Háskólinn á Akureyri
University of Akureyri

Viðskipta- og raunvísindasvið
School of Business and Science

TJALDUR

aðgangstýrikerfi fyrir tjaldsvæði

(Campsite RFID Access System)

Final Year Project
2010

Ásgeir Hreiðarsson

Háskólinn á Akureyri
University of Akureyri

TJALDUR

aðgangstýrikerfi fyrir tjaldsvæði

(Campsite RFID Access System)

Final Report

Ásgeir Hreiðarsson

Supervisor: Nik Whitehead

Computer Science Division
Faculty of Natural Resource Science

School of Business and Science
University of Akureyri

Submitted April 2010, in partial fulfilment of
the conditions of the award of the degree BSc.

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

Signature _____

Date _____/_____/_____

Table of Contents

Table of Figures

Table of Tables

Abstract

This report is about a RID access system for campsites. It is done as a final year project in the University of Akureyri. In this project the aim is to build a simple system that can manage the access control of the campsite and give statistical information both that are obligated by authorities and also can provide practical information about the practice. The program will be written in C# using MS SQL as a database and ISO 15693 13,56 Mhz data carriers. The minimal solution is supposed to be able to take care only f the access control but if possible cause of time issue a full version would also take care of other parts of the campsite service such as, electricity, laundry, rentals and internet.

1 Introduction project objective

1.1 Description of the problem

One of the major issues for all businesses is how to have good control of the products they are offering. The idea must be to have a minimal cost of maximizing the income. At a Campsite the product you are selling, is the access of your facilities, to customers who are coming and going. So to be able to get the most out of your product you have to be able to control, who is entering and leaving the site. Since campsites are generally thought of as low budget options, there is not a lot of money to invest in expensive solutions for monitoring and controlling the guests at campsites. This problem is properly the same or similar for all campsites, but in this project the focus is on a local campsite called Hamrar. There the monitoring and control is through visual observations of a staff member. The sale process is all done in hands making handwritten receipts, stickers and papers for the customer to have visible in their cars, on their tents or caravans. These papers are used to validate the payment at each entrance of a customer. Afterwards the information is put in to a computer system to manage the reports that are demanded from the authorities and help with financial reports. Since the campsite is rather big it takes staff members a long time to walk to every tent/caravan/car and check for valid payment receipts. There is only one access point to the campsite and at that point there is a gate barrier manually controlled by a staff member, if the staff members are not at the gate the control is switch to self control by customers and then anybody can manually open the gate.

1.2 Motivation for solving the problem

Being a manager at Hamrar, one of the biggest campsites in Iceland for seven years made me realize that controlling the access to the site is one of the big problems this business faces. Although having a fairly big site the debate becomes how to minimize the cost of staff against maximizing the income. This also relates to the issue of making the customers happy and giving them better access to the site and its facilities. There is commercial software out there that would properly meet most of the demands for the Icelandic marked, but it is in a price range that our market does not afford. Altering some normal cash desk solutions is also a possibility but would properly proof to have high cost in the end. These reasons have driven me to try to make a rather simple solution to this problem that would not cost so much to manufacture taking both hardware and software into consideration.

1.3 Proposed solution

The aim of this project is to make a solution that will take care of controlling the access to the campsite and making statistical and financial reports on the sales. It should also be able to take care of other items that are part of the service at the campsite such as electricity, laundry, internet and rentals. The main focus is the access system and making that in such a manner that it will prove to be efficient and user-friendly.

1.3.1 Programming language

I decided to use C# as the programming language for me solution for two reasons. One was that I wanted to learn more about .net programming since those kinds of programs are very common now a days and learning about more about it might help me later on in me professional live. The other reason is that I saw that I could get support from the manufactures of the selected hardware solution

I had chosen in C#: This help was also available in other programming languages but since I was also interested in learning a more about .Net programming this was the obvious choice. The hardware provided examples and libraries' for other languages such as Java, C, C++ and Delphi. I have taken courses in the University of Akureyri about both Java and C. Therefore it might be easier for me to program in either of those languages but I found it more challenging and interesting to learn also about C# and, as I already mention above it gives me new skills to bring from my studies.

1.3.2 Tools

For this project I use survival tools and software packages to both develop and write the program. My choices in these tools are not always delivered as the best possible/available tools but more or less those how I had Access to and was able to use to do the job.

1.3.2.1 Microsoft Windows XP

The operation system used for this project is Microsoft Windows XP 2002 service pack 3.0

1.3.2.2 Microsoft Office 2003 and 2007

The Office 2003 and 2007 packages are use extensively for all documentations, spreadsheets, communication and demonstrations of the solutions.

1.3.2.3 Freehand 10

Drawing the storyboards I used Freehand 10 from Macromedia. I have a good background knowledge about this program from my days in the printing business, where I worked as a layout person for five years.

1.3.2.4 Gliffy

I used Gliffy a web based program to draw all my diagrams. This enabled my to provide my instructor an easy access to what I was drawing.

1.3.2.5 Freemind

I use Freemind 0.8.0 to draw a Mind map of the possible solution for this project.

1.3.2.6 Microsoft Visual Studio 2008

Since I had access to Microsoft Visual Studio I decided to use that software. It gives me an opportunity to study common and widely spread software that offers good support and documentation on C#.

1.3.2.7 Microsoft Project 2007

When planning the project I used a evaluation version of Microsoft Project to make a Gantt chart of the plan.

1.3.2.8 SourceMonitor

I used a software called SourceMonitor, version 2.5.1.0 from Campwood Software (www.campwoodsw.com) which is a tool that measures and records source code metrics.

1.3.3 Hardware

When selecting the hardware to use my main focus was on the cost because of personal reasons and also because that is one of the issues that will come up when the campsite would be deciding if they want to use this solution.

1.3.3.1 RFID reader

When looking at available readers I decided to focus on products that had code examples offered and since I had decided to use C# I looked at if this hardware had support for that language. There are a lot of different versions out there with various different frequencies and standards. The reader I am using is working on 13,56 Mhz using the ISO 15443 standard. It is called APSX RW-310 RFID Reader/Writer Module (TTL). I have to add a connector to the reader to connect it to PC computer and that is called APSX C-100 TTL to RS232 Level Converter. Both these Items are made by a company called APSX, LLC.

1.3.3.2 USB digital in and output

I needed a solution that would give an outgoing digital signal to the gate barriers to open. I found a USB connection card that provides 5 digital inputs and 8 outputs plus 2 analog inputs. It is called K8055 (USB Experiment Interface Board) and is manufactured by Velleman Instruments. This was a cheep solution giving me exactly that, it has a simple USB connection to the computer. There are test buttons to test inputs and signal lights to test outputs. It also provides code examples and dll files to support it.

1.3.4 Database

When choosing what database to use I looked at several options including Oracle, MySQL and SQL2005. Oracle offers a free version called Oracle Database 10g Express Edition which has a good reference pages on how to work with Visual Studio. MySQL

1.4 Timeframe

Planning this project I used two methods first I made a Gantt chart to plan the project, using Microsoft Project 2007. This gave me an overview of how to divide my time to each parts of the project. Then Later on I made a task list for me based on the Gantt chart and used that to follow up on my plans. There I had a more detailed description of each task and that helped me to organize my work. This task list ended up as the first draft of the interim report. I will update and evaluate both in January. Both the Gantt chart and the task list can be found in Appendix A.

2 Access control

A system that enables an authority to control access to areas and computers is called an access control system. In reality it is an every day phenomenon like a lock to a car, a Pin to ATM at a bank or a bouncer standing in front of a night club. All these things are a form of an access system. When talking about access systems we can define them as physical access or computer security (Wikipedia, 2009). In this document we focus on the physical access part since that is the main issue of the project. Computer security is of course also an issue regarding every system that is based on a computer working program but that is another thing that will not be dealt with now.

2.1 Physical access

The physical access to a place may limited based on payment, statistical information or only to control traffic such as one way pass through. Using different method this can be achieved, a doorman, a ticket checker, a security guard, a turnstile or with a barrier gate or a turnstile. With this would likely be some kind of fencing or walls to control or guide people to the access control point. Normally this is done on an entry point but could also be done at the exiting point such as at a shops

check out. A physical access control is a matter of whom, where and when. It controls the entry and exit of a physical thing. Keys and Locks have though history been used to do this but using a normal key and lock you can't control when and you won't be able to look at records of who and when the lock is opened. Using a computerized electric lock solves a lot of the limitations of mechanical key and locks. Using a electric lock and key allows the controller to decide when, where and who can open the lock, it can register the time, date, and key holder that opens the lock. It also gives the opportunity to deny access to a lost key so changing locks and keys if no longer a problem (Wikipedia, 2009). The authorities at a given place can get a lot of data about the lock and actually also about the users using that lock.

2.2 What information to gather

Normally a electric lock would check that the electric key has a valid number that the computer software recognizes and stores in a database. The information that is gathered in the access system can vary a lot between systems. A system can combine the key number to series of other transactions collecting a vast amount of data regarding the key holder. Often the access key can be used as an identification card for the key holder. This races the issue of personal integrity, while giving the system owner an opportunity to use this data to analyze the user's statistics. This also provides the owner of the access system an opportunity to gather data that he is required to collect and turn in to governmental authorities. This is the case in Iceland for campsite. According to a regulation nr. 163 21/2007 about Statistics Iceland, they are allowed to require campsites to return information about number of people from each country staying at the site each day. The financial profit of having a good database about the campsite customers in order to advertise, sending special offers and giving discounts to frequent users, can be significant. On the other hand it can cost a lot of money, space and time to collect to much information in to the system.

2.3 Personal integrity

When collecting data about customers the issue of personal integrity arises. The boundaries of public and private have to be considered. Gary T. Marx wrote an article about ethics for surveillance, in the Information Society, Vol. 14, No. 3, 1998. There he summarizes his findings about the ethics of a surveillance activity, in 29 questions arguing that it must be judged according to the means, the context and conditions of data collection and the uses/goal. A table with the questions can be found in Appendix E. They can be used to realize better the ethical question of whether the data collected is personal or private. Not only is this an ethical question because according to Icelandic laws personal integrity has to be considered (Regulation nr. 70/2000). To avoid problems related to this a statement from the customer about his awareness of the information gathering of the system might be needed.

2.4 Measures to control the access

Using access control the authority in place faces the question of how to control or what methods to use. The first question is whether the control will be done visually by a person or automatically by a computerized system.

2.4.1 Visual observation

If a visual observation is used then the person involved has to look at some kind of verification, paper ticket as used in many heathers and cinemas, trains, buses, stadiums etc. Discos and nightclubs

sometimes use stamps on the hand and the often using an invisible ink and black light to verify it. Stickers are used on places like campsites and ski areas.

2.4.2 Computerized systems

A computerized system often uses a single factor transaction to compare the credentials of the access control. These factors are often characterized as,

- Something you have, such as an access badge, ID card, barcode ticket, implant or key ring,
- Something you know, e.g. a PIN or password
- Something you are, typically a biometric input (fingerprint, eye scan, voice recognition, picture recognition).

Using only one of these factors especially the first one the security threat becomes that a user passes his badge or pass around to other users. To prevent this many system use at least two factors a pass end PIN number or password (Wikipedia, 2009).

2.5 Have a look at existing programs for campsites

Looking for access control for campsite I found three products that all are commercial and offer a booking system for campsites (Campground manager, Campground master, NO Q Caravan). I downloaded a trial version of all of them and had a look at how they work. Since the all worked in a similar matter I will not go in to detail of them all. On figure 1 is a screen shoot of Campground Master booking form where is can be seen that all bookings are made to a certain place on the campsite. That means that the campsite has to be divided up to small units where a customer can order a specific place on the campsite or at least when he arrives he will get a certain place to camp on. This is something that would not work at Hamrar at present at least. All these software solutions also seem to require a lot of data input about the customer, which would mean increased time in each transaction. There are some interesting points that can be learned from looking at these programs especially regarding bookings in advanced although I will not be able to use it in my solution. (Campground manager, n.d.; Campground master, n.d.; NO Q Caravan, n.d.).

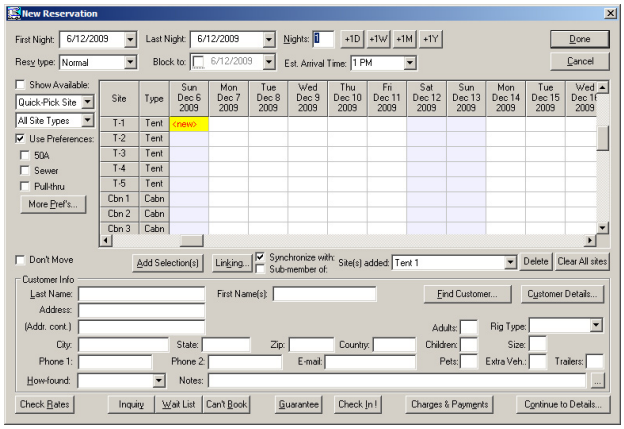


Figure 1 – Sales window for a access control system (Campground master, n.d.)

3 RFID

3.1 What is RFID

The abbreviation RFID stands for radio frequency identification (Finkenzeller, 1999). RFID was first conceived in 1948, but it took many years for the technology to evolve to the point where it was sufficiently affordable and reliable for widespread use. RFID is today a generic term for technology that uses radio waves to automatically identify objects or people. RFID is, in its simplest form, similar to bar coding, but its improvement over bar codes is in terms of information density, non-optical proximity communication and two-way communication ability (Roberts, 2006). FFID is similar to smart card system in a way that data is stored on an electronic data carrying device – the transponder. Unlike smart card it uses magnetic or electromagnetic fields. This means that the power supply to the data carrying device and the data exchange between data carrying device and reader are achieved without the use of galvanic contacts (Finkenzeller, 1999). The RFID system is always made from two components: The transponder that is located on the object to be identified and the interrogator or reader, which, may be a read or write/read device depending upon design and the technology used. A reader normally consists of radio frequency module (transmitter and receiver), a control unit and a coupling element to the transponder. Often an additional interface (RS 232, RS 485) is fitted with the reader enabling it to forward the data received to another system (PC, robot control system). The actual data carrying device of an RFID system is there for the transponder which typically contains a coupling element and an electronic microchip. When the transponder is not within the interrogation zone of a reader it is totally passive. It only activates when it is within the interrogation zone of a reader (Finkenzeller, 1999). Figure 2 shows a typical RFID system that is normally comprised by following objects: an RFID device (tag), a tag reader (with an antenna and transceiver) and a host system or connection to an enterprise system (Roberts, 2006).

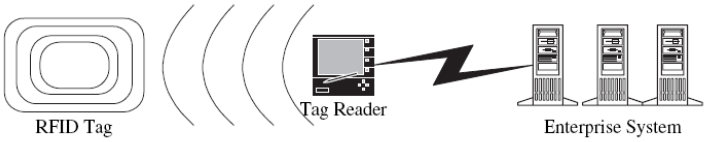


Figure 2 – Typical RFID system

3.2 RFID devices fall into two broad categories

"Active tags" which has read/write devices and are with power supply (battery) and "passive tags" which are read only and unpowered devices. Passive tags are smaller and cheaper than active tags. They have unlimited live and are lighter and they require a higher-power reader than active tags. Active tags have not more than 10 years of live time, due to the usage of a battery, but they have bigger storage capability and a longer read range than passive tags. There are also semi-passive tags where the battery runs the chip's circuitry but the device communicates by drawing power from the reader. Tags are available in different shapes, sizes and protective housings. Devices injected beneath the skin of an animal for tracking purpose are approximately 10 mm long and 1 mm in diameter. The smallest devices commercially available are thinner than a sheet of paper (0.4 x 0.4) (Roberts, 2006).

The transponder usually doesn't possess its own voltage supply (battery) so the power required to activate the transponder is supplied to the transponder through the coupling unit (contactless) as it is the timing pulse and data (Finkenzeller, 1999)

3.3 Frequencies

RFID is radio frequency base equipment that uses electromagnetic frequency to communicate or to obtain power. The electromagnetic spectrum in the range that RFID uses is normally divided up to four categories, low frequency (LF), high frequency (HF), ultra high frequency (UHF), and microwave. This difference between these frequencies is demonstrated in table 1. These frequencies are set a site for ISM (Industrial Scientific Medical) so they won't interfere with other, protected applications, such as emergency service radios or television transmissions.(Bhatt & Glover, 2006) From that table it can be seen that that HF is the suitable frequency for building access control, but if we wanted to have a longer range then UHF might be a better choice. This would mean that a customer at the campsite would not have to leave the car to get the reader to read their card. On the other hand when thinking of a broader use of the cards it would not be vice to have such a long reading range for opening of doors, operating laundry or electricity. Therefore the HF is the best solution for this project.

Table 3-1. RFID frequency ranges

Name	Frequency range	Typical max. read range for passive tags	ISM frequencies	Some typical applications
LF	30300 kHz	50 centimeters	< 135 kHz	Pet identification and close reads of items with high water content
HF	330 MHz	3 meters	6.78 MHz, 13.56 MHz, 27.125 MHz, 40.680 MHz	Building access control
UHF	300 MHz-3 GHz	9 meters	433.920 MHz, 869 MHz, 915 MHz	Boxes and pallets
Microwave	> 3 GHz	> 10 meters	2.45 GHz, 5.8 GHz, 24.125 GHz	Vehicle identification of all sorts

Table 3-1 – RFID frequency ranges (Bhatt & Glover, 2006)

3.4 Standards

It is very important for the manufacture of any device that is it is produced according to a certain standard to insure that different manufactures will produce a product that work with products from other manufactures. To do this a technical committee of ISO develops a standard. The ISO is a worldwide union of national standardisation, similar to DIN (Germany) and ANSI (USA). RFID uses several ISO standards depending on the use of the RFID. Animal identification uses ISO 11784 and 11785, Contactless Smart cards uses ISO 10536, 14443 and 15693 standards, data carriers for tools and clamping devices uses ISO 69873 standard and Container identification uses ISO 10374 standard (Finkenzeller, 1999). In his project the focus is on Contactless Smart cards. There are three standards to choose from the first is the ISO 10536 close coupling cards that are designed for ranges between 0.1 cm and a maximum of 1 cm (Finkenzeller, 1999) . The second is the ISO 14443 or the Proximity smart cards with reading range up to 10 cm (Axess AG, 2009). The third is the ISO 15693 or the Vicinity coupling smart card with reading range up to 50 cm(Axess AG, 2009). Using the ISO 15693 the user doe not have to be as close to the reader as the others and by that he could easily get the

reader to read the card without having to go out of the car, but still not activate a read just by passing by.

3.5 RFID Tags

Because RFID tags must physically attach data to items of different shapes and sizes in different environments, they come in a wide assortment of shapes and sizes. Furthermore, they may be housed in many different kinds of materials. Some of the physical characteristics of various tags include:

- PVC or plastic buttons and disks, usually including a central hole for fasteners. These tags are durable and reusable.
- RFID tags shaped like credit cards, which are called "contactless smart cards ."
- Tags made into the layers of paper in a label, called "smart labels ." These may be applied with automated applicators similar to those used for bar code labels.
- Small tags embedded in common objects such as clothing, watches, and bracelets. These small tags may also come in the form of keys and key chains.
- Tags in glass capsules, which can survive even in corrosive environments or in liquids.

(Bhatt & Glover, 2006)

3.6 Looking at RFID solutions

RFID is used for a lot of different things. Amongst those are access control systems. Working on a solution for such a system limited the scale of things to look at. In chapter 3.7 I will talk about RFID for campsites but in this chapter I talk about how a possible solution could be in using an RFID system not intended for access control.

3.6.1 NFC solutions

Near Field Communication (NFC) is a new technology designed to extend RFID and has the potential to bring mobile devices and physical objects together. Using NFC the aim is to enable users to interact with augmented objects or other devices intuitively, offer a long duration of connection establishment and high power consumption (Falke et al., 2007). The Idea of using an every day object like the mobile phone as a access key in access systems is very appealing. The NFC technique offers not only the usage of the phone as a data carrier but also as a RFID reader to read from other RFID tags. This option also opens a security threat since the phone can read other RFID cards and simulate them. The main problem with this solutions is though that it is not a wide spread solutions in mobile phones and therefore not likely to be feasible to use at the moment, but if they continue to manufacture phone with this function it could be an interesting addition to my project.

3.6.2 Mirror/rabbit

My instructor pointed out to me a product that works with RFID and can be controlled over the internet. It is a product call the Mir:ror manufactured by Violet. It allows the user to program an RFID tag, that comes in form of rabbits and stamps. The user can decide what action will be taken by the computer (software) when the Mirror (RFID reader) reads the RFID tag of the rabbit or the stamp. The stamps can be glued on any thing the user would like to use to manipulate the system. When the tag is read the software will start any programmed feature chosen. This could be to change a status on the MSN to drinking coffee, when a coffee mug with a stamp on is brought to the mirror. It could also be to run a Marco that the user has written. Although it is a neat product given great options to

make the computer able to interact with physical things, I found it to be to complex to work with for my solution and therefore decided not to use it (Violet, n.d.).

3.7 Existing programs for campsites

Looking for RFID solutions for campsites I did not find many products, in fact only two. These products are both used in a lot of campsite all over Europe at least. I was not able to get a lot of technical information about these products since they are both commercial products. I used their homepages and material form there to investigate how they work and what they offer.

