Team Spectrum

Design Document

March 24, 2014

Brody Berson, Mario Galeno, Kyle Schelhaas, and Michael Torres

Contents

Programming Languages

Aero Med Spectrum needs a stable platform for their application. It will be used to keep track of very critical procedures they perform during a flight. In addition to being used during their flights, authorized personnel will be able to gather large-scale data on these procedures and their implementations during missions. Therefore the data should be easily accessed and always reliable. Aero Med's preferred platform is the Apple iPad. Therefore, Objective-C will be the only programming language that will be used in the final implementation. CSS and HTML have been used as mockup languages for milestone presentations.

Frameworks, Libraries & API's

The application is an initial application with no predecessors to pull frameworks, libraries, or API's from. The group will do much research to find the appropriate tools to implement the most reliable application for Aero Med. TestFlight has been used to create early prototypes and has worked exceptionally well. We have not used Pixate to style the app using CSS stylesheets. Our hope is that Pixate will make it easier to fix the user interface. With Bootstrap we were able to create a website for presentation reasons. Parse is the NoSQL database that has been adopted for this project. Great documentation and support has helped us when we encounter any problems.

Code Repository

Using Apple's XCode IDE has been great. Git support has been built into XCode and pulling and pushing our updates has been relatively simple. We have had no major problems in merging all of our changes to the project.

Database Organization

Parse allows use us to create a NoSQL database. Currently, we have three classes in the database. One class holds the navigation structure for all the documents. The other two classes contain two different types of documents that we have transferred from the PDF's we received from our client. There are still a few more classes for documents that need to be created.

Client/Server Organization

Currently, the application has to have an internet connection to be able to log in. Once the user logs in all the documents are pulled from the database and saved into the device so that users can access them when they don't have a connection. We plan on having the transport documents being saved to the device and pushed once an active connection has been established.

User Interface

The user interface for the application is information driven. Aero Med personal will search for a common medical condition and perform the necessary procedures that are pulled from the server on login.

When the application is initially opened, the user is shown an option to log in or sign up for a new account. The signup view simply asks for a username, password and email. A valid Spectrum email must be used to successfully signup. In the login view, you can select a username via the scroll list that appears and enter a password with the device keyboard.

Once logged in, the transport screen shows all transports that the account has been on. A new transport can be created with the '+' icon in the upper right corner. A new transport form asks for the transport number, crew and other information and notes about the flight. Once it is created it shall be added to the transport screen. The user can also open up the slide out menu by sliding their finger to the right or clicking on the 'Menu' button.

The slide out menu, implemented with a separate library, has options for the transport screens, documents, trend graphs, or logging out. Only admin users will be able to see the graphs. The documentation screen uses a card layout to navigate through all the different type of documents. The card layout responds to sliding and tapping gestures. It was our hope to have the card resemble real life tabs and folders for a more intuitive feel to the user.

Approach

We have continued to build upon the iPhone and iPad views in this single application. The information and views have been modified for both devices and have the same functionality.

Testing Practices

Seeing as most of our back end development will be done in Objective-C we will use unit tests to test our code against expected results. Once the model driven development phase of this project runs its course we will develop use cases and port those into unit test. This will give us a solid base to launch the test driven development phase of the project.

As for testing the database, we will develop sample data that violates each of the constraints and verify that none of the rows were accepted. Then we will test the database with data that it should accept.

The front end of the application will be tested with mostly user and acceptance testing. Other testing frameworks that will be used will be Testflight, Crashlytics, emulators and real devices.

Development Environment

We have continued using Xcode for our development. Xcode has all the needed tools to develop this applicaiton. XCode has integration of GIT within it, which will make it even easier to organize our code via a private GitHub account. It also allows us to easily add frameworks as needed.

For the most part, we will divide up the work as much as possible and work on it at our own homes. Thanks to GitHub, we can easily share our work without meeting constantly. We will still have occasional meetings and we will work together when deemed necessary.

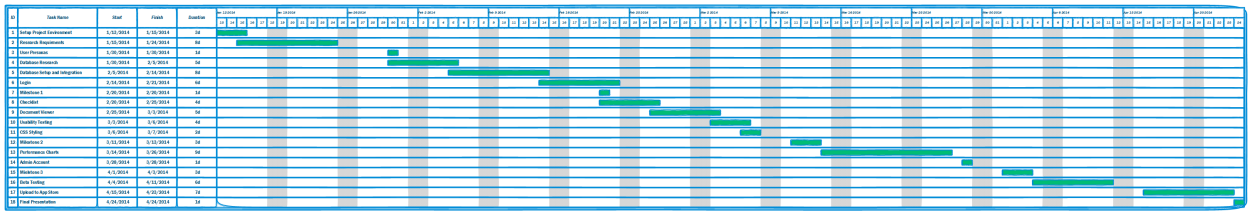
Division of Labor

We have all created a Trello account specifically for the Aero Med project. Here we put up ideas for the application and list the functions that need to be implemented. Each person selects a task and moves it to a separate “In Progress” card while they are working on it. Thanks to Trello we have not had any conflicts when making changes to the application.

