The Statistical Analyzer

Bjarni Gunnar Bjarnason

Supervisor: Timothy William Perkin

Faculty of Information Technology,
University of Akureyri.

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature ______________________

Date _____/_____/_____

Abstract:

This dissertation outlines the work made while doing this final year project. This is a web-based expert system which looks for patterns of faults in a golfer's statistics. A Standardized Statistical Form was made to have the inputted statistics in the same format. A database with the ability to take in all necessary information for the statistical form and the recommendation inside were made. The system scans the golfer's statistics to spot errors in his game and gives him recommendations on how to improve.

Table of contents

Table of Figures

1 Introduction

1.1 Motivation of the work

The motivation of the work was to give golfers another way to improve their golf game.

1.2 Description of the problem – Golfers want to improve their golf game

The problem is that golfers want to improve their game. It is in a nature of the golfer to want to improve his/her game. A golfer can improve in many different areas in golf. Of course like in any sport there is the mental aspect, the physical aspect and the technical aspect. This project is going to try to help with the technical and if possible the mental aspects of the game. In the mental area things to look for in the statistics are things like, does the golfer always have a bad drive after a good putt or a good hole. In the physical area the things to think about are stamina, muscle strength and over all physical shape. In the technical area there is a lot to think about. It is not without reason that golf is called the most difficult technically sport in the world. In the technical area you have the swing, short game and putts. To give an example of how many elements there is to cover in the technical area. The swing can break down to posture, foot placement, club placement, backswing and etc. The short game is every shot within 100 meters from the green, e.g. sand shot and a flop shot which is one of many kind of chips. There are equally many things to consider for putts (play on the green).

1.3 How can golfer improve their game

There are a few ways that golfer can improve his game. I am going to state four ways to improve once golf game, however there are many more. The first way would for the golfer to hire a Golf Teacher to help him. The second way would be to buy some golf videos or golf books. The third way would be to take their statistics themselves. Finally the fourth way would to use an expert system which would analyze the golfer's statistics.

1.3.1 Golf Teacher

The golfer could hire golf pro to help him with improving his game. That is probably the best way to improve his golf game. The golf teacher can help in any level of the game. He can give exercises for the physical and the mental part. His specialty is of course the technical part where he can spot errors just by looking at a couple of swings. The downfall of going to a golf teacher is that you probably have to go to him regularly for a few times so he can monitor your development and they tend to be very expensive.

1.3.2 Buying books and videos

The golfer could buy himself a golf book or a golf video to help improve his game. This is not an expensive way of trying to improve your golf game. Golf books and golf videos are usually on a fair price. They are most of the time written or made by famous teachers or professionals. Both the books and the videos commonly have good exercises and explanations. The downfall to buying books or videos is that they do not spot errors for the golfer. That means the golfer himself has to find his errors first to able to fully use these kind of approach to improve his game. Usually, golfers are very blind on their errors. That is why professionals have a teacher on their payroll all the time.

1.3.3 Taking golf statistics and analyze on their own

The golfer could take the golf statistic down and analyze it himself. That is something that would stop some golfers in taking their statistics themselves. The simplest way to store this information would be using Microsoft Excel or another similar program. This would not cost anything except the cost of the Excel. The cost would be none if the golfer just would use pen and paper. The downfall of the golfer taking his statistics for himself would be that it would mean that he would have to do all the calculation of the all the form and at least the average statistics. If the golfer would have to know how to use the Microsoft Excel or any another similar program to calculate and store his data. If just stored on paper it would be a lot of paper work after each round. They would also have to know what the statistics how to read from it. Also how the information read from the statistics could be used to improve their game.

1.3.4 Web-based Expert system which analysis the golfers statistics

The fourth solution is an expert system which would analyze the golfer's statistics. In this system the golfer would start by inputting in his statistics from his golf rounds. When he has put in some rounds he would let the expert system analyze his statistics. It would look for patterns in his statistics e.g. if he is driving to often to the right of the fairway or if he is missing to often putts under 1 meter from the hole. When the expert system has finished scanning the golfer's statistics it will give him some tips on how to improve his golf game. The tips given are got from the patterns found by the expert system.

1.4 Chosen solution

I have chosen to create an expert system for this project as it is a new and original concept that I believe will be of great assistance in helping golfers improve their game. Capturing the knowledge of the experts in the game of golf to aid golfer to improve is a worthy project. One other reason is that the other solution gives me little opportunities to make a good project.

2 Golf Statistics

Golfers tend to record their statistics for each round they play. Recording the statistics is not very hard and every golfer can do it. Unfortunately the majority of the golfers that record their statistics today are professionals or golfers with handicap under 6. This project wants to try to change that. The statistics can be taken with different requirements. Everywhere from just taken down score of each hole to length of each shot played on each round.

2.1 How golfer record statistics

Golfers are divided into two groups when remembering the details of their last round(s) of golf. A golfer either remembers everything he (and maybe his "team mates") did on the last round(s) of golf or he has to write down after each hole how he played the hole. While golfers are playing a round of golf they are usually writing down their own score and most probably the score of one of their "team mates" if playing in a tournament. It is taught that a golfer should write down the score after each hole. Writing down the statistics is the same. A golfer either writes it while he is playing, after he has finished a round at the clubhouse while it is still fresh in his mind or just when after he comes home. If the golfer writes down his statistics while playing he either has an extra piece of paper from home, another scorecard that he uses to write in or he write in some of the boxes in the one he is using. There is no standard on how golfers record their statistics today.

2.2 What is record in statistics

Statistics are recorded with different kind of details. The golfer is the one which decides how much detail he wants to record. In my opinion what can be called recording a statistic in golf is when a golfer record his score, hit fairways in drives, hit green in regulation (GIR) and putts on each hole. Just to take an example of how detailed the maximum statistical recording can be is e.g. record length of shot on the round in meters and with what club they were made and where on the club face the ball hit in each shot. The recording detail of statistics can vary much between golfers. In Appendix B I have made two statistical recording forms and information what each field in them stands for. It will enhance your understanding on the statistics. Please look at them for more detail on this section.

2.3 Analysis of the statistics

Analysis of the statistics today is usually just the average of field in the statistics like shown in Appendix A. What golfers are looking for some repeated patterns that occur while they are playing or if their total average scores of those fields that have scores are above some standards they have set or are set by their golf teacher. This project is going to use a pattern matcher to find these patterns. Examples of a patterns, golfers are looking for, are drives are often going to the left of the fairway in their drives and they never one putt after we have been in the bunker. From this kind of information the golfer can find out this error and improve his game. Just these two examples can point more than a dozen things that need improvement. But it is a fact that majority of the golfers would not be able to spot all the patterns that can be found. His is one of the things that this project is going to try to help with.

2.4 The reason golfers take statistics

The reason that golfers take statistics is to spot their weaknesses but that requires that they have to spot them themselves. Very good amateur golfers and professionals use statistics to spot their weaknesses as said above and they often have the ability to see it out for themselves or have a teacher that will spot them out for them.

3 Description of the work

In this section are two things that are going to be discussed. First what is this project is going to achieve and second how it is suppose to function.

3.1 What is this project going to achieve

This project will do an AI approach on pattern searching golf statistics. This approach will search for repeated patterns in the golf statistics which the golfer inputs. These patterns will then have different kind of tips for the golfer. The golfer can then pick one or all of these recommendations and then he would get ways on how to fix that error. This will help golfers to improve by finding the areas in the technical part of the game they need to improve. The pattern search is the most important part of this project and what is most important to be able to implement it. The second most important stuff is to be able to make this a web-based system. The reasons for that are: no install from upcoming users, updating the system will be easier and accessibility would be good.

3.1.1 Database of statistics

This will approach by making a fully functional database with statistics that will input by the designer. These statistics will all the patterns that the pattern search will be able to find. This database will hopefully have every pattern a golfer can think about from the drive on the tee and to when the ball finally goes into the hole.

3.1.2 Database of recommendations

Database of recommendations will be a database that will be full of recommendations how the golfer can improve his game. This database will contain at maximum 10 tips for each pattern the user can get when his statistics are analyzed. This database will grow larger as more patterns in the pattern search will appear. In shortage, if no pattern will be implemented then his database will probably not be made. Here will the AI pattern search come in and connect the recommendations with the inputted statistics. When the search finds a pattern it will trigger the recommendations that are associated with that pattern. The recommendations will be stored in the database as name of each tip in one field in database and then the text attached to that tip in another field. These recommendations will hopefully come from ex-professional, the head pro at the local golf club, from books and finally from the internet.

3.1.3 Pattern search/Statistical Analyzer

The pattern search is the main object of this project. Like I see the search, it will go through the data it is represented with and it will look for some patterns that will trigger the search to say, "Hello, partner there is a repeated occurrence in your game. Here are up to 10 reason why this can be happening." Under these reasons are then up to 10 tips how to fix this error and improve the golfer's game. Another approach is to give him all the reason with tips under each right away. The different is that the golf does not have to pick a reason himself. It could be done that golfers with handicap under 6 will get just the reason and can pick put the others just get it all straight away. The reason for that is that golfers with handicap under 6 have more knowledge about the technical parts of the game.

3.1.4 Standardized Statistical Form

I will need to make a minimum standardizes statistical form. The reason I say that is because even though in a complete web site the golfer can use input the most minimum statistics, it does not mean that it can be used to search for patterns. Like in Appendix A the minimum amount of statistics can not be searched for patterns that will in any good way give the golfer tips about what needs to improved.

3.1.5 Make a website with all the functionalities to take in statistics (If time allows it)

I will approach this by making a web site that has the functionality that will be enough so that the user will be able to input his statistics so that I can run and test the pattern search. The functionalities that the golfer needs to input his data are into the database are: Standardizes statistical forms, Golf courses, the ability to add golf course, the ability to log on to the web site with its own login and password, and finally, which is pretty important, he needs his statistics from the last play round.

3.1.6 Allow the golfer to compare himself with other golfers in the database (If time allows it)

The web site will also have enough functionality so the golfer can compare to other golfers in the database. He will be able to compare himself with everyone in the database, with e.g. players with 5-9 in handicap and etc. Say that the golfer has 9 in handicap and he compares himself with golfer with 4-8 in handicap he will be able to see in what areas he needs improve in the comparison with this group of golfers. Here is also the possibility to have a pattern search on the comparison.

3.2 How it is meant to function

This chapter has two use cases on how the functionality of the system should be. The first one is a use case on how the system is meant to function if just the database of statistics and recommendations, the Standardized Statistical Form, the Statistical Analyzer and the ability to log on to the system will be ready in time. The second use describes the functionality if all the functionalities stated above will be ready.

3.2.1 Use case 1

The system should function like this. A golfer will be able to register to the system and chose a login name and a password. The golfer logs onto the system with a user name and a password. When inside the system he will be able to add a new golf course and all the information needed with a golf course. He will be able to add a new round that he would then use when he is inputting the statistics for that particular round. He will be able to input his statistics into one standard form and let the system calculate his average statistics. Finally and the most important part he will be able to let the system analyze his average statistics and give him basic recommendations.

3.2.2 Use case 2

A golfer has finished his round in a tournament or just playing with his buddies. He goes home and sits down in front of the computer with his statistics from the round he just played. He logs onto the system with his user name and his password or if this is his first time he registers himself and then logs in. The course he played in already in the system, so he goes and inputs the name of his round into the system. If his course is not in the system he can add him in. Now he is set to input his statistics into the systems. He picks the course he played, the tees the played from and the name of the round. Next he has to pick between several different kinds of input forms which vary in their level of detail. He picks one and starts to input his statistics into the input form he chose. When finished he inputs the statistics into the system. Then he can go and let the system calculate his average statistics in several different details. Now if he have put in some rounds into the system he can go and let the system analyze his average statistics to give him recommendations on how to improve his game. He will get 10 reasons on what could be wrong and under each reason are at least 5 practises on how to fix that error. When he has found out what he needs to improve he wants to compare himself with a player with a better handicap and him to see where the difference is between them. This best friend also uses this system and he wants to compare their statistics together and does that. Next he goes and looks for a new driver because he is on a web site where everything can be found related to golf.

4 Related Work

In this section will be talked about several things. First there is going to be a talk about what is new is my approach to this project. Secondly there will be an introduction what expert systems are and how they work. Thirdly there will be talked about one existing expert system. Finally there will be talked about existing golf system.

4.1 What is new about my approach to the project

In the field of golf statistics this project has one new thing. In no other system found during the research for this project have I found a system that gives back to the golfers recommendations/suggestions based on their inputted statistics that can help them improve their game. There are systems out there that take in the statistics and calculate an average. Also there are systems/sites that give golfers some advice about how to improve their game but none these have the ability to connect these things together.

4.2 Expert System

What is an expert system? To answer that question lets start of with a definition of an expert system.

> *"Computer software that can solve a narrowly defined set of problems using information and reasoning techniques normally associated with a human expert. A computer system that performs at or near the level of a human expert in a particular field of endeavour."* **[13]**

An expert system is software that takes in knowledge, usually, from a human expert in some field and tries to be a replacement for that expert. It can be used both to solve a problem and/or for giving advice. This is a good way to store human expertise. The reason for that is that a computer does not forget and a computer is consistent. Expert systems are also known as Knowledge-based systems.

A simple explanation on how expert system works. Expert system takes in knowledge and input it into a knowledge base as facts. These facts are always true, they are the truth. Then it applies rule to the knowledge to pick out the information it wants from the knowledge base. The rules are build up from IF THEN relationships. All facts in the IF have to be true to the THEN to execute. An example IF you hit your golf ball into a middle of a deep hazard THEN you will get a penalty.

There are different types of expert systems like rule-based, fuzzy logic and neutral network. The difference between is how the deal with the reasoning.

4.2.1 Rule-based system

One kind of expert system is rule-based system. Here is a definition of a rule-based system.

> *"A rule-based system is a system that uses rules to derive conclusions from premises: Given the gum-chewing rule and the premise that you are in school, you*

(as an advanced kind of rule-based system yourself might conclude that it's time to spit out your gum."
[12, page 17]

A rule-based system consists of four main parts, an inference engine, a working memory, a rule base and an execution engine.

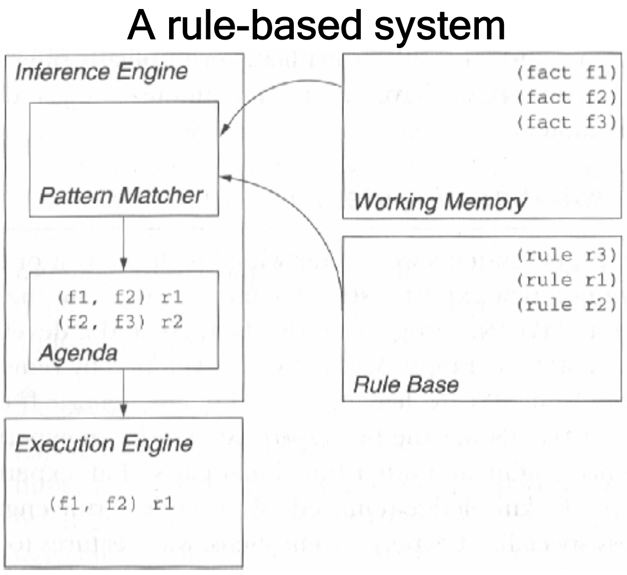


Figure 1: A rule-based system **[12, page 20]**

The inference engine is the part inside the system that compares the rules from the rule base with the facts from the working memory. The inference engine has two parts, a pattern matcher and an agenda. The pattern matcher is the mechanism that compares the facts with the rules and decided which rules are allowed to execute. The agenda decides if there are more the rules that can execute which of them should be executed first. Working memory is another name for a knowledge base. Working memory is where the facts are stored. Rule base is where the rules are stored. **[12]**
A rule is built up very similar to IF THEN like in any programming language. A rule has two sides, a left hand side (LHS) and a right hand side (RHS). The LHS is like the IF and includes one or more facts that need to be true. The RHS is like the THEN and includes what should be done if the LHS is true.
Rule-based programming is a declarative programming. By declarative programming is meant that the programmer describes what the computer should do but also gives it a lot of rules on how to do it. **[12]**
The reason rule-based systems are covered specially is because Java Expert System Shell or JESS which will used in the system is a rule-based system.

4.3 Existing Expert System

In this section I am going to talk about existing expert system. No similar expert system to the one in this project was found on the internet in a massive search there. Most of the recommendation expert systems found on the internet where recommendation system involving e-commerce. These systems were most of them design to analyze online buyers. What they were going to buy, what they would be likely to buy and etc.

4.3.1 Feature-based recommendations for one-to-one marketing

The reason this expert system was chosen as an existing expert system is because it is a recommendation system. What this recommendation system is meant to do is to give recommendations to customers on the internet on what they should buy depending on their former internet purchasing. This is going to be achieved by one-to-one personalized marketing and by very accurate recommendation system that should be able to give a customer recommendation on a product. This accurate recommendation system will be able with enough information about the customer will be able to pick sizes, colour, brands and etc. for the every customers in the system.
This recommendation system influenced this project in the way it was going to handle its customers. Not that system is going to implement one-to-one marketing but it is hopefully going to have every customer unique. It will handle every customer like it is its only customer. Hopefully golfers using this system will feel the same thing. Also hopefully in the future will this system be able tell the golfer what kind of practises will suit him the best and with more information maybe he best size of clubs like the recommendation system is going to pick sizes and brands for its customer. **[10]**

4.4 Other golf existing systems

This section covers the existing systems I have been looking at. I was mostly looking at web sites that offered golf statistics for free. There was one that had pretty good statistics and a couple that were okay. However, all the sites I have found have been missing is the AI part I am aiming to do in this project. I still haven't found a web site/expert system that uses pattern search in golf statistics to help the golfer to improve.
The best web site I have found this far is home site of the GameTrack system which is built to try to improve the golfer game. It has reasonable statistics, it lacks a lot but can be used. Then it offers an analysis of the statistics. The analysis they offer are course analysis that is just the average of the golfers statistics on each course for itself. Also it offered something called the VirtualPro. All this VirtualPro does is to have tips on fields in the technical part of golf. The tips that they offer are not connected with the statistics in anyway, like what I want to do in my project. They don't identify the golfer's weaknesses and gives him tips on how to improve his game. They are general tips on backswing, putts, grip and sand shots. A golf teacher provides these tips. It analysis of the game is not very helpful for the golfer in the sense that he has to find all his weaknesses himself from the statistics. The problem with this site is that it doesn't let you do everything for free. There are many features that you will have to pay for to be able to use. Its statistics is not very detailed and like I said earlier it lacks a lot fields in the statistics that are needed to make good. The tips from the VirtualPro are not connected to the statistics of each golfer. They are just there for each golfer to look at and see if he/she can use any of them. So, if this is what is called analysis then what I want to do will probably called SuperHuman VirtualPro **[1]**.
Another web site I found was www.goforgolf.com. Their statistics are not very detailed but it could probably used by some part of the golf community. It's analysis of the game is poor and only feedback of the statistics with the total or the average. It advanced

summary should be called a minimum summary, there is just nothing advanced about it. It has no help for the golfer. It offers no real tips on how to improve your game **[7]**.
Third and last web site that is worth mentioning is www.egolfcard.com. This site has minimum statistics and no analyzes of the game. There is just nothing more to say about this site **[4]**.
What all those web sites are missed is the part the project want to accomplish in this final year project and that is the pattern search.

4.5 The reason this is an expert system

There has to be a reason for why this could be called an expert system. This is one explanation of what can be called an expert system.

> *"Often the rules represent the heuristic knowledge of a human expert in some domain, and the knowledge base represents the state of an evolving situation (perhaps an interview or an emergency). In this case, the rules are said to constitute an expert system."* **[12, page 37]**

In the case of this system the domain is golf. The human expert or domain expert has to be an expert of golf. The reason this can be called an expert system is because the rules in this system constitute the knowledge of an expert in this domain. The expert of this system and the domain of golf is a former golf teacher and a national player of Iceland. His name is Birgir Haraldsson.

5 Design

In this section will be talked about how system was designed. It will be split into four chapters. First is the design of the input. Second is the design of the database. Thirdly is the design of the Statistical Analyze. Finally is the design of the recommendations.

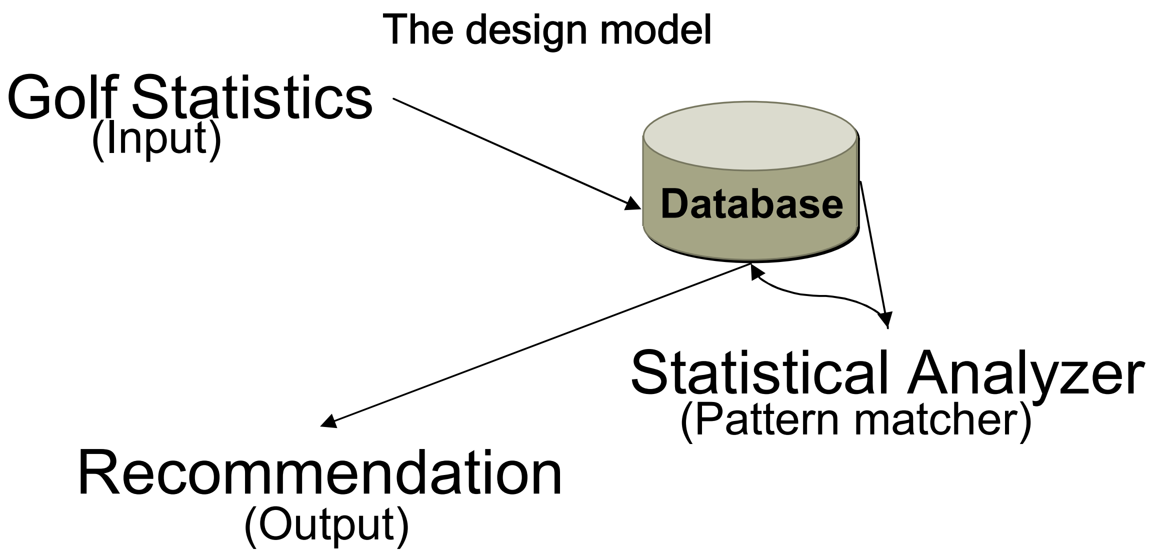


Figure 2: The design model

This figure shows the design model of the system. First the golfer inputs his statistics and they are stored in the database. The Statistical Analyzer then analyzes in the inputted statistics looking for pattern of faults. When the Statistical Analyzer is finished he gives out recommendations on the faults the found. These recommendations, which are stored in the database, are then displayed for the golfer to use to help him improve his game.

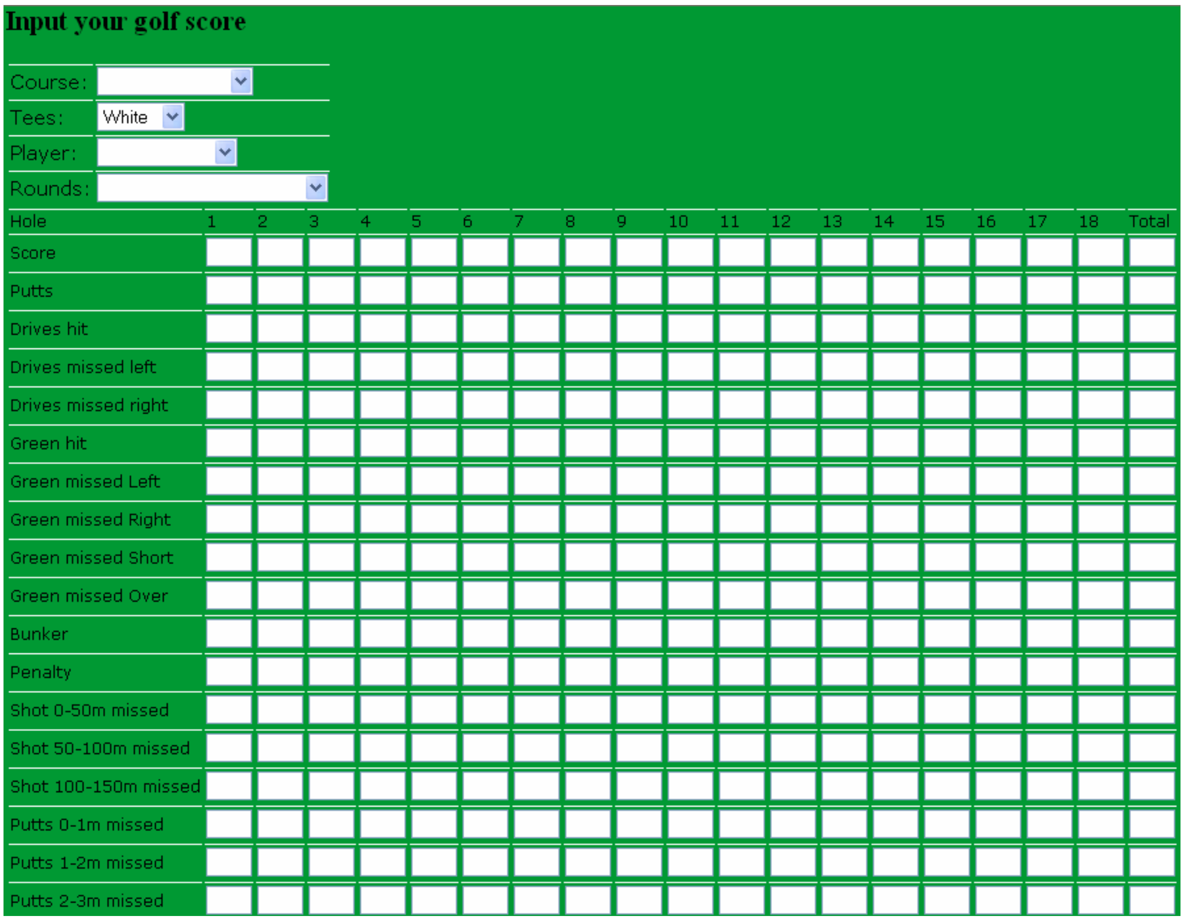
5.1 Input/Golf Statistics

The inputs into this system are golf statistics and information about the player, golf courses and golf rounds. At this point the system only has one statistical form. That form is the Standardized Statistical Form which next chapter is all about.

5.1.1 Standardized Statistical Form

When this project was started it was obvious it would need some Standardized Statistical Form(s) that the project was going to work from in making the system. The outcome of these thought were to make a statistical form that would take in the minimal amount of information needed for the pattern matcher/Statistical Analyzer to work on. The things that had to be thought about were in what detail the data needed to be so it could be able to analyze and recommendations given out. What I decided was to talk to a former golf teacher and have his opinion on what the minimal amount of information were needed in the statistical form. The outcome of our discussion was the form here below. This is what could be referred to as the Standardized Statistical Form.

The Standardized Statistical Form



Input your golf score

Course:	
Tees:	White
Player:	
Rounds:	

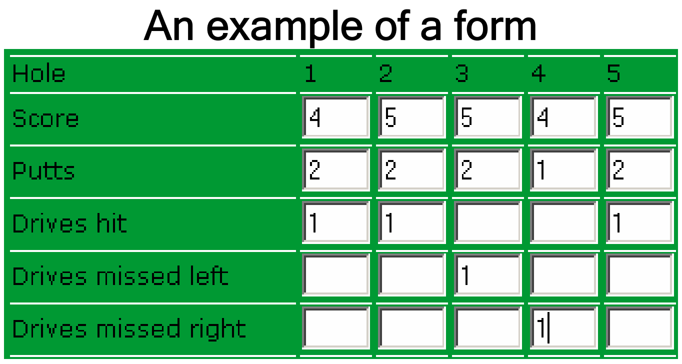
Hole	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Total
Score																			
Putts																			
Drives hit																			
Drives missed left																			
Drives missed right																			
Green hit																			
Green missed Left																			
Green missed Right																			
Green missed Short																			
Green missed Over																			
Bunker																			
Penalty																			
Shot 0-50m missed																			
Shot 50-100m missed																			
Shot 100-150m missed																			
Putts 0-1m missed																			
Putts 1-2m missed																			
Putts 2-3m missed																			

Figure 3: The Standardized Statistical Form

As can be seen when looking over the form most of its fields are self-explaining but nevertheless it is better to walk you through so there will be no misunderstanding.
First four elements are course, tees, player, rounds. In course the player just picks the course he played the round he is going to enter the statistics for. In tees he picks the tees he played this round on. Then he would pick the player which would be him in this case. Then he would pick the name of the round this statistics came from. All these information would have to be inputted and stored into the system before the player would use input his statistics.
Now to it would good to explain why these information's of the statistics were chosen over others. First thing I and the golf teacher did was to try to cover the all the major part of the golf game. They are Drives, Putts, Pitches and Shots. Drives are the first shot on each par 4 and par 5 holes. Putts are usually the last shots on each hole. Pitches are shot that are made from less than 100 meters from the green. With more detailed statistical form this shots can be broken more down. Finally there are the shots. Shots are all shot from more than 100 meters from the green. In the form above we have only shots from 100-150 meters from the green to compare to that group but that is fine.

5.1.2 Filling out the form

Filling out the form is straight forward. In the score and putts boxes the golfer inputs the number strokes and putts made on each hole. In the rest of the form you just put 1 if applicable or just leave empty if not applicable. To explain better what I mean I am going to give an example. Say if the golfer is on the 4th hole and he drives to the right of the fairway in the drive then he wound put 1 in the box drive missed right as shown in picture below.



