6-7-2019

# Bronco Books: Textbook E-commerce Platform

Vineet Joshi

Terry Shih

David Taylor

# SANTA CLARA UNIVERSITY
## DEPARTMENT OF COMPUTER ENGINEERING

Date: June 7, 2019

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

**Vineet Joshi**
**Terry Shih**
**David Taylor**

ENTITLED

# Bronco Books: Textbook E-commerce Platform

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

Thesis Advisor

Department Chair

# Bronco Books: Textbook E-commerce Platform

by

Vineet Joshi
Terry Shih
David Taylor

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

Santa Clara, California
June 7, 2019

# Bronco Books: Textbook E-commerce Platform

Vineet Joshi
Terry Shih
David Taylor

Department of Computer Engineering
Santa Clara University
June 7, 2019

ABSTRACT

College students purchase textbooks for the classes they take every quarter, but current solutions for selling back those textbooks are insufficient, requiring that the student pay to utilize the selling platforms or that the student build rapport within a given community. Our project, Bronco Books, offers a solution by being a native mobile application open to only SCU students. Bronco Books will be free to access and will act as an e-commerce platform where students go to sell their textbooks. We were motivated to create Bronco Books primarily because we wanted to help alleviate the financial burden that comes with purchasing textbooks. t This document covers the requirements, use cases, and activity diagrams for Bronco Books. We also explain our rationale behind some of our design decisions, such as the technologies we will be using for this project, and our followed timeline. We conclude the document by discussing the lessons we learned and the developmental future of Bronco Books.

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

College students purchase textbooks for their classes during each academic period. These textbooks are expensive and are often not useful after the end of the academic period for which they were purchased. The question becomes: what to do with these remaining textbooks? Students often get frustrated looking for a way to sell their textbooks, and they end up keeping the textbook without ever using it again or recycling the book. For those who do attempt to sell their books, there are no satisfying solutions to the problem.

One way students can deal with this problem is to sell their textbooks on large e-commerce platforms, such as Amazon, eBay, or Craigslist. However, it is hard to sell items on these large e-commerce platforms because buyers prefer to buy from dependable sellers, and becoming a dependable seller requires building a reputation over time [6]. Since most students are not frequent sellers on these platforms, building a reputation becomes difficult. Also, these platforms often require the seller to ship the product to the customer, and most college students don't want to deal with the hassle of shipping. Students can also sell their textbooks back to a bookstore, but bookstores generally have a limited list of textbooks they buy back. More importantly, the buyback rates are extremely low. For instance, the buyback price at the SCU Bookstore for a like-new $200 physics textbook was $40. Rentals, another alternative, are offered by bookstores, but many books do not have a rental option. More often than not, only the cheapest textbooks have a rental option, which forces students to buy the expensive textbooks. The old-fashioned way of selling used textbooks is to post a listing on bulletin boards in dorms. This solution is non-ideal since the listings are unavailable to students who live off campus and to students who live in different dorm buildings than the seller.

## 1.2 Solution

Our solution is a native mobile application that allows students to buy and sell used textbooks. Students sign in using their SCU logins, and can post their used textbooks for sale. We will integrate barcode scanning so students can simply take a picture of their book's barcode, which will auto-fill the book information fields. The listings will also include textbook condition and at least one image that will justify the price to the buyer. This platform will be available only to college students, so sellers don't need to build their reputation over time, as there is a much higher expectation of trustworthiness between fellow students than unknown sellers on e-commerce platforms. Students interested in a listing will be able to let the seller know directly within the app, which will notify the seller to initiate further contact through email or phone. Sellers will deliver the books in person so students don't have to hassle with shipping. We will also integrate Apple Pay for iOS and Google Pay for Android to remove the need for physical transactions, such as cash or check. Most importantly, our solution will create a centralized platform where college students can sell any textbook regardless of demand, and can set their own price.

# Chapter 2

# Requirements

In this chapter, we have listed the features that must be included in our design. Functional requirements are the specific system functions/features that our design is required or should aim to have, while non-functional requirements are more broad statements about how our design should function. We have divided the functional and non-functional requirements into 3 sections each: critical, recommended, and suggested which represent the priority of each requirement.

## 2.1 Functional

### 2.1.1 Critical

- System will store textbook listings created by sellers
- System will facilitate communications between buyer and seller
- System will enable logins using SCU student accounts
- System will allow payment through Apple Pay/Google Pay
- System will allow sellers to edit or remove textbook listings

### 2.1.2 Recommended

- System will scan barcodes to auto-fill information for textbooks
- System will provide a mechanism for direct communication between buyers and sellers

### 2.1.3 Suggested

- System will provide push notifications when contacted by a buyer or seller

## 2.2 Non-Functional

### 2.2.1 Critical

- The system will be easy to use for buyers and sellers
- The system will be secure for buyers and sellers
- The system will be reliable for all users

### 2.2.2 Recommended

- The system will have high availability

### 2.2.3 Suggested

- The system will be scalable to other educational institutions

## 2.3 Design Constraints

- The system must run on native Android and native iOS.

# Chapter 3

# Use Cases

There are two kinds of users using this system: buyers and sellers. Buyers and sellers share a majority of the use cases for our system: they can login, view textbook listings, search for textbooks, and contact their opposite. On top of these actions, buyers can of course purchase a textbook, and sellers can publish a textbook listing to the app.

