Team Spectrum

Mason Mahoney, Rosa Fleming, Aaron Teague

Overview

Team Spectrum looks to build two mobile applications that will allow doctors who are part of Spectrum's Transplant Surgery team to communicate with doctors outside of Spectrum. Team Spectrum will accomplish this by developing two application using the Android Strack and iOS stack to create the Front-end of the applications and develop the Back-end using ASP.NET and C#. Both application will also utilize the Model View Presenter framework. The applications simplify the process for Spectrum doctors to make referrals to doctors outside of Spectrum as well as make easy for Spectrum doctors and outside doctors to communicate about status updates on patients, referral questions, and professional advice.

Languages

The two native apps that we will be developing in Android Studio and Xcode will be using Java and Swift programming languages, respectively. To complement the front-end technology stack we will also be using HTML/CSS/Javascript web languages to implement the API called TransplantReferral.

In the back-end of our technology stack we will using .NET Core MVC which will be using the C# programming language. Along with that we will be using a little bit SQL as well to pull information from databases such as Microsoft SQL server.

Framework, Libraries, and APIs

Accordingly at Spectrum requests two mobile applications will be created. One will need to run on the Android OS and one will run on IOS. The Android application will be created in Java using Android Studio 3.1.4. The IOS version will be in Swift using Xcode. The supporting backend will be developed in ASP.NET and C# using Visual Studio. The project will include an API called TransplantReferral, provided by the Spectrum Development Team, that will be separated into four parts TransplantReferralAPI, TransplantReferralAPI.Core, TransplantReferralAPI.Data, and TransplantReferralAPI.Services. TransplantReferralAPI will handle all web related aspect of the project this includes HTML views, controllers to provide data, the CSS files, and JavaScript files. TransplantReferralAPI.Core will control and handle utility class files will be share through out the project. TransplantReferralAPI.Data will handle all tasks pertaining to data storage and data retrieval, all backend databases will be stored here along with all classes that need to access the database. Finally TransplantReferralAPI.Services will handle the business logic, this includes algorithms and filtering data, of the applications an example of this will be transforming the raw data from the database into actual meaningful objects. We will also be implementing a firebase framework to keep track of users and grant them certain permissions.

For the Android mobile app we will be using retrofit for networking at http protocols along with dagger2 for dependency, and support libraries will be built in. To touch more on our API we will also be using stylecop, and an entity framework.

Code Repository Organization

Our main code repositories will al be hosted on Azure DevOps which is the new and improved VSTS. This is an application that has been set up and is controlled by Spectrum. Within these repositories will be Swift code for the iOS native app, Java code for the Android native app, and HTML/CSS/JavaScript for the web API. Each repository will also have sub repositories for each individual component that needs to be implemented into the apps.

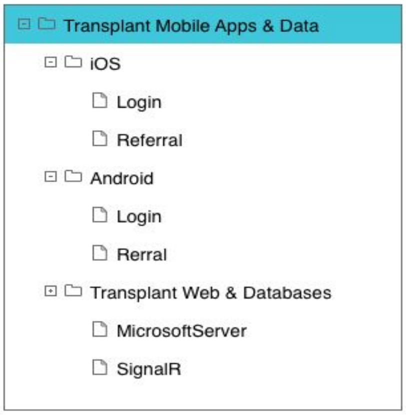


Figure 1: Database Structure

Database Organization

We will create one database that will contain three tables. One table will represent the Users, this table will represent the doctors who are a part of the Spectrum network. Another table will represent Transplant Specialists, these are doctors who aren’t part of the Spectrum network. Finally one table will represent the Super Administrators, these will be system administrators who will be able to monitor conversation on the app and web portal and control the privileges of all end users. We will generate the Database with the Entity Framework and house it on Microsoft SQL Server.

