Abstract

Gagnasmiðjan is a part of the University of Akureyri that handles technical issues within the university, they use a system that handles data about equipment that they have and handles information about students and what equipment is lent out to students or classrooms in the university. This current system is not suitable for them, and they want a system that is designed specifically for their use, this dissertation goes through the research, design and implementation procedures that were done in designing the system that Gagnasmiðjan needs.

Introduction

This is a dissertation for a final year project done within the Information Technology department of the University of Akureyri, this dissertation covers work that has been done within one school year, expanding 8 months, and will be graded as one module towards a BSc degree in computer science.

In September 2004, Adam Bridgen, a staff member of the Information Technology department of the University of Akureyri approached the final year students in the department and told them that Gagnasmiðjan wanted a system designed and built for them and it could be a good opportunity to make it a final year project, the situation being that no final year project had been found so far, the decision was made to meet with the people of Gagnasmiðjan and discuss the issue and possibly find some solution that would work both as a final year project and a problem solution for them.
At the first meeting i met with Jóna Jónsdóttir and Viðar Sigmarsson from Gagnasmiðjan, they addressed the problem that they had and were eager to find a solution, so the choice was made that i would take on the project and start doing research on how it would be best to solve the problem.
After doing some background research it was obvious that the approach was to build a new system for them rather than making changes to an existing system, but a project like this requires some supervision and after confronting the staff of the faculty, it was the faculty's dean, Mark O'Brien that agreed to supervise the project.
The process of the project and relevant information is discussed in this report and the first thing is to introduce Gagnasmiðjan and what they do.

What is Gagnasmiðjan?

Gagnasmiðjan is a department within the University of Akureyri that handles technical aspects within the school, they manage the universities computers systems, information systems and they provide user- and technical services, Gagnasmiðjan is located at the university's facility at Sólborg and the head office is located right next to the university's library.

Gagnasmiðjan currently employees 8 people:

Adam Óskarsson: Adam is a system engineer and is the backup systems administrator within the university, he also handles a system called WebCT, which is a web based system that teachers use to get material to the students at the university.

Erlendur Steinar Friðriksson: handles the remote teaching and handles seminars that are held within the university, he also handles the operation of all remote teaching equipment.

Erlingur Harðarson: the systems administrator, he is responsible for all general system services and making sure that the universities intranet and wireless networks are working properly.

Gerardo Reynaga: a computer scientist and a backup systems administrator, he is working at temporary assignments, related to databases and programming solutions for the university.

Jóna Jónsdóttir: the head person of the communication facilities within the university also works on marketing and promotional work for the university.

Óskar Þór Vilhjálmsson: works on the help desk at Gagnasmiðjan and is the person that handles complaints, he makes sure that problems get fixed.

Stefán Jóhannsson: a specialist in information systems and is the author of a system call Stefania, an Access based database system that is used to hold information about university students, he also manages and runs the universities web pages.

Viðar Sigmarsson: a technician and a backup systems administrator, he also handles the computer facilities, making sure that the machines themselves are in working order and the also handles the technical aspect of the remote teaching equipment.

Gagnasmiðjan owns all sorts of equipment, digital still cameras, digital video cameras, projectors and voice recorders, they also provide the facility to edit both pictures and videos, have the equipment to make hardbound copies of documents.

What these people to is they maintain the university's wireless internet connections, maintain the public computers that are in the university's library, maintain the computers that are in classrooms, as well as overhead projectors, remote teaching equipment, photocopiers and other technical equipment that is used within the university. They handle all the organization of the remote teaching that is done within the university and they schedule all seminars that are held, they also manage the system that students use to check their schedules and view their personal information, and if you are in trouble with any equipment within the university, they are the people to talk to about getting it sorted out. Gagnasmiðjan has office hours from 08:00 to 16:00 on week days and they can also be contacted by telephone at 460 8070 or they can be contacted by e-mail at gagn@unak.is.

Gagnasmiðjan also lends their equipment out to both students and staff members,
The system that Gagnasmiðjan uses now is called Gegnir, it is a system that is based upon another system that is called Aleph 500.

The Aleph system

Aleph 500 is a library bases system published by Ex Libris (http://www.exlibris.co.il/), the system was originally built in 1980, and has expanded a lot since then, and as of June 2004, the system is used in 1250 libraries in 51 countries across the world.
In 2001, the Icelandic government and the controlling body of 26 towns in Iceland bought the Aleph system and a group was formed about the purchase, this group was named Landskerfi and the Icelandic version of the system bares the name Gegnir, but bear in mind that the Icelandic version of the system is no different than the English version.