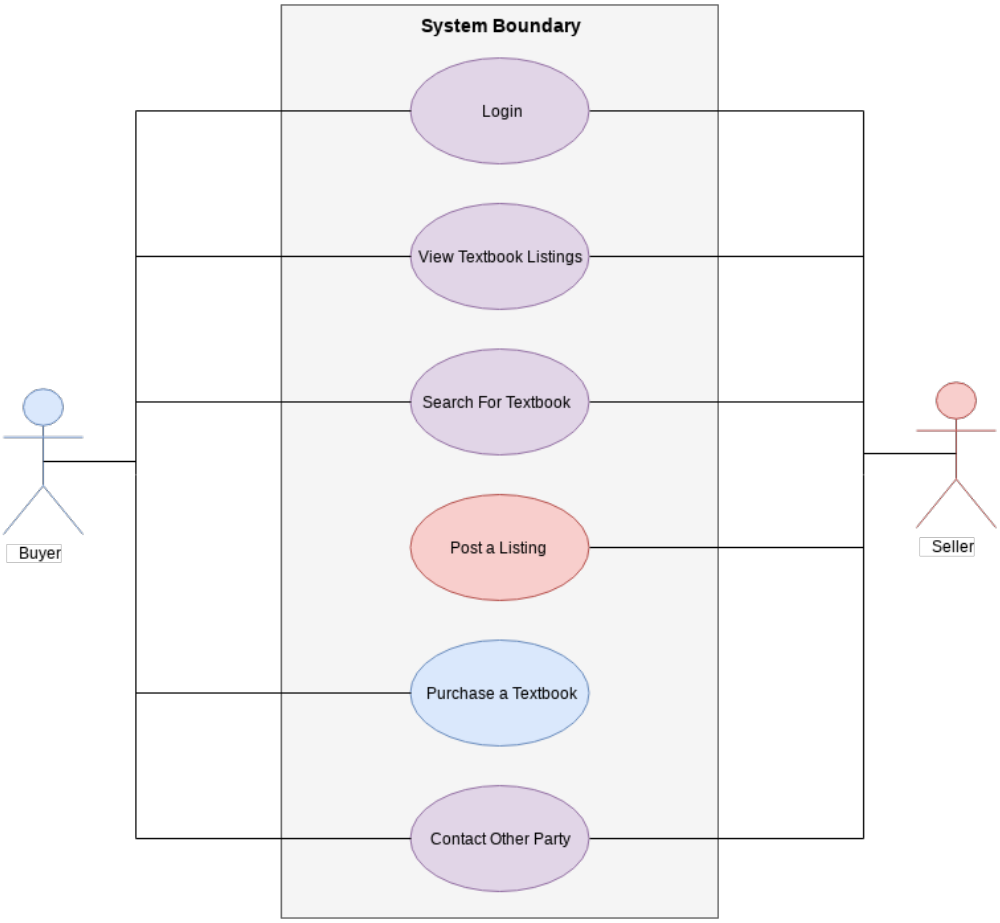


Figure 3.1: Use Case Diagram

## 3.1 Use Case Scenarios

- Login
  - **Actors:** Buyer, Seller
  - **Goal:** Log in the user into his or her SCU account.
  - **Preconditions:** The user has a SCU login.
  - **Steps:** Use SCU authentication (username, password) to log in the user.
  - **Postconditions:** The user is logged in if he or she entered valid credentials; otherwise, the user is prompted to login again.
- View Textbook Listings
  - **Actors:** Buyer, Seller
  - **Goal:** Show all textbook listings to the user.
  - **Preconditions:** User must be logged in.
  - **Steps:** Scroll through the textbook listings.
  - **Postconditions:** None.
- Search for Textbook
  - **Actors:** Buyer, Seller
  - **Goal:** Search for a textbook using filters.
  - **Preconditions:** User must be logged in.
  - **Steps:** Enter filters to search for a specific textbook.
  - **Postconditions:** The textbooks matching the filters are shown.
- Post a Listing
  - **Actors:** Seller
  - **Goal:** Post a textbook listing for buyers to potentially purchase.
  - **Preconditions:** User must be logged in and must have a textbook to list.
  - **Steps:** Use barcode scanning or manual entry to fill out textbook info, take photos of textbook, specify a price and preferred payment method.
  - **Postconditions:** The listing is posted on the server, and all users can view it.

- Purchase a Textbook
  - **Actors:** Buyer
  - **Goal:** Purchase a textbook from a specified listing.
  - **Preconditions:** User must be logged in and must select a listing.
  - **Steps:** Tap the buy button and initiate payment.
  - **Postconditions:** The listing is marked as sold, and the specified dollar amount is transferred from buyer to seller.
- Contact Other Party
  - **Actors:** Buyer, Seller
  - **Goal:** Contact other users about textbook listing.
  - **Preconditions:** User must be logged in.
  - **Steps:** Choose method of contact (such as email or phone), and start communication.
  - **Postconditions:** The users are able to initiate communication with each other from the application.

# Chapter 4

# Activity Diagram

Figure 4.1 shows the general flow for a buyer. After successfully logging in, the buyer lands on the textbook listings page. From there, the buyer can view their profile and see any listings they may have posted as a seller. From the textbook listings page, buyers can select a particular listing in detail before getting into contact with the seller to facilitate purchasing the textbook.

The seller's activity diagram, depicted in Figure 4.2, shares many aspects with the buyer's activity diagram. Seller's can log in, browse the textbook listings and view their own profiles. The main distinction between the two activity diagrams is the flow chart depicting the process of uploading a textbook list to the system. A seller can either scan the barcode of their textbook or input the book information manually. AFter that, they confirm the information, input a price, then submit the listing to the system.