3.7.1 Compusoft

Is a Danish company that produces an access system used by Campsites, Hotels & Hostels, Resorts and Swimming pool's. The offer a wide range of options including, Web-applications, Online booking, Self-service terminals, cash desk terminals Text Message Gateway and many others. In their system the user can control, gates, doors, electricity, laundry, internet, lighting, heating, burglary alarms and so on. It seems to offer a complete solution for every aspect of running a campsite. There are more that 2000 campsites using this solution already with over 80% of all campsites in Denmark (Compusoft, n.d.). This product uses RFID cards but I was not able to get any technical information from there website about standards or frequencies. They also offer a picture of registration numbers to compare the car at a gate which is an interesting idea.

3.7.2 MADA

Is a company from Germany that specialises in RFID techniques. Their products are visiting control, campsite control and petrol pumps control. They also handle with data carries for RFID and printing on data carries. Their campsite system offers a wide range of possibilities controlling doors, gates, laundry, electricity and lockers. They offer a web booking system, self service automats, charging stations and cash desk solution (MADA, n.d.). This seems to be a simple hardware that giving most of the things needed for any campsite. According to their website there are over 100 campsites using their product already. I found little information about the technique used in the system but the state that the system works for main RFID card producers (Mada, n.d).

4 Requirement analysis

To be able to analyze the specific needs for this project several methods where used, including interviewing former staff members from the campsite, brainstorming with them, interviewing the manager of the campsite about his expectations to such a system, background reading and testing on other products. Then using object oriented methods I started to form use case diagrams of the possible solutions to this project.

4.1 Interview former staff about expectations

On a meeting with two of former staff members in Reykjavík on the 18th October 2009 a walkthrough of the normal workday was done. We especially focused on the work procedures of making a sale to a customer. There where discussions about how a computerized system might be evolved and what aspects had to be considered in such a system. Today the transaction goes something like this.

1. Staff member writes down the date and his/her name on the receipt.
2. Customer is asked for:

- Nationality (only one is optional per receipt)
- How many nights he/she is staining
- How many adult (14 and older)
- How many children (13 and younger)
- If he/she wants electricity hookup

3. Staff member calculates the total amount and finishes writing the receipt.
4. Staff member writes a sticker for the tent/caravan/trailer
5. Staff member writes a ticket to have on display in the car (different colors depending on departure day).
6. Customer pays and collects all three validation papers (receipt, sticker and car paper) and gets info about where to stay and info about the service on site.
7. Staff member puts the money in the cash register and later on will put the info from the receipt into a Microsoft Access database for statistical purposes.

This procedure demands a lot of handwriting different validation papers and offers a lot of possible errors to creep in. It takes some time to write all these things down and the form used limits somewhat different combinations of Customer s needs. This can for example be when the customers are travelling to gather but have different nationalities or are planning on staying for different length of time. These situations would require staff to make two or more receipts for one transaction. An other issue that came up in our discussion was the fact that it can be hard to notice through the window if the customer in a car has the right colored piece of paper in the car window or even if it is a valid kind of paper. And further more if there are many Customer s waiting for service the handling of control the gate often becomes a disturbance for the staff and annoyance for customer with valid entrance papers but waiting for the staff to open the gate.

After going over how things are, we started to discuss more how this solution could work in practice. How other items in the service could intervene in a computer based version of the cash register. We ended up with a brainstorming session where all possible things where open for suggestions about the possibilities of such a program.

4.2 Interview manager of campsite about expectations

Talking to the manager of the campsite quickly brings up the issue of cost, both regarding investment and maintenance. The campsite is open only 3-4 months a year and having as low as 5-6 weeks of that period as the high season. Although Hamrar is a fairly big campsite the income doses not support big spending on computer systems. Leaving that issue to be dealt with later, we focus on what elements would be beneficiary for the site to have in this solution. The key issues the manager had was that:

- it would be easy to operate,
- it would prolong the time it takes to serve one Customer ,
- in case of a computer failure there should be an easy backup system to operate the access control manually,
- be able to handle fast service on busy days,
- Give an clear info about number of guest on the site at a given day
- Make the closure of the cash desk easy

- Minimize the workload of inputting data to the system
- Be able to add cash desk computers if necessary
- Have an opportunity to presale on line

From this I got a different aspect of things since I have focused a lot on how to make everything computer involved, but he comes up with points about what to do when the computers or software might fail. Also the issue of fast service for busy days is something to bear in mind while making this solution.

4.3 Mind map of solutions

After having a meeting both with former staff members and the current manager of the site I had a better picture of what the possible solution could involve. Taking those points I made a Mind map of the solution using the Freemind 0.8.0 software my instructor, Dr. Nik Whitehead had pointed out to my during a Lab in Human-computer-interactions course. (See Appendix B). Using this tool I was able to visualize how different ideas fitted together and started to formulate the overall picture of the possible solution.

4.4 Use case diagrams

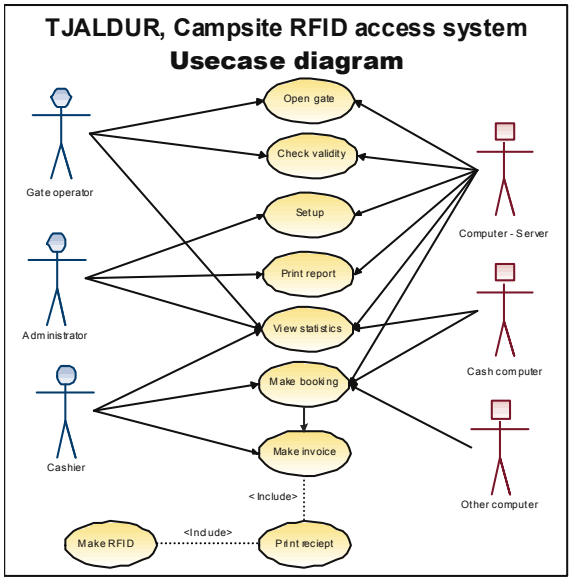


Figure 3 - Use case diagram

Having a better image of the possible solution I started by using Gliffy a web program to make a Use case diagram. Use case diagrams are used to show the functionality that the system will provide and to show which users will communicate with the system in some way to use that functionality (Bennett, McRobb and Farmer., 2006). On the use case diagrams I have three different user types (Administrator, cashier and gate controller) as actors and 3 different computers (server, cash desk and other) running the system. When designing the use cases I kept in mind that one of the managers issues was to have a back up system of using handwritten receipts in case of a failure of the computer system. As can be seen on the use case diagram picture Then most system actions are ran through the server though the server can not make invoices by it self but the cash desk must take care of that part. Also I want the cash desk to be able to run without a fixe connection with the database. Therefore I assume that the database will be on all computers using scheduled synchronisation to keep all computers up to date. This will also help with backups of the database. Nothing will though prevent the campsite from having all interfaces running on one computer with only one database. Looking at the use case diagram we can see that the Gate operator can both work through the server and also manually control the gate. If he is working on the server he can also have a look at statistical information's about number of guest. The Administrators main function is to setup the system and get statistical information and reports about the operation. The cashiers main function is to make booking, invoices, and receipts but he can also get statistical info about the sales. The Server will operate both as a database that all users can make queries to and working with the gate handling the validity check and the opening the gate. The server

will also provide internet sales cannel to the database, enabling a customer to make a booking online. The cash computer will mainly focus on the booking and making receipts and RFID cards. Other computers will be bale to do booking and through the cash computer also to print receipts and RFID cards. In the following sub sections each use case will be explained.

4.4.1 Opening gate

1. Brief Description: After identifying the customers cards validity the system will open the gate or display a red light indicating invalid cards
2. Preconditions: Waiting for a read of RFID card
3. Business Trigger: Customer arrives at the gate reader

4. Basic Flow:

5. Interface sends a signal to output device to open gate
6. Output device opens the gate
7. Interface displays a message about the RFID card
8. Interface sends signal to output device to light a green signal light
9. Gate closes automatically

10. Post Condition: Gate closes and system is ready to open again

11. Alternate Flow: Manual open of gate

12. When in 5
13. User at gate presses manual open button in interface, then:.
14. Interface recognize a push of a manual open button
15. Interface send signal to output device to open gate.
16. Output device opens the gate
17. Interface sends message to database about manual open including time, date, and user
18. Database registers the transaction
19. Gate closes automatically

20. Post Conditions: System is ready to read in a new RFID card

4.4.2 Check validity

1. Brief Description: When a customer comes to the gate he puts the card to the reader and the reader sends the cards info to the computer. The interface checks if that card has a valid pass on.
2. Preconditions: Waiting for a read of RFID card
3. Business Trigger: Customer arrives at the gate reader

4. Basic Flow:

5. Reader reads the card number from the RFID card
6. The interface sends a query to the database asking for the validity of that card number
7. The data base responds with sending information's about what is registered on that card
8. Interface sends message to the output device to open the gate
9. Interface sends to screen info about the owner of the RFID card.
10. Interface sends a signal to a green light indicating that it was a successful run
11. Interface send message to database about the truncation of the RFID card.
12. Database registers the transaction

13. Post Condition: The system is ready to read in again.

14. Alternate Flow: Invalid RFID card at the reader

15. When in 7 the data base sends a message to the interface of an invalid card

16. When the RFID card has an expired date of departure, then:
17. Interface sends a signal to the output device to light a red signal light indicating that the card is expired
18. Interface sends message to screen about the expired card
19. **Post Conditions:** The system is ready to read in again
20. **Alternate Flow:** Empty RFID card
21. When in 7 the database sends a message to the interface of an no record of RFID card
22. When the RFID card has no record in the database, then:
23. Interface sends a signal to the output device to light a yellow signal light indicating that the card is empty
24. Interface sends message to screen about the empty card
25. **Post Conditions:** The system is ready to read in again

4.4.3 Setup the program

1. Brief Description: The setup of the system must be done in the beginning but it can also be updated as needed
2. Preconditions: The system must be installed first
3. Business Trigger: Campsite administrator that wants to use the system
4. **Basic Flow:**
5. User puts in login, password and selects server
6. System allows the user to select setup from menu
7. User select setup menu and selects company Info
8. System displays a fill in form for company info
9. User puts in the info and presses ok
10. System saves settings of company info to Database and will use for all reports and receipts
11. User presses ok
12. **Post Condition:** System has information about the company stored in database
13. **Alternate Flow:** Setup of prices
14. When in 7
15. User select setup prices from the setup menu, then:.
16. System displays prices and discount fill in form
17. User puts prices in to the fill in form
18. User puts in discounts in to the fill in form
19. User presses ok button
20. System stores the prices and discount in the database
21. **Sub flow;**
22. User select cancel
23. System returns to displaying setup menu
24. **Post Conditions:** System stores the data and displays the setup menu again.
25. **Alternate Flow:** Setup of Nations
26. When in 7
27. User select setup nationalities from the setup menu, then:.
28. System displays nationalities fill in form
29. User puts nations and ID number and selects add
30. **Sub flow;**

31. User repeats number 29 as often as needed
32. User select ok
33. System saves list of nations to database
34. Post Conditions: System stores the data and displays the setup menu again.
35. Alternate Flow: Setup of Users
36. When in 7
37. User select setup users from the setup menu, then:.
38. System displays users fill in form
39. User puts users, user type and ID number and selects add
40. Sub flow;
41. User repeats number 39 as often as needed
42. User select ok
43. System saves list of users to database
44. Post Conditions: System stores the data and displays the setup menu again.
45. Alternate Flow: Setup of Map of site
46. When in 7
47. User select setup map of site from the setup menu, then:.
48. System displays maps of site fill in form
49. User hits browse
50. System displays a normal select file form folder window
51. User selects picture file from window and selects ok
52. System displays a preview of the picture
53. Sub flow;
54. User hits cancel
55. System shows setup map of site window again
56. User select ok
57. System saves the map file to the database.
58. System displays a setup menu window
59. Post Conditions: System stores the data and displays the setup menu again.
60. Alternate Flow: Setup of connections
61. When in 7
62. User select setup connections from the setup menu, then:.
63. System displays connections fill in form
64. User fills in info about computers connected to the system
65. System displays list of connected computers
66. User select ok
67. System saves the info about connections to the database.
68. System displays a setup menu window
69. Post Conditions: System stores the data and displays the setup menu again.

4.4.4 Print reports

1. Brief Description:	The system will view reports for certain periods via online request through queries
2. Preconditions:	The user must be logged in
3. Business Trigger:	Campsite administrator that wants to get info about statistic from the system

4. Basic Flow:
5. User puts in login, password and selects server
6. System allows the user to select view menu from menu
7. User select reports from view menu, then:
8. System displays date selection window to choose periods
9. User puts in the period he wants to view
10. System displays a selection window to select what things to view
11. User selects items to view (Guests, Electricity, Laundry, Dryer, Internet, Rental and Other)
12. System displays selected view with details about amounts, numbers names ect.
13. User presses ok button
14. Sub flow;
15. User select cancel
16. System returns to displaying view menu
17. Sub flow;
18. User select print
19. System displays a print dialog window
20. User selects pages to print, how many copies and presses print button
21. System prints out selected pages.
22. System returns to the report displayed on the screen
23. User presses ok button
24. System returns to view menu
25. Post Condition: System displays view menus

4.4.5 View statistics

1. Brief Description: The system will show statistics via online request through queries
2. Preconditions: The user must be logged in
3. Business Trigger: Campsite administrator that wants to get info about statistic from the system

4. Basic Flow:
5. User puts in login, password and selects server
6. System allows the user to select view menu from menu
7. User select view menu and selects status of cash drawer
8. System displays a views that can be chosen from, Amount in cash drawer, Amount of transactions, Amount of guest to day, Amount of sold electricity connections, Availability of electric connections, Status on laundry, Status on dryer, Status on Internet or Status on rental
9. User selects the view he wants to look at
10. System makes query to the database about that information.
11. System displays the spreadsheet about that information's
12. User selects ok button
13. Sub flow:
14. User selects print
15. System displays a print dialog
16. User selects print preferences and hits print button
17. System prints out selected information.
18. System returns to view menu
19. Post Condition: System displays view menus

4.4.6 Make booking

4.4.6.1 Cash computer booking

1. Brief Description: A user can make a booking from a cash register, or through another device such as PDA.
2. Preconditions: The user must be login to the system
3. Business Trigger: User making a booking for a customer on site

4. Basic Flow:

5. User puts in login, password and selects cash computer
6. System displays a booking window with current date as a default arrival date then:
7. User selects how many nights the guest wants to buy
8. User selects how many adults there are
9. User selects how many children there are
10. User selects from what nation the customer is
11. User selects a discount value if needed
12. System displays a line for each person with detail info about nights, arrival date, nation and discount
13. User alters the needed info about each line if needed
14. System calculates price for each line and a total price for the booking

15. Post Conditions: System returns to booking window

16. Alternate flow: Add info about customer

17. When in 12
18. User select add info about addresses for each line
19. System displays a fill in window for (Car number, First name, Last name, ID number(kennitala), Home Address, Postal code, City, Country, Email, Phone number(on site if possible), Photo)
20. User fills in the info about the customer and hits ok button
21. System stores info about customers in database

22. Post Conditions: System returns to booking window

23. Alternate flow: Delete a line from booking

24. When in 12
25. user selects to delete a line , then:
26. System takes that line out of the displayed lines and recalculates the total price

27. Post Conditions: The system returns to booking window

28. Alternate Flow: User selects electricity

29. When in 6
30. User selects electricity hook-up if needed, then:
31. System displays a line for electricity with arrival date, nights, location and hook number
32. User selects number of nights, location, and hook number
33. System displays price for electricity and adds it to total price

34. Post Conditions: The system returns to booking window

35. Alternate Flow: User fills in info about customer

36. When in 15
37. User fills in info about customer, then:
38. System displays a line for travel arrangement, Accommodation and registration number of car
39. User selects kind of travel arrangement

40. User selects kind of accommodation
41. User enter the registration number of the car
42. **Sub flow:** User adds info about customer (Car number, First name, Last name, ID number(kennitala), Home Address, Postal code, City, Country, Email, Phone number(on site if possible), Photo)
43. System adds to price if customer comes by car i.e. needs an RFID card
44. **Post Conditions:** The system returns to booking window
45. **Alternate Flow:** User selects laundry
46. When in 6
47. User selects laundry, then:
48. User selects the amount of washes, driers and powder the customer wants
49. System adds the price to the end of the line and adds it to total price
50. User selects to book a time for the washer or drier
51. System displays a booking window with available times
52. User selects a time and registers info about customer
53. **Post Conditions:** The system returns to booking window
54. **Alternate Flow:** User selects internet
55. When in 6
56. User selects internet, then:
57. User select the time that the customer wants on the internet
58. System adds the price to the end of the line and adds it to total price
59. **Post Conditions:** The system returns to booking window
60. **Alternate Flow:** User selects Rental
61. When in 6
62. User selects rental, then:
63. User selects what the customer wants rent
64. User selects the time of the rental
65. User enters the price for the rental
66. System adds the price to total price
67. **Post Conditions:** The system returns to booking window
68. **Alternate Flow:** User selects other
69. When in 6
70. User selects other, then:
71. User enters the price for the other
72. System adds the price to total price
73. **Post Conditions:** The system returns to booking window
74. User selects kind of payment
75. System displays the total price and the vat in the bottom of the booking window
76. User selects ok
77. **Sub flow:** User select cancel
78. System clears the sales window
79. **Post Conditions:** System saves the booking to the database and makes receipts.

4.4.6.2 Online internet booking

1. Brief Description: A user can make a booking from the internet
2. Preconditions: The user must enter a website
3. Business Trigger: Customer how wants to make an online booking

4. Basic Flow:

5. Customer enters the campsite website
6. System displays a booking window then:
7. Customer selects the arrival date.
8. Customer selects how many nights the guest wants to buy
9. Customer selects how many adults there are
10. Customer selects how many children there are
11. Customer selects from what nation the customer is
12. System displays a line for each person with detail info about nights, arrival date and nation
13. Customer alters the needed info about each line if needed
14. System calculates price for each line and a total price for the booking
15. Customer selects ok
16. System displays a window to add info about every line(Car number, First name, Last name, ID number(kennitala), Home Address, Postal code, City, Country, Email, Phone number(on site if possible), Photo)
17. User fills in info about every line.
18. User hits ok
19. System brings up a payment window
20. User enters payment details
21. System checks validity of payment.
22. System ask if customer has RFID card number
23. User puts in his RFID card number
24. System displays a line for travel arrangement, Accommodation and registration number of car
25. User selects kind of travel arrangement
26. User selects kind of accommodation
27. User enter the registration number of the car
28. System registers the information to the database using the RFID number as reference.

29. Alternate flow: User dose not have a RFID card

30. When in 22
31. User select that he has no RFID card
32. System saves info about customer to database using ID number as reference
33. System displays a message about customer having to talk to staff up on arrival to get RFID card
34. Post Conditions: System returns to booking
35. System displays a confirmation window.
36. User select confirmation
37. System displays a display with all the booking info and a receipt number
38. User selects to print out a copy of the receipt

39. Post Conditions: System returns to booking window

4.4.7 Make invoice

1. Brief Description: After making a booking the system will make invoices.

2. Preconditions: The user must have made a booking
3. Business Trigger: User finish the booking and hits ok
4. Basic Flow:
5. User has finished the booking and hits ok
6. System saves the data to the database then:
7. System prints out a normal receipt with all legal info needed
8. System prints out a sticker for the accommodation
9. System prints out displays a window with selection option of paper ticket or RFID card
10. User selects Paper ticket
11. System prints out a ticket to show at the entrance
12. User selects finish
13. Post Conditions: System returns to booking window
14. Alternate flow: User selects RFID card
15. When in 9
16. User select RFID card then:
17. System displays a message about putting the RFID card to the reader
18. User puts the RFID card to the reader.
19. System reads the RFID cards number and stores it in the database with reference to the registration number of the car.
20. System displays a message about removal of the RFID card
21. User removes the RFID card from the reader.
22. Post Conditions: System returns to booking window.

4.4.8 Make receipts
Included in 4.4.7. Make invoice

4.4.9 Make RFID cards
Included in 4.4.7. Make invoice

5 System design
In any system design the Human computer interaction is a vital part of the process. When I started to make my solution of the system I started by making a task analyses of my proposed system. From that task list I draw up a storyboard for the interface. I tried to be consistent to the way things are done at the campsite today and have the interface as similar to the handwritten receipts that are used today. Then I tried to take in consideration things that the staff and manger had given as expectation to this solution. Those things include having a possibility to have different nationalities and/or different length of staying time at the campsite. I also used my own experience from working at the campsite and from my studies. When deciding on how to design the system I draw up a block diagram of how things would work together. To be minimizing the risk of disturbance from computer failure I decided to build the system so it could work even though computers would crash or lose network connections. This is achieved by having a copy of the database not only on the server but also on each cash desk and syncing them to the server by schedule. The cash computers could also ask for synchronisation if they needed an update on sales through internet or other cash computers.

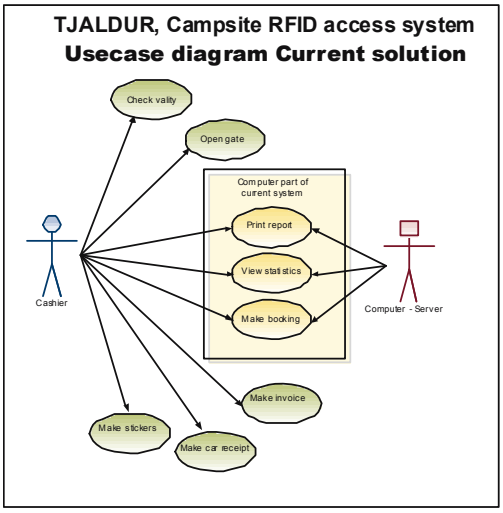


Figure 4 - Use case diagram– current solution

5.1 Interface

Designing the interface I used the current procedure of sales as a guide, so that the change for the staff would be as little as possible. But still would make things faster and more secure. Taking the handwritten receipts (See Appendix C) and the part that is computerized today I made a use case diagram of these procedures (see figure) When comparing the current use case diagram with the new version it can be seen that I am in fact making the current version adapt the old procedures so that they can all be done in the system but not manually by hand as it is today. This means that the cashier doesn't have to do as many things and the system is taking care of the validation, opening the gate, making the receipts and other materials. Even though the RFID part would not be implemented or would break down the system would still save a lot off manual work done by the staff. In the new version there opens a possibility for making internet sales although that part would be better to use of the campsite would be divided up to smaller areas so that certain spots could be recovered for a customer. This is really hard to do given the current organisation of the campsite.

