E-Patient Telemedicine Project

Iván Febles Trujillo
Thesis written by Iván Febles Trujillo for the Computer Science’s Department of Växjö University.
Supervised by Mathias Hedenborg

Växjö (Sweden), August 2006
Abstract:

The aim of this thesis is to develop a Java web application with MySql that provides a useful tool for those people that have to handle with diabetes in their normal life.

The idea is that both patients and doctors will use this prototype for exchanging useful information and thus reduce some of the multiple inconvenient that this disease has.

The solution proposed in this project can be generalized to cover other aspects of the communication between the patient and the doctor.

Acknowledgements:

First I would like to thank to my supervisor Mathias Hedenborg for his comprehension and support on this project.

I would also like to thank Doctor Lars Nilsson (General Practioner, Teleborg Health Centre, Växjö) for his help and his patience during our interviews through the development of this project.

Finally, I would like to thank my parents for supporting me throughout my studies and helping me during my stay in Sweden, where I have developed this project.
# Table of content:

1. Problem definition  
2. Introduction  
   2.1. Background  
   2.2. Purpose  
   2.3. Limitations  
   2.4. Structure of the report  
   2.5. Telemedicine  
   2.6. Previous work  
3. System overview  
   3.1. Application overview  
   3.2. Software development method  
   3.3. Use cases  
      3.3.1. Use case 1  
      3.3.2. Use case 2  
      3.3.3. Use case 3  
4. Requirements  
   4.1. Functional requirements  
   4.2. Non-functional requirements  
5. Proposed solution  
   5.1. Application design  
   5.2. Database design  
   5.3. Security issues  
6. Interfaces  
   6.1. Public interface  
   6.2. Private interface  
   6.3. Special users  
7. Conclusion  
8. Future Work  
9. References  

## Appendix A

A.1 NetBeans  
A.2 MySql Administrator  
A.3 MySql Query Browser  

## Appendix B

B.1 Public interface  
B.2 Private interface  
B.3 Doctor’s interface
1. Problem definition

I live in the Canary Islands, a group of 7 islands (that belong to Spain) in the Atlantic Ocean. The two main islands (Tenerife and Gran Canaria) are the ones with the big hospitals and most of the specialists, and the other islands have small medical centres (now they are growing and some of them have incoming hospitals) with limited resources.

Sometimes, when somebody who lives in a small island needs to visit a specific doctor (heart’s specialist, surgeon, etc); they have to go to one of the main islands. Thus they have to spend money in the plane/boat tickets, and when these trips are usual, this cost a lot of money.

But not only is the money a problem. When a patient has a job and has to travel to another island, they need to ask for some free days in their jobs, and that is not always possible. The same happens for those people with children, and they are forced to let them with their families. Thus, these are only some of the large list of inconvenient that this kind of illness can cause.

When we talk about diabetes, sometimes these visits are only for regular controls, like checking that the glucose levels are not out of range. In such situations, these trips are useless. Instead of that, we can send the information in some other way, and there is where this application comes into scene.

This project proposes to use internet to reduce these problems, and to allow patients to use a web application to send/receive some information, and avoid or at least reduce the number of times per year that a patient must travel to one of the main islands.

By using this application, the patients can get some important information about diabetes (what diabetes is, the different types of diabetes, read some FAQ, etc.). They can also send their own information to the doctor (Glucose level in blood, weight, height, arterial pressure, etc.). Doctors will use this information to control the evolutions of their patients.

Whenever a doctor finds one of the measurements out of the normal range, he can send an email to the patient, telling him to come to the hospital for a check, or to take some specific medicine.

This way, the patients do not need to go to the main islands so often, and then they can reduce these annoying and expensive trips.

This web application can be also be useful in Sweden, where distances are longer and sometimes you need some hours travelling by car to go to the hospital. It can be also useful in some cases like when a patient have some difficulties in going to the hospital so often (handicap people, elder people who have nobody to take care of them, etc.).

Summarizing, the main goal of this project is to provide a useful tool that the users (patients, doctors and internet surfers) can utilize to reduce some of the problems that diabetes causes.

Since different kind of people will use this tool, it must be simple to use and to understand.

This application must allow the interaction between the patient and the doctor.

Since we are working with databases, the security is an important factor to take into consideration. The application must be reliable and not allow unauthorized access to the data.

A primary goal for the doctor is to be able to view the personal and medical information of their patients. They must be able to read all the forms that the patient has sent or to be able to read the appointment requests of the patients.
Another goal for the doctors is to be able to use the application to contact the patient (via email).

In the case of the patients, the goal is to receive some medical information about diabetes, and to be able to send information to the doctor, and thus avoid the long trips to the hospital that they have to do more often than they would like to.
2. **Introduction**

2.1. **Background**

In these days that internet and new technologies have become so important and they are so widely spread around the world, we have to take advantage of the possibilities that they offer to us. Nowadays we are able to turn on the lights of our house by clapping our hands, to open our car just putting our finger over the lock or to do the shopping by using our fridge because it says that we are running out of milk. Internet is a powerful tool and in this project, I will try to make a good use of it.

In this thesis I will explain the idea that I have about how to use these new technologies to make our life easier and more comfortable. This project consists in a web application for people with diabetes. This application can be used either to receive educational information about diabetes or to send some medical information into a database. The doctors will use this web application to check if their patients have any problem with their disease.

In this case, we talk about diabetes, but this application can be used in many different fields of the Telemedicine world. It can be easily applicable to other diseases, or it can be modified to support videoconference with the doctor or to send some extra information like x-ray pictures.