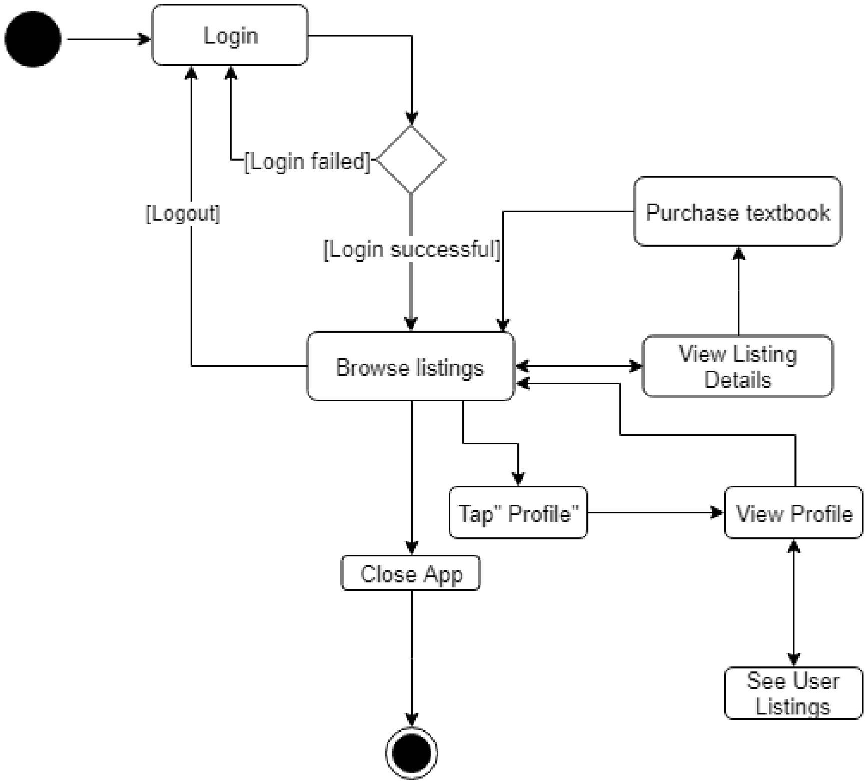


Figure 4.1: Buyer Activity Diagram

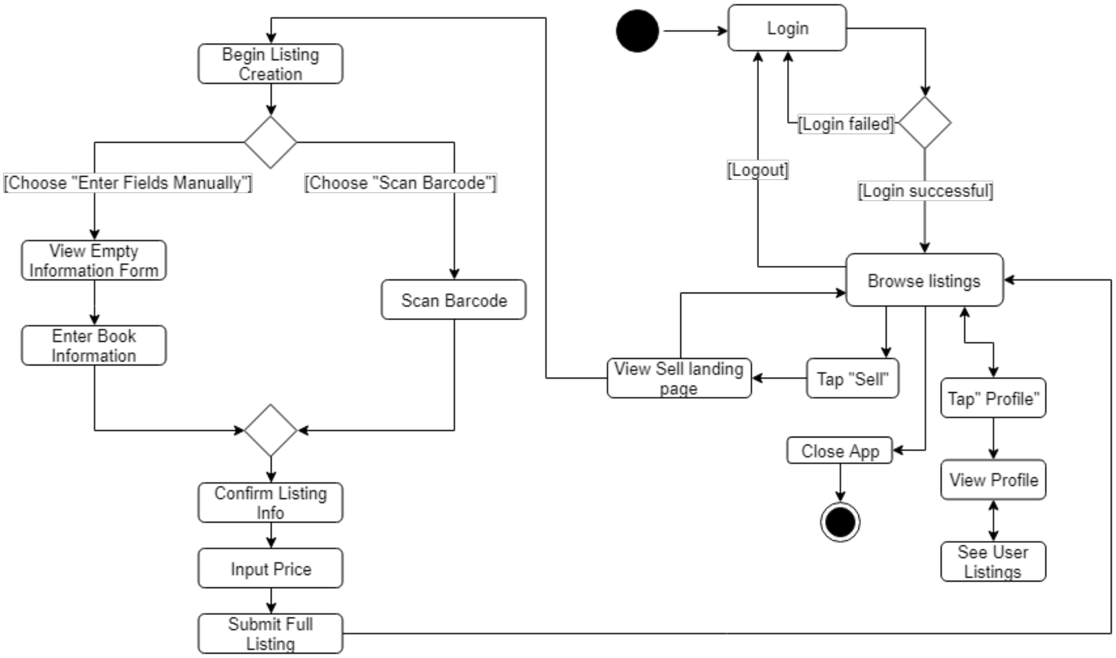


Figure 4.2: Seller Activity Diagram

# Chapter 5

# Architecture

We used a data-centric architecture for our system due to its nature as a consumer-to-consumer e-commerce platform, as this type of system needs data to be accessible to everyone, and as such users store no relevant data themselves. Instead, the generated data is stored in a centralized database so that it can be universally accessible to all of our users. Firebase is the middleware in this architecture, since it provides an interface into the backend database, which is hosted on Google servers.