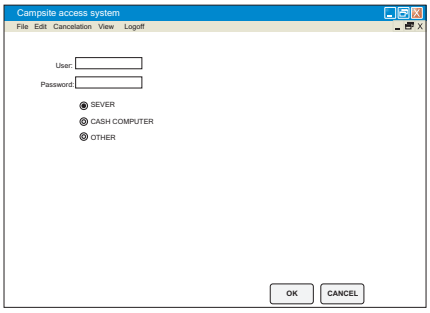
5.1.1 Tasks analyses

After having made the use case diagram of the proposed system I made a task analysis of the entire tasks that I thought that would be needed for the system. There I used the three different kinds of computers that would be used and made a list of all tasks that should be in them. Focusing on the server and the cash computer, but also thinking of how to add more items into the system later on. The complete task list can be seen in Appendix D

5.1.2 Storyboards

From the task list and use case diagram I made a storyboard of how the interface would work. I made a storyboard for the server setup part and the sales in the cash desk. Storyboards for other parts of the interface could be done later provided that there will be time to do so. To begin with I will focus on making a operational version of the cash desk and server.

5.1.2.1 Login



Here the user can put in the user name and password, then select what kind of interface he would like to get, server, cash desk or other type. The default user and login name will be admin and the default password for that user the same(admin). After this the user can choose OK button which will activate the top menu. In the top menu are normal controls, File, edit and help, plus setup and view. When the user selects setup he will get a drop down menu with all possible setup features.

Figure 5 – Login window

5.1.2.2 Setup company info

Here the user can enter all the information about the company using the system (Name, address, city, postal code, Id number, vat number and receipt text). This information will be used on all material that will be printed out like receipts, reports and views. The user can also add the companies logo by hitting on browse. Then a selection window will be displayed where the user can look for the logo image an import it. Hitting the OK button brings the user back to an empty window only with the top menu saving the info to the database. If cancel is chosen the same thing happens except that nothing is stored in the database. From there the user can select the next item to setup, view or logout.

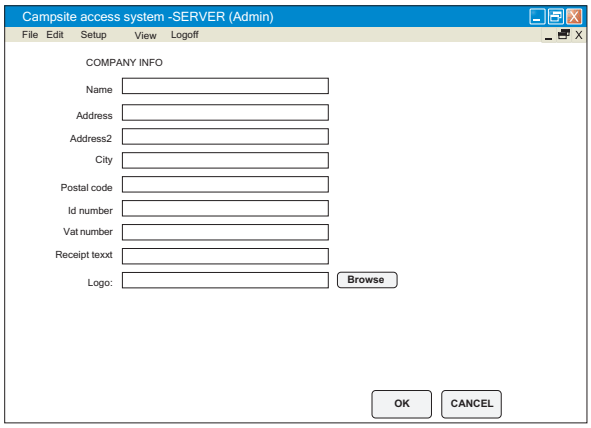


Figure 6 - Setup company info

5.1.2.3 Setup prices

If the user selects the prices from the setup in the top menu he will get this window. Here he can enter prices of the units that have fixed price. There is also a possibility to add items to the list. Choosing add item would give you a fill in window with item name and Id number for Item. On this page you can also setup the discounts for the system. In the same way you can add items to discounts with discount name and id number for that item. When you have finished you press OK button or CANCEL button. Both will bring you back to selecting from the top menu, the only difference is that by pressing OK the inputted data is saved to the database.

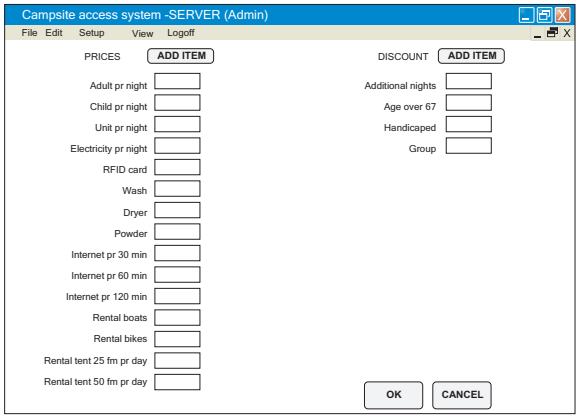


Figure 7 - Setup prices

5.1.2.4 Setup nationality

If the user selects the nationalities from the setup in the top menu, he can add nationalities to the nations list. He enters a shortcut number and the nation and presses add. Pressing the cancel button beside add clears both number and the nations input. If he has pressed add the nation will be shown in the list to the right. By highlighting nations in the list and pressing the single arrow by the bottom of the list he removes the highlighted nations from the list.

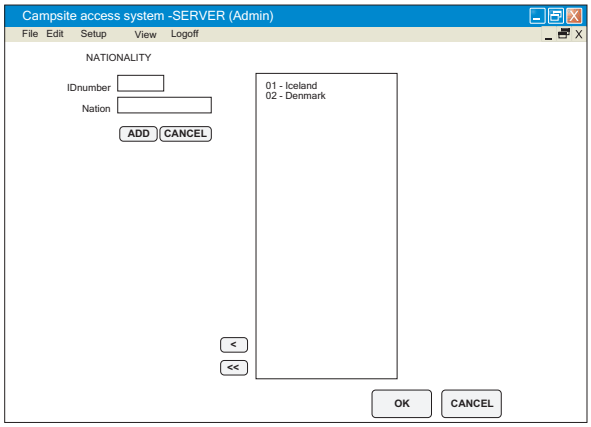


Figure 8 – Setup nationalities

By pressing the double arrow button all nations are clear from the list. When you have finished you press OK button or CANCEL button. Both will bring you back to selecting from the top menu, the only difference is that by pressing OK the inputted data is saved to the database.

5.1.2.5 Setup users

If the user selects the selects setup users from the top menu he will get this window. Here he can add users. A default user and password will be added during install (admin/admin). The users are given a number, name and password. Then by pressing add a new user is added to the users list in the middle. By pressing Cancel the textboxes are cleared. When the user is in the list it is possible to decide what right he gets, by pressing on the listed user types to the right. Then they will become highlighted with red color. To remove the rights you simply press again on the user type. It is also possible to create new user types by pressing add item. By highlighting user in the list and pressing the single arrow by the bottom of the list he removes the highlighted user from the list. By pressing the double arrow button all users are clear from the list(only default remaining). When you have finished you press OK button or CANCEL button. Both will bring you back to selecting from the top menu, the only difference is that by pressing OK the inputted data is saved to the database.

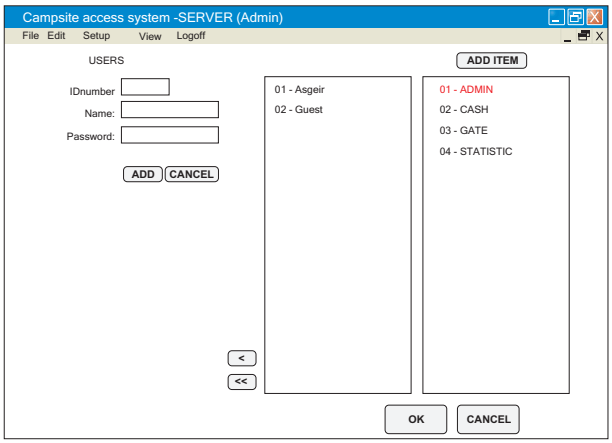


Figure 9 – Setup users

5.1.2.6 Setup map of site

If the user selects setup map from the top menu he will get this window where he can add a jpg file with a picture of the campsite. At this moment this is just for display in the system but later it could be added with the area divided down in areas that customers could reserve. Here the user presses browse and gets a file finder window, after finding the picture and selecting it, the picture is displayed in the window. To change pictures the user simple presses browse again. When you have finished you press OK button or CANCEL button. Both will bring you back to selecting from the top menu, the only difference is that by pressing OK the inputted data is saved to the database. After this the setup is completes and the user can either select view or logout from the top menu. If he selects view he will get a window with a possibility to select what to view and the period (This will be done if there is time). If he selects logout he will get the screen from 5.1.2.1. Login.

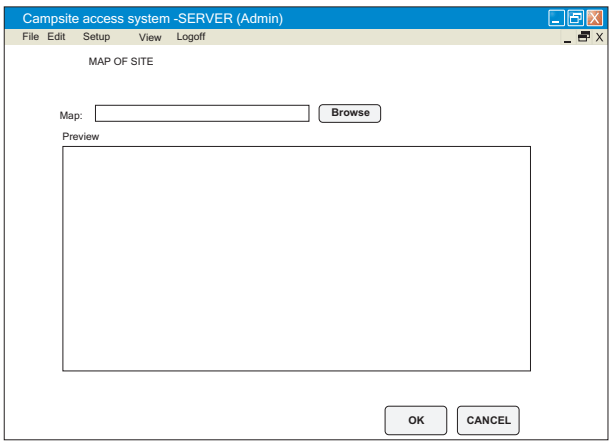


Figure 10 - Setup map of site

5.1.2.7 Sales window

If the user logs into a cash computer he will get a sales window like in figure 11. There he will get a top menu with file, edit, cancelations, view, and logoff. From file he can choose quit and logoff, from edit he can choose undo/redo, cut, copy, paste and find. In cancelation he can cancel or refund bookings. In view he can choose either to get statistical information about the sales or to close the day and send the updates to the server. Finally the user can choose logout and will be brought back to screen 5.1.2.1 Login. In the sales window the user will first see the current date, time and invoice number which are all automatically filed in by the system. Next is the fist line that will be handling most of the cash computers transaction. In this line there is the main information about the sale and about 90 % of the time this is the only line needed to fill in for the sales. Arrival date will be filled automatically by the system with current date. The user can changes that if the customer has arrived earlier or is making a booking in advance. Then number of nights are selected, number of adults and children. Nest the nationality is selected (can be done by using shortcut number) finally discount is given if that is necessary. When these information's are ready the user can press a plus or minus sign behind the line. Pressing the minus sign will clear the first line and you can start over. Pressing the plus sign new lines will be created by the system, one for each person. There it is possible to alter the info in each line if there is an error and that also gives the opportunity to have different length of staying time, different discount rates for each person, and different nationalities. By pressing the "i" button behind each line more info about that person can be added. The minus sign will erase that line. Under all the lines with persons a total amount for the camp fee will be displayed. Next we have electricity, there the arrival date is also the current date by default but can be changed. The number of nights will be taken from the first line but can also be changed. Then a info about where the plug in is and what number the hook is. From these information's the system calculates the price. Behind the line is also a plus and a minus sign to add lines or clear the line. Below electricity a info about the travel arrangement can be added, by choosing travel by car, bike, foot. Then the user selects what kind of accommodation the customer uses, a tent, a camper, a trailer, or a hut. The last thing in this line is to register the car registration number. This has to be used if there is a RFID card connected to the customer. In the bottom part of the window there are alternative things from the campsites service. Though those things might not be implemented in the RFID part of this solution because of time issues, they can still be in the interface so that it can be used as a cash register and takes care of all sold items. These items are laundry (washer, drier, powder), internet, rentals and other. All these items could be registered to a certain customer if needed. That might be especially good for the laundry part. In the last line the user can set the payment method to cash, card or invoice. There is also info about the amount going to the VAT and the grand total of this invoice amount. Choosing OK will finish the transaction and save it to the database, then the receipts will be printed and the RFID card coded if that is used. After that a new sales window will come up with the next invoice number. Pressing Cancel will clear all info from the sales window and the user can start from the beginning.

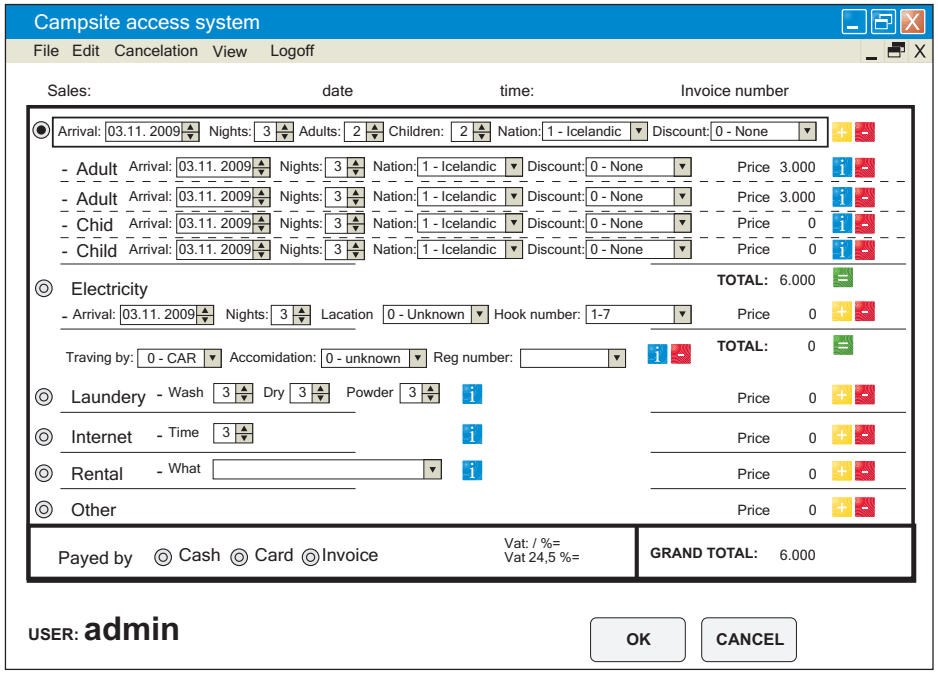
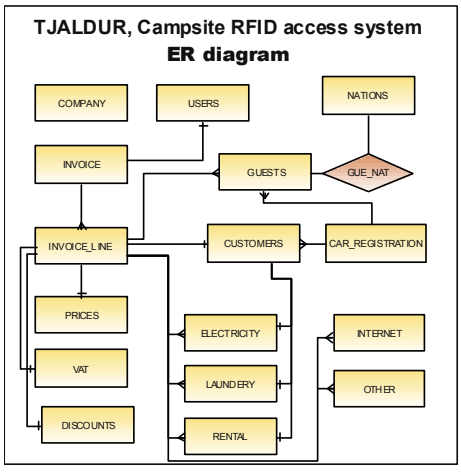


Figure 11 – Sales window

5.2 Database

When designing the database I used an ER diagram to draw up the fist draft having all attributes in the diagram. Then I changed it and made a ER diagram with only the tables and there connections. Then I made a relational database schema with all the attributes and how the all connect together indentifying primary and foreign keys. As I have said before I decided to have the database on both the server and the cash computers so that they could work independently. I will use OCDB to connect them together and make synchronization schedules to have everything in sync. The interface will also have a update buttons to make an unscheduled syncs if needed.

5.2.1 ER diagram of database



In my first draft I had all the attributes included in the ER diagram just to realize better the whole extent of the database. That also helped me to make the Relational database schema. I do not include this first draft here because every thing from there is in the ER diagram and the Relational schema. Making the ER diagram again I just show the tables, their connections and showing if they are connecting one tone ore one to many. It can be clearly seen that the INVOICE_LINE table will have a lot of connections and will be the main data table. The invoice table will gather up the info about the related lines. It can also be seen that Customers table has a lot of connections, but they might not be filled in because filling in the customer info is an option that is not mandatory in the system.

Figure 12 - ER diagram

5.2.2 Relational database schema

A relational database schema is used to measure the quality of the design of the database. In chapter 10 in the book Fundamentals of Database systems 5th editions by Elmari and Naviathe from 2007 they discuss the usage of Relational database schema and state that there are four informal design guidelines for relational schemas used to measure the quality of it. They are: 1. semantics of the attribute, 2. reducing the redundant information tuples, 3. reducing the NULL values in tuples and 4. disallowing the possibility of generating spurious tuples. Then they go on and talk about how to work with functional dependencies and the Normal forms of the database. Having these things in mind I draw up a schema for the database where all the tables are represented, their primary keys are indicated by underlining the attribute and then the foreign keys are indicated by a small black box by each attribute, colored lines are used to show the connections between the tables. From this it can be seen that the biggest table will be the IVN_LIN or the lines in the invoices, linking together all to different items in the sales procedure.

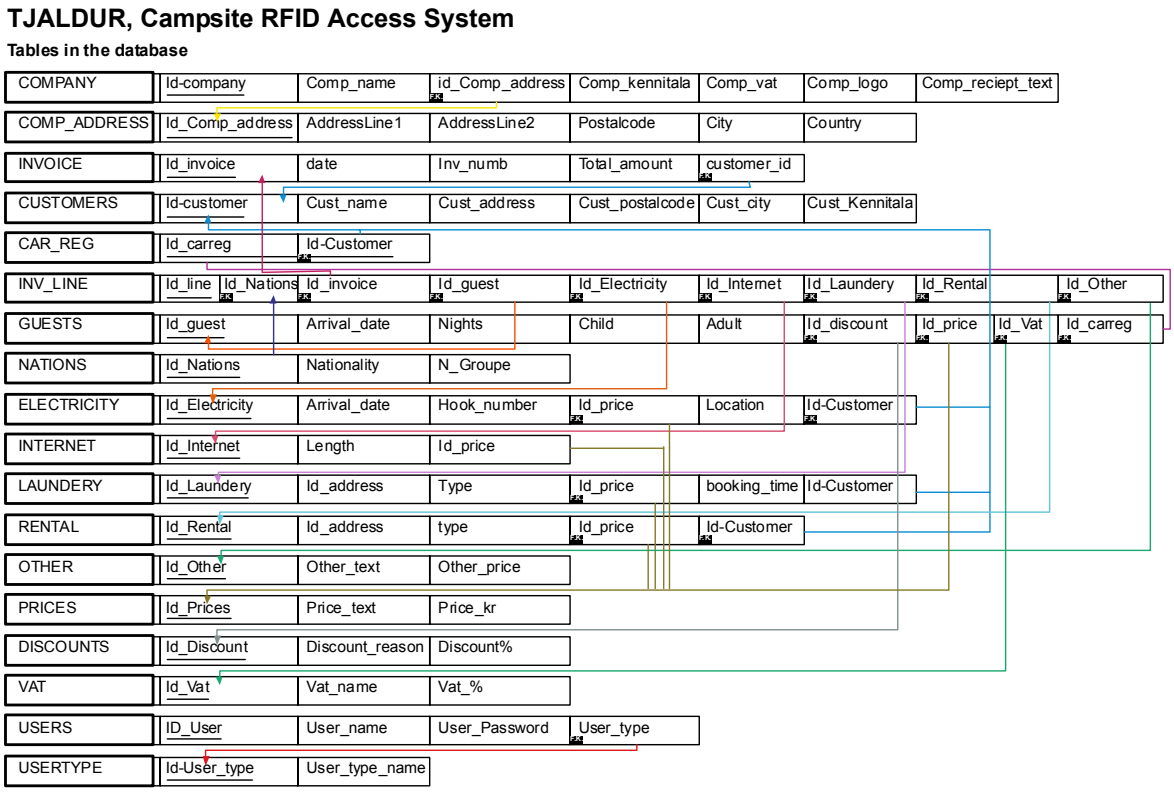


Figure 13 - Relational database schema

6 Implementation

I started working on the implementation in the beginning of the year. My first action was to start reading up about C# and SQL. I used two book to help me with this work C# in easy steps by Tim Andersson and Sams Teach Yourself C# in 21 days by Bradley L. Jones. I soon found out that I gained more from looking at videos and posts on the internet. I used mostly three sites on the internet, one is the www.msdn.microsoft.com (http://msdn.microsoft.com/en-us/vcsharp/bb798022.aspx) an other is the www.c-sharpcorner.com(http://www.c-sharpcorner.com/Articles/ArticleListing.aspx?SectionID=1&SubSectionID=143) and the third is the www.learnvisualstudio.net (http://www.learnvisualstudio.net/) which I had to buy an access to. These websites have given me most off the knowledge that I have needed for this project. During the implementation I encountered couple of problems, some off whom I have solved but others still stop me from finishing the first part of the project. In the next chapter I will talk about the technology I used, the problems that I have encountered and how I have solved them.

6.1 Technology platform

6.2 Hardware

For this project I use my laptop computer (Dell Latitude D830) running Windows XP service pack 3. It has a Intel (R) Corel(TM)2 Duo CPU T7700 @ 2.4 GHz and a 2,39GHz 2.00 GB of RAM.

To implement the RFID part I used a reader/writer from APSX called RW-210 using a C-100 module to connect the RW-210 to a computer through RS232 (serial port). The APSX reader/writer supports the ISO15693 RFID tags with an operating frequency of 13.56MHz.

6.3 Software

I choose to write the program in C# because I wanted to add to my experience a knowledge about C# and .net programming. I used Visual Studio 2008, working with ADO.net Entity Framework. Through the Entity Framework I connected my classes to MS SQL2005 Express edition database. Other software and tools used for this project are described in chapter 1.3.2 (Tools).