One of the advantages of new technologies and computers is that they are always evolving, and the range of applications grows exponentially, so I think that this application can be a good beginning.

2.2. **Purpose**

As I said previously, patients with diabetes have many difficulties because they have to visit the hospital quite often for checking their glucose level in blood, their blood pressure, etc.

The purpose of this project is to help them reducing the number of these trips by using a simple web application.

This web application is also useful for doctors, who will have the chance to access to the patient’s information 24 hours a day, 7 days a week, and then, they will not be limited to the office hours in the hospital.

This project offers then the chance to send some specific information (glucose values, blood pressure, ask for an appointment, etc.) to the doctor.

This information can be used by the doctor to determine if there is something wrong and then, communicate the patient that he has to go to the hospital.

Another characteristic of this project is that it gives the chance to the patients (and to the doctors as well) to interact with the system from any computer (with internet connection) at any time, not only during the office hours.

2.3. **Limitations**

This application can be used by everybody. The only thing that a patient needs is to connect to internet for using the website.

Once they are in the main page, they can get the general information without logging in or they can access to the private part of this application (with just a few requirements).

For accessing to these options a patient must have a private username and a password that must be previously given by the doctor at the hospital.
There are some things that we should also take into consideration:

- Some people (elder people, handicap people, etc.) are not so comfortable with new technologies, so unless they have some help for using the application, they will not be able to understand how it works.
- A computer with internet connection is necessary, so the patient will need to use their own computer (at home or at work) or use a public one.
- Even when this application allows the user to send some information to the doctor and thus reduce the visits to the hospital, for some other specific features, they will have to go to the hospital.

This application is just a prototype, and it does not solve all the problems that diabetes can cause. However, the aim of this project is, at least, to reduce them. The other kind of limitations (technological ones) will not be solved in this project.

2.4. Structure of the report

In this section I will describe the structure of the report. After the first chapters, where the reader can get an overview of the problem treated in this project, we can find a chapter that offers the reader an overview of the system and how it works.

Then, I will define the requirements of the system (functional and non-functional).

After that I will explain the solution proposed in this project. I will talk about the application, its design, the structure of the database, etc.; I will also explain the different interfaces that we can find depending of the type of user that we are, etc.

Once the reader is familiarized with the application’s appearance and how it works, he will find another chapter where the conclusions are. In that chapter I make a reflection of the project and the results obtained.

Finally we can find another chapter called future work, with possible modifications to the project, or improvements to extend this application to other fields.

2.5. Telemedicine

Before I start with the definition of the system, I would like to explain what Telemedicine is and why this project is related to it.

I founded this definition in the Wikipedia, the free encyclopedia of internet. In this page, they define the term Telemedicine.

The term Telemedicine is the delivery of medicine at a distance. The term is composed of the Greek word τελε (tele) meaning 'far', and medicine. Telemedicine may be as simple as two health professionals discussing a case over the telephone, or as complex as using satellite technology and video-conferencing equipment to conduct a real-time consultation between medical specialists in two different countries.

Telemedicine generally refers to the use of communications and information technologies for the delivery of clinical care.

Care at a distance (also called in absentia care), is an old practice which was often conducted via post; there has been a long and successful history of in absentia health care, which - thanks to modern communication technology - has metamorphosed into what we know as modern telemedicine.

As you can see in the definition, Telemedicine not only means video, it can be applied to every technology used for the delivery of clinical care.
2.6. Previous work

A few years ago I participated in an exchange between University of La Laguna and University of Missouri, Columbia.

During this exchange, I worked in a project helping to develop a call centre for people with diabetes and there is where the idea of this project comes from.

This project was quite big to be developed in a bachelor level project, so we decided to modify it and use a web application instead of a call centre.

I have been searching for similar projects in the net, and I have found some interesting ones, but the decisions I have taken for designing this application, come from the previous idea that I already had.

I have found some telemedicine projects about diabetes, like the one of WCCHC, but even when they are quite interesting, they offer a slightly different approach to this project.

As it says in the website of WCCHC, this innovative project enables patients with diabetes to receive computers connected to home blood glucose monitoring devices. Then, a dedicated internet site allows the doctor to monitor daily blood glucose information, and allows the patient to access his own glucose patterns, along with educational tips in the website.

This project involves that some specific blood glucose monitoring devices are installed in the patient’s place, and this is not the idea of my project. This way the patient needs to be at home to use the application, and this is a limitation that I try to avoid with my project.

Another interesting project that I founded is IDEATel (Informatics for Diabetes Education And Telemedicine).

The IDEATel project will establish web-based computing and telecommunications networks in both urban and rural economically disadvantaged areas within New York State.

This new technology allows the patients:

- To see and talk with nurses specialized in diabetes
- Check their sugar level in blood and arterial pressure
- Learn more about his health care

However, this project uses a webcam to allow visual communication between the patient and the nurses, and this is not one of the intended features of my project.

Furthermore, this project only offers a limited variety of parameters (sugar level in blood and arterial pressure) and I wanted to offer a wider range, so this solution is not useful for my purpose, at least not without some modifications.

To have a deeper knowledge of this project, see references, where the link of this project is.

I also found a paper written for the Uppsala University, about “Telemedicine and patient data management”. This paper explains a solution they have implemented to allow patients with diabetes and health care personnel to communicate.

