Project: Smart E-health Consulting System

Java – Objected Oriented Programming
M.Sc. Luigi La Blunda

Authors:
Gregor Napierski (1219667)
Ali Al-Haidary (1353109)
Elia Funk (1367315)
Muhammed Calli (1363678)

February 10, 2022

DECLARATION OF AUTHORSHIP

We hereby certify that the project report we are submitting is entirely our own original work except where otherwise indicated. We are aware of the University's regulations concerning plagiarism, including those regulations concerning disciplinary actions that may result from plagiarism. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Name	**Matriculation-nr.**	**Date**
Gregor Napierski	1219667	10.02.2022
Ali Al-Haidary	1353109	10.02.2022
Elia Funk	1367315	10.02.2022
Muhammed Calli	1363678	10.02.2022

1 Inhalt

2 Project Description

The objective of the project was to implement a user-specified smart e-health consulting system, in which patients can make appointments for a specialized medical doctor based on a selected health problem that he/she is having. The patient can cancel or shift appointments if needed. The second user in the system is the administrator who has an overview over all users and can edit and delete user profiles.

The system provides a multi-user login for patient and admin accounts. Each user is supposed to have a unique account. In case a patient does not have an account already, they can register by clicking the "Create an account" button. In the registration panel they must set a first and last name, address information, the date of birth, the insurance name and type, a username and a password. At the end of the registration process the user is set to patient by default and an email is sent as a registration verification.

To get access to the functionalities of the system, the users have to login with their username or email and the corresponding password. The salt from the database is added to the password and afterwards they are getting hashed. After that the input will be compared with the hashed password entry in our database. Depending on the database entries the user gets redirected to the specific panel.

The patient dashboard panel provides several different options. The patient gets the option to edit their profile including first and last name, date of birth, address information, insurance name and type and the password. They can also set up their health information and download them as a pdf or txt file. There are also possibilities to make, cancel and reschedule an appointment. At the end of each of these three processes the user gets an email confirmation.

(Ali Al-Haidary 1353109) (Elia Funk 1367315)

3 Project motivation

For some people making a doctor's appointment over the phone is a real challenge. That is why we are motivated to make peoples every day live easier with our Smart E-Health Consulting System. With this system, patients can book appointments with their doctors 24/7 a week without having to talk to another human being. Since the doctor‘s offices are usually busy throughout the day or you are stuck in the waiting loop forever, it is way more convenient for the patients to use an alternative. In addition, patients can easily find suitable doctors in their area. After booking an appointment, patients receive an e-mail confirmation with all relevant information. Besides that, they can view every appointment they made and choose when they get reminded for that appointment.

With this system, all appointments of different doctors can be seen easily. Furthermore, appointments can easily be rescheduled and even be cancelled. The digitalisation in the health sector ensures that the doctors and medical staff save a lot of work with this system, because they no longer have to deal with making or rescheduling appointments. So, using this system, a doctor's office has more time to deal with the patients and surrender the bureaucracy to our system.

(Muhammed Calli 1363678)

4 Requirements

4.1 GUI (Graphical User interface)

- Provide a multi-user login with username and password authentication
    - For Patient and Admin
- Storing data in an online database and encrypt sensitive data
    - Relational Diagramm
- Admin operations
    - Access to all user profiles
    - Edit user profiles
    - Delete user profiles
- Patients can edit Health- and account information

4.2 Make an Appointment

- Make an appointment for a specialized medical doctor
    - based on a selected health problem doctors are displayed in a list
    - Distance of search and specific time
    - Appointment confirmation via email (Name and address of doctor, Date and time of the Appointment)
- Reminder functionality

4.3 Shift or Cancel the Appointment

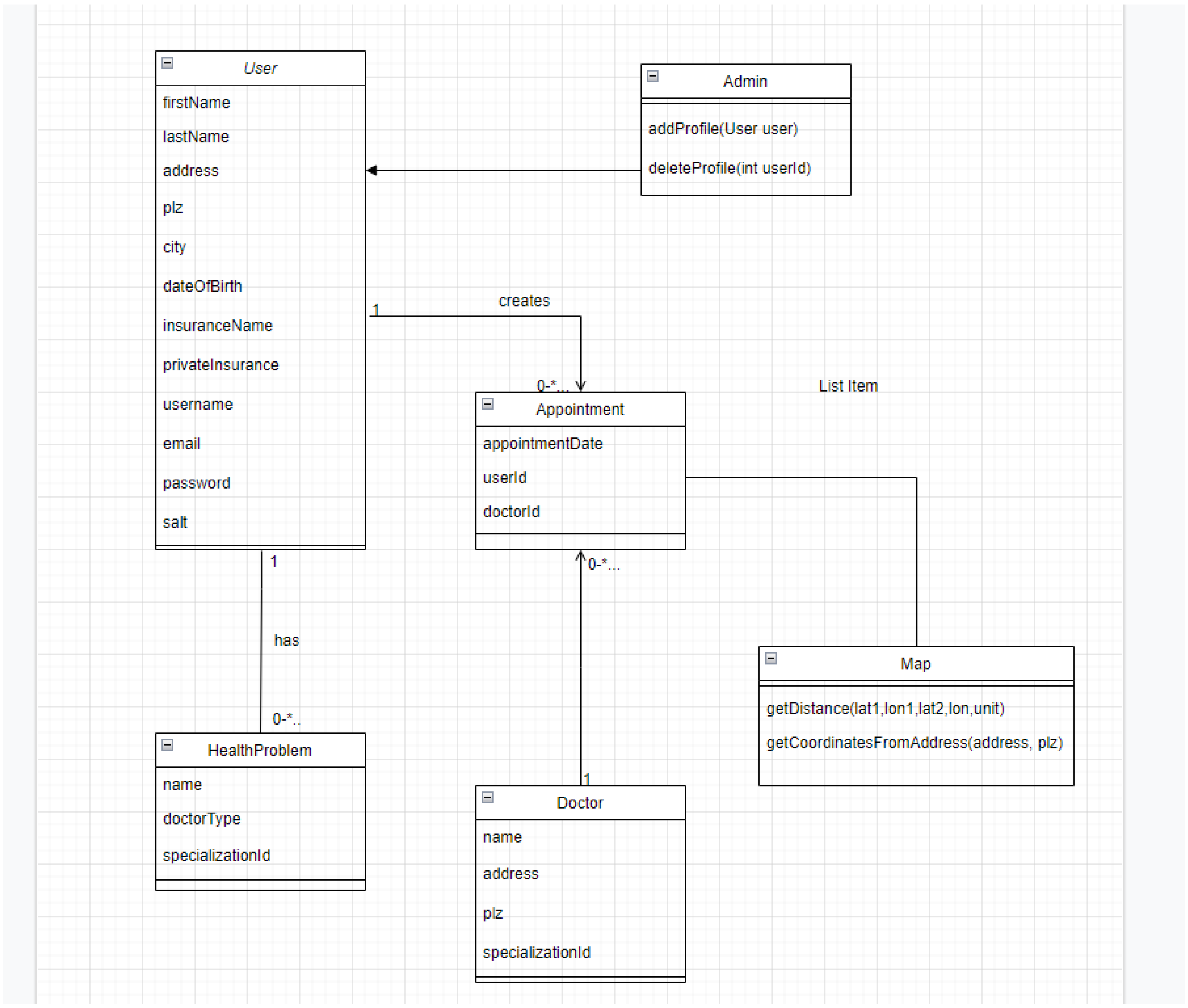
- User can shift or cancel appointments
    - Notification via email