6.4 Problems encountered and solutions

6.5 Hardware

The only problem I ran into regarding the hardware was that it took a very long time to get the reader from the United States and it arrived here in Iceland in middle of March. This did not give me a lot of time to implement the reader into the system but since I could get good support and a DLL file to import into my project I managed to get the reader working in two days.

Because I did not manage to go further with the project I did not look at the implementation of the digital usb connection card intended to use for the gate opening.

6.6 Software

6.6.1 SQL

I used the visual studio to make the tables in the database. This was an easy thing to do but I later found out that my original design of the database was not sufficient. I had to change name of entities re-think connections and make new tables to satisfy my needs. When I had made the database and was staring to test the connections between the database and my software I found a thing called ADO.NET Entity Framework which looked and sounded very tempting to use. I decided to do that and by this way the database is mapped to an Entity set (classes) of the software using a data access layer. Having this all linked up in the visual studio saved me a lot of time in writing code to do all the connections. On the other hand I sometimes encountered problems that I had hard time finding solutions to and from what I could see from reading blogs on forum's some of those issues have no clear solution yet. The biggest problem I had was to get connections between tables using foreign key references. On figure is the diagram of the database as it looks today. The main difference from the original design is that I have skipped the invoice line table and I am connection everything directly to the invoice number. Then I have lost the connections on a couple of tables that comes from realising that some of the tables where just intended to give info to the forms such as prices and vat. Instead I needed to add field for things (price and vat) into other tables. Further development needs to take place as the code is developed.

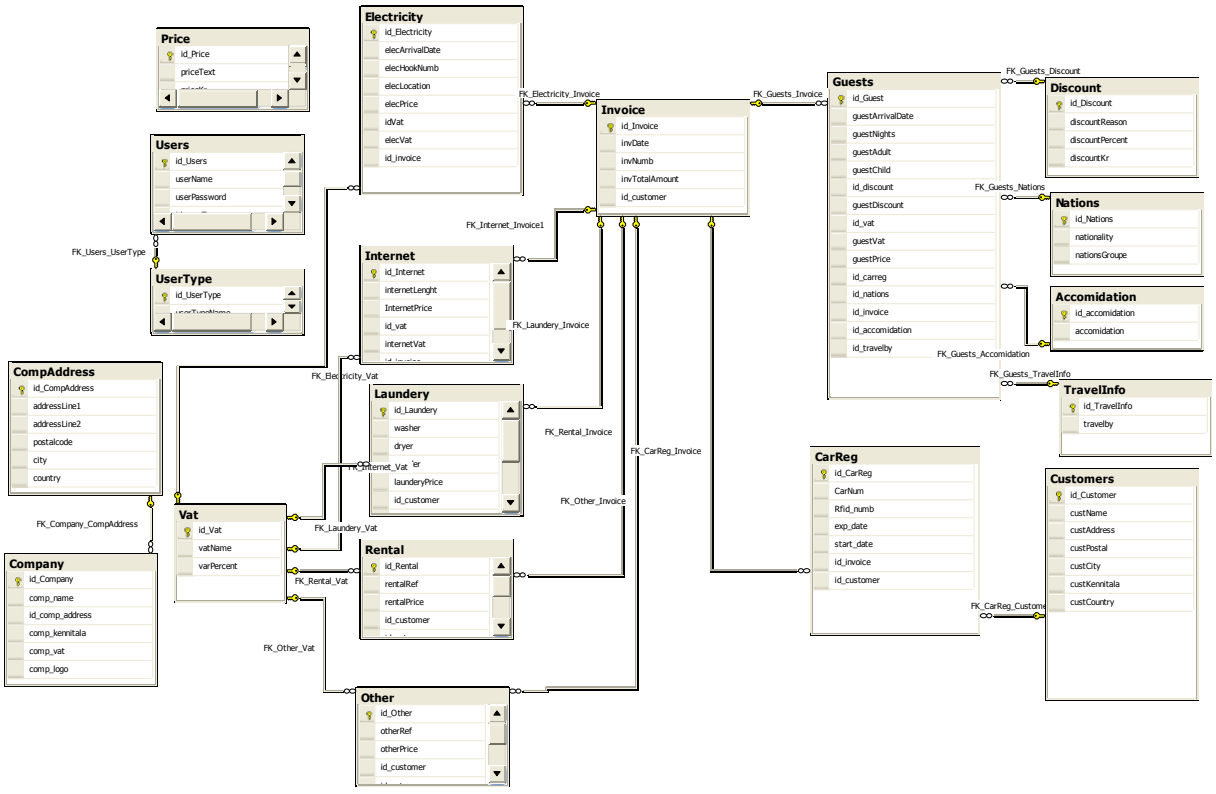


Figure 14 – Database for Tjaldur

6.6.2 C# - Visual Studio

Getting the hang of C# is not so hard when you have studied another programming language like Jva or C. But there are though some differences which at time are time-consuming to find the correct

statements to use. Visual offers good support and references to websites with loads of information for those how have the resources. I quickly found out that using the help search in visual often gave me an answer. As I have said before I also used a lot of videos and books from the web.

6.7 My software

The implementation has taken more time then I had realised in the beginning and therefore I have only managed to work on one of nine use cases. This is the main sales window. Other things like the setup and configuration part, the gate monitor and other wait further developments. The original design off the interface is pretty closed to my product as it is today. Most features are there although slide changes have been made to the interface as can be seen on figure 15 The firt thig is the data grid that I use to store each guest line, and the fact that I have not included a button by each row to set the persons data into, but only after you have selected further info button. In general I have cut down the number of buttons on the form to make it simpler. I have also added a departure window after a comment on that might be a good option to have.

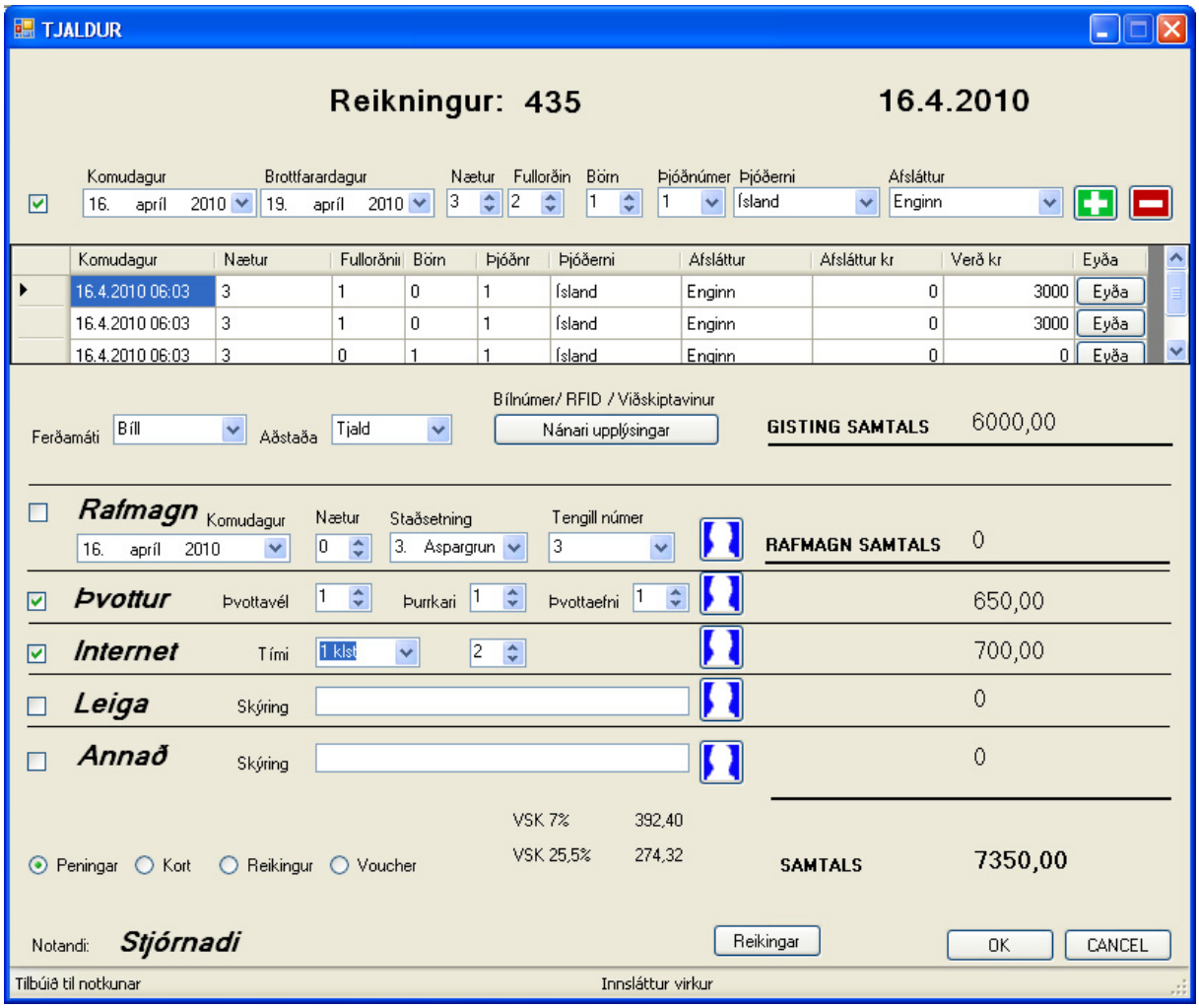


Figure 15 – the sales window interface

One of the elements that I forgot to put in the database in the beginning was the registration of the data carries. I had put in a place for a car registration information but forgot the RFID id number. This I added to the car registration table that is working fine at the moment.

When further info button is pressed a new window opens with the car registration and RFID read in. This form I had not designed before so that was design on the fly. In this window the dates are automatically filled with info form the main form about arrival date and then the number of nights are used to calculate the expiration date, giving it a fixed time of the day value. There you can add info about the car number and a customer. This is so all connected to the number that can be added on from the data carrier, giving the transaction an Id number of the card. On figure 16 we can see the car registration window.

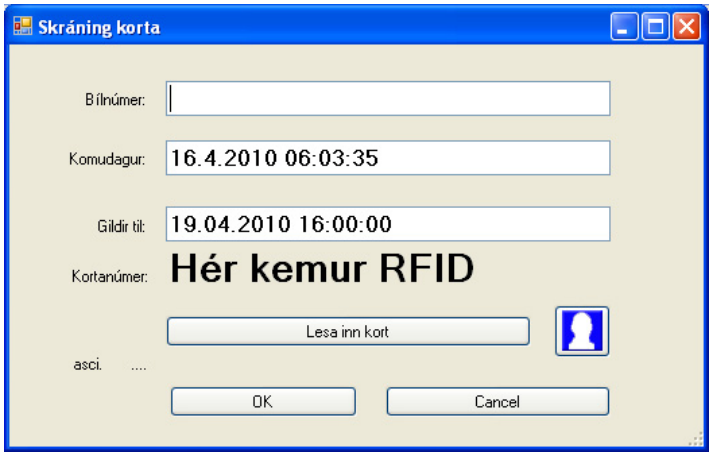


Figure 16 – car registration window

On this car registration window there is a person sign indication that further info can be added to the car registration form. This is the customer window, which I did not design either in the beginning. On figure 17 we see how that window looks like. There you can search for a customer by his name or kennitala in the database or enter a new one.

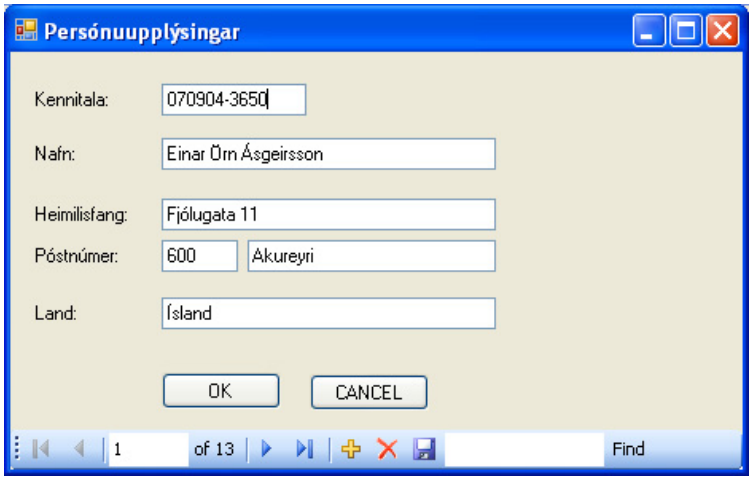


Figure 17 – customer window

The laundry, internet and electricity are added in the main window but to add a rental or other you get a popup window. After talking to the staff at the campsite it would properly make sense to add a combo box with the items known to be rented. This would give better opportunities to statistical information about the rentals. See figure 18. The other window is at the moment exactly the same as this one.

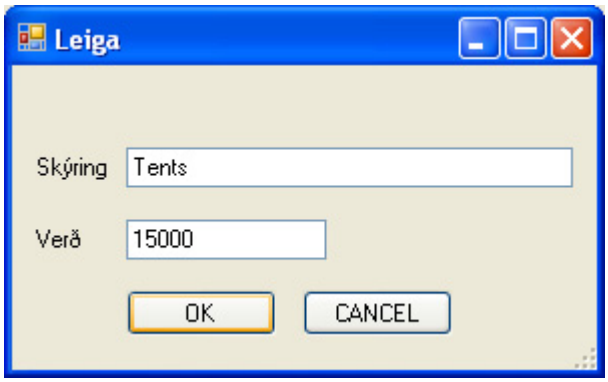


Figure 18 - rental window

I have started looking at reporting from the system to make receipts and such things but that is just on the beginning stage so I won't talk about that now.

I decided when I started the implementation of the code that I would try to be consistent in using English as the names of all variables, labels and buttons although the text shown in all window would be in Icelandic. I also tried to name all things with suitable names that would be more or less self explainable. The near the end I used try and catch to control the error exceptions. All in all the project is doing most things needed at this stages but it needs to be further worked on and developed to be a suitable solution for the campsite.

7 Evaluation

The evaluation is an important part of the design process. I started to evaluate the design in cooperation with my instructor. This has been done on our scheduled meeting where I have gotten her advice and comments on my original design. On the 15th of February I had a meeting with the camp manager and went over the implementations of what I had done by that point. This was not a formal evaluation session but an opportunity for me to show him how the design would look and to get feedback on it. In general he was happy about the design and no special issues needed to be addressed at that point. While implementing the code I used the debug features of the Visual studio to help me do use case testing on the code. A thing the keep in mind about the evaluation is that I only managed to work on one of nine of the use cases for the program. In this chapter I will talk about how further testing where done on the code. My aim was also to do cognitive walkthrough of the interface but I never found the time to do it. Some errors might have been caught during that evaluation and I see now that I should have listen better to my instructor to start that process much sooner.

7.1 Use case testing

As mentioned here above I did most of the use case testing on the code while implementing it, the use case I started working on was for the main sales window and is described in chapter 4.4.6 Make booking on page 20. Doing these testing I have found a couple of vulnerabilities that I have in some cases solved but in others actions need to be taken in case of further work with this program. The things that I found are:

- Number of nights at first the number of night had a min value of 0 so a line could be created with no nights behind it. This would not matter for the calculations but makes no sense to allow so I changed the min value to 1. The max value is set to 100. It is not possible to input numbers under or over those values. While testing the max number I found that display of guest total had some limitation so big numbers so I put a format ("F2") to the string and then got the right displayed number with two decimal points.

- Nationality is displayed through the entity set but a value can be typed in. Adding a value though the form should not be possible only through the setup. I have not found the solution for this yet.

- Nation id number did not change if the number was put in by typing. This is not solved yet, but I found out that if I use the arrow down key (not enter) after typing the number it is updated to the right number.

- Adding and deleting with the plus and the minus lines from the data grid was throwing exceptions some times. This I manage to solve in the code so it does throw the exceptions but displays a message about the error instead of crashing the program. I used try and catch to do this and also found out that I needed to clear the data grid and the table used to fill in the data grid to solve it.

- Deleting a line in the data grid was not working correctly, at first I was not getting the button to work at all. But after manually making all the columns in the datagrid I was able to get this to work, using a listener for the button clicked to delete the selected line.

- In the rental and other forms I had a error if i put anything else than numbers in the postal code field. I solved this with try and catch, reporting the error of missing number and by using Causes Validation property in the Visual Studio. That prevents the user from putting in the wrong type of data. Used the same for the Kennitala field that only requires numbers.

- Twice I had some problems in connecting the RFID reader to the computer. That seemed to work again after a reboot of the computer.

In total the use case testing went well but, there where not a lot of erreosr that came up and I have managed to solve most of them.

7.2 Heuristic Evaluation

I decided to get experienced programmers to do an evaluation of my software. I aimed on getting 2-3 persons to do this but in the end I only got response from one of them. I an evaluation session I monitored this person using the software and noted down his comments on it. The nodes form the session can be found in appendix III. Before the session started I gave the evaluator a sheet with list of 10 heuristics to look fore (see appendix IV. From this session I got several good points about the program and software in general. From the evaluator experience in software development two things ar very important and those are tab order and shortcut keys. I have improved the tab order much since the session put shortcut keys wait further developments. In total where there 21 points that I did node down and here is the list of them and how or if I have solved them

1. Adult should be default 1, so if you press plus button it gives at least one line.*(Fixed)*
2. What about other items sold as Showers.*(A thing that need to be decided by the users)*
3. Discount does not work for 1 night .*(Fixed)*
4. Tab order is not correct – very important .*(is improved but not totally fixed)*
5. No shortcut keys – always wanted.*(Waits for further development)*
6. Washer should have the most used one as default at least 1 wash, maybe 1 of each.*(Fixed)*
7. Put error handling on rental and other popup screens.*(Fixed)*
8. The user login should display notandi to keep up with other language settings.*(Fixed)*
9. When using number input and enter the country is not updated.*(have not solved but found out that the arrow down key works instead of enter)*.
10. Discount rules for additional nights discount so if more than 2 nights then rule would activate discount. .*(Fixed, but found out later that the manager of the campsite did not want to have the discount added automatically so I removed it again)*
11. Nationality allows for input of new names, should not be possible*(not fixed yet, I have not found out how to make the combo box read only)*
12. Text in Discount is not all visible when dropdown menu opens *(this is because of the column in the database is 20 char and I fixed this by enlarging the combo box but future solution is to changes the database)*
13. Delete line in data grid button does not work with using Enter when selected *(fixed by adding the column manually to the data grid and putting an event handler on a column button)*
14. If a line in data grid is deleted the info is not updated (price)*(fixed)*
15. In the report the sum is for all data in the list not only current page*(fixed by removing the line from the report or now but the development of reports waits further development)*
16. Current customer should be displayed as default inside the same invoice *(not fixed yet differs from my first ideas and has to be decided how to solve, waits further development)*
17. The OK button does not save the new customer has to hit save button to store new customer.*(Fixed)*
18. Searching is dependent on – in Kennitala*(not fixed yet waits further development)*
19. Prices are not align to right*(not fixed yet waits further development)*
20. Possible to have both adult and children in same data line in data grid. *(not fixed yet waits further development)*
21. No default value on radio button at the bottom*(Fixed)*

7.3 Cooperative Evaluation

I got two staff member from the campsite to attend a supervised session where i monitored the usage of the software on a extra display and noted down the problems that come up and discussed the possibilities to solve them with these staff members. They have both good experiences in collection fees for the campsite, but don't have the same knowledge about computers. One is very used to working on a computer and the other has less experience. From these sessions I got about 40 points to look at, the nodes from these sessions can be found in appendix III:

1. Format of day and time in data grid He wanted to get rid of the time so changing the day would be easier*(not fixed yet, waits further work)*
2. Discount options needed to be more or even a possibility to changes to percents of the amount*(not fixed yet ,has to decided how should work, waits further work)*
3. Deleting a line in the data grid through exceptions *(Fixed)*
4. Extra night discount should not be default for more than two nights*(Fixed)*
5. Possibility of making a fee for children some times*(not fixed yet ,has to decided how should work, waits further work)*

6. Car number is it necessary will the staff member ever be able to see it*(not fixed yet ,has to decided how should work could use cameras to provide the viability inside the office, waits further work)*
7. Should car number have a special format*(After discussing this we agreed on not having any required format because cars can have a lot of different numbers)*
8. How to register a company as the customer?*(this can be solved by using the customer info box as it is, but the only thing is that it is properly better to have only the option of one customer pr invoice, this was something that came up in the heuristic evaluation also)*
9. OK in tool strip panel was confusing – evaluator tried to push it *(the tool menu strip had a ok text implying that the system was ready, one user was trying to use the display as a control. Changed the text of this display to try to clearly this.*
10. Customers information buttons where not clearly marked by the image on the button*(Not fixed yet)*
11. Add a combo box to the Rental window with the know elements that are being rented, such as bouncer, boats, tents, tables and chairs, the houses and cottages, barbeques. *(Not fixed yet, and should properly be done in the database and then use an entity connection, waits further work)*
12. Internet should also have a 24 hours options for those with their own computers *(Not fixed yet, but should be easy)*
13. Other should also be able to connect to a customer*(fixed)*
14. If laundry option is selected it does not add to the price only the value of one*(fixed)*
15. Default money should be on the payment radio buttons *(fixed)*
16. Cards should maybe be a combo box with all kind of cards or at least debit/ credit *(Not fixed yet, and should properly be done in the database and then use an entity connection, waits further work)*
17. How to make credit invoices??? *(Not fixed yet, and has to be also implemented in the database and then use an entity connection, waits further work)*
18. Departure date should also be displayed *(Fixed)*
19. RFID reader did not work when tested*(Started working after a restart of the computer)*
20. Date in Electric arrival date should be automatically change with arrival date*(Fixed)*
21. Tab order on main form is incorrect*(Has been fixed to some extent but need further work)*
22. Unclear how to search in customer form *(not been fixed yet)*
23. Postal code does not bring up known cities*(not been fixed yet, should get a costal code)*
24. Save of customer info does not work *(Fixed)*
25. Internet time was unclear value mixed with time length*(Fixed by moving the value to select behind the text to select)*
26. Where should additional, often used things be put like boats, bicycle-cars.*(see point 11 above)*
27. How to fix if errors are done??? *(see point 17 above)*
28. Error while pushing + after – buttons *(Fixed)*
29. When resizing the form, parts of the form did look right(*Fixed by having the form non resizable)*
30. Part of the Vat amount is not displayed*(Fixed)*
31. How to put reference text to a certain invoice like reason for how it is done *(This is can only be done though customers at the moment but could be added if needed, waits further work)*
32. How to but reason for voucher or more info about the voucher*(This is can be done though customers)*
33. Discount is always based up on nights and always uses full price for first night*(Fixed)*

34. When deleting a line in data grid by using the delete button two lines sometimes go together*(Fixed)*
35. No reason for having a combo box for the amount of discount in first line*(Fixed)*
36. Changes on the combo boxes using the keyboard do not work *(See point 9 in heuristic evaluation)*
37. Inner search in combo boxes does not work*(Works to type the beginning of the word and then use arrow down, like in point 9 in heuristic evaluation)*
38. Location and number for electricity does give any possible values*(Fixed with collections but might be better to add a table in the database and do entity connection to it)*
39. Possibility to order time for laundry *(Not done yet but would be a cool feature to have, waits further work)*
40. The checkbox for guest does not delete guest in a form if unselected*(Fixed)*

From this it can be seen that a lot of point have come up, some have been solved other wait for future work to be done on this project. In the conclusions chapter I will talk more about future work.

