6-9-2015

Flippit: a local marketplace mobile application

Kyle Alwyn
Santa Clara University

Taylor Roden
Santa Clara University

Santa Clara University
DEPARTMENT of COMPUTER ENGINEERING

Date: June 9, 2015

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Kyle Alwyn and Taylor Roden

ENTITLED

Flippit: A Local Marketplace Mobile Application

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING AND BACHELOR OF SCIENCE IN WEB DESIGN AND ENGINEERING

THESIS ADVISOR

DEPARTMENT CHAIR

Flippit: A Local Marketplace Mobile Application

by

Kyle Alwyn and Taylor Roden

SENIOR DESIGN PROJECT REPORT

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering &
Bachelor of Science in Web Design and Engineering
School of Engineering
Santa Clara University

Santa Clara, California

May 28, 2015

Abstract

Every year, university students spend hundreds, if not thousands on textbooks. However, students cannot effectively sell off or return the used books. Current solutions do exist, but they are slow, difficult to use, and do not give a reasonable refunded amount. To solve the problem, we implemented a hybrid mobile application, built to be available on both the Apple App Store and Google Play Store.

Throughout the development process, several competitors emerged in the local book-selling market, therefore, our team pivoted and went more broad, allowing users to buy and sell anything, rather than restricting the platform to only books. Using real-time technology and geographic locations, we were able to create an instantaneous, localized marketplace that students could use to sell off their unwanted items. We facilitate community interaction and socio-economic improvement through a simple, usable interface. Moving forward, more thought and resources would have to be put into the technological infrastructure to aid in scaling the application, if it were to go viral.

Acknowledgements

This thesis is dedicated to our family, because without their love and support, we would not have had the opportunity to complete this project.

Also, a big thank you to Professor Daniel Lewis and Professor Loring Pfeiffer for helping us through this long, intensive process. We could not have done it without them.

Table of Contents

List of Figures

Chapter 1: Introduction

Problem Statement

Every year, people spend hundreds, if not thousands, of dollars on items that they end up not using after a short period of time. When said item, whether it be furniture, clothing, or electronics, will wind up sitting on a shelf unless the owner attempts to sell their item to a potential buyer. Finding a potential buyer efficiently comes with challenges, as the owner will advertise the item on various sites to increase their chance of successfully selling said item.

Currently, there are several platforms for which people can sell their belongings. Companies such as Amazon, and eBay have rolled out websites and services that assist in matching a buyer with a seller. The issue with these sites is that the quality of the item is unknown, unless it is from a trusted buyer, and then the buyer must wait for the item to arrive. These items obviously need to be shipped, increasing the price and creating another problem of a potential backorder, further delaying the shipment of the item. These sites are not simple to sell items on either and usually require the seller to mail said item on their own, causing some to not even bother with the process. Facebook groups are another alternative for selling items, but once again, many groups exist where users can post these items, and the buyer of an item may not find the specific group the seller has posted the product in.

Our solution is a mobile application, built for both Android and iOS, that acts as a local craigslist for people's items that they no longer need. The app will be powered using Adobe Phonegap hybrid application technologies. A user, wanting to sell an item, will be able to take a picture of their item and fill out relevant information, such as a title, pricing, and a description. When the user decides to post the item, we will gather the user's current location, via the phone's geolocation. On a separate tab, a user will be able to set a variable range, in miles, for which the app will pull posts by taking the searching user's current latitude and longitude, and comparing it with that of all the item posts. If the user chooses 5 miles, the app will return all items for sale

within that range. If a user likes an item, they can click on the post, in the feed, in order to see the details of the item. On the detail page, a user may elect to message the user, creating a real time conversation using Firebase real-time server technologies, or directly text message the seller through their phones SMS system, given that the seller has input their phone number in the description. There are many people looking to dispose of items that others may deem desirable in the community around us, so we want to establish a socio-economic buying platform, that not only supplies a valid option for those that want to sell or purchase items, but also creates a local community.

Other Projects

Throughout the development of this application, many similar projects arose that influenced the final implementation. Initially, our project started as a local forum for students to easily buy, sell and trade textbooks. As development continued, other services began to surface that shared a similar vision to this senior design. One such example is Booksmart (Booksmartapp.com), an application created by San Jose State students that attempts to achieve the same goals and solve the same problems. As a result, this application expanded not only from textbooks, but to an overall marketplace to stimulate buying, selling and trading of goods throughout the community.

Chapter 2: Project Analysis

Requirements

In order to implement this program, a few requirements must keep in mind. Requirements lay out what the system will do or needs to do. It is typically defined by the customer and must be prioritized. Requirements are split into three categories: functional, non-functional and design constraints. Functional requirements define what must be done in the system. They are usually evaluated as a true or false and is non-negotiable. Non-functional requirements define the manner in which the functional requirements need to be achieved. They are usually evaluated as a degree of satisfaction. Design constraints are restrictions on the way the solution is implemented.