The aleph system is based on and written in XML, that is done because the system has always been thought of as having two functions, being a system that is useable both as a local system and having a web based interface that is accessible from where ever there is an internet connection.
the system used an Oracle database and incorporates standards such as Z39.50, refers to the international standards, ISO 23950, which is a technical data format and transmission protocol specifications, Unicode, which is a single character encoding scheme that allows characters from Western European, Eastern European, Cyrillic, Greek, Arabic, Hebrew, Chinese, Japanese, Korean, Thai, Urdu, Hindi and all other major world languages, living and dead, to be encoded in a single character set, and MARC 21, acronym for Machine, which is a protocol by which computers exchange, use and interpret bibliographic information

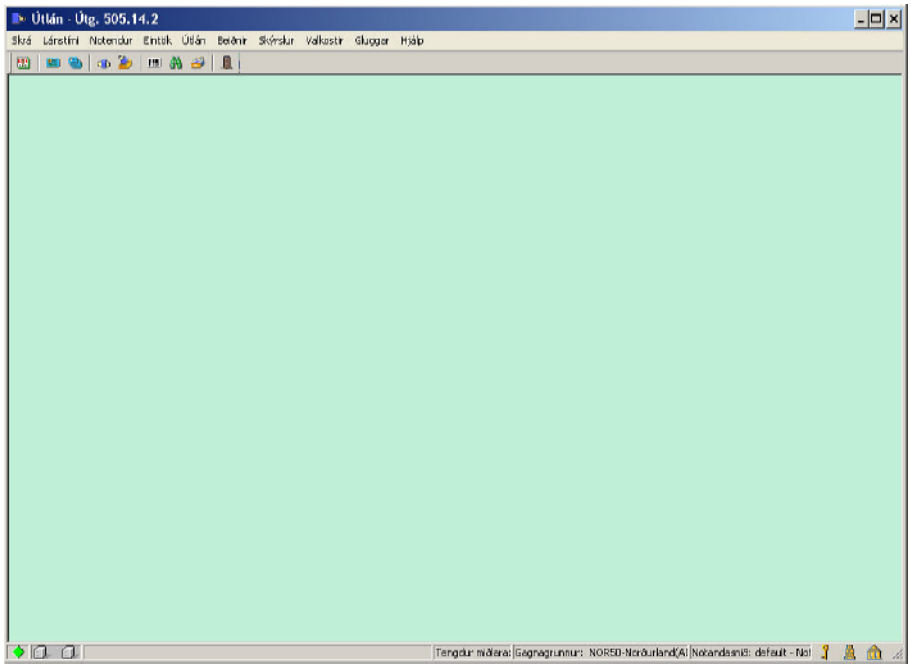
What the system does it that it incorporates all aspects of running a library, it holds information about users in a database, user history, has the ability to make lists, handling preorders, ordering material from vendors, but the most important part of the system it's ability to combine the handling of all the different material, books, magazines, CD's and videos, and also making this function available online too.

Structure

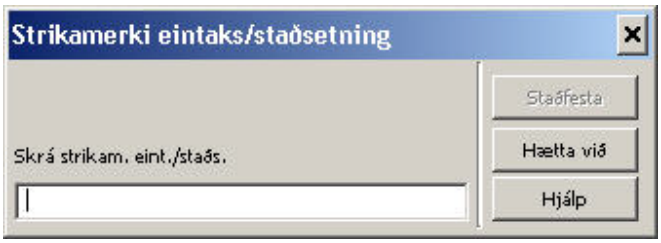
The structure of the system is divided into two parts, local and web based version and each will be discussed separately, but there is one thing that cannot be issued and that is the database structure. A request was made to the vendor of the system to gain a temporary access to the database for educational purposes, look at the structure and see how it works, but the request was denied on the ground that the system is owned by so many that an acceptance would take too much time and there was a lot of personal information stored in the database, ant that they simply would not take the risk of revealing the information, so now database structure will be discussed in this section.

Local version:

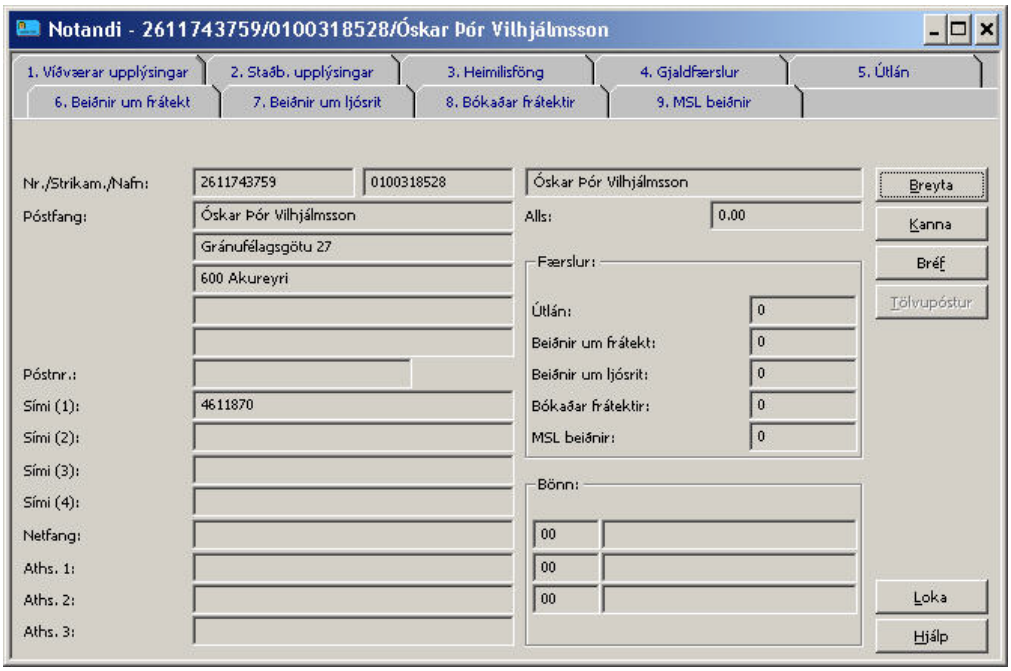
designed as a windows based window environment and acts and behaves in a similar manner that windows programs do, the interface is standard, with a file menu that has basic functions, quit, edit, preferences, help and incorporates other features that the system has and the most commonly used functions are incorporated into buttons on the screen but the interface is not all that simple, but users learn to use it quickly.