Design Methodologies

The design methodology that we have been using so far is model driven development. Right now we have focused on getting prototype ideas out there and looking ahead to assume what we will run into and plan for that. We will continue this until we have a proper model for the application including design prototypes, database schema, code structure, and needed functionality. At that point we will have enough planned out where we can split the work up and continue with a more test driven method of development. That way we can have all of the tasks split up and we can develop in our own areas.

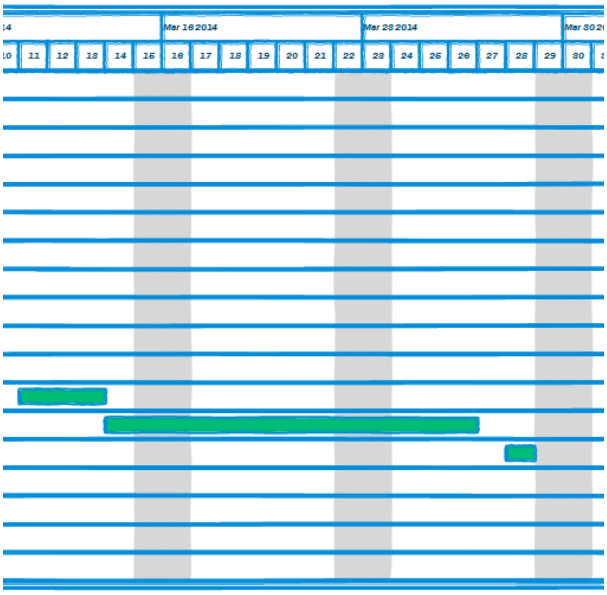
Gantt Chart



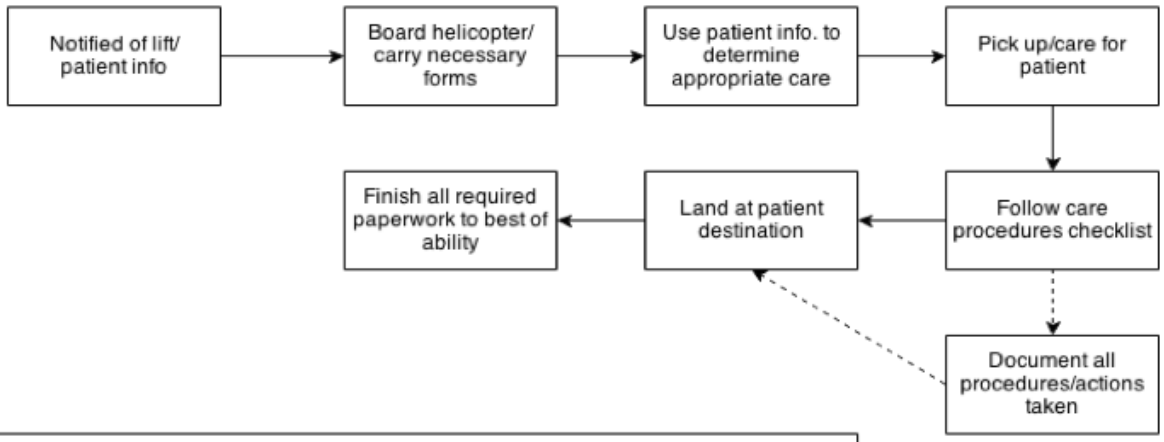
Our Gantt Chart might be a little difficult to read because of its large scope. There are some close ups at the top of the following page in order to more easily read the chart. For a better view, head to our project website at http://www.brodyberson.com/aeromed.

We’ve mostly been on schedule with the exception of the performance charts. We’ve moved the development and implementation of this feature further down our Gantt chart, but still before Milestone 3.