Functional Requirements:

1. The user must be able to post their own items to sell on the application
2. There must be a way for the seller to get in contact with the buyer
3. Only postings within a variable distance must appear to the user

Non-functional Requirements:

1. The system can be used on iOS and Android devices
2. The system will be aesthetically pleasing so it will be easier for users to navigate the application
3. The application will limit the number of steps required by the user in order to browse items

Design Constraints:

1. Must be available as a mobile application

Conceptual Models

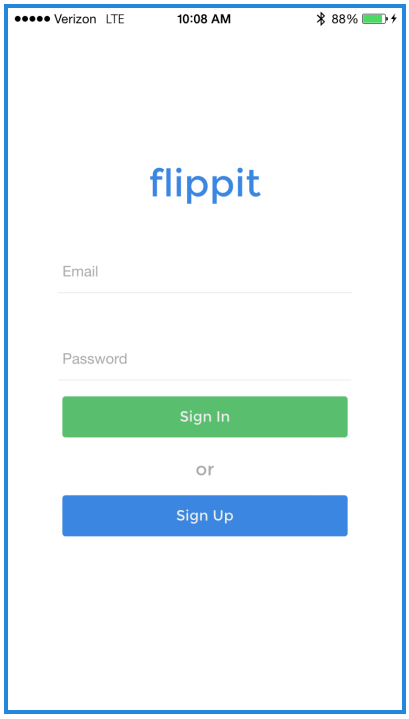

Figure 1: Login Panel
The user can login or create a new account

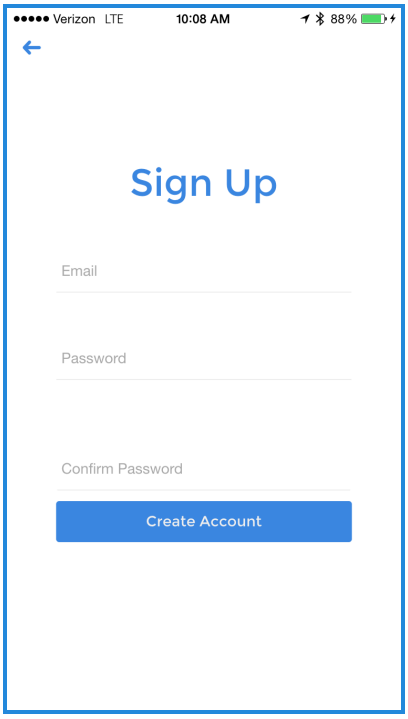

Figure 2: Sign Up Panel
The user can sign up for Flippit on this page

After gathering the requirements, the application must be conceptualized into a basic wireframe, showing the main interfaces and the interactions that the user will be performing. With our system being a mobile application, we are constrained to smaller rectangular screens, therefore we model our design following the mobile industry paradigms. In **Figure 1**, we see the login screen for our application. If the user selects "Sign Up", they will be taken to the screen shown in **Figure 2**.

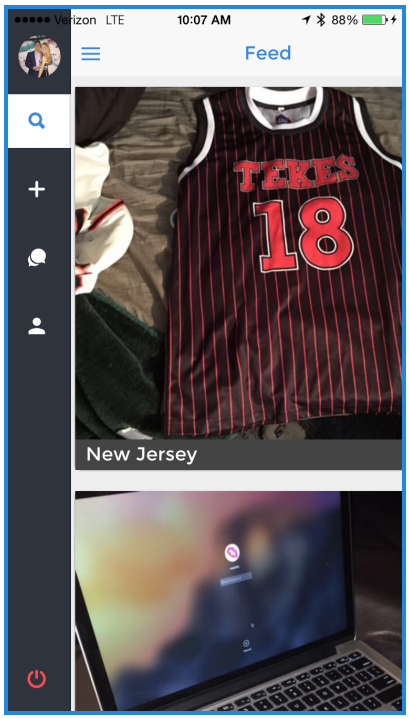

Figure 3: Side Menu Panel
The Side Menu presents the user with different options for navigating the app