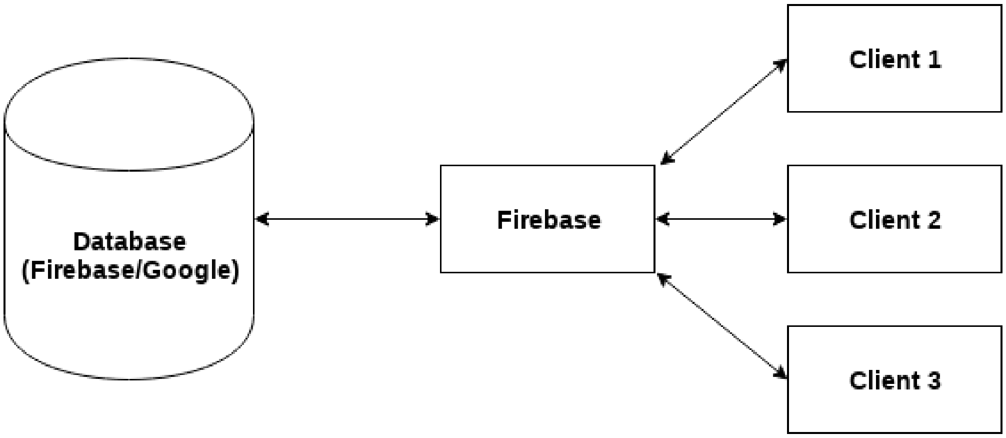


Figure 5.1: Software Architecture Diagram

# Chapter 6

# Technologies

As stated before, we developed both a native iOS and Android version of our app. Beginning with the front end of each app, the iOS version has its front end developed in **Xcode**[2] using Swift as the development language. The Android version was developed in **Android Studio**[3] using Java as the development language. Meanwhile, the back end of both apps were developed using **Firebase**[4] as our framework. As part of Firebase, we used a tool called **ML Kit**[8], which we used in tandem with the **Google Books API**[5] in order to allow for scanning textbook barcodes and auto-filling listing fields.

## 6.1 Xcode

Xcode is the official development environment for iOS app development. Beyond just letting you write your code, it has built in tools to help test your code, tools to aid in building the user interface for your app, and a simulator to allow you to run your app as you design it. When you're ready to release your system, Xcode handles the process of distributing the app on the App Store.

## 6.2 Android Studio

Similar to its counterpart Xcode, Android Studio is the official development environment for building Android apps. With tools to help with testing your app, designing your app's interface, and emulating your app as you build it, Android Studio has everything needed for efficient app development. Once you're ready to release your app, Android Studio takes care of the distribution.

## 6.3 Firebase

Firebase is a Google-owned mobile development platform. While Xcode and Android Studio handle the interface of an app, Firebase is used to develop back end code. Firebase has a large suite of features that help with app development:

- Realtime database hosting

- Cloud storage for app data and files/images
- User authentication
- Web asset hosting
- Crash Analytics
- In-app messaging
- Remote app updates (without deploying a new version)
- Testing tools

and much more. Along with all of this, Firebase also has access to a feature called ML Kit, which we also utilized.

### 6.3.1 ML Kit

ML Kit is a built-in tool for Firebase that includes a number of ready to use APIs for a number of different functions. In our case, ML Kit provided us barcode scanning functionality for our app. Beyond that, ML Kit also allows for adding easy to use machine learning features to mobile apps.

## 6.4 Google Books API

The Google Books API takes in an ISBN number and returns all of the attributes for the book with that ISBN. These attributes include the title, authors, publisher, date published, etc. As mentioned before, we used ML Kit to allow users to scan the ISBN number from textbook barcodes, which is fed to the Google Books API to retrieve the relevant textbook information. We used this to auto-fill all associated fields when posting a textbook for sale, so the user doesn't have to enter these fields manually.

# Chapter 7

# Design Rationale

## 7.1 Why Native Apps?

Our decision to develop our solution as a mobile app was driven by our desire for our design to be as convenient and easily accessible for the user as possible. Designing on a mobile platform allows us to include features such as using smart-phone cameras to scan textbook barcodes and take pictures of textbooks to include in the listings. We are developing for native iOS and Android apps rather than a mobile web app because a native app allows us to have more convenient access to camera functionality, as mentioned before, and access to built-in payment methods, such as Apple Pay and Google Pay.

## 7.2 Why Firebase?

We decided to use Firebase for our back end because it allows us to create a back end without needing to write any server-side code. Also, it provides us with ML Kit, which provides functionality for barcode scanning. Since our app stores images of textbooks, we also needed a development platform that allows for easy storage of images. Also, Firebase provides a framework for authenticating with Google using the user's Gmail account, which greatly simplified the code we have to write for authentication.

## 7.3 Why SCU Login?

Because our focus is on the SCU community, we chose to utilize the SCU Login system. As students ourselves, we recognize the apathy towards making additional accounts. By using the SCU login system, we can circumvent this issue. Furthermore, by forcing users of our app to login through SCU login, we can ensure that only SCU members are using the app.

# Chapter 8

# Test Plan

Beyond standard unit, integration, and system testing of the code itself, our Test Plan included thorough testing of all the functionality for Buyers and Sellers. We created test accounts for 2 buyers and 2 sellers, each on Android and iOS. Using the Seller account, we created a new listing by using the smartphone camera to scan the textbook barcode (which auto-filled textbook information). We completed the listing by using the smartphone camera to take textbook pictures, specifying the price, and specifying the payment method (Apple Pay, Google Pay, or cash). Once the listing was posted, we used the Buyer account to view the listing, and purchase the textbook using either the native payment method (depending on mobile platform) or specifying cash. We also tested the messaging feature between buyer and seller, which should send a text message or email to the respective parties. After we tested it ourselves, we also performed usability testing by having outside parties attempt to go through the process of both buying and selling a textbook with no prior knowledge of the exact functionality of the app.