7.4 Metrics Evaluation

I used a software called SourceMonitor, version 2.5.1.0 from Campwood Software (www.campwoodsw.com) which is a tool that measures and records source code metrics. From this software I got a table of the comments, methods, complexity, block dept and amount of lines in each class of the project. The following table shows the statistics from this software.

File Name	Lines	Statements	Percent Comment Lines	Methods per Class	Max Complexity	Calls per Method	Statements per Method
Tjaldur04\carregform.cs	141	76	28,4	16	7	1,31	2,25
Tjaldur04\customerForm.cs	145	65	25,5	10	4	1,8	3,9
Tjaldur04\Form1.cs	1056	487	24,9	53	10	3,34	7,58
Tjaldur04\otherform.cs	80	42	22,5	8	3	1,13	2,13
Tjaldur04\pFrm.cs	171	72	5,3	6	9	2,33	8,67
Tjaldur04\Program.cs	21	11	0	1	1	4	3
Tjaldur04\rentalform.cs	83	42	20,5	8	3	1,13	2,13
Tjaldur04\report2.cs	30	18	3,3	2	1	1,5	1,5
Tjaldur04\startUp.cs	26	15	0	2	1	0,5	1
Total	**1753**	**828**	**22%**	**11,77**	**10**	**2,42**	**5,39**

Table 7.4 – Metrics for the classes

As seen In table the lines of code are 1753, having 828 statements and 22% comment intensity. Calls pr method have an average value of 2,42 and statements per method have an average of 5,39 and the maximum complexity is 10.

8 Conclusions

This project has been a very interesting and enlightening journey, I have learned allot about developing software solutions and how much work and effort one must put in to such projects. I took me a while time to getting the hang of C# and especially the usage of the ADO.net Entities Framework. I should have read more about the ADO.net because I had a lot of problems relating to connections of primary and foreign keys in the Framework. Although the reader I ordered arrived very late, the fact that the supplier provides both support and DLL for the hardware it proved to be a good choice since it took only 2 days to get it working with the program. I hope I will have the time and resources to finish this project sooner than later. Of course there are a many things I would do differently but that is one of the aim of a project like this.

8.1 Future work

8.1.1 Finish the implementation

First thing is to finish the work that I have already planned, there are 8 use cases waiting to be implemented and also still some things left of the first part. My next step in that work would be to start looking at the interface for the gate and to make the receipts and other printouts work. Then I would work on the interface for the server, setup and administration. And finally I would look at the internet solution, making the possibility of pre sales through the internet.

8.1.2 New features

There are a lot of possibilities regarding RFID systems and in the planned solution I am only using a tiny part at the moment. First thing would be to expand the system so it could handle all access to all doors and gate at the site. Next issue could be to use the fact that you can store information on the data carries for presold items, personal information or any other thing.

An important thing to look at also would be what is needed to do with such a system so it can be ran as a certified cash register, so that it could replace a normal cash register.

9 References

Anderson T, 2004, C# in easy steps, Warwickshire: Computer Steps.

Axess AG, 2009. *Knowhow.* [Online] (Updated 16 July 2009) [Online] Available at: http://www.teamaxess.com/en/xtra/knowhow/nh2.htm [Accessed 6 Desember 2005]

Bennett, S. McRobb, S. & Farmer, R., 2006. *Object-oriented systems analysis and design using UML.* 3rd ed. Berkshire: McGraw-Hill education.

Bhatt H., Glover B., 2006. *RFID Essentials.* 1st ed. Sebastopol (CA): O'Reilly

Campground manager, n.d. System Tour [Online] Available at: http://www.campgroundmanager.com/index.php/main/system_tour_more/customer_in_park_list/ [Accessed 18 September 2009]

Campground master, n.d. *Reservation Software for Campgrounds and RV Parks.* [Online] Available at: http://www.campgroundmaster.com/zip/campground_master_information.pdf [Accessed 18 September 2009]

Compusoft, n.d. *Reservation & Management for Campsites*. [Online] Available at: http://www.compusoft.dk/index.asp?mode=forside [Accessed 10 October 2009]

Elmasri, R & Navathe, S.B., 2007. *Fundaments of database systems*. 5th ed. Boston: Pearson education.

Falke, O. et al., 2007. Mobile services for near field communication. *Technical report.*[Online] Available at:http://media.csee.ltu.se/courses/smd162_Pervasive_Computing/papers/B5%20Falke%20-%20Mobile%20Services%20for%20Near%20Field%20Communication%20-%202007.pdf [Accessed 16. September 2009]

Finkenzeller, K., 1999. *RFID handbook: radio-frequency identifications, fundamentals and applications.* Translated from German by R.Waddington. Chichester: John Wiley & sons Ltd.

Jones L. B, 2002. SAMS: Teach Yourself C# in 21 days. Indiana: Sams Publishing

Mada, n.d. *MADA - competence & experience in RFID.* [Online] Available at: http://www.mada.de/uploads/media/Camping_and_leisure_A4.pdf [Accessed 10 October 2009]

Marx G., 1998. An Ethics For The New Surveillance. *The Information Society*. Vol. 14, No. 3. [Online] Available at: http://web.mit.edu/gtmarx/www/ncolin5.html [Accessed 4 December 2009]

NO Q Caravan, n.d. *NO Q Caravan & Campsite Reservation 1.0.1 specifications*. [Online] Available at: http://download.cnet.com/NO-Q-Caravan-Campsite-Reservation/3010-2064_4-10286708.html [Accessed 18 September 2009]

Regulation nr. 163 21, 2007. *Lög um Hagstofu Íslands og opinbera hagskýrslugerð,* [Online] Available at: http://www.althingi.is/altext/135/s/0446.html [Accessed 6 December 2009]

Regulation nr. 70, 2000. *Lög um persónuvernd og meðferð persónuupplýsinga Persónuvernd,* [Online] Available at: http://www.althingi.is/lagas/nuna/2000077.html [Accessed 4 December 2009]

Roberts, C.M., 2006. Radio frequency identification (RFID). *Computers & security,* 25, pp 18-26.

Violet, n.d. *Mir:ror the RFID reader that gives power to your Objects.* [Online], Available at: http://www.violet.net/_mirror-give-powers-to-your-objects.html [Accessed 28 September 2009]

Wikipedia contributors a, 'Access control', *Wikipedia, The Free Encyclopedia,* [Online], Available at: http://en.wikipedia.org/w/index.php?title=Access_control&oldid=329655898 [Accessed 6 December 2009]

Appendix A

Class Form1

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using TjaldurDAL;



namespace Tjaldur04
{
    public partial class Form1 : Form
    {
        public TjaldurEntities tje;

        int washTotal, dryerTotal, powderTotal, internetTime;//variables for the
calculation of prices
        int washCount, dryerCount, powderCount, elecCount;// variables for the
calculation of prices
        float guestTotal = 0;// variable for the total amount to display and use
for calulation
        int elecTotal = 0;// variable for the total amount to display and use for
calulation
        int launderyTotal = 0;// variable for the total amount to display and use
for calulation
        int internetTotal = 0;// variable for the total amount to display and use
for calulation
        int rentalTotal = 0;// variable for the total amount to display and use for
calulation
        int otherTotal = 0;// variable for the total amount to display and use for
calulation
        float grandTotal = 0; // variable for the total amount to display
        float vat255Total;// variable for the vat calculation
        float vat7Total;// variable for the vat calculation
        public string Tbx_rental_refIN;
        public rentalform popupFormLeiga = new rentalform();
        public otherform popupFormOther = new otherform();
        public carregform carForm = new carregform();
        public DataTable guestLinetable;
        public Invoice inv1;
        public Int64 _currentInv =0;

        private Int64 custNumbLaundery, custNumbRental, custNumbElec,
custNumbInternet, custNumbOther;

        public Form1()
        {
            InitializeComponent();
        }


        private void Form1_Load(object sender, EventArgs e)
        {
            Lbl_CurrentTime.Text = DateTime.Now.ToString("d");

           InitDataOnForm();
```

```
}

public void InitDataOnForm()
{
    // make an new instance of the Entity set
    tje = new TjaldurEntities();
    // create a new invoive
    inv1 = Invoice.CreateInvoice(1);
    // add the new invoice to the entity set
    tje.AddToInvoices(inv1);
    // save the new invoce to the entity set so I can use the ivoicenumber
    tje.SaveChanges();

    // making the connection to the Entity set for the nationality
    cob_guest_nationality.DisplayMember = "nationality";
    cob_guest_nationality.DataSource = tje.Nations;
    // making the connection to the Entity set for the accomidation
    Cob_facilities.DisplayMember = "accomidation1";
    Cob_facilities.DataSource = tje.Accomidations;
    // making the connection to the Entity set for the travel method
    Cob_travel_method.DisplayMember = "travelby";
    Cob_travel_method.DataSource = tje.TravelInfos;
    // making the connection to the Entity set for the nationality
    cob_guest_nationumber.DisplayMember = "id_Nations";
    cob_guest_nationumber.DataSource = tje.Nations;
    // making the connection to the Entity set for the discount
    Cob_guest_discount.DisplayMember = "discountReason";
    Cob_guest_discount.DataSource = tje.Discounts;
    // making the connection to the Entity set for the discount
    Cob_guest_discountKr.DisplayMember = "discountKr";
    Cob_guest_discountKr.DataSource = tje.Discounts;
    // making the connection to the Entity set for the invoices
    cob_inv_numb.DisplayMember = "id_Invoice";
    cob_inv_numb.DataSource = tje.Invoices;
    // making a temp variable to hold the invoce number
    Int64 tempInvNumb = Int64.Parse(inv1.id_Invoice.ToString());
    Lbl_InvNumb.Text = "" + tempInvNumb;
    // Make a new data table
    CreateDataTable();
    // make the cash radiobutton default
    Rbu_cash.Checked = true;
}
// to get and set the current invoice number to the other forms
public Int64 currentInv
{
    get { return _currentInv; }
    set { _currentInv = value; }
}


private void CreateDataTable()
{
    // Add a CellClick handler to handle clicks in the button column.
    dataGridView1.CellClick +=
    new DataGridViewCellEventHandler(dataGridView1_CellClick);
    // Here we create a DataTable with nine columns
    guestLinetable = new DataTable();
    //make the colunms for the data table
    guestLinetable.Columns.Add("guestArrivaldate", typeof(DateTime));
    guestLinetable.Columns.Add("guestNights", typeof(int));
    guestLinetable.Columns.Add("guestAdult", typeof(short));
    guestLinetable.Columns.Add("guestChild", typeof(short));
    guestLinetable.Columns.Add("id_nations", typeof(int));
    guestLinetable.Columns.Add("nationality", typeof(string));
    guestLinetable.Columns.Add("id_discount", typeof(string));
    guestLinetable.Columns.Add("discount", typeof(float));
    guestLinetable.Columns.Add("guestPrice", typeof(float));
    CreateColumns();
```

```
            dataGridView1.Columns["discount"].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleRight;
            dataGridView1.Columns["guestPrice"].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleRight;

            dataGridView1.DataSource = guestLinetable;
        }

      private void CreateColumns()
        {
          // Initialize the columns in the datagrid
            DataGridViewButtonColumn eydaTakkiColumn =
                new DataGridViewButtonColumn();
            DataGridViewTextBoxColumn guestArrivaldateColumn =
                new DataGridViewTextBoxColumn();
            DataGridViewTextBoxColumn guestNightsColumn =
                  new DataGridViewTextBoxColumn();
            DataGridViewTextBoxColumn guestAdultColumn =
                  new DataGridViewTextBoxColumn();
            DataGridViewTextBoxColumn guestChildColumn =
                  new DataGridViewTextBoxColumn();
            DataGridViewTextBoxColumn id_nationsColumn =
                  new DataGridViewTextBoxColumn();
            DataGridViewTextBoxColumn nationalityColumn =
                  new DataGridViewTextBoxColumn();
            DataGridViewTextBoxColumn id_discountColumn =
                   new DataGridViewTextBoxColumn();
            DataGridViewTextBoxColumn discountColumn =
                   new DataGridViewTextBoxColumn();
            DataGridViewTextBoxColumn guestPriceColumn =
                   new DataGridViewTextBoxColumn();
          // makeing the setting for each colunm
          // first colunm for arrivaldate
            guestArrivaldateColumn.Name = "guestArrivaldate";
            guestArrivaldateColumn.HeaderText = "Komudagur";
            guestArrivaldateColumn.DataPropertyName = "guestArrivaldate";
            // secound colunm for number of nights
            guestNightsColumn.Name = "guestNights";
            guestNightsColumn.HeaderText = "Nætur";
            guestNightsColumn.DataPropertyName = "guestNights";
            guestNightsColumn.Width = 80;
            // third colunm for adults
            guestAdultColumn.Name = "guestAdult";
            guestAdultColumn.HeaderText = "Fullorðnir";
            guestAdultColumn.DataPropertyName = "guestAdult";
            guestAdultColumn.Width = 50;
            // fourth colunm for children
            guestChildColumn.Name = "guestChild";
            guestChildColumn.HeaderText = "Börn";
            guestChildColumn.DataPropertyName = "guestChild";
            guestChildColumn.Width = 50;
            // fifth colunm for nations id number
            id_nationsColumn.Name = "id_nations";
            id_nationsColumn.HeaderText = "Þjóðnr";
            id_nationsColumn.DataPropertyName = "id_nations";
            id_nationsColumn.Width = 50;
            // sixth colunm for natioanlity
            nationalityColumn.Name = "nationality";
            nationalityColumn.HeaderText = "Þjóðerni";
            nationalityColumn.DataPropertyName = "nationality";
            nationalityColumn.Width = 90;
            // seventh colunm for discount reason
            id_discountColumn.Name = "id_discount";
            id_discountColumn.HeaderText = "Afsláttur";
            id_discountColumn.DataPropertyName = "id_discount";
            id_discountColumn.Width = 90;
            // eigth colunm for amount of discount
            discountColumn.Name = "discount";
```

```
            discountColumn.HeaderText = "Afsláttur kr";
            discountColumn.DataPropertyName = "discount";
            discountColumn.Width = 90;
            // ninth colunm for price
            guestPriceColumn.Name = "guestPrice";
            guestPriceColumn.HeaderText = "Verð kr";
            guestPriceColumn.DataPropertyName = "guestPrice";
            guestPriceColumn.Width = 90;
            // tenth colunm for button deleting rows
            eydaTakkiColumn.Name = "eydaTakki";
            eydaTakkiColumn.HeaderText = "Eyða";
            eydaTakkiColumn.Text = "Eyða";
            eydaTakkiColumn.Width = 50;

           // Use the Text property for the button text for all cells rather
            // than using each cell's value as the text for its own button.
            eydaTakkiColumn.UseColumnTextForButtonValue = true;

            // Add the columns to the control.
            // first colunm is the arrival date
            dataGridView1.Columns.Insert(0, guestArrivaldateColumn);
            // second colunm is the number of nights
            dataGridView1.Columns.Insert(1, guestNightsColumn);
            // third colunm is the number of nights
            dataGridView1.Columns.Insert(2, guestAdultColumn);
            // fourth colunm is the number of nights
            dataGridView1.Columns.Insert(3, guestChildColumn);
            // fifth colunm is the number of nights
            dataGridView1.Columns.Insert(4, id_nationsColumn);
            // sixth colunm is the number of nights
            dataGridView1.Columns.Insert(5, nationalityColumn);
            // seventh colunm is the number of nights
            dataGridView1.Columns.Insert(6, id_discountColumn);
            // eight colunm is the number of nights
            dataGridView1.Columns.Insert(7, discountColumn);
            // ninth colunm is the number of nights
            dataGridView1.Columns.Insert(8, guestPriceColumn);
            // tenth colunm for button deleting rows
            dataGridView1.Columns.Insert(9, eydaTakkiColumn);

        }

        // update the time for the form if click on
        private void CurrentTime_Lbl_Click(object sender, EventArgs e)
        {
            Lbl_CurrentTime.Text = DateTime.Now.ToString();
        }
        // a method to update the grand total and the vat calculations for each
invoice
        private void updated_GrandTotal()
        {
            grandTotal = guestTotal + internetTotal + launderyTotal + elecTotal +
rentalTotal + otherTotal;
            vat255Total = (float)((internetTotal + launderyTotal + elecTotal +
rentalTotal+ otherTotal) * 0.2032);
            Lbl_vat25.Text = vat255Total.ToString("F2");
            vat7Total = (float)((guestTotal) * 0.0654);
            Lbl_vat7.Text = vat7Total.ToString("F2");
            Lbl_grandtotal.Text = grandTotal.ToString("F2");

        }
        // a method to update the guesttotal after takeing the discount off
        private void update_guestTotal()
        {
            // here it goes through the datatable and adds to the guest total
            guestTotal = 0;
            foreach (DataRow dr in guestLinetable.Rows)
            {
```

```
                guestTotal += (float)(dr["guestPrice"]) - (float)(dr["discount"]);
            }
            // puts the guest total to the label
            Lbl_guests_price.Text = guestTotal.ToString("F2");
            updated_GrandTotal();
        }


        // a method to update the electicity total (has a fixed price that needs to
be connected to price table later)
        private void updated_ElecTotal()
        {
            elecCount = (int)Nub_elec_nights.Value;
            elecTotal = elecCount * 500;
            Lbl_elect_price.Text = elecTotal.ToString("F2");
        }
        // a method to update the internet total (has a fixed price that needs to
be connected to price table later)
        private void updated_InternetTotal()
        {
            internetTotal = internetTime * 350;
            Lbl_internet_price.Text = internetTotal.ToString("F2");
        }
        // a method to update the laundery total (has a fixed price that needs to
be connected to price table later)
        private void updated_LaunderyTotal()
        {
            launderyTotal = washTotal + dryerTotal + powderTotal;
            Lbl_laundery_price.Text = launderyTotal.ToString("F2");
        }
        // a method to update the rental total (has a fixed price that needs to be
connected to price table later)
        private void updated_RentalTotal()
        {
            Lbl_rental_price.Text = rentalTotal.ToString("F2");
        }
        // a method to update the other total (has a fixed price that needs to be
connected to price table later)
        private void updated_OtherTotal()
        {
            Lbl_other_price.Text = otherTotal.ToString("F2");
        }
        // Calulate the price of laundery if value changes in amount of washes
        // (has a fixed price that needs to be connected to price table later)
        private void Nub_washer_ValueChanged(object sender, EventArgs e)
        {
            washCount = (int)Nub_washer.Value;
            washTotal = washCount * 250;
            // checks if all values for laundery are zero or not
            if (washCount != 0 || dryerCount != 0 || powderCount != 0)
            {
                Chb_laundery.Checked = true;
                updated_LaunderyTotal();
            }
            else
            {
                Chb_laundery.Checked = false;
            }
            updated_LaunderyTotal();
        }
        // Calulate the price of laundery if value changes in amount of dryers
        // (has a fixed price that needs to be connected to price table later)
        private void Nub_dryer_ValueChanged(object sender, EventArgs e)
        {
            dryerCount = (int)Nub_dryer.Value;
            dryerTotal = dryerCount * 350;
            // checks if all values for laundery are zero or not
```

```
            if (washCount != 0 || dryerCount != 0 || powderCount != 0)
            {
                // set the laundery option to selected
                Chb_laundery.Checked = true;
            }
            else
            {
                // set the laundery option to unselected