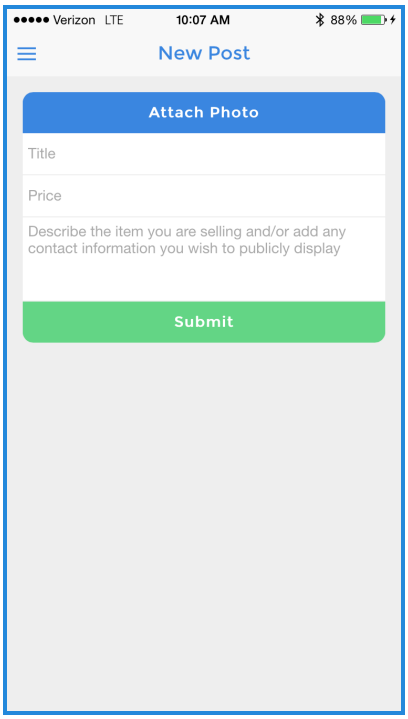

Figure 4: New Post Panel
A page where a user can create new postings

After logging in, the user can swipe left on the screen or press the top left-hand button to access the side menu, as shown in **Figure 4**. Users have the option to view their profile, browse the feed, add a new post, see their current messages, and logout. If the user were to press the "+" symbol, they will be taken to the "New Post" screen in **Figure 5**. On this page, a user can access their phone's camera roll to upload an already existing photo, or they can access the phone's camera to take a picture of the item if they have it in front of them. The user can then fill out other relevant details, such as the title, price, and description of the item. If the user goes back to the feed, they can click the top right button to access the "Feed Settings," shown in **Figure 6**. These settings allow the user to display items within a range of 2-100 miles. After setting the desired range, a user can browse the items in the "Feed", shown in **Figure 7**. The feed consists of users' postings for others to browse, displaying the post's picture, title, and price. The feed also utilizes the infinite scroll feature, much like existing social media platforms, so that items populate the page once the bottom of the page is seemingly reached.

Figure 5: Feed Settings Panel
Sets the variable range for the feed

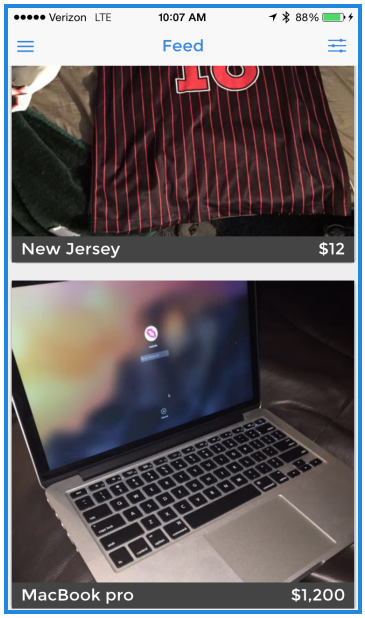

Figure 6: Feed Panel
The feed of items presented to the user

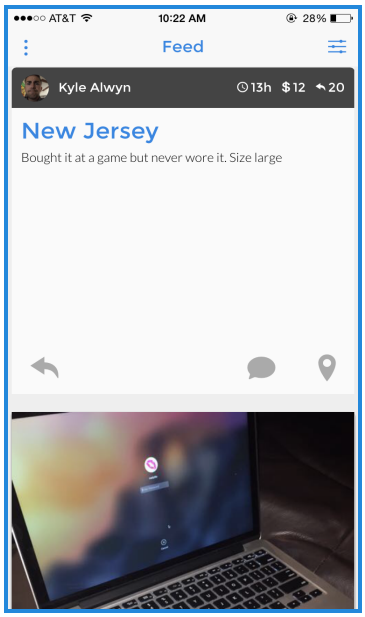

Figure 7: Posting Details Panel
Displays the details of a post that the user clicks on

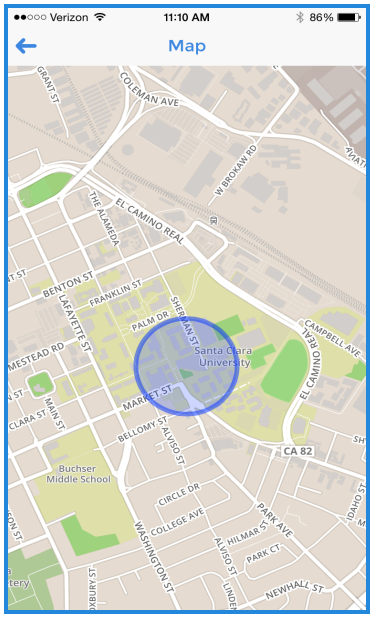

Figure 8: Map Panel
Shows where the item is located on a map

If the user wants to view the details of an item, they can click the picture of the posting and the image will "flip" around to display the details of the posting. In order to produce the flip effect, the image was rotated at a 45 degree angle about the x and y-axes. **Figure 8** shows said details, as the user then has the ability to see who posted the item, when it was posted, the asking price, as well as a counter to track posting views. The body of the posting includes the item's title and description. At the bottom, the user can press the arrow to flip the posting back to the image, message the owner, or view on the map the location of said item. If the user elects to view the map, they will be taken to the screen shown in **Figure 9**, where a rough estimate of the item's location is presented to the user. Once the user decides they want to purchase an item, or if they have other inquiries, they can message a user, where a chat will be created, as seen in **Figure 10**. The user can also view their messages and open conversations, depicted in **Figure 11**.