4.4 Export the health information as a PDF or TXT file

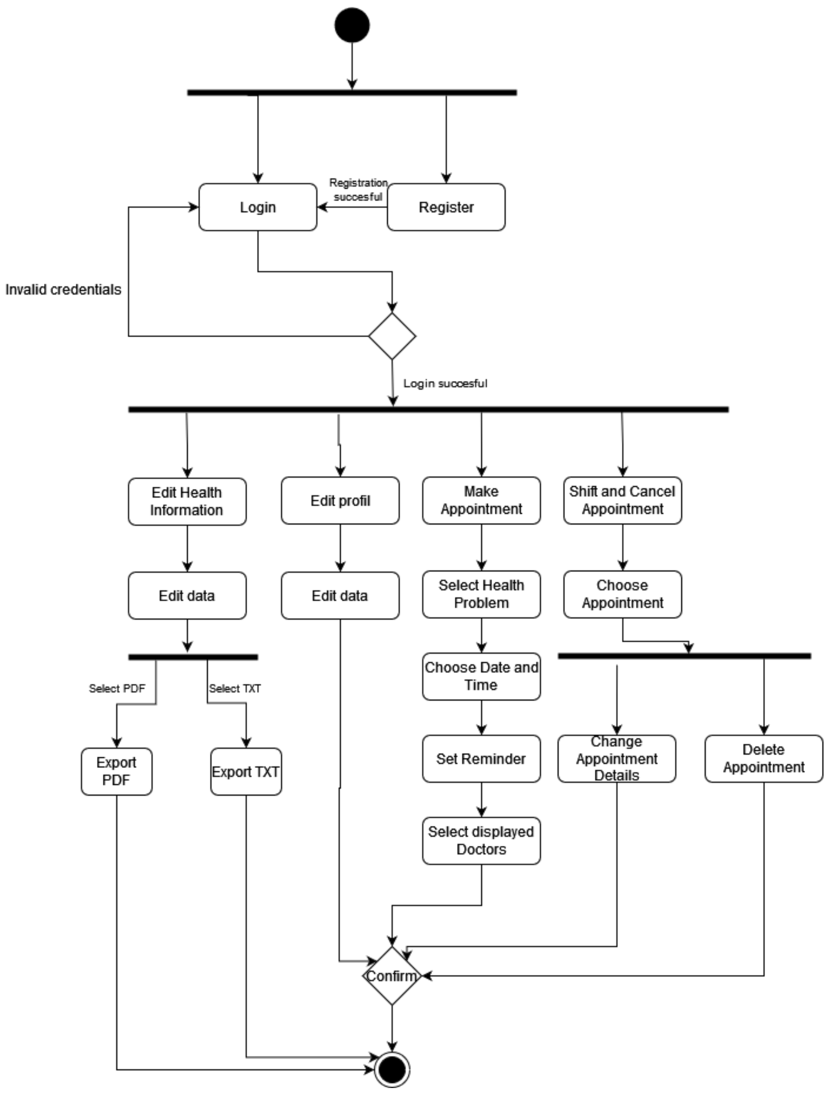
- Export as pdf or txt

5 Technical description of solution

5.1 Class diagram



5.2 Activity diagram (User)



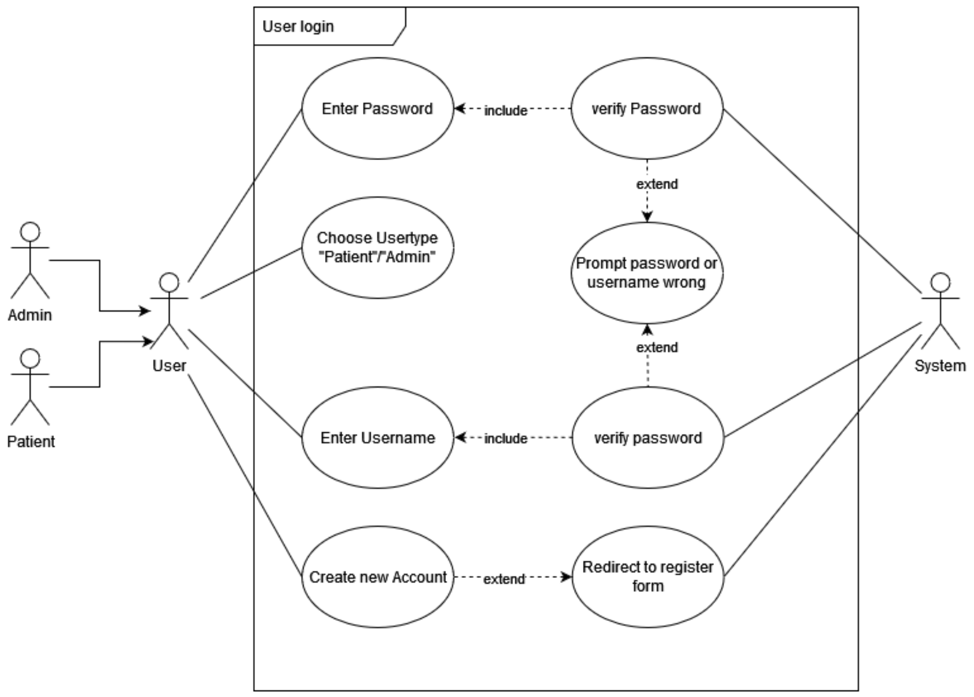
The diagram shows the path of the patient in the system. The patient can either login or register as a new patient. After the login process, the patient gets redirected to the dashboard. The patient has several choices what to do. The application provides the option to edit the health information and export these as a pdf or txt file. In addition, the profile data can be changed and a password reset option is given. There is also the option to make an appointment and if needed shift as well as cancel it.

To successfully create an appointment the user must select one of the displayed health problems. After that he needs to set a date and a time as well as a search radius and a remainder which will send an email