



UNIVERSITY OF NEW YORK TIRANA

SYSTEM ANALYSIS & DESIGN: PROJECT

Hospital Management System

Professor:
Olta Sula
System Analysis & Design
Department of Computer Science

Group: Regi Nishani,
Xhuliano Kushova,
Daniele Llazo
Juxhin Gjata

Deadline: 24 November 2021

Contents

1	Introduction	2
1.1	Scope	2
1.2	User Characteristics	2
1.2.1	First Page	2
1.2.2	Doctor	2
1.2.3	Patient	2
1.2.4	Administrator	2
2	Feasibility Study	4
3	Technologies Used	5
3.1	FrontEnd and BackEnd implementations	5
3.2	Database	7
4	Testing	9
4.1	Different types of tests	9
4.2	Functionality Testing	9
5	Diagrams	11
5.1	UML	11
5.2	Use Case Diagram	12
5.2.1	Dashboard	12
5.2.2	Patient	13
5.2.3	Doctor	14
5.2.4	Administrator	15

1 Introduction

1.1 Scope

Our team decided to build a HMS (Health Management System). The aim of this system is to serve the community to have the chance of a better service from Medical Institutions and also to serve these institutions' staff to offer better service to their patients. In order to make it as user friendly as possible we decided to separate the system into three different sectors. The idea to build this project came from the covid-19 pandemic crises that had a large financial cost for the governmental and an even larger logistical cost in human and other resources, due to the unplanned events that occurred in the past two years. This application tends to have a highly running efficiency and a low building and maintaining cost.

1.2 User Characteristics

1.2.1 First Page

After you open the application, you will be directed to the first page. In the first page you may choose between Doctor, Patient, or Administrator, depending on the person who is going to use the website. You may also choose contact option above, where you may inquire about our services.

1.2.2 Doctor

You are required to log in first. Once you are logged in you will be directed to dashboard. On the menu panel on the left, a doctor may see his patients, appointment history, or simply search what he is looking for. On the top right, he can check his profile, update the password or log out.

1.2.3 Patient

You are required to log in first. Once you are logged in you will be directed to dashboard. On the menu panel on the left, a patient can check his medical history, appointment history, or book an appointment. On the top right, he can check his profile, update password, or simply log out.

1.2.4 Administrator

You are required to log in first. Once you are logged in you will be directed to dashboard, where he will be shown the most useful options. Administrator is the user with the most options of all. He can check all the users, patients, doctors,

appointment history, reports, search for specific patient, and even watch their latest logs. He can even add new queries for the database. Administrator must have knowledge on database to use the last option. The administrator should also respect patients and doctors privacy, since he can access sensitive information.

2 Feasibility Study

Feasibility is the determination of whether or not a project is worth doing the process followed making this determination is called feasibility study. This of determines if a project can and should be taken. Once it has been determined that a project is feasible, the analyst can go ahead and prepare the project specification which finalizes project requirements. Generally, feasibility studies are undertaken within right time constraints and normally culminate in a written and oral feasibility report. The contents and recommendations of such a study will be used as a sound basis for deciding whether to proceed, postpone or cancel the project. Thus, since the feasibility study may lead to the commitment of large resources, it becomes necessary that it should be conducted competently and that no fundamental and that no fundamental errors of judgment are made.

To meet the needs of the users, the web application that we built is a dynamic website, which means that the users of the website can change the content of the website. We used different types of technologies on process of building this website. The front end of the website:

We used HTML 5 in order to build the structure of the website for the main reason of offering the user the best and the most offering experience on the application. For the design of the website, we used CSS 3 in order to design text, images, forms and also buttons. In order to make the work seem more professional we used a prebuild CSS library called "Bootstrap".

The last technology we used in front-end was JavaScript. The file containing JavaScript code is found under "JS" folder under the name of "responsiveslides.min". In the back-end we used different technologies, such as:

MYSQL in order to build the database of the application for which we gave a detailed information earlier on this paper.

PHP- we used php for this website to connect to the database and give the user the ability to create, update, read and delete data from the database. We started by building the log in pages and making them communicate safely with the database. After the log in information is confirmed for each user, we made sure that the log in page redirected users to their specific service. The hardest work of this all was the user dashboard which has a lot of services offered to user.

The last technology we used was Apache server in order to make it possible for the website to work locally on our machines.

All the technology used in this application is separated in different folders which makes it more understandable and the work easier for the entire group.

3 Technologies Used

3.1 FrontEnd and BackEnd implementations

We separated this application into three sectors: The patient sector (regular user), the doctor sector (Hospital Staff), and the admin sector. When user try to access this application, they will be presented to the main page of the website (index.html). The main page is made of three sectors:

The Nav Bar which contains three link that can link within the website:

The Home Page(index.html), the About page, at the moment doesn't show anything since the about page info is included in this report, and the contact page on which both groups of users can as for technical assistance on cases when they face difficulties while using the application.