Hole	1	2	3	4	5
Score	4	5	5	4	5
Putts	2	2	2	1	2
Drives hit	1	1			1
Drives missed left			1		
Drives missed right				1	

Figure 4: An example of a form

Here you can see that the golfer hit the fairway in the drive on holes 1, 2 and 5. Missed the drive left of the fairway on hole 3 and missed the drive right of the fairway on hole 4. This is the same for the rest of the form.

5.1.3 The data requirements for the input

The data requirements are golf statistics from a golf round. The minimal amount of statistical data the Standardized Statistical Form. So, the data requirements are all the information needed to fill out the Standardized Statistical Form.
Some more data could be needed by the golfer, e.g. if the course he played is not in the system he has to get the needed information for that course. These information's involve e.g. length of each hole, par of each hole, handicap of each hole and the course rating. These information can all be found on a scorecard from that particular golf course or if it has a web site it can be found their.

5.1.4 Interface

The interface for this system is made in PHP, which stands for Hypertext Pre-processor and HTML, which stands for HyperText Markup Language. PHP is a widely used scripting language which is well suited for web development.
The reason PHP is used in the system is because of its ease of use and very good documentation. It can standalone, be embedded in HTML, to execute programming language like Java and much more. The main reason is though its connection to MySQL database which is used to store the data in this system. PHP has a whole section of function to access and manipulate data from MySQL. (www.php.net)
The reason HTML is used in this system is because how easy it is to use it to make web sites. It was decided to make the HTML in this system a XHTML standard. XHTML

stands for EXtensible HyperText Markup Language. XHTML is a cleaner and a stricter version of HTML.
When designing this system the decision where made to use the interfaces so that the user could input the data needed for the Statistical Analyzer to work and to display the output of the Statistical Analyzer but only make them pretty if time would allow.

5.1.4.1 Apache

The use of PHP, in this case, means the use of the Apache HTTP/web server. The Apache server is open source software meaning it is free for everyone to use. The reason Apache is used in this system is because it can interpret PHP and because it is free.

5.2 Database

The database stores both all the information needed for the system to analyze a player statistics and the recommendations that the system give to the golfer after the system has analyzed his statistics. The design of this database took a lot of time. It needed to be good and precise to make it easier to implement the database later.

5.2.1 Database requirements

When designing the database there were a lot of things that needed to be thought about. There would have to be some information on the golfer. There would also have to be information about the golf courses and something that would uniquely identified different rounds the golfer played. Every golfer had to have a unique identity. Right from the start it was known that the table that would store the statistics would be the biggest table in the database.
The information about a golf course was split into three different tables. That decision was made because of three things. First reason was that it would to have some general information about the golf course. Second reason was that each hole on a golf course needed to have several information's about itself. The third and last reason was that golf course can have more than one kind of tees on it, most golf courses have four.
To be able to design the table that would store the information about the statistics the Standardized Statistical Form had to be design first because the statistic table would be identical with that form. It would take in all the information from the form and store it. The decision in making that table lied with the decision made when the Standardized Statistical Form were made.
There would need to be a table which stores the recommendations which are given to the golfer after his statistics has been analyzed.

5.2.2 The design structure of the database

The database will probable consists of 8 tables. Seven of these tables are used for the input. They will be in an alphabetic order: avg_stats, course, golfrounds, holes, player, stats and tees. These names are self explaining on what information will be stored in them and will be in more detail in the implementation chapter. Then the recommendation table is the table which stores the recommendations. The seven tables taking in the data from the input will be connected together in some way but the recommendation will always stand alone. This means that this will be relational database. The database will probably look something like this.

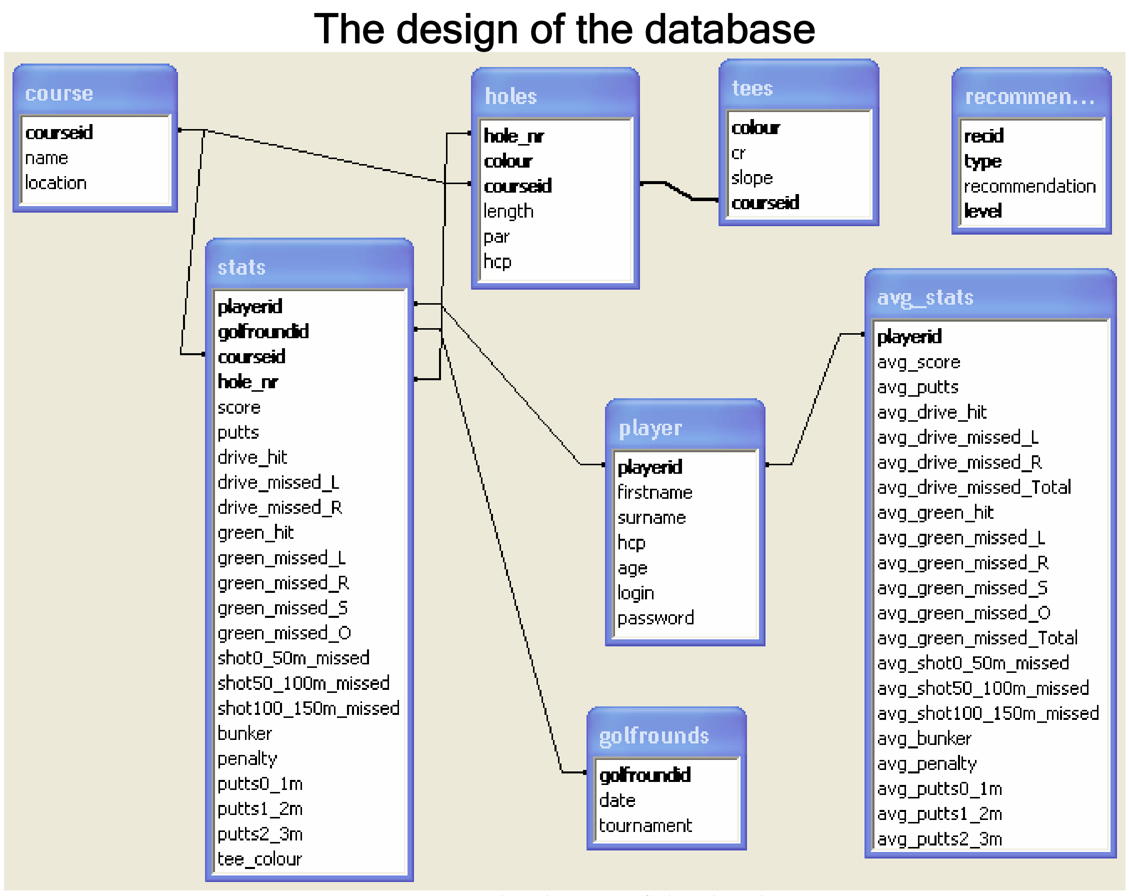


Figure 5: The design of the database

This table shows the design of the database with it fields and how the tables connect together making this a relational database. The bolded words are the primary key(s) of each table. The lines between the tables represent the connections between the tables. The only standalone table is the recommendation table.

5.2.3 Relational database

A relational database is a database were some or all the tables in it are connected together in some way. A definition of a relational database is:

> *"A relational database is a collection of data items organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables."* **[9]**

This database will become a relational database because for that only reason that the information needed in the stats table come from many different tables.

5.2.4 MySQL Database

In this system MySQL database server will be used to store the data. The MySQL is open source database and is free for all but very large commercial companies and the license costs 440 euros a year **[8]**. This system will never become a very large commercial company so it is free of charge. The reason MySQL is used in this system because it is pretty fast, has a good admin tool, has good documentation, is open source and previous experience of the designer. The main reason is though because of its connection with PHP.

5.3 Pattern Matcher/Statistical Analyzer

The statistical analyzer or the pattern matcher is the heart of this system. This is the mechanism that analyses the golfer's golf statistics and gives him recommendation on how he could improve his game. First will be discussed what expert system was chosen to be used inside the analyzer. Then will be discussions how the analyzer should work and its design. Finally the design of the facts and rules will be introduced.

5.3.1 Introduction to JESS

JESS stands for Java Expert System Shell. JESS is a ruled based system. JESS can be used both as rule engine and also as general-purpose programming language. Because it was written in Java it can access all the Java's powerful API's such as networking and database access. JESS can be used from a command line, a GUI or be embedded. This gives JESS a great flexibility. **[12]**

5.3.1.1 Little bit of a history

JESS was developed at Sandia National Laboratories and was first released in 1995. It is based on CLIPS technology but is in reality quite different from CLIPS. CLIPS was used as an insight for the group which wrote JESS. It is free for non-commercial use and commercial use inside the United States of America (USA) but is negotiable for commercial use outside the USA. **[12]**

5.3.1.2 JESS usage with or without Java

JESS can be in used many different way. To explain this better this text from the book JESS in action gives a good idea.

> *"Jess is a powerful environment for processing rules and scripting the Java platform. You can use it in a wide range of applications, built purely using the Jess rule language, purely in Java, or with some mixture of the two."*
> **[12, page 39]**

As this text explains JESS can be used in many different ways. It can vary from standing alone to having some Java code to being manipulated inside a Java code. This is one of things that give JESS its great flexibility. **[12]**

5.3.1.3 The reason for choosing

There were four reasons why JESS was chosen over other expert systems/rule-based systems to be used in the project. The first reason was its ability to work with Java and its API's. The second reason was because JESS has this relationship with Java API's it can be used in both Java servlets and applets if needed. The third reason was that by having inside a Java class I could let PHP run the program if I wanted to take that approach. The fourth and last reason I chose JESS was because of my former personal experience with Java.

5.3.1.4 How JESS was used

In this project JESS embedded in a Java code. The system creates a new engine inside Java. This engine has its own working memory, rules and functions just like a engine in JESS. Into that it inputs the facts the knowledge base and makes the rules needed. It also runs the engine. It then stores the output that is then retrieved by Java after the engine has stopped.

5.3.2 Alternatives

5.3.2.1 CLIPS

CLIPS was first released in 1986 from NASA and it was developed to be a tool to make the development of software to model human expertise easier. It is an expert tool that has a user-friendly programming environment. It is free for non-commercial use but is negotiable for commercial use. It can integrate languages like C, C++ and Ada. CLIPS is older and that makes it have advantages over JESS in user experience and more documentation which could be useful. It is more used in basic applications. In web based application JESS rules because of its connection to Java and its Object Oriented (OO) possibilities **[5].**

5.3.2.2 LISP and Prolog

The two languages are LISP which stands for LIST Processing and PROLOG which stands for Programming in logic. These are similar languages in AI. LISP is much more used in the USA as for PROLOG is much more used in Europe. Speed: LISP is better because the speed of PROLOG depends on the size of the vocabulary. Backtracking: PROLOG has the advantage because it has it build in but LISP does not have build in but it can be build. Analyses: It comes naturally to LISP to produce analyses. Structures: It comes naturally to PROLOG to build structures but LISP doesn't build structures.
PROLOG has the advantage in comprehensibility and readability because its syntaxes grammarlike structure **[2]**.

5.3.3 Briefly how the analyzer should work

The analyzer takes in average statistic for a player and puts them into its knowledge base as facts. Then the rules, which are the pattern which look for a match in the fact, are also inputted into JESS. These rules are hard coded into the system. The user can neither change these rules nor add in a new rule. Only the administrator of the system can do that. On runtime the rules scan through facts looking for a match. When a rule is finds a match it stores its recommendation id in a value. This value is then retrieved by Java which uses it to give the player the recommendation for the recommendation id.

5.3.4 Design of the analyzer

The analyzer or pattern searcher will be design in Java with JESS embedded where the pattern matching will take place. It will take in the playerid when executed. The reason for that is so the average statistics which will be inputted as facts into the system will come from the right golfer. The playerid will ensure that. The recommendation will have to be retrieved from the pattern matching so that the recommendations will something to output.

5.3.4.1 Fact design

There were couple of ways to design the fact into the system. One was to make a template in JESS called deftemplate, which would take in the facts from the average statistics. A deftemplate would be structured like this.

```
(deftemplate avg_stats
      (slot avg_score)
      (slot avg_putts)
      (slot avg_drive_hit)….)
```

Then the information would go into slot in the template. Each template can be used to make many alike facts to put into the knowledge base. This could take in more than one golfer at a time.
The other way was to just put the average statistics straight in as facts using no template. Just using the assert function in JESS that inputs fact into the knowledge base. The reason that is was chosen for this system is that there is only one golfer analyzing his average statistics at a time not many. The golfer has just one set of average statistics not many.

5.3.4.2 Rule design

When designing the rules it came clear that this could become the largest part of the system. One can almost make an ocean of rules but the designer think it is more useful to have three that works then many that does not work. The reason for that was the time factor of the project. The objective was to make rules that would find patterns of faults in the inputted statistics. The rule could be structure in that way that when the average statistics goes over a predefined number it will fired and a recommendation given how to improve that. When a rule fires it execute the commands on the RHS. Here is an example of a rule, where 6.5 and 5 are the predefined numbers.

```
(defrule drives-right
    (avg_drive_hit ?drvhit &:(< ?drvhit 6.5))
    (avg_drive_missed_R ?drvright &:(> ?drvright 5))
=>
    (store Drives (+ 0 1)) )
```

This is an example of the rule in JESS. The name of the rule is drives-right. The facts the have to be match for this rule to fire are the average drive hit has to be fewer than 6.5 on average and average drive missed to the right is over 5 fairways on average. Then the recommendation id 1 will be stored in Drives.
Each rule was designed to have its own id which is stored along with the type. The reason for that will be covered in the recommendation design part.
It was also decided to not give the golfer the ability to change or add rules in the system. That would be in the hands of the administrators of the system. But that is something that could be design in next versions. The reason for that is that the recommendation ids given inside here have to be the same as in the recommendation table for the system to give out the right recommendation. The golfer hasn't got access to the place where the rules are made only the administrator or designer.

5.3.5 The data requirements for the analyzer

As briefly mention in the section above the analyzer takes in average statistics for a player. These average statistics are found in the average statistical tables in the database. The data found in that table is as the name suggests an average of the statistics that the user has inputted into the system. The reason the data requirements are the average statistics but not just the stats for each round is that the average statistics sum together the different elements of the statistics into one place. This makes it easier for the rules to work with the data.

5.4 Recommendations

The system will have to be able to give recommendations in different detail to golfer with different ability. The reason for that is that you can give a player with a low handicap recommendation in more technical detail then a player with high handicap. The most obvious way is to use the golfer handicap.
The recommendations are got from values that the statistical analyzer stores when looking for patterns. These values are then used to query the recommendation tables in the database. The outcome is then outputted for the golfer to use to help him improve his golf game.
The recommendations themselves are inputted by the administrator of the system. The recommendation are made by the administrator and reviewed by a golf teacher. The recommendations are both practises and technical details.
The design of the recommendation was made with a former golf teacher. The recommendations made has to able connect to the right rule in the Statistical Analyzer. That is why both each rule and recommendation was given a unique number. The number of the rule and recommendation that it was going to give would have to be the same so they could match when the analyzer needed an outcome after a run.
This is one of the reasons why the golfer can not input rules and recommendations himself because these numbers have to match for the analyzer to give back the right recommendation.

6 Implementation

In this section will be talked about the implementation of this system. It will be split into four chapters like design section. First is the implementation of the input. Second is the implementation of the database. Thirdly is the implementation of the Statistical Analyze. Finally is the implementation of the recommendations.

6.1 Input/Golf Statistics

The implementation of the input was a big and constantly changing process. The way the implementation will be presented is as follows. First I am going to talk about common factors in the interface and input files. Then the heading of the chapters below will be in the names of the interface files, in alphabetic order. Inside these chapters will be both talked the file in the header and the file it sends its information to. The input system is built up in that way that all of these interfaces are made of HTML code and they are using the form method in HTML to POST the inputted data. Then PHP takes these information and input them into the database or uses them in any other way.

6.1.1 How the form method=”post” works

The form is the method in HTML used to take in and send information a user inputs or picks from a box to another file. That file then inputs these information input into the database, which is the case with this system. This is like this for almost all interfaces in the system. For an example of a form method is the form method code from course.php.

```
<form method="post" action="course_entry.php">
    <table>
    <tr>
      <td><span>Name:</span></td>
      <td><input type="text" size="30" name="name"/></td>
    </tr>
    <tr>
      <td><span>Location:</span></td>
      <td><input type="text" size="30"
name="location"/></td>
    </tr>
    <table>
    <input type="submit" name="submit" value="Input"/>
    <input type="reset" name="reset" value="Clear"/>
```

The form starts at the top by declare what method to use and what action to take. In this system the method is always post and the action is always an input file. In this case it is course_entry.php. Inside this form is a table to make the form look “more even and more standard”. Then if we cut to the important part then next is the input tag. The input can be in the form of a text box, buttons, checkboxes, submit and more. In this system is mainly used text box which are type text. Also used a lot in this system are combo box which do not have the tag input but select. Back to the example as can be seen there the input is of type=”text” size=”30” name=”name”. Type text stands for text box, size stands for the size of the textbox and name stands for the name of this input. The bottom two inputs of the type submit and reset are represented as buttons. The type submit sends in information inputted in the form and type reset resets the form to original state. When the submit button the press the information are send or posted to the file declared in the action, in this case course_entry.php. Below you can see the course.php interface.

6.1.2 How PHP takes the information and sends them to the database

The information submitted by a form are sent to a PHP file. This is the case for almost all the entry files in the system unless otherwise stated in the chapters about the entry files. Lets start with a seeing the code of the course_entry.php

```
<?php
include("db_connect.php");
$name = $_POST['name'];
$location = $_POST['location'];

$sql  =  "INSERT  INTO  course  (name,  location)  VALUES
('$name', '$location')";

$login_sql_check = mysql_query("SELECT * FROM course WHERE
name='$name'");
      $login_check = mysql_num_rows($login_sql_check);

if($login_check == 0)
{
      if ($name != "" && $location != "")
      {
            $query = $sql;

            mysql_query($query) or
                        die (mysql_error());

      echo  "<p>The  course  has  been  stored  in  the
database</p>";
      include("tees.php");
      }
      else
      {
            echo "<p>All boxes have to be filled out</p>";
            include("course.php");
      }
}
else
{
      echo "<p>The course name already exists</p>";
      include("course.php");
}
?>
```

The PHP text is always within these tags <? and ?>. This file starts to get connection to the database. That is done by including db_connect.php in the input files which makes a connection with the database. This is done is all input files the need a database connection. Then it takes the inputted information for the form and puts them into a variable. The $_POST['location'] is calling the information inputted in the input name="location" in the form in course.php. The reason it is store in a variable right in the

beginning is for ease of use because the information could be used in many places in the file.
Then it makes a SQL statement string and store it in a variable called $sql. This statement is inputting into the table course. The table course has fields name and location. Into name is put the information stored in the variable $name and into location is stored the information from the variable $location. These variables contain the information from the form in course.php. The reason for storing SQL statement in a variable is so that the query looks nicer and if it is used in another place in the file.
Next there is variable by the name $login_sql_check which queries the database in this case for all information about course in the course table of the database where the name of the course is the same as the name stored in the $name variable. It then stores how many rows where in the course table with that name. There should not be any.
The if statement if($login_check == 0) is saying that if the number of rows from the query equal zero then continue else the course exists in the system. Then a string saying that is displayed along with course.php.
If the $login_check does equal zero then next is check the if statement if ($name != "" && $location != ""). If says if both variable $name and variable $location contain information then the $sql is stored in $query. The function mysql_query queries the database with the information in the $query variable which in this case $sql variable with an insert statement. If the SQL has an error of any kind mysql_error is executed. If the SQL is inputted the string “The course has been stored in the database” is display along with the file tees.php. If on the other information is missing in either $name or $location then the string “Please fill in all information needed” along the course.php. The reason the course.php included when some error occurs in the input is to make easier for the golfer to input the information again. If this would not be it would mean that the golfer would have to get back to the course.php, for example, by using back on the browser. This is the typical structure of a PHP that inputs or queries the database in this system.

6.1.3 avg_stats.php and avg_stats_entry.php

This is the interface for the golfer to ask the system to calculate his average statistics. The information input here or in better word chosen here will enable a calculation of the average of a golfer statistics and will input that information into the avg_stats table in the database. This is approached in this way.
The reason the system does not do this automatically is because golfers could want to put in many rounds of golf before looking at their statistics. Then it would be lighter on the system to let the golfer decide when to calculate the average statistics.
The interface is a single combo box which includes the name of all the golfers in the system. That information’s are got from querying the player table.
What the avg_stats_entry.php does is very much and pretty complicated. It only takes in one posted data and that is the playerid. First it checks if the golfer has any information in the stats table because if he hasn’t then his average stats can not be calculated. That is done the routine login check.
Because the statistics of players is stored in the stats table the avg_stats has to query like this, `select avg(score*18) from stats where playerid=$playerid`, where the playerid in the stats table equals in the playerid picked in the avg_stats.php. The avg(score*18) is multiplying the average score of the golfer with that playerid with 18 to

get the average on 18 holes. This is done because all statistics in golf are aimed from a round which is 18 holes. The queries where stored in variables which were then put into the function mysql_result to get the result of the queries. The results were then stored in other variables which were then used as input in the avg_stats table. This is done for all the statistics except the drives which are a bit different. The different lies in that the beginning shots on par 3 holes are not considered drives.
What needed to be done was to count all holes the golfer had play and where not par 3 holes. Also needed to find out how many rounds he had played. Then by dividing the count of all holes which where not par 3 with the total of played rounds you get the average of the holes that are not par 3. That number can be used instead of 18 to multiply with to get in average statistics of the drives. One problem was when getting this to work and it was because it was needed to join the stats and holes tables together. The problem was that the count of the holes would always come out exactly of double size. The problem was in the join and the best way around it was to divide the count always with 2. That works great and causes not problems. The outcome of all this is stored in variable called $drive_count which is then used instead of 18 which is used in the other queries.
Then there are both an insert SQL statement and an update SQL statement. The reason for that is that if the golfer is calculated his average statistics for the first time then the insert statement is used otherwise the update statement is used. That is checked in the same as a login check.
After the average statistics have been put into the database the database is queried and the results displayed.

6.1.4 course.php and course_entry.php

This is the interface where you input the general information about a golf course. Information inputted here will be input into the course table in the database. This is approached in this way.
Inside the form in the course.php is one table which contain two text boxes. These text boxes are both of size 30. These boxes take in information about the name of the course and its location.
What course_entry.php then does is to take the posted information and put them into variables. Then is one SQL insert statement which is inputted into the database if there is data in both of the variables with the posted information and the course does not exist in the database. If that is the case then a string is displayed along with tees.php.
If on the other hand data is missing for either of the variables and a string is displayed and course.php displayed.

6.1.5 golf.php and golf_entry.php

This is the interface for the Standardized Statistical Form. This is the biggest interface in the system. The information inputted here will go into the stats table in the database. This is approached in this way.
Inside the form in the golf.php are two tables. In the upper one is four combo boxes to pick from. The player, course and rounds box have information got from querying the database from these tables. The tees combo box is regular option combo box where the colour are just hand put into the code. Then is the rest of the interface text boxes take in all the information that will be inputted into the form. A total box is with every row and it

is generated from a JavaScript. There came up a problem there because I could not make the JavaScript can in the name of each row and then only use one function so I had to make one functions for each total box.
The golf_entry.php file takes in all the posted information and put variables. Then it makes 18 SQL insert statement, one for each hole, with all the information that should go with a hole into the state table in the database. It has a routine login check and then 18 if statements to input each of the SQL statements.

6.1.6 holes.php and holes_entry.php

This is the interface where you input the information about each hole on a golf course. Information inputted here will be input into the holes table in the database. This is approach in this way.
Inside the form in the holes.php are two tables. The first one has two combo boxes, one where the course is picked and the other the colour of the tee. The course combo box queries the course table for information about all courses in the database. The tees combo box is regular option combo box where the colour are just hand put into the code.
The second table contain the input text boxes of size 2 for the length, par and handicap of the holes.
Also inside is a JavaScript which calculates on the fly the total of the length and score. The reason that is includes is to make it easier for the golfer to keep track of this total when inputted the data involved.
What holes_entry.php then does is to take the posted information and put them into variables. Then are 18 SQL insert statement one for each hole. On each hole there is inputted the courseid, colour of tee, hole number, length, par and handicap. There are three kind of check before the information's are inputted. First is the usual check whether these holes exist in the database. Second is check whether there is data in the courseid and tee colour and there inside are 18 if statement to check if there is data in the par for each hole. Here was a problem encountered. It would be better to have one check and then input to the database but there was problems running 18 MySQL queries at the same time and putting them into separate statement made it possible so it was done. It is a problem if say par is missing on just one hole then the input on the other will go through. Not the best way but it works on most cases. It could be solved by having the check for all par boxes in the first if statement. It will be fixed. If everything is successful then a string is displayed along with the round.php and golf.php.
If some data is missing for the checks then a string is displayed along with the holes.php to input again.

6.1.7 index.php, index_main.php, index_entry.php and main.php

This is the login interface for the system. Here the golfer enters his login name and password to get access to the system. The information input here query the player table in the database to check whether this login name and password are correct. It also displays the main interface of the system.

6.1.8 player.php and player_entry.php

This is the interface where a player input information about himself. Information inputted here will be input into the player table in the database. This is approached in this way.
The golfer inputs required information into the form on the player.php.
The player_entry.php checks if all the information were inputted, if the login name is at right size and does not exits and if the password is at right size.

6.1.9 recommendation.php and recommendation_output.php

This is the interface where the golfer asks the system to let the Statistical Analyzer to analyze his statistics and get back recommendation. The information selected here will trigger a execution of the Statistical Analyzer and it will query the avg_stats table for information and then it will inputted in recommendation table for recommendation after it has finished analyzing. That information will then be outputted.
The recommendation.php is just a combo box where the golfer is able to pick a golfer to get analyzed.
The recommendation_output.php is an important file because it executes the Main.java which is the Statistical Analyzer. It also gets back the recommendations from recommendation.java and then displays them in a nice table format. To be able to execute the Statistical Analyzer the recommendation_output.php has to give the class path to two .jar files. These files are jess.jar and mysql-connector.jar and these files are stored in the same directory as the interface and input files are. Also to execute, Java has to be installed and in path so that the Statistical Analyzer.

6.1.10 recommendation_admin.php and recommendation_admin_entry.php

This is the interface to input new recommendation for the golfer. Information inputted here will go into the recommendation table in the database. The reason there is a recommendation_admin.php file is because the golfer can not put in recommendations the administrator has to do that. This is approach in this way.
The interface consists of two checkboxes where the golfer choices the type and level of the recommendation. The recommendation id has to put in by hand in a text box. The reason the recommendation id is put in by hand is that it has to match with the recommendation id in the rule in the Statistical Analyzer that is going to give this recommendation. Then the recommendation itself is put into a larger text box which is of input type textarea.
The recommendation_admin_entry.php then inputs the data into the database in the usual way.

6.1.11 round.php and round_entry.php

This is the interface where the golfer input information about a new golf round. Information inputted here will go into the golfrounds table in the database. This is approach in this way.
The only thing this does is to put a new round into the golfrounds table. This has one text box to input the name of the round.

6.1.12 tees.php and tee_entry.php

This is the interface where the golfer input information about a new tee on an existing golf course. Information inputted here will go into the tees table in the database. This is approach in this way.
This takes in information about colour of a tee on a golf course along with information's about the course rate and slope for that tee. The golfer picks the colour of the tee and course from combo boxes and inputs into text boxes the course rate and slope.
What the tee_entry.php does the routine login check and check if the variables have some data and if that is the case inputs the tee into the database. If not then displays an message and the tees.php to try to input again.

6.1.13 Other files in the input/interface

Other files in the input/interface are the db_connect.php and new.css. The db_connect.php handles the connection with the database.
The new.css is a Cascading Style Sheet (CSS) which helps with consistency in the interface look. It makes sure that no interface looks different from another in background colour, the tables through out the system will look alike and more. The reason for using CSS is because is helps with consistency of the interfaces and prior experience.

6.2 Database

In this chapter will be gone through the implementation of the database for this system. As I said the design part my database has eight tables. Now I am going to walk though the making of each table and try to explain why I chose to do it the way I did. I will walk through the tables in alphabetic order. MySQL database was used as the database for this system.

6.2.1 avg_stats:

This stands for average statistics. This is the table that the statistical analyzer uses to get the data it uses for the analyses of the statistics. There is one foreign key in this table and that is the playerid. The reason for that is that each player in the system has his unique id. If this would not be a foreign key then the player could have more than one id which is unacceptable. The table below shows the description of the table in the MySQL database.