# Chapter 9

# Risk Analysis

In every project, there are situations that can arise that interfere with the projects completion. Below is a table analyzing the likely hood of some of these situations, and how severe their impact on the overall project would be.

| **Risk** | **Severity** | **Probability** | **Impact** | **Mitigation Strategies** |
|---|---|---|---|---|
| Programs Bugs | 4 | 1.0 | 4 | 1. Start testing early<br>2. Stick to detailed test plan |
| Poor Time Management | 7 | 0.5 | 3.5 | 1. Cut feature(s)<br>2. Proper delegation of work and sticking to timeline |
| Insufficient Understanding of Technologies | 7 | 0.5 | 3.5 | 1. Get familiar with development environment before beginning work<br>2. Have multiple members assigned to less understood development sections |
| Loss of Code or Development Environment | 8 | 0.1 | 0.8 | 1. Backup work and use cloud storage<br>2. Use a version control tool (Github) |

Figure 9.1: Risk Analysis Table

# Chapter 10

# Timeline

Below is the timeline that we followed for the development of this project. We began by adding functionality to create and post listings to the database, and then finished user authentication. Next, we moved on to barcode scanning to auto-fill the listing information fields, and then added the option to upload photos for a listing. We took a break from development during our final exams and spring break. Afterwards, we added functionality to edit a listing, and then handled the payment system using Apple Pay and Google Pay. We worked on the user interface and all forms of testing over the entirety of this time period as well.

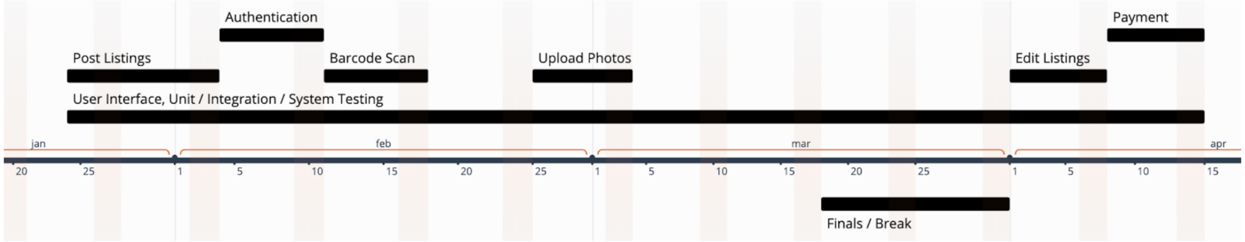


Figure 10.1: Our Development Timeline

# Chapter 11

# Societal Issues

- Ethical
  - Our main concern with this project is that, like other peer to peer e-commerce platforms like Craigslist, there is always some danger when meeting someone from online to purchase something from them. We attempted to mitigate this by making our app closed to SCU students only, as we believe there is a level of trust between fellow students. However, the risk is always there and needs to be acknowledged. Another ethical concern we considered was the possibility that our app could be used in a black market type situation, using books as a cover to sell other items, though we consider this to be very unlikely.
- Social
  - We believe our app encourages the SCU student community to support each other during their college years by helping each other mitigate one expensive part of attending college.
- Political
  - The only political concern that we believe could come up with our app is the issue that peer-to-peer purchases could be viewed as a way to avoid paying taxes on certain purchases, though this is not our intent with the app.
- Economic
  - We do not believe there are any major economic considerations with the cost of our app, as we used only free services in its development, and intend for it to be available on the app store for free as well. A minor concern could be that, with a large enough user base using it at the same time, we may need a paid version of our database service in order to accommodate all those requests at once, though we do not believe that will occur.

- Health and Safety
  - As mentioned in the ethical concerns, with peer-to-peer platforms, there is always a danger of something happening during the in person hand-off of the product. However, since most if not all exchanges will likely be happening on the SCU campus itself, we believe this to be an unlikely occurrence. Another concern would be the safety of the user information we store, which we accounted for when picking a database service in order to ensure it was properly encrypted.
- Manufacturability
  - We do not believe there are any manufactuability issues with our app, due to it being easily deployed en masse on app stores with no further costs other than the initial development time costs and the cost of any future updates/maintenance.
- Sustainability
  - Our product is designed to be viable indefinitely without many updates by its very nature, and if updates are needed, it is also easily modified and updated to fit whatever new functionality is necessary for it to continue to be useful. A sustainability issues would likely only ever arise in a situation where physical textbooks are no longer used on college campuses.
- Environmental Impact
  - The main environmental impact our app has would likely be an increase of the reuse of books. By allowing students to sell their books to each other, the hope is that more students will be used books, hopefully helping to contribute to less books needing to be printed (with new editions and newly written books being an exception), leading to less paper production, and less trees being cut down.
- Usability
  - Usability was one of our critical requirements for this app, and as such we designed it to as straightforward and easy to understand as possible.
- Lifelong learning
  - Over the course of this project, we were forced to research and learn new technologies that none of us had worked with before, without the benefit of a teacher to help us understand the technologies. This was incredible practice for having to do this in the future as new technologies are made and we need to learn them in order to continue to be up to date for our jobs.

- Compassion
  - The driving force behind this project was a desire to help our fellow students and relieve some of the struggle associated with buying textbooks every quarter and then, when they no longer need them after the quarter, only getting a fraction of what they paid for the book back. By providing students with an alternative that allows them to not only get a more fair price for their book, but help out other students with getting their books for the next quarter, we believe our app is one of compassion.

# Chapter 12

# Conclusion

In our concluding section, we will go over some of the lessons learned during the development of Bronco Books, some of the advantages and disadvantages we thought of for our system, and finally go over some future plans should we continue to develop Bronco Books.

## 12.1 Lessons Learned

### 12.1.1 Thorough Research

We learned very quickly the importance of thorough research during the planning phase of our projects. The technologies that we thought could assist us ended up slowing down the development process by either being unusable to us (as was the case with ISBNdb API, which we had believed was free but had turned into a paid service in the middle of our planning phase) or by not working the way we intended for it to (as was the case with Apple Pay and Google Pay, which require that we become licensed sellers in order to integrate their services directly into our app [1]). Had we encountered these hurdles during the planning and research phase rather than in the middle of development, we could have saved a significant amount of time, as could the next topic of discussion.