according to the given selected index. As a result all doctors fitting the given radius and health problem are displayed in a list. The patient has to select one of the displayed doctors and create an appointment. After creating an appointment, a verification email is sent to the patient.

(Ali Al-Haidary 1353109)

5.2 Login Process

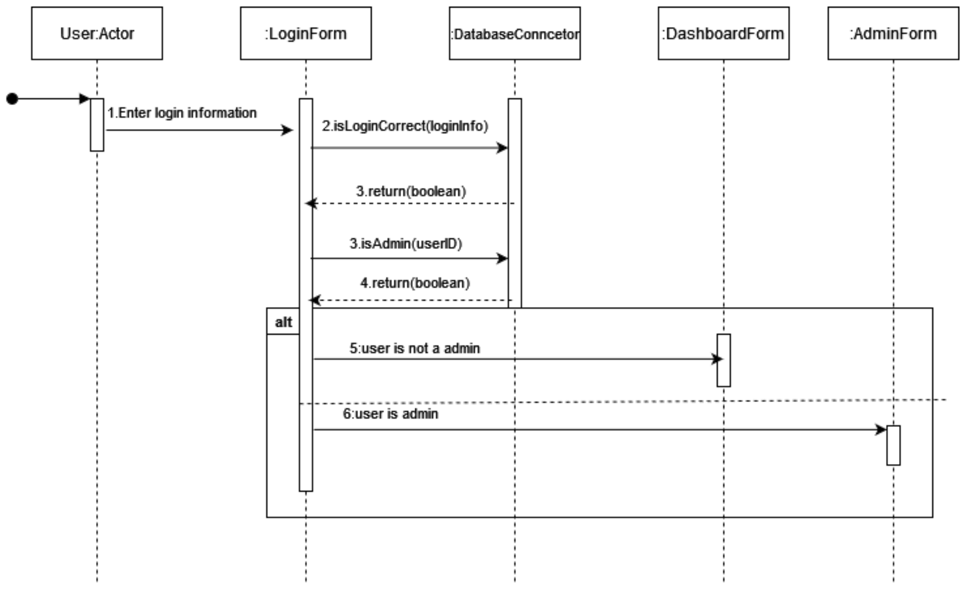
5.2.1 Use-case User login



The user has two options on how to access the application. He or she can access the application by entering the email address or username with the fitting password. When the user clicks on the login button, the application compares the given input data with the data stored in the database. If the password, username or email is not matching with the dataset entry in the database an error prompt is displayed on the screen. If a patient did not create an account yet, the create new account button redirects the user to the registration panel.

(Ali Al-Haidary 1353109)