The description of the avg_stats table

```
+--------------------------+-----------------------------+------+-----+---------+-------+
| Field                    | Type                        | Null | Key | Default | Extra |
+--------------------------+-----------------------------+------+-----+---------+-------+
| playerid                 | int(10) unsigned            |      | PRI | 0       |       |
| avg_score                | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_putts                | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_drive_hit            | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_drive_missed_L       | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_drive_missed_R       | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_drive_missed_Total   | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_green_hit            | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_green_missed_L       | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_green_missed_R       | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_green_missed_S       | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_green_missed_O       | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_green_missed_Total   | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_shot0_50m_missed     | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_shot50_100m_missed   | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_shot100_150m_missed  | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_bunker               | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_penalty              | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_putts0_1m            | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_putts1_2m            | decimal(5,2) unsigned       |      |     | 0.00    |       |
| avg_putts2_3m            | decimal(5,2) unsigned       |      |     | 0.00    |       |
+--------------------------+-----------------------------+------+-----+---------+-------+
```

Figure 6: The description of the avg_stats table

Field	Explaination
playerid	The playerid is a foreign key from the player table. The type of the field is integer, int(10) stands for integer of maximum size of 10 digits. By having the field unsigned it prevents the field of getting a negative number. This field is used to connect the average statistics to the player who owns them. The reason his is a foreign key is so that a player has one unique id throughout the database.
All the rest	All except for the playerid field are of type decimal(5,2). The reason they are decimals are because they are taking in values from a calculation of an average number of the inputted stats from the golfer and those kinds of calculations tend to end in decimals. By having the field unsigned it prevents the field of getting a negative number. The structure decimal(5,2) stands for maximum length of 5 digits with two of them being in the decimal place e.g. a correct structure is 85,12, this has 4 digits and two in the decimal place. By using that structure it means that the biggest number is 999,99. I would that to be enough because the highest numbers that will come into this system is the score of the golfer and that tend to stay under 200 strokes per round on average. This is an average of the total so the number will not exceed that. The reason I found only two digits in the decimal place to be enough is because golf is always in whole numbers and more than two digits in the decimal would not help they golfer in any way. These fields store the average statistics which are derived from taking the average of the statistics stored in the stats table.

6.2.2 course:

In this table the name and the location of the golf courses are stored. This table stores the general information about the golf course. This is the first table of three which stores information about a golf course like was described in the design. The courseid is used as a foreign key in many tables and it is the thing that uniquely identifies each golf course in the system.

The description of the course table

```
+----------+------------------+------+-----+---------+----------------+
| Field    | Type             | Null | Key | Default | Extra          |
+----------+------------------+------+-----+---------+----------------+
| courseid | int(10) unsigned |      | PRI | NULL    | auto_increment |
| name     | varchar(30)      |      |     |         |                |
| location | varchar(30)      |      |     |         |                |
+----------+------------------+------+-----+---------+----------------+
```

Figure 7: The description of the course table

Field	Explanation
courseid	The courseid is the id of a golf course in the database. It is the primary key of this table. This field is auto_incremented meaning that when a new course is added to the table that course will get the next available id. If last inserted course had the id 12 then the new course will get the id 13. The reason it is auto_incremented is to make sure that no two courses will get the same courseid and too make it easier on the designer. The type is integer which can be 10 digit long.
name	The name field takes in the name of the golf course. Its type is varchar which mean it can take in any character, digit and spaces. The type varchar stands for variable characters. The difference between char and varchar is that char always stores the full length of the data even if the data is shorter. An example of this is when varchar is of length 30 and the data is only 5 characters then varchar only stores the 5 characters while char would fill up the field with spaces **[9]**. The maximum length of the varchar is 30. The reason for that is that I do no know any course name that will exceed 30 characters.
location	The location field takes in the location of the golf course. Its type is varchar which mean it can take in any character, digit, spaces and etc. The maximum length of the varchar is 30. The reason for that is that I do no know any location that will exceed 30 characters.

6.2.3 golfrounds:

In this table goes the name of a golfer's tournament round or just a round he played and the date he played the round. The reason for this table is to uniquely identify each round that the golfer plays. The tournament field could be called name of round but it was decided to name it tournament because most of the rounds inputted as statistics are tournament rounds.

The description of the golfrounds table

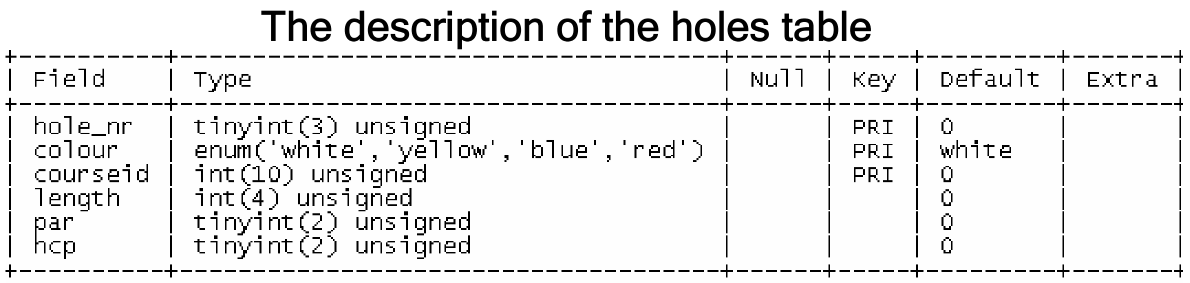
```
+-------------+------------------+------+-----+---------+----------------+
| Field       | Type             | Null | Key | Default | Extra          |
+-------------+------------------+------+-----+---------+----------------+
| golfroundid | int(10) unsigned |      | PRI | NULL    | auto_increment |
| date        | int(10) unsigned |      |     | 0       |                |
| tournament  | varchar(35)      |      |     |         |                |
+-------------+------------------+------+-----+---------+----------------+
```

Figure 8: The description of the golfrounds table

Field	Explanation
golfroundid	The golfroundid is the id of a golf round in the database. It is the primary key of this table. It is unsigned meaning that is can not be a negative number. It is auto_incremented so that there will not be two golf rounds with the same id.
date	The date is the date of when this round was inputted into the system. The type is integer of length 10. The reason for that is that the system uses UNIX timestamp to store the date and that is of length 10. The reason for using UNIX timestamp is because the comfort of it in PHP.
tournament	The tournament field takes in a tournament or just a name of a golf round that the golfer is going to into. Its type is varchar which mean it can take in any character, digit, spaces and etc. The maximum length of the varchar is 35. The reason for that is that I do no know any tournament name that will exceed 35 characters.

6.2.4 holes:

This table stores the information about each hole on a golf course. This the second table of three which stores information about a golf course. With each hole it stores the colour of the tee played from, the id of the course is stored in the courseid. The finally The reason for his tables is to identify each hole on a golf course by hole number, colour of tee and courseid. Also it stores the length of the hole, its par and its handicap which are information needed about each hole.



The description of the holes table

```
+----------+-------------------------------------+------+-----+---------+-------+
| Field    | Type                                | Null | Key | Default | Extra |
+----------+-------------------------------------+------+-----+---------+-------+
| hole_nr  | tinyint(3) unsigned                 |      | PRI | 0       |       |
| colour   | enum('white','yellow','blue','red') |      | PRI | white   |       |
| courseid | int(10) unsigned                    |      | PRI | 0       |       |
| length   | int(4) unsigned                     |      |     | 0       |       |
| par      | tinyint(2) unsigned                 |      |     | 0       |       |
| hcp      | tinyint(2) unsigned                 |      |     | 0       |       |
+----------+-------------------------------------+------+-----+---------+-------+
```

Figure 9: The description of the holes table

Field	Explanation
hole_nr	This field stores the number of hole of a golf course. This is a primary key but also colour and courseid are primary keys of this table. The reason for that is of course that a primary key of a table should be unique and it is only unique if all of those three are primary keys. All courses have hole number from 1-18 so that would not be unique. Have just courseid and hole_nr primary keys would not be unique because on each course there are usually 4 kind of tees which are identified with different colours. That is why all three of them are primary keys. The type of this field is tinyint which is takes upto three digit numbers. By having it unsigned I moved the tinyint value from “-128 to 127 to 0-255”.
colour	This field stores the number of the colour of hole. The type is enum and it works in that way that it can only take in the “strings” which are defined in the enum itself. Like in this field it can only take in white, yellow, blue or red

	because that is the colour of the tees in Iceland. It is a primary key and the reason why was explained above.
courseid	This field stores the id of the course which this hole is on. It is a foreign key from the course table.
length	This field stores the length of the hole. The type of this field is integer of the length of 4. The reason this is not a tinyint is because tinyint highest number is 255 but length of golf holes can be up to 600 meters long.
par	This field stores the par of the hole. The type of this field is tinyint. The reason for that is that it is enough for this field.
hcp	Thie field stores the handicap of the hole. The type of this field is tinyint. The reason for that is that it is enough for this field.

6.2.5 player:

This table stores all the information about a player in this system. The information stored in this table are the unique id every player has, his name, his handicap, his age, his login name and finally his password. The reason for this table is to keep all the general information about the player in one table. The playerid is used in many other tables as a foreign key because it uniquely identifies every player in the system. That is very important for this system because it is working with statistics from many players in the same tables. So, the playerid is used to retrieve the information for a specific player from different tables.

The description of the player table

```
+-----------+-----------------------+------+-----+---------+----------------+
| Field     | Type                  | Null | Key | Default | Extra          |
+-----------+-----------------------+------+-----+---------+----------------+
| playerid  | int(10) unsigned      |      | PRI | NULL    | auto_increment |
| firstname | varchar(30)           |      |     |         |                |
| surname   | varchar(30)           |      |     |         |                |
| hcp       | decimal(4,1) unsigned |      |     | 0.0     |                |
| age       | tinyint(3) unsigned   |      |     | 0       |                |
| login     | varchar(10)           |      |     |         |                |
| password  | varchar(32)           |      |     |         |                |
+-----------+-----------------------+------+-----+---------+----------------+
```

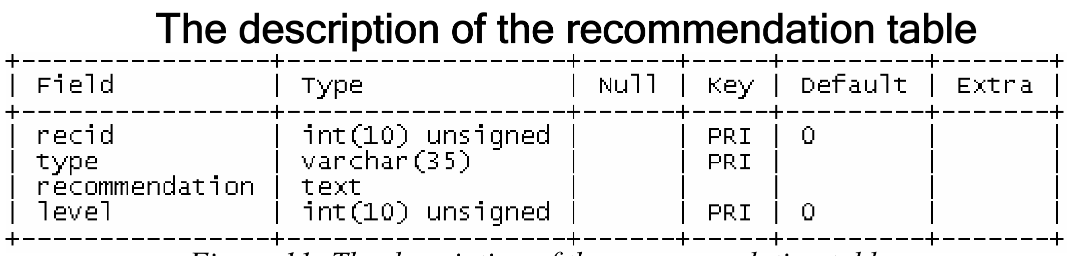
Figure 10: The description of the player table

Field	Explanation
playerid	This field stores the unique id for each player in the system. It is the primary key of the table. It is auto_inceremented so that no player gets the same id. The type is integer
firstname	This field stores the firstname of the player. The type is varchar with the maximum length of 30. The reason name is divided into firstname and surname is to able to use them separately.
surname	This field stores the surname of the player. The type is varchar with the maximum length of 30. The reason name is divided into firstname and surname is to able to use them separately.
hcp	This field stores the handicap of the player. The type of this field is tinyint. The reason for that is that it is enough for this field. This field will be useful in the recommendation part of the system when recommendations are split

	into 3 groups depending on the player's handicap. The players handicap also reflects his ability.
age	This field stores the age of the player. The type of this field is tinyint. The reason for that is that it is enough for this field.
login	This field stores the login name the player chose himself when he registers into the system. The type is varchar of the length of 10. The reason for that is that the maximum length of the login name in this system is 10 characters.
password	This field stores the password the player chose himself when he registers into the system. The type is varchar of the length of 32. The reason for that is that the passwords are stored encrypted in md5 encryption standard.

6.2.6 recommendation:

This is the only table in the database that does not have any relational with another table. This table stores the recommendations which the golfer will get. This table has three fields that form the primary key. These keys are recid, type and level. Then left is the recommendation field and it stores the text of the recommendations.


The description of the recommendation table

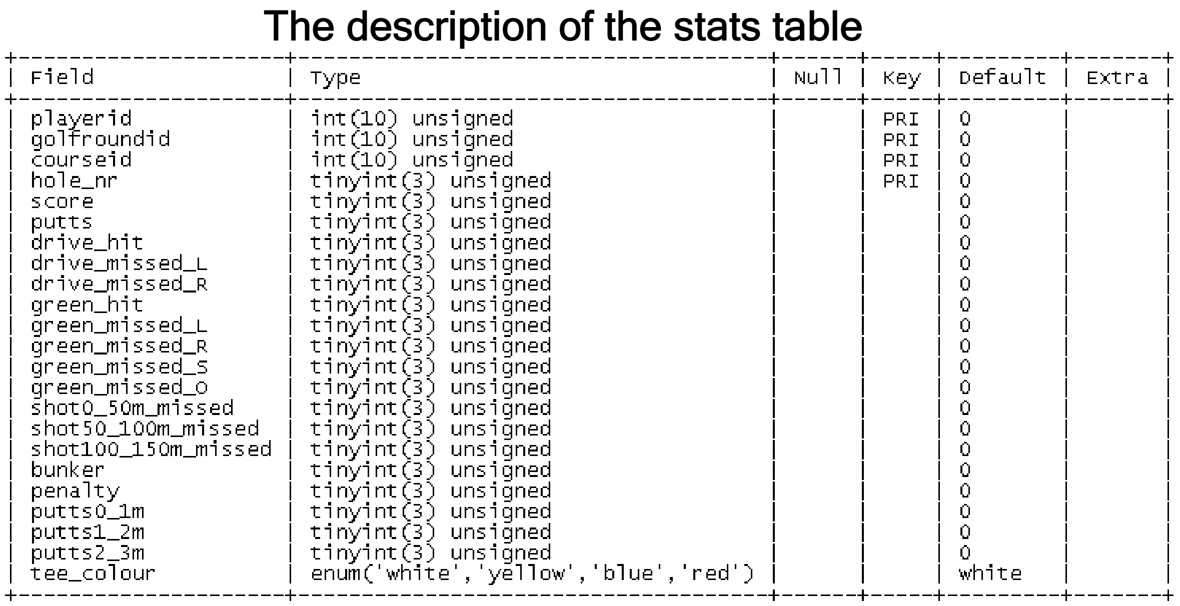
```
+----------------+------------------+------+-----+---------+-------+
| Field          | Type             | Null | Key | Default | Extra |
+----------------+------------------+------+-----+---------+-------+
| recid          | int(10) unsigned |      | PRI | 0       |       |
| type           | varchar(35)      |      | PRI |         |       |
| recommendation | text             |      |     |         |       |
| level          | int(10) unsigned |      | PRI | 0       |       |
+----------------+------------------+------+-----+---------+-------+
```

Figure 11: The description of the recommendation table

Field	Explanation
recid	This field stores the id of the recommendation in the database. This is one of three keys that form the primary key. The type is integer of length 10. The reason this is not a auto_increment field and is not a unique id for this table is because many recommendation can have the same recid but both at different level and different type.
type	This field stores the type of the recommendation. The type is varchar of length 35. The type takes in the types of the recommendations.
recommendation	This field stores the recommendations given to the golfer. The type is text. The reason for that is that it can take in all the text needed when a recommendation is made. The recommendations are text in very different length.
level	This field stores the level of the recommendation. The type is integer of length 10.

6.2.7 stats:

This is the most important table in the database. It stores the inputted statistics for every golfer in the system. This table is a replicate of the Standard Statistical Form is that way that each column in the database represents one column in the form. It covers all the fields in the form. There are three foreign keys in this table, they are playerid, golfroundsid and courseid. All of the foreign keys along with the hole_nr field form the primary key of the table. This table doesn't have one specific primary key like for example statsid, because then each column in the statistical form would have a unique id and that is not something which would be used in the table. One round has 18 holes and it wouldn't be useful to have each hole in that round to have specific stats id. The three foreign keys do not form a primary key together because each column in the database stands for one hole played. But adding the hole_nr to the three does the trick. Then this table takes in the statistics from the statistical form. Finally it has a field called tee_colour which takes in the colour of the tee played.


The description of the stats table

```
+---------------------+----------------------------------------+------+-----+---------+-------+
| Field               | Type                                   | Null | Key | Default | Extra |
+---------------------+----------------------------------------+------+-----+---------+-------+
| playerid            | int(10) unsigned                       |      | PRI | 0       |       |
| golfroundid         | int(10) unsigned                       |      | PRI | 0       |       |
| courseid            | int(10) unsigned                       |      | PRI | 0       |       |
| hole_nr             | tinyint(3) unsigned                    |      | PRI | 0       |       |
| score               | tinyint(3) unsigned                    |      |     | 0       |       |
| putts               | tinyint(3) unsigned                    |      |     | 0       |       |
| drive_hit           | tinyint(3) unsigned                    |      |     | 0       |       |
| drive_missed_L      | tinyint(3) unsigned                    |      |     | 0       |       |
| drive_missed_R      | tinyint(3) unsigned                    |      |     | 0       |       |
| green_hit           | tinyint(3) unsigned                    |      |     | 0       |       |
| green_missed_L      | tinyint(3) unsigned                    |      |     | 0       |       |
| green_missed_R      | tinyint(3) unsigned                    |      |     | 0       |       |
| green_missed_S      | tinyint(3) unsigned                    |      |     | 0       |       |
| green_missed_O      | tinyint(3) unsigned                    |      |     | 0       |       |
| shot0_50m_missed    | tinyint(3) unsigned                    |      |     | 0       |       |
| shot50_100m_missed  | tinyint(3) unsigned                    |      |     | 0       |       |
| shot100_150m_missed | tinyint(3) unsigned                    |      |     | 0       |       |
| bunker              | tinyint(3) unsigned                    |      |     | 0       |       |
| penalty             | tinyint(3) unsigned                    |      |     | 0       |       |
| putts0_1m           | tinyint(3) unsigned                    |      |     | 0       |       |
| putts1_2m           | tinyint(3) unsigned                    |      |     | 0       |       |
| putts2_3m           | tinyint(3) unsigned                    |      |     | 0       |       |
| tee_colour          | enum('white','yellow','blue','red')    |      |     | white   |       |
+---------------------+----------------------------------------+------+-----+---------+-------+
```

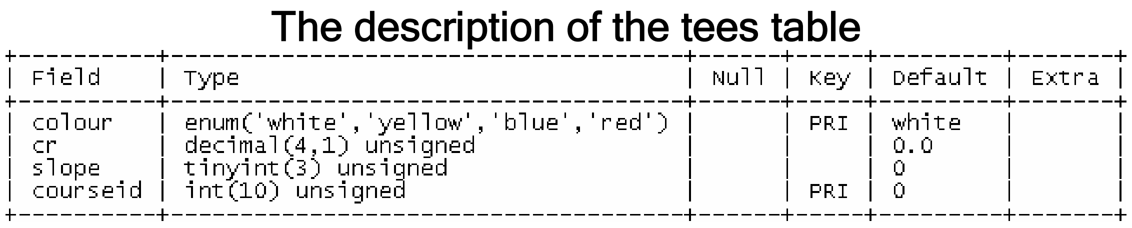
Figure 12: The description of the stats table

Field	Explanation
playerid	This field stores the id of the player. It is a foreign key from the player table. It is also one of four keys that form the primary key for this table. The reason for this to be a foreign key is because each player in the system has a unique id. The reason this is a primary key is for the same reason as why it is a foreign key. The type is like with any other id integer of length 10.
golfroundid	This field stores the id of the golfround that was played with the inputted statistics. It is a foreign key from the golfrounds table. It is also one of four keys that form the primary key for this table. The reason for this to be a foreign key is because each golfround in the system/database has a unique id. The reason this is a primary key is for the same reason as why it is a foreign key. The type is like with any other id integer of length 10.
courseid	This field stores the id of the course that was played with the inputted statistics. It is a foreign key from the course table. It is also one of four

	keys that form the primary key for this table. The reason for this to be a foreign key is because each course in the system has a unique id. The reason this is a primary key is for the same reason as why it is a foreign key. The type is like with any other id integer of length 10.
hole_nr	This field stores a hole number. This is an important key in the primary key. The type of this field is tinyint. The reason for that is that it is enough for this field.
Statistical fields	These fields store the statistical information of the system. The type of these fields is tinyint. The reason for that is that it is enough for these fields.
tee_colour	This field was not in the primary design of this table but was needed to add in. This field stores the colour of the tee being played off. The type is enum and it works in that way that it can only take in the “strings” which are defined in the enum itself. Like in this field it can only take in white, yellow, blue or red. The reason for this field is stores off what tee are played from which is important for the player to know when looking at the statistics later on. This field should probably in the golfrounds table because then it would only be put in once for the round but not for every hole.

6.2.8 Tees

This table stores the information about tee colour, course rate, slope for a golf course. This is the final table of three that store information about a golf course. There are two foreign keys is in this table and that is courseid which comes from the course table and the colour which comes from the holes table. The primary key consists of the foreign key because the uniquely identify a tee. Each golf course has usually has more the four different kinds of tees which are identified by different colour.



The description of the tees table

```
+----------+-------------------------------------+------+-----+---------+-------+
| Field    | Type                                | Null | Key | Default | Extra |
+----------+-------------------------------------+------+-----+---------+-------+
| colour   | enum('white','yellow','blue','red') |      | PRI | white   |       |
| cr       | decimal(4,1) unsigned               |      |     | 0.0     |       |
| slope    | tinyint(3) unsigned                 |      |     | 0       |       |
| courseid | int(10) unsigned                    |      | PRI | 0       |       |
+----------+-------------------------------------+------+-----+---------+-------+
```

Figure 13: The description of the tees table

Field	Explanation
colour	This field stores the colour of the each tee in the database. This is a one of the primary keys in this table and this is also a foreign key. The type is enum and it works in that way that it can only take in the strings which are defined in the enum itself. Like in this field it can only take in white, yellow, blue or red. The reason for that is that these are the only colours here in Iceland. This would have to be changed if this system goes out of Iceland.
cr	This field stores the course rating of a golf course. The type is decimal with 4 digits and one of them in the decimal place. The reason for that is that course rates are almost all in the format two digits comma one digit.

slope	This field stores the slope of the golf course. The type is tinyint. The reason for that is that slope will never in the near future go much over 160.
courseid	This field stores the courseid of the tees in this table. This is a primary key of this table and a foreign key from the course table.

6.3 Statistical Analyzer/Pattern Matcher

This chapter is about the implementation of the Statistical Analyzer. It will be spilt out into a walkthrough of the how the system works, talk about JESS embedded in Java and talk about the facts and the rules. The Statistical Analyzer is the part of the system that analyzes the statistics of the golfer looking for the faults in his game. It uses JESS rule engine to find the patterns of faults in an inputted fact which come from the average statistics of the golfer.

6.3.1 Walkthrough of how it works

The execution of the Statistical Analyzer is done in the PHP file recommendation_output. It calls executes the Main.java with one argument which is the playerid. The reason the playerid is put in as a parameter is because to get the right golfer from the database to analyze his playerid is needed. Now Main.java goes to work and takes that parameter (playerid) and puts it into a variable. It then executes the Main constructor with the argument inside (new Main(1)). Now the system is in business and gets it connection to the database through DB.java. It then uses the method PreparedStatement to query the database with a prepared SQL string. Into that string it inputs playerid to get the avg_stats of the right golfer. The ResultSet of the query is then gone through to retrieve the information from the average statistics table. All the average statistics, if any, for that query are then put into variables. Another query is then made to get the handicap of the golfer. Now the Statistical Analyzer kicks off. It starts by making a JESS engine. Then the systems use the store and fetch method inside the engine. Next it takes in all the average statistics saved in variables before and stores them in JESS Value. Next are the rule defined. First rule to be defined is a rule named start. This rule asserts the facts into the database along with storing the four JESS variables later used to stores the recommendation ids. They are stored with the value of zero because if no recommendation is found for any of the types the recommendation would be zero and no recommendation outputted. If they wouldn’t be stored in before hand then it would output an error if no recommendation would be found. Then the rules which will be used to find the faults in the average statistics are defined. Then the engine is executed. The final part of the Main constructor fetches the values containing the recommendation ids, which were stored in the engine, and stores them in variables. These variables are then set as parameters along with the handicap of the golfer in a call to the recommendation.java file (new recommendation(putts, drive, pitch, shots, handicap)). Now has the Statistical Analyzer finished is work and handed the ball to the recommendation/output part of the system.

6.3.2 JESS embedded in Java

JESS embedded in Java was the option taken in making the engine of this analyzer. As said in the design the reasons that were several and one of them were because that would mean using Java.
When starting JESS inside Java the first thing was to make a JESS engine (Rete StatisticalAnalyzer = new Rete()). This engine has inside its own working memory, rules and functions. The working memory is also called the fact base or knowledge base. Then the StatisticalAnalyzer.executeCommand is used inside the engine to assert facts and rules as needed. This is exactly how it is used in this system. Also used are the StatisticalAnalyzer.store and StatisticalAnalyzer.fetch commands to stores and fetch values that are inside the engine. Values can be put into the machine and got from the machine. Then in the end StatisticalAnalyzer.reset() and StatisticalAnalyzer.run() are called to reset the fact base and start the engine.

6.3.3 Facts implementation

After the information from the average statistics had been stored in JESS Values the first rule is made. This is the rule that inserts the facts into the working memory. The rules name is startup and it uses the JESS function assert to input the facts. This is an example of an assert function (assert (id (fetch playerid))). This assert function inserts a fact with the name id and the value of playerid into the working memory. This is done for all the values from the average statistics. That means that all the statistics from the average statistical table the inputted as facts in the working memory.

6.3.4 Rules implementation

The rules made in this system are all except the start rule described in the fact implementation made to try to patterns of faults in the golfer average statistics. The rules are all built in a similar way. On the LHS the have facts the need to be true so that the rule fires and the RHS will be activated. The RHS of every rule has a value being stored. To explain this better lets get a rule from the system.

```
"(defrule drives" +
"     (avg_drive_hit ?b &:(< ?b 6.5))" +
"     (avg_drive_missed_Total ?a &:(> ?a 8))" +
" =>" +
"     (store Drives (+ 0 3)) )"
```

This is how the rules were implemented into the system. The name of the rule is drives. The LHS says that if average drive hit is under 6.5 and the total drive missed is over 8 on average then store the recommendation id 3 to Drives. Each of the four types of recommendations, that are drives, putts, pitch and shot, will get a recommendation id if any of there rules are fired. Each rule will be explained in the next chapter.
The rules are grouped into four groups. These groups are the same as in the Standardized Statistical Form. They are drives, putts, pitches and shots. The reason for that change for the implementation is that a problem was encountered with the store of the rules. They problem was that when running the analyzer only the value of the first rule executed was stored even though there were other that should execute also. After a bit a struggle and conversions with the golf teacher it was decided to group this down even though it would

mean that only one recommendation would come from each group. The reason that is alright is because the golfer should only fix one error at a time in each of these fields in his game. But that also meant that the rules would have to be ordered in that way that the order with the highest priority would be at the top of each group.

6.3.5 The rules in the system now

In this chapter it will be told what rules are already in the system.
Drives
In drives are three rules already in the system. These rules are:

1. drives, if average drive hit is under 6.5 and the total drive missed is over 8 on average then give drive recommendation with id number 3.

This rule aims to check whether the golfer is overall a bad driver.

2. drives-right, if average drive hit is under 6.5 and the drives missed right is over 5 on average then give drive recommendation with id number 1.

This rule aims to check whether the golfer hit his drives too often to the right of the fairway.

3. drives-left, if average drive hit is under 6.5 and the drives missed left is over 5 on average then give drive recommendation with id number 2.

This rule aims to check whether the golfer hit his drives too often to the left of the fairway.

Putts
In putts are three rules already in the system. These rules are:

1. putts0_1m, if average putts is over 30 putts and average putts missed less than 1 meter from the hole is more than 1 then give putts recommendation with id number 1.

This rule aims to check whether the golfer missed too many putts less than 1 meter from the hole.

2. putts1_2m, if average putts is over 30 putts and average putts missed between 1 and 2 meters from the hole is more than 1 then give putts recommendation with id number 2.

This rule aims to check whether the golfer missed too many putts between 1 to 2 meters from the hole.

3. putts2_3m, if average putts is over 30 putts and average putts missed between 2 and 3 meters from the hole is more than 1 then give putts recommendation with id number 3.

This rule aims to check whether the golfer missed too many putts between 2 to 3 meters from the hole.

Pitch
In pitch are four rules already in the system. These rules are:

1. pitches, if average green hit is under 8 and both average shots less than 50 meters from the green and between 50 to 100 meters from the green are over 1 then give pitch recommendation number 1.

This rule aims to check whether the golfer is missing the green too often from less than 100 meters from the green.

2. pitch0_50m, if average green hit is under 8 and average shots less than 50 meters from the green is over 1 then give pitch recommendation number 2.

This rule aims to check whether the golfer is missing the green too often from less than 50 meters from the green.

3. pitch50_100m, if average green hit is under 8 and average shots between 50 to 100 meters from the green are over 1 then give pitch recommendation number 3.

This rule aims to check whether the golfer is missing the green too often from between 50 to 100 meters from the green.

4. pitch_closer, if average shots less than 50 meters from the green is less than 2, average shots between 50 to 100 meters from the green are less than 2, average putts is over 30 putts, average putts missed less than 1 meter from the hole is less than 1 and average putts missed between 1 and 2 meters from the hole is less than 1 then give pitch recommendation number 4.