### 12.1.2 Project Management

Working in a group environment taught us basic project management skills. As a team, we would often fall into familiar pitfalls, such as pieces of the system taking much longer than expected to develop and us having to work around the time taken from that underestimation. We also struggled with us overestimating how much time parts of the system would take to develop, often leaving us scrambling to find the next step to accomplish.

## 12.2 Advantages and Disadvantages

The biggest advantage Bronco Books has is the fact that it is a mobile app. In college campuses, it is exceedingly rare for a student to not have access to a smart phone. Another positive is that Bronco Books' community is exclusively

Santa Clara students. A buyer looking for a textbook for a given course is likely to find a listing for it because the other users are also Santa Clara students who are likely to have taken that course. Our biggest shortcoming is the lack of a payment system that is fully integrated into Bronco Books.

## 12.3 Future Plans

### 12.3.1 Payment Methods

Currently, Apple Pay and Google Pay are implemented through the corresponding text messaging services. As was mentioned in Section 12.1.1, direct integration of Apple and Google Pay into Bronco Books would require that we become licensed sellers. We want to look into services such as Venmo or PayPal to see if we can implement an integrated payment system into Bronco Books

### 12.3.2 Phone Numbers

Because we use Firebase to access Google Authentication, our current implementation cannot save the user's phone number (to be used for contact through text messaging) to the system. We would want to add this functionality so that the app does not need to prompt for the user's phone number every time that user logs into Bronco Books.

### 12.3.3 Polishing Android-side

The Android version of our app is slightly behind the iOS version. Future work would include adding search functionality for browsing textbook listings and hooking up the barcode scanning for book information. We know how to implement these features, it was simply time constraints and prioritization that prevented us from implementing these features

### 12.3.4 Integration with the Registrar

Another interesting feature that would be nice to have is for the users to search and tag their books under the course that the book is relevant for. This way, users can search for textbooks based on the courses that they need texbooks for.

### 12.3.5 Price Comparison

This feature is about pulling pricing data of a given textbook from the internet, so that users can see statistics, such as average price, related to the price of the textbook. This is useful for buyers who want to know if the listed price is fair and for sellers who may need the information to set a fair price.

# References

[1] About Apple Pay for Merchants. Apple Support, 13 May 2019, support.apple.com/en-us/HT204274.

[2] Apple Inc. Xcode - Apple Developer. Xcode - Apple Developer, developer.apple.com/xcode/.

[3] Download Android Studio and SDK Tools. Android Developers, developer.android.com/studio.

[4] Firebase. Firebase, Google, firebase.google.com/.

[5] Google Books API. Google Books APIs — Google Developers, Google, developers.google.com/books/.

[6] How to Sell on Amazon - Amazon Services - Amazon.com. Amazon, Amazon, services.amazon.com/selling/benefits.html?ld=SCNASFButton.

[7] Joshi, Vineet. Bronco Books iOS Repository. GitHub, github.com/vineet0713/Bronco-Books.

[8] ML Kit for Firebase. Firebase, Google, firebase.google.com/products/ml-kit/.

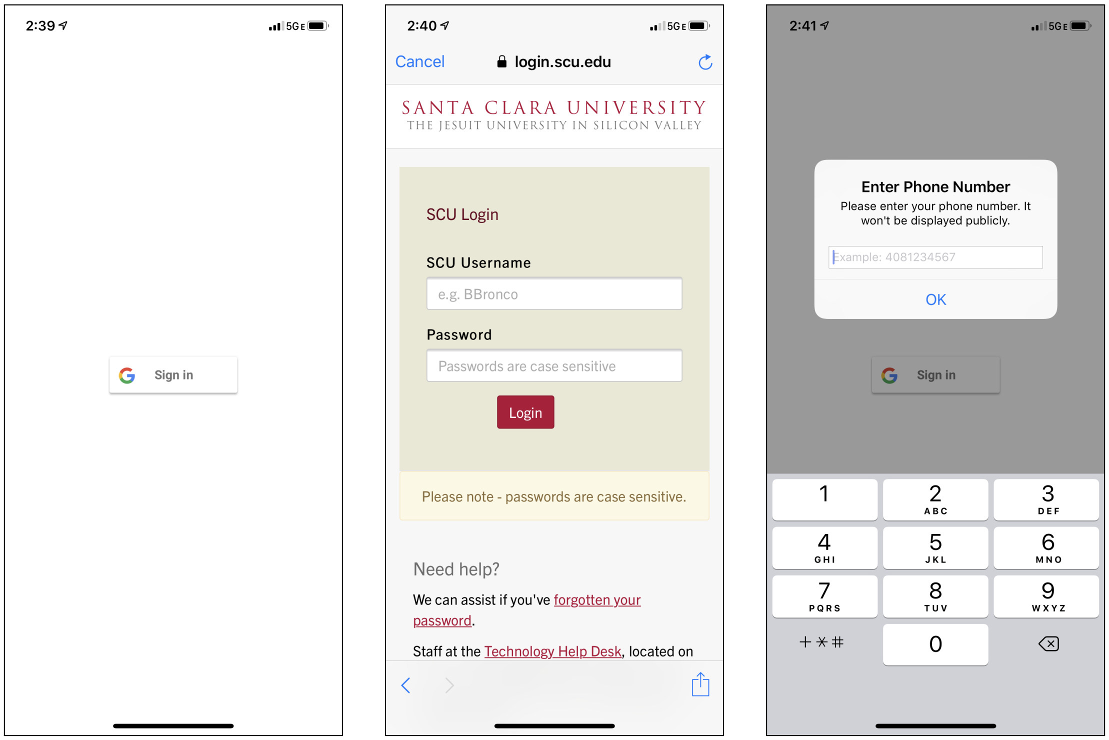
[9] Shih, Terry, and David Taylor. Bronco Books Android Repository. GitHub, github.com/terrie9876/Bronco-Books-Android.

