Health Provider Application

Jackson Kimball and Ryan Pufall

1. Introduction and Background

For our final project, we created an application for Android using the Ionic framework that would be used by a doctor in a hospital. The project was to create an application that a doctor could log into and manage and view information that they would have. Information would include patients, tests, allergies of patients, when their last test was, etc. This project came from a couple of graduate students that provided examples of projects for the class to do that if done how requested, could help them with their final projects for the year. Initially, this project was a client/server-based rule editor, but about a week and a half before the due date, we were informed that this would be too difficult to complete, and we were given the project that we did.

1.1 Problem Statement

The problem that we worked on for our final project was the development of an Android application that can be logged into by a “doctor” and browsed for information including the patients of that doctor, a history of tests completed on that patient, list of allergies for each patient, information about the patient, and information about when each test was completed and the results of that test. Asked of us to do was make the application, connect it with a Jetstream server, and to do a few user interface quality of life features. Quality of life features included; recent tests, ease of navigation, and a clean appearance. In regard to recent tests, we were asked to add a list in the home page of the doctor that would show tests that the doctor has not yet viewed, and upon viewing, would be removed from the list. Second, we were asked to make the application easy to navigate through, Ionic assisted us in this because it already had a back button feature that used a stack of previous pages, but to navigate through the website otherwise we made items in lists clickable so that it is easy to get to the next page. Lastly, we used CSS, a style sheet language used for describing the presentation of a document written in a markup language like HTML, to make the application visually appealing.

1.2 Previous Work

Previously, many applications have been created using Ionic, the most common being; MarketWatch, Pacifica, and Swordkit [1]. MarketWatch operates a financial information website that provides business news, analysis, and stock market data. Pacifica is a mental health app that offers numerous ways to handle stress. Lastly, Swordkit is an application that is used for fitness, it includes a wide range of exercises and workouts. All of these apps use Ionic because of the ease of making it compatible on IOS and Android.

Jetstream is a Java in-memory database engine. Jetstream claims to be the only one database which stores data exactly in the same way as Java does itself. They also claim to be faster with no mapping, do not require a query language, and have no network bottleneck. Previous work in Jetstream was hard for us to find which is why we switched to a MySQL database. MySQL is an open source relational database management system. It is also something that we have used before and is more widely known so there is more information online about it. Previous work that we have done in MySQL includes a database project in a class a couple years ago and during the summer one of us used it a lot at an internship.

1.3 Background

Our goal in our project is to create an Android Ionic application that allows a doctor to login, navigate through a database, and have a friendly and intuitive user experience. We plan to make it possible to view any information that is in the database in a manner that makes sense and also have some information visible without needing to dig through the database to find it, for example, new tests should be visible right after logging into the system. In addition to using Ionic, we plan to use MySQL instead of Jetstream for our database because it has been used before and is well documented as opposed to Jetstream, which is a very poorly documented database service. In addition, we want our application to be able to run on Android mobile devices.

2. Project Description

To do this project, we were asked to use the Ionic framework to create an Android application that could be used by a health provider. Doctors would have accounts that could log in and view essential information like unread tests, patient information, and all tests per patient. We were also asked to use Jetstreamdb, a database storage engine. We tried to use this tool for a long time, but eventually switched to a mySQL database because unlike Jetstreamdb, there is a helpful guide and past work on mySQL.

2.1 Functional Specification

We have many functions that we would like our application to be able to perform. Functions include, logging into the system, displaying unread tests, viewing tests, and viewing a list of patients. The first function that we implement is the ability to login to the application. In our database there are currently two doctor profiles, each with their own username and password, and entering their credentials correctly gets the user in. This function creates security so that patient information can be protected. The second function is displaying a list of unread tests on the home page for ease of access. Also, we would like to make it so that the user can easily navigate between pages to view patients and tests.

2.1.1 Functions Performed

Our project meets some of our functional specifications. We have made it so that it is possible to easily navigate between pages to view tests and patients. And connect to a database to display information from that database onto various pages. We created a login that requires a username and password, and created an unread tests list, but those are not fully implemented and are just laid out for the time being ready to be implemented by the graduate students, should they choose to build off of what we have done for this project.

For ease of navigation between pages, we use Ionics clickable button features. In Ionic, you can create what is called an ion-button and clicking on that calls a command, for our buttons the command redirects you to the page the button displays. Our application contains 3 different buttons. The first button is the login button, the login button is on the first page of the application, and must be used to enter. The next button in the application is the patient. Displayed on the doctors home page is a list of patients that can be searched through, and each patient in the list is shown by name and age, and can be clicked on to dig deeper into the application. Once on a patient page, the application displays the name of the patient, followed by an allergy list, followed by a list of tests performed on the patient, then lastly by ways to contact the patient. Which brings us to our last button, the patient tests list is clickable, and clicking on a test brings you to the test page where information about the tests is shown.

Another function of our project is the connection to a mySQL database. Connecting to the database is what allows us to display all of the information on each page that we currently display, including names, ages, tests, allergies and more. Connecting to the database was a large task because we initially were asked to use Jetstreamdb as our database software. Unfortunately, Jetstreamdb was not only poorly explained online with a useless guide that was provided to us, but it was also not really used by anyone. We could not find anyone that had previously used this database engine nor could we figure out how to make it work for this project. MySQL is a database engine that is well used, well explained, and convenient.

Our next specification is our unread tests list. Currently, the layout for the recent notifications list is all laid out but is not fully implemented. To implement this list, one would need to add a new row to the tests table named something like unread or recent, that would be a Boolean. This value would be used to determine if it should be displayed in the list or not, and once it has been viewed, should be changed to not show up in the list in the future. If we had more time to work on the project we would have worked on this feature, but since we only had about a week it was not very high in the priority list.

Lastly, the secure login is a feature that was requested of us to make. The secure login is just the use of database values username and password to authenticate a user. This was also not high on our priority list like the unread tests list, so like that list, this is not fully implemented. Each user in the database already has a username and password, the only thing that would need to be done to fully implement verification would be to check the database for the username and password from the fields on the page before continuing to the doctor home page.

2.1.2 Limitations and Restrictions

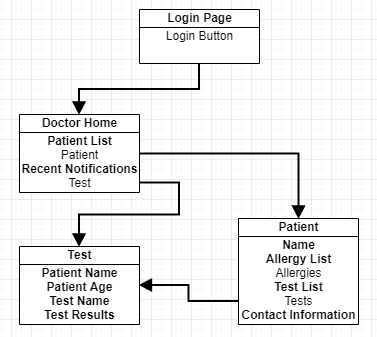
One of the big limitations and restrictions on the project was the fact that we didn’t know what the final data our program would interact with was like. This meant that a lot of our work was focused on creating a UI based on a rough idea of what was required and working with mock data. This may mean that certain aspects of the UI will need to be redone in order to comply with the requirements of the real data sets.

Another limitation was the fact that our team was two individuals instead of the max of four. This means that we weren’t able to accept as big of a project scope, however I think given the team size and project scope given to us, this wasn’t a major limiting factor.

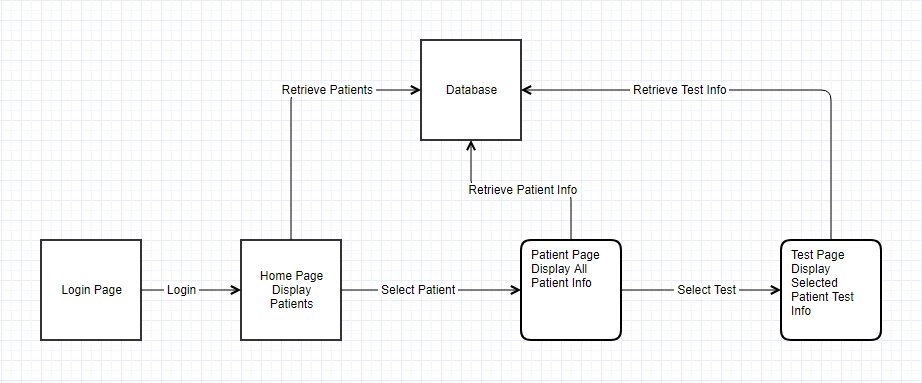
2.2 Design Specification

2.2.1 System Data Flow Diagrams

When running our application, as shown on the image, you begin on the login page. From the login page you can login and reach the Doctor page. From there the user can either select a patient to view and move to the patient page, or they can select a recent test and go straight to the test page. If the user goes to the patient page, they can then make their way to the test page by selecting a test.



2.2.2 System Structure Chart



2.2.4 Equipment Configuration

For the majority of development, a Windows 10 PC utilizing an intel processor, standard SATA hard drive and 16GB of RAM was the extent of the hardware. There was also an android virtual device used, specifically an emulator for the Pixel XL android phone.

2.2.5 Implementation Languages

For the server, an application called WAMP was used, which essentially sets up an easy to use server/database dev stack. The stack WAMP sets up utilizes apache, mysql, php and phpMyAdmin.

For the application the ionic framework was utilized, which essentially consists of ionic snippets, typescript, CSS and HTML, with php being used to communicate with the database.

Additionally, the cordova debugging tools were used in the testing of the application.

2.3 Implementation

2.3.1 Project management

The project code was managed in bitbucket which is an extension of github. As for deciding what each member was going to work on, the two members of the project didn’t have to do much planning on communication, since they are roommates and can just ask each other questions at any time during the project.

Other than utilizing github to manage shared code and communication between the two of us during development, there was not much management.

2.3.1 Deliverable Items

The deliverables for this project include this project report outline, the code from the database, the php files, the source code for the application, a readme file and two videos showcasing our project in the likely event that the environment for the application cannot be set up.

3. Results and discussion

In the end we have application source code that provides a functional demonstration of our UI as well as a functional backend to show that we can successfully connect the two and display data on our application. We have some areas that are lacking, such as no login data, no direct functionality for recent notifications and a non-functional search feature on the patient list. This project was meant primarily to serve as a demonstration of a UI as well as practice for creating a backend and connecting the two; I believe the project fulfills that purpose.

4. Conclusions

Overall, given our timeline for the project, we are happy with the progress we have made. With a little more time there are just a few things we would like to implement better, however it is satisfactory as is. Looking back, I think the biggest thing we would have changed is to not bother with trying to use jetstreamDB. Since we are students, utilizing an extremely new or underused technology like jetstreamDB was a waste of time. There simply wasn’t enough reference material referring to jetstreamDB for us to be able to quickly utilize it given our timeline.

5. References

[1]. <https://csform.com/top-10-apps-built-with-ionic-framework/>

[2] <http://masteringionic.com/blog/2016-12-15-using-php-and-mysql-with-ionic/>