                Chb_laundery.Checked = false;
            }
            updated_LaunderyTotal();
        }
        // Calulate the price of laundery if value changes in amount of powder
        // (has a fixed price that needs to be connected to price table later)
        private void Nub_powder_ValueChanged_1(object sender, EventArgs e)
        {
            powderCount = (int)Nub_powder.Value;
            powderTotal = powderCount * 50;
            // checks if all values for laundery are zero or not
            if (washCount != 0 || dryerCount != 0 || powderCount != 0)
            {
                // set the laundery option to selected
                Chb_laundery.Checked = true;
            }
            else
            {
                // set the laundery option to unselected
                Chb_laundery.Checked = false;
            }
            updated_LaunderyTotal();

        }
        // displays the customerform and listens for a ok button push to get the
control back
        private void Ibu_car_info_Click(object sender, EventArgs e)
        {
            // creates a new customerform for electricity
            customerForm custFormElec = new customerForm();
            // Listens for the push of ok button ín customer form to get back
control
            if (custFormElec.ShowDialog() == DialogResult.OK)
            {
            }
            // putting the id number  of a customer to
            custNumbElec = custFormElec.currentCust;
        }

        // displays the customerform and listens for a ok button push to get the
control back
        private void Ibu_internet_info_Click(object sender, EventArgs e)
        {
            // creates a new customerform for internet
            customerForm custFormInternet = new customerForm();
            // Listens for the push of ok button ín customer form to get back
control
            if (custFormInternet.ShowDialog() == DialogResult.OK)
            {
            }
            // set gets the customers internet id from the customers form
            custNumbInternet = custFormInternet.currentCust;

        }
        // displays the customerform and listens for a ok button push to get the
control back
        private void Ibu_retal_info_Click(object sender, EventArgs e)
        {
            // creates a new customerform for rental
```

```
            customerForm custFormRental = new customerForm();
            // Listens for the push of ok button ín customer form to get back
control
            if (custFormRental.ShowDialog() == DialogResult.OK)
            {
            }
            // set gets the customers rental id from the customers form
            custNumbRental = custFormRental.currentCust;
        }
        // displays the customerform and listens for a ok button push to get the
control back
        private void button1_Click(object sender, EventArgs e)
        {
            // creates a new customerform for other
            customerForm custFormOther = new customerForm();
            // Listens for the push of ok button ín customer form to get back
control
            if (custFormOther.ShowDialog() == DialogResult.OK)
            {
            }
            // set gets the customers other id from the customers form
            custNumbOther = custFormOther.currentCust;

        }
        // displays the customerform and listens for a ok button push to get the
control back
        private void Lbl_elec_price_TextChanged(object sender, EventArgs e)
        {
            updated_GrandTotal();
        }
        // checks for changes being made to the internet price and updates the
grandtotal
        private void Lbl_internet_price_TextChanged(object sender, EventArgs e)
        {
            updated_GrandTotal();
        }
        // checks for changes being made to the laundery price and updates the
grandtotal
        private void Lbl_laundery_price_TextChanged(object sender, EventArgs e)
        {
            updated_GrandTotal();
        }
        // checks laundery is unselected and puts all value to zero
        private void Chb_laundery_CheckedChanged(object sender, EventArgs e)
        {
            // checks if laundery is deselected and then sets all value to zero
            if ((Chb_laundery.Checked == false))
            {
                Nub_washer.Value = 0;
                Nub_dryer.Value = 0;
                Nub_powder.Value = 0;
            }

            else
            {
                Nub_washer.Value = 1;
                updated_LaunderyTotal();
            }
        }
        // checks for changes being made to the rental price and updates the
grandtotal
        private void Lbl_rental_price_TextChanged(object sender, EventArgs e)
        {
            updated_GrandTotal();
        }
        // checks for changes being made to the other price and updates the
grandtotal
        private void Lbl_other_price_TextChanged(object sender, EventArgs e)
```

```
        {
            updated_GrandTotal();
        }

        // checks for changes being made to the internet value and updates the
internettotal
        private void Nub_internet_ValueChanged(object sender, EventArgs e)
        {
            // checks if the value is zero
            if (Nub_internet.Value == 0)
            {
                // set the internet option to unselcted
                Chb_internet.Checked = false;
                // clears the text box
                Comb_internet_Length.Text = "";
                // set the interval to 0
                internetTime = 0;
                // updates the internat price
                updated_InternetTotal();
            }
            // checks the value of internet index and then calc the price
            // (has a fixed price that needs to be connected to price table later)
            else
            {
                // check if internet is seleceted
                Chb_internet.Checked = true;
                // check if internet index of internet and sets it to 30 min
                if (Nub_internet.Value == 1)
                {
                    // set the text explaination to 30 min
                    Comb_internet_Length.Text = "30 min";
                    // set the interval to 1
                    internetTime = 1;
                    // updates the internat price
                    updated_InternetTotal();
                }
                // check if internet index and sets it to 1 hour
                else if (Nub_internet.Value == 2)
                {
                    // set the text explaination to i hour
                    Comb_internet_Length.Text = "1 klst";
                    // set the interval to 2
                    internetTime = 2;
                    // updates the internat price
                    updated_InternetTotal();
                }
                // check if internet index and sets it to 2 hour
                if (Nub_internet.Value == 3)
                {
                    // set the text explaination to 2 hour
                    Comb_internet_Length.Text = "2 klst";
                    // set the interval to 3
                    internetTime = 3;
                    // updates the internat price
                    updated_InternetTotal();
                }
            }
        }
        // checks for changes being made to the internet text in combobox is
changed and updates the internettotal
        private void Comb_internet_Length_SelectedIndexChanged(object sender,
EventArgs e)
        {
            // check if internet text 30 min is index and sets the index to 1
            if (Comb_internet_Length.Text == "30 min")
            {
                // set the interval to 1
                Nub_internet.Value = 1;
```

```
                internetTime = 1;
                updated_InternetTotal();
            }
            // check if internet text 1 hour is index and sets the index to 2
            else if (Comb_internet_Length.Text == "1 klst")
            {
                // set the interval to 2
                Nub_internet.Value = 2;
                internetTime = 2;
                updated_InternetTotal();
            }
            // check if internet text 2 hours is index and sets the index to 3
            if (Comb_internet_Length.Text == "2 klst")
            {
                // set the interval to 3
                Nub_internet.Value = 3;
                internetTime = 3;
                updated_InternetTotal();
            }

        }
        // updating the grandtotal if label is clicked
        // used while developing the code can be deleted later
        private void Lbl_grandtotal_Click(object sender, EventArgs e)
        {
            grandTotal = launderyTotal + internetTotal;
            Lbl_grandtotal.Text = grandTotal.ToString();
        }
        // check if internet option has been selected or deselected
        private void Chb_internet_CheckedChanged(object sender, EventArgs e)
        {
            // if deselected then chandes values and prices to zero
            if ((Chb_internet.Checked == false))
            {
                Nub_internet.Value = 0;
                Comb_internet_Length.Text = "";
                internetTime = 0;
                updated_InternetTotal();
            }
            // if selected set the default value to 1 and 30 min
            else
            {
                Nub_internet.Value = 1;
                Comb_internet_Length.Text = "30 min";
                internetTime = 1;
                updated_InternetTotal();
            }
        }
        // Checks if value of elecrticety has been changes and updates prices
        private void Nub_elec_nights_ValueChanged(object sender, EventArgs e)
        {
            // if the value is zero then set deselect electricity option and clear
text
            if (Nub_elec_nights.Value == 0)
            {
                Chb_elec.Checked = false;
                Cob_elec_location.Text = "";
                Cob_elec_hooknumb.Text = "";
                updated_ElecTotal();
            }
            // if the value is not zero set deselect electricity option and update
prices
            else
            {
                Chb_elec.Checked = true;
                updated_ElecTotal();
            }
        }
```

```
        // Checks if option of elecrticety has been changes and updates prices
        private void Chb_elec_CheckedChanged(object sender, EventArgs e)
        {
            // if the option is not selected then put the value to zero and update
prices
            if ((Chb_elec.Checked == false))
            {
                Nub_elec_nights.Value = 0;
                updated_ElecTotal();
            }
            // if the option is selected then get the value and update prices
            else
            {
                Cob_elec_arrivaldate.Value = Nub_guest_arrivaldate.Value;
                Nub_elec_nights.Value = Nub_guest_nights.Value;
                updated_ElecTotal();
            }
        }
        // Checks if retal has been selecet of deselected
        public void Chb_rental_CheckedChanged(object sender, EventArgs e)
        {
            // if it is selected then open a new window with rental info
            if (Chb_rental.Checked == true)
            {
                // makes a new form for rental infomation
                popupFormLeiga = new rentalform();
                {
                    // listens for the ok button pushed in the rentalform to get
the control back
                    if (popupFormLeiga.ShowDialog() == DialogResult.OK)
                    {
                    }
                updated_RentalInfo();
                }
            }
            // if it is not selected then clear the fields and update prices
            else if (Chb_rental.Checked == false)
            {
                Tbx_rental_ref.Text = "";
                rentalTotal = 0;
                updated_RentalTotal();
            }
        }
        // Checks if other has been selecet of deselected
        private void Chb_other_CheckedChanged(object sender, EventArgs e)
        {
            // if it is selected then open a new window with other info
            if (Chb_other.Checked == true)
            {
                // makes a new form for other infomation
                popupFormOther = new otherform();
                {
                    // listens for the ok button pushed in the otherform  to get
the control back
                                        if (popupFormOther.ShowDialog() ==
DialogResult.OK)
                    {
                    }
                    updated_OtherInfo();
                }
            }
            // if it is not selected then clear the fields and update prices
            else if (Chb_other.Checked == false)
            {
                Txb_other_ref.Text = "";
                otherTotal = 0;
                updated_OtherTotal();
            }
```

```
        }
        // method to update the rental info
        public void updated_RentalInfo()
        {
            // get the rental status for the option button
            Chb_rental.Checked = popupFormLeiga.rentalSelected;
            // get the rental info text from the rental form
            Tbx_rental_ref.Text = "" + popupFormLeiga.ref_rental;
            // get the rental info price from the rental form
            rentalTotal = popupFormLeiga._price_rental;
            // update the price of rental
            updated_RentalTotal();
        }
        // method to update the other info
        public void updated_OtherInfo()
        {
            // get the other status for the option button
            Chb_other.Checked = popupFormOther.otherSelected;
            // get the other info text from the other form
            Txb_other_ref.Text = "" + popupFormOther.ref_other;
            // get the other info price from the other form
            otherTotal = popupFormOther._price_other;
            // update the price of other
            updated_OtherTotal();
        }
        // eventhandler to make new lines in the datagrid
        private void Ibu_guest_plus_Click(object sender, EventArgs e)
        {
            try
            {
                // set guest total to zero
                guestTotal = 0;

                // Here DataRows are added according to amount of adult.
                for (int i = 1; i <= Nub_guest_adult.Value; i++)
                {
                    float calcDis = 0; // temp variable to calc discount
                    float calcPrice = 0; // temp variable to calc price
                    // here is checked if the discaount is because of more than one
night
                    if (Cob_guest_discount.Text == "Aukanótt            ")
                    {
                        calcDis = (float)(Nub_guest_nights.Value - 1) *
float.Parse(Cob_guest_discountKr.Text);
                    }
                    // here is if the discaount type is because of any other than
extra nights
                    else
                    {
                        calcDis = (float)(Nub_guest_nights.Value) *
float.Parse(Cob_guest_discountKr.Text);
                    }
                    // This is to calculate the price for each customer
                    // (has a fixed price that needs to be connected to price table
later)
                    calcPrice = (float)(Nub_guest_nights.Value * 1000) - calcDis;
                    // this adds the line to the datagridtable
                    guestLinetable.Rows.Add(Nub_guest_arrivaldate.Value,
Nub_guest_nights.Value, 1, 0, cob_guest_nationumber.SelectedValue,
cob_guest_nationality.Text, Cob_guest_discount.Text, calcDis, calcPrice);
                }

                // Here DataRows are added according to amount of  children.
                for (int j = 1; j <= Nub_guest_child.Value; j++)
                {
                    guestLinetable.Rows.Add(Nub_guest_arrivaldate.Value,
Nub_guest_nights.Value, 0, 1, cob_guest_nationumber.SelectedValue,
cob_guest_nationality.Text, Cob_guest_discount.Text, 0, 0);
```

```
                }
                // Here the total price for the datagrid table is calculated
                foreach (DataRow dr in guestLinetable.Rows)
                {
                    guestTotal += (float)(dr["guestPrice"]);
                }
                // here the totalprice is updated and guest tolal is set to a label
                Lbl_guests_price.Text = guestTotal.ToString("F2");
                updated_GrandTotal();
            }
            catch (Exception ex)
            {
                MessageBox.Show("VILLA " + ex);
            }
        }



        // listning for changes made in the datagrind
        private void dataGridView1_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
        {
            // allows changes in the datagrid
            guestLinetable.AcceptChanges();
            // trying to update each line of the datagrid
            // not working yet!!!
            foreach (DataRow dr in guestLinetable.Rows)
            {
                //int TmpNight = (int)(dr[1]);
                //int TmpAdult = (int)(dr[2]);
                //float Tmpprice = (float)(TmpNight * TmpAdult);
                //float TmpDiscount = (float)(dr["discount"]);
                //dr["guestPrice"] = Tmpprice-TmpDiscount;
                //guestTotal += (float)(dr["guestPrice"]);
            }
            // update teh guest total price
            update_guestTotal();
            // put the updated value to the display label
            Lbl_guests_price.Text = guestTotal.ToString("F2");
            // update the grandtotal price
            updated_GrandTotal();
        }
        // checks for changes on the label of guest price and updates total price
        private void Lbl_guests_price_TextChanged(object sender, EventArgs e)
        {
            updated_GrandTotal();
        }


        private void OK_button_Click(object sender, EventArgs e)
        {
            // check if the table is empti
            if (guestLinetable != null)

                 try
                {

                    {
                        // adds line for each adult or child to the datagrid
                        foreach (DataRow dr in guestLinetable.Rows)
                        {
                            // adds the guest to add to each line
                            Guest CurrentGuest = new Guest();
                            EntityKey guestKey = new EntityKey();
                            Guest guest1 = Guest.CreateGuest(0);
                            guest1.guestArrivalDate =
(DateTime)dr["guestArrivaldate"];
                            guest1.guestNights = (int)dr["guestNights"];
```

```
                            guest1.guestAdult = (short)dr["guestAdult"];
                            guest1.guestChild = (short)dr["guestChild"];
                            guest1.guestPrice = (float)dr["guestPrice"];
                            guest1.NationsReference.EntityKey = new
EntityKey("TjaldurEntities.Nations", "id_Nations", (int)dr["id_nations"]);
                            guest1.DiscountReference.EntityKey = new
EntityKey("TjaldurEntities.Discounts", "id_Discount",
Cob_guest_discount.SelectedValue);
                            guest1.InvoiceReference.EntityKey = new
EntityKey("TjaldurEntities.Invoices","id_Invoice",Int64.Parse(inv1.id_Invoice.ToStr
ing()));
                            // saves the guest to the guest table
database/entityset
                            tje.AddToGuests(guest1);
                            // saves the changes in the database/entityset
                            tje.SaveChanges();
                            //
                            CurrentGuest = guest1;
                            //MessageBox.Show(CurrentGuest.id_Guest.ToString());


                        } // end of for each

                        // get all info about the carregistration for this invoice
                        CarReg CurrentCar = new CarReg();
                        CurrentCar.start_date =
DateTime.Parse(carForm.nights_calc.ToString());
                        CurrentCar.exp_date =
DateTime.Parse((carForm.nights_calc.AddDays(carForm.carNights)).ToString());
                        CurrentCar.CarNum = carForm.currentCarNumb.ToString();
                        CurrentCar.Rfid_numb = carForm.currentRfId.ToString();
                        CurrentCar.InvoiceReference.EntityKey = new
EntityKey("TjaldurEntities.Invoices", "id_Invoice",
Int64.Parse(inv1.id_Invoice.ToString()));
                        CurrentCar.CustomersReference.EntityKey = new
EntityKey("TjaldurEntities.Customers", "id_Customer", carForm.custNumb);

                        // saves all the info about carregistration
                        tje.AddToCarReg(CurrentCar);

                        Electricity CurrentElec = new Electricity();
                        CurrentElec.elecArrivalDate = Nub_guest_arrivaldate.Value;
                        //CurrentElec.elecLocation =
Cob_elec_location.SelectedText; look into
                        //CurrentElec.elecHookNumb =
Int32.Parse(Cob_elec_location.SelectedText); look into
                        CurrentElec.elecPrice = Int32.Parse(elecTotal.ToString());
                        CurrentElec.elecVat =
(float)Int32.Parse(elecTotal.ToString())* 0.203;
                        tje.AddToElectricities(CurrentElec);


                        tje.SaveChanges();
                        // set the date on the invoice to now
                        inv1.invDate = DateTime.Now;
                        // get the number of the invoice to save
                        inv1.invNumb = Int64.Parse(Lbl_InvNumb.Text.ToString());
                        _currentInv = (Int64)inv1.invNumb;
                        // get the total amount to save
                        inv1.invTotalAmount = (double)grandTotal;
                        // save data to the database/entityset
                        tje.SaveChanges();
                    }
                    //  here the reports will be printed automatically but for know
it is on a special button
                    //  report2 Testreport = new report2();
                    //  if (Testreport.ShowDialog() == DialogResult.OK)
                    //  {
```

```
            //  }

            // clearing all data on the form for the next
            clearAllDataOnForm();
          //  this.dataGridView1 = new System.Windows.Forms.DataGridView();
            InitDataOnForm();
        }
         catch (Exception ex)
         {
             MessageBox.Show("Exception caught. " + ex);
         }
    else
        toolStripStatusLabel1.Text = "Engar færslur";
}
// if cancel button is used the the forme is cleared
private void Btn_Cancel_Click(object sender, EventArgs e)
{
    // clear the data on the form
    clearAllDataOnForm();
    // make the datagrid again
    CreateDataTable();
   // Close();
}
// checks if the guest option button is seleceted or not
private void Chb_guest_CheckedChanged(object sender, EventArgs e)
{
    // Check if guest is selected or not
    if (Chb_guest.Checked != true)
    {
        // make the buttons for adding and deleting unselectable
        Ibu_guest_plus.Enabled = false;
        Ibu_guest_minus.Enabled = false;

        // Clear the datagrid and update prices
        if (guestLinetable.Rows != null)
        {
            guestLinetable.Rows.Clear();
            guestLinetable = null;
            guestTotal = 0;
            Lbl_guests_price.Text = guestTotal.ToString("F2");
            updated_GrandTotal();
            // Add the button column to the control.
            dataGridView1.Columns.Clear();

            dataGridView1.DataSource = null;
            guestLinetable = null;
            CreateDataTable();
        }
    }
    // enebles the plus and minus buttons again
    else
    {
        Ibu_guest_plus.Enabled = true;
        Ibu_guest_minus.Enabled = true ;
    }

}



// Calls the Employee.RequestStatus method.
void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
    // Ignore clicks that are not on button cells.
        if (e.RowIndex < 0 || e.ColumnIndex !=
            dataGridView1.Columns["eydaTakki"].Index) return;
```

```
                {
                    //MessageBox.Show(dataGridView1[1,
e.RowIndex].Value.ToString());
                    dataGridView1.CurrentRow.Selected = true;

                    if (this.dataGridView1.SelectedRows.Count > 0 &&
                        this.dataGridView1.SelectedRows[0].Index !=
                        this.dataGridView1.Rows.Count - 1)
                    {
                        this.dataGridView1.Rows.RemoveAt(
                            this.dataGridView1.SelectedRows[0].Index);
                    }
                    update_guestTotal();
                }

            }
            catch (Exception ex)
            {
                MessageBox.Show("" + ex);
            }
        }
        //checks if the laundery customer button is selected
        private void Ibu_laundery_info_Click(object sender, EventArgs e)
        {
            // creates a new customer form for the electricity
            customerForm custFormLaundery = new customerForm();
            // listens for the ok button on the costomerform to get bak control
            if (custFormLaundery.ShowDialog() == DialogResult.OK)
            {
            }
            // gets the value of the id of the customer using the laundery
            custNumbLaundery = custFormLaundery.currentCust;
        }
        // checks if the more info or carreg button is used
        private void btn_carreg_Click(object sender, EventArgs e)
        {
            try
            {
                // gets the value of the car ergistration id
                carForm.nights_calc = Nub_guest_arrivaldate.Value;
                carForm.carNights = (float)Nub_guest_nights.Value;
                // listensfor the ok button push on the carregform to get control
again
                if (carForm.ShowDialog() == DialogResult.OK)
                {
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("" + ex);
            }
        }
        // a method to update the first line of the form to default
        private void updateGuestlineTodefault()
        {
            // set the arrivaldate to current day
            Nub_guest_arrivaldate.Value = DateTime.Now;
            // set the number of nights to 1
            Nub_guest_nights.Value = 1;
            // set the amount of adult to 1:
            Nub_guest_adult.Value = 1;
            // set the amount of child to 0:
            Nub_guest_child.Value = 0;
            // set the nationality to Icelandic:
            cob_guest_nationumber.SelectedIndex = (0);
            // set the discount to zero:
            Cob_guest_discount.SelectedIndex = (0);
```

```
}

private void Ibu_guest_minus_Click(object sender, EventArgs e)
{
    // set the value of data in firts line to default
    try
    {
        updateGuestlineTodefault();
    }
    catch (Exception ex)
    {