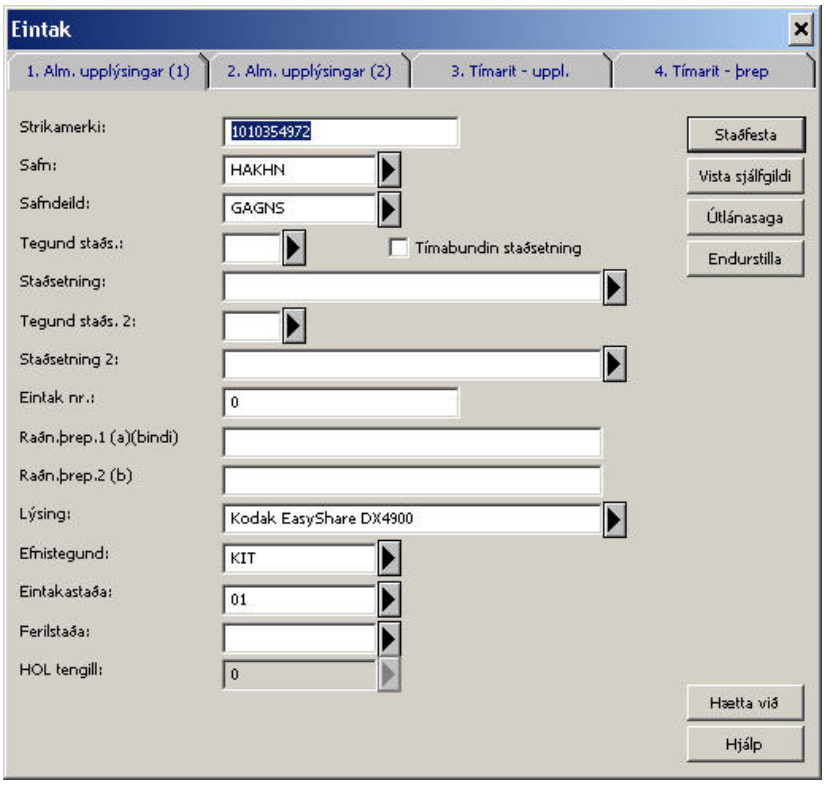
The main window of the Aleph system



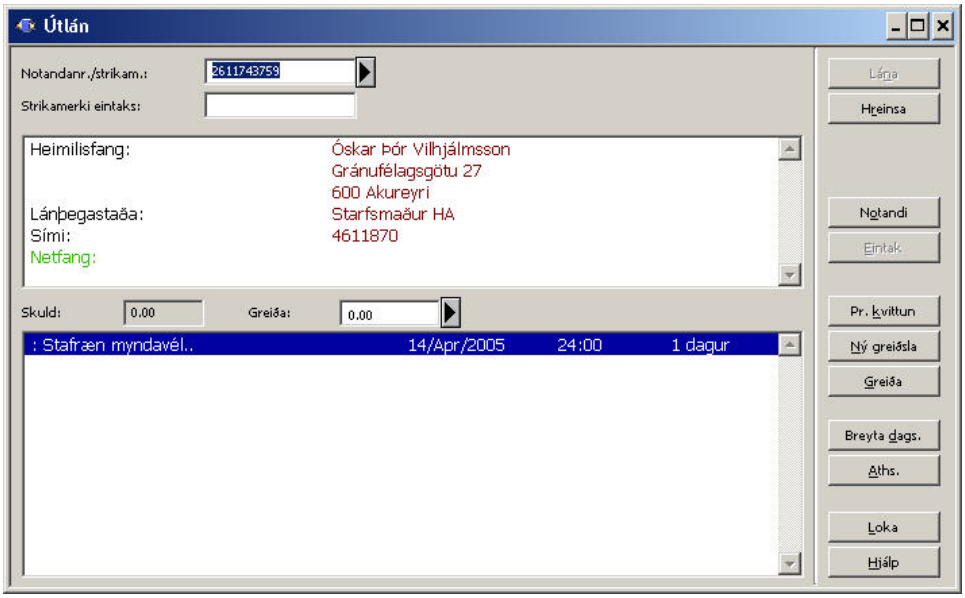
Search window of the Aleph system



User information window from the Aleph system



Device information window from the Aleph system

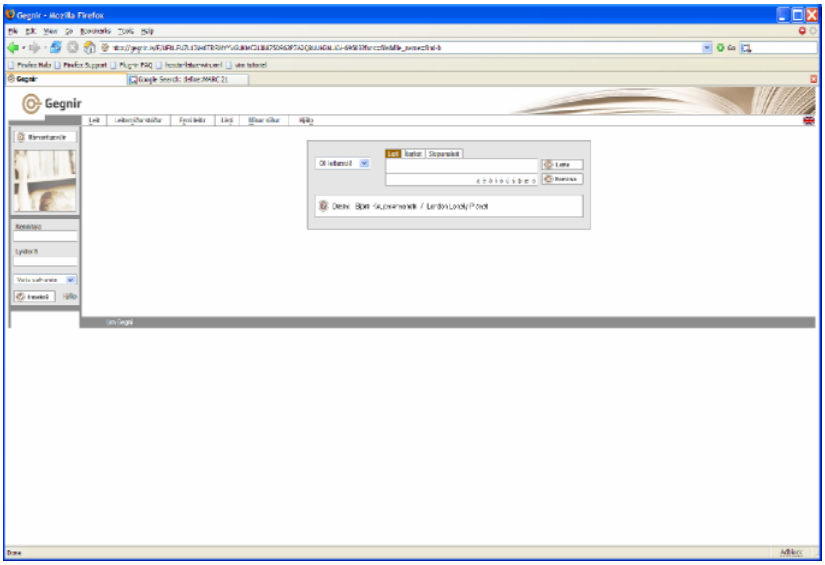


Borrow equipment window from the Aleph system

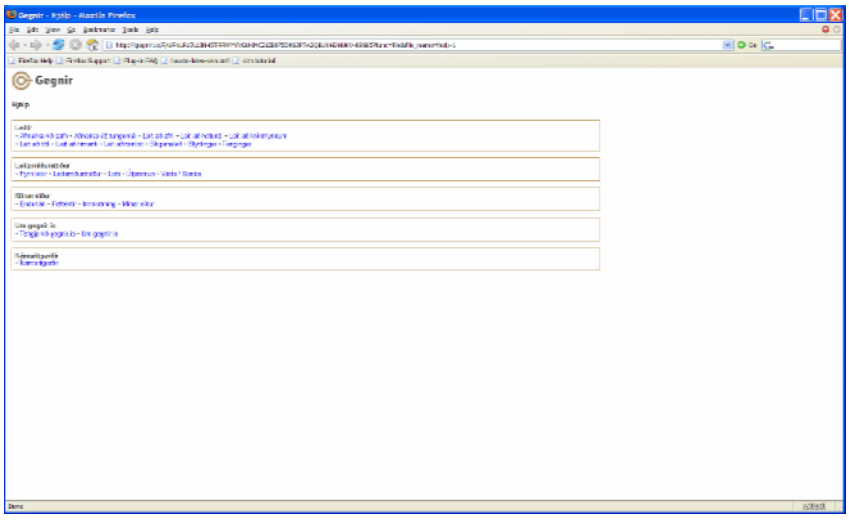
Aðalgeir Sævar Óskarsson

Online version:

Online version is designed as a webpage with clickable links, pull down menus and search fields, users are prompted with a sign in field at the start page and there is a menu structure that has the most common fields, search, preferences and help and the interface is very simple.



Screenshot from the main window in the online version



Screenshot from the help section in the online version

Reports

The system like this is not very useful if it cannot produce any reports, here is a list of the reports that the system can produce