5.3.2 Sequence diagram User login process

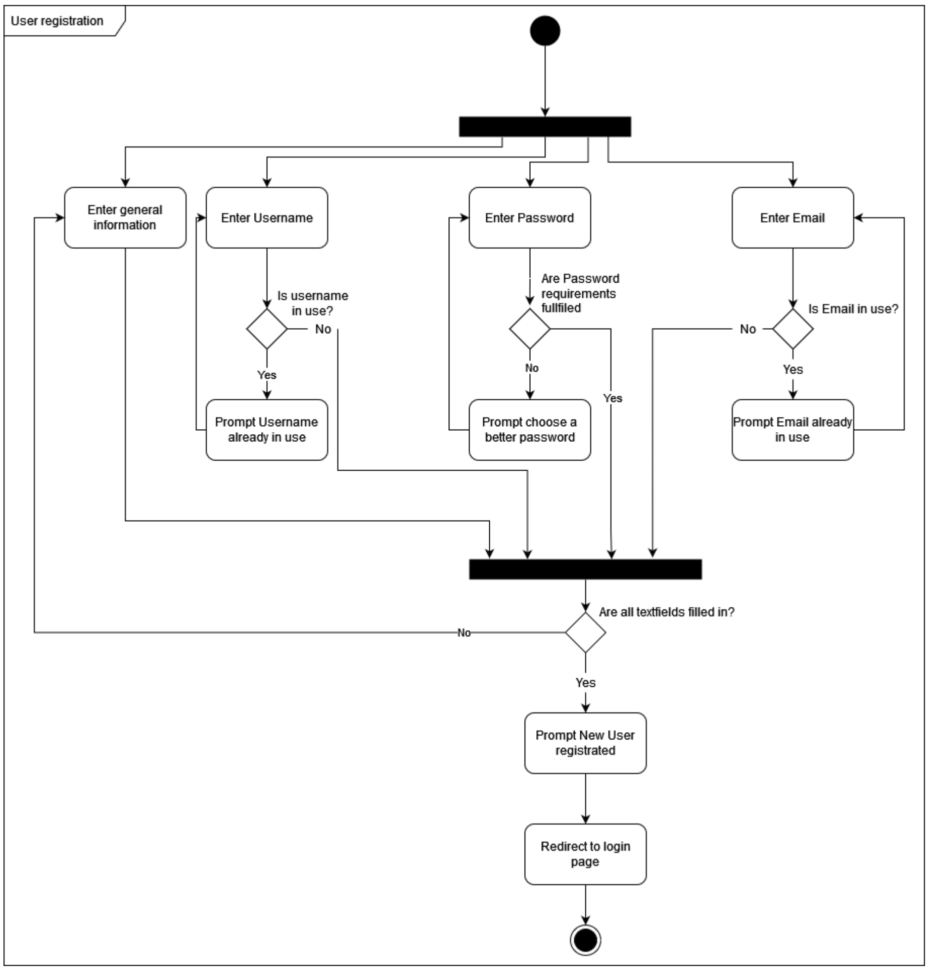


The user enters the login information. The loginForm class sends a request to the database with the method of the class databaseConnector isLoginCorrect(loginInfo). If the user entry is found in the database, the function will return true. After that the loginForm asks the databaseConnector via the isAdmin() function if the user is an admin or not. If that is not the case the user will be redirected to the patient dashboardForm. If the return value is true, the user gets redirected to the adminForm.

(Ali Al-Haidary 1353109) (Elia Funk 1367315)

5.3 Registration Process

5.3.1 Use-case User registration


User registration
Enter general information
Enter Username
Enter Password
Enter Email
Is username in use?
No
Yes
Prompt Username already in use
Are Password requirements fullfiled
No
Yes
Prompt choose a better password
Is Email in use?
No
Yes
Prompt Email already in use
Are all textfields filled in?
No
Yes
Prompt New User registrated
Redirect to login page

By clicking the create new account button in the login frame, the user gets redirected to the registration page. There the user needs to fill in all the displayed text fields. The most important ones are the username, password and email text field. If a text field is not filled in correctly, meaning that the text field requirements are not fulfilled (e.g., the password is not strong enough) an error prompt is getting displayed on the screen.

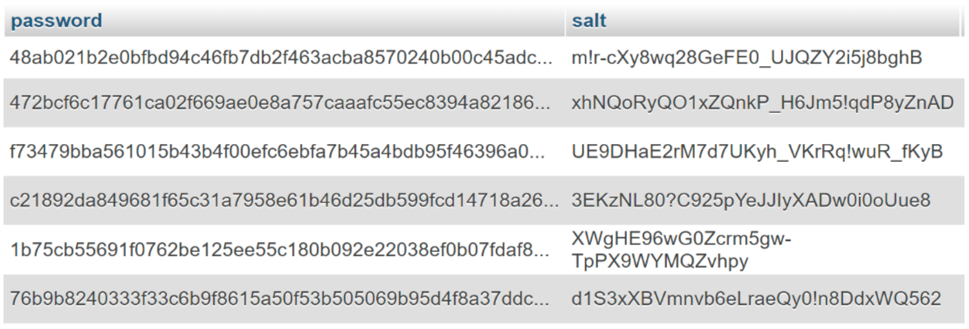
Every text field needs to be filled with an input. If that is not the case an error message is displayed on the screen. Another requirement for a successful registration is that the username and the email are unique. Therefore, a post request is created which will verify if the username and the email are already in use. At the end of the registration process the user will be redirected to the login frame.

(Ali Al-Haidary 1353109)

5.4 Encryption and User authentication

Because the trust of our Users is very important for us, security is one of our priorities. That’s why every password that gets added to our database is combined with a globally unique 32-char (that’s even overkill) salt and gets hashed afterwards. The hashing algorithm we use is SHA-256. After hashing the password together with the salt, the corresponding hash and the salt are added to our Database. The password the user entered on their keyboard never leaves their system, and even we the administrators don’t have a way of knowing what their password is. [4][5]

5.4.1 Database entries in our database



password	salt
48ab021b2e0bfbd94c46fb7db2f463acba8570240b00c45adc...	m!r-cXy8wq28GeFE0_UJQZY2i5j8bghB
472bcf6c17761ca02f669ae0e8a757caaafc55ec8394a82186...	xhNQoRyQO1xZQnkP_H6Jm5!qdP8yZnAD
f73479bba561015b43b4f00efc6ebfa7b45a4bdb95f46396a0...	UE9DHaE2rM7d7UKyh_VKrRq!wuR_fKyB
c21892da849681f65c31a7958e61b46d25db599fcd14718a26...	3EKzNL80?C925pYeJJlyXADw0i0oUue8
1b75cb55691f0762be125ee55c180b092e22038ef0b07fdaf8...	XWgHE96wG0Zcrm5gw-TpPX9WYMQZvhpy
76b9b8240333f33c6b9f8615a50f53b505069b95d4f8a37ddc...	d1S3xXBVmnvb6eLraeQy0!n8DdxWQ562

When the user tries to log in, first the salt and the hashed password are fetched from the database where either the username or the email used for logging in matches. After that the salt is appended to the password, the user put into the password field, and this combined long String is hashed. If the now gotten hash matches the hash stored in the database, we then now that the user used the correct password when logging in.