This rule aims to check whether the golfer is hitting his shots less than 100 meters from the hole onto the green but not close enough to the hole to one putt.

Shots

In shots are three rules already in the system. These rules are:

1. length_problem_shot, if average green hit is under 8 and if both average green missed short or long is over 3 then give shots recommendation id 2.

This rule aims to check whether the golfer is missing the green too often short or long.

2. accuracy_problem_shot, if average green hit is under 8 and if both average green missed right or left is over 3 then give shots recommendation id 3.

This rule aims to check whether the golfer is missing the green too often right or left.

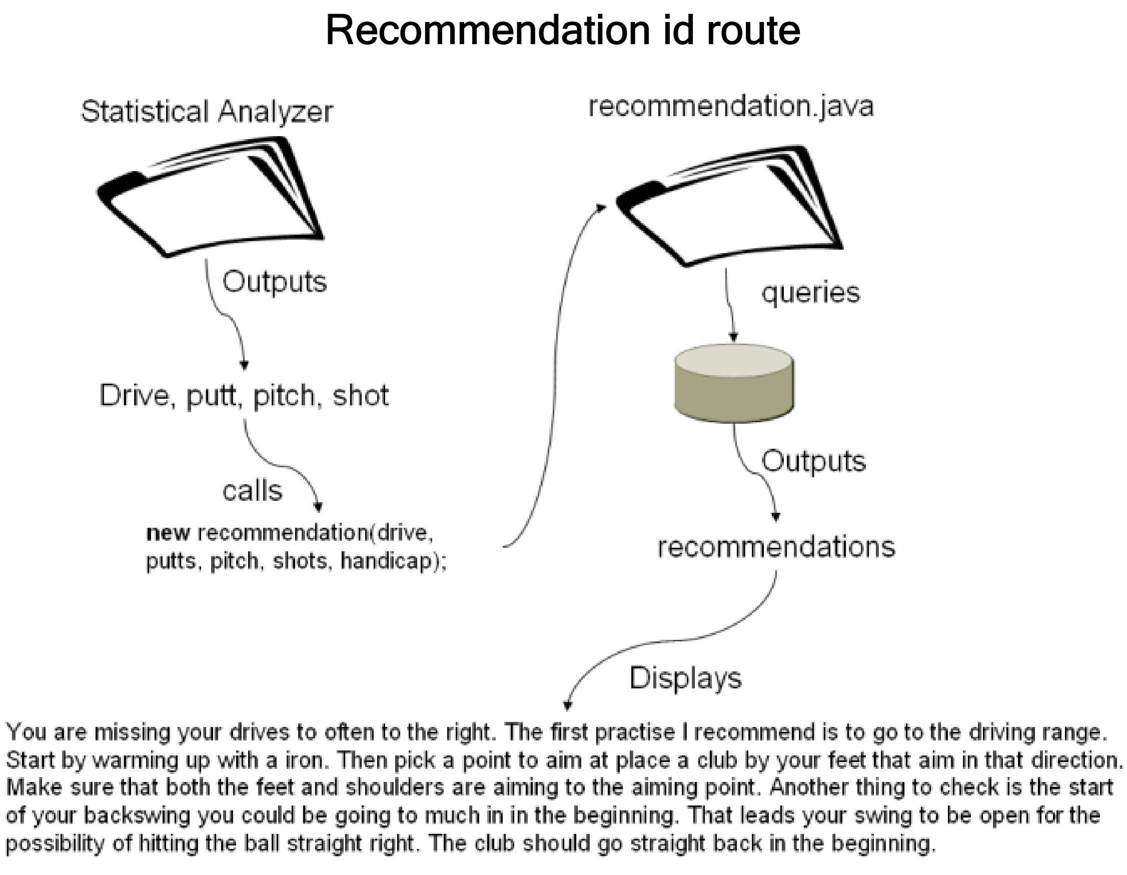
3. GIR_shot, if average green hit is under 8 and average shots between 100 to 150 meters from the green are more than 3 then give shots recommendation number 1.

This rule aims to check whether the golfer is missing the green too often overall.

In total these are 13 rules they needed to be tested thoroughly. Only these 13 rules will be in this version of the system.

6.4 Recommendations/Output

After the design of the recommendations was ready then the rest was just getting the help of the golfer teacher to make the recommendations. The recommendations Java file/class, which name is recommendation.java, gets the recommendations by using the types name and their recommendation id got from the Statistical Analyzer. The constructor takes in five parameters when called. These values are int putt, int driv, int shot, int pitc and float hand. The first four contain the recommendations ids for the types from the analyzer and the hand variable contains the golfer's handicap. It first thing done is to check the level of the golfer by checking his handicap and stores that is a variable named handicap. Then it connects to the database and queries the recommendation table. It connects to the database through the DB.java which handles. The query string includes the recommendation from the analyzer and the level of the golfer. This is done by using the PreparedStatement functions in Java. That function returns a ResultSet from the query which then is "queried" for the recommendation. Then it prints out the recommendation from the query with system.out.println. These sections queries and prints out recommendations for all four types of the game that is drives, putts, pitch and shots. The file recommendation_output.php which executed the Statistical Analyzer in the first place then takes the system.out.println from the recommendation.java and displays them in a nice way. Doing it like this, that is, using PHP to execute the java class files lets me slide past using servlets or applets.



You are missing your drives to often to the right. The first practise I recommend is to go to the driving range. Start by warming up with a iron. Then pick a point to aim at place a club by your feet that aim in that direction. Make sure that both the feet and shoulders are aiming to the aiming point. Another thing to check is the start of your backswing you could be going to much in in the beginning. That leads your swing to be open for the possibility of hitting the ball straight right. The club should go straight back in the beginning.

Figure 14: The recommendation id route

What this diagram is showing is the way from when the Statistical Analyzer output is recommendation id to when the recommendation are displayed on the screen of the golfer. The Statistical Analyzer output the recommendation id and they are put into 4 variables which are named the names of the recommendations types. The variables along with the handicap of the golfer are set in as parameters when the recommendation.java is called. The recommendation.java takes in the parameters and uses them to query the database for the recommendations which are then printed out by the recommendation.java in system.out.println. The recommendation_output.php then takes the output from the recommendation.java and displays the recommendations, if any, in a nice way.

7 Evaluation

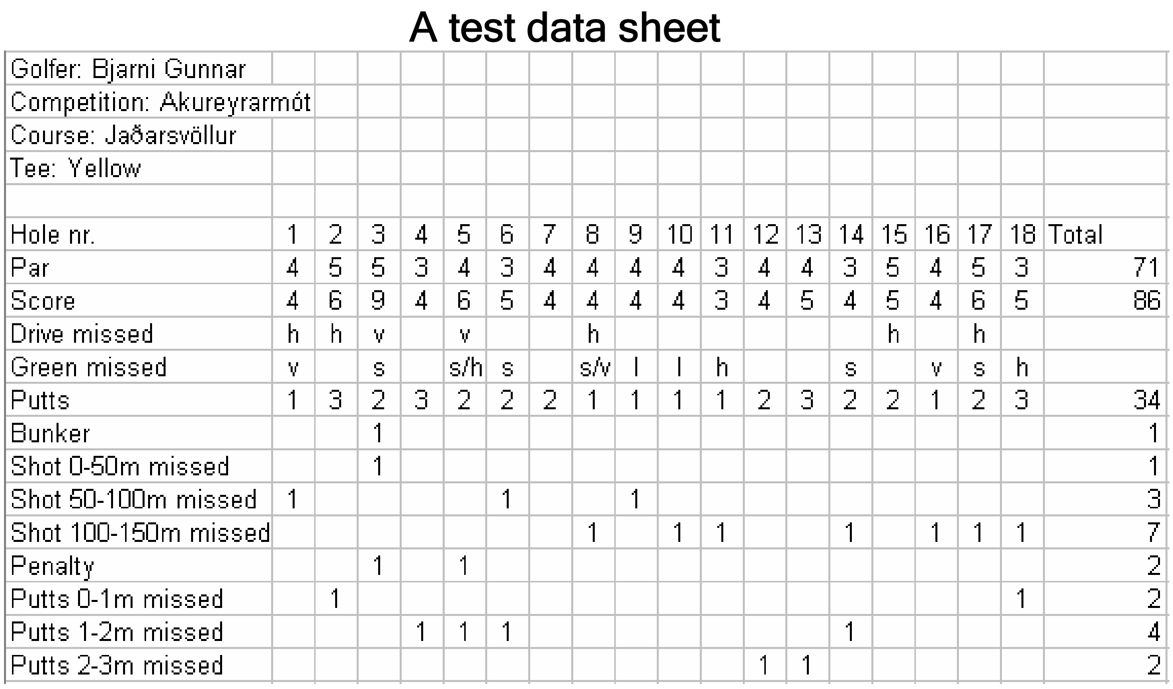
In this chapter will be talked about the tests made on this system so far, what was achieved of the things stated in the chapter description of the work and then it will end with a conclusion.

7.1 Testing

In this chapter will be told from the tests that have been made on the system so far. First the test data will be explained and then the tests made.

7.1.1 Test data

The tests on the system have in major part aimed for testing the statistical analyzer to check whether it gives out the right recommendation depending on the input statistics. To do this test data has been made in the form the statistics. This data has aimed to check the rules that are already in the system. The test data was made in Excel and one statistical form looks like this.



A test data sheet

Golfer: Bjarni Gunnar																			
Competition: Akureyrarmót																			
Course: Jaðarsvöllur																			
Tee: Yellow																			
Hole nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Total
Par	4	5	5	3	4	3	4	4	4	4	3	4	4	3	5	4	5	3	71
Score	4	6	9	4	6	5	4	4	4	4	3	4	5	4	5	4	6	5	86
Drive missed	h	h	v		v			h							h		h		
Green missed	v		s		s/h	s		s/v	l	l	h			s		v	s	h	
Putts	1	3	2	3	2	2	2	1	1	1	1	2	3	2	2	1	2	3	34
Bunker			1																1
Shot 0-50m missed			1																1
Shot 50-100m missed	1					1			1										3
Shot 100-150m missed								1		1	1			1		1	1	1	7
Penalty			1		1														2
Putts 0-1m missed		1																1	2
Putts 1-2m missed				1	1	1								1					4
Putts 2-3m missed												1	1						2

Figure 15: A test data sheet

This is a test statistical form. The information from the test form was inputted into the system and it executed.

7.1.2 Testing the system

When testing the system most of the test where to test the Statistical Analyzer. This chapter will split into two things. First testing made on the Statistical Analyzer and recommendations and the testing made on the input and database. The reason for this split is that these pars work closely together in the system.

7.1.2.1 Statistical Analyzer and Recommendations

The system now contains 13 rules and have all of them been tested. To able check every single on it was needed to input 13 statistical forms with 13 different players. The reason for that is that it is the best way of making sure that all of the rules work properly. The outcome of the tests will be displayed a table here below.

The inputted rules			What recommendation came out	
Rule name	Type	id	Type	id
drives	drive	3	drive	3
drives-right	drive	1	drive	1
drives-left	drive	2	drive	2
putts0_1m	putt	1	putt	1
putts1_2m	putt	2	putt	2
putts2_3m	putt	3	putt	3
pitches	pitch	1	pitch	1
pitch0_50m	pitch	2	pitch	2
pitch50_100m	pitch	3	pitch	3
pitch_closer	pitch	4	pitch	4
length_problem_shot	shot	2	shot	2
accuracy_problem_shot	shot	3	shot	3
GIR_shot	shot	1	shot	1

As can been seen from this test table all the tests were a success. This is exactly what was needed to been seen. The reason for that is that this is the heart of the system. To able to show just one pattern found and give recommendations would have been a success in my eyes. But be able to show more is just great.

7.1.2.2 Input and Database

The tests made on the input have not been very organized. They have just been done when the interfaces and the files with the input have been made. The inputs have been compared with the data in the database. Not much power has been put into interfaces because they are an extra feature in this project. Tests were made on the input into both the stats table and the average statistics table after they have been made to verify that the calculation of the input where correct. The database needs to tested more thoroughly and hopefully it will be done soon.

7.2 What was achieved

In this chapter will be talked about what was achieved of the things stated in the chapter description of the work. This will just be a recitation what was achieved and what was not achieved.

7.2.1 Achieved

The database of statistics is fully functional. The database of recommendations changed a little bit. It just has one recommendation for each pattern not 10 reason and then recommendations. The Statistical Analyzer is working for the rules inputted and finds patterns of faults. The Standardized Statistical Form is ready and is used when the golfer inputs its statistics. These were the things that were supposed to be functional and they are. Also ready is the web site with all the functionalities to input the statistics from a web site.

7.2.2 Not achieved

The comparing two players together is the only thing that is not ready but hopefully it will come soon enough.

7.3 Conclusion

In this conclusion of my final year project I am going to talk about where I see the future of my system.
The future of this system looks good. In the future I see more input forms with more detail. This will increase the depth of detail the system can go into when giving recommendations to the golfer. Not that the current form isn't enough but this would aim more to the professionals. The system will be able to give about 10 different reason for each pattern matched in the system. Under these 10 reasons will be practises, technical and mental details. This requires a lot of work in the recommendation part and maybe some slight changes in the rules part. Each pattern will have to be more specific, that means more rules. More security is needed and to make the interface more user friendly.
How the statistics are inputted today is long procedure. This procedure could be made easier by making the input form in that way that they can be used in a palm. Also by using the GPS technology it could be made to track the golf round of the golfer down to the smallest detail. Making large golfers website is a possibility where the golfer can find everything he needs.
I have shown proof of concept and the above further implementations would take the system further.

Reference

1. "GameTrack Version 3.0" [online], (2002), (www.freeinfogolf.com),
2. Available at: <URL: http://www.axiagolf.com/gametrack3/index.asp?ax=fgi> (Accessed: 2003, October 10).

3. von Bremem, Klaus. "Prolog Parsers.Parsing techniques in Prolog" [online]. Available at: <URL: http://utenti.lycos.it/parsers/> (Accessed: 2003, November 20).

4. "egolfcard.com - The UK and Irelands Premier Online Golf Score Recording and Analysis Website" [online], (2003), (www.egolfcard.com), Available at: <URL: http://www.egolfcard.com/> (Accessed: 2003, October 18).

5. Van Learhoven (2003, 21.January – last update), "CLIPS versus JESS" [online], (www.comp.lancs.ac.uk/~kristof/), Available at: <URL: http://www.comp.lancs.ac.uk/~kristof/research/notes/clipsvsjess/> (Accessed: 2003, November 24).

6. Perdue, Tim. PHPBuilder.com – "MySQL and PostgreSQL Compared" [online],(www.PHPBuilder.com). Available at: <URL: http://www.phpbuilder.com/columns/tim20000705.php3> (Accessed: 2003, November 20).

7. "Harvard Style examples" [online], (2003, 26.September – last update), (www.lib.monash.edu.au/), Available at: <URL: http://www.lib.monash.edu.au/vl/cite/harvex.htm> (Accessed: 2003, November 27).

8. "Gofor GOLF - free golf statistics analysis and golf course directory - web database software to track and improve golf scores [online], (2003), (www.goforgolf.com), Available at: <URL: http://www.goforgolf.com/> (Accessed: 2003, October 10).

9. "MySQL® Database Server" [online], (2003), (www.mysql.com), Available at: <URL: http://www.mysql.com/products/mysql/index.html> (Accessed: 2003, November 21).

10. "HIS-Database Terms Dictionary" [online], (2003), (http://his.osu.edu/), Available at: <URL: http://his.osu.edu/help/database/terms.cfm> (Accessed: 2003, Mars 30).

11. "ScienceDirect – Expert Systems with Applications : Feature-based recommendations for one-to-one marketing" [online], (2003, 27.november – Available online), Available at: <URL: http://www.sciencedirect.com/science?_ob=ArticleURL&_aset=B-WA-A-A-D-

MsSAYZA-UUW-AUYVWDVCEU-AUYAYCCBEU-BCBDBABBA-D-U&_rdoc=2&_fmt=full&_udi=B6V03-4B3NGWB-1&_coverDate=05%2F31%2F2004&_cdi=5635&_orig=search&_st=13&_sort=d&view=c&_acct=C000039758&_version=1&_urlVersion=0&_userid=1829798&md5=61aa407a7ebe6277f4c2d2336099644a> (Accessed: 2003, February 21).

12. Friedman-Hill, Ernest. (2003), *JESS in action*, Manning Publications Co., Greenwich, United State of America.

13. "SPHINcsX: Glossary" [online], (2004), (http://www.stanford.edu), Available at: <URL: http://www.stanford.edu/~buc/SPHINcsX/bkhm15.htm> (Accessed: 2003, Mars 31).

Appendix A

Here is all the code written in this final year project. The headings are the names of the files which the codes belong to.

Main.java

```
import jess.*;
import java.sql.*;

/*
      Class : Main
      Author   : Bjarni Gunnar
      Description : This class drives the Statistical Analyzer.
                          It gets necessary input for the Statistical
Analyzer/JESS Engine.
                          It loads fact and rules into the JESS engine
and fires it up.
                          It fetches the recommendation id after the
JESS engine has finished
                          and uses them as parameter to call
recommendation.java.
*/

public class Main
{
      //declaring some variables
      private final static String getAvgStats = "select * from
avg_stats where playerid=?";
      private final static String getHandicapStmt = "select hcp from
player where playerid=?";
      int playerid, drive, putts, pitch, shots;
      float avg_score, avg_putts, avg_drive_hit, avg_drive_mL,
avg_drive_mR, avg_drive_mT;
      float avg_green_hit, avg_green_mL, avg_green_mR, avg_green_mS,
avg_green_mO, avg_green_mT;
      float avg_shot0_50m, avg_shot50_100m, avg_shot100_150m,
avg_putts0_1m, avg_putts1_2m, avg_putts2_3m;
      float avg_bunker, avg_penalty, handicap;

  public static void main(String[] args) throws Exception
  {
      int i = Integer.parseInt(args[0]);
      new Main(i);
  }

  /*
            Constructor : Main
            Author   : Bjarni Gunnar
            Description : This constructor drives the Statistical
Analyzer.
                                It takes in one parameter. It gets
necessary input for the
                                Statistical Analyzer/JESS Engine by
using the input arguments as a
                                key for querying the database.
```

```
                              It loads fact and rules into the JESS
engine and fires it up.
                              It fetches the recommendation id after
the JESS engine has finished
                              and uses them as parameter to call
recommendation.java.
            Parameters : int id
            Returns:
   */

  public Main(int id)
  {
      //resets the connection
    Connection conn = null;
      try
      {
            //gets connection to the database
            conn = DB.getConnectionToDB();

                  //Queries the database with the Prepared Statement
and the id
                  PreparedStatement stmt =
conn.prepareStatement(getAvgStats);
                  stmt.setInt(1, id);
                  // Creates a Result Set
                  ResultSet rs = stmt.executeQuery();
                  // Fetches each row from the Result Set rs and puts
them into variables
                  // It is fetching the average statistics
                  while (rs.next())
                  {
                        playerid = rs.getInt("playerid");
                        avg_score = rs.getFloat("avg_score");
                        avg_putts = rs.getFloat("avg_putts");
                        avg_drive_hit = rs.getFloat("avg_drive_hit");
                        avg_drive_mL =
rs.getFloat("avg_drive_missed_L");
                        avg_drive_mR =
rs.getFloat("avg_drive_missed_R");
                        avg_drive_mT =
rs.getFloat("avg_drive_missed_Total");
                        avg_green_hit = rs.getFloat("avg_green_hit");
                        avg_green_mL =
rs.getFloat("avg_green_missed_L");
                        avg_green_mR =
rs.getFloat("avg_green_missed_R");
                        avg_green_mS =
rs.getFloat("avg_green_missed_S");
                        avg_green_mO =
rs.getFloat("avg_green_missed_O");
                        avg_green_mT =
rs.getFloat("avg_green_missed_Total");
                        avg_shot0_50m =
rs.getFloat("avg_shot0_50m_missed");
                        avg_shot50_100m =
rs.getFloat("avg_shot50_100m_missed");
```

```
                        avg_shot100_150m =
rs.getFloat("avg_shot100_150m_missed");
                        avg_bunker = rs.getFloat("avg_bunker");
                        avg_penalty = rs.getFloat("avg_penalty");
                        avg_putts0_1m = rs.getFloat("avg_putts0_1m");
                        avg_putts1_2m = rs.getFloat("avg_putts1_2m");
                        avg_putts2_3m = rs.getFloat("avg_putts2_3m");
                  }

                  //if not disconnection then closes the connection to
the database
                  if(conn != null)
                  {
                        conn.close();
                  }
            }
            catch (SQLException e)
            {
            }

      try
      {
            //gets connection to the database
            conn = DB.getConnectionToDB();

            //Queries the database with the Prepared Statement and the
id
            PreparedStatement getHandicap =
conn.prepareStatement(getHandicapStmt);
            getHandicap.setInt(1, id);
            ResultSet getHandi = getHandicap.executeQuery();
            // Fetches each row from the Result Set getHandi, is
fetching the handicap
            while (getHandi.next())
            {
                  handicap = getHandi.getFloat("hcp");
            }

            //if not disconnection then closes the connection to the
database
            if(conn != null)
                  {
                        conn.close();
                  }
      }
            catch (SQLException e)
            {
            }

    try
      {
        // Create the JESS engine
        Rete StatisticalAnalyzer = new Rete();

        // Stores the values of the average statistics inside the
engine
```

```
        StatisticalAnalyzer.store("playerid", new Value(playerid,
RU.INTEGER));
        StatisticalAnalyzer.store("avg_score", new Value(avg_score,
RU.FLOAT));
        StatisticalAnalyzer.store("avg_putts", new Value(avg_putts,
RU.FLOAT));
        StatisticalAnalyzer.store("avg_drive_hit", new
Value(avg_drive_hit, RU.FLOAT));
        StatisticalAnalyzer.store("avg_drive_missed_L", new
Value(avg_drive_mL, RU.FLOAT));
        StatisticalAnalyzer.store("avg_drive_missed_R", new
Value(avg_drive_mR, RU.FLOAT));
        StatisticalAnalyzer.store("avg_drive_missed_Total", new
Value(avg_drive_mT, RU.FLOAT));
        StatisticalAnalyzer.store("avg_green_hit", new
Value(avg_green_hit, RU.FLOAT));
        StatisticalAnalyzer.store("avg_green_missed_L", new
Value(avg_green_mL, RU.FLOAT));
        StatisticalAnalyzer.store("avg_green_missed_R", new
Value(avg_green_mR, RU.FLOAT));
        StatisticalAnalyzer.store("avg_green_missed_S", new
Value(avg_green_mS, RU.FLOAT));
        StatisticalAnalyzer.store("avg_green_missed_O", new
Value(avg_green_mO, RU.FLOAT));
        StatisticalAnalyzer.store("avg_green_missed_Total", new
Value(avg_green_mT, RU.FLOAT));
        StatisticalAnalyzer.store("avg_shot0_50m_missed", new
Value(avg_shot0_50m, RU.FLOAT));
        StatisticalAnalyzer.store("avg_shot50_100m_missed", new
Value(avg_shot50_100m, RU.FLOAT));
        StatisticalAnalyzer.store("avg_shot100_150m_missed", new
Value(avg_shot100_150m, RU.FLOAT));
        StatisticalAnalyzer.store("avg_bunker", new Value(avg_bunker,
RU.FLOAT));
        StatisticalAnalyzer.store("avg_penalty", new Value(avg_penalty,
RU.FLOAT));
        StatisticalAnalyzer.store("avg_putts0_1m_missed", new
Value(avg_putts0_1m, RU.FLOAT));
        StatisticalAnalyzer.store("avg_putts1_2m_missed", new
Value(avg_putts1_2m, RU.FLOAT));
        StatisticalAnalyzer.store("avg_putts2_3m_missed", new
Value(avg_putts2_3m, RU.FLOAT));

            // First rule: This rules inputs the facts into the fact
base and
            // zeros the variables that take in the recommendation id
        StatisticalAnalyzer.executeCommand("(defrule startAnalyzing" +
                              " =>" +
                              "  (assert (id (fetch playerid)))" +
                              "  (assert (avg_score (fetch avg_score)))"
+
                              "  (assert (avg_putts (fetch avg_putts)))"
+
                              "  (assert (avg_drive_hit (fetch
avg_drive_hit)))" +
                              "  (assert (avg_drive_missed_L (fetch
avg_drive_missed_L)))" +
                            "  (assert (avg_drive_missed_R (fetch
avg_drive_missed_R)))" +
                            "  (assert (avg_drive_missed_Total (fetch
avg_drive_missed_Total)))" +
                            "  (assert (avg_green_hit (fetch
avg_green_hit)))" +
                            "  (assert (avg_green_missed_L (fetch
avg_green_missed_L)))" +
                            "  (assert (avg_green_missed_R (fetch
avg_green_missed_R)))" +
                            "  (assert (avg_green_missed_S (fetch
avg_green_missed_S)))" +
                            "  (assert (avg_greem_missed_O (fetch
avg_green_missed_O)))" +
                            "  (assert (avg_green_missed_Total (fetch
avg_green_missed_Total)))" +
                            "  (assert (avg_shot0_50m_missed (fetch
avg_shot0_50m_missed)))" +
                            "  (assert (avg_shot50_100m_missed (fetch
avg_shot50_100m_missed)))" +
                            "  (assert (avg_shot100_150m_missed (fetch
avg_shot100_150m_missed)))" +
                            "  (assert (avg_putts0_1m (fetch
avg_putts0_1m_missed)))" +
                            "  (assert (avg_putts1_2m (fetch
avg_putts1_2m_missed)))" +
                            "  (assert (avg_putts2_3m (fetch
avg_putts2_3m_missed)))" +
                            "  (store Drives (+ 0 0))" +
                            "  (store Shots (+ 0 0))" +
                            "  (store Putts (+ 0 0))" +
                            "  (store Pitch (+ 0 0)) )");

            // There are the rules that looks for the faults made
        StatisticalAnalyzer.executeCommand(

                                    //Rule looks for too many missed
drives
                                    "(defrule drives" +
                            "  (avg_drive_hit ?b &:(< ?b 6.5))" +
                            "  (avg_drive_missed_Total ?a &:(> ?a 8))"
+
                            " =>" +
                            "  (store Drives (+ 0 3)) )" +

                            //Rule looks for too many drives missed
right
                            "(defrule drives-right" +
                            "  (avg_drive_hit ?b &:(< ?b 6.5))" +
                            "  (avg_drive_missed_R ?a &:(> ?a 5))" +
                            " =>" +
                            "  (store Drives (+ 0 1)) )" +

                            //Rule looks for too many drives missed
left
                            "(defrule drives-left" +
                            "  (avg_drive_hit ?b &:(< ?b 6.5))" +
                            "  (avg_drive_missed_L ?a &:(> ?a 5))" +
                            " =>" +
                            "  (store Drives (+ 0 2)) )" +

                            //Rule looks for too many putts missed less
than 1m from hole
                            "(defrule putts0_1m" +
                            "  (avg_putts ?c &:(> ?c 30))" +
                            "  (avg_putts0_1m ?a &:(> ?a 1))" +
                            " =>" +
                            "  (store Putts (+ 0 1)) )" +

                            //Rule looks for too many putts missed 1-2m
from hole
                            "(defrule putts1_2m" +
                            "  (avg_putts ?c &:(> ?c 30))" +
                            "  (avg_putts1_2m ?a &:(> ?a 1))" +
                            " =>" +
                            "  (store Putts (+ 0 2)) )" +

                            //Rule looks for too many putts missed 2-3m
from hole
                            "(defrule putts2_3m" +
                            "  (avg_putts ?c &:(> ?c 30))" +
                            "  (avg_putts2_3m ?a &:(> ?a 1))" +
                            " =>" +
                            "  (store Putts (+ 0 3)) )" +

                            //Rule looks for too many pitches missed
50-100m from green
                            "(defrule pitch50_100m" +
                            "  (avg_green_hit ?c &:(< ?c 8))" +
                            "  (avg_shot50_100m_missed ?a &:(> ?a 1))"
+
                            " =>" +
                            "  (store Pitch (+ 0 3)) )" +

                            //Rule looks for too many pitches missed 0-
100m from green
                            "(defrule pitches" +
                            "  (avg_green_hit ?c &:(< ?c 8))" +
                            "  (avg_shot0_50m_missed ?a &:(> ?a 1))" +
                            "  (avg_shot50_100m_missed ?b &:(> ?b 1))"
+
                            " =>" +
                            "  (store Pitch (+ 0 1)) )" +

                            //Rule looks for too many pitches missed 0-
50m from green
                            "(defrule pitch0_50m" +
                            "  (avg_green_hit ?c &:(< ?c 8))" +
                            "  (avg_shot0_50m_missed ?a &:(> ?a 1))" +
                            " =>" +
                            "  (store Pitch (+ 0 2)) )" +

                            //Rule looks for that golfer isn't pitching
close enough to the hole
```

```
                            "(defrule pitch_closer" +
                            "  (avg_shot0_50m_missed ?ac &:(< ?ac 1))"
+
                            "  (avg_shot50_100m_missed ?bc &:(< ?bc
2))" +
                            "  (avg_putts ?pc &:(> ?pc 30))" +
                            "  (avg_putts0_1m ?p1c &:(< ?p1c 1))" +
                            "  (avg_putts1_2m ?p2c &:(< ?p2c 1))" +
                            " =>" +
                            "  (store Pitch (+ 0 4)) )" +

                            //Rule looks for missed green too short and
too long
                            "(defrule length_problem_shot" +
                            "  (avg_green_hit ?c &:(< ?c 8))" +
                            "  (avg_green_missed_S ?a &:(> ?a 4))" +
                            "  (avg_greem_missed_O ?b &:(> ?b 3))" +
                            " =>" +
                            "  (store Shots (+ 0 2)) )" +

                            //Rule looks for missed green right and
left too often
                            "(defrule accuracy_problem_shot" +
                            "  (avg_green_hit ?c &:(< ?c 8))" +
                            "  (avg_green_missed_R ?a &:(> ?a 4))" +
                            "  (avg_greem_missed_L ?b &:(> ?b 3))" +
                            " =>" +
                            "  (store Shots (+ 0 3)) )" +

                            //Rule looks for too many GIR shots missed
                            "(defrule GIR_shot" +
                            "  (avg_green_hit ?c &:(< ?c 8))" +
                            "  (avg_shot50_100m_missed ?b &:(> ?b 1))"
+
                            " =>" +
                            "  (store Shots (+ 0 1)) )" );


        // Resets the fact base and run the rules

        StatisticalAnalyzer.reset();
        StatisticalAnalyzer.run();

        // Fetch the results of the running of the Statistical Analyzer
        // Fetch the recommendation id's and puts them into variables
        Value testValue = StatisticalAnalyzer.fetch("Putts");
        putts =
testValue.intValue(StatisticalAnalyzer.getGlobalContext());
        Value driveValue = StatisticalAnalyzer.fetch("Drives");
        drive =
driveValue.intValue(StatisticalAnalyzer.getGlobalContext());
        Value pitchValue = StatisticalAnalyzer.fetch("Pitch");
        pitch =
pitchValue.intValue(StatisticalAnalyzer.getGlobalContext());
        Value shotsValue = StatisticalAnalyzer.fetch("Shots");
        shots = 
shotsValue.intValue(StatisticalAnalyzer.getGlobalContext());
      }
      catch (JessException re)
      {
        re.printStackTrace();
      }

      // Method calls the recommendation.java with the recommendation
id's and handicap as parameter
      new recommendation(putts, drive, pitch, shots, handicap);
  }
}
```