        MessageBox.Show("" + ex);
    }
}
// Clear all values and put them to a default posission
public void clearAllDataOnForm()
{
    // checks if the datagrid is empty
    if (dataGridView1.DataSource != null || guestLinetable != null)
        {
            // sets all the data on the form to defaulr
            try
                {
                // clears the guest datagrid
                dataGridView1.Columns.Clear();
                dataGridView1.DataSource = null ;
                guestLinetable = null;
                // set sthe guest total to zero
                guestTotal = 0;
                // set the electricity total to zero
                elecTotal = 0;
                // set the laundery total to zero
                launderyTotal = 0;
                // set the internet total to zero
                internetTotal = 0;
                // set the rental total to zero
                rentalTotal = 0;
                // set the other total to zero
                otherTotal = 0;
                // set the grand total to zero
                grandTotal = 0;
                // deselect the electricity option
                if (Chb_elec.Checked)
                    Chb_elec.Checked = false;
                // deselect the  internet option
                if (Chb_internet.Checked)
                    Chb_internet.Checked = false;
                // deselect the  laundery option
                if (Chb_laundery.Checked)
                    Chb_laundery.Checked = false;
                // deselect the  other option
                if (Chb_other.Checked)
                    Chb_other.Checked = false;
                // deselect the  rental option
                if (Chb_rental.Checked)
                    Chb_rental.Checked = false;
                // put the guest total text on the label to "0"
                Lbl_guests_price.Text = guestTotal.ToString("F2");

                updated_GrandTotal();
                // set the guest line to default
                updateGuestlineTodefault();
                // setting the travelby and accomidation to default
                Cob_travel_method.SelectedIndex = (0);
                Cob_facilities.SelectedIndex = (0);
            }
    catch (Exception ex)
```

```
                {
                    MessageBox.Show("Exception caught. " + ex);
                }
            }

        else
            // Info about no data being in the form displayed to menustrip
                toolStripStatusLabel1.Text = "Ekki má eyða ef engar færslur
eru";
        }

        private void btn_reports_Click(object sender, EventArgs e)
        {
            // makes a new form for the reports to get the control back
            report2 Testreport = new report2();
            // listens to the ok button of the new form
            if (Testreport.ShowDialog() == DialogResult.OK)
            {
            }
        }
        // listening to the event of changed value of the amount of nights
        private void Nub_guest_nights_ValueChanged(object sender, EventArgs e)
        {
            // update the depaturedate depending on the number of nights
            if (Nub_guest_nights.Value > 1)
            {
                DateTime tempdate = Nub_guest_arrivaldate.Value;

                Nub_guest_departuredate.Value =
tempdate.AddDays((int)Nub_guest_nights.Value);
            }
        }
        // listening to the event of changed depature date
        private void Nub_guest_departuredate_ValueChanged(object sender, EventArgs
e)
        {
            // update the arrival depending on the number of nights

            DateTime tempdate = Nub_guest_departuredate.Value;

            Nub_guest_arrivaldate.Value = tempdate.AddDays(-
(int)Nub_guest_nights.Value);
        }


    }
}
```

class carregform

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using TjaldurDAL;

namespace Tjaldur04
{
    public partial class carregform : Form

    {
        public DateTime _nights_calc = DateTime.Now; // variable for the arrival
date
        public float _carNights = 0; // variable for the nights stayed
        private  string _currentRfId ="óþekkt"; // variable for the RFID id
        private string _currentCarNumb = "óþekkt";// variable for the
carregistration number
        private Int64 _custNumb; // variable for the custumer ID number
        // create an now instance of the customerform class
        public customerForm custForm2 = new customerForm();
        // create a object of the apsx reader class
        APSXDLL.apsxrfid apsx = new APSXDLL.apsxrfid();
        byte[] uidbytes = null; // cariable for the id number of the card

        public carregform()
        {
            InitializeComponent();
        }

        private void carregform_Load(object sender, EventArgs e)
        {
            // set the text of the depature date
            tbx_carreg_expDate.Text =
_nights_calc.AddDays(_carNights).ToString("dd.MM.yyyy 16:00:00");
            // set the text of the depature date
            tbx_carreg_arrivalDate.Text = _nights_calc.ToString();
            // gets the value of the
            tbx_carreg_expDate.Text =
_nights_calc.AddDays(_carNights).ToString("dd.MM.yyyy 16:00:00");
            // opens the serial port used for the connection of the reader
            apsx.OpenPort("COM1");
        }
        //  to get and set customer id nubber
        public Int64 custNumb
        {
            get { return _custNumb; }
            set { _custNumb = value; }
        }
        //  to get and set customer RFID tag
        public string currentRfId
        {
            get { return _currentRfId; }
            set { _currentRfId = value; }
        }
        // to get and set customer car registration number
        public string currentCarNumb
        {
            get { return _currentCarNumb; }
            set { _currentCarNumb = value; }
        }
        //  to get and set customer arrival date
```

```
        public DateTime nights_calc
        {
            get { return _nights_calc; }
            set { _nights_calc = value; }
        }
        //  to get and set customer nights
        public float carNights
        {
            get { return _carNights; }
            set { _carNights = value; }
        }
        //checks if the RFID read button is used
        private void btn_carreg_readRFID_Click(object sender, EventArgs e)
        {
            // Variable for the version number of the reader
            byte dsbyte = 0;
            // check if the serilaport is open or not
            if (apsx.OpenPort("COM1") == false)
            {
                MessageBox.Show("Tenging fékst ekki við COMMPORT", "VILLA",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            //if the serialport is open the it check if the version of reader is
higher than 103
            else
            {
                // a mothod from the APSX.dll to get the version
                dsbyte = apsx.GetAPSXVersion();
                // if the software cant connect to the hardware of the reader i.e.
it is lower that 103
                if (dsbyte < 103)
                    MessageBox.Show("Tenging fékst við EKKI lesara " +
dsbyte.ToString(), "VILLA", MessageBoxButtons.OK, MessageBoxIcon.Error);
                // the connection is establish
                else
                {
                   // reading the ID number from the manudfacturer
                    if (apsx.GetUID(ref uidbytes) == true)
                    {
                        // this is the ascii version iof the tag. Later I would use
this to make simpler tags
                        //label1.Text = apsx.BytesToAsciiString(uidbytes);
                        // put the id number to the label in the form
                        Lbl_RFIDtag.Text = apsx.BytesToDecString(uidbytes);
                        // set the tag as current tag for the main form
                        currentRfId = Lbl_RFIDtag.Text;

                    }
                    else
                        Lbl_RFIDtag.Text = "Kort lest EKKI";
                }
            }
        }
        // check if the customer button of the carregform is selected
        private void Ibu_car_info_Click(object sender, EventArgs e)
        {
            //create a new object of the customer form
            customerForm custForm2 = new customerForm();
            // listens to the ok button of the new customer form to get back the
control
            if (custForm2.ShowDialog() == DialogResult.OK)
            {
            }
            // Set the current customer for the main form
            _custNumb = custForm2.currentCust;
        }
        // checks for the use of the cancelbutton on the form
        private void btn_carreg_Cancel_Click(object sender, EventArgs e)
```

```
        {
            // close the serial port after usage
            apsx.ClosePort();
            // close the form
            this.Close();
        }
        //checks for the use of the OK button on the form
        private void btn_carreg_OK_Click(object sender, EventArgs e)
        {
            // close the serial port after usage
            apsx.ClosePort();
             // get the carreg carnumber for the mainform
            _currentCarNumb= tbx_carregNumb.Text.ToString();
            // close the form
            this.Close();
        }
    }
}
```

class customerForm

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using TjaldurDAL;
using System.Data.Objects;



namespace Tjaldur04
{
    public partial class customerForm : Form
    {
        // create a instance of the entity set
        private TjaldurEntities tjeCust = new TjaldurEntities();
        // variabel for the customer id number
        public Int64 _currentCust;
        // create a object of the customer kind
        public Customer c1;

        public customerForm()
        {
            InitializeComponent();
        }
        // form load method
        private void customerForm_Load(object sender, EventArgs e)
        {
            // defines the dataconnection to the entity set
            this.customerBindingSource.DataSource = tjeCust.Customers;
            // searshstring set as empty
            TBX_Search.Text = "";
        }
        //To get and set the customer number
        public Int64 currentCust
        {
            get { return _currentCust; }
            set { _currentCust = value; }
        }
        // method to make a query to the database
        private void BindData()
        {
            // object query list that you an use to search for name or kennital
            ObjectQuery<Customer> custqur = tjeCust.CreateQuery<Customer>(
                    "SELECT VALUE c FROM Customers AS c WHERE c.custName LIKE '%" +
TBX_Search.Text + "%' OR c.custKennitala LIKE  '" + TBX_Search.Text + "%'ORDER BY
c.custName");
            this.customerBindingSource.DataSource = custqur;
        }
        // Check if the savebutton in the navigatorbar is used
        private void customerBindingNavigatorSaveItem_Click(object sender,
EventArgs e)
        {
            try
            {
                // saves the selected customer as current customer for the main
form
                Customer selectedCostomer =
(Customer)customerBindingNavigator.BindingSource.Current;
                _currentCust = selectedCostomer.id_Customer;
                //check if the customer has been added or is from the list of
customers
```

```
                if (_currentCust == 0)
                {
                    updateNewcustomer();

                    MessageBox.Show("Nýr stofnaður viðskiptavinur");
                }
                // check if the customer was not added just updated
                else
                {
                    // saves the upated info to the entity set
                    tjeCust.SaveChanges();
                    // message to show that the customer has been updated
                    MessageBox.Show("Uppfærsla viðskiptavinar");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Póstnumer verður að vera tölustafir " + ex);
            }
        }
        // checks if the search button on the tool strip manu was used
        private void toolStripButton1_Click(object sender, EventArgs e)
        {
            // method to seach the database
            BindData();
        }
        // checks if the ok button is used
        private void Btn_Custinfo_OK_Click(object sender, EventArgs e)
        {
                // get the id number of the current customer
                Customer selectedCostomer =
(Customer)customerBindingNavigator.BindingSource.Current;
                _currentCust = selectedCostomer.id_Customer;
                // checks if the customer has been added or is from the database
                if (_currentCust == 0)
                {
                    //makes a new customer
                    updateNewcustomer();
                }
                // saves the customer to the database
                tjeCust.SaveChanges();
                //closes the form
                this.Close();
        }
        // check if the canel button is used
        private void Btn_Custinfo_Cancel_Click(object sender, EventArgs e)
        {
            //closes the form with out saving
            this.Close();
        }
        // a method to add new customers
        private void updateNewcustomer()
        {
            try
            {
                // creates a customer if he did not exists already
                Customer c1 = Customer.CreateCustomer(0);
                // puts the value of the customer name to the customer
                c1.custName = custNameTextBox.Text;
                // puts the value of the customer address to the customer
                c1.custAddress = custAddressTextBox.Text;
                // puts the value of the customer kennitala to the customer
                c1.custKennitala = maskedTextBox1.Text;
                // puts the value of the customer city to the customer
                c1.custCity = custCityTextBox.Text;
                // puts the value of the customer postal code to the customer
                c1.custPostal = int.Parse(custPostalTextBox.Text.ToString());
                // puts the value of the customer country to the customer
```

```
                c1.custCountry = custCountryTextBox.Text;
                // adds the customer to the entity set
                tjeCust.AddToCustomers(c1);
                // saves the chaneges of the entity set
                tjeCust.SaveChanges();
                // gets the id number of the current customer for the main form
                _currentCust = c1.id_Customer;
                // to show what has happend

            }
            catch (Exception ex)
            {

                MessageBox.Show("Verður að hafa póstnúmer með tölustöfum");
            }

        }
    }
}
```

class rentalform

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using TjaldurDAL;
using System.Data.Objects;

namespace Tjaldur04
{
    public partial class rentalform : Form
    {
        public Boolean rentalSelected =true;// boolean to triggar an event in the
main form
        public string _ref_rental;// variable for the reference of the rental form
        public int _price_rental = 0;// variable for the price of the rental form

        public rentalform()
        {
            InitializeComponent();
        }

        // to get and st the reference to the main form
        public string ref_rental
        {
        get { return _ref_rental; }
        set { _ref_rental = value; }
        }
        // to get and st the price to the main form
        public int price_rental
        {
            get { return _price_rental; }
            set { _price_rental = value; }
        }
        // loadmethod of the form
        public void testform_Load(object sender, EventArgs e)
        {
        }
        // checks if the ok button is used
        private void Btn_Rental_popup_ok_Click(object sender, EventArgs e)
        {
            try
            {

            // check if the price is zero
                if (Txb_Rental_popup_price.Text.Equals(""))
                {
                    // if price is zero the update the label to "0"
                    _price_rental = int.Parse(Txb_Rental_popup_price.Text = "0");
                    //deselect the rental option
                    rentalSelected = false;
                }
                // set the reference text  from the textbos to the main form
                _ref_rental = (string)Txb_Rental_popup_skyring.Text.ToString();
                // set the price text  from the textbos to the main form
                _price_rental = int.Parse(Txb_Rental_popup_price.Text.ToString());
                // close the form
                this.Close();
            }
            catch (Exception ex)
            {
                // displays the info about the price should be numbers
                MessageBox.Show("Verða að vera tölustafir í verð reitnum!!!");
```

```
                // and set the rental option as deselected
                rentalSelected = false;
            }

        }


        private void Btn_Leiga_popup_cancel_Click(object sender, EventArgs e)
        {
            // checks if the cancel button has been used
            rentalSelected = false;
            // close the form
            this.Close();
        }


    }
}
```

class otherform

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using TjaldurDAL;
using System.Data.Objects;

namespace Tjaldur04
{
    public partial class otherform : Form
    {
        public Boolean otherSelected =true;// boolean to triggar an event in the
main form
        public string _ref_other;// variable for the reference of the other form
        public int _price_other = 0; //variable for the price of the other form

        public otherform()
        {
            InitializeComponent();
        }
        // to get and st the reference to the main form
        public string ref_other
        {
        get { return _ref_other; }
        set { _ref_other = value; }
        }
        // to get and st the price to the main form
        public int price_other
        {
            get { return _price_other; }
            set { _price_other = value; }
        }
        // loadmethod of the form
        public void otherform_Load(object sender, EventArgs e)
        {
        }
        // checks if the ok button is used
        private void Btn_Other_popup_ok_Click(object sender, EventArgs e)
        {
            try
            {
                // check if the price is zero
                if (Txb_Other_popup_verd.Text.Equals(""))
                {
                    // if price is zero the update the label to "0"
                    _price_other = int.Parse(Txb_Other_popup_verd.Text = "0");
                    //deselect the other option
                    otherSelected = false;
                }
                // set the reference text  from the textbos to the main form
                _ref_other = (string)Txb_Leiga_popup_skyring.Text.ToString();
                // set the price text  from the textbos to the main form
                _price_other = int.Parse(Txb_Other_popup_verd.Text.ToString());
                // close the form
                this.Close();
            }
            catch (Exception ex)
            {
                // displays the info about the price should be numbers
                MessageBox.Show("Verða að vera tölustafir í verð reitnum!!!");
                // and set the other option as deselected
                otherSelected = false;
```

```
            }
        }

        // checks if the cancel button has been used
        private void Btn_Other_popup_cancel_Click(object sender, EventArgs e)
        {
            // and set the other option as deselected
            otherSelected = false;
            // close the form
            this.Close();
        }


    }
}
```

Class report2 auto generated

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.Objects;
using TjaldurDAL;

namespace Tjaldur04
{
    public partial class report2 : Form
    {
        private TjaldurEntities tjeInv = new TjaldurEntities();
        public report2()
        {
            InitializeComponent();
        }

        private void report2_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'Tjaldur02DataSet1.kvittanalisti' table. You can move, or remove it, as needed.

this.kvittanalistiTableAdapter.Fill(this.Tjaldur02DataSet1.kvittanalisti);

            this.reportViewer1.RefreshReport();
        }
    }
}
```

class Program auto generated

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace Tjaldur04
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
```

```
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Autoconfig (connection to database)

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add name="TjaldurEntities"
connectionString="metadata=res://*/TjaldurModel.csdl|res://*/TjaldurModel.ssdl|res:
//*/TjaldurModel.msl;provider=System.Data.SqlClient;provider connection
string='Data Source=.\SQLEXPRESS;AttachDbFilename=&quot;C:\Program Files\Microsoft
SQL Server\MSSQL.1\MSSQL\Data\Tjaldur02.mdf&quot;;Integrated Security=True;Connect
Timeout=30;User Instance=True;MultipleActiveResultSets=True'"
      providerName="System.Data.EntityClient" />
    <add name="Tjaldur04.Properties.Settings.Tjaldur02ConnectionString"
      connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=&quot;C:\Program
Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\Tjaldur02.mdf&quot;;Integrated
Security=True;Connect Timeout=30;User Instance=True"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

Appendix B

9.1 Gantt chart of Project

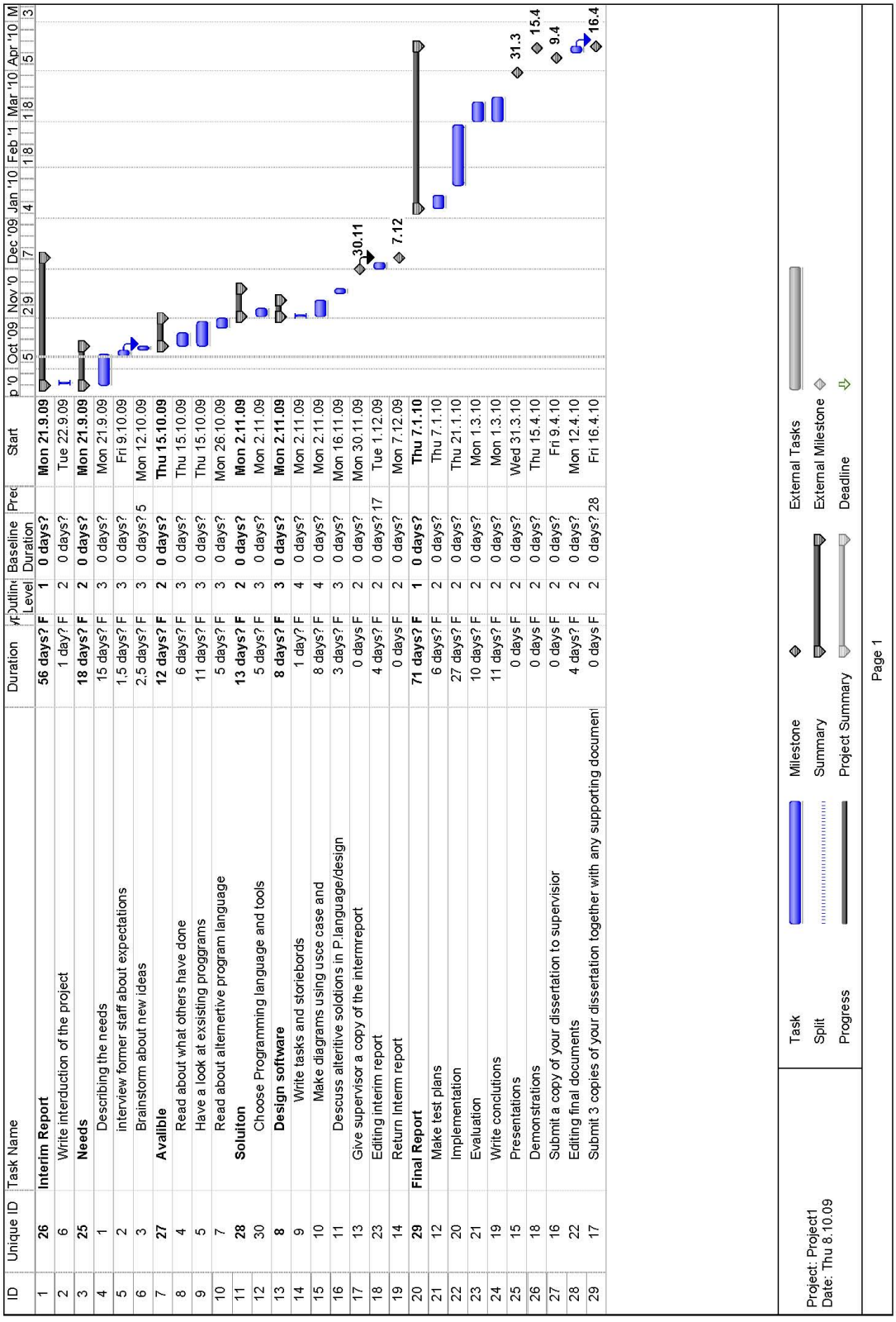


Figure 19 – Gantt chart of project

Task list for programmer of TJALDUR. campsite RFID Access system

1. **Interim Report**
   - 1.1. Write introduction of the project
     - 1.1.1. Describe the problem
     - 1.1.2. Describe the motivation for solving
     - 1.1.3. Describe the possible solution
     - 1.1.4. Describe alternatives solutions
     - 1.1.5. Justify the choice of language
     - 1.1.6. Justify the choice of tools
   - 1.2. Related work and Requirements Analysis and Design
     - **1.2.1. Needs**
       - 1.2.1.1. Describing the needs for the system in words
       - 1.2.1.2. Interview former staff about expectations
       - 1.2.1.3. Interview manager of campsite about expectations
       - 1.2.1.4. Brainstorm about new ideas (meeting with former Employee)
       - 1.2.1.5. Make Mind map of solutions
     - **1.2.2. Available**
       - 1.2.2.1. Read about what others have done
         - 1.2.2.1.1. Looking at RFID solutions
         - 1.2.2.1.2. Looking at NFC solutions
       - 1.2.2.2. Have a look at existing programs
         - 1.2.2.2.1. Compusoft
           Made in Denmark for campsites, hostels and carparks.
         - 1.2.2.2.2. MADA
           Made in Germany for Campsites
         - 1.2.2.2.3. Campground manager
         - 1.2.2.2.4. Campground master
         - 1.2.2.2.5. NO Q Caravan
       - 1.2.2.3. Read about alternative program language
         - 1.2.2.3.1. C
         - 1.2.2.3.2. C++
         - 1.2.2.3.3. C#
         - 1.2.2.3.4. Java
     - **1.2.3. Solution**
       - 1.2.3.1. Choose Programming language
         - 1.2.3.1.1. C#
       - 1.2.3.2. Choose Hardware to use
         - 1.2.3.2.1. RFID reader 13 mhz
         - 1.2.3.2.2. USB digital in and output
       - 1.2.3.3. Choose Software to use
         - 1.2.3.3.1. Microsoft Word
         - 1.2.3.3.2. Microsoft Excel
         - 1.2.3.3.3. Freehand
         - 1.2.3.3.4. Gliffy
         - 1.2.3.3.5. Microsoft Visual Studio 2008
         - 1.2.3.3.6.
       - 1.2.3.4.
       - 1.2.3.5. Choose database to use
         - 1.2.3.5.1.
     - 1.2.4. Discuss alternative solutions in P.language
       - 1.2.4.1. Libraries for other programming language
       - 1.2.4.2.
     - 1.2.5. Discuss alternative hardware
       - 1.2.5.1. Looking at Mirror/rabbit
       - 1.2.5.2. Examining frequency for RFID
     - 1.2.6. Discuss alternative software
       - 1.2.6.1.
     - **1.2.7. Design software**
       - 1.2.7.1. Write tasks list
         - 1.2.7.1.1. Evaluate task list
       - 1.2.7.2. Make storyboards
         - 1.2.7.2.1. Evaluate Storyboards
       - 1.2.7.3. Make diagrams using use case
         - 1.2.7.3.1. Evaluate Use case diagrams
       - 1.2.7.4. Make ER diagram of database
         - 1.2.7.4.1. Evaluate ER diagrams
   - 1.3. Give supervisor a copy of the interim report
   - 1.4. Editing interim report
   - 1.5. Return Interim report

2. **Final Report**
   - 2.1. Make test plans
     - 2.1.1. Get evaluators involved
     - 2.1.2. Make a questioner
     - 2.1.3. Make software test cases

2.2. Implementation
    2.2.1. Implement interface
    2.2.2. Implement database
    2.2.3. Implement code for hardware
    2.2.4.
2.3. Evaluation
    2.3.1. Get users feed back
    2.3.2. Run test cases
    2.3.3. Get feedback from evaluators
    2.3.4.
2.4. Write conclusions
2.5. Submit a copy of your dissertation to supervisor
2.6. Editing final documents
2.7. Submit 3 copies of your dissertation together with any supporting document
2.8.