This solution differs from the one I propose in different aspect. The public part of this application has a forum, where the users can discuss and exchange their experiences of Diabetes. The private part of this application is based in an email system between the patients and the doctors / nurses. Some of these options are not applicable to my solution, so I have decided to create my own one.
Once I have explained the different approaches I have found and why they are not suitable for my project, in the next chapter I will start explaining the solution that I propose in this project.
3. System overview

This chapter offers a general overview of the system and it works. It will not go deep into details because they will be explained in chapter 4 and chapter 5.

It will start giving an overview of the web application, and how different types of user will use it.

Then I will explain the software development method used in the development of this project and finally I will show some use cases that will provide a better understanding of the possible applications for this project.

3.1 Application overview

This application is very easy to understand. On one end of the internet we have a normal user that wants to use the application to get some information about diabetes and for sending some information to the doctor. If the user is a doctor, he will not store information in the database, he will only read the information that the patient has sent, and then he will have the chance to order a check or some specific test to the patient.

On the other end we have a database that the web application will use to store the information (from the user) and to show it to the doctor.

The picture 3.1 gives a graphical idea of how the application works.

![Picture 3.1, system overview](image)

The web application is the centre of the whole project. It will give the chance to patients and doctor to interact by just using a simple database to store the information.

A deeper explanation will be given in chapter 5.
3.2 Software development method

To develop this application I have used Extreme Programming (XP). XP is an agile software development methodology that places a higher value on adaptability than on predictability.

I have used this methodology because it is a good approach when requirements can change at any point during the project development.

This is a more realistic and better approach than attempting to define all the requirements at the beginning of the project and then expending effort to control changes to the requirements.

3.3 Use cases

For having a better understanding of the application, I will show three use cases, one for each kind of user that can use the system.

3.3.1 Use case 1: non registered user

To have an example of a non-registered user, I had an interview with Nesibe Molla, a 26 years old student at Växjö University. She is quite familiarized with computers and internet, but only at user level, so she can be a good example of the typical internet surfer.

In this meeting, I asked her to use the application and tell me if she could understand what it was for, the actions that she could perform without logging in, etc.

She declared that she liked the appearance of the web pages. She said that the colours used and the amount of pictures used in the pages made the navigation quite pleasant.

She also stated that liked that the menu in the left hand side of the page was visible in every page she visited, because that way she could not get lost in the system.

This user also appreciated the chance to send an email to the administrator to get some information, because she could not find a telephone where to call to get information.

3.3.2 Use case 2: a patient

Now is the turn of the patient example. In this case, I created an imaginary user called Magnus Ferguson.

Magnus is a 40 year old man that works in a car factory in Goteborg. He has a nice cottage at 2 hours by car from Goteborg and he likes to escape there every time that he has the chance. So these trips would not allow him to be too far from the hospital, at least not for a long time.

He has been received a valid username and password and he has started using the application.

In his opinion, this can be a very useful tool, and the simplicity of the interface provides big potential to this application.

Once the patient has logged in, it is able to:

- Send a form to the doctor
- Ask for an appointment
- Other options (not implemented yet)
- Access to the public interface to read this information
3.3.3 Use case 3: a doctor

To illustrate the doctor use case I created an imaginary doctor called Henry Radisson, a 50 year old doctor at VårdCentralen in Växjö.

This doctor is a registered user, so after logging in he has been able to:

- Read the information of the patient
- Check all the forms sent by a single patient
- Send an email to the patient telling him to come to the hospital
- See all the appointment requests sent by a patient
- Other options (not implemented yet)
- Access to the public interface to read the information about diabetes

The doctor has talked about the commodities of having this kind of applications. He has found the application very interesting but he has admitted that he would like to have a wider variety of features within the application.
4. Requirements

To allow patients to access to the web application, there are some technical details to consider.

The users can access to the public part of this application and surf through it without logging in (no necessary username and password). On the other hand, if they want to access to the private part of the application and send information to the doctor, they will need a valid username and a password.

The username and the password must be given by the doctor in the hospital, is not possible to get them through the web application.

These are some of the requirements needed. To explain them better, we can divide them in functional and non-functional requirements. The next lines give us an idea of the requirements that this project will have.

4.1 Functional requirements

The functional requirements for the application are different for the different types of people that use the application.

The requirements for the normal (non-registered) internet users are:

• The application must give the chance to normal users to surf through the application and get information (what is diabetes, types, FAQ, interest links, etc.) about diabetes without logging into the system.

• It should allow the user to contact the administrator for getting some extra information like what does a user have to do for being able to log into the system or how to contact the doctor.

• Accessing to the private area of the application is forbidden unless the user has a valid username and password.

There are some specific options (or services) that, for security reasons, must be only available for registered users.

• When a user log into the system, he must be able to see his/her personal information in the screen, but he wont be able to modify it. To do this kind of modifications, the patient must go to the hospital and communicate it to them.

• The patient must be able to access to the public area of the application.

• Once the patient is in his homepage, he must be able to send a form to the doctor with the medical information about his disease.

• It must also be able to ask for an appointment with the doctor.

• All the fields of these two features (form and appointment) must be filled or the information will not be added to the database.

• The patient must not be able to access to the doctor features, like checking the information of another patient.

There is another type of user that can use the application so there are specific requirements for them.

• The doctors must be able to access to the public area of the application.

• Once the doctor has logged in, they must be able to see a list of all the patients and then, after selecting one of them, access to the information of that patient.

• For each patient, the doctor must be able to see all the forms that the patient has sent as well as the appointment requests.
• The doctor must be able to send an email to the patient to ask him to go to the hospital if any parameter in the forms is out of the normal range.
• The doctor must not be able to access to the patients options like sending a form or requesting an appointment.