Items overdue

Debts

Pre paid orders

Request for items

Pre orders

List of users

Orders to vendors

Status of orders

Lists of popular items

Lists of items on one day pass

Lost items

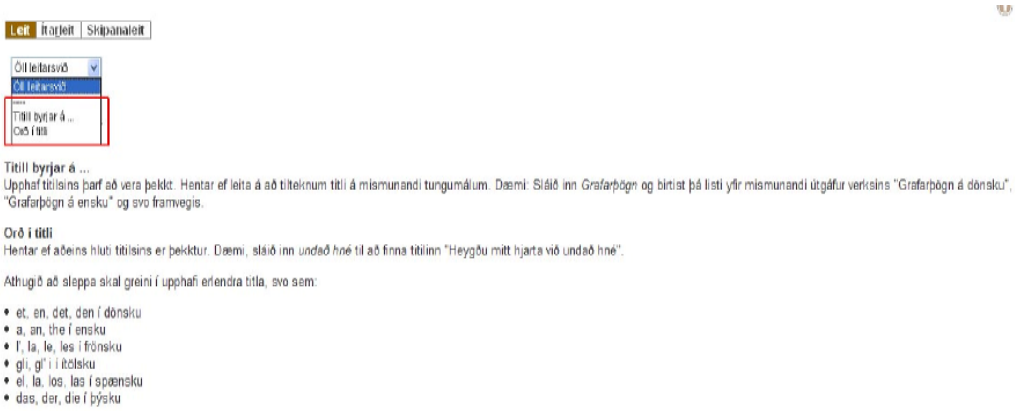
List of vendors

List of all items in the library

Statistics of users and dates

Manuals and help system

Manuals for the system are good and detailed both for the local and online system. In the local system the access to the manual is under the Help menu and there are examples on how to perform all basic functions of the system and examples are given, and the online version has a similar system but it is in the form of a website, the site is divided into sections for whether you are looking for help with searching in the system, displaying pages, changing personal preferences or want information about the system.
Here is an example of the help system in the online version



System requirements

For window version:

Processor: Pentium 4, 2, 4 GHz clock speed

Memory: 512 MB minimum

Hard drive: 40 GB of hard disk space

Operating system: Windows 2000 or newer

Other: screen size of at least 15”, mouse and keyboard, sound card and speakers are optional but are recommended

For web version:

No hardware requirements mentioned, but internet explorer of Netscape, version or above is the minimum requirements, screen resolution of 1024x768 pixels, mouse and keyboard, sound card and speakers are optional but are recommended

Evaluation

The problems that Gagnasmiðjan has can be categorized in two ways

System control:

When dealing with a system like this, changes has to be made frequently, and since Gagnasmiðjan has no real control over the system that makes it hard.
The system is in control of Landskerfi and all changes made to the system are made in Reykjavík where Landskerfi have their main office.
Gagnasmiðjan has to add information to the system quite frequently, and not having control of the system is hard, the information that is added to the database is:
Student information, new student profile or editing current profile
Adding device information, a new device is added or information about an older device is changed.
Making these sorts of changes takes time, a formal request has to be made and sent to the head office at Landskerfi for the change to be made, that takes at least one working day which means that a new device cannot be rented out, and a new student has to wait for at least that one day to get the service that he wants.

Internal problems:

When using a system for a purpose it wasn't designed for, there are bound to be some problems, and that is what Gagnasmiðjan has encountered.
The Aleph system is library based, and data input is designed as a unified schema for books, but when trying to use the fields to input data that describes equipment, things go wrong.

Each device is registered as having a unique serial number, color, description, category and name, and trying to fit this into a schema where you have other database entries, and there is no good way of fitting the information into any table so that it makes sense.

Because of this database problem, other thins in the system do now function properly, trying to make reports for instance is a mess, and there is no good way of making them.
The reports that are missing from the system are:
Device history, seeing who has borrowed each device
User history, see what devices each uses has borrowed
Backtracking history, there is no way of seeing all history of items, there is a time limit of how long history is kept, and the limit is to short, only one month.

Why not change the current system?

The current system is good for what it is designed for, otherwise it would not be used so widely, but it does not function for Gagnasmiðjan, but why is the system not adapted so that they can make use of it?
There are 2 main reasons for it:

Overhead: the system is in control of the government and many others, dealing with the government would take a long time and would be costly, the system is also big and complex and modifying it to fit the needs would take time, and taking such a large system and make changes to it is costly and it is something that is just not acceptable, after all, Gagnasmiðjan is only one organization that has special needs and vendors for systems at this scale just do not bother putting in the effort for such a little organization.

Budget: Gagnasmiðjan does not have the resources to go into changing the current system, besides the before mentioned reason that the vendors do not want to put in the effort, the change to the system would be very expensive, and they do not have the kind of budget that could cover such a change, and spending millions on a project like that is not acceptable.

Proposed new system

It is proposed that a new system will be built that will be developed specially for Gagnasmiðjan, it will be developed according to their desired specification and build according to their standards. This section will go through the system design and the steps that are necessary to take when designing and implementing a new system.

System design

During the systems development process, the employees of Gagnasmiðjan have analyzed and pointed out the desired specifications of the system, normal features like exit and minimize are not presented here, these specifications are improvements from the previous system and some of these are functions that the previous system was not able to perform.

System functions

The functions that the new system will be able to perform are all functions that are needed in the old system but are not available, the functions in the Aleph system have already been discussed but the new system will have the basic functions from the older system plus these functions:

Create users in the system
Control privileges of users
Have access to universities student lists
Being able to register devices within the system
Being able to create lists of:
- All current students
- Student's usage history
- All current devices
- What devices are available/not available?
- Device usage history

Being able to print out all lists
Being able to set time limits when students book devices
Notify automatically when a device becomes overdue
Issue a late return fee to the users account if a user does not return a device in time

Optional specifications for future development:

If the system is a success, Gagnasmiðjan has some ideas for future development, some of these ideas easy to implement and some are hard, but in an agreement with Gagnasmiðjan, these are the features that could be developed in the future.