Example:

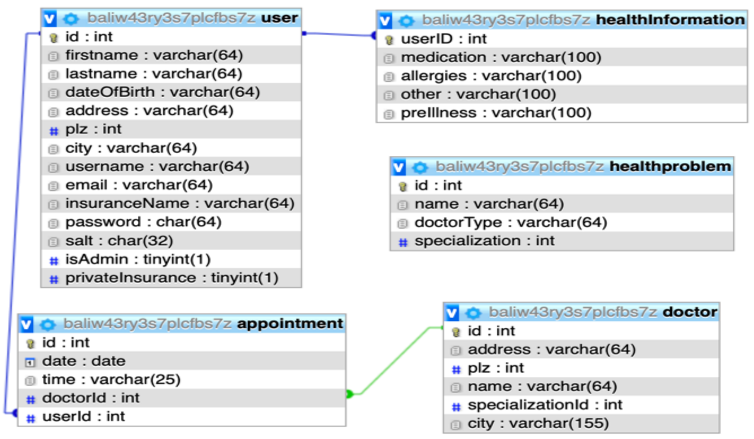
My password is password, the salt is a randomly generated 32 characters long String starting with UE9DHa(...).

Hash(password+UED9Ha(...)) = f73479bba...(The resulting hash is 256bits or 64 characters long) <- This is getting stored in the database together with the salt. Now it’s almost impossible for an attacker to guess what my original password was because only the correct password together with the salt would create the same exact hash the original password did.

(Elia Funk 1367315)

5.5 Database

5.5.1 Database model



baliw43ry3s7plcfbs7z **user**
- id : int
- firstname : varchar(64)
- lastname : varchar(64)
- dateOfBirth : varchar(64)
- address : varchar(64)
- plz : int
- city : varchar(64)
- username : varchar(64)
- email : varchar(64)
- insuranceName : varchar(64)
- password : char(64)
- salt : char(32)
- isAdmin : tinyint(1)
- privateInsurance : tinyint(1)

baliw43ry3s7plcfbs7z **healthInformation**
- userID : int
- medication : varchar(100)
- allergies : varchar(100)
- other : varchar(100)
- prellIness : varchar(100)

baliw43ry3s7plcfbs7z **healthproblem**
- id : int
- name : varchar(64)
- doctorType : varchar(64)
- specialization : int

baliw43ry3s7plcfbs7z **appointment**
- id : int
- date : date
- time : varchar(25)
- doctorId : int
- userId : int

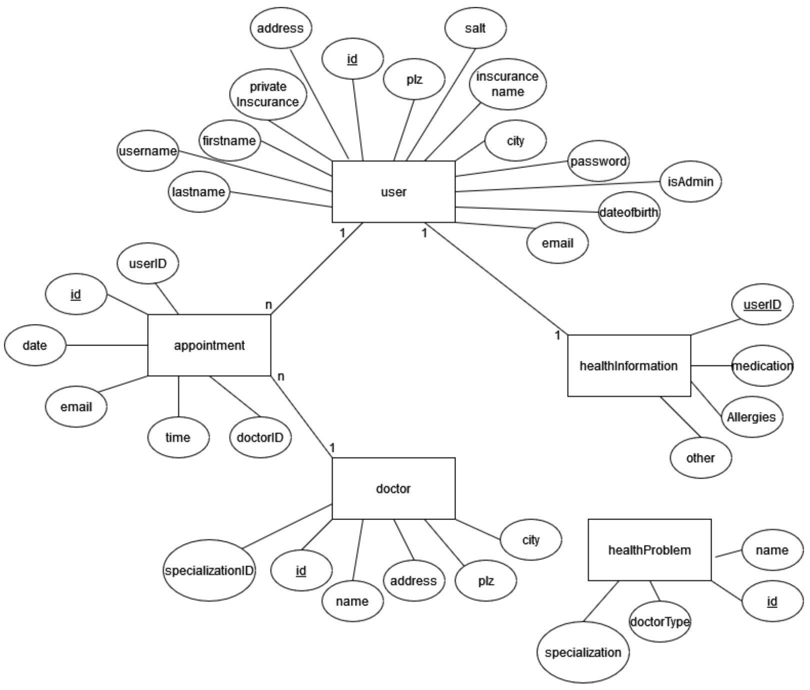
baliw43ry3s7plcfbs7z **doctor**
- id : int
- address : varchar(64)
- plz : int
- name : varchar(64)
- specializationId : int
- city : varchar(155)

At the beginning of our project, we made the decision to use a web hosting service for our database. Therefor we agreed on using clever cloud as our database management system. To establish a connection, we used JDBC Connector.

The database consists of 5 tables: user, appointment, healthInformation, healthProblem and doctor.

(Ali Al-Haidary 1353109)

5.5.2 Relational model of our database



(Ali Al-Haidary 1353109)

User-Appointment: 1:n relation

User (ID, firstname, lastname, username, privateInscurance, address, plz, salt, inscuranceName, city, password, IsAdmin, dateOfBirth, email)

Appointment (ID, date, time, !doctorID, !userID)

A user can create several appointments and every appointment has only one user assigned to it. The appointment entity got the userID as a foreign key.

User-healthInformation: 1:1 relation

User (ID, firstname, lastname, username, privateInscurance, address, plz, salt, inscuranceName, city, password, IsAdmin, dateOfBirth, email)

HealthInformation (!UserID, medication, allergies, other)

A user has only one health information. That means the attributes included in the table healthInformation are connected to the specific ID of a user.