4.2 Non-Functional requirements

• The application must be accessible 24 hours a day, 7 days a week. So a computer with internet connection (server) will be necessary. The application and the database will be stored in this computer.
• A 1280x800 pixels resolution is recommended to visualize the application in the optimal conditions, but other resolutions have been tested and the result is also good.
• To allow patients to use this application it is necessary to buy an internet domain (a hypothetical one could be www.epatient-telemedicine-project.com) and associate it to the application.
• The system must be able to recover from failures like power down, so it should be convenient to have a power generator that runs automatically if the power supply fails.
• The system will be used by multiple users at the same time, so it has to offer a good response time (real time interaction). For that we will need a computer with good enough characteristics (HD space, RAM memory, CPU speed, etc.).
• Depending of the number of patients that will use the application, we will need more or less hard disk space to store the database. A 120 GB hard disk would be enough for the normal use of the application but it should be convenient to have a second hard disk drive of at least 100 GB for storing the security copies of the database.
• It should be convenient to have at least a computer with 1GB of RAM memory and a CPU speed of 2.0 GHz.
• Technology used:
  o This project has been developed using Java and Windows XP.
  o The main application used for developing has been a tool called NetBeans IDE 5.0 (from SUN Microsystems Inc.). For more information about NetBeans, see Appendix.
  o This tool includes apache Tomcat, the server used for running the application.
  o I have also used MySql Administrator (GPL license) for creating and handling the tables of the database.
  o To work with a graphical interface, I have used a tool called MySql Query Browser (GPL license) that has allowed me to consult the database using a graphical interface.
• This approach (Java NetBeans with MySql) has been selected for multiple reasons (as you can see in the appendix)
  o The first one is that they are free, so I do not need to buy any license
  o NetBeans is a complete develop environment that includes the Apache Tomcat, so it facilities the job of the developer.
  o Java is portable; it can run applications in Windows, Linux, or Solaris. If I had decided to use.Net, I could only use Windows.
  o I have used MySql because is freeware, and it is also a very good tool for admin the databases in a graphical environment.
5. Proposed solution

This web application has been designed with the aim of being very easy to use and understand, because people who will use it can be not so familiarized with computers and internet, so I have tried to make it as simple as possible.

On one end of the communication system we have the patient, who can get useful information about diabetes by just using a computer and interacting with the web application.

The web application will do the following tasks:

- It will offer public information to the users who are using it.
- It will offer the chance to login into the system to access to the private area of the application.
- It will write information in the database (the forms or appointments that the patient has previously filled).
- It will read information from the database and will show it to the user (patient or doctor).

In the sections 5.1 and 5.2, I will describe the way the application and the database have been designed.

To develop this project I have decided to use Java and MySql. The reason because I have chosen it is because Java is a portable language (valid for Windows, Linux, Solaris…) and because it is one of the most widely used programming languages all over the world. A Java freeware tool offered by Sun Microsystems is NetBeans, which offers a friendly environment to develop Java applications. It also includes Apache Tomcat server so, it’s the one that I have chosen.

About MySql, the reason is the same, it has freeware license and SQL is one of the most widely databases languages used. It also has many features and tools (like the MySql Query Browser) that make easier the interaction with the database.

5.1 Application design

To have a better understanding of how this web application works, we can take a look to the picture 5.1, where we can see an explanation of the relations between the pages of the application and how we browse in the system.
The application starts in the welcome page. If the user logs in, he will be able to access to the private interface of the application, if not, he will be only able to surf in the public interface.

In the public interface, we can access some static pages that will give us some information about diabetes, but we will not be able to send information to the doctor.

If the user wants to log in, he has the chance to do it from any of the pages of the public interface.

Once he has logged in, there are two possibilities:

If the registered user is a patient, he will be able to:
- Send a form to the doctor
- Ask for an appointment
- Other options (not implemented yet)
- Access to the public interface to read this information

If the registered user is a doctor, he will be able to:
- Read the information of the patient
- Check all the forms sent by a single patient
- Send an email to the patient telling him to come to the hospital
- See all the appointment requests sent by a patient
- Other options (not implemented yet)
- Access to the public interface to read the information about diabetes

In the diagram above, the circles correspond to an action that can only be performed in the interface from they are hanging from.
5.2 Database design

The database has been developed using MySql 5.0.

This database consists of four tables that store the information of the patients. The reasons for choosing four tables instead just one are simplicity and security.

For example, the users will not be able to modify the content of the `user_info` table or the `authentication` table. For security reasons, this information will be added by the system administrator in the hospital.

Thus the patient will need other tables where they can store the information they want to send to the doctor. I have created two tables called `form` and `appointment`. The patients will be able to add information to these tables. I have used two tables instead of one to ensure the data integrity.

The `form` table has two primary keys (userid and form_id) that allow a patient to have more than a form stored in the table.

The `appointment` table also has two primary keys (userid and appointment_id) that allow the patient to have more than one appointment request stored in the table.

Keeping this information in separated tables simplifies the understanding of the database and avoids conflicts with duplicated rows.

In this database scheme, each patient will have a unique USERID that will identify him/her in the tables.

For example, the patient “Test” have the USERID = 1 in the authentication table. So the table USER_INFO will have a row (only one for each patient) with the personal information (name, surname, address, etc.) of this user (userid field = 1).

In the tables FORM and APPOINTMENT, each row with the field USER = 1 will identify the user “Test”.