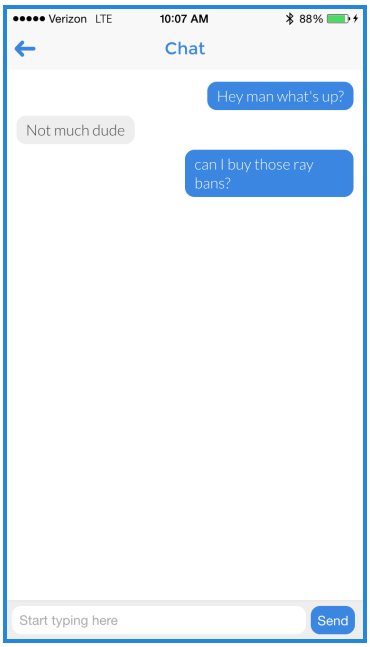

Figure 9: Message Panel
A message conversation

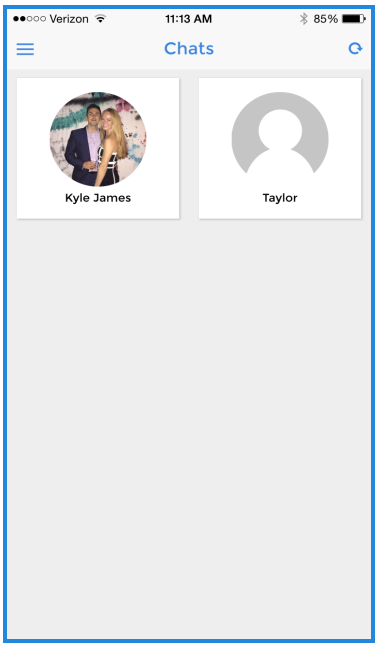

Figure 10: Chats Panel
Displays all open chats

Furthermore, a user can edit their profile settings to edit their information. **Figure 12** shows the user's profile page, where they can view their profile picture, name, and current postings. If they swipe left on an item, they can choose to delete or sell the item, as shown in **Figure 13**. The user can also change their information by clicking the "Edit Profile" button, where they will be taken to the screen shown in **Figure 14**. On this page, the user can edit their information, such as their name and picture.

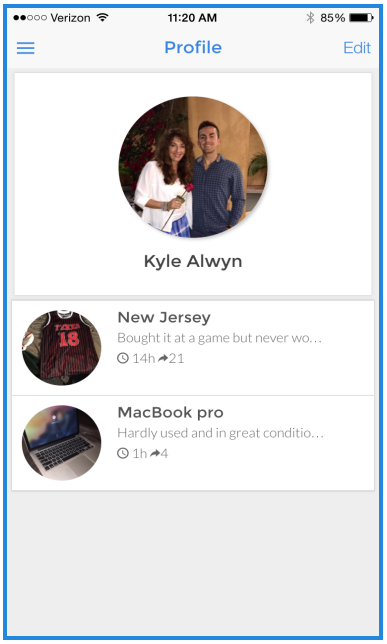

Figure 11: Profile Panel
The user's profile page

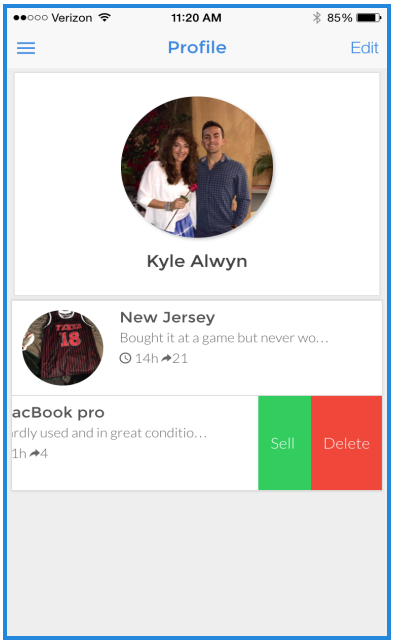

Figure 12: Edit Posts Panel
The user can delete their
items or mark them as sold

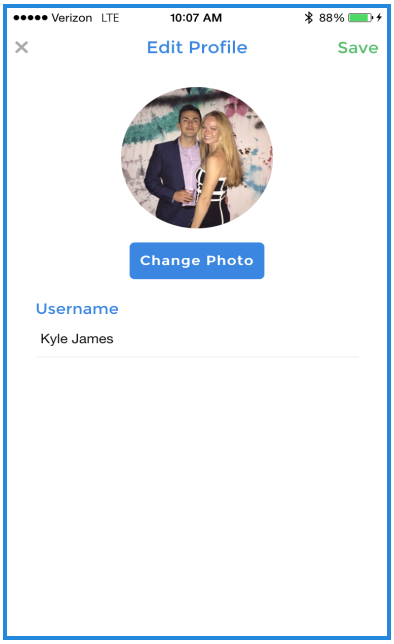

Figure 13: Edit Profile Panel
The user can change their
name and picture

Use Cases

After conceptualizing our model, we must determine how a user will interact with our program. We create use cases in order to simulate the scenarios. A use case is each action that an end user will typically perform while interacting with the program, software, or application as shown in **Figure 15**.