System Organization

First when registering the application we will need to confirm if they are a real doctor, a spectrum doctor, or outside doctor. Upon a successful login a user will be able to initiate a conversation. Another user using the app will be notified and will be able to respond. The original sender will then be notified of the response. A permission model be implemented to distinguish three groups, users, Transplant Staff, and Super Administrators. Users will only be able to messages that are corresponding to a Transplant Staff. Transplant Staff will be able to see all messages that have been to them. Super Administrators will be given the same privileges as Transplant Staff as well as the ability to change an end user’s privileges. The Spectrum team will create a web portal that only Transplant staff and Super Administrators will be able to login into. The webportal will allow Transplant staff to respond to messages and will Super Administrators to see all users of the app and will be able to change a user’s permissions. On the front end our API will also be using POST and GET protocols to get data from the server that our application is hosted on.

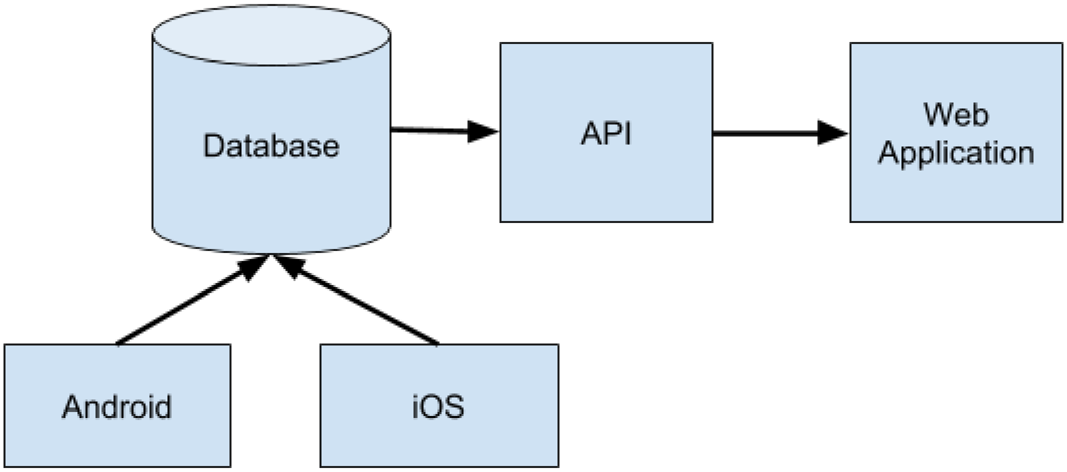


Figure 2: System Organization

Development

The mobile applications that we will be developing in this project will be done in our local environments pulled down from Azure DevOps into Android Studio and Xcode. For each feature we will be using branches off of the master and then merging back into the master as each feature is developed. Our web app is also going to be hosted on Microsoft SQL server and authentication will come from Microsoft Identity.

Non-Trivial Requirements

Chatting with Spectrum

Using their Android or iOS app, a user will be able to download and register for an account in the app. Upon logging in, a user will be able to initiate a conversation with Spectrum Health. A Spectrum Health transplant surgeon, using the same app, will then be notified of the new message, They will be able to respond directly within the app, which will then notify the original sender of the response.

Permissions

A permissions model will need to be implemented for the apps to distinguish between users, transplant staff members, and super users. Users will only see their corresponding message threads. Transplant staff members will be able to see all message threads sent to them. Super Administrators will have the same privileges as Transplant staff members but will be able to change a user's privileges.

Administration Portal

Transplant staff and super administrators will have a web-based portal to sign in to and to respond to messages. Super administrators will be able to see users of the app and will be able to change a user's permissions.

Division of Labor

To maintain continuity, we all be working on each feature the Android and iOS apps. We will be taking a paired programming approach to this project. Each task highly overlays and relies one another to begin creating the next task's functionality so paired programming will allow us to all understand how each feature works and how to approach our next task. Each member will have an equal hand in completing and designing every feature created.

Work Policies

Test Plan

We will be using JUnit to testing for both the Android and XCode applications. By using these testing methods it will allow us to not only check our work but see that our implemented features are working. We will also be using Lint also for code coverage and to make sure our code is written in a professional and proper way. Web testing will be done using Jasmine.

Deliverables and Sprint Plans

Sprint 1

- Create the user login screen for Android.
- Create the user login screen for iOS.
- Design out the database tables for users, transplant staff, and super administrators, and referrals.
- Design the referral form for the transplant doctors.
- Create overall design and layout of each application.