Appointment-Doctor: n:1 relation

Appointment (ID, date, time, !doctorID, !userID)

Doctor (ID, address, plz, name, specializationID, city)

Every appointment has one doctor assigned to it, but a doctor can be assigned to more than one appointment.

HealthProblem:

Healthproblem (id, name, doctorType, specialiatzion)

(Ali Al-Haidary 1353109) (Elia Funk 1367315)

5.6 Admin operations

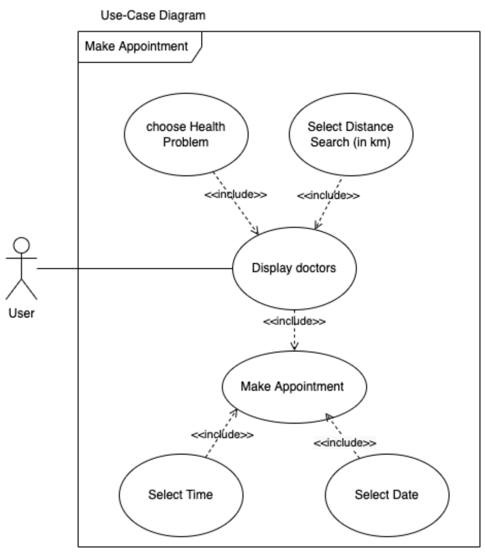
When an admin logs in and the input data matches with the admin account stored in the database the user is redirected to the admin dashboard. There he has the option to log out or press the button “admin” which will redirect the user to the show all user's panel. All users which are stored in the database will be displayed in a table. By clicking on one specific entry in the table all user data will be displayed in the corresponding text fields. The admin can change every entry except the userID. By pressing the button update an updated request will be send to the database and the data is changed. Because of data security the passwords are getting displayed in the hashed and added salt format. We chose this approach to ensure maximum user security and prevent data abuse. The admin still has the option to change the password. The new password gets automatically hashed and a salt is added.

The admin also has the option to delete accounts by clicking on one displayed dataset in the table and clicking the button delete.

(Ali Al-Haidary 1353109)

5.7 Make an Appointment

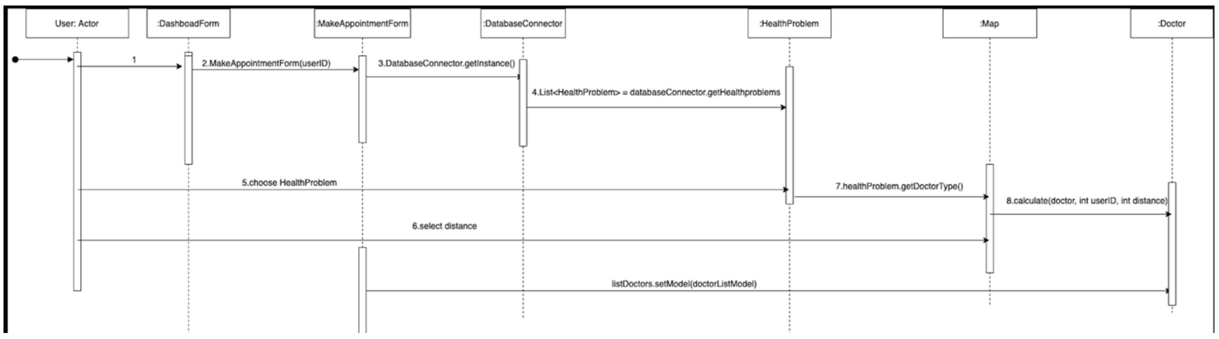
5.7.1 Use Case make an Appointment



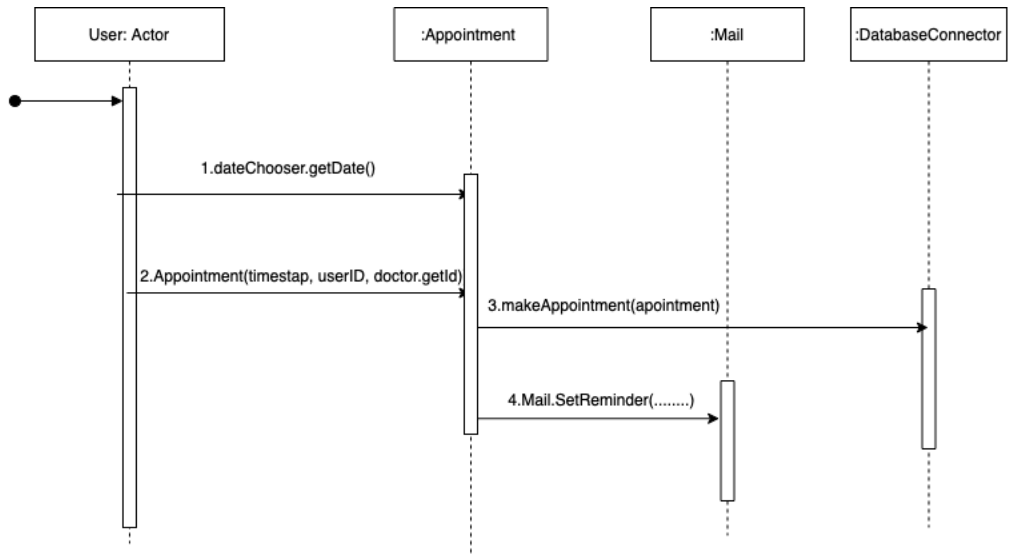
To make an appointment, the user must choose a suitable doctor. To find a suitable doctor, the user first needs to specify his or her health problem. The problems are displayed in a list. In addition, the user must indicate the maximum distance to the doctor in kilometres. After the input, the matching doctors are filtered out and displayed in a list.