Web based interface
Send out e-mail notifications when a device is overdue
Book devices over the internet

GUI

What all larger systems must have is a Graphical User Interface (GUI) that is appealing to the users and is easy to use, but incorporating the right interface design is more that just applying buttons and text areas where they fit on the screen, it must be taken into effect that users are not all experts and some even have very limited computer skills, that means interfaces have to be designed with that in mind.

Interface design

Users have some general idea of what they want when it comes to user interfaces, they are used to some things, like having a "File" menu, and having a "Quit" option under the File menu, things that are in common in most all interfaces, take for instance Microsoft software, they have almost the same general features in all their products, if you can work with one Microsoft product, you can basically figure out all of the because of the integrated design and standards. When it comes to these design things, you follow some standards and these are:

- Having a menu at the top that goes from left to right
- Having at least "File", "Edit", and "Help" options
- When creating new things, the option for that is placed under "File"
- The "Quit" option is also placed under "File"
- When changes are gone, the options for that are set under "Edit"
- Also if there is a possibility to change preferences, that is also set under "Edit"
- "Instructions" and "About" options are set under the "Help" menu
- controls in the main window are usually set on the right side of the screen

When validating an interface, there are some things that must be kept in mind, users do not want to search for things, they want interfaces to be accessible and easy to use, placing things in such a way that it takes more than 5 mouse clicks to access is not acceptable, and the design of this interface was done with that in mind

Having some experience of design interfaces, there are some approaches that are better than others, the approach taken in this project was heuristic evaluation.
What that means is that the interface is designed and the designer is the one that evaluates the interface, according to some standards that he has set, in this case following the standard windows feel of having a window environment with a standard file menu, there are other design methods available, like doing a user based evaluation, but that takes incredible amount of time and is expensive, going with the heuristic technique saves time and money.

Database design

Designing a database is probably the most important thing when doing a system like this, without a good database the system will not function properly.
What makes a good database? Good design of course, but to achieve a good design there has to be some problem analysis.
This system has basically two things, users and equipment, when it comes to users, they have certain things, they all have a name and Kennitala, they have a standard university issued e-mail address and most of the have telephones, and these are the key features in the student database table
What all equipment has is a serial number, a description, color, name and they are categorized, and those are the key features in the equipment database table.
There is still a big thing missing, a table that stores the history, who borrowed which peace of equipment, when making a table for that, the input is slightly different from the who other tables, you have to input some identifier for a user, some identifier for equipment, and the date, to be able to do that, relationships have to be assigned.
What makes a person special is the Kennitala, every person in Iceland has a unique Kennitala, so that is by far the best user identifier, and the same thing goes for equipment, all equipment have serial numbers, and no piece of equipment has the same serial number, so when data is added to the history table, a pointer is added that points to a persons Kennitala, thereby knowing what person it is, and another pointer is added that points to a serial number of a device, thereby knowing what device it is, and the date is added so that the loan history can be backtracked and searched properly.

Implementation

this segment will discuss the technical side of the project, what software is used for development, what programming language is used, what kind of system is used for developing the code, what kind of system it will be running on after development and some background information about the vendors of all these systems and utilities that are used during the process. This report will also include as an appendix, use cases of how the system functionality will be and how the basic functions of the system will be utilized.

When doing a project like this, it is essential to choose the software and systems that suit the job. In this case, since it is a final student project, the possibilities are not endless, this is a non budged system so the tools have to be free and the utilities have to match the system that is currently in use, because buying the best possible hardware is not an option due to the same reason.
The hardware used for development is owned by the developer and the university does not have to supply any hardware or software for the job.
The systems and tools that have been chosen are listed in the next section of this document, the section is split into sub sections that cover each system and/or tool, and the sections give some background information about each vendor, technical information about the system or tool, and a reason for choosing that tool.
As an appendix, use cases are presented that represent the usability of the system, this is done to better understand what the user wants out of the system and how it is best to build the system according to the users specification, also there is a section with the appendix that explains how the basic use case is set up.

Technical information

System used for development

The system used for software development is a Dell laptop computer and a pc desktop system, specifications are as follows:

Laptop

Model: Inspiron 8600

Processor: Intel Pentium Mobile, type 725, 1, 7 GHz clock speed, 400 MHz system bus

RAM: 1 gigabyte DDR (double density ram)

Hard drive: 80 gigabyte

Operating system: Windows XP professional with service pack 2

Other features: DVD burner, built in wireless card, built in modem, built in 10/100 Ethernet card, USB 2, 17" display.