A use case typically includes:

- Name
- Goal
- Actors
- Precondition: things that must be true before the use case can be performed
- Postcondition: things that will be true after the use case has been successfully performed
- Steps required: enumerated list of steps, sometimes w/ substeps or alternatives
- Exceptions: common errors and how they are resolved

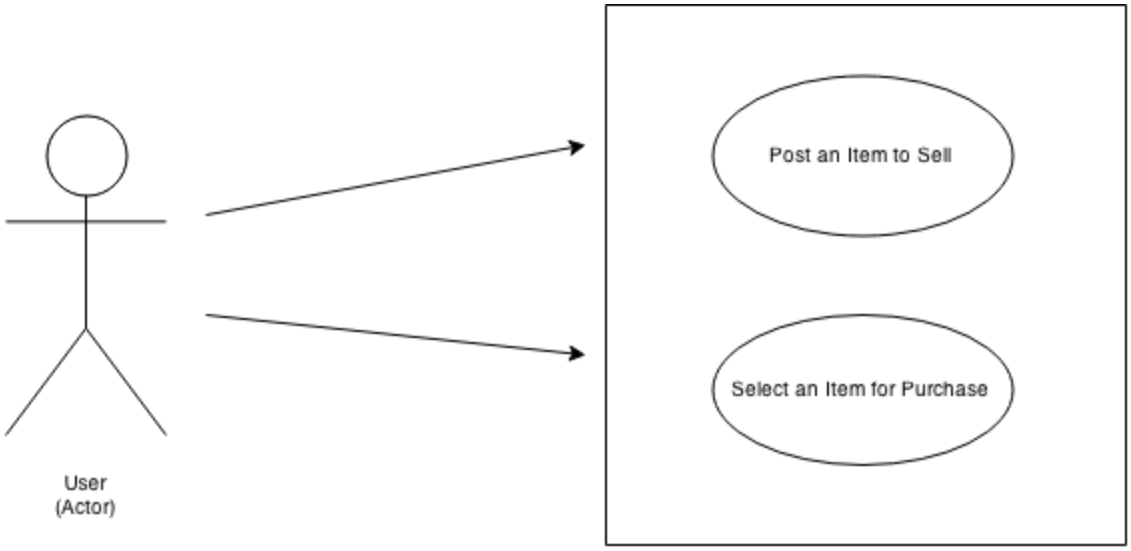

Figure 14: General actions available to every user for our mobile application

Use Case #1

Goal: The user posts their item for sale on the system for other users to view

Precondition:

- ❖ The user has the item in their possession and necessary information for their posting

Postcondition:

- ❖ A post for the item is created on the application and available for all users to see

Steps Required:

1. The user uploads a picture to the "Add Post"
2. The user fills out relevant information, such as asking price and other details

Exceptions:

- ❖ The user is not at the location they want their post to be viewed from

Use Case #2

Goal: The user locates a post with an item they want to purchase

Precondition:

- ❖ There are posts within range for the user to view

Postcondition:

- ❖ The user can contact the seller to purchase the item

Steps Required:

1. The user locates an item they want to purchase on the feed
2. The user views the item's information and messages the seller

Exceptions:

- ❖ The book desired may not have a posting within range

Chapter 3: Design Decisions

Architectural Design

Once we have determined the user's interactions with the program, we must organize what is happening on the back-end for each interaction. The architectural design, seen in **Figure 16**, gives a high level overview of what is happening behind the scenes when a user is interacting with our application.

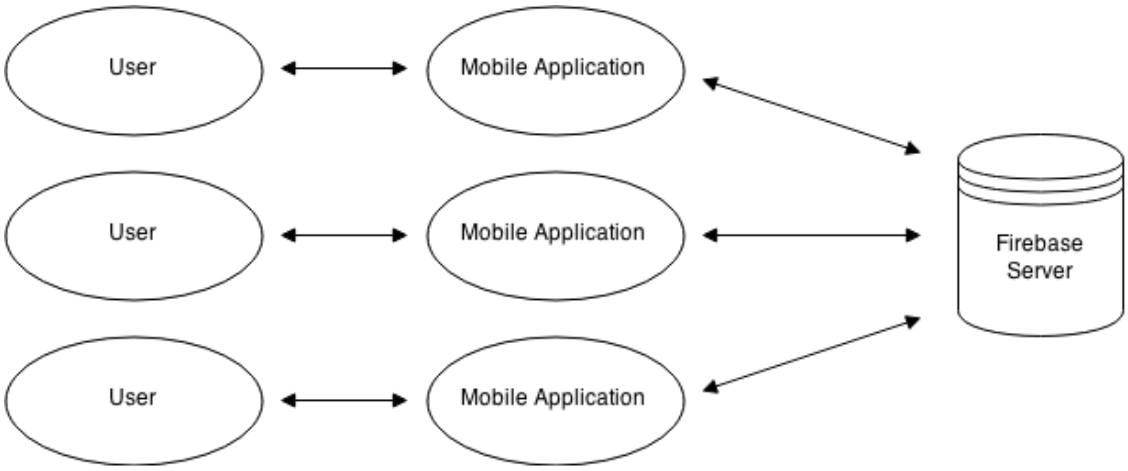

Figure 15: The high level overview of our system architecture

Instant Messaging

We also developed an in-app messaging client for our users. The design is shown in **Figure 17**, and demonstrates what is occurring when a user messages another user.