The Body Section:

The body section is made of a covering image and below it three well designed links. The reason the body of the application was built this way is to help users find the right section to log in immediately and it is very natural and user friendly, so that the users won't wonder around on the website.

The three sections found on this part of the body (three links) are:

Patients, links the patient to the log in page where he/she can log in to their account.

Doctors, links to the log in page of the hospital staff.

Admin, links to the admin log in page.

And the last section of the main page of the website (index.html) is the footer. Which was mostly put on the website for site identity and it does not have any other function.

The contact page on the website is made of a single contact form which requires the user to define his/her name, his/her email, his/her mobile number and the body of the message. When the submit button is pressed the user will be presented with a message by the application which lets the user know if their message was or wasn't sent to the admin. All the fields on the form are required to be filled by HTML code on the source code (contact.php).

The main goal of this website is to serve three different users:

The patient (regular user), the hospital staff (doctors, nurses) and the admins who will manage this application and report any problem to the developers if they find

any. After regular users (patients) manage to find their log in page they have to insert their credentials in order to have access to their user dashboard. Since at the moment we do not have any database provided by the intended buyer we have given the chance to anyone to sign up on the system (mostly for testing reasons). If the user does not have any account to access the system, they can sign up by click the link on the log in page which will redirect them to registration.php page. On this page the user will be asked for his credentials such as: Full name, Home Address, City, his/her gender, Email, Password. After the user agrees with the terms and conditions set by the website, he can submit the personal information and have the chance to access the user dashboard. For testing and demonstration purposes we have created a testing account with the credentials:

Email: test@gmail.com Password: Test@123 When the user is granted access to the user dashboard, she/he will be shown a menu bar on the left side of the screen which contains:

Dashboard, Book Appointment, Appointment History, and Medical History. The dashboard contains three sections on which you can find page such as: Update Profile(edit-profile.php) on which the user can change his/her personal information. My Appointments(appointment-history.php) which the user can see their id, Doctor name the Fee of the appointment, Time Date and even more.

This type of service saves the hospital staff a lot of their time and helps them to manage their resources better. Book Appointment(book-appointment.php) on which the user can book an appointment, choose the specialist he/she thinks it's the best for their medical problem. After having chosen the Specialist, the system automatically calculates the fee for the patient. Then the patient can choose the date and the time. On the left menu the patient has the possibility to also view his/her medical history with his details. The user has the possibility to change his/her dashboard style by simply clicking the button on the left. Where he can choose to have dark colors on the dashboard or have the menu fixed. The patient is offered with the log out option. On this case we created a session to stop the user to return to the dashboard unless they log in again. This is done by a session.

After the Doctor uses his/her credentials to access the Dashboard, they will be shown a personalized dashboard. On the dashboard the doctor has the possibility to update their profiles and see the appointments that he has for the day. This offers the doctors the possibility to manage their work hours better. On the left menu the doctors have the possibility to view appointments history where she/he can cancel different appointments or view his/her patients' data. Then the doctors are offered with two other options on which the patients list can be seen or if the

doctors cannot find patients on the list can search them with the search option on the left menu. The third and the last option is to log in as an Admin. The admin log in information is listed below:

Username: admin Password: Test@12345 This account was created for testing reasons. You cannot register as admin. The admin account has all the possible options found in the application. The admin can see all the listed doctors. The admin can add specializations, can add or remove doctors from the list. The admin has also the capability to see all the users added on the system and can or remove them. On the left menu the admin can also find the possibility to see all the appointments history for the hospital, the same way a doctor can see for his patients. This is very helpful to the hospital since this type of option removes the hard work and time-consuming work of going through endless files on the hospital to find the patients work. This way of technique makes it possible for data to be safe and not lost due to physical problems in the hospital problems. The users have the ability to report different problems or contact the admin for different problems that they might have on the application. We managed to build an option which makes it possible for the admin to see all the reports that are committed to the database. In order to keep the site safe from the intruders the admin has two session logs on the menu:

One of them is for the doctors and the other one is for the users. This session log gives the admin to view the users id, personal information, the log in time, the log out time and the Ip from which the site was accessed from. In the end the admin has the option to search for the hospital patients via a search bar.

3.2 Database

We build a database using MySQL. The database is made of ten tables. The tables are:

Admin, Appointment, Doctors, Doctorslog, Doctorspecilization, Tblcontacus, Tbltblmedicalhistory, Tblpatient, Userlog, Users. The database saves all the information from our webpage. It contains 10 tables. Admin table has the authentication information and data for the administrator. The appointment table saves the information for all the appointments via E-health system. Doctors table has information about doctor authentication and their profile. Doctor's log contains doctor's activity information. Doctor specialization contains all types of doctors available in the E-health System. Table tblcontact contains information from the inquiries. Table tblmedicalhistroy contains information on the diagnoses of the patients diagnosed by the doctors of this system. Table userlog has information about patients activity in the system. Users table has authentication information of the users and information

about their profile.