Now the user can select a doctor. Before he can make an appointment, he has to enter the date and time of the appointment. Then he can create an appointment and a confirmation e-mail is send automatically to the user.

5.7.2 sequence diagram make an appointment


User: Actor
:DashboadForm
:MakeAppointmentForm
:DatabaseConnector
:HealthProblem
:Map
:Doctor
1
2.MakeAppointmentForm(userID)
3.DatabaseConnector.getInstance()
4.List<HealthProblem> = databaseConnector.getHealthproblems
5.choose HealthProblem
7.healthProblem.getDoctorType()
8.calculate(doctor, int userID, int distance)
6.select distance
listDoctors.setModel(doctorListModel)

This sequence diagram shows how the user finds a doctor who is in his desired radius and is help for his health problem.


User: Actor
:Appointment
:Mail
:DatabaseConnector
1.dateChooser.getDate()
2.Appointment(timestap, userID, doctor.getId)
3.makeAppointment(apointment)
4.Mail.SetReminder(........)

This sequence diagram shows how the user make an appointment with their selected doctor.

(Muhammed Calli 1363678)

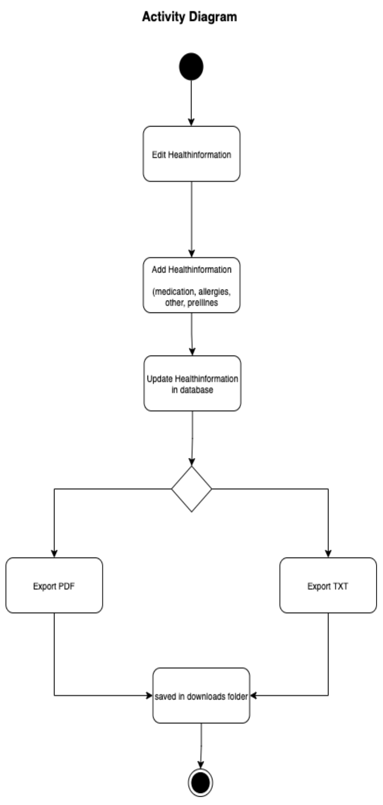
5.8 Shift or Cancel Appointment

In the dashboard the user also has an option to cancel/shift appointments. Clicking the Cancel/Shift Appointment button opens a new window displaying all current upcoming appointments for that user. He can then select an appointment and cancel it. After that he receives an email indicating the appointment was cancelled. If he clicks on shift appointment, he has the option to move the time and date of the appointment and gets an email if the shift was successful.

(Gregor Napierski 1219667)

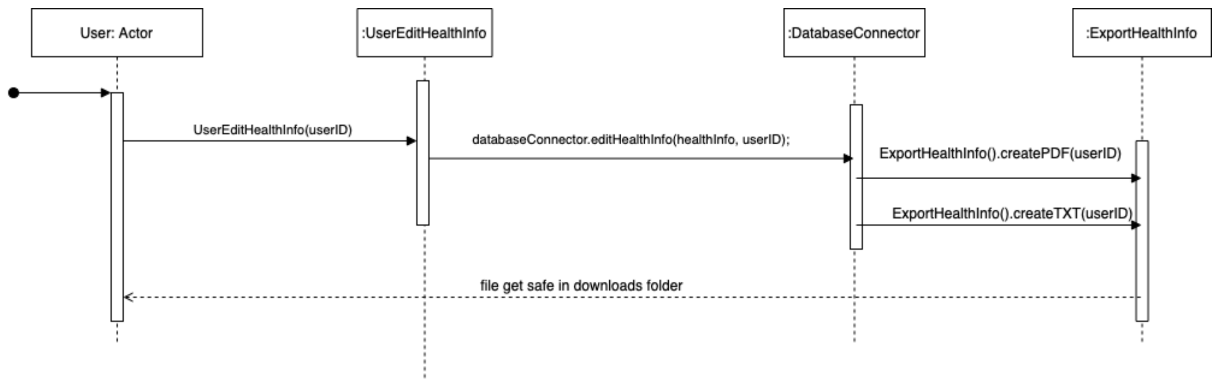
5.9 Export the health information as a PDF or TXT file

5.9.1 Use Case Export the health information


Activity Diagram
Edit Healthinformation
Add Healthinformation
(medication, allergies, other, prellines
Update Healthinformation in database
Export PDF
Export TXT
saved in downloads folder

This activity diagram shows how the user can edit their health information and then export it as a Pdf or TXT file. In the dashboard, the user presses the button edit health information. There the user selects the information (medication, allergies, other, bouncing). With the button update, the information is then also saved in the database. In addition, the user can choose to export the health information as a pdf or txt file. The information is then saved in the Downloads folder

5.9.2 Sequence diagram Export the information


User: Actor
:UserEditHealthInfo
:DatabaseConnector
:ExportHealthInfo
UserEditHealthInfo(userID)
databaseConnector.editHealthInfo(healthInfo, userID);
ExportHealthInfo().createPDF(userID)
ExportHealthInfo().createTXT(userID)
file get safe in downloads folder

The user calls the function UserEditHealthInfo(userID) with the button edit Health Information and userID is passed. If the user changes something in the health information, the change is also saved in the database with the databaseConnector.editHealthInfo(healthInfo, userID). Then the user can call the methods ExportHealthInfo().createPDF(userID) or ExportHealthInfo().createTXT(userID) with the buttons Export PDF or Export TXT. With these methods, the health information entered is created as a document and then saved in the Downloads folder.