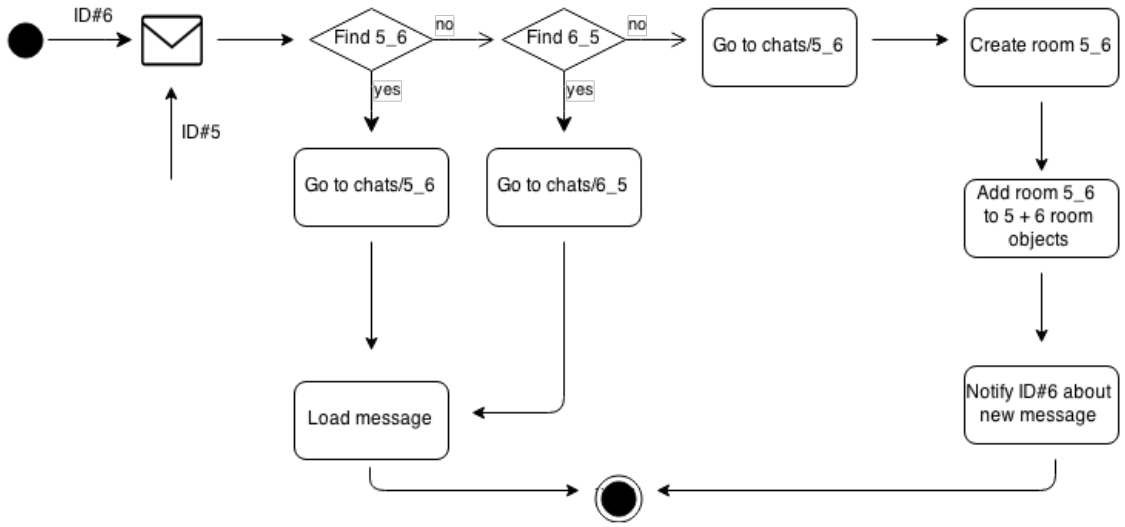

Figure 16: Overview of Instant Messaging between example client IDs #5 and #6

Technologies Used

To implement our application, we had to decide which technologies languages would be best for carrying out the requirements. We used the following:

- HTML (HyperText Markup Language): holds the displayable content for the application
- CSS(Cascading Style Sheets): used to style the application, making it more user friendly and aesthetically pleasing
- JavaScript: used to make the website interactive and provide access to the DOM (Document Object Model)
- Ionic Framework: a subset of Cordova and Adobe Phonegap, a system that bundles and compiles web languages into a native iOS or Android application
- AngularJS Framework: Ionic is built on the AngularJS framework, because of its powerful client side manipulation abilities and implicit data binding to aid in real-time interface updates
- Firebase: a real-time server that stores data as JSON (Javascript Object Notation) objects, making the data easy to pull and use on the client side.

Design Rationale

To begin, we went the route of a mobile application because it is easily the one of the largest industries in the world. It is growing and has no sign of stopping. Because our team has background in web programming and hybrid applications are becoming more relevant, we decided to build using Ionic, so we did not have to learn another programming language. We are using the most modern web languages, with HTML5, CSS3, and AngularJS.

In terms of the looks and user experience, we want to make it as simple as possible. We used an off-canvas side menu in order to maximize screen real-estate, so tabs on the bottom would not overlap search results. The 3 bar button in the top left is a web paradigm that resembles the menu, which will show on touch. The colors are subtle and provide enough contrast to allow users with vision impairment to recognize color differences throughout the application. Images are all responsive, so the width and height will be determined based on the user's screen size, meaning the users will grow on an iPhone 6 and shrink on an iPhone 4, while still looking natural. The FontAwesome CSS Icon Library was used in order to substitute icons for words, such as a price tag for how much the book will cost, or an eyeball for how many views the posting has received.

Throughout the application, transitions and motion are used to show the user that the app is responding. For example, the user can click an item to view its details, where a flip effect is applied to the image, creating a "two-sided page" effect. This provides the user a feeling that something is happening behind the scenes and optimizes the user experience. Other uses of feedback are success screens, which prompt the user that a successful action has taken place. Occurrences of a success screen will show after changing information on the settings page, or after posting an item. Users want to make sure that their actions are saved so they do not repeat themselves.