We connected with the database through the function:

```
define('DB_SERVER','localhost');  
define('DB_USER','root');  
define('DB_PASS','');  
define('DB_NAME','hms');  
$con = mysqli_connect(DB_SERVER,DB_USER,DB_PASS,DB_NAME);
```

(While connection to the database please check the root user on your machine, and also make sure that the user has all the privileges. The server might be defined with the MYSQL port on your machine. For example: localhost:3306. This result came as a problem we faced during the code execution).

The connect() / mysqli_connect() function opens a new connection to the MySQL server.

4 Testing

4.1 Different types of tests

We tested our website for many error that could occur. We ran all the possible option that we created for the users of the platform and they all work. We had a problem in front-end with our header image in the index.html page where the image was showing properly but we simply fixed it by removing:

`align-items: flex-wrap;`

and by adding: `align-items: center; center;`

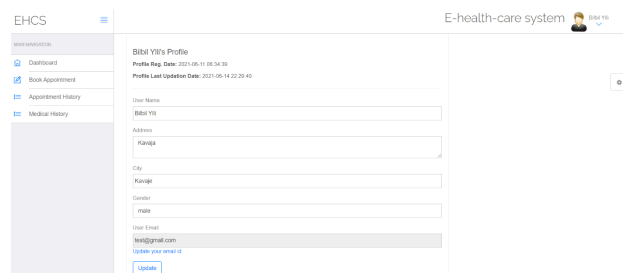
While building the website we did not notice the problem with the database connectivity until we starting using all the possible options. We simply fixed it by adjusting the root credentials and adjusting the privileges. The code showed this error:

Failed to connect to MySQL: No connection could be made because the target machine actively refused it.

4.2 Functionality Testing

We have done functionality testing for all the website functions. We are giving one example of it:

We tried a simple test to check if the updated personal of a user would be updated on the database.



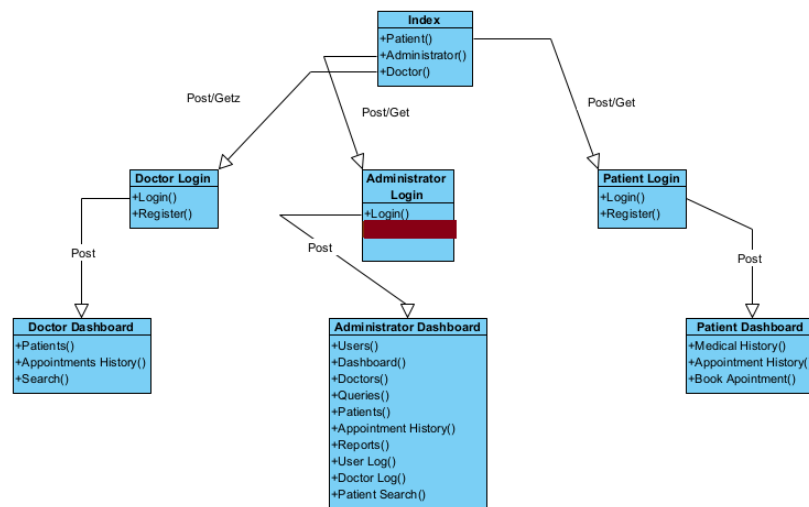
The system notifies the user that his information was successfully updated on the database.

As we see below the information is updated on the database and the password encryption is functioning as well.

