College Swap

Requirements Document

Patrick Boatner

Charodd Richardson

Chris Popovich

Justin Kerber

Change History:

Date	Summary	Author(s)
1/22/2015	Start of doc. Outline of requirements and use cases.	Team
1/29/2015	Draft of introduction, description, and requirements.	Team
1/29/2015	Draft of class diagram.	Patrick Boatner
2/03/2015	Draft of activity and use-case diagrams. Refined class diagram.	Team
2/05/2015	Finished activity diagrams.	Chris, Charodd
2/05/2015	UML class diagram	Patrick Boatner
2/05/2015	Use case diagram and descriptions	Justin Kerber
2/17/2015	Class diagram description	Patrick Boatner

Table of Contents

1. Introduction

1.1 Motivation/Purpose

College Swap is an Android application that will serve as a platform, much like Craigslist, where college students can exchange goods and services with other students. The motivation behind this project is that, as college students, we often find ourselves in need of an easy and secure way to sell and buy things with fellow students. The main item we find ourselves in need of exchanging is textbooks for school which is the main motivation behind this project.

1.2 Scope

Initially this app will be made to serve the University of Alabama but could later be expanded to work with any other colleges. The users who would be able to access the application would be limited to those who have a 'crimson.ua.edu' email address in this version although if expanded to other colleges we would add support to allow those users as well. The app will not have support for purchasing through the app but will instead encourage the buyer/seller to meet at a safe location on campus. The app will be bounded to cell phones and will not be available for use on a tablet. However, this could in the future be an added feature.

1.3 Goals

The goals of College Swap are to provide a user friendly and secure platform for college students to trade textbooks, football tickets, and subleasing apartments with their peers.

1.4 Key Definitions

Metadata: Data about data.
Listing: A collection of information describing a good or service that a student wants to sell or rent. It includes data like date, title, description, and various fields related to the type of listing.
Transaction: A successful sale or rent from one student to another, which was facilitated by a listing.
Poster: The student who creates or 'posts' a listing so other students can view it.

2. Project Description

2.1 Selling

The app will provide a reasonable interface for entering information on items users want to sell. The users will be able to enter information such as the item description, category, and price. The app will also allow users to select their preferred contact method for the item. Once the information has been posted, users will be able to view the current offers or remove the item from listings. Once the item has been sold, the seller will be able to rate and post a review on the buyer.

2.2 Buying

The app will allow buying users to view, filter, and sort items by information such as category, price, and seller history. During searches, the items will be displayed in a list with their basic information. Selecting an item displays the detailed description of the item, as well as seller information and ratings. Buyers will be able to rate and review the seller of an item they buy.

2.3 Users

The app will require users to log in with a crimson.ua.edu email, which will be verified during acount creation. Users will have preferred settings for contact information and alerts. The server will also collect and make available buyer/seller ratings and reviews.

3. Requirements

3.1 Functional

The app should provide an account for each user. The app must use a “crimson.ua.edu” email address to verify the student is from UA. The app must also allow each user to configure settings for their account, which at least includes contact settings. The user can choose whether others can contact them regarding their listings via phone, push notification, or email.

The app should allow users to rate transactions with other users. It should also allow users to view these transaction and rating histories for another given user.

The app should allow users to post listings to sell their textbooks, football tickets, and subleases. It should allow users to view these listings from other users. It should allow the user to view a summary page of multiple listings and a detailed page of a single listing. It should allow users to sort the summary page by date and popularity.

The app should allow users to contact the poster in order to purchase the listing.

The app should provide different categories of listings. The app should include fields specific to each category and make use of these fields whenever relevant, including posting a listing and searching for listings.

The app should allow users to search within a category for listings whose fields match certain criteria.

The app should provide push notification support to users. Users should be able to receive push notifications when others want to buy or make an offer for their listing. Users should be able to contact other users using push notifications. Users should be able to register for push notifications whenever listings matching certain criteria become available.

3.1 Non-Functional

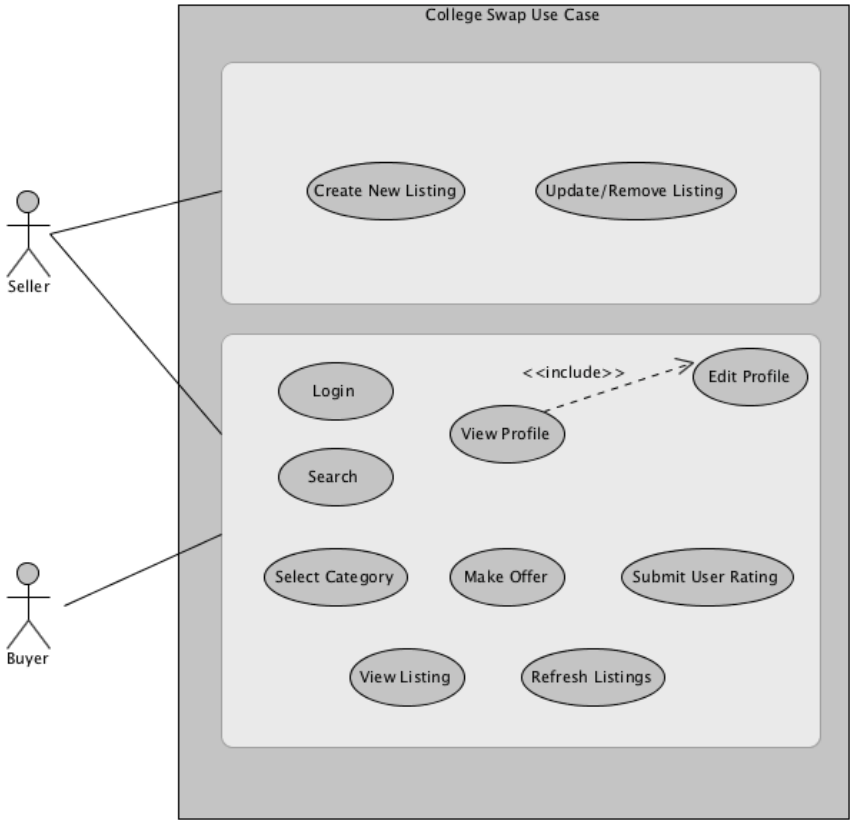
The app should provide security and confidentiality for contact information.

The app should provide a reasonably responsive user interface.

The app should be portable across a large variety of android phones.

4. Diagrams