The following scheme shows how the tables are related:

![Database Scheme](image.png)

**Picture 5.2, database scheme used in this application**
The picture 5.2 shows the database scheme used in this web application.
As I mentioned before, this scheme contains four tables. I will describe them briefly
to give you a better understanding of the whole application.

**Authentication:** This is the first table used in the application. When a patient wants to
send information to the doctor, he has to log in. This table contains the username,
password and user’s ID of the users allowed accessing to the private area of the
application, as well as the doctors who will control them. Let’s take a look to the fields
of this table:

- **UserID:** This is a unique number assigned to a patient. This number identifies
  only to one patient, and it will be used to search for the patient information in
  the other tables. (This is the primary key, so any other patient will be allowed to
  have the same UserID).
- **Username:** It specifies the name that the user will use to log into the application.
- **Password:** It specifies the secret password of the user for accessing to the web
  application

**User_info:** This table contains the personal information of each one of the patients
(name, surname, address, etc). This table can not be modified by the patient. This table
will be filled in by the administrator (at the hospital) once the patient have applied for a
personal username and password for using the web application. The fields of this table
are:

- **UserID:** This is a unique number assigned to a patient. This number identifies
  only to one patient, and it will be used to search for the patient information in
  the other tables. (This is the primary key in this table, so any other patient will
  have the same UserID).
- The other fields like name, surname, address, etc., correspond to the patient’s
  personal information.

**Form:** This is one of the tables that can be filled by the user by using the web
application. This table contains the information of the forms that the patient sends to the
doctor (glucose level in blood, insulin amount that the patients take, blood pressure,
etc.). This table will have two primary keys, UserID and Form_ID, so the same patient
will be able to have more than one form (more than a row in this table). The fields of
this table are:

- **UserID:** This is a unique number assigned to a patient. This number identifies
  only to one patient, and it will be used to search for the patient information in
  the other tables. This field is one of the two primary keys in the table.
- **Measure_val:** This field is used to identify when the measurement
  was performed (before breakfast, after lunch, etc.).
- **Measure:** This field identifies the glucose level on blood, the amount obtained in
  that measurement.
- **Time:** This field specifies the time of the measurement in a more precise way.
- **Insulin_type:** This field indicates the type of insulin that the patient takes.
- **Insulin_amount:** This field specifies the amount of insulin that the patient has
  taken in that case.
- **Weight:** Weight of the patient.
- **Height:** Height of the patient.
- **Min_pressure:** It indicates the patient’s minimum blood pressure in that moment.
• **Max_pressure:** It indicates the patient’s maximum blood pressure in that moment.

• **Hypoglycemia:** It specifies if the patient has had a hypoglycemia attack in the last month.

• **Test_value:** This field indicates the result of the Microalbuminuria test. (This is a simple test that can be made by the patient at home).

• **Form_ID:** This field corresponds to the number of the form in the table. This field is also the primary key. This way, the same patient can have multiple forms in the same table.

**Appointment:** This is the second table that can be filled by the user through the web. This table contains the information of the appointments that the patients have requested. Let’s take a look to the fields of this table:

- **UserID:** This is a unique number assigned to a patient. This number identifies only to one patient, and it will be used to search for the patient information in the other tables.

- **Comment:** It indicates the own appointment. For example, if the patient wants to change the date or the time of an appointment, etc.

- **Date:** Specifies the date in which the appointment request was done.

- **Appointment_ID:** This is a unique number (Primary key) assigned to an appointment. This field (and the User ID) will be used to identify all the appointments of a given patient. If I have only used the UserID field as a primary key in this table, a UserID (patient) would not be allowed to appear in more than one row (have more than one appointment).

### 5.3 Other security issues

At the moment, the only security measure in this system is the login process. Only users with a valid username and password will be able to access the private areas of the application.

There are other security issues that this system should take into consideration, but because of time constraint, they have not been implemented yet. Future versions of this project should contain these features.

A basic improvement of the existing system should be that when a patient is going to insert some information (like sending a form), there should be a mechanism to control that the values that the patient is adding are correct. At the moment, the system checks if all the parameters have been filled in, but does not control if there is a value out of a normal range.

For example, if a patient introduces a weight value of 780 kilos, the system should detect it as an error. At least, the system should send an alert message to the user, telling him that the weight parameter is out of range. Then the patient should have the chance to modify this value, or to communicate the doctor that there was an error.

Another security measure that should be added is encryption. To ensure the data confidentiality, the information sent by the user should use some kind of encryption mechanism. This encryption could be unencrypted at the destination computer, before adding it to the database.

The next step in the development of this project, should be the addition of these security measurements, otherwise the usability of this project would decrease considerably.
6. Interfaces

This application will be accessible by different types of users, so each of them will need a different interface.

The types of users that can use this application are:

- Normal internet users (the ones that types “diabetes” in Google and find our website. They will only be able to use the public interface, but nothing else.
- Patients, they will have a username and a password that will allow them to access to the private interface for sending forms or appointments to the doctor.
- Doctors that are a special type of private users. They will access to the private interface, but they will have different options than the patients.

The picture 6.1 shows the different interfaces that a user can find, depending of the type of user.

Picture 6.1, shows the different interfaces of the system
All the users will have the chance to access to the public interface, and surf through their pages, but only authorized users will be allowed to access to the private interfaces.

As you can see in the picture 6.1, the patient’s interface will be accessible only to the patients. The same happens with the doctor’s interface that only will be shown to the doctors not to the patients or to the normal users.

