Group 11
Jon Heikkinen
Alejandro Cabral
Michael Chau
Alen Peter

Testing and Inspection Report Summary

Project overview/domain

Doctor's office is an application that allows patients to seamlessly book appointments with any specialist within their network, view the locations of each specialist, as well as easily see both their upcoming appointments as well as the location of that appointment. In order to implement most of the functionality the team used a MYSQL schema containing three tables: 'patients', 'medicalprofessionals', and 'appointments'. This is where all the information used throughout the application is stored. The patients and medicalprofessionals tables are also used to differentiate which login portal should be displayed when the user logs in.

Testing

For testing we wanted to make our tests as wide covering as possible and we designed our components tested around that desire. Many of our tests revolved around our database because that is where the majority of our information is fetched from. We extensively tested multiple different users and user types in order to make sure that the program was working for everybody, and that there were no issues seen in some users that weren't present in other users.

Test results

By the end of the project we were able to have each of our test cases result in a PASS, and so we are feeling pretty confident in where our project is in terms of bugs. Luckily most if not all of the tests had easy verification of results and so were easy to see whether they passed or not.

Inspection

For inspection procedures we made sure that team members that were not involved in the creation of the feature were inspecting it, in order to eliminate any bias and also allow fresh eyes to take a look and find anything the original creator may have missed. This also allowed each member to get exposure to multiple parts of the project and have a deeper understanding of how it works in all phases.

Project issues

Open issues for the project include OS compatibility issues, Threading, and communication with hospitals and pharmacies. The google API seems to run into more issues on Windows OS than Apple OS. The system would crash more often on Windows as well as run

slower. This program was not originally designed with multithreading in mind and as such the code was not tailored for that implementation. So if that functionality were to be implemented it would probably require a non-trivial amount of changes to the code base. As for communication with hospitals and pharmacies, this would require a collaboration with the targeted hospitals and pharmacies in order to gather the required information of both the specialist doctors, as well as the available supply of pharmaceuticals from the pharmacies.

Comparison with original project

The original project included functionality for nurses to also log in and be able to assist both doctors and patients, as well as patients being able to check stock of their prescriptions at their preferred pharmacies within the network. Due to time constraints this functionality was not able to be implemented. This project added the google maps functionality that allows patients to see where the appointments they've booked are located, as well as where the specialists they are looking to see are located.

Project deliverables

For the first release the main objective was to establish a database that would allow the creation of both patient and doctor objects. This was the release that would set up the rest of the project down the line, as without a database connection and proper implementation of account creation we would not be able to move into the next steps. For the second release our main goal was to get the implementation of the separated patient/doctor portals which would contain different features, as well as the ability to actually log-in to the application. This release also included the addition of the appointments table which is what allowed patients and doctors to actually book/cancel and see their appointments. Most of the features we had discussed were implemented in this release.

Project retrospective

Things that went well with our project included the database implementation and our use of XML files in order to design the GUI. We implemented a database utilizing JDBC and SQL and it worked very well for the project. The XML files gave us a lot of control on what each page would look like from a front-end perspective, and allowed us to implement all of the functionality that we desired. If we were going to do the project again I think we would've kept both the database and GUI implementation. Things that did not work out well was the webview that we used for the Google Embedded Maps API. This is because there were quite a few bugs that arose from this implementation, including the project completely crashing upon searching for a hospital location. The webview also felt delayed and not as smooth as we would have liked. If we were to do this project again we would have gone a different route for the Google API. Overall we are happy with the product that we have delivered and feel given more time we would have been able to get to the other features we had originally planned.