10

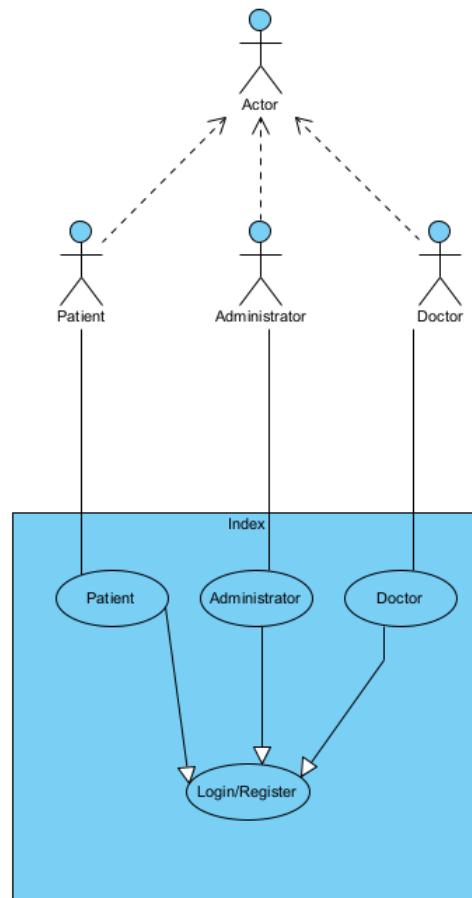
5 Diagrams

5.1 UML

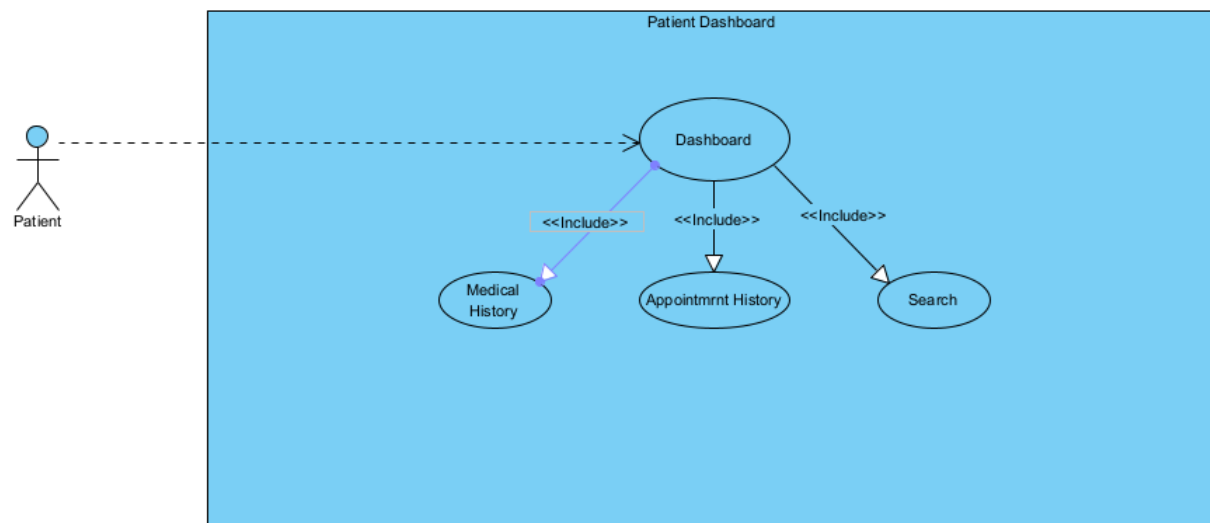


5.2 Use Case Diagram

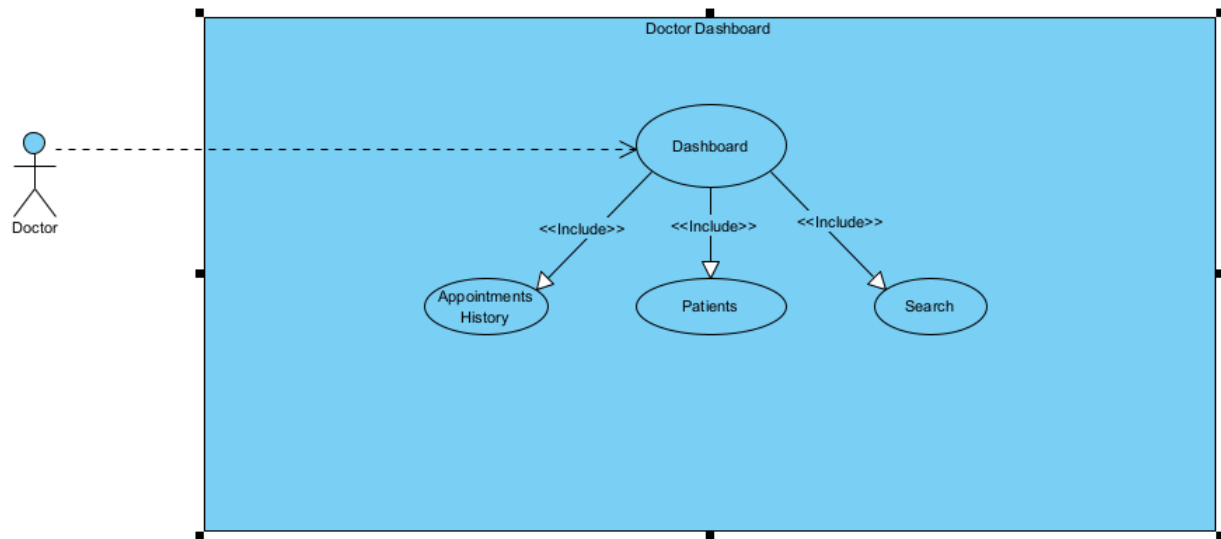
5.2.1 Dashboard



5.2.2 Patient



5.2.3 Doctor



5.2.4 Administrator