# Chapter 13

# User Manual

We have provided a list of instructions and associated screenshots to make it easier for users to understand the application flow. Our app is divided into a 4 different ”sections”:

1. **Setup** (Figure 13.1)
    - Tap the Google Sign In button
    - Sign in with your SCU username and password
    - Enter your phone number (Ex. 4081234567)
2. **Buy** (Figure 13.2)
    - View all listings posted for sale
    - Search for a specific listing based on title, author, and seller
    - Tap a listing to view its information, contact the seller, edit the fields, or purchase it
3. **Sell** (Figure 13.3)
    - Tap the Scan Barcode button and point your smartphone camera on your textbook’s barcode (the barcode should be automatically detected)
    - Edit the listing fields and upload textbook photos
    - Tap the Post button to post your listing for sale
4. **Profile** (Figure 13.4)
    - View your name, email, and phone number
    - View all textbooks you have posted for sale or have already sold
    - Tap the Logout button to sign out of your SCU account



SANTA CLARA UNIVERSITY
THE JESUIT UNIVERSITY IN SILICON VALLEY

SCU Login

SCU Username

e.g. BBronco

Password

Passwords are case sensitive

Login

Please note - passwords are case sensitive.

Need help?

We can assist if you've forgotten your password.

Staff at the Technology Help Desk, located on

Enter Phone Number

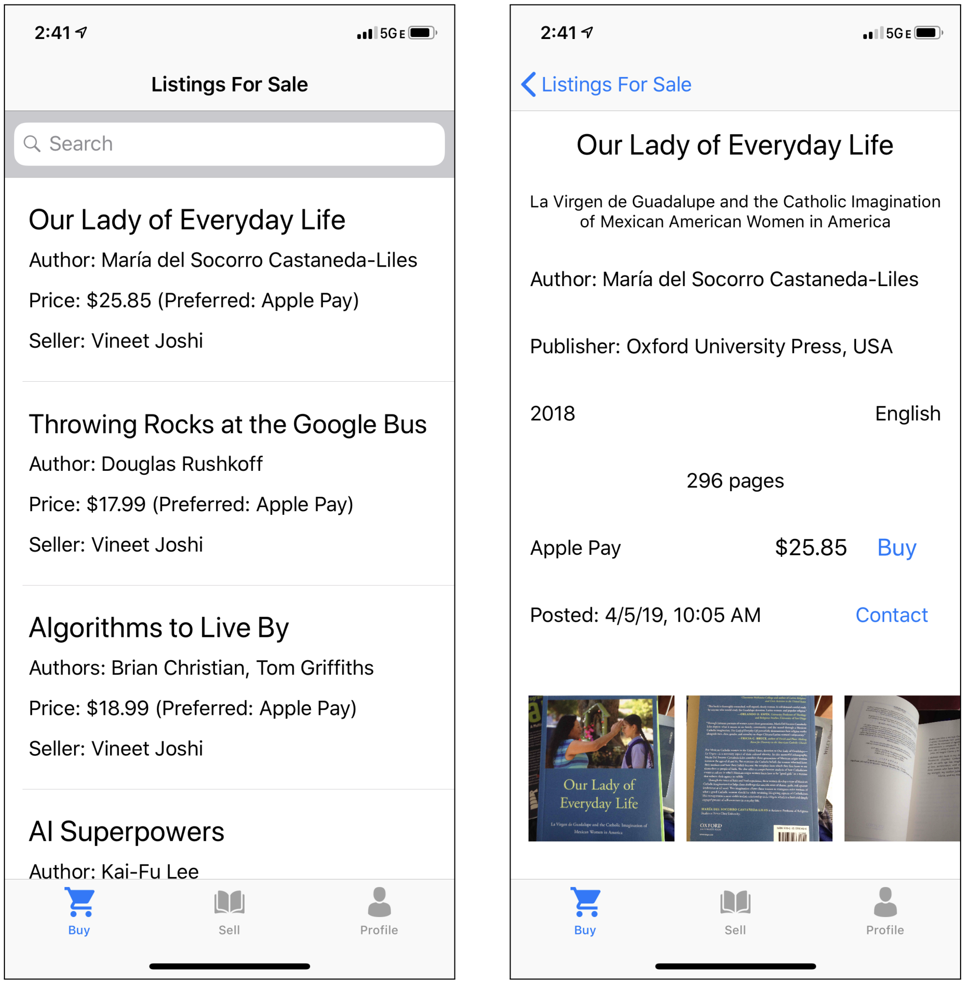
Please enter your phone number. It won't be displayed publicly.