In the sections 6.1, 6.2 and 6.3, I will give a brief explanation of each interface. For a deeper explanation, check the Appendix B.

6.1 Public interface

This interface will be available for all kind of users. In this case, the user will not be able to add information into the database. The only thing that these users can do is to read the information displayed on the public pages of the application.

6.2 Private interface

This interface will be available only for those patients that have been authorized previously by the system administrator at the hospital. This interface allows the user to add information into the database (forms and appointment requests) as well as accessing to the public interface.

6.3 Doctor’s interface

This is a private interface that will only be available for doctors. It will not be accessible for patients or non-registered users.

By using this interface, the doctors will be able to read the information sent by the patients and check if there is any value out of range.
7. Conclusions

In this chapter I will take a look to the goal criteria stated in the chapter one and I will discuss about how this application has covered those goals.

One of the goals of this project is to offer an interactive and useful tool in the war against diabetes.

Different users have tested the application and they have found it very useful and easy to understand.

Nowadays many people are familiarized with internet and they way it works, so the test users did not have any problems in understanding how to navigate through this application.

Another goal that we wanted to achieve was to offer some free information to the normal internet users (non-registered users) and with the public interface, this goal is achieved, because they do not need to login to see these pages.

About the usability, they have declared that by using this application they can reduce the number of visits to the hospital that they have to make along the year. This way they can save money and time, and reduce the problems that these trips can cause.

About the security, the application has different interfaces. Only authorized users are allowed to access the private interfaces, so the personal and medical information is safely stored in the database.

There is also a separated interface for patients and doctors, so the patient will not be able to use the doctor interface, and the doctor will not be able to use the patient interface. This way, the application ensures a correct use for the different kind of users.

This kind of disease changes the life of the people who suffers it; I can affirm that because I have experienced it in my own family.

With this web application, I have tried to find a way to make their life easier. Internet and new technologies offer us a wide range of tools that we could use to reduce these barriers.

This is a simple application that allows users to exchange information with the doctors and it can also be used as a reference whenever a patient has a doubt about diabetes.

On the other hand, doctors can also be beneficed by using this application. If a patient does not need to go to the hospital so often, other people who really need to go there will not have to wait that long queues, what is in theory, more efficient. Then the working environment can be less stressful for the doctors and everybody will gain: patients and doctors.

I said “in theory” because maybe there are so many patients that the waiting list will be long anyway. At least, this application can help to reduce that list.

The doctors will have the commodity to use the application from their own home computer, so they can check everything from their own place without going to the hospital.

By using this application they can also send an email to the patient (another of the proposed goals) whenever they detect a value out of range, and that way the patient should go to the hospital for a regular check.

New technologies offer us many possibilities, and I think that this application could be useful for many people and even it could be extended to be used with different purposes, not only the diabetes control.

There are some slogans like “Impossible is nothing” or “The future is now” that we have listened to, but I honestly think that initiatives like this project (even when its only a student project with some limitations) are the first step in the long way that we have in front of us.
8. Future work

This web application could be modified to be used with a wide range of diseases, not only diabetes. Even more features could be added to the already existing ones, extending the usability of this project.

For example, if the doctor wants the patient to take a new medicament, he will need a recipe to buy it in the pharmacy. It this application would use a fax or a scanner with a printer plus the corresponding security checks, the patient would be able to print the recipe at home and go with it to the pharmacy.

With some small modifications, this application could be used, for example, to send an X-ray picture to the doctor (using a scanner), or a picture of a burned arm, it could handle a videoconference (adding a camera) with a psychologist, etc.

It could be also modified, to use the phone instead of the website. There are some people who are not so comfortable by using computers, they prefer the phone. So this application could be used like a call centre who gives the information to the patients by listening to some educational messages or storing the information into the database by using the phone keypad.

Another feature that could be added is to send an SMS to a mobile phone. If the doctor finds a value out of range, the application can send an SMS to the patient telling him/her to visit the doctor as soon as possible.

This could be implemented, for example, with the button “other” that I have added to the application. Whenever the doctor presses that button, he would be able to send an SMS to the mobile phone of the patient. The phone number is already stored in the database, so this could be a new feature that could be added to the current application.

About the security, this is an area that should definitely be improved in future versions.

For example, the next version of this application should include a mechanism that sends an alert message to the patient and the doctor when the value of one of the parameters of a form, is out of range. The next version should also allow the patient to modify the values of a specific form.

Another security measure necessary in newer versions is encryption. Information sent by the patient to the database and read by the doctor from the database, should be encrypted to ensure confidentiality.

Without these security measures this project would be useless, because confidentiality of data is a basic requirement in every application that works with databases.

I think this is a very interesting area, and maybe in a future I will be able to keep working in this project.
9. References


dicine%20diabetes%20sweden%22](http://www.medsci.uu.se/mie/project/dino/NBC2005_shortpaper.pdf#search=%22telem
dicine%20diabetes%20sweden%22), accessed 2006-09-16.


Appendix A

A.1 NetBeans IDE

NetBeans is a successful open source project with a huge user’s database, a community that is continuously growing (more than 100 societies all over the world).

NetBeans IDE is a robust, open source Java IDE (Integrated Development Environment) that has everything software developers need to develop cross-platform desktop, web and mobile applications straight out of the box. You can write, compile, debug and deploy Java programs for the Solaris, Windows, Linux and Macintosh platforms.

This open source product is free for commercial and non commercial use. The source code is available for reusing following the Common Development and Distribution License (CDDL).