Desktop

Model:

Processor: Intel Pentium hyper threading, 3, 0 GHz clock speed, 800 MHz system bus

RAM: 1 gigabyte DDR (double density ram)

Hard drive: 3, total storage space of 560 gigabytes

Operating system: Windows XP professional with service pack 2

Other features: DVD burned, 10/100/1000 Ethernet card, USB 2

Java

About SUN Microsystems

SUN Microsystems is a silicon valley based company, founded by students from Stanford University in Palo Alto, California, The company name, "SUN", stood for Stanford University Network. The company started in 1982 and went public in 1986; the founders were Vinod Khosla, Scott McNealy, Bill Joi and Andy Bechtolsheim. The company was originally only working with UNIX and developing network based computing programs, but they expanded their work and developed JAVA, and they also had a lot to do with the creation of XML specification

About Java

The java platform was developed on the early 1990s and was developed with the objective of allowing programs to function regardless of the device they were running on, hence the popular java slogan "Write once, run everywhere".

The java platform is in essence three major parts, the programming language, the java virtual machine and the java API (application programming interface), which is an extensive library of routines that the programming language provides, and java is also an object oriented language, in which a software system is modeled as a set of objects interact with each other. Since 1995, java has been increasing its popularity and is now one of the most used programming languages.

The Java virtual Machine

The virtual machine is what makes Java so easy to use on all systems.

When java code is developed, it gets compiled into what is called Java byte code, this java byte code by itself is practically useless because it cannot be run on any machine directly.

What the java virtual machine does is that it interprets java byte code into machine code, so that the machine can understand and run the program.

The java virtual machine is in itself a computer within a computer! The virtual machine is available for all platforms and that is the reason why Java is so popular, many programs are platform specific and have to be rewritten of be suitable for other systems, because of this java byte code compilation, the program is only written once but can be used on all

systems, as long as they have the virtual machine installed for that system. Another result of this is that all java programs appear and act exactly the same on all systems

Reasons for choosing Java

The reasons for choosing java are the platform independence, it is a good thing to not have to worry about what system to run your program on, with java you just write the program and then get the virtual machine for the platform you are using.

Other reasons for choosing Java is that the language is free of charge, no leases are required, and because of that, java is very popular and lots of code samples and help can be found on the internet, SUN also has an extensive website that has all information that a developer can possibly need while programming, including a very extensive API that covers all aspects of the programming language.

Borland J Builder

About Borland Inc.

Borland international inc. was founded in 1983 in Scotts valley, California by a programmer call Philippe Kahn, his goal vas to develop and market innovative and high performance programming languages and applications software based for the ever growing personal computer market.

Since 1983, things have changed quite a bit, Borland has been in the process of simplifying the process of developing world class software and as of 2005 the company has over 1300 employees worldwide, has expanded to 29 countries and has all year round support services for companies and individuals, all across the world.

The Borland Company is in a way a pioneer in the software development community for they are the ones that developed one of the first development environments designed for the PC, the Turbo Pascal.

The official website for Borland is: http://www.borland.com

The jBuilder.

The jBuilder was created and first announced in 1997, it is a highly capable tool for java development and it has won several consecutive awards as the most powerful professional Java programming tool in the world.

The jBuilder has very nice features, such as:

A method list that pops up when appropriate
Error checking
Debugging tools
Can run java web applets
Comes with the Java API specification built in
Has a current version systems tool
Automatically formats code layout
Auto completion of methods when typed in

And many other features that are useful when developing. This developer is one of, or even the best java developer available, and the main reason for picking this developer in front of others is that it is available in a student edition that is free of charge

Minimum system requirements

Processor: Pentium III or higher

Memory: 256 MB minimum, 512 MB recommended

Hard drive: 700 MB or hard disk space minimum

Operating system: Windows XP recommended

Other: high resolution monitor (1024x768 minimum), mouse or other pointing device

Microsoft Access

Very brief history of Microsoft

Microsoft was formed in 1975 in Albuquerque, New Mexico by 2 school buddies named Bill Gates and Paul Allen, the name Microsoft is something that Bill Gates came up with when discussing “microcomputer software”, the company wrote some unsuccessful software in the past but Microsoft's big break was when they landed a deal to write the operating system for the I.B.M. PC, released in 1981

Microsoft is today one of the most powerful and successful computing companies in the world, which is a big improvement from their original company, Traf-O-Data, which made car counters for highway departments. In just 3 years, the company achieved revenues of $500,000 and employed 15 people and by 1992 the numbers had increased to 2.8 billion dollars in revenue (50% of which are from exports), and over 10,000 employees, figures of revenue and employee numbers have not been released for the current year.

About Access

Access is a database management system that combines the Jet relational database engine with a graphical interface, intended to make it possible for new and un-experienced programmers to build front ends to databases, and one of the benefits of Access for a programmer is that that is compatible with SQL and queries may be viewed and edited as SQL statements.

Reasons for choosing Access

There are databases available that could have been more appropriate for the job, but the reason for choosing Access was that Gagnasmiðjan chose to use it, a system they are using now, called Stefanía, is using the Access database, and this new system has to interact with the Stefania system, and the choice they made was to use the same database, since the university has a lease from Microsoft and owns a copy of Access, also Access is a simple database that is easy to maintain, and for a database of this size, access is a good choice.