ID	Task Name	Start	Finish	Duration
1	Setup Project Environment	1/13/2014	1/15/2014	3d
2	Research Requirments	1/15/2014	1/24/2014	8d
3	User Personas	1/30/2014	1/30/2014	1d
4	Database Research	1/30/2014	2/5/2014	5d
5	Database Setup and Integration	2/5/2014	2/14/2014	8d
6	Login	2/14/2014	2/21/2014	6d
7	Milestone 1	2/20/2014	2/20/2014	1d
8	Checklist	2/20/2014	2/25/2014	4d
9	Document Viewer	2/25/2014	3/3/2014	5d
10	Usability Testing	3/3/2014	3/6/2014	4d
11	CSS Styling	3/6/2014	3/7/2014	2d
12	Milestone 2	3/11/2014	3/13/2014	3d
13	Performance Charts	3/14/2014	3/26/2014	9d
14	Admin Account	3/28/2014	3/28/2014	1d
15	Mielstone 3	4/1/2014	4/3/2014	3d
16	Beta Testing	4/4/2014	4/11/2014	6d
17	Upload to App Store	4/15/2014	4/23/2014	7d
18	Final Presentation	4/24/2014	4/24/2014	1d



Target Audience Workflow



- Application should provide appropriate guidelines to treat the patient's condition
- Guidelines should be updated as needed
- Staff will easily record information about protocols through application
- Application should keep track of and display staff performance trends
- Leaders will use information to gather for auditing and ensuring quality care