Example: 4081234567

OK

(a) Splash Screen (b) SCU Login (c) Enter Phone Number

Figure 13.1: Setup



Our Lady of Everyday Life

Author: María del Socorro Castaneda-Liles

Price: $25.85 (Preferred: Apple Pay)

Seller: Vineet Joshi

Throwing Rocks at the Google Bus

Author: Douglas Rushkoff

Price: $17.99 (Preferred: Apple Pay)

Seller: Vineet Joshi

Algorithms to Live By

Authors: Brian Christian, Tom Griffiths

Price: $18.99 (Preferred: Apple Pay)

Seller: Vineet Joshi

AI Superpowers

Author: Kai-Fu Lee

Our Lady of Everyday Life

La Virgen de Guadalupe and the Catholic Imagination of Mexican American Women in America

Author: María del Socorro Castaneda-Liles

Publisher: Oxford University Press, USA

2018 English

296 pages

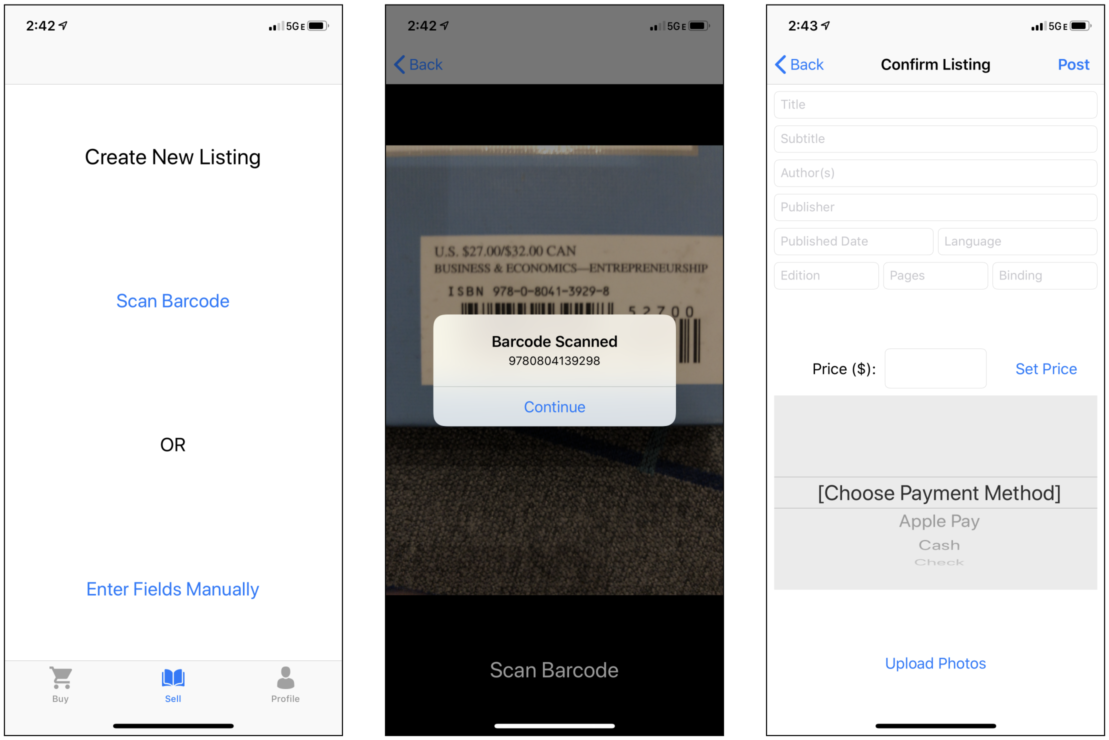
Apple Pay $25.85

Posted: 4/5/19, 10:05 AM

(a) Listings For Sale

(b) Listing Details

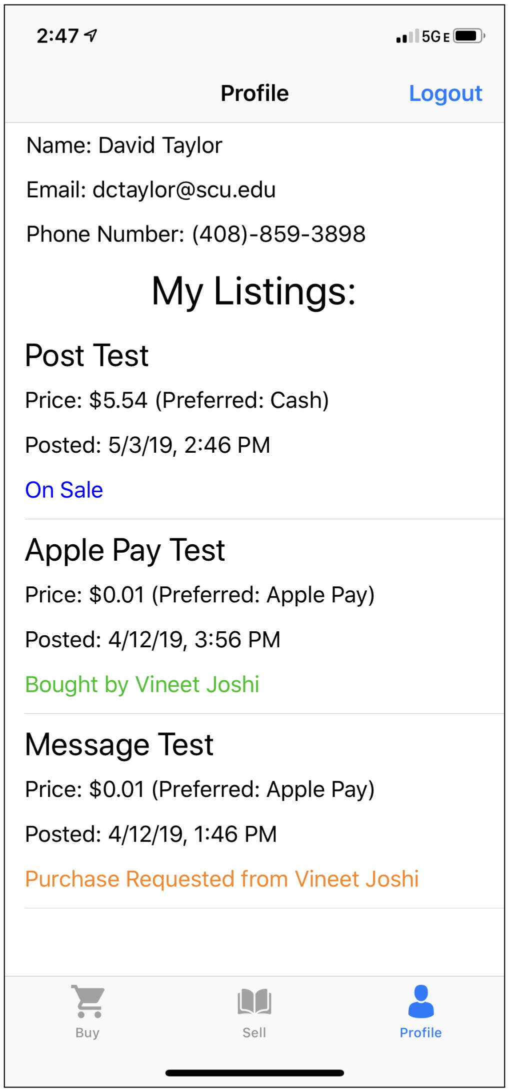
Figure 13.2: Buy



Barcode Scanned
9780804139298

(a) Start Sell Process (b) Barcode Scanning (c) Confirm Listing

Figure 13.3: Sell



Name: David Taylor

Email: dctaylor@scu.edu

Phone Number: (408)-859-3898

My Listings:

Post Test

Price: $5.54 (Preferred: Cash)

Posted: 5/3/19, 2:46 PM

On Sale

Apple Pay Test

Price: $0.01 (Preferred: Apple Pay)

Posted: 4/12/19, 3:56 PM

Bought by Vineet Joshi

Message Test

Price: $0.01 (Preferred: Apple Pay)

Posted: 4/12/19, 1:46 PM

Purchase Requested from Vineet Joshi

Figure 13.4: Profile