Minimum system requirements

Computer and processor: personal computer with an Intel Pentium 233 MHz or faster **processor:** Pentium 3 or higher recommended

Memory: 128 MB of RAM or greater

Hard drive: 180 MB of available hard-disk space and optional installation cache files Requires an additional 200 MB of available hard-disk space, and these cache files are recommended

Operating system: Microsoft windows 2000 with service pack 3 (SP3), windows XP, or later

Internet connection: Internet functionality requires dial-up or broadband Internet access

Stefanía

Stefanía is an Access based database system that manages student records, the system was build by Stefán Jóhannsson, an employee at the University of Akureyri,
Stefanía is used by the university faculty to keep student records.
Stefanía is both a local and a web based system, the local system being for the faculty and the web based system used by the students to check their time schedules, rooms, student lists and their personal profile.

Reasons for choosing Stefanía

Stefania is build for the University, it is used by Gagnasmiðjan and the most logical step was to implement the current database instead of build a new one, the student records are already in the database, so the work of adding all student records into a new database is also eliminated.

Database connectivity

The database connection is established by using the java database connection function, java has build in database connection methods that work just as well as third party database drivers, and the database driver used by the operating system is a Microsoft Access mdb driver that is supplied specially for Microsoft Access.

Evaluation

Evaluation is hard for an incomplete system, this system has its two major components working, the database and the interface, although bits and pieces need developing. The evaluation will take place in two phases, Gagnasmiðjan will be given a local copy of the software, a copy that will have the database and the software stored on one computer and they will be asked to evaluate it and point out possible bugs, the second phase will take place after the bug fixing and then the system will be installed and taken into use, then real life situation and use will determine how the system performs, if the system performs well, the older system will be replaces by the new system.

Problems encountered

Access

Working with the Access database means that a version of Access has to be available to work with, the university has a lease for Access, but that is only for use within the University, so getting Access was a bit of a problem.
The action was taken to send an e-mail to Microsoft with the request of getting a student evaluation edition of the Access software for use in the project, the reply from them was negative, they do not hand out any student evaluation editions of their products.
The solution to the problem was to buy the student/teacher edition of the Microsoft office package, that has the Access database in it and the project could go on as planned.

Gathering information

When doing research for a project like this, some information gathering takes place, and during the project some database and software information had to be gathered. The software marked is very rough, and companies try to keep their software well protected, and that makes information gathering hard. E-mails were sent out to many companies, some gave out information, Landskerfi were very helpful and provided information for the Gegnir system, some refused to give out information or services, like Microsoft and Ex libris that build the Aleph system, and some just did not take the time to answer e-mails even after several reminders, like Borland.

The Stefania system

After spending about 3 weeks on trying to connect to the Stefanía database, it was obvious that something was not working right, after trying nearly all known methods of database connection the answer had to be that something was not present.
Gagnasmiðjan said that the connection should be possible and gave information about the database and the server, but after having a talk with faculty member Adam Bridgen, it was obvious that the connection with Stefania was not going to happen, the way the

system is implemented it that the Stefania is used locally within the University's intranet, and not accessible from the outside.
The solution was to change the plans and create a local database for the G.E.M.S system with only two days until the system was due, it took some code changing and altering, but it worked and the system has a function database.

Current status of system

After the setback of knowing that the main database design would not work, the design and implementation changed a bit, but the current status is that the system has a functional interface, a main window with functioning buttons and text boxes, a working file menu with all menu items functioning and short cut keys defined for the main keys, and there is a help, and about window working also.
The database connections is working also, the system prints out a list from one table just to show that the database works, but there was not enough time to implement the interface to the database connection, creating tables and adding data to tables was tried and it works properly, so all the functionality is in place, the only thing that has to be done is to implement functions in the interface to the database.

Conclusion

The implementation of this project is that a system needs to be built and a dissertation be written, there is no clause of delivering a complete system with full functionality.
The system at hand is not complete, the major components of the system are functioning, but the system needs to be developed more, the aim is to complete the system so that Gagnasmiðjan can utilize the system in the upcoming school year and finally have a system that has the functionality that they need.

Resources

Online: http://microsoft.com

Accessed 28/02/2005

Online: http://borland.com

Accessed 12/02/2005

Online: http://www.microsoft.com/office/access/prodinfo/sysreq.mspx

Accessed 15/02/2005

Online: http://www.wikipedia.com

Accessed 15/02/2005

Online: http://www.google.com

Accessed: frequently

Online: http://java.sun.com

Accessed: 15/02/2005

Online: http://www.javacoffeebreak.com/articles/jdbc/

Accessed: 13/04/2005

Online: http://www.landskerfi.is

Accessed: 20/03/2005

Online: http://www.gegnir.is

Accessed: 20/03/2005

Online: http://www.exlibris.co.il

Accessed 20/03/2005

Bishop, Judith. Java Gently, 3rd edition, London, Pearson education, 2001

Heller,Philip; Roberts, Simon; Zukowski, John; Vanhelsuwé, Laurence; Holzner, Steven Java 2 Complete, Alameda, Sybex, 1999