Appendix A – Current Application UI

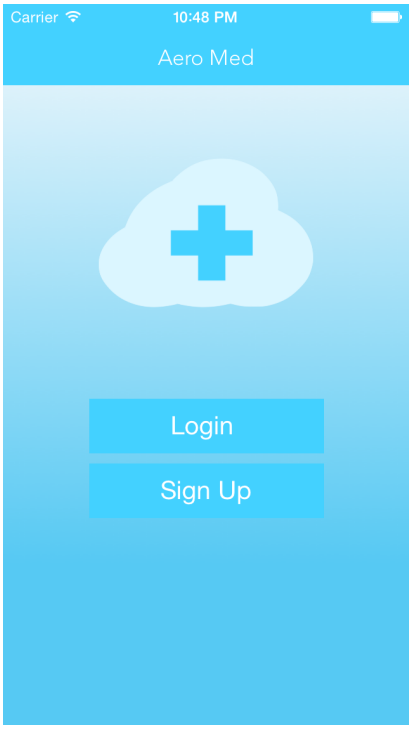


Figure 1. Splash Screen

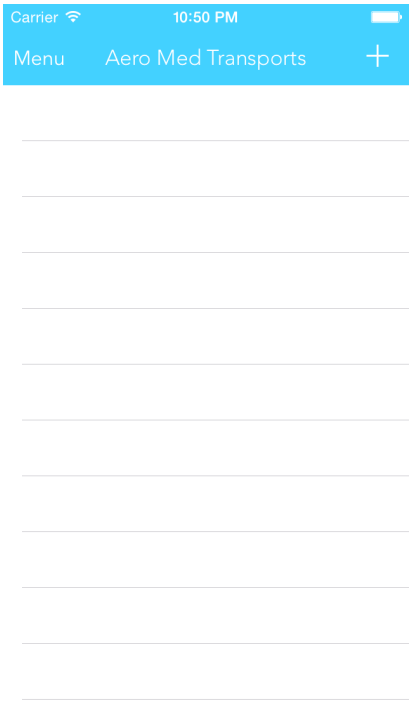


Figure 3. Transport View

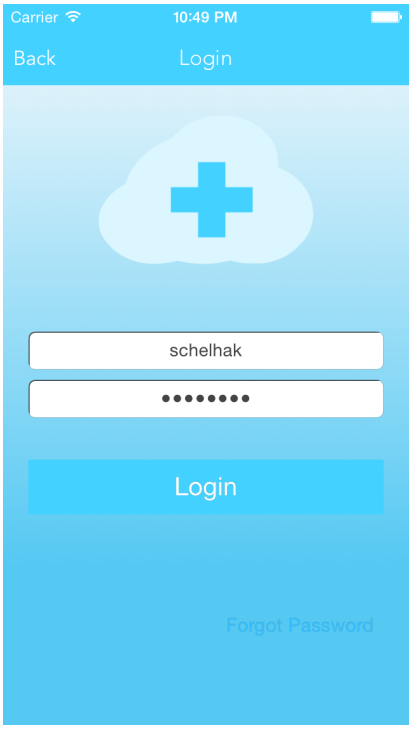


Figure 2. Login View

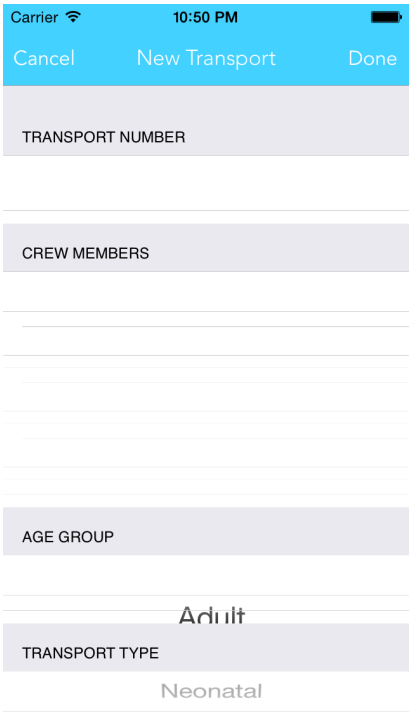


Figure 4. New Transport View

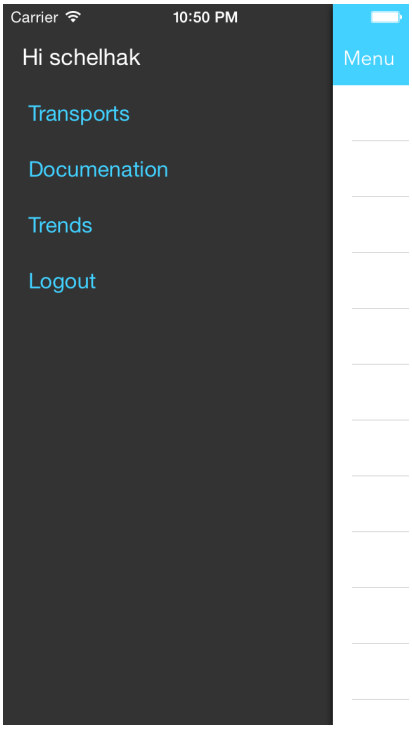


Figure 5. Slide-Out Menu

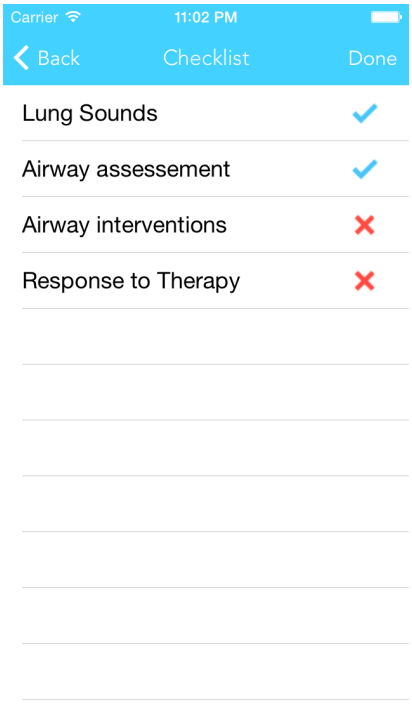


Figure 7. Checklist View

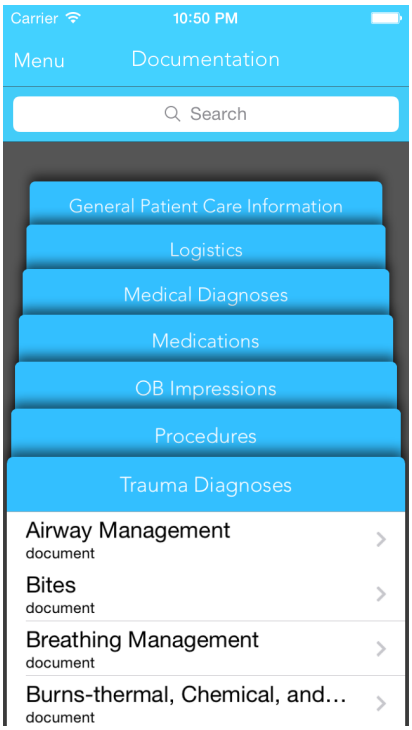


Figure 6. Document Viewer

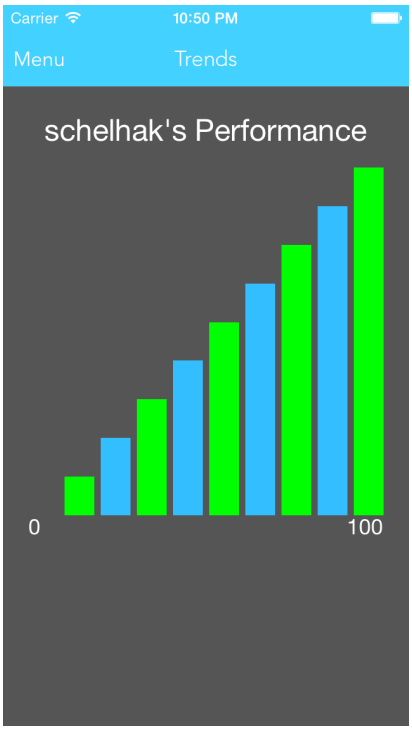


Figure 8. Trend/Performance Chart View