NetBeans IDE 5.0 requires J2SE SDK, version 1.4.2 or higher.

The NetBeans information of this appendix has been obtained from the Sun Microsystems homepage. More information and downloads in the web page http://www.sun.com or in http://www.netbeans.org/index_es.html.

Here we have a screenshot of the NetBeans interface:

![Picture A.1, NetBeans interface screenshot.]
A.2 MySql Administrator

MySql Administrator is a powerful visual administration console that enables you to easily administer your MySql environment and gain significantly better visibility into how your databases are operating. MySql Administrator now integrates database management and maintenance into a single, seamless environment, with a clear and intuitive graphical user interface.

MySql Administrator enables developers and DBAs to easily perform all the command line operations visually including configuring servers, administering users, and dynamically monitoring database health. Other common administrative tasks such as monitoring replication status, backup and restore, and viewing logs can also be performed through the MySql Administrator graphical console.

The information about MySql Administrator of this appendix has been obtained from MySql homepage. More information and downloads are available in [www.mysql.com/products/administrator](http://www.mysql.com/products/administrator).

The picture A.2 shows a screenshot of the MySql Administrator interface:

![Picture A.2, MySql Administrator screenshot](image)

MySql Administrator allows us to modify the tables in a very easy way, as the picture A.3 shows:
Since I have been working with Windows, I feel more comfortable working with an easy windows interface, instead of using the command line. MySql offers a variety of tools that help the users to work with a graphical interface instead of using the command line. One of these tools is MySql Query Browser.

MySQL Query Browser is the easiest visual tool for creating, executing, and optimizing SQL queries for your MySQL Database Server. The MySQL Query Browser gives you a complete set of drag-and-drop tools to visually build, analyze and manage your queries.

It can be downloaded from http://www.mysql.com/products/tools/query-browser/, page where I have get the information for this part of the appendix.

The picture A.4 shows a screenshot of how does this tool works:
Picture A.4, shows how the tables can be modified easily by using this tool.

This is the main query window. In the upper side we can see the SQL query and in the lower part we can see the result of that query.
Appendix B

B.1 Public interface

This application has a public side that will be accessible for the user without logging in.

In this case, the users will be able to surf through the pages of our application without registering. They will be able to read the educational information that the different pages offer, like what is diabetes, the existing types of diabetes, the frequently asked questions, the interesting links, etc.

In this case, there will not be interaction with the database and there the user will not be able to send information to the doctor.

When the application starts, the user will find the welcome page. In this page, the user will receive some information about this project, the purpose of it, what can the user find surfing through the pages, etc.

The main page will also explain the user that he has the possibility to login and access to the private options that this web application offers.

In this page, I will also explain the actions (in general terms) that the doctor can perform by using this interface.

The picture B.1 shows a screenshot of the main page, the one that the user will find when he uses this application:

![Picture B.1, welcome page of the web application](image)

Once the user has accessed the application and he is in the main page, he will find a menu with some options in the left hand side of the screen. He will be able to access to this options without logging in.
The first option of this menu is “What is Diabetes”. This is the page that the user will access if he/she clicks on that link:

![Picture B.2, screenshot of the page “what is diabetes”](image)

In this page, we can find a small explanation about what is diabetes and why is it a health problem.

In this case I have just tried to give an overview of this disease but the information can be easily updated or modified by a doctor.

To have a better understanding of the illness and the information that could be interesting for a normal user (patient or not) I have been supervised by the Swedish doctor Lars Nilsson (NSV VC in Teleborg, Växjö).

The second link that the user can access to is the “Types of diabetes”. This page gives a brief explanation about the different types of diabetes existing, the causes, the group of people (age ranges) more common for each type, etc.

The picture B.3 shows a screenshot of this page:
The next option that we can find in this menu is “Taking care”. This page explains that there is no cure for this disease and also explains the kind of treatment that the patient must receive for each type of diabetes.

Under these lines we can see the corresponding screenshot for this page.
Taking care

There is no cure for diabetes, so the key to good health for people who have this disorder is to control it: to keep the blood sugar level as near to normal as possible. Good control can go a long way toward Prevention of Complications of Diabetes related to the heart and circulatory system, eyes, kidneys and nerves. Good control of blood sugar levels is very possible by planning what you eat, staying physically active, taking your medication as directed and checking your blood sugar level often.

A number of advances in recent years have made improved blood sugar control easier to achieve. Many people with Type I diabetes have been improved control with intensive insulin therapy, using multiple daily injections or an insulin pump. The Food and Drug Administration (FDA) has approved Hiralog, a new, very fast-acting insulin, which should help control the rise in blood sugar that occurs immediately after eating. And research is continuing on the development of an implantable insulin pump that would make injections a thing of the past.

New pills for the treatment of Type II diabetes are now available. Glucosegen (metformin) works by sensitizing the body to insulin. Unlike other diabetes pills, which tend to promote weight gain (Glucosegen is often noted as weight reduction. Some people with Type II diabetes who have been taking insulin are able to stop it when Glucosegen is added to their treatment program.

Precise (acarbose), another new pill, works by blocking the absorption of starch, resulting in less of a rise in blood sugar immediately after eating. Improvements continue to be made in home blood glucose monitoring devices, with newer instruments being smaller, faster, and requiring less blood than older models. One of the best indicators of how well your diabetes is controlled is the Glycosylated Hemoglobin Test, which shows your average blood sugar level over the past three months. By using these test results to improve your diabetes control, you can reduce your risk of complications of diabetes.