recommendation.java

```
import java.sql.*;

/*
      Class : recommendation
      Author   : Bjarni Gunnar
      Description : This class finds the level of the golfer by
checking his handicap.
                          Then it queries the database with
recommendation id and the level
                          of the golfer to get the recommendation. It
then prints out the
                          recommendation.
*/

public class recommendation
{
      // declaers some variables
      private final static String getDriveRec = "select recommendation
from recommendation where recid=? and type='drive' and level=?";
      private final static String getPuttsRec = "select recommendation
from recommendation where recid=? and type='putts' and level=?";
      private final static String getShotsRec = "select recommendation
from recommendation where recid=? and type='shots' and level=?";
      private final static String getPitchRec = "select recommendation
from recommendation where recid=? and type='pitch' and level=?";

      public int handicap;

      /*
            Constructor : recommendation
            Author   : Bjarni Gunnar
            Description : This constructor finds the level of the
golfer by checking his handicap.
                                Then it queries the database with
recommendation id and the level
                                of the golfer to get the
recommendation. It then prints out the
                                recommendation.
            Parameters : int putt, int driv, int shot, int pitc, float
hand
            Returns:
```

```
   */

      public recommendation(int putt, int driv, int shot, int pitc,
float hand)
      {
            //resets the connection
            Connection conn = null;

            // Check the level of the golfer by checking his handicap
            // Stores the level the in handicap
            if(hand <= 10)
            {
                  handicap = 1;
            }
            if(hand > 10 && hand <= 20)
            {
                  handicap = 2;
            }
            if(hand > 20 && hand <= 42)
            {
                  handicap = 3;
            }

            try
            {
                  //gets connection to the database
                  conn = DB.getConnectionToDB();

                  //stores the value of handicap in hcp
                  int hcp = handicap;

                  //Queries the database with the Prepared Statement,
the type and level
                  PreparedStatement puttstmt =
conn.prepareStatement(getPuttsRec);
                  puttstmt.setInt(1, putt);
                  puttstmt.setInt(2, hcp);

                  // Fetches each row from the Result Set putts and
puts them into variables
                  ResultSet putts = puttstmt.executeQuery();

                  // It is fetching the recommendation and prints it
out
                  while (putts.next())
                  {
                        String puttrecommend =
putts.getString("recommendation");
                        System.out.println(puttrecommend);
                  }

                  //Queries the database with the Prepared Statement,
the type and level
                  PreparedStatement drivestmt =
conn.prepareStatement(getDriveRec);
                  drivestmt.setInt(1, driv);
                  drivestmt.setInt(2, hcp);
                  // Fetches each row from the Result Set drive and
puts them into variable
                  ResultSet drive = drivestmt.executeQuery();

                  // It is fetching the recommendation and prints it
out
                  while (drive.next())
                  {
                        String driverecommend =
drive.getString("recommendation");
                        System.out.println(driverecommend);
                  }

                  //Queries the database with the Prepared Statement,
the type and level
                  PreparedStatement shotstmt =
conn.prepareStatement(getShotsRec);
                  shotstmt.setInt(1, shot);
                  shotstmt.setInt(2, hcp);

                  // Fetches each row from the Result Set shots and
puts them into variable
                  ResultSet shots = shotstmt.executeQuery();

                  // It is fetching the recommendations and printing
them out
                  while (shots.next())
                  {
                        String shotrecommend =
shots.getString("recommendation");
                        System.out.println(shotrecommend);
                  }

                  //Queries the database with the Prepared Statement,
the type and level
                  PreparedStatement pitchstmt =
conn.prepareStatement(getPitchRec);
                  pitchstmt.setInt(1, pitc);
                  pitchstmt.setInt(2, hcp);

                  // Fetches each row from the Result Set pitch and
puts them into variable
                  ResultSet pitch = pitchstmt.executeQuery();

                  // It is fetching the recommendations and printing
them out
                  while (pitch.next())
                  {
                        String pitchrecommend =
pitch.getString("recommendation");
                        System.out.println(pitchrecommend);
                  }

                  //if not disconnection then closes the connection to
the database
                  if(conn != null)
                  {
                        conn.close();
                  }
            }
            catch (SQLException e)
            {
            }
      }
}
```

DB.java

```
import java.sql.*;

/**
 * Created by IntelliJ IDEA.
 * User: Bjarni
 * Date: Feb 26, 2004
 * Time: 9:38:40 PM
 * To change this template use Options | File Templates.
 */
public class DB {

    public static java.sql.Connection getConnectionToDB() {
        Connection conn = null;
        try  {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            conn =
DriverManager.getConnection("jdbc:mysql://localhost/bjarni?user=bjarni&
password=BgB123");
            conn.setAutoCommit(false);
        } catch (InstantiationException e) {
            e.printStackTrace();  //To change body of catch statement
use Options | File Templates.
        } catch (IllegalAccessException e) {
            e.printStackTrace();  //To change body of catch statement
use Options | File Templates.
        } catch (ClassNotFoundException e) {
            e.printStackTrace();  //To change body of catch statement
use Options | File Templates.
        } catch (SQLException e) {
            e.printStackTrace();  //To change body of catch statement
use Options | File Templates.
        }

        return conn;
    }

    public void ConnectionClose(Connection conn) {
        if(conn != null)
        {
            try
            {
                conn.close();
            }
            catch (SQLException e)
            {
                e.printStackTrace();  //To change body of catch
statement use Options | File Templates.
            }
        }
    }


}
```

avg_stats.php

```
<html>
<head>
            <title>Average stats for players</title>
</head>

<body>
<h2>Pick a player to calculate his average statistics</h2>

<?
//Including the connection to the database
include("db_connect.php");

//This is querying the database for playerid and firstname
$sql1="SELECT playerid, firstname FROM player";
$result1=mysql_query($sql1);

$options1="";

//This put the result into an array and fetch the playerid and
firstname
//until all player have been fetched. This is then used in a combo box
while ($row1 = mysql_fetch_array($result1)) {

    $id1=$row1["playerid"];
    $thing1=$row1["firstname"];
    $options1.="<OPTION VALUE=\"$id1\">$thing1</option>";
}
?>

    <form method="post" action="avg_stats_entry.php">
    <span>Player:</span>
    <select name="playerid">
    <option value="0">
    <?=$options1?>
    </SELECT>

    <input type="submit" name="submit" value="Calculate"/>

    </form>

    <?php
            //include recommendation.php in the document
            include("recommendation.php");
    ?>
</body>
</html>
```

avg_stats_entry.php

```
<?php
//Connects to the database
include("db_connect.php");

//Puts in the Cascading Stylesheet
echo "<link rel=\"Stylesheet\" type=\"text/css\" href=\"new.css\" />
";

//Takes in the posted data from avg_stats.php and stores in variable
$playerid = $_POST['playerid'];

//Queries the database to check whether the player has data in the
stats table
$inputted_rounds_query = mysql_query("SELECT * FROM stats WHERE
playerid='$playerid'");
$inputted_rounds_check = mysql_num_rows($inputted_rounds_query);

//If the player has data in the stats table then continue
if($inputted_rounds_check > 0)
{
    //calculating the average of all the statistics the chosen player
has

    //average score
    $avg_score_query = mysql_query("select avg(score*18) from stats
where playerid=$playerid") or die (mysql_error());
    $avg_score = mysql_result($avg_score_query, 0);
    //average putts
    $avg_putts_query = mysql_query("select avg(putts*18) from stats
where playerid=$playerid") or die (mysql_error());
    $avg_putts = mysql_result($avg_putts_query, 0);

    //average bunker
    $avg_bunker_query = mysql_query("select avg(bunker*18) from stats
where playerid=$playerid") or die (mysql_error());
    $avg_bunker = mysql_result($avg_bunker_query, 0);

    //average penalty
    $avg_penalty_query = mysql_query("select avg(penalty*18) from stats
where playerid=$playerid") or die (mysql_error());
    $avg_penalty = mysql_result($avg_penalty_query, 0);

    //average green hit
    $avg_green_hit_query = mysql_query("select avg(green_hit*18) from
stats where playerid=$playerid") or die (mysql_error());
    $avg_green_hit = mysql_result($avg_green_hit_query, 0);

    //average green missed left
    $avg_green_missed_L_query = mysql_query("select
avg(green_missed_L*18) from stats where playerid=$playerid") or die
(mysql_error());
    $avg_green_missed_L = mysql_result($avg_green_missed_L_query, 0);

    //average green missed right
    $avg_green_missed_R_query = mysql_query("select
avg(green_missed_R*18) from stats where playerid=$playerid") or die
(mysql_error());
    $avg_green_missed_R = mysql_result($avg_green_missed_R_query, 0);

    //average green missed short
    $avg_green_missed_S_query = mysql_query("select
avg(green_missed_S*18) from stats where playerid=$playerid") or die
(mysql_error());
    $avg_green_missed_S = mysql_result($avg_green_missed_S_query, 0);

    //average green missed over
    $avg_green_missed_O_query = mysql_query("select
avg(green_missed_O*18) from stats where playerid=$playerid") or die
(mysql_error());
    $avg_green_missed_O = mysql_result($avg_green_missed_O_query, 0);

    //average green missed total
    $avg_green_missed_Total_query = mysql_query("select avg(18-
(green_hit*18)) from stats where playerid=$playerid") or die
(mysql_error());
    $avg_green_missed_Total =
mysql_result($avg_green_missed_Total_query, 0);

    //average shot 0-50m missed
    $avg_shot0_50m_query = mysql_query("select avg(shot0_50m_missed*18)
from stats where playerid=$playerid") or die (mysql_error());
    $avg_shot0_50m = mysql_result($avg_shot0_50m_query, 0);

    //average shot 50-100m missed
    $avg_shot50_100m_query = mysql_query("select
avg(shot50_100m_missed*18) from stats where playerid=$playerid") or die
(mysql_error());
    $avg_shot50_100m = mysql_result($avg_shot50_100m_query, 0);

    //average shot 100-150m missed
    $avg_shot100_150m_query = mysql_query("select
avg(shot100_150m_missed*18) from stats where playerid=$playerid") or
die (mysql_error());
    $avg_shot100_150m = mysql_result($avg_shot100_150m_query, 0);

    //average putts 0-1m missed
    $avg_putts0_1m_query = mysql_query("select avg(putts0_1m*18) from
stats where playerid=$playerid") or die (mysql_error());
    $avg_putts0_1m = mysql_result($avg_putts0_1m_query, 0);

    //average putts 1-2m missed
    $avg_putts1_2m_query = mysql_query("select avg(putts1_2m*18) from
stats where playerid=$playerid") or die (mysql_error());
    $avg_putts1_2m = mysql_result($avg_putts1_2m_query, 0);

    //average putts 2-3m missed
    $avg_putts2_3m_query = mysql_query("select avg(putts2_3m*18) from
stats where playerid=$playerid") or die (mysql_error());
    $avg_putts2_3m = mysql_result($avg_putts2_3m_query, 0);

    //calculate the count of all played hole except par 3
    $drive_holes_query = mysql_query("select (count(stats.hole_nr)/2)
from stats, holes where stats.courseid=holes.courseid and
playerid=$playerid and par!=3 and stats.hole_nr=holes.hole_nr") or die
(mysql_error());
    $drive_holes = mysql_result($drive_holes_query, 0);

    //count the total rounds played
    $played_rounds_query = mysql_query("select (count(golfroundid)/18)
from stats where playerid=$playerid") or die (mysql_error());
    $played_rounds = mysql_result($played_rounds_query, 0);

    //average of drive holes a round, used instead of 18 for the other
average statistic
    $drive_count = $drive_holes/$played_rounds;

    //average drive hit
    $avg_drive_hit_query = mysql_query("select
avg((drive_hit)*$drive_count) from stats, holes where
stats.courseid=holes.courseid and playerid=$playerid and par!=3 and
stats.hole_nr=holes.hole_nr") or die (mysql_error());
    $avg_drive_hit = mysql_result($avg_drive_hit_query, 0);

    //average drive missed right
    $avg_drive_missed_L_query = mysql_query("select
avg((drive_missed_L)*$drive_count) from stats, holes where
stats.courseid=holes.courseid and playerid=$playerid and par!=3 and
stats.hole_nr=holes.hole_nr") or die (mysql_error());
    $avg_drive_missed_L = mysql_result($avg_drive_missed_L_query, 0);

    //average drive missed left
    $avg_drive_missed_R_query = mysql_query("select
avg((drive_missed_R)*$drive_count) from stats, holes where
stats.courseid=holes.courseid and playerid=$playerid and par!=3 and
stats.hole_nr=holes.hole_nr") or die (mysql_error());
    $avg_drive_missed_R = mysql_result($avg_drive_missed_R_query, 0);

    //average drive missed total
    $avg_drive_missed_Total_query = mysql_query("select avg((1-
drive_hit)*$drive_count) from stats, holes where
stats.courseid=holes.courseid and playerid=$playerid and par!=3 and
stats.hole_nr=holes.hole_nr") or die (mysql_error());
    $avg_drive_missed_Total =
mysql_result($avg_drive_missed_Total_query, 0);

    //an insert SQL statement to put the average statistics of a player
in for the first time
    $insertsql = "INSERT INTO avg_stats (playerid, avg_score,
avg_putts, avg_bunker, avg_green_hit, avg_green_missed_L,
avg_green_missed_R, avg_green_missed_S, avg_green_missed_O,
avg_green_missed_Total, avg_shot0_50m_missed, avg_shot50_100m_missed,
avg_shot100_150m_missed, avg_penalty, avg_putts0_1m, avg_putts1_2m,
avg_putts2_3m, avg_drive_hit, avg_drive_missed_L, avg_drive_missed_R,
avg_drive_missed_Total) VALUES ('$playerid', $avg_score, $avg_putts,
$avg_bunker, $avg_green_hit, $avg_green_missed_L, $avg_green_missed_R,
$avg_green_missed_S, $avg_green_missed_O, $avg_green_missed_Total,
$avg_shot0_50m, $avg_shot50_100m, $avg_shot100_150m, $avg_penalty,
$avg_putts0_1m, $avg_putts1_2m, $avg_putts2_3m, $avg_drive_hit,
$avg_drive_missed_L, $avg_drive_missed_R, $avg_drive_missed_Total)";

    //an update SQL statement to update an average statistics which is
already in the database
    $updatesql = "UPDATE avg_stats SET playerid=$playerid,
avg_score=$avg_score, avg_putts=$avg_putts, avg_bunker=$avg_bunker,
avg_green_hit=$avg_green_hit, avg_green_missed_L=$avg_green_missed_L,
avg_green_missed_R=$avg_green_missed_R,
avg_green_missed_S=$avg_green_missed_S,
avg_green_missed_O=$avg_green_missed_O,
avg_green_missed_Total=$avg_green_missed_Total,
avg_shot0_50m_missed=$avg_shot0_50m,
avg_shot50_100m_missed=$avg_shot50_100m,
avg_shot100_150m_missed=$avg_shot100_150m, avg_penalty=$avg_penalty,
avg_putts0_1m=$avg_putts0_1m, avg_putts1_2m=$avg_putts1_2m,
avg_putts2_3m=$avg_putts2_3m, avg_drive_hit=$avg_drive_hit,
avg_drive_missed_L=$avg_drive_missed_L,
avg_drive_missed_R=$avg_drive_missed_R,
avg_drive_missed_Total=$avg_drive_missed_Total where
playerid=$playerid";

    //if player was chosen in the avg_stats.php then continue
    if ($playerid != "")
    {
      //querying the database checking whether the player has data here
or not
        $res = mysql_query("select playerid from avg_stats where
playerid=$playerid");

      //if he doesn't then use the insert SQL statement
        if (mysql_num_rows($res) != 1)
        {
                        $query = $insertsql;

            mysql_query($query) or
                die (mysql_error());
        }
      //else use the update SQL statement
        else
        {
                        $query = $updatesql;

                        mysql_query($query) or
            die (mysql_error());
        }

    ?>

    <?php
            //querying and stores the name of the player
            $name_query = mysql_query("select firstname from player
where playerid=$playerid");
            $name = mysql_result($name_query, 0);

            //querying for the average statistics of the player
```

```
             $output_result = mysql_query("select * from avg_stats
where playerid=$playerid") or
             die (mysql_error());

             echo "<h2>$name\n</h2>";
             // start table
             echo "<table>";

      //fetch information from the avg_stats table and store in
variables
        while ($row = mysql_fetch_array($output_result))
      {
            $avg_score_output = $row['avg_score'];
            $avg_putts_output = $row['avg_putts'];
            $avg_drive_hit_output = $row['avg_drive_hit'];
            $avg_drive_missed_L_output = $row['avg_drive_missed_L'];
            $avg_drive_missed_R_output = $row['avg_drive_missed_R'];
            $avg_drive_missed_Total_output =
$row['avg_drive_missed_Total'];
            $avg_green_hit_output = $row['avg_green_hit'];
            $avg_green_missed_L_output = $row['avg_green_missed_L'];
            $avg_green_missed_R_output = $row['avg_green_missed_R'];
            $avg_green_missed_S_output = $row['avg_green_missed_S'];
            $avg_green_missed_O_output = $row['avg_green_missed_O'];
            $avg_green_missed_Total_output =
$row['avg_green_missed_Total'];
            $avg_shot0_50m_output = $row['avg_shot0_50m_missed'];
            $avg_shot50_100m_output = $row['avg_shot50_100m_missed'];
            $avg_shot100_150m_output = $row['avg_shot100_150m_missed'];
            $avg_putts0_1m_output = $row['avg_putts0_1m'];
            $avg_putts1_2m_output = $row['avg_putts1_2m'];
            $avg_putts2_3m_output = $row['avg_putts2_3m'];

            //make a table row and column
                  echo "<tr>\n";
                  echo "<td>Score</td>\n";
                  echo "<td>Putts</td>\n";
                  echo "<td>Drives hit</td>\n";
                  echo "<td>Drives m.Left</td>\n";
                  echo "<td>Drives m.Right</td>\n";
                  echo "<td>Drives m.Total</td>\n";
                  echo "<td>GIR</td>\n";
                  echo "<td>GIR m.Left</td>\n";
                  echo "<td>GIR m.Right</td>\n";
                  echo "<td>GIR m.Short</td>\n";
                  echo "<td>GIR m.Over</td>\n";
                  echo "<td>GIR m.Total</td>\n";
                  echo "<td>Shot 0-50m m.</td>\n";
                  echo "<td>Shot 50-100m m.</td>\n";
                  echo "<td>Shot 100-150m m.</td>\n";
                  echo "<td>Putts 0-1m m.</td>\n";
                  echo "<td>Putts 1-2m m.</td>\n";
                  echo "<td>Putts 2-3m m.</td>\n";
                  echo "</tr>\n";

            //make a table row and display the average statistic in the
columns
```

```
            echo "<tr>\n";
                  echo "<td>$avg_score_output</td>\n";
                  echo "<td>$avg_putts_output</td>\n";
                  echo "<td>$avg_drive_hit_output</td>\n";
                  echo "<td>$avg_drive_missed_L_output</td>\n";
                  echo "<td>$avg_drive_missed_R_output</td>\n";
                  echo "<td>$avg_drive_missed_Total_output</td>\n";
                  echo "<td>$avg_green_hit_output</td>\n";
                  echo "<td>$avg_green_missed_L_output</td>\n";
                  echo "<td>$avg_green_missed_R_output</td>\n";
                  echo "<td>$avg_green_missed_S_output</td>\n";
                  echo "<td>$avg_green_missed_O_output</td>\n";
                  echo "<td>$avg_green_missed_Total_output</td>\n";
                  echo "<td>$avg_shot0_50m_output</td>\n";
                  echo "<td>$avg_shot50_100m_output</td>\n";
                  echo "<td>$avg_shot100_150m_output</td>\n";
                  echo "<td>$avg_putts0_1m_output</td>\n";
                  echo "<td>$avg_putts1_2m_output</td>\n";
                  echo "<td>$avg_putts2_3m_output</td>\n";
                  echo "</tr>\n";
        }

      // End the table
      echo "</table>\n";

      mysql_free_result($output_result);
      echo "<a href=\"javascript:history.back(0)\">Back</a>";
      }
      //else go back and pick a player
      else
      {
            echo "<p>You have to pick a player</p>";
            include("avg_stats.php");
      }
}
//else the player has not put in any round into the system
else
{
      echo "<span>This player has not put any rounds into the
system</span>";
      echo "<span>Please pick another player or input some
statistics</span>";
      include("avg_stats.php");
}
?>
```

Border.php

```
<html>
<head>

  <title></title>
  <link rel="stylesheet" type="text/css" href="new.css" />
  <base target="contents">
</head>
```

```
<table border="0" width="100%" height="59" cellspacing="0"
cellpadding="0" style="border-collapse: collapse">
  <tr>
    <td><img border="0" src="DT_Solo.jpg"></td>
    <td class="big">The Golfhelper</td>
  </tr>

</table>

</body>
</html>
```

Course.php

```
<html>
<head>
      <title>Golfcourse input</title>
      <link rel="stylesheet" href="new.css" />
</head>

<body>
<h2>Insert a new course</h2>

    <form method="post" action="course_entry.php">
    <table>
    <tr>
      <td><span>Name:</span></td>
      <td><input type="text" size="30" name="name"/></td>
    </tr>
    <tr>
      <td><span>Location:</span></td>
      <td><input type="text" size="30" name="location"/></td>
    </tr>
    </table>
    <input type="submit" name="submit" value="Input"/>
    <input type="reset" name="reset" value="Clear"/>

    </form>

</body>
</html>
```

Course_entry.php

```
<?php
//Connects to the database
include("db_connect.php");

//Takes in the posted data from course.php and stores in variables
$name = $_POST['name'];
$location = $_POST['location'];

//A SQL insert statement into the course table
$sql = "INSERT INTO course (name, location) VALUES ('$name',
'$location')";

//Queries the database to check whether the course is in the course
table
```

```
$login_sql_check = mysql_query("SELECT * FROM course WHERE
name='$name'");
      $login_check = mysql_num_rows($login_sql_check);

//If he isn't the go ahead
if($login_check == 0)
{
      //If both name and location have data into the course into the
course table
      if ($name != "" && $location != "")
      {
            $query = $sql;

            mysql_query($query) or
                        die (mysql_error());

      echo "<p>The course has been stored in the database</p>";
      include("tees.php");
      }
      //else displays course.php again and asks to fill in all
information
      else
      {
            echo "<p>All boxes have to be filled out</p>";
            include("course.php");
      }
}
//else, course already in the course table
else
{
      echo "<p>The course name already exists</p>";
      include("course.php");
}
?>

Db_connect.php
<? @mysql_connect('localhost', 'bjarni', 'BgB123') ?
mysql_select_db('bjarni') : die(mysql_error()); ?>
```