Muhammed Calli (1363678)

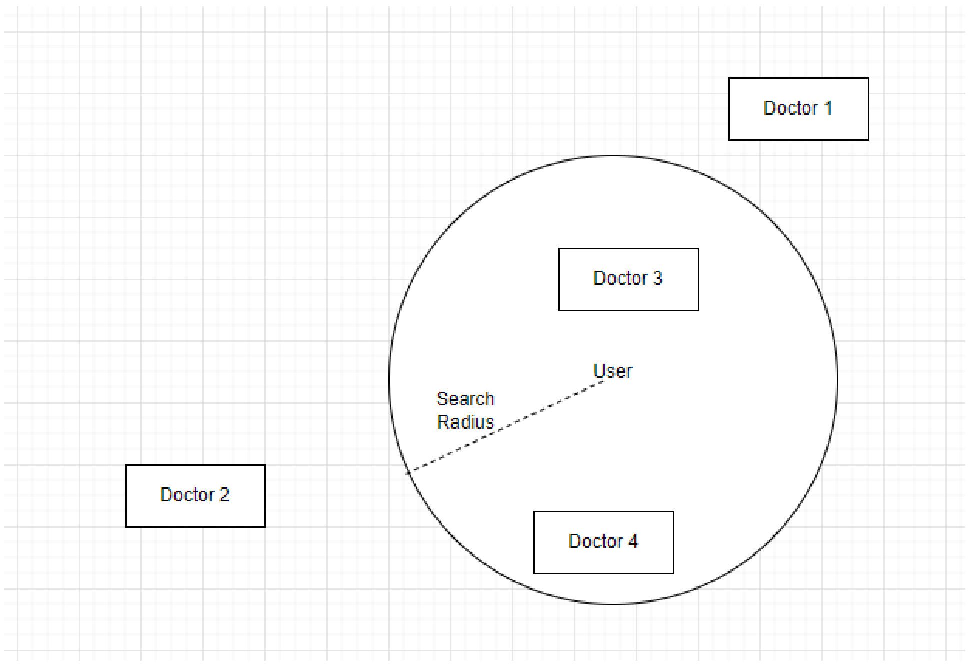
5.10 Map

The Map class contains methods to convert an address and an area code into latitude and longitude coordinates. To achieve this, a HTTP request is created and sent to openstreetmap.org. The response is a json Array containing all available information on the location including the latitude and longitude coordinates.

The coordinates are used in another method called getDistance which performs several mathematical operations on these coordinates to get the distance between two points in either kilometres or miles depending on the unit given.

These methods are used in the application form when displaying doctors inside the given radius from the user. Doctors outside the radius, meaning distance between the user's location and the doctor are not displayed.

The representation below shows how the calculation works. First, the search radius given by the user is used as an upper bounds to the doctors that are shown. After the distance to each doctor is calculated, only doctors within the circle remain. Then those doctors get filtered by their specialization according to the specialization selected by the user (I.e. cardiology) and the final list of doctors is then displayed in a selection box. In this case doctor 3 and 1 are cardiology specialists so only doctor 3 is displayed.



(Gregor Napierski 1219667)

6 Solution of challenges

6.1 How to pass the user information between frames

At the beginning of our project we did not really know how we should edit user specific database entries and how we would determine which user is logged in now. We tried to create a session, but we found another simpler solution. The whole Graphical user interface is implemented in java swing. Every time a new frame opens the unique userID which is stored in the database is fetched and transmitted to the opened frame. This process begins after a user is logged in successfully. After that every time a new frame opens the user id is send to the new frame.

(Ali Al-Haidary 1353109)

6.2 Calculating distance of search

Calculating distance between doctors and the patient has been a challenge for the whole team. First we tried to use google Street maps API to get the distance automatically but the documentation was confusing and creating an API key was not easy. After some research we finally decided to use openstreetmaps instead. By using a HTTP request we could get coordinates for any address written in the URL. These coordinates (in latitude and longitude) could be used to calculate the distance. A series of mathematical operations are necessary to calculate the distance between two coordinates. We found these by googling.[3]

(Gregor Napierski 1219667)

6.3 Export into specific folder

We had the problem that the exported pdf and txt files were saved directly in the project folder by default. We changed that by using the system.getProperty() method. Doing that and adding the parameter “user.home” will send the exported pdf and txt files into the home directory. [6]

(Ali Al-Haidary 1353109)

7 Team organization

Task	Elia	Ali	Muhammed	Gregor
Creation and design GUI		X		
Login and Registration process	X	X		
Encryption	X			X
Admin operations	X	X		
Edit Health Information of the user	X	X	X	
Make new Appointment	X		X	
Get distance between patient and doctor	X		X	X
Automatic email function	X	X		X
Reminder email functionality	X	X		
Shift and cancel appointments	X			X
Export as pdf or txt	X		X	
Database creation and population	X	X	X	X
Project structure and class design	X	X		X
Write documentation	X	X	X	X

8 Conclusion

This project was a good learning experience for the whole team. We went through all the phases of software development from concept to finalizing the code. We got an insight into software engineering and faced many challenges during the process.

We had to research several new concepts in varying fields. I.e. a timed email reminder functionality for appointments, implementing a search radius based on real location data, database and GUI interactions or securely hashing and retrieving a password in a database.

Working in a group also showed the necessity of using an online source control software (Git) since everyone was working from home and in their own time. All in all we learned a lot that we could use in real world software engineering.

(Gregor Napierski 1219667)

9 References

Github Link: https://github.com/gnap1/JavaProjekt

https://www.stackoverflow.com

[3] https://www.movable-type.co.uk/scripts/latlong.html

[4] https://www.geeksforgeeks.org/sha-256-hash-in-java/

[5] https://stackoverflow.com/questions/7111651/how-to-generate-a-secure-random-alphanumeric-string-in-java-efficiently

[6] https://stackoverflow.com/questions/585534/what-is-the-best-way-to-find-the-users-home-directory-in-java/586345