Chapter 4: Planning Phases

Project Risks

Risks	Consequences	Probability	Severity	Impact	Mitigation Strategy
Time	Not meeting deadlines	.7	8	5.6	Follow the created Gantt Chart
Bugs	The system fails to work properly	1.0	5	5.0	Thoroughly test all cases as we develop the application. Have classmates alpha test the system to find unknown bugs
Group Management	Not being able to meet regularly	.4	5	2.0	Keep an open line of communication between group members and the advisor

Test Plan

Throughout the implementation process, various tests have been conducted to to ensure our application runs smoothly and successfully. We used Unit Testing and fed the application mock data in order to check the inputs and outputs of each component. We also used End-to-End (E2E) testing to ensure all of our components integrated properly. For example, we tested the ability to create new posts by uploading items to the feed and checking that their information is accurate both on the feed and the server. Furthermore, we had friends and family test out the application to provide us with user feedback. While the Unit and E2E tests helped us with debugging the application, the user groups gave us feedback on user experience and possible improvements for the application.

Development Plan

The test plan is a part of our overarching development plan, which is presented on **Figure 17**. The development plan provides a timeline by which we implemented our application.

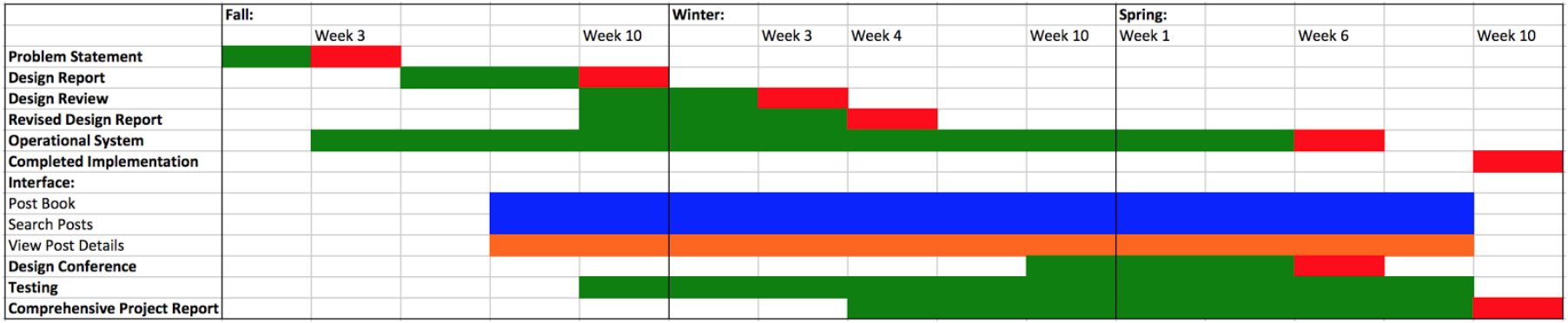

Figure 12: The Gantt chart details the timeline by which our project will be completed. The legend designates the color that matches to a specific member of the team.

Chapter 5: Societal Issues

Ethical

With any project comes ethical consequences, good or bad. Being a software project, several ethical issues could be raised. First, when dealing with users on our platform, we have the ability to gather and store personal information. It is our duty, as developers and creators, to not exploit the user. Second, some users are not comfortable with divulging their location through an app, which would defeat our purpose. Ethically, we want to include everyone and not leave out a piece of the market, so some would want us to find a way to use the app without gathering geolocation.

Social

As a buy and sell platform, social interactions between the buyer and seller are inevitable. With our localized geo-concept, our plan is for users to create a tight-knit, trustworthy community centralized by commerce. Social interactions can, and most likely will, go awry at times. Users may send hateful message within our instant messaging feature, or could “rip off” a user at the time of transaction when they meet, **as explained in our ethics section.** When we build this product, we rely on the users to be ethically and socially correct, but if negative interactions begin occurring, then we must intervene, through new security policies or the banning of users.

Political

We do not feel that there is much political relevance within our application.

Economic

With any long-term project, there will be expenses in order to properly acquire necessary materials. Fortunately for Computer Engineers, we can easily reduce some of these costs. For example, Apple requires users to pay for a Developer Account in order to upload applications to the App Store and for testing on a physical iOS device. To cut cost, we only got one Developer Account for testing purposes, and simulated the system when appropriate. The project overall was not costly, but if we were to continue to expand this application, we would need to pay for the infrastructure to be put in place, as we would need our servers to support a large user base.

Beyond the costs for the project, we also considered potential monetary gain. There are numerous ways to profit from application, with the most common being a paid application or advertisements. In order to increase our user base and create a socio-economic platform, we felt that putting a price tag on the application would work against that goal. As a result, we decided that we would go with minimal advertising if we were attempting to make a profit. Ultimately though, we decided against further modification of the application for monetary gain, as we built the application for experience purposes.

Health and Safety

The safety of our users was a great concern, as personal information needs to be protected in a system where users eventually meet up face to face. We promoted anonymity within the application by allowing users to change their display name to any name of their choosing. This allows people the option to select a unique name that differs from their given birth name.

Furthermore, we built in a chat feature so our users do not have to exchange contact information unless they buyer and seller agree to share it with each other within the chat room.