Golf.php

```
<html>
<head>
            <title>Input golf statistics</title>
          <link rel="stylesheet" href="new.css" />

<script language="Javascript">
    //here are the functions that calculate the total in the total
boxes
    //the name of the functions as for what row they are
    function calcscore()
    {
      var valscore, totalscore;
      totalscore=0;

      for(i=1;i<19;i++)
      {
          valscore=document.forms[0].elements["score"+i].value;
          if(parseInt(valscore))
          {
            totalscore=totalscore+parseInt(valscore);
          }
      }

      document.forms[0].elements["totalscore"].value=totalscore;
    }
    function calcputts()
    {
      var valputts, totalputts;
      totalputts=0;

      for(i=1;i<19;i++)
      {
          valputts=document.forms[0].elements["putt"+i].value;
          if(parseInt(valputts))
          {
            totalputts=totalputts+parseInt(valputts);
          }
      }

      document.forms[0].elements["totalputts"].value=totalputts;
    }
    function calcdrives()
    {
      var valdrives, totaldrives;
      totaldrives=0;

      for(i=1;i<19;i++)
      {
          valdrives=document.forms[0].elements["drive"+i].value;
          if(parseInt(valdrives))
          {
            totaldrives=totaldrives+parseInt(valdrives);
          }
      }

      document.forms[0].elements["totaldrivehit"].value=totaldrives;
    }
    function calcdriveL()
    {
      var valdrives, totaldrives;
      totaldrives=0;

      for(i=1;i<19;i++)
      {
          valdrives=document.forms[0].elements["drive_L"+i].value;
          if(parseInt(valdrives))
          {
            totaldrives=totaldrives+parseInt(valdrives);
          }
      }

      document.forms[0].elements["totaldriveL"].value=totaldrives;
    }
```

```
function calcdriveR()
{
  var valdrives, totaldrives;
  totaldrives=0;

  for(i=1;i<19;i++)
  {
      valdrives=document.forms[0].elements["drive_R"+i].value;
      if(parseInt(valdrives))
      {
        totaldrives=totaldrives+parseInt(valdrives);
      }
  }

  document.forms[0].elements["totaldriveR"].value=totaldrives;
}
function calcgir()
{
  var valgir, totalgir;
  totalgir=0;

  for(i=1;i<19;i++)
  {
      valgir=document.forms[0].elements["gir"+i].value;
      if(parseInt(valgir))
      {
        totalgir=totalgir+parseInt(valgir);
      }
  }

  document.forms[0].elements["totalgir"].value=totalgir;
}
function calcgirL()
{
  var valgir, totalgir;
  totalgir=0;

  for(i=1;i<19;i++)
  {
      valgir=document.forms[0].elements["gir_L"+i].value;
      if(parseInt(valgir))
      {
        totalgir=totalgir+parseInt(valgir);
      }
  }

  document.forms[0].elements["totalgirL"].value=totalgir;
}
function calcgirR()
{
  var valgir, totalgir;
  totalgir=0;

  for(i=1;i<19;i++)
  {
      valgir=document.forms[0].elements["gir_R"+i].value;
      if(parseInt(valgir))
```

```
        {
          totalgir=totalgir+parseInt(valgir);
        }
    }

    document.forms[0].elements["totalgirR"].value=totalgir;
  }
  function calcgirS()
  {
    var valgir, totalgir;
    totalgir=0;

    for(i=1;i<19;i++)
    {
        valgir=document.forms[0].elements["gir_S"+i].value;
        if(parseInt(valgir))
        {
          totalgir=totalgir+parseInt(valgir);
        }
    }

    document.forms[0].elements["totalgirS"].value=totalgir;
  }
  function calcgirO()
  {
    var valgir, totalgir;
    totalgir=0;

    for(i=1;i<19;i++)
    {
        valgir=document.forms[0].elements["gir_O"+i].value;
        if(parseInt(valgir))
        {
          totalgir=totalgir+parseInt(valgir);
        }
    }

    document.forms[0].elements["totalgirO"].value=totalgir;
  }
  function calcbunker()
  {
    var valbunker, totalbunker;
    totalbunker=0;

    for(i=1;i<19;i++)
    {
        valbunker=document.forms[0].elements["bunker"+i].value;
        if(parseInt(valbunker))
        {
          totalbunker=totalbunker+parseInt(valbunker);
        }
    }

    document.forms[0].elements["totalbunker"].value=totalbunker;
  }
  function calcpen()
  {
    var valpenalty, totalpenalty;
    totalpenalty=0;

    for(i=1;i<19;i++)
    {
        valpenalty=document.forms[0].elements["pen"+i].value;
        if(parseInt(valpenalty))
        {
          totalpenalty=totalpenalty+parseInt(valpenalty);
        }
    }

    document.forms[0].elements["totalpen"].value=totalpenalty;
  }
  function calcs50m()
  {
    var valshots, totalshots;
    totalshots=0;

    for(i=1;i<19;i++)
    {
        valshots=document.forms[0].elements["shot0_50m"+i].value;
        if(parseInt(valshots))
        {
          totalshots=totalshots+parseInt(valshots);
        }
    }

    document.forms[0].elements["totals50m"].value=totalshots;
  }
  function calcs100m()
  {
    var valshots, totalshots;
    totalshots=0;

    for(i=1;i<19;i++)
    {
        valshots=document.forms[0].elements["shot50_100m"+i].value;
        if(parseInt(valshots))
        {
          totalshots=totalshots+parseInt(valshots);
        }
    }

    document.forms[0].elements["totals100m"].value=totalshots;
  }
  function calcs150m()
  {
    var valshots, totalshots;
    totalshots=0;

    for(i=1;i<19;i++)
    {
        valshots=document.forms[0].elements["shot100_150m"+i].value;
        if(parseInt(valshots))
        {
          totalshots=totalshots+parseInt(valshots);
          }
        }

        document.forms[0].elements["totals150m"].value=totalshots;
      }
      function calcputt1m()
      {
        var valputts, totalputts;
        totalputts=0;

        for(i=1;i<19;i++)
        {
            valputts=document.forms[0].elements["putt0_1m"+i].value;
            if(parseInt(valputts))
            {
              totalputts=totalputts+parseInt(valputts);
            }
        }

        document.forms[0].elements["totalputt1m"].value=totalputts;
      }
      function calcputt2m()
      {
        var valputts, totalputts;
        totalputts=0;

        for(i=1;i<19;i++)
        {
            valputts=document.forms[0].elements["putt1_2m"+i].value;
            if(parseInt(valputts))
            {
              totalputts=totalputts+parseInt(valputts);
            }
        }

        document.forms[0].elements["totalputt2m"].value=totalputts;
      }
      function calcputt3m()
      {
        var valputts, totalputts;
        totalputts=0;

        for(i=1;i<19;i++)
        {
            valputts=document.forms[0].elements["putt2_3m"+i].value;
            if(parseInt(valputts))
            {
              totalputts=totalputts+parseInt(valputts);
            }
        }

        document.forms[0].elements["totalputt3m"].value=totalputts;
      }

</script>

</head>
```

```
<body>
<h2>Input your golf score</h2>

<?
include("db_connect.php");

$sql="SELECT courseid, name FROM course";
$result=mysql_query($sql);

$options="";

while ($row = mysql_fetch_array($result)) {

    $id=$row["courseid"];
    $thing=$row["name"];
    $options.="<OPTION VALUE=\"$id\">$thing</option>";
}
$sql1="SELECT playerid, firstname FROM player";
$result1=mysql_query($sql1);

$options1="";

while ($row1 = mysql_fetch_array($result1)) {

    $id1=$row1["playerid"];
    $thing1=$row1["firstname"];
    $options1.="<OPTION VALUE=\"$id1\">$thing1</option>";
}
$sql2="SELECT golfroundid, tournament FROM golfrounds";
$result2=mysql_query($sql2);

$options2="";

while ($row2 = mysql_fetch_array($result2)) {

    $id2=$row2["golfroundid"];
    $thing2=$row2["tournament"];
    $options2.="<OPTION VALUE=\"$id2\">$thing2</option>";
}
?>

    <form name="statistics" method="post" action="golf_entry.php">
    <table>
    <tr>
      <td><span>Course:</span></td>
      <td><select name="courseid"><option
value="0"><?=$options?></select></td>
      <td><a href ="course.php">New course</a></td>
    </tr>
    <tr>
      <td><span>Tees:</span></td>
      <td>
          <select name="colour">
          <option value="white">White</option>
          <option value="yellow">Yellow</option>
          <option value="blue">Blue</option>
          <option value="red">Red</option>
      </td>
      <td><a href ="tees.php">New tee</a></td>
    </tr>
    <tr>
      <td><span>Player:</span></td>
      <td><select name="playerid"><option
value="0"><?=$options1?></select></td>
      <td><a href ="player.php">New player</a></td>
    </tr>
    <tr>
      <td><span>Rounds:</span></td>
      <td><select name="golfroundid"><option
value="0"><?=$options2?></select></td>
      <td><a href ="round.php">New round</a></td>
    </tr>
    </table>

    <table>
          <tr>
                <td>Hole</td>
                <td>1</td>
                <td>2</td>
                <td>3</td>
                <td>4</td>
                <td>5</td>
                <td>6</td>
                <td>7</td>
                <td>8</td>
                <td>9</td>
                <td>10</td>
                <td>11</td>
                <td>12</td>
                <td>13</td>
                <td>14</td>
                <td>15</td>
                <td>16</td>
                <td>17</td>
                <td>18</td>
                <td>Total</td>
            </tr>
            <tr>
                <td>Score</td>
            <td><input type="text" size="2" name="score1" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score2" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score3" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score4" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score5" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score6" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score7" value=""
onchange="calcscore()"/></td>
```

```
                <td><input type="text" size="2" name="score8" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score9" value=""
onchange="calcscore()"/></td>
            <td><input type="text" size="2" name="score10" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score11" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score12" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score13" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score14" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score15" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score16" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score17" value=""
onchange="calcscore()"/></td>
                <td><input type="text" size="2" name="score18" value=""
onchange="calcscore()"/></td>
                <td><input readonly="text" size="2"
name="totalscore"/></td>
            </tr>
            <tr>
                <td>Putts</td>
                <td><input type="text" size="2" name="putt1" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt2" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt3" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt4" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt5" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt6" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt7" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt8" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt9" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt10" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt11" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt12" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt13" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt14" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt15" value=""
onchange="calcputts()"/></td>
```

```
                <td><input type="text" size="2" name="putt16" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt17" value=""
onchange="calcputts()"/></td>
                <td><input type="text" size="2" name="putt18" value=""
onchange="calcputts()"/></td>
            <td><input readonly="text" size="2"
name="totalputts"/></td>
            </tr>
            <tr>
                <td>Drives hit</td>
                <td><input type="text" size="2" name="drive1" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive2" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive3" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive4" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive5" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive6" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive7" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive8" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive9" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive10" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive11" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive12" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive13" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive14" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive15" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive16" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive17" value=""
onchange="calcdrives()"/></td>
                <td><input type="text" size="2" name="drive18" value=""
onchange="calcdrives()"/></td>
            <td><input readonly="text" size="2"
name="totaldrivehit"/></td>
            </tr>
            <tr>
                <td>Drives missed left</td>
                <td><input type="text" size="2" name="drive_L1"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L2"
value="" onchange="calcdriveL()"/></td>
```

```
                <td><input type="text" size="2" name="drive_L3"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L4"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L5"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L6"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L7"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L8"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L9"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L10"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L11"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L12"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L13"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L14"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L15"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L16"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L17"
value="" onchange="calcdriveL()"/></td>
                <td><input type="text" size="2" name="drive_L18"
value="" onchange="calcdriveL()"/></td>
            <td><input readonly="text" size="2"
name="totaldriveL"/></td>
            </tr>
            <tr>
                <td>Drives missed right</td>
                <td><input type="text" size="2" name="drive_R1"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R2"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R3"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R4"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R5"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R6"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R7"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R8"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R9"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R10"
value="" onchange="calcdriveR()"/></td>
```

```
                <td><input type="text" size="2" name="drive_R11"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R12"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R13"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R14"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R15"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R16"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R17"
value="" onchange="calcdriveR()"/></td>
                <td><input type="text" size="2" name="drive_R18"
value="" onchange="calcdriveR()"/></td>
            <td><input readonly="text" size="2"
name="totaldriveR"/></td>
            </tr>
            <tr>
                <td>Green hit</td>
                <td><input type="text" size="2" name="gir1" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir2" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir3" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir4" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir5" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir6" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir7" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir8" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir9" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir10" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir11" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir12" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir13" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir14" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir15" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir16" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir17" value=""
onchange="calcgir()"/></td>
                <td><input type="text" size="2" name="gir18" value=""
onchange="calcgir()"/></td>
```

```
            <td><input readonly="text" size="2" name="totalgir"/></td>
            </tr>
            <tr>
                <td>Green missed Left</td>
                <td><input type="text" size="2" name="gir_L1" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L2" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L3" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L4" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L5" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L6" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L7" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L8" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L9" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L10" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L11" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L12" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L13" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L14" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L15" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L16" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L17" value=""
onchange="calcgirL()"/></td>
                <td><input type="text" size="2" name="gir_L18" value=""
onchange="calcgirL()"/></td>
            <td><input readonly="text" size="2" name="totalgirL"/></td>
            </tr>
            <tr>
                <td>Green missed Right</td>
                <td><input type="text" size="2" name="gir_R1" value=""
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R2" value=""
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R3" value=""
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R4" value=""
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R5" value=""
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R6" value=""
onchange="calcgirR()"/></td>
```

```
                <td><input type="text" size="2" name="gir_R7" value="" 
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R8" value="" 
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R9" value="" 
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R10" value="" 
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R11" value="" 
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R12" value="" 
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R13" value="" 
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R14" value="" 
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R15" value="" 
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R16" value="" 
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R17" value="" 
onchange="calcgirR()"/></td>
                <td><input type="text" size="2" name="gir_R18" value="" 
onchange="calcgirR()"/></td>
            <td><input readonly="text" size="2" name="totalgirR"/></td>
            </tr>
            <tr>
                <td>Green missed Short</td>
                <td><input type="text" size="2" name="gir_S1" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S2" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S3" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S4" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S5" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S6" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S7" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S8" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S9" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S10" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S11" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S12" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S13" value="" 
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S14" value="" 
onchange="calcgirS()"/></td>
```

```
                <td><input type="text" size="2" name="gir_S15" value=""
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S16" value=""
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S17" value=""
onchange="calcgirS()"/></td>
                <td><input type="text" size="2" name="gir_S18" value=""
onchange="calcgirS()"/></td>
            <td><input readonly="text" size="2" name="totalgirS"/></td>
            </tr>
            <tr>
                <td>Green missed Over</td>
                <td><input type="text" size="2" name="gir_O1" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O2" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O3" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O4" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O5" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O6" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O7" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O8" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O9" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O10" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O11" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O12" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O13" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O14" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O15" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O16" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O17" value=""
onchange="calcgirO()"/></td>
                <td><input type="text" size="2" name="gir_O18" value=""
onchange="calcgirO()"/></td>
            <td><input readonly="text" size="2" name="totalgirO"/></td>
            </tr>
            <tr>
                <td>Bunker</td>
                <td><input type="text" size="2" name="bunker1" value=""
onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker2" value=""
onchange="calcbunker()"/></td>
```

```
                <td><input type="text" size="2" name="bunker3" value=""
onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker4" value=""
onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker5" value=""
onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker6" value=""
onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker7" value=""
onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker8" value=""
onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker9" value=""
onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker10"
value="" onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker11"
value="" onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker12"
value="" onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker13"
value="" onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker14"
value="" onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker15"
value="" onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker16"
value="" onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker17"
value="" onchange="calcbunker()"/></td>
                <td><input type="text" size="2" name="bunker18"
value="" onchange="calcbunker()"/></td>
            <td><input readonly="text" size="2"
name="totalbunker"/></td>
            </tr>
            <tr>
                <td>Penalty</td>
                <td><input type="text" size="2" name="pen1" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen2" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen3" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen4" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen5" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen6" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen7" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen8" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen9" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen10" value=""
onchange="calcpen()"/></td>
```

```
                <td><input type="text" size="2" name="pen11" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen12" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen13" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen14" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen15" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen16" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen17" value=""
onchange="calcpen()"/></td>
                <td><input type="text" size="2" name="pen18" value=""
onchange="calcpen()"/></td>
            <td><input readonly="text" size="2" name="totalpen"/></td>
            </tr>
            <tr>
                <td>Shot 0-50m missed</td>
                <td><input type="text" size="2" name="shot0_50m1"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m2"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m3"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m4"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m5"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m6"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m7"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m8"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m9"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m10"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m11"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m12"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m13"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m14"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m15"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m16"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m17"
value="" onchange="calcs50m()"/></td>
                <td><input type="text" size="2" name="shot0_50m18"
value="" onchange="calcs50m()"/></td>
            <td><input readonly="text" size="2" name="totals50m"/></td>
            </tr>
            <tr>
                <td>Shot 50-100m missed</td>
                <td><input type="text" size="2" name="shot50_100m1"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m2"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m3"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m4"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m5"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m6"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m7"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m8"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m9"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m10"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m11"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m12"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m13"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m14"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m15"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m16"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m17"
value="" onchange="calcs100m()"/></td>
                <td><input type="text" size="2" name="shot50_100m18"
value="" onchange="calcs100m()"/></td>
            <td><input readonly="text" size="2"
name="totals100m"/></td>
            </tr>
            <tr>
                <td>Shot 100-150m missed</td>
                <td><input type="text" size="2" name="shot100_150m1"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m2"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m3"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m4"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m5"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m6"
value="" onchange="calcs150m()"/></td>
```

```
                <td><input type="text" size="2" name="shot100_150m7"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m8"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m9"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m10"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m11"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m12"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m13"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m14"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m15"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m16"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m17"
value="" onchange="calcs150m()"/></td>
                <td><input type="text" size="2" name="shot100_150m18"
value="" onchange="calcs150m()"/></td>
            <td><input readonly="text" size="2"
name="totals150m"/></td>
            </tr>
            <tr>
                <td>Putts 0-1m missed</td>
                <td><input type="text" size="2" name="putt0_1m1"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m2"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m3"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m4"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m5"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m6"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m7"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m8"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m9"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m10"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m11"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m12"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m13"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m14"
value="" onchange="calcputt1m()"/></td>
```

```
                <td><input type="text" size="2" name="putt0_1m15"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m16"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m17"
value="" onchange="calcputt1m()"/></td>
                <td><input type="text" size="2" name="putt0_1m18"
value="" onchange="calcputt1m()"/></td>
            <td><input readonly="text" size="2"
name="totalputt1m"/></td>
            </tr>
            <tr>
                <td>Putts 1-2m missed</td>
                <td><input type="text" size="2" name="putt1_2m1"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m2"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m3"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m4"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m5"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m6"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m7"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m8"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m9"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m10"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m11"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m12"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m13"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m14"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m15"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m16"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m17"
value="" onchange="calcputt2m()"/></td>
                <td><input type="text" size="2" name="putt1_2m18"
value="" onchange="calcputt2m()"/></td>
            <td><input readonly="text" size="2"
name="totalputt2m"/></td>
            </tr>
            <tr>
                <td>Putts 2-3m missed</td>
                <td><input type="text" size="2" name="putt2_3m1"
value="" onchange="calcputt3m()"/></td>
```

```
                <td><input type="text" size="2" name="putt2_3m2"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m3"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m4"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m5"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m6"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m7"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m8"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m9"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m10"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m11"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m12"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m13"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m14"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m15"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m16"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m17"
value="" onchange="calcputt3m()"/></td>
                <td><input type="text" size="2" name="putt2_3m18"
value="" onchange="calcputt3m()"/></td>
            <td><input readonly="text" size="2"
name="totalputt3m"/></td>
            </tr>
            <tr>
                <td>Hole</td>
                <td>1</td>
                <td>2</td>
                <td>3</td>
                <td>4</td>
                <td>5</td>
                <td>6</td>
                <td>7</td>
                <td>8</td>
                <td>9</td>
                <td>10</td>
                <td>11</td>
                <td>12</td>
                <td>13</td>
                <td>14</td>
                <td>15</td>
                <td>16</td>
                <td>17</td>
                <td>18</td>
```

```
          </tr>
        </table>

    <input type="submit" name="submit" value="Insert"/>
    <input type="reset" name="reset" value="Clear"/>

    </form>

</body>
</html>
```