3. Presentations
    3.1. Make slide show
    3.2. Make handouts
4. Demonstrations
    4.1. Get computers to run software on

Appendix B
Mindmap of possible solution

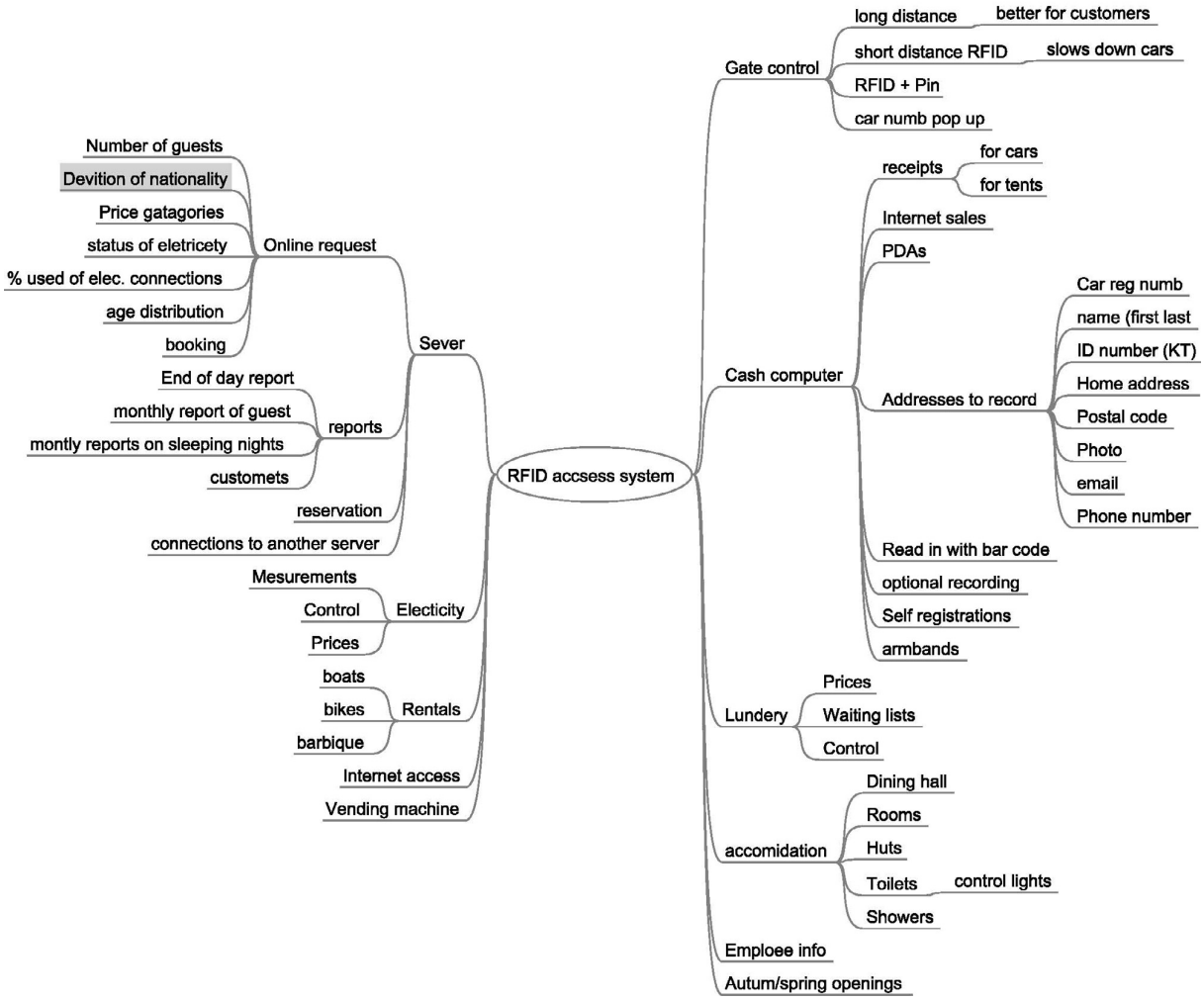


Figure 20 – Mind map

Appendix C

Evaluation 12.04.2010 at 20:00 – 21:00

Evaluator: A staff member at the campsite that has worked at the site for 8 summers and has good computer skills

The evaluation took place at my home where I had the program running on my Dell Latitude 830 laptop computer and had the APSX reader connecter to it. I used a extra display to monitor the actions of the user. I explained to the evaluator the reasons for this project, what it was supposed to do and told him how much of the whole project I had implemented to this point. During the evaluation I answered questions if ask if the evaluator needed information to continue the evaluation. Here are the points noted down at the evaluation session:

1. Departure date should also be displayed
2. RFID reader did not work when tested
3. Date in Electric arrival date should be automatically change with arrival date
4. Tab order on main form is incorrect
5. Unclear how to search in customer form
6. Postal code does not bring up known cities
7. Save of customer info dose not work
8. Internet time was unclear value mixed with time length
9. Where should additional often used things be put like boats, bicycle-cars.
10. How to fix if errors are done???
11. Error while pushing + after –
12. When resizing the form, parts of the form did look right
13. Part of the Vat amount is not displayed
14. How to put reference text to a certain invoice like reason for how it is done
15. How to but reason for voucher or more info about the voucher
16. Discount is always based up on nights and always uses full price for first night
17. When deleting a line in data grid by using the delete button two lines sometimes go together
18. No reason for having a combo box for the amount of discount in first line
19. Changes on the combo boxes using the keyboard do not work
20. Inner search in combo boxes does not work
21. Location and number for electricity does give any possible values
22. Possibility to order time for laundry
23. The checkbox for guest does not delete guest in a form if unselected

Evaluation 13.04.2010 at 14:30-15:30

Evaluator: IT person 33 years old, studied in HR University and is working as a Programmer for an Icelandic company developing Microsoft Dynamics solutions.

The evaluation took place at my home where I had the program running on my Dell Latitude 830 laptop computer and had the APSX reader connecter to it. I used a extra display to monitor the actions of the user. I explained to the evaluator the reasons for this project, what it was supposed to do and told him how much of the hole project I had implemented to this point. I also gave him a list of Heuristics to look for in the program(see appendix c). During the evaluation I answered questions if ask if the evaluator needed information to continue the evaluation. Here are the points noted down at the evaluation session:

1. Adult should be default 1, so if you press plus button it gives at least one line.
2. What about other items sold as Showers.
3. Discount does not work for 1 night
4. Tab order is not correct – very important
5. No shortcut keys – always wanted
6. Washer should have the most used one as default at least 1 wash, maybe 1 of each
7. Put error handling on rental and other popup screens
8. The user login should display notandi to keep up with other language settings
9. When using number input and enter the country is not updated.
10. Discount rules for additional nights discount so if more than 2 nights then rule would activate discount.
11. Nationality allows for input of new names, should not be possible
12. Text in Discount is not all visible when dropdown menu opens
13. Delete line in data grid button does not work with hitting Enter when selected
14. If a line in data grid is deleted the info is not updated (price)
15. In the report the sum is for all data in the list not only current page
16. Current customer should be displayed as default inside the same invoice
17. The OK button does not save the new customer has to hit save button to store new customer.
18. Searching is dependent on – in Kennitala
19. Prices are not align to right
20. Possible to have both adult and children in same data line in data grid.
21. No default value on radio button at the bottom

Evaluation 14.04.2010 at 23:00 – 02:00

Evaluator: The manager at the campsite that has worked at the site for 2 years and has avarage computer skills

The evaluation took place at my home where I had the program running on my Dell Latitude 830 laptop computer and had the APSX reader connecter to it. I used a extra display to monitor the actions of the user. I explained to the evaluator the status of the project and how I would see the rest of the work. During the evaluation I answered questions if ask if the evaluator needed information to continue the evaluation. Here are the points noted down at the evaluation session:

1. Format of day and time in data grid He wanted to get rid of the time so changing the day would be easier
2. Discount options needed to be more or even a possibility to changes to percents of the amount
3. Deleting a line in the data grid through exceptions
4. Extra night should not be default for more than two nights
5. Possibility of making a fee for children some times
6. Car number is it necessary will the staff member ever be able to see it
7. Should car number have a special format
8. How to register a company as the customer?
9. OK in tool strip panel was confusing – evaluator tried to push it
10. Customers information buttons where not clearly marked by the image on the button
11. Add a combo box to the Rental window with the know elements that are being rented, such as bouncer, boats, tents, tables and chairs, the houses and cottages, barbeques.
12. Internet should also have a 24 hours options for those with their own computers
13. Other should also be able to connect to a customer
14. If laundry option is selected it does not add to the price only the value of one
15. Default money should be on the payment radio buttons
16. Cards should maybe be a combo box with all kind of cards or at least debit/ credit
17. How to make credit invoices???

Appendix D

Task list for TJALDUR, campsite RFID access system

1. Login

1.1.Sever

1.1.1.Setup

1.1.1.1. Company info

1.1.1.1.1. Name

1.1.1.1.2. Address

1.1.1.1.3. Vat number

1.1.1.1.4. Logo

1.1.1.1.5. Receipt text

1.1.1.2. Prices

1.1.1.2.1. Price

1.1.1.2.2. Discount groups

1.1.1.2.3.

1.1.1.3. Nationalities

1.1.1.4. Users

1.1.1.5. Map of site

1.1.1.6. Connections

1.1.2.On line requests

1.1.2.1. Status from cash computer

1.1.2.1.1. Amount in cash drawer

1.1.2.1.2. Amount of transactions

1.1.2.1.3. Amount of guest to day

1.1.2.1.4. Amount of sold electricity connections

1.1.2.1.5. Availability of electric connections

1.1.2.1.6. Status on laundry

1.1.2.1.7. Status on dryer

1.1.2.1.8. Status on Internet

1.1.2.1.9. Status on rental

1.1.3.Reports for chosen period

1.1.3.1. Guests

1.1.3.1.1. Pr. Night

1.1.3.1.2. Pr. Stay

1.1.3.1.3. Pr. Nation

1.1.3.1.4. Pr. customer

1.1.3.2. Electricity

1.1.3.2.1. Income

1.1.3.2.2. Amount of connections

1.1.3.2.3. Usage pr. Connection

1.1.3.2.4. Total usage

1.1.3.3. Laundry

1.1.3.3.1. usage

1.1.3.3.2. income

1.1.3.4. Dryer
1.1.3.4.1. Usage
1.1.3.4.2. income
1.1.3.5. Internet
1.1.3.5.1. Usage
1.1.3.5.2. income
1.1.3.6. Rental
1.1.3.6.1. Usage
1.1.3.6.2. income
1.1.3.6.3. Other
1.1.4. View of entrances
1.1.4.1. Show exp. Date
1.1.4.2. Show car number
1.1.4.3. Show number of people in car
1.1.4.4. Reports on statistic
1.1.4.4.1. How many guest at cite at given period
1.1.4.4.2. How many stay over's at given period
1.1.4.4.2.1. Based on nationality
1.1.4.4.2.2. Based on person category

1.2. Cash computer

1.2.1. Sales
1.2.1.1. Normal sales
1.2.1.1.1. Arrival Date
1.2.1.1.2. Number of nights
1.2.1.1.3. Number of guests
1.2.1.1.4. Nationality
1.2.1.1.5. Discount
1.2.1.1.6. Location on the campsite(option)
1.2.1.1.7. Travelling info
1.2.1.1.7.1. Walking
1.2.1.1.7.2. Driving
1.2.1.1.7.3. group
1.2.1.1.8. Accommodation
1.2.1.1.8.1. Tent
1.2.1.1.8.2. RV
1.2.1.1.8.3. Camper
1.2.1.1.8.4. Foldable camper
1.2.1.1.8.5. Tent camper
1.2.1.1.8.6. Hut
1.2.1.1.8.7. Room
1.2.1.1.8.8. Visit
1.2.1.2. Additional things
1.2.1.2.1. RFID cards
1.2.1.2.2. Electricity
1.2.1.2.2.1. Select location

1.2.1.2.2.2.
1.2.1.2.3. Laundry
1.2.1.2.3.1. Time order
1.2.1.2.4. Dryer
1.2.1.2.4.1. Time order
1.2.1.2.5. Internet
1.2.1.2.6. Rentals
1.2.1.2.6.1. Boats
1.2.1.2.6.2. Bikes
1.2.1.2.6.3. Barbeque
1.2.1.2.6.4. Frisbee disk
1.2.1.2.7. Vending machines
1.2.1.2.8. Other
1.2.1.3. Addresses(option)
1.2.1.3.1. Car number
1.2.1.3.2. First name
1.2.1.3.3. Last name
1.2.1.3.4. ID number(kennitala)
1.2.1.3.5. Home Address
1.2.1.3.6. Postal code
1.2.1.3.7. City
1.2.1.3.8. Country
1.2.1.3.9. Email
1.2.1.3.10. Phone number(on site if possible)
1.2.1.3.11. Photo
1.2.1.4. Payment
1.2.1.4.1. Cash
1.2.1.4.2. Debit Card
1.2.1.4.3. Credit card
1.2.1.4.4. Invoice
1.2.1.4.5. Free
1.2.1.5. Receipts
1.2.1.5.1. Normal receipt
1.2.1.5.2. Sticker for the tent
1.2.1.5.3. Car label
1.2.1.5.4. Info on the RDID
1.2.1.5.4.1. Cards
1.2.1.5.4.2. Wristbands
1.2.2. Cancelations
1.2.2.1. Using receipt number
1.2.2.2. Erasing out of RFID cards
1.2.3.Viewing/statistic
1.2.3.1. Amount in cash drawer
1.2.3.2. How many on cite
1.2.3.3. Electric connections

1.2.3.3.1. What connections points are active
1.2.3.3.2. How much usage pr connection
1.2.3.3.3. How much total usage
1.2.3.4. Laundry
1.2.3.4.1. Timetable / waiting list
1.2.3.5. Dryer
1.2.3.5.1. Timetable / waiting list
1.2.3.6.
1.2.3.7. Internet
1.2.3.8.
1.2.4. Reports/day finance
1.2.4.1. End of day report
1.2.4.2. Statistic of the day
1.2.4.3. Transaction list for a period

1.3. Other cash computers

1.3.1. Internet
1.3.1.1. Reservations
1.3.2. Self service computer at site
1.3.2.1. To speed up sales
1.3.3. PDA
1.3.3.1. Smaller and simpler sales window
1.3.3.2. Using a printer with desktop computer
1.3.3.3.
1.3.4. GSM

1.4. Logout

Appendix E

HAMRAR

Tjaldsvæðið að Þórunnarstræti
Tjaldsvæðið að Hömrum
Box 135 - 602 Akureyri
Sími 461 2264 - Fax 461 2263
Kt. 430698-3469 - Vsk. nr. 58471

Kvittun/*Receipt*

Greitt til/paid until

Nafn
Name

Heimilisfang
Address

Kennitala
Social security number

Þjóðerni
Nationality

Athugið!
Engin ábyrgð er tekin á eigum tjaldbúa

Warning!
We accept no responsibility for visitors belongings

Dagsetning/*Date*

Tjaldvörður/*Camp warden*

Framlenging dvalar
Staying Extension

Nætur
Nights

Fullorðnir
Adults X

Börn
Children

Afsláttur
Discount –

Rafmagn
Electricity

SAMTALS (Total amount) kr.

Vsk. innifalinn - Vat. included

Figure 21 – Receipt used at the campsite

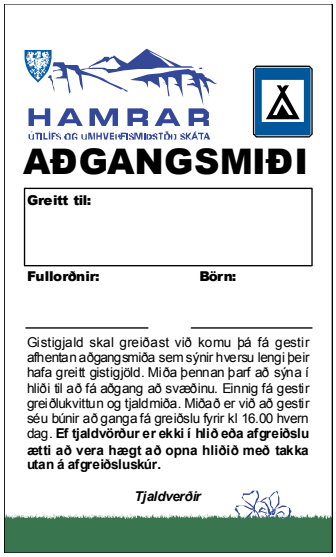


Figure 22 – Car receipt from the campsite

Figure 23 – Sticker from the campsite

Figure 24 - Database used at the campsite

Appendix F

Questions To Help Determine The Ethics of Surveillance

A. The Means
1. Harm: does the technique cause unwarranted physical or psychological harm?
2. Boundary: does the technique cross a personal boundary without permission (whether involving coercion or deception or a body, relational or spatial border)?
3. Trust: does the technique violate assumptions that are made about how personal information will be treated such as no secret recordings?
4. Personal relationships: is the tactic applied in a personal or impersonal setting?
5. Invalidity: does the technique produce invalid results?
6. Awareness: are individuals aware that personal information is being collected, who seeks it and why?
7. Consent: do individuals consent to the data collection?
8. Golden rule: would those responsible for the surveillance (both the decision to apply it and its actual application) agree to be its subjects under the conditions in which they apply it to others?
9. Minimization: does a principle of minimization apply?
10. Public decision making: was the decision to use a tactic arrived at through some public discussion and decision making process?
11. Human review: is there human review of machine generated results?
12. Right of inspection: are people aware of the findings and how they were created?
13. Right to challenge and express a grievance: are there procedures for challenging the results, or for entering alternative data or interpretations into the record?
14. Redress and sanctions: if the individual has been treated unfairly and procedures violated, are there appropriate means of redress? Are there means for discovering violations and penalties to encourage responsible surveillant behavior?
15. Adequate data stewardship and protection: can the security of the data be adequately protected?
16. Equality inequality regarding availability and application:a. is the means widely available or restricted to only the most wealthy, powerful or technologically sophisticated? b. within a setting is the tactic broadly applied to all people or only to those less powerful or unable to resist c. if there are means of resisting the provision of personal information are these equally available, or restricted to the most privileged?
17. The symbolic meaning of a method: what does the use of a method communicate more generally?
18. The creation of unwanted precedents: is it likely to create precedents that will lead to its application in undesirable ways?
19. Negative effects on surveillors and third parties: are there negative effects on those beyond the subject and if so can they be adequately mediated?
B. Uses

20. Beneficiary: does application of the tactic serve broad community goals, the goals of the object of surveillance or the personal goals of the data collector?
21. Proportionality: is there an appropriate balance between the importance of the goal and the cost of the means?
22. Alternative means: are other less costly means available?
23. Consequences of inaction: where the means are very costly, what are the consequences of taking no surveillance action?
24. Protections: are adequate steps taken to minimize costs and risk?
25. Appropriate vs. inappropriate goals: are the goals of the data collection legitimate?
26. The goodness of fit between the means and the goal: is there a clear link between the information collected and the goal sought?
27. Information used for original vs. other unrelated purposes:is the personal information used for the reasons offered for its collection and for which consent may have been given and does the data stay with the original collector, or does it migrate elsewhere?
28. Failure to share secondary gains from the information: is the personal data collected used for profit without permission from, or benefit to, the person who provided it?
29. Unfair disadvantage: is the information used in such a way as to cause unwarranted harm or disadvantage to its subject?