Since the ultimate goal of the application is for our users to be able to sell and buy items from each other, we also encourage users to meet in a public setting to avoid harm. We want to create a community that benefits society, so health and safety precautions are vital to the application's success.

Manufacturability

Apart from a code editor and test devices, no other manufacturing costs or machines need be used.

Sustainability

In order to make our application fully sustainable, we need it to be easily scalable to account for an increase in user base. This can be achieved by increasing the load the server can handle, but we would most likely need to switch to a more traditional-based server model. We aim to create a community within the application that would help it be sustainable for years to come.

Environmental Impact

While we want our application to be sustainable in terms of longevity and use, we also want our application to be sustainable for the environment. Although it does not have an immediate application, our system promotes sustainability between our users. Instead of throwing away

unneeded items, our application promotes reuse, as users can sell their used items to other people through the application.

Usability

Today's world has seen an exponential increase in interface and experience design. Because of this, we heavily focused on creating and simple, easy, and seamless experience for the end-user. With the amount of apps available, the experience and functionality are the true differentiators. We also believe first impression is important for retention. If a user encounters any problems on their first visit, the likelihood they will open the app again or even keep it on their phone dwindles. We spent much time developing an onboarding flow to satisfy and retain a user on first load. We also worked hard to make our application accessible, for users with poor eyesight or for users without much experience on smart devices. If usability and user experience is not at the forefront of designing an app in industry today, it will not see success.

Lifelong Learning

As students to the computer engineering profession, we cannot stop learning, or we will be left in the stone age. Relatively to many other industries, software is young and still changing rapidly, therefore we must stay on the cutting edge in order to keep an edge. We experienced this directly in our project. We worked with young technologies (Ionic, Firebase, and Angular), and encountered several changes in their functionality throughout our design and development process. They are constantly iterating on their product, and would release a change or update that we needed, but would break our old code, causing us to refactor several times. This demonstrated

to us first hand that we need to be involved and aware of not only the tech we are working with, but the tech that is being developed around us.

Compassion

After studying at Santa Clara University the past four years, we have had the 3 C's ingrained within us: Competence, Conscience, and Compassion. The latter was a significant drive to this project. We have experienced the problem of buying and selling on campus, and it is because of this that we do not want others to suffer the same problem. Although this is not as critical a problem as solving world hunger or building infrastructure in developing countries, it has the ability to enhance everyday life in communities around the world, and this is something we strive for.

Chapter 6: Conclusion

Improvements

Through testing and user feedback, there are features that we would like to integrate that go beyond our initial development plan. Our users wanted the ability to search and find specific items with ease. Through this feedback, a new feature we would introduce is to search for specific items based on description or item title. Also, we wanted to expand the settings so users can filter their feed of items. The settings tab would also include a variable range for the cost of products and the type of product (electronics, clothing, sporting goods, etc.) to supplement the distance already set by the user. Furthermore, our users want the ability to track their prior sales and progress. While this idea gives the user a true mapping of their activity, it also provides incentive for the user to continue using the app. For instance, there could be incentive bonuses, such as a reliability rating, for those that reach a certain number of successful sales or purchases so that other users know they are trustworthy.

Marketing Plan

Beyond asking our users for feedback through testing, they are also an integral role in our marketing plan. In order to spread the application, we would want the users that enjoyed the application to share it with their own network of friends. In addition, our immediate families can easily spread the application to their workplaces to reach markets outside of a campus setting. If the application is successful locally, our extended family can spread the application to other areas once we are ready to handle a greater user base. Besides word of mouth, we can extend our marketing by creating an instructional video, advertisements on social media (Facebook, Twitter, Instragram), and advertisements around school campuses utilizing existing portals such as TV's or flyers.

Lessons Learned

Internally, we have learned quite a bit over the course of this one year project. First, native applications are still far more powerful than hybrid applications, due to the direct access you have to system components that are not available using web technologies. We have also learned that it is difficult working with young technologies such as Firebase and Ionic. They are still growing and evolving, and their frameworks still have bugs uncovered daily. This hindered our progress during certain times of our development until an answer, fix or workaround could be found. Lastly, we learned how to adapt and read the market. Someone had also realized this was a problem and wanted to solve it just as we had, so when they rolled out their "solution," we had to adapt and grow ourselves in order to beat out our competitors.

Concluding Remarks

Through this year-long process, we have created a community-oriented, socio-economic platform that allows for users to better connect to those in their immediate surroundings. We used modern web technologies (AngularJS, Firebase, and Ionic) to create a hybrid application that allows users to post items within a variable range for others to view and potentially purchase. Although we had to work around emerging projects with similar goals as ours, it expanded our knowledge on adapting to unforeseen dilemmas.

Works Cited

Booksmart. Booksmart App. Vers. 1.3.1. Booksmart Inc., 2015. Web. 15 Jan. 2015. <www.Booksmartapp.com>.