Golf_entry.php

```
<?php
//Connects to the database
include("db_connect.php");

//Takes in the posted data from golf.php and stores in variables
$roundid = $_POST['golfroundid'];
$playerid = $_POST['playerid'];
$courseid = $_POST['courseid'];
$tee_colour = $_POST['colour'];

//the score
$s1 = $_POST['score1'];
$s2 = $_POST['score2'];
$s3 = $_POST['score3'];
$s4 = $_POST['score4'];
$s5 = $_POST['score5'];
$s6 = $_POST['score6'];
$s7 = $_POST['score7'];
$s8 = $_POST['score8'];
$s9 = $_POST['score9'];
$s10 = $_POST['score10'];
$s11 = $_POST['score11'];
$s12 = $_POST['score12'];
$s13 = $_POST['score13'];
$s14 = $_POST['score14'];
$s15 = $_POST['score15'];
$s16 = $_POST['score16'];
$s17 = $_POST['score17'];
$s18 = $_POST['score18'];
$total = $s1 + $s2 + $s3 + $s4 + $s5 + $s6 + $s7 + $s8 + $s9 + $s10 +
$s11 + $s12 + $s13 + $s14 + $s15 + $s16 + $s17 + $s18;


//the hole number
$h1 = 1;
$h2 = 2;
$h3 = 3;
$h4 = 4;
$h5 = 5;
$h6 = 6;
$h7 = 7;
$h8 = 8;
$h9 = 9;
$h10 = 10;
```

```
$h11 = 11;
$h12 = 12;
$h13 = 13;
$h14 = 14;
$h15 = 15;
$h16 = 16;
$h17 = 17;
$h18 = 18;


//the putts
$p1 = $_POST['putt1'];
$p2 = $_POST['putt2'];
$p3 = $_POST['putt3'];
$p4 = $_POST['putt4'];
$p5 = $_POST['putt5'];
$p6 = $_POST['putt6'];
$p7 = $_POST['putt7'];
$p8 = $_POST['putt8'];
$p9 = $_POST['putt9'];
$p10 = $_POST['putt10'];
$p11 = $_POST['putt11'];
$p12 = $_POST['putt12'];
$p13 = $_POST['putt13'];
$p14 = $_POST['putt14'];
$p15 = $_POST['putt15'];
$p16 = $_POST['putt16'];
$p17 = $_POST['putt17'];
$p18 = $_POST['putt18'];


//the drive hit
$d1 = $_POST['drive1'];
$d2 = $_POST['drive2'];
$d3 = $_POST['drive3'];
$d4 = $_POST['drive4'];
$d5 = $_POST['drive5'];
$d6 = $_POST['drive6'];
$d7 = $_POST['drive7'];
$d8 = $_POST['drive8'];
$d9 = $_POST['drive9'];
$d10 = $_POST['drive10'];
$d11 = $_POST['drive11'];
$d12 = $_POST['drive12'];
$d13 = $_POST['drive13'];
$d14 = $_POST['drive14'];
$d15 = $_POST['drive15'];
$d16 = $_POST['drive16'];
$d17 = $_POST['drive17'];
$d18 = $_POST['drive18'];

//the drive missed left
$d_L1 = $_POST['drive_L1'];
$d_L2 = $_POST['drive_L2'];
$d_L3 = $_POST['drive_L3'];
$d_L4 = $_POST['drive_L4'];
$d_L5 = $_POST['drive_L5'];
```

```
$d_L6 = $_POST['drive_L6'];
$d_L7 = $_POST['drive_L7'];
$d_L8 = $_POST['drive_L8'];
$d_L9 = $_POST['drive_L9'];
$d_L10 = $_POST['drive_L10'];
$d_L11 = $_POST['drive_L11'];
$d_L12 = $_POST['drive_L12'];
$d_L13 = $_POST['drive_L13'];
$d_L14 = $_POST['drive_L14'];
$d_L15 = $_POST['drive_L15'];
$d_L16 = $_POST['drive_L16'];
$d_L17 = $_POST['drive_L17'];
$d_L18 = $_POST['drive_L18'];

//the drive missed right
$d_R1 = $_POST['drive_R1'];
$d_R2 = $_POST['drive_R2'];
$d_R3 = $_POST['drive_R3'];
$d_R4 = $_POST['drive_R4'];
$d_R5 = $_POST['drive_R5'];
$d_R6 = $_POST['drive_R6'];
$d_R7 = $_POST['drive_R7'];
$d_R8 = $_POST['drive_R8'];
$d_R9 = $_POST['drive_R9'];
$d_R10 = $_POST['drive_R10'];
$d_R11 = $_POST['drive_R11'];
$d_R12 = $_POST['drive_R12'];
$d_R13 = $_POST['drive_R13'];
$d_R14 = $_POST['drive_R14'];
$d_R15 = $_POST['drive_R15'];
$d_R16 = $_POST['drive_R16'];
$d_R17 = $_POST['drive_R17'];
$d_R18 = $_POST['drive_R18'];

//the green hit
$g1 = $_POST['gir1'];
$g2 = $_POST['gir2'];
$g3 = $_POST['gir3'];
$g4 = $_POST['gir4'];
$g5 = $_POST['gir5'];
$g6 = $_POST['gir6'];
$g7 = $_POST['gir7'];
$g8 = $_POST['gir8'];
$g9 = $_POST['gir9'];
$g10 = $_POST['gir10'];
$g11 = $_POST['gir11'];
$g12 = $_POST['gir12'];
$g13 = $_POST['gir13'];
$g14 = $_POST['gir14'];
$g15 = $_POST['gir15'];
$g16 = $_POST['gir16'];
$g17 = $_POST['gir17'];
$g18 = $_POST['gir18'];

//the green missed left
$g_L1 = $_POST['gir_L1'];
$g_L2 = $_POST['gir_L2'];
```

```
$g_L3 = $_POST['gir_L3'];
$g_L4 = $_POST['gir_L4'];
$g_L5 = $_POST['gir_L5'];
$g_L6 = $_POST['gir_L6'];
$g_L7 = $_POST['gir_L7'];
$g_L8 = $_POST['gir_L8'];
$g_L9 = $_POST['gir_L9'];
$g_L10 = $_POST['gir_L10'];
$g_L11 = $_POST['gir_L11'];
$g_L12 = $_POST['gir_L12'];
$g_L13 = $_POST['gir_L13'];
$g_L14 = $_POST['gir_L14'];
$g_L15 = $_POST['gir_L15'];
$g_L16 = $_POST['gir_L16'];
$g_L17 = $_POST['gir_L17'];
$g_L18 = $_POST['gir_L18'];

//the green missed right
$g_R1 = $_POST['gir_R1'];
$g_R2 = $_POST['gir_R2'];
$g_R3 = $_POST['gir_R3'];
$g_R4 = $_POST['gir_R4'];
$g_R5 = $_POST['gir_R5'];
$g_R6 = $_POST['gir_R6'];
$g_R7 = $_POST['gir_R7'];
$g_R8 = $_POST['gir_R8'];
$g_R9 = $_POST['gir_R9'];
$g_R10 = $_POST['gir_R10'];
$g_R11 = $_POST['gir_R11'];
$g_R12 = $_POST['gir_R12'];
$g_R13 = $_POST['gir_R13'];
$g_R14 = $_POST['gir_R14'];
$g_R15 = $_POST['gir_R15'];
$g_R16 = $_POST['gir_R16'];
$g_R17 = $_POST['gir_R17'];
$g_R18 = $_POST['gir_R18'];

//the green missed short
$g_S1 = $_POST['gir_S1'];
$g_S2 = $_POST['gir_S2'];
$g_S3 = $_POST['gir_S3'];
$g_S4 = $_POST['gir_S4'];
$g_S5 = $_POST['gir_S5'];
$g_S6 = $_POST['gir_S6'];
$g_S7 = $_POST['gir_S7'];
$g_S8 = $_POST['gir_S8'];
$g_S9 = $_POST['gir_S9'];
$g_S10 = $_POST['gir_S10'];
$g_S11 = $_POST['gir_S11'];
$g_S12 = $_POST['gir_S12'];
$g_S13 = $_POST['gir_S13'];
$g_S14 = $_POST['gir_S14'];
$g_S15 = $_POST['gir_S15'];
$g_S16 = $_POST['gir_S16'];
$g_S17 = $_POST['gir_S17'];
$g_S18 = $_POST['gir_S18'];
```

```
//the green missed over
$g_O1 = $_POST['gir_O1'];
$g_O2 = $_POST['gir_O2'];
$g_O3 = $_POST['gir_O3'];
$g_O4 = $_POST['gir_O4'];
$g_O5 = $_POST['gir_O5'];
$g_O6 = $_POST['gir_O6'];
$g_O7 = $_POST['gir_O7'];
$g_O8 = $_POST['gir_O8'];
$g_O9 = $_POST['gir_O9'];
$g_O10 = $_POST['gir_O10'];
$g_O11 = $_POST['gir_O11'];
$g_O12 = $_POST['gir_O12'];
$g_O13 = $_POST['gir_O13'];
$g_O14 = $_POST['gir_O14'];
$g_O15 = $_POST['gir_O15'];
$g_O16 = $_POST['gir_O16'];
$g_O17 = $_POST['gir_O17'];
$g_O18 = $_POST['gir_O18'];

//the bunker hit
$b1 = $_POST['bunker1'];
$b2 = $_POST['bunker2'];
$b3 = $_POST['bunker3'];
$b4 = $_POST['bunker4'];
$b5 = $_POST['bunker5'];
$b6 = $_POST['bunker6'];
$b7 = $_POST['bunker7'];
$b8 = $_POST['bunker8'];
$b9 = $_POST['bunker9'];
$b10 = $_POST['bunker10'];
$b11 = $_POST['bunker11'];
$b12 = $_POST['bunker12'];
$b13 = $_POST['bunker13'];
$b14 = $_POST['bunker14'];
$b15 = $_POST['bunker15'];
$b16 = $_POST['bunker16'];
$b17 = $_POST['bunker17'];
$b18 = $_POST['bunker18'];

//the penalties
$pen1 = $_POST['pen1'];
$pen2 = $_POST['pen2'];
$pen3 = $_POST['pen3'];
$pen4 = $_POST['pen4'];
$pen5 = $_POST['pen5'];
$pen6 = $_POST['pen6'];
$pen7 = $_POST['pen7'];
$pen8 = $_POST['pen8'];
$pen9 = $_POST['pen9'];
$pen10 = $_POST['pen10'];
$pen11 = $_POST['pen11'];
$pen12 = $_POST['pen12'];
$pen13 = $_POST['pen13'];
$pen14 = $_POST['pen14'];
$pen15 = $_POST['pen15'];
$pen16 = $_POST['pen16'];
```

```
$pen17 = $_POST['pen17'];
$pen18 = $_POST['pen18'];

//the shot 0-50m missed
$s0_50m1 = $_POST['shot0_50m1'];
$s0_50m2 = $_POST['shot0_50m2'];
$s0_50m3 = $_POST['shot0_50m3'];
$s0_50m4 = $_POST['shot0_50m4'];
$s0_50m5 = $_POST['shot0_50m5'];
$s0_50m6 = $_POST['shot0_50m6'];
$s0_50m7 = $_POST['shot0_50m7'];
$s0_50m8 = $_POST['shot0_50m8'];
$s0_50m9 = $_POST['shot0_50m9'];
$s0_50m10 = $_POST['shot0_50m10'];
$s0_50m11 = $_POST['shot0_50m11'];
$s0_50m12 = $_POST['shot0_50m12'];
$s0_50m13 = $_POST['shot0_50m13'];
$s0_50m14 = $_POST['shot0_50m14'];
$s0_50m15 = $_POST['shot0_50m15'];
$s0_50m16 = $_POST['shot0_50m16'];
$s0_50m17 = $_POST['shot0_50m17'];
$s0_50m18 = $_POST['shot0_50m18'];

//the shot 50-100m missed
$s50_100m1 = $_POST['shot50_100m1'];
$s50_100m2 = $_POST['shot50_100m2'];
$s50_100m3 = $_POST['shot50_100m3'];
$s50_100m4 = $_POST['shot50_100m4'];
$s50_100m5 = $_POST['shot50_100m5'];
$s50_100m6 = $_POST['shot50_100m6'];
$s50_100m7 = $_POST['shot50_100m7'];
$s50_100m8 = $_POST['shot50_100m8'];
$s50_100m9 = $_POST['shot50_100m9'];
$s50_100m10 = $_POST['shot50_100m10'];
$s50_100m11 = $_POST['shot50_100m11'];
$s50_100m12 = $_POST['shot50_100m12'];
$s50_100m13 = $_POST['shot50_100m13'];
$s50_100m14 = $_POST['shot50_100m14'];
$s50_100m15 = $_POST['shot50_100m15'];
$s50_100m16 = $_POST['shot50_100m16'];
$s50_100m17 = $_POST['shot50_100m17'];
$s50_100m18 = $_POST['shot50_100m18'];

//the shot 100-150m missed
$s100_150m1 = $_POST['shot100_150m1'];
$s100_150m2 = $_POST['shot100_150m2'];
$s100_150m3 = $_POST['shot100_150m3'];
$s100_150m4 = $_POST['shot100_150m4'];
$s100_150m5 = $_POST['shot100_150m5'];
$s100_150m6 = $_POST['shot100_150m6'];
$s100_150m7 = $_POST['shot100_150m7'];
$s100_150m8 = $_POST['shot100_150m8'];
$s100_150m9 = $_POST['shot100_150m9'];
$s100_150m10 = $_POST['shot100_150m10'];
$s100_150m11 = $_POST['shot100_150m11'];
$s100_150m12 = $_POST['shot100_150m12'];
$s100_150m13 = $_POST['shot100_150m13'];
```

```
$s100_150m14 = $_POST['shot100_150m14'];
$s100_150m15 = $_POST['shot100_150m15'];
$s100_150m16 = $_POST['shot100_150m16'];
$s100_150m17 = $_POST['shot100_150m17'];
$s100_150m18 = $_POST['shot100_150m18'];

//the putts 0-1m missed
$p0_1m1 = $_POST['putt0_1m1'];
$p0_1m2 = $_POST['putt0_1m2'];
$p0_1m3 = $_POST['putt0_1m3'];
$p0_1m4 = $_POST['putt0_1m4'];
$p0_1m5 = $_POST['putt0_1m5'];
$p0_1m6 = $_POST['putt0_1m6'];
$p0_1m7 = $_POST['putt0_1m7'];
$p0_1m8 = $_POST['putt0_1m8'];
$p0_1m9 = $_POST['putt0_1m9'];
$p0_1m10 = $_POST['putt0_1m10'];
$p0_1m11 = $_POST['putt0_1m11'];
$p0_1m12 = $_POST['putt0_1m12'];
$p0_1m13 = $_POST['putt0_1m13'];
$p0_1m14 = $_POST['putt0_1m14'];
$p0_1m15 = $_POST['putt0_1m15'];
$p0_1m16 = $_POST['putt0_1m16'];
$p0_1m17 = $_POST['putt0_1m17'];
$p0_1m18 = $_POST['putt0_1m18'];

//the putts 1-2m missed
$p1_2m1 = $_POST['putt1_2m1'];
$p1_2m2 = $_POST['putt1_2m2'];
$p1_2m3 = $_POST['putt1_2m3'];
$p1_2m4 = $_POST['putt1_2m4'];
$p1_2m5 = $_POST['putt1_2m5'];
$p1_2m6 = $_POST['putt1_2m6'];
$p1_2m7 = $_POST['putt1_2m7'];
$p1_2m8 = $_POST['putt1_2m8'];
$p1_2m9 = $_POST['putt1_2m9'];
$p1_2m10 = $_POST['putt1_2m10'];
$p1_2m11 = $_POST['putt1_2m11'];
$p1_2m12 = $_POST['putt1_2m12'];
$p1_2m13 = $_POST['putt1_2m13'];
$p1_2m14 = $_POST['putt1_2m14'];
$p1_2m15 = $_POST['putt1_2m15'];
$p1_2m16 = $_POST['putt1_2m16'];
$p1_2m17 = $_POST['putt1_2m17'];
$p1_2m18 = $_POST['putt1_2m18'];

//the putts 2-3m missed
$p2_3m1 = $_POST['putt2_3m1'];
$p2_3m2 = $_POST['putt2_3m2'];
$p2_3m3 = $_POST['putt2_3m3'];
$p2_3m4 = $_POST['putt2_3m4'];
$p2_3m5 = $_POST['putt2_3m5'];
$p2_3m6 = $_POST['putt2_3m6'];
$p2_3m7 = $_POST['putt2_3m7'];
$p2_3m8 = $_POST['putt2_3m8'];
$p2_3m9 = $_POST['putt2_3m9'];
$p2_3m10 = $_POST['putt2_3m10'];
```

```
$p2_3m11 = $_POST['putt2_3m11'];
$p2_3m12 = $_POST['putt2_3m12'];
$p2_3m13 = $_POST['putt2_3m13'];
$p2_3m14 = $_POST['putt2_3m14'];
$p2_3m15 = $_POST['putt2_3m15'];
$p2_3m16 = $_POST['putt2_3m16'];
$p2_3m17 = $_POST['putt2_3m17'];
$p2_3m18 = $_POST['putt2_3m18'];

//18 SQL insert statement into the stats table, one for each hole
$sql1 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h1', '$s1', '$p1', '$d1',
'$d_L1', '$d_R1', '$g1', '$g_L1', '$g_R1', '$g_S1', '$g_O1',
'$s0_50m1', '$s50_100m1', '$s100_150m1', '$b1', '$pen1', '$p0_1m1',
'$p1_2m1', '$p2_3m1', '$tee_colour')";
$sql2 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h2', '$s2', '$p2', '$d2',
'$d_L2', '$d_R2', '$g2', '$g_L2', '$g_R2', '$g_S2', '$g_O2',
'$s0_50m2', '$s50_100m2', '$s100_150m2', '$b2', '$pen2', '$p0_1m2',
'$p1_2m2', '$p2_3m2', '$tee_colour')";
$sql3 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h3', '$s3', '$p3', '$d3',
'$d_L3', '$d_R3', '$g3', '$g_L3', '$g_R3', '$g_S3', '$g_O3',
'$s0_50m3', '$s50_100m3', '$s100_150m3', '$b3', '$pen3', '$p0_1m3',
'$p1_2m3', '$p2_3m3', '$tee_colour')";
$sql4 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m,tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h4', '$s4', '$p4', '$d4',
'$d_L4', '$d_R4', '$g4', '$g_L4', '$g_R4', '$g_S4', '$g_O4',
'$s0_50m4', '$s50_100m4', '$s100_150m4', '$b4', '$pen4', '$p0_1m4',
'$p1_2m4', '$p2_3m4', '$tee_colour')";
$sql5 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h5', '$s5', '$p5', '$d5',
'$d_L5', '$d_R5', '$g5', '$g_L5', '$g_R5', '$g_S5', '$g_O5',
'$s0_50m5', '$s50_100m5', '$s100_150m5', '$b5', '$pen5', '$p0_1m5',
'$p1_2m5', '$p2_3m5', '$tee_colour')";
$sql6 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h6', '$s6', '$p6', '$d6',
'$d_L6', '$d_R6', '$g6', '$g_L6', '$g_R6', '$g_S6', '$g_O6',
'$s0_50m6', '$s50_100m6', '$s100_150m6', '$b6', '$pen6', '$p0_1m6',
'$p1_2m6', '$p2_3m6', '$tee_colour')";
$sql7 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h7', '$s7', '$p7', '$d7',
'$d_L7', '$d_R7', '$g7', '$g_L7', '$g_R7', '$g_S7', '$g_O7',
'$s0_50m7', '$s50_100m7', '$s100_150m7', '$b7', '$pen7', '$p0_1m7',
'$p1_2m7', '$p2_3m7', '$tee_colour')";
$sql8 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h8', '$s8', '$p8', '$d8',
'$d_L8', '$d_R8', '$g8', '$g_L8', '$g_R8', '$g_S8', '$g_O8',
'$s0_50m8', '$s50_100m8', '$s100_150m8', '$b8', '$pen8', '$p0_1m8',
'$p1_2m8', '$p2_3m8', '$tee_colour')";
$sql9 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h9', '$s9', '$p9', '$d9',
'$d_L9', '$d_R9', '$g9', '$g_L9', '$g_R9', '$g_S9', '$g_O9',
'$s0_50m9', '$s50_100m9', '$s100_150m9', '$b9', '$pen9', '$p0_1m9',
'$p1_2m9', '$p2_3m9', '$tee_colour')";
$sql10 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h10', '$s10', '$p10', '$d10',
'$d_L10', '$d_R10', '$g10', '$g_L10', '$g_R10', '$g_S10', '$g_O10',
'$s0_50m10', '$s50_100m10', '$s100_150m10', '$b10', '$pen10',
'$p0_1m10', '$p1_2m10', '$p2_3m10', '$tee_colour')";
$sql11 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h11', '$s11', '$p11', '$d11',
'$d_L11', '$d_R11', '$g11', '$g_L11', '$g_R11', '$g_S11', '$g_O11',
'$s0_50m11', '$s50_100m11', '$s100_150m11', '$b11', '$pen11',
'$p0_1m11', '$p1_2m11', '$p2_3m11', '$tee_colour')";
$sql12 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
```

```
('$courseid', '$roundid', '$playerid', '$h12', '$s12', '$p12', '$d12',
'$d_L12', '$d_R12', '$g12', '$g_L12', '$g_R12', '$g_S12', '$g_O12',
'$s0_50m12', '$s50_100m12', '$s100_150m12', '$b12', '$pen12',
'$p0_1m12', '$p1_2m12', '$p2_3m12', '$tee_colour')";
$sql13 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h13', '$s13', '$p13', '$d13',
'$d_L13', '$d_R13', '$g13', '$g_L13', '$g_R13', '$g_S13', '$g_O13',
'$s0_50m13', '$s50_100m13', '$s100_150m13', '$b13', '$pen13',
'$p0_1m13', '$p1_2m13', '$p2_3m13', '$tee_colour')";
$sql14 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h14', '$s14', '$p14', '$d14',
'$d_L14', '$d_R14', '$g14', '$g_L14', '$g_R14', '$g_S14', '$g_O14',
'$s0_50m14', '$s50_100m14', '$s100_150m14', '$b14', '$pen14',
'$p0_1m14', '$p1_2m14', '$p2_3m14', '$tee_colour')";
$sql15 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h15', '$s15', '$p15', '$d15',
'$d_L15', '$d_R15', '$g15', '$g_L15', '$g_R15', '$g_S15', '$g_O15',
'$s0_50m15', '$s50_100m15', '$s100_150m15', '$b15', '$pen15',
'$p0_1m15', '$p1_2m15', '$p2_3m15', '$tee_colour')";
$sql16 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h16', '$s16', '$p16', '$d16',
'$d_L16', '$d_R16', '$g16', '$g_L16', '$g_R16', '$g_S16', '$g_O16',
'$s0_50m16', '$s50_100m16', '$s100_150m16', '$b16', '$pen16',
'$p0_1m16', '$p1_2m16', '$p2_3m16', '$tee_colour')";
$sql17 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h17', '$s17', '$p17', '$d17',
'$d_L17', '$d_R17', '$g17', '$g_L17', '$g_R17', '$g_S17', '$g_O17',
'$s0_50m17', '$s50_100m17', '$s100_150m17', '$b17', '$pen17',
'$p0_1m17', '$p1_2m17', '$p2_3m17', '$tee_colour')";
$sql18 = "INSERT INTO stats (courseid, golfroundid, playerid, hole_nr,
score, putts, drive_hit, drive_missed_L, drive_missed_R, green_hit,
green_missed_L, green_missed_R, green_missed_S, green_missed_O,
shot0_50m_missed, shot50_100m_missed, shot100_150m_missed, bunker,
penalty, putts0_1m, putts1_2m, putts2_3m, tee_colour) VALUES
('$courseid', '$roundid', '$playerid', '$h18', '$s18', '$p18', '$d18',
'$d_L18', '$d_R18', '$g18', '$g_L18', '$g_R18', '$g_S18', '$g_O18',
'$s0_50m18', '$s50_100m18', '$s100_150m18', '$b18', '$pen18',
'$p0_1m18', '$p1_2m18', '$p2_3m18', '$tee_colour')";

//Queries the database to check whether these statistics for this round
have been inputted before
$login_sql_check = mysql_query("SELECT * FROM stats WHERE
courseid='$courseid' and playerid='$playerid' and
golfroundid='$roundid'");
      $login_check = mysql_num_rows($login_sql_check);

//If the statistics haven't been inputted before then input them now
if($login_check == 0)
{
      //18 if statement each one checking if score has some data
      //if it has then input the hole
      if($s1 != "")
      {
          $query1 = $sql1;

              mysql_query($query1) or
                  die (mysql_error());
      }

      if ($s2 != "")
      {
          $query2 = $sql2;

              mysql_query($query2) or
                  die (mysql_error());
      }

      if ($s3 != "")
      {
          $query3 = $sql3;

              mysql_query($query3) or
                  die (mysql_error());
      }

      if ($s4 != "")
      {
          $query4 = $sql4;

              mysql_query($query4) or
                  die (mysql_error());
      }

      if ($s5 != "")
      {
          $query5 = $sql5;

              mysql_query($query5) or
                  die (mysql_error());
      }

      if ($s6 != "")
      {
          $query6 = $sql6;

              mysql_query($query6) or
                  die (mysql_error());
      }

      if ($s7 != "")
      {
          $query7 = $sql7;

              mysql_query($query7) or
                  die (mysql_error());
      }

      if ($s8 != "")
      {
          $query8 = $sql8;

              mysql_query($query8) or
                  die (mysql_error());
      }

      if ($s9 != "")
      {
          $query9 = $sql9;

              mysql_query($query9) or
                  die (mysql_error());
      }

      if($s10 != "")
      {
          $query10 = $sql10;

              mysql_query($query10) or
                  die (mysql_error());
      }

      if ($s11 != "")
      {
          $query11 = $sql11;

              mysql_query($query11) or
                  die (mysql_error());
      }

      if ($s12 != "")
      {
          $query12 = $sql12;

              mysql_query($query12) or
                  die (mysql_error());
      }

      if ($s13 != "")
      {
          $query13 = $sql13;
```

```
              mysql_query($query13) or
                  die (mysql_error());
      }

      if ($s14 != "")
      {
          $query14 = $sql14;

              mysql_query($query14) or
                  die (mysql_error());
      }

      if ($s15 != "")
      {
          $query15 = $sql15;

              mysql_query($query15) or
                  die (mysql_error());
      }

      if ($s16 != "")
      {
          $query16 = $sql16;

              mysql_query($query16) or
                  die (mysql_error());
      }

      if ($s17 != "")
      {
          $query17 = $sql17;

              mysql_query($query17) or
                  die (mysql_error());
      }

      if ($s18 != "")
      {
          $query18 = $sql18;

              mysql_query($query18) or
                  die (mysql_error());
      }

                  echo "<p>The statistics have been stored in the
database</p>";
                  include("main.php");
}
//else stats for this round already been inputted
else
{
    echo "<p>Stats for this round have already been inputted</p>";
    include("golf.php");
}
?>
```

Holes.php

```
<html>
<head>
            <title>Golfholes input</title>
          <link rel="stylesheet" href="new.css" />

<script language="Javascript">
    function calclength()
    {
      var vallength, totallength;
      totallength=0;

      for(i=1;i<19;i++)
      {
          vallength=document.forms[0].elements["length"+i].value;
          if(parseInt(vallength))
          {
            totallength=totallength+parseInt(vallength);
          }
      }

      document.forms[0].elements["totallength"].value=totallength;
    }
    function calcpar()
    {
      var valpar, totalpar;
      totalpar=0;

      for(i=1;i<19;i++)
      {
          valpar=document.forms[0].elements["par"+i].value;
          if(parseInt(valpar))
          {
            totalpar=totalpar+parseInt(valpar);
          }
      }

      document.forms[0].elements["totalpar"].value=totalpar;
    }

</script>

</head>

<body>
<h2>Input your golf holes on your golf course</h2>

<?
include("db_connect.php");

$sql="SELECT courseid, name FROM course";
$result=mysql_query($sql);

$options="";

while ($row = mysql_fetch_array($result)) {
    $id=$row["courseid"];
    $thing=$row["name"];
    $options.="<OPTION VALUE=\"$id\">$thing</option>";
}
?>

    <form method="post" action="holes_entry.php">

    <table>
      <tr>
          <td><span>Course:</span></td>
          <td><select name="courseid">
            <option value="0">
            <?=$options?>
            </select>
          </td>
      </tr>
      <tr>
          <td><span>Tees:</span></td>
          <td><select name="colour">
            <option value="white">White</option>
            <option value="yellow">Yellow</option>
            <option value="blue">Blue</option>
            <option value="red">Red</option>
            </select>
          </td>
      </tr>
    </table>
    <table>
      <tr>
                <td>Hole</td>
                <td>1</td>
                <td>2</td>
                <td>3</td>
                <td>4</td>
                <td>5</td>
                <td>6</td>
                <td>7</td>
                <td>8</td>
                <td>9</td>
                <td>10</td>
                <td>11</td>
                <td>12</td>
                <td>13</td>
                <td>14</td>
                <td>15</td>
                <td>16</td>
                <td>17</td>
                <td>18</td>
            <td>Total</td>
            </tr>
            <tr>
                <td>length</td>
                <td><input type="text" size="2" name="length1" value=""
onchange="calclength()"/></td>
```

```
                <td><input type="text" size="2" name="length2" value=""
onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length3" value=""
onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length4" value=""
onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length5" value=""
onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length6" value=""
onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length7" value=""
onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length8" value=""
onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length9" value=""
onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length10"
value="" onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length11"
value="" onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length12"
value="" onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length13"
value="" onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length14"
value="" onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length15"
value="" onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length16"
value="" onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length17"
value="" onchange="calclength()"/></td>
                <td><input type="text" size="2" name="length18"
value="" onchange="calclength()"/></td>
                <td><input readonly="text" size="2"
name="totallength"/></td>
            </tr>
            <tr>
                <td>Par</td>
                <td><input type="text" size="2" name="par1" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par2" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par3" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par4" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par5" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par6" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par7" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par8" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par9" value=""
onchange="calcpar()"/></td>
```

```
                <td><input type="text" size="2" name="par10" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par11" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par12" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par13" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par14" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par15" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par16" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par17" value=""
onchange="calcpar()"/></td>
                <td><input type="text" size="2" name="par18" value=""
onchange="calcpar()"/></td>
                <td><input readonly="text" size="2"
name="totalpar"/></td>
            </tr>
            <tr>
                <td>Handicap</td>
                <td><input type="text" size="2" name="hcp1"/></td>
                <td><input type="text" size="2" name="hcp2"/></td>
                <td><input type="text" size="2" name="hcp3"/></td>
                <td><input type="text" size="2" name="hcp4"/></td>
                <td><input type="text" size="2" name="hcp5"/></td>
                <td><input type="text" size="2" name="hcp6"/></td>
                <td><input type="text" size="2" name="hcp7"/></td>
                <td><input type="text" size="2" name="hcp8"/></td>
                <td><input type="text" size="2" name="hcp9"/></td>
                <td><input type="text" size="2" name="hcp10"/></td>
                <td><input type="text" size="2" name="hcp11"/></td>
                <td><input type="text" size="2" name="hcp12"/></td>
                <td><input type="text" size="2" name="hcp13"/></td>
                <td><input type="text" size="2" name="hcp14"/></td>
                <td><input type="text" size="2" name="hcp15"/></td>
                <td><input type="text" size="2" name="hcp16"/></td>
                <td><input type="text" size="2" name="hcp17"/></td>
                <td><input type="text" size="2" name="hcp18"/></td>
            </tr>
        </table>

    <input type="submit" name="submit" value="Input"/>
    <input type="reset" name="reset" value="Clear"/>

    </form>

</body>
</html>
```

Holes_entry.php

```
<?php
//Connects to the database
include("db_connect.php");
```

```
//Takes in the posted data from course.php and stores n variables
$colour = $_POST['colour'];
$courseid = $_POST['courseid'];

//the length of the holes
$l1 = $_POST['length1'];
$l2 = $_POST['length2'];
$l3 = $_POST['length3'];
$l4 = $_POST['length4'];
$l5 = $_POST['length5'];
$l6 = $_POST['length6'];
$l7 = $_POST['length7'];
$l8 = $_POST['length8'];
$l9 = $_POST['length9'];
$l10 = $_POST['length10'];
$l11 = $_POST['length11'];
$l12 = $_POST['length12'];
$l13 = $_POST['length13'];
$l14 = $_POST['length14'];
$l15 = $_POST['length15'];
$l16 = $_POST['length16'];
$l17 = $_POST['length17'];
$l18 = $_POST['length18'];

//the holes number
$h1 = 1;
$h2 = 2;
$h3 = 3;
$h4 = 4;
$h5 = 5;
$h6 = 6;
$h7 = 7;
$h8 = 8;
$h9 = 9;
$h10 = 10;
$h11 = 11;
$h12 = 12;
$h13 = 13;
$h14 = 14;
$h15 = 15;
$h16 = 16;
$h17 = 17;
$h18 = 18;

//the par of the holes
$p1 = $_POST['par1'];
$p2 = $_POST['par2'];
$p3 = $_POST['par3'];
$p4 = $_POST['par4'];
$p5 = $_POST['par5'];
$p6 = $_POST['par6'];
$p7 = $_POST['par7'];
$p8 = $_POST['par8'];
$p9 = $_POST['par9'];
$p10 = $_POST['par10'];
$p11 = $_POST['par11'];
```

```
$p12 = $_POST['par12'];
$p13 = $_POST['par13'];
$p14 = $_POST['par14'];
$p15 = $_POST['par15'];
$p16 = $_POST['par16'];
$p17 = $_POST['par17'];
$p18 = $_POST['par18'];

//the handicap of the holes
$hand1 = $_POST['hcp1'];
$hand2 = $_POST['hcp2'];
$hand3 = $_POST['hcp3'];
$hand4 = $_POST['hcp4'];
$hand5 = $_POST['hcp5'];
$hand6 = $_POST['hcp6'];
$hand7 = $_POST['hcp7'];
$hand8 = $_POST['hcp8'];
$hand9 = $_POST['hcp9'];
$hand10 = $_POST['hcp10'];
$hand11 = $_POST['hcp11'];
$hand12 = $_POST['hcp12'];
$hand13 = $_POST['hcp13'];
$hand14 = $_POST['hcp14'];
$hand15 = $_POST['hcp15'];
$hand16 = $_POST['hcp16'];
$hand17 = $_POST['hcp17'];
$hand18 = $_POST['hcp18'];

//18 SQL insert statements into the holes table, one for each hole
$sql1 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h1', '$l1', '$p1', '$hand1')";
$sql2 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h2', '$l2', '$p2', '$hand2')";
$sql3 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h3', '$l3', '$p3', '$hand3')";
$sql4 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h4', '$l4', '$p4', '$hand4')";
$sql5 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h5', '$l5', '$p5', '$hand5')";
$sql6 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h6', '$l6', '$p6', '$hand6')";
$sql7 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h7', '$l7', '$p7', '$hand7')";
$sql8 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h8', '$l8', '$p8', '$hand8')";
$sql9 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h9', '$l9', '$p9', '$hand9')";
$sql10 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h10', '$l10', '$p10',
'$hand10')";
$sql11 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h11', '$l11', '$p11',
'$hand11')";
$sql12 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h12', '$l12', '$p12',
'$hand12')";
```

```
$sql13 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h13', '$l13', '$p13',
'$hand13')";
$sql14 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h14', '$l14', '$p14',
'$hand14')";
$sql15 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h15', '$l15', '$p15',
'$hand15')";
$sql16 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h16', '$l16', '$p16',
'$hand16')";
$sql17 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h17', '$l17', '$p17',
'$hand17')";
$sql18 = "INSERT INTO holes (courseid, colour, hole_nr, length, par,
hcp) VALUES ('$courseid', '$colour', '$h18', '$l18', '$p18',
'$hand18')";

//Queries the database to check whether this course and this tee colour
is in the holes table
$login_sql_check = mysql_query("SELECT * FROM holes WHERE
courseid='$courseid' and colour='$colour'");
      $login_check = mysql_num_rows($login_sql_check);

//If it isn't then go ahead
if($login_check == 0)
{
      //checks whether information were put into the course and colour
box in holes.php,
      //if so and insert these pair of holes.
      if ($courseid != "" && $colour != "")
      {
            //18 checks where the par has not been inputted
            if($p1 != "")
            {
                  $query1 = $sql1;

                  mysql_query($query1) or
                        die (mysql_error());
            }

            if ($p2 != "")
            {
                $query2 = $sql2;

                  mysql_query($query2) or
                  die (mysql_error());
            }

            if ($p3 != "")
            {
                $query3 = $sql3;

                mysql_query($query3) or
                  die (mysql_error());
            }
```

```
if ($p4 != "")
{
    $query4 = $sql4;

      mysql_query($query4) or
          die (mysql_error());
}

if ($p5 != "")
{
    $query5 = $sql5;

    mysql_query($query5) or
      die (mysql_error());
}

if ($p6 != "")
{
    $query6 = $sql6;

    mysql_query($query6) or
      die (mysql_error());
}

if ($p7 != "")
{
      $query7 = $sql7;

      mysql_query($query7) or
         die (mysql_error());
}

if ($p8 != "")
{
      $query8 = $sql8;

      mysql_query($query8) or
          die (mysql_error());
}

if ($p9 != "")
{
      $query9 = $sql9;

      mysql_query($query9) or
          die (mysql_error());
}

if($p10 != "")
{
      $query10 = $sql10;

      mysql_query($query10) or
         die (mysql_error());
}
```

```
if ($p11 != "")
{
      $query11 = $sql11;

      mysql_query($query11) or
          die (mysql_error());
}

if ($p12 != "")
{
      $query12 = $sql12;

      mysql_query($query12) or
                die (mysql_error());
}

if ($p13 != "")
{
      $query13 = $sql13;

      mysql_query($query13) or
          die (mysql_error());
}

if ($p14 != "")
{
      $query14 = $sql14;

      mysql_query($query14) or
          die (mysql_error());
}

if ($p15 != "")
{
      $query15 = $sql15;

      mysql_query($query15) or
          die (mysql_error());
}

if ($p16 != "")
{
      $query16 = $sql16;

      mysql_query($query16) or
          die (mysql_error());
}

if ($p17 != "")
{
      $query17 = $sql17;

      mysql_query($query17) or
          die (mysql_error());
}

if ($p18 != "")
```

```
            {
                  $query18 = $sql18;

                    mysql_query($query18) or
                        die (mysql_error());
            }

            echo "<p>Teigurinn er kominn í grunninn</p>";
            include("round.php");
            include("golf.php");
      }
      else
      {
            echo "<p>Eitthvað ekki að virka</p>";
            include("holes.php");
      }
}
else
{
      echo "<p>These holes already exists</p>";
      include("holes.php");
}
?>
```

Index.php

```
<html>

<head>
<title>The Golfhelper</title>
</head>

<frameset rows="120,*" framespacing="0" border="0" frameborder="0">
      <frame name="banner" scrolling="no" noresize target="contents"
src="border.php">
      <frameset cols="142">
                  <frame name="main" src="index_main.php"
scrolling="auto" target="_self">
      </frameset>
</frameset>


</html>
```

Index_entry.php

```
<?php
//Connects to the database
include("db_connect.php");

//Takes in the posted data from course.php and stores n variables
$login = $_POST['login'];
$password = $_POST['password'];

//If either login was left empty or the password then insert
information and try again
if(($login != "") || ($password != "")){
echo "Please enter ALL of the information! <br />";
include ("index.php");
exit();
}

//Convert password to md5 encryption string
$password = md5($password);

//Queries the database to check whether the login name and password fit
together
$sql = mysql_query("SELECT * FROM player WHERE login='$login' AND
password='$password'");
$login_check = mysql_num_rows($sql);

//If they do, the golfer in system
if($login_check > 0)
{
    include("main.php");
}
//else either password or login wrong, try again
else
{
    echo "You could not be logged in! Either the username and password
do not match!<br />
    Please try again!<br />";
    include ("index_main.php");
}
?>
```

Index_main.php

```
<html>
<head>
      <title>Golfhelper</title>
      <link rel="stylesheet" href="new.css" />
</head>

<body>
<form method="post" action="index_entry.php">
     <table>
    <tr>
      <td><span>Login name:</span></td>
      <td><input type="text" size="10" name="login"/></td>
    </tr>
    <tr>
      <td><span>Password:</span></td>
      <td><input type="password" size="10" name="password"/></td>
    </table><br />

    <input type="submit" name="submit" value="Login"/>
    <input type="reset" name="reset" value="Reset"/>

    </form>
    <div><a href="player.php">New Player</a></div>

</body>
</html>
```

Main.php

```
<?php
echo "<a href=\"golf.php\">Input Statistics</a>";
include("round.php");
include("course.php");
include("tees.php");
include("avg_stats.php");
?>
```

Player.php

```
<html>
<head>
      <title>Players</title>
      <link rel="stylesheet" href="new.css" />
</head>

<body>
<span class="head">Registration of a new player</span>
<div>Please fill out all the columns. Login name should contain 3-10
</div><div>characters and the password should contain 6-10
characters.</div><br />


<form method="post" action="player_entry.php">
    <table>
    <tr>
      <td><span>First Name:</span></td>
      <td><input type="text" size="30" name="firstname"/></td>
    </tr>
    <tr>
      <td><span>Surname:</span></td>
      <td><input type="text" size="30" name="surname"/></td>
    </tr>
    <tr>
      <td><span>Handicap:</span></td>
      <td><input type="text" size="2" name="hcp"/></td>
    </tr>
    <tr>
      <td><span>Age:</span></td>
      <td><input type="text" size="3" name="age"/></td>
    </tr>
    <tr>
      <td><span>Login name:</span></td>
      <td><input type="text" size="10" maxlength="10"
name="login"/></td>
    </tr>
    <tr>
      <td><span>Password:</span></td>
      <td><input type="password" size="10" maxlength="10"
name="password"/></td>
    </tr>
    <tr>
      <td><span>Password again:</span></td>
      <td><input type="password" size="10" maxlength="10"
name="password_again"/></td>
    </tr>
    </table><br />

    <input type="submit" name="submit" value="Input"/>
    <input type="reset" name="reset" value="Reset"/>

    </form>

</body>
</html>
```

Player_entry.php

```
<?php
//Connects to the database
include("db_connect.php");

//Takes in the posted data from player.php and stores in variables
$firstname = $_POST['firstname'];
$surname = $_POST['surname'];
$hcp = $_POST['hcp'];
$age = $_POST['age'];
$login = $_POST['login'];
$password = $_POST['password'];
$password_again = $_POST['password_again'];

//Check if the login name is of rigth size
if((strlen ($login) > 2) && (strlen ($login) < 11))
{
      //Queries the database to check whether this login name exits
      $login_sql_check = mysql_query("SELECT * FROM player WHERE
login='$login'");
      $login_check = mysql_num_rows($login_sql_check);

      //If it does then please pick another login name
      if($login_check > 0)
      {
            echo "<p>Login name already exists</p>";
            echo "<p>Please try another one</p>";
            include("player.php");
      }
      //If it does not, continue
      else
      {
            //checks if the password is at right length and matches
with the retyped password
            if($password != $password_again && (strlen ($password) < 6)
&& (strlen ($password) > 10))
            {
                  echo "<p>Password incorrectly inputted</p>";
            }
            else
            {
                  //SQL insert statement to input player into player
table
                  $sql = "INSERT INTO player (firstname, surname, hcp,
age, login, password) VALUES ('$firstname', '$surname', '$hcp', '$age',
'$login', md5('$password'))";
```

```
                  //if firstname, surname and hcp have been entered by
player then input player
                  if ($firstname != "" && $surname != "" && $hcp != "")
                  {
                        $query = $sql;

                        mysql_query($query) or
                        die (mysql_error());

                        echo "<div>The player has been
registered</div>";
                        echo "<div>Please login</div>";
                        include("index_main.php");
                  }
                  //else please insert all information
                  else
                  {
                        echo "<p>Every box need to be inputted</p>";
                        include("player.php");
                  }
            }
      }
}
//else the length of the login name is incorrect
else
{
    echo "<p>Login name length incorrect</p>";
    include("player.php");
}
?>
```

Recommendation.php

```
<html>
<head>
            <title>Analysis my statistic</title>
          <link rel="stylesheet" href="new.css" />
</head>
<body>
<h2>Pick a player to analyze this statistics</h2>

<?
include("db_connect.php");

$sql1="SELECT playerid, firstname FROM player";
$result1=mysql_query($sql1);

$options1="";
while ($row1 = mysql_fetch_array($result1)) {

    $id1=$row1["playerid"];
    $thing1=$row1["firstname"];
    $options1.="<OPTION VALUE=\"$id1\">$thing1</option>";
}
?>
    <form method="post" action="recommendation_output.php">
    <span>Player:</span>
    <select name="playerid">
    <option value="0">
    <?=$options1?>
    </SELECT>

    <input type="submit" name="submit" value="Analyse this"/>
    </form>

</body>
</html>
```

Recommendation_admin.php

```
<html>
<head>
      <title>Inputting recommendations</title>
      <link rel="stylesheet" href="new.css" />
</head>

<body>

    <form method="post" action="recommendation_admin_entry.php">

    <table>
    <tr>
      <td><span>Type of recommendation:</span></td>
      <td><select name="type"><option selected="selected"
value="drive">drive</option><option value="putts">putts</option><option
value="pitch">pitch</option><option
value="shots">shots</option></SELECT></td>
    </tr>
    <tr>
      <td><span>Recommendation Id:</span></td>
      <td><input type="text" size="2" name="recid"/></td>
    </tr>
    <tr>
      <td><span>Type of recommendation:</span></td>
      <td><select name="level"><option selected="selected"
value="1">1</option><option value="2">2</option><option
value="3">3</option></SELECT></td>
    </tr>
    <tr>
      <td><span>The recommendation:</span></td>
      <td><textarea rows="15" cols="40"
name="recommendation"></textarea></td>
    </tr>
    </table>
    <input type="submit" name="submit" value="Input"/>
    <input type="reset" name="reset" value="Reset"/>

    </form>

</body>
</html>
```

Recommendation_admin_entry.php

```
<?php
//Connects to the database
include("db_connect.php");

//Takes in the posted data from the recommendation_admin.php and stores
in variables
$type = $_POST['type'];
$recommendation = $_POST['recommendation'];
$recid = $_POST['recid'];
$level = $_POST['level'];

//SQL insert statement to input the recommendation into the
recommendation table
$sql = "INSERT INTO recommendation (recid, type, recommendation, level)
VALUES ($recid, '$type', '$recommendation', '$level')";

//routine check whether this recommendation has been added before
$login_sql_check = mysql_query("SELECT * FROM recommendation WHERE
recid='$recid' and level='$level' and type='$type'");
      $login_check = mysql_num_rows($login_sql_check);

//If it has not been added before continue
if($login_check == 0)
{
      //check whether all information needed have been inputted
      if ($type != "" || $recommendation != "" || $level != "")
      {
            $query = $sql;

            mysql_query($query) or
                        die (mysql_error());

            echo "<p>Recommendation inserted into the database</p>";
            include("recommendation_admin.php");
      }
      //else input them
      else
      {
            echo "<p>All information have to inputted</p>";
            include("recommendation_admin.php");
      }
}
//else recommendation has been added before
else
{
      echo "<p>This recommendation with the recommendation id $recid
and level $level is already in the database</p>";
      include("recommendation_admin.php");
}
?>
```

Recommendation_output.php

```
<?php
//Connects to the database
include("db_connect.php")
;
```

```
echo "<link rel=\"Stylesheet\" type=\"text/css\" href=\"new.css\" />
";
//Takes in the posted data from recommendation.php and stores in
variables
$playerid = $_POST['playerid'];

//Queries the database to check whether this player has any information
in the avg_stats table
$inputted_rounds_query = mysql_query("SELECT * FROM avg_stats WHERE
playerid='$playerid'");
$inputted_rounds_check = mysql_num_rows($inputted_rounds_query);

//if the player has the execute the Main.java which is the Statistical
Analyzer
if($inputted_rounds_check > 0)
{
      //This is made for a Linux server. So this execution will work on
Windows
      //the : has to be changed to ; and / has to be changed to \ in
the cp variable
      //This is the classpath to jess.jar and mysql-connector...jar
which are needed when
      //executing the Statistical Analyzer
      $cp = "./:./jess.jar";
      $cp .= ":./mysql-connector-java-3.0.11-stable-bin.jar";

      //the execution string to the Statistical Analyzer
      $cmd = "java -cp $cp Main $playerid";

      //executs the Statistical Analyzer and puts the output into a
variable
      $lastline = exec($cmd,$outputarray);

      //Queries the player table to get the first name of the player
involved
      $name_query = mysql_query("select firstname from player where
playerid=$playerid") or die (mysql_error());
      $name = mysql_result($name_query, 0);

      //Displays this string
      echo "<h2>Here are your recommendation $name \n</h2>";

      //table starts
      echo "<table>";

      //foreach recommendation stored in outputarray display
      foreach($outputarray as $value)
      {
            echo "<tr>";
            echo "<td><span>The recommendation is:</span><td>";
            echo "<td>$value<td>";
            echo "</tr>";
      }
      //table ends
      echo "</table>";
      echo "<a href=\"javascript:history.back(0)\">Back</a>";
}
```

```
//else the player has not let the system calculate his average
statistics
else
{
      echo "<span>This player has not let this system calculate his
average statistics.\n</span>";
      echo "<span>Please pick calculate the average statistics or input
some statistics.\n</span>";
      include("avg_stats.php");
}
?>
```

Round.php

```
<html>
<head>
        <title>Rounds</title>
        <link rel="stylesheet" href="new.css" />
</head>

<body>
<h2>Insert a new round</h2>

    <form method="post" action="round_entry.php">
    <table>
    <tr>
      <td><span>Tournament:</span></td>
      <td><input type="text" size="30" name="tournament"/></td>
    </tr>
    </table>
    <input type="submit" name="submit" value="Input"/>
    <input type="reset" name="reset" value="Clear"/>

    </form>

</body>
</html>
```

Round_entry.php

```
<?php
include("db_connect.php");
$tournament = $_POST['tournament'];
$time = time();

$sql = "INSERT INTO golfrounds (tournament, date) VALUES
('$tournament', $time)";

if ($tournament != "")
{
    $query = $sql;

        mysql_query($query) or
            die (mysql_error());

echo "<p>Hringurinn er komin í grunninn</p>";
include("golf.php");
}
```

```
else
{
        echo "<p>Allir reitir verða að útfylltir</p>";
    include("round.php");
}
?>
```

Tees.php

```
<html>
<head>
      <title>Golftees input</title>
      <link rel="stylesheet" href="new.css" />
</head>

<body>
<h2>Insert a new tees on your golf course</h2>

<?
//Connection to the database
include("db_connect.php");

$sql="SELECT courseid, name FROM course";
$result=mysql_query($sql);

$options="";

//getting the name and courseid of the courses in the database
while ($row = mysql_fetch_array($result)) {

    $id=$row["courseid"];
    $thing=$row["name"];
    $options.="<OPTION VALUE=\"$id\">".$thing;
}
?>

    <form method="post" action="tee_entry.php">

    <table>
    <tr>
      <td><span>Tee Colour:</span></td>
      <td><select name="colour">
          <option value="white">White</option>
          <option value="yellow">Yellow</option>
          <option value="blue">Blue</option>
          <option value="red">Red</option>
          </select>
      </td>
    </tr>
    <tr>
      <td><span>Course:</span></td>
      <td><select name="courseid"><option
value="0"><?=$options?></select><td>
    </tr>
    <tr>
      <td><span>Course Rate:</span></td>
      <td><input type="text" size="4" name="cr"/></td>
    </tr>
    <tr>
      <td><span>Slope:</span></td>
      <td><input type="text" size="3" name="slope"/></td>
    </tr>
    </table>

    <input type="submit" name="submit" value="Input"/>
    <input type="reset" name="reset" value="Clear"/>

    </form>

</body>
</html>
```

Tee_entry.php

```
<?php
//Connects to the database
include("db_connect.php");

//Takes in the posted data from recommendation.php and stores in
variables
$colour = $_POST['colour'];
$courseid = $_POST['courseid'];
$cr = $_POST['cr'];
$slope = $_POST['slope'];

//SQL insert to input the tee into the tee table
$sql = "INSERT INTO tees (colour, courseid, cr, slope) VALUES
('$colour', '$courseid', '$cr', '$slope')";

//Queries the database to check whether this tee has any information in
the tees table
$login_sql_check = mysql_query("SELECT * FROM tees WHERE
colour='$colour' and courseid='$courseid'");
      $login_check = mysql_num_rows($login_sql_check);

//if it has no information then continue
if($login_check == 0)
{
      //checking if all neccessary information have been inputted
      //if so input the tee into the tee table
      if ($colour != "" && $cr != "" && $slope != "" && $courseid !=
"")
      {
            $query = $sql;

            mysql_query($query) or
                        die (mysql_error());

            echo "<p>The tee has been added to the database</p>";
            include("holes.php");
      }
      //Please input the neccessary information
      else
      {
            echo "<p>All boxes have to be filled or chosen</p>";
            include("tees.php");
      }
}
//else the tee is already in the database
else
{
      echo "<p>This tee is already in the database</p>";
      include("main.php");
}
?>
```

Appendix B

Here I am going to show you two kind of statistical recording forms. I will explain what each of the fields mean and what is being recorded. The minimum statistics is as follows:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Total
Score	4	5	6	3	5	4	5	4	4	4	3	4	4	3	4	4	5	4	75
GIR	1	1		1	1			1		1	1		1	1	1				10
Drive	1		1		1			1					1		1		1		7
Putts	2	2	2	2	3	2	2	2	1	2	2	1	2	2	1	1	1	2	32

Score stands for itself, just the score on each hole.
GIR stands for Greens in Regulation, that means that the golfer has hit the green in as many shots he is should. Meaning e.g. at a hole which is par 4 the golfer should make the green in 2 shots and then have 2 putts on the green to get par. The par of the hole usually depends on her length. On par 5 the golfer should reach the green in 3 shots and on par 3 the golfer should reach the green in 1 shot. Drive stands for drive on fairway or not. This is either done with 1 or a tick and the total are counted. Putts stands for putts on the each and the total are counted.
With just this the golfer can just see if he is hitting enough GIR's and fairways and how many putts he made. (The average putts should not exceed 30 putts per round. Good putters are making around 25 putts a round). This is not enough to give him any good tips on how to improve his game. There are very few patterns that can be found from this kind of statistics.
This is the statistics form I was using this summer, it's pretty good though it is far from being complex. I would say that it is somewhere in the middle. It looks like this:

Hole	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Total
Par	4	4	3	5	3	4	4	4	4	4	4	3	5	3	4	4	4	4	70
Score	5	5	5	6	3	5	5	4	6	5	4	4	6	6	5	4	6	5	89
Drive fairway hit		1	1		1			1				1	1	1		1	1		9
Drive fairway missed L/R	L			L		L	R		R	R	L				R			R	
GIR			1		1			1		1	1	1				1			7

Green miss S/O L/R	S	S		R		S	S		S				S	R	S		S	S	
Putts	2	2	4	2	2	2	2	2	1	3	2	3	2	2	2	2	2	2	39
Bunker																			0
Green 0-50m														2					2
Green 50-100m	1			1			1		1				1					1	6
Green 100-150m						1									1		1		3
Water																	1		1
OB																			
Lost ball									1										1
Putts 0-1m			1							1									2
Putts 1-2m																			0
Putts 2-3m			1	1								1							3

Now there are a lot of changes from the first example to this one. I just start at the top and work my way down.

Score	The score on each hole.
Drive fairway hit	Drive fairway hit stands for hit fairways on the round.
Drive fairway missed L/R	Drive fairway missed stands for missed fairway. Here has been added in the letters L and R which stand for left and right respectively. This places the ball left or right of the fairway after the drive on each hole.
GIR	Green in regulation. (Explained above)
Green miss S/O L/R S/O L/R	Green missed stands for missed GIR's. The letters stand for S = short to the green, O = over the green, L = left of the green, R = right of the green. This places the ball of the golfer on the shot in his GIR shot.
Putts	Total putts on each hole and then the total for the whole round.
Bunker	Bunker stands for the times the golfer went into a bunker on the round.
Green 0-50m	Green 0-50m stands for shots that miss the green from 0-50 meters distance
Green 50-100m	Green 50-100m stands for shots that miss the green from 50-100 meters distance from the green.
Green 100-150m	Green 100-150m stands for shots that miss the green from 100-150 meters distance from the green.
Water	Water is marked if the golfer has put his ball into water on the round.
OB	OB stands for Out of Bounds or Out of the path marked for the course. This is marked when the golfer plays the ball outside the marked path of the course.
Lost ball	Lost ball is like the name suggests, is marked when the player looses his ball. This is not marked when the golfer hits the water or OB. This is marked e.g. when the player plays the ball into rough, trees/bushes or over a hill and does find the ball.
Putts 0-1m	Putts from 0-1m is marked when a golfer misses a putt from 1 meters or less from the hole.
Putts 1-2m	Putts from 1-2m is marked when a golfer misses a putt from between 1 and

	2 meters from the hole.
Putts 2-3m	Putts from 2-3m is marked when a golfer misses a putt from between 2 and 3 meters from the hole

The shots from 0-100m away from the green are the shots that different between a good golfer and a not as good golfer. If a golfer can always get these shots on the green, his handicap will go down and he will improve.
Water, OB and lost ball all stand for penalties on the round. The different between hitting the water from the other two is that you can drop beside or behind the water and get one shoot penalty but if you go OB or loose your ball you have to play the ball from the same spot as you where when you played the ball before. OB and lost ball often cost the golfer two shots because, as stated above, he haves to play the same shot again plus a one shoot penalty.
Like the shots from 0-100m are vital to improve the game the 0-3m putts are the same. A golfer how can always put down his putts from that distance is in most cases a better golfer then how is struggling with these lengths.
From this kind of statistics there are a lot of patterns that can be found. E.g. a pattern of the golfer drives can be found.

Appendix C

The journal for my final year project. It can be found at:
http://dyna35.skrin.is/bjarni/journal

First day of Jess	09.01.2004	01:50:47
Today I started to read an introduction to the JESS language. JESS stands for Java Expert System Shell. I found out that it is similiar to LISP in many ways like it has lists and atoms. The basic JESS are 5 things. Atoms which are very much like identifiers in other languages and it can contain letters, numbers and symbols. The three magic atoms are nil, TRUE and FALSE.Numbers, Strings and Comments are also basics in JESS but do not need any further explaination. Then finally it is lists which are very much like lists in LISP and are like arrays in JAVA. The first element in a list is called the list's head and is the same as car in LISP. I also read through Functions, Variables, Deffunctions and Defadvice and more on that will come later.		
Second day of Jess	12.01.2004	00:12:19
Have continued to read about Jess. Need to sleep on what I have been reading. More detail on Jess will be in tomorrow.		
Jess continues	13.01.2004	00:18:51
Well, what I have read on Jess looks very good for my project. First some notes I wrote on paper while I was reading. Jess has something called deftemplates which are equivalent to Java Beans. Deftemplates are unordered facts. Deftemplates can extend. ppdeftemplates, where pp stands for pretty print. There are inheritance in Jess. I do not fully understand this Bean connection from Jess to Java and vice versa. Look better at		

Java Beans, Jess defclass and definstance. LSH = match facts. RSH = function calls. I took no notes on what I read today. Now I am going to go through what I have read so far and pick out things I want on paper. From the paper Functions: Function calls in Jess are simply lists. Ex. (+ 2 3). The outer function is responsible for the inner function calls. Variables: Programming variables start with ? and multivariables with $. Valid global variables names look like this ?*a* and are not destroyed when reset is called. Deffunctions: Just read from papar. Defadvice: Defadvice is a tool to help you alter a function without having to change all the code you have written. It is a way to for Jess to add e.g. 1 to a function without having to change the interal code. Java reflection: Java reflection is as the name suggests have Jess reflects on Java and vice versa. Best to just read chapter in paper and look at table. Next are The knowlegde base and Rules & Queries. I will add that in tomorrow.		
Assignment from Tim	17.01.2004	22:47:17
I went to see Tim on thursday and he gave me an assignment which I have to finish before tuesday. The Assignment was that I should download and install Jess, learn all the examples which come with Jess and make a little program by myself. Hope it will be a good sunday. I will write about he knowlegde base and Rules & Queries soon. Now I am try to look at a Jess book which has the name **Jess in action** by Ernest Friedman-Hill. It looks good a maybe worth buying. Enough for now.		
Yesterday	19.01.2004	21:29:18
Yesterday, was a bit odd. I went to UNAK about 12.30 to get Jess and some space to work. I started with some local web browsing like usually. Then when I went to get Jess to internet connection for outside Iceland was dead. So, I waited for about 1 and half hour. I easied my wait by talking to the girls. Then about 14.30 I was getting tired of the wait and said that I was going to wait about half an hour longer for the chance to get Jess and then I was going home. Then shouted Maria, do you need Jess? I got Jess right here on my laptop :-). Great. So, I unzipped Jess and got all the examples to run before I had dinner at my mothers house. Enough about that odd day.		
New assignment	22.01.2004	14:37:13
1. Introdution to Jess for Tim. 2. Input, Rules and Output. 3. What are the inputs going to be.		
Making the introductory	25.01.2004	14:36:05
I have skipped writing to the journal for some day now. The main reason was that I was going through the examples which came with Jess and trying to understand them and the languge better. I think I can use Jess with this project. Today I am going to do a two page introductory to Jess for my dissertion and for Tim so		

he can follow the project better. So, on I go.		
More Jess reading	25.01.2004	21:38:56
Today I have been reading more about Jess from the book Jess in Action which I loaned from Óli Jóns. I am both trying to get more familiar with Jess and in the way trying to get more information to put into the Introductory I have to do for Tim and my dissertation. Here are notes which I took from the book. After reading more from the book I decided to buy it from amazon.com. It should arrive in two weeks time. Enough for now.		
Last couple of days	29.01.2004	14:19:08
I have not written much in for the last couple of days. I have not been in the mood. On tuesdaynight I fell asleep around 8pm and last night I was not is the mood for the project. On the other hand yesterday I continued to decide how the input into the Jess should be. I have come a long way in designing the database I am going to use in the project. I will put in a photo of my structure tomorrow. Designing the database was much harder then I expected. My newest design has the tables: Course, Holes, STATS, Rounds, Players and AVG_STATS. About the introductory I have not gone far with it because of the time spent on the db design but it will be made. Enough for now, I am going to a meeting with Tim. We decided to have weekly meetings at 14.15 on thursdays nowon.		
Last week	10.02.2004	11:52:29
Hello, journal, long time no see. Last week I have been designing and then starting to build the database I will be using in this project. I has been much harder then I thought it would be. Over the weekend I have making interfaces in PHP for every table I will have to input into before I can input into the big stats table and the average stats table. The average stats table will be made out of queries from the other tables. I will write more tomorrow because I will have to finish three things for Tim so I will not have to pay for the next beer and milk night.		
A good 9 days	19.02.2004	23:08:32
Well, Well long time no writing into my beautiful journal. Last time I wrote I was trying to make Tim's deadline which I did. He has to pay beer and milk night :). Well, on the night of 11.february I had ready all the interfaces I think are needed for inputting the data into the database. These interfaces are: golf.php, round.php, tees.php, holes.php, course.php and player.php. These name are pretty much self-explaining about what is being inputted into the database. On that same night all except one of the tables where ready in the database but with very little or no data. These tables are in the same order as the interfaces: stats, golfrounds, tees, holes, course and player. Only table missing was avg_stats which will contain the average stats for each field in the stats for each golfer. Then it happend, on the 12th the server I keep my database and files on broke down. It lost connection to the internet and I could not access it. It was fixed the day after but then I was going to Reykjavik to be a roadie for the great band Hvanndalsbræður. I did no work on the project over the weekend. This week has gone into finishing the avg_stats table in the database and producing SQL statements to query the stats table to get the data needed for the avg_stats table. The data		

in the avg_stats table will then be used as the data which goes into Jess. Jess will then have rules which will be activated if fullfilled.
All the SQL queries are ready to rock and roll except for the statement for the drive stats. The reason for that is that holes that are par 3 are can not be included into the drive stats because a golfer doesn't drive on par 3 holes. I am almost there, just have small problem joining tables together, nothing that will not be solved.
Next steps are to finish these queries, get the avg_stats data into Jess and try to produce some rules for that data. It looks easy just saying it but it will be pain in the ass :-). I think that is good for now.

Saturday after Ytri-vik	22.02.2004	10:48:36

Yesterday, was not very productive. I went to Ytri-vik with the department and then to Kaffi Akureyri on friday night and wasn't home until 5am. Tim also gave me a very good sandwich. The only thing I did yesterday was to start to put in some **test data** into the database. That is all for now

Great day	27.02.2004	00:02:04

Finally after great deal of fuc**** work the SQL queries seem to work. Now I will put my energy into finishing SIM and COM and then I will continue this project by starting to get the data I need into Jess expert system. When I have finished that I will make some rules in Jess to help the golfer to get better at golf. Enough, I am going to sleep.

Finally time	16.03.2004	22:10:19

Hello my wonderful journal. I had finally time to work on my final year project after more then two weeks of projects in the other courses. I would think that the teachers would talk together to spread the projects but no, all in the same week. Enough about that Well, today and yesterday I have been reading in the Jess book to have get some ideas for the next part of this project which is the "putting the data into Jess facts" part. This will be done by taking the data from the database into Java and from Java into Jess which will do some great thing with it. Either Jess itself or Java will then give the final output to the user. This is the most important part of the project. If I can make rules for two or three things and output it, the project is save :-).
In this reading through the Jess book I have learnt a lot of things I can use the assignment for the Expert System, that is just great.
The final part of the system would then to have a database of suggestion how the user could improve his golf game. I will keep my fingers crossed for all of this to work out before 17.april. Well this has been nice, hopefully I will write a lot this week.

Yesterday	18.03.2004	06:08:19

Yesterday I tried to get a connection from my MySQL database to Java but I did not have much luck. I am pretty convident about that next steps I am going to take in this project. I will continue today on getting that connection going and hopefully also I will be able to get some final idea on how I am going to get the needed fact into Jess. Hopefully then the rule will not be much problem. Enough for now.

The connections	19.03.2004	22:20:23

Well, this evening I have been reading about how Jess is used inside Java and how this then can be put onto the web by using Java Servlet and Tomcat. This is a little bit of a

shock for me but nothing that I can not overcome. I was hoping I would not have to involve Java Servlet and maybe I will not have to that will time see. Jess is just imported into Java like any other package. After that it can be used almost like you were in Jess. One thing I have not completely understand and that is where to store the Jess file containing the Rules. Deftemplates, deffacts and etc. can be inputted into Jess while making a Java program but the rules seems to have to be made in a serparate Jess file. I will figure that out :-). I have briefly start to make a draft of the deftemplates I am going to use for this project and also some simple rule that will fire up the result of the diagnostic of the inputted average stats . This Jess book is going to be a good help in the project. The more I dig into it, more I see how useful she is. Enough for tonight.		
Jess Licensed Version	20.03.2004	09:19:36
I have continued to read through the jess trying to make sense to it all. Now I am hoping that it will all make sense when and if I will get a licensed version of Jess. I just a evaluation version and it does contain the source code which I am sure I will need to finish the project this way. So, until I will get a licensed version I will concentrate on making the rules ready. Enough for now.		
Pretty good weekend	23.03.2004	08:28:04
Good morning, this last weekend was very good for the project. I was able to put in facts, which where build from the avg_stats from mysql, and make some rules and get out some feedback from Jess. I am going to throw in some more rules and hopefully Tim will think that is enough for this middle part of the project. The Jess I use is rather simple but I think it enough so I can write in my dissertations that more can be done with this project in the future. When I am finish with making the rules I will start on a simple database tables which will contain the suggestion I want to give the player so he can improve his golf game. Hopefully I will be able to stick to the time plan from the interim report and be finish coding on 1st of april, sounds like a good joke :-).		
Rules and Output :-)	24.03.2004	23:40:18
Since last time I wrote I have managed to make some more rules and get output. What I did for output was that I made I new database tables which will store the recommendations for the player. I will need to put in some data into the database but I am going to wait a little bit with that. I do not thing that it is the most important thing now. Next is to let all the project bits work together as a whole. I will be trick because the are in three different languages which will require two different mahcines (most likely apache and tomcat). When that is finish then my programming is finished. I will probably have to make my code on pretty and nice but that will be saved for last. Tomorrow I have a meeting with Tim about what I have done so far, what my next steps are and the dissertation. Enough for now.		
Last days	29.03.2004	18:21:39
Since my last confession I have made some changes to the interface of the program. I have added in the recommendation.java that user can get different recommendations depending on their handicap. You can give a player with handicap of 6 recommendations		

in much more details then a player with the handicap of 35. I have started writing and constructing my dissertation. I am trying to use as much as I can from the interim report but much more have to added into the dissertation. Tim gave my a rough outline of how the dissertation should be and I am working on it with that as my guiding light ;-). Today I started to make my presentation. I have no clue what to put into it but I will try to have it on topics I now something about so I can answer some questions afterwards. Well, enough for now.		
Presentation and days after	05.04.2004	13:52:37
Well, now is the time to complete the work. Now there are 11 days until the dissertation is suppose to be in. So, lets get on with the butter. After the presentation on thursday, which went OK I think, I had a complete voltage drop and felt like this project was over but it sure is not :-). The plan for the last 11 days is like this. Tomorrow I will continue with writing the design. On wednesday I will write the implementation. On thursday I will write the last chapter. On saturday and sunday I will do the related work. On monday I will put all the things together and try to form a final dissertation. The last four days will be just to make the report and the system good enough for a passing grade. This looks like a plan to me ;-). Well lets get on with the work.		
Connection Interfaces	06.04.2004	08:26:14
Yesterday, I was going through the interfaces and connection them together so the user of the system will have a good flow through the system. I will not spend anymore time on the system itself until the dissertation is ready. The dissertation is much more important than the system being fully functional. Lets keep on working.		
A good 9 days to go	07.04.2004	16:37:47
Now is the time to dig my heel into the ground a work nostop for the last nine days of this project. Last couple of days have gone into finalising the system and structuring the dissertation. No time write anything.		
7 days to go	09.04.2004	10:30:53
Hello, now there are only 7 days to go. Yesterday I was working on restructuring the dissertations and writing more in the design part. Hopefully , today will be a good day. Today I am going to try to find some related work. Enough for now.		
The last confession	15.04.2004	15:55:01
This will be my last confession into this journal of my final year project. The last 7 days have been days and night work and hopefully that comes to an end tonight. The hand in deadline is 10 o'clock tomorrow morning. I would like to thank my journal for the support is has given me over the journey. Bye Bye forever		