The following screenshot corresponds to the next link in the menu, the FAQ (Frequently Asked Questions).

In this page, the user will find a list of the most common question for the patients with this disease. Here we have the picture:
The last option of the menu is called “links”. In this page the user will find a list of links of interest where the user can click to receive more information about this disease.

In this page we can find the link to some world well known associations about diabetes where the patient can maybe found more specific information that he could not find in our website.

This is the corresponding screenshot:

![Picture B.6, screenshot of a page with interesting links related to diabetes](image)

The last option of the public side of this application is the “contact us” option. By clicking this option, the user will be able to send an email to the network administrator (in this case me) who will be able to answer it or to forward to a doctor (if is a medical question).

For the rest of options, it will be necessary to be logged in the system.
B.2 Private interface

If the public interface was destined to any user (a normal Internet user that types “telemedicine” or “diabetes” in Google), the private side of the Project is only destined to those patients that have previously registered in the hospital, and that have received a valid username and password.

In picture B.7 you can see how the patient logs into the system by tipping the username and password.

The red circle in the screenshots shows how it happens.

When the patient clicks the login button, the application will make a query in the database, to check if the user exists or not.

If the login is successful, a page will show us a message telling us that we are logged into the system. The screenshot (picture B.8) illustrates this case:
Once we have logged in and we have clicked “continue”, we will access to the private area of this application.

In this case, the application redirects us to the patient homepage.

In this page we will find, on the left hand side menu, the same options that normal users have without login but in the centre of the page we will find something different.

First, we can see the patient’s personal information (obtained from the user_info table).

Under this information, we will find 3 small buttons with different purposes.

- **Send information button:** This button will take us to the page for sending a form to the doctor. This form contains the medical information (glucose level, min and max pressure, etc.) that the doctor needs to determine if there is something wrong and we should go to the hospital for a check or if everything goes well.

- **Ask for an appointment button:** This button will take us to a different page. In that page we will be able to ask for an appointment to the doctor, or to modify the date/time of the existing one.

- **Other button:** This button has been added for offering an open field for future features that can be added to this application.

The picture B.9 shows the screenshot of a patient homepage:
Now I will show the two screenshots corresponding to the two main options that a patient has by using this application.

This is the page that the patient will access by clicking the “send information” button:

![Picture B.9, patient home page](image)

![Picture B.10, where the patient can send the form to the doctor](image)
The picture B.10 shows a form that the patient is going to send to the doctor (store it in the database).

A requirement for adding a form in the database is that the patient must fill all the fields of the form, if any of those fields is empty, an error message will appear. In this case, the information will not be added to the database; instead, the user will be redirected to the form and will be asked to fill all the fields before storing it in the database.

The next screenshot (picture B.11) shows the error page that a patient will find if any of the form fields are empty.

As you can see in the picture B.11, if any field is empty, the patient will have to go back to the previous form and fill all the fields before storing it into the database.

Once all the fields have been filled, the information will be added to the database.

The other option that the patient has is to ask for an appointment. The process is basically the same.

The patient will be required to fill the comment field, in which he/she will have to explain the reason of the change, and specify if there is any date more convenient for the meeting.

The hospital will answer them by phone or email with the new appointment.

The picture B.12 shows a screenshot of the appointment page:
At the moment these are the two specific options that the patient can use to send information to the doctor. There is a third button that can be used to add more features to this project.

In the left hand side we can find a new field, only available for registered users, that can take the patient directly to his/her homepage.

**B.3 Special users: Doctors**

This application is also a tool that the doctor can use to check the information of his patients.

The doctors will have a different interface than the patients. They will not need to fill any forms; instead, they will have to check the information sent by the patients to see if there is anything parameter out of range. Thus, the doctors will have a different appearance in their homepage.

On one hand, the doctors will have a menu on the left hand side of the page. This menu is the same as the patients have. So the doctor will be able to read all the information about diabetes that a normal patient can have.

In the centre of the page, the doctor will have specific information that a normal patient does not have.

The picture B.13 shows a screenshot of the doctor’s homepage.
On it, the doctors will have a list with all their patients. If they want to check the information of any of them, they only have to select one and click on the “view patient” button.

One the doctor has selected a patient, and clicked the button, he will access to the patient information.

The picture B.14 shows an example of a selected patient, with his personal information and some other options.
The picture B.14 shows a selected patient. First we can find his personal information (name, age, nationality, etc.) and under this, the doctor has some buttons with different purposes:

- **View forms button**: the doctor can click this button to see all the forms sent by this patient.
- **View appointment button**: this button can be clicked by the doctor to see all the appointment requests sent by that patient.
- **Other button**: this is also a button that I added for future options that can be added to the application.
- **Email patient button**: with this button the patient can send an email to the patient if there is anything wrong.

The picture B.15 shows the list of forms sent by a single patient to the doctor. By checking this table, the doctor can determine if there is any parameter out of range and then order the patient to come to the hospital for a check.

This can be done by sending an email to the patient’s email address or could also be done by a phone call from the hospital’s administration staff.
If the doctor clicks on the “view appointments” button, he will get another table with all the requests sent by the patient, and then he can set a new appointment. The picture B.16 shows the appointments list sent by a single patient.

This appointment should be set by the hospital and not by the doctor, because maybe the doctor does not know the available dates or the waiting list that the hospital have, that is the reason because the name of the button is “generate new appointment”.
At the moment, that are all the options that a doctor has. New features can be added in the future and that is the reason because I added an extra button to the doctor’s homepage.