Sprint 2

- Create the referral from for the transplant physicians in the Spectrum organization.
- Successfully retrieve data from the Transplant referral web API.
- Implement database tables in Microsoft SQL Server.
- Begin creating chat feature within mobile applications.

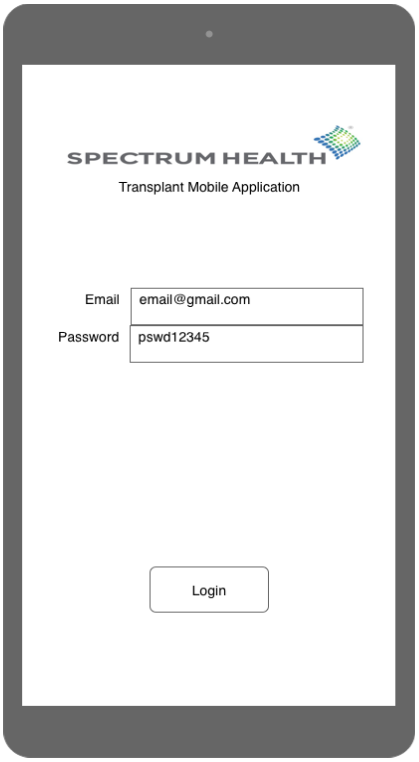
Sprint 3

- Permission model implemented to distinguish between users, Transplant staff members, and Super Administrators.
- Administrator portal which will be web based so Transplant staff and Super Administrators can sign in and respond to messages.
- Notify the doctors that a referral has been sent to them.

- Design touches and brand the app to fit Spectrum Health’s standards.

- Successfully finish chat feature with apps.

Mock Ups:



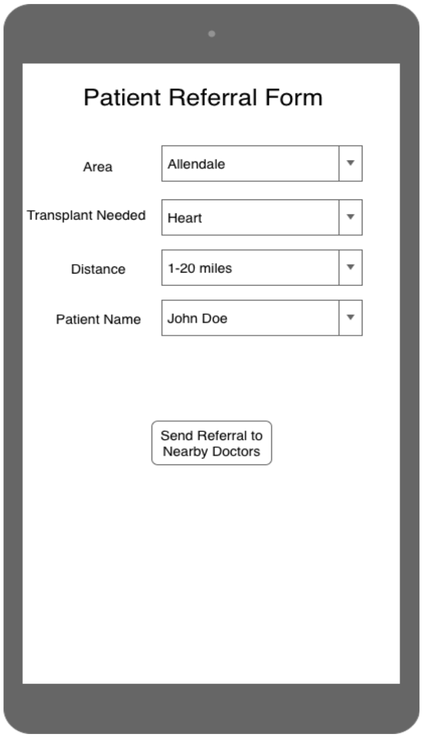
SPECTRUM HEALTH

Transplant Mobile Application

Email: email@gmail.com

Password: pswd12345

Figure 3: Login Screen



Patient Referral Form

Area: Allendale

Transplant Needed: Heart

Distance: 1-20 miles

Patient Name: John Doe

Figure 4: Referral from

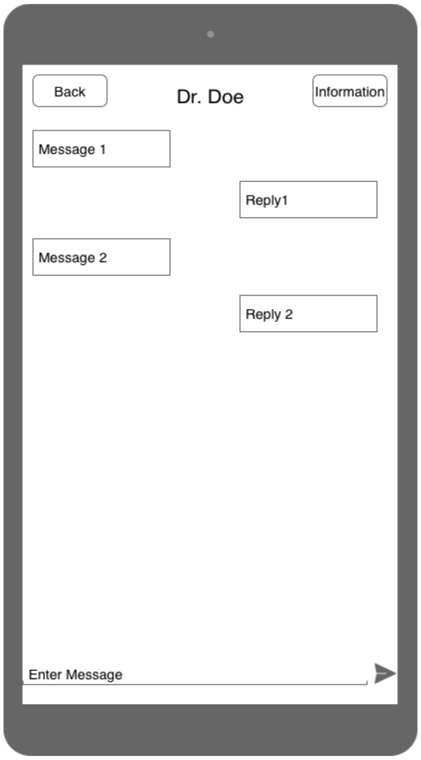


Figure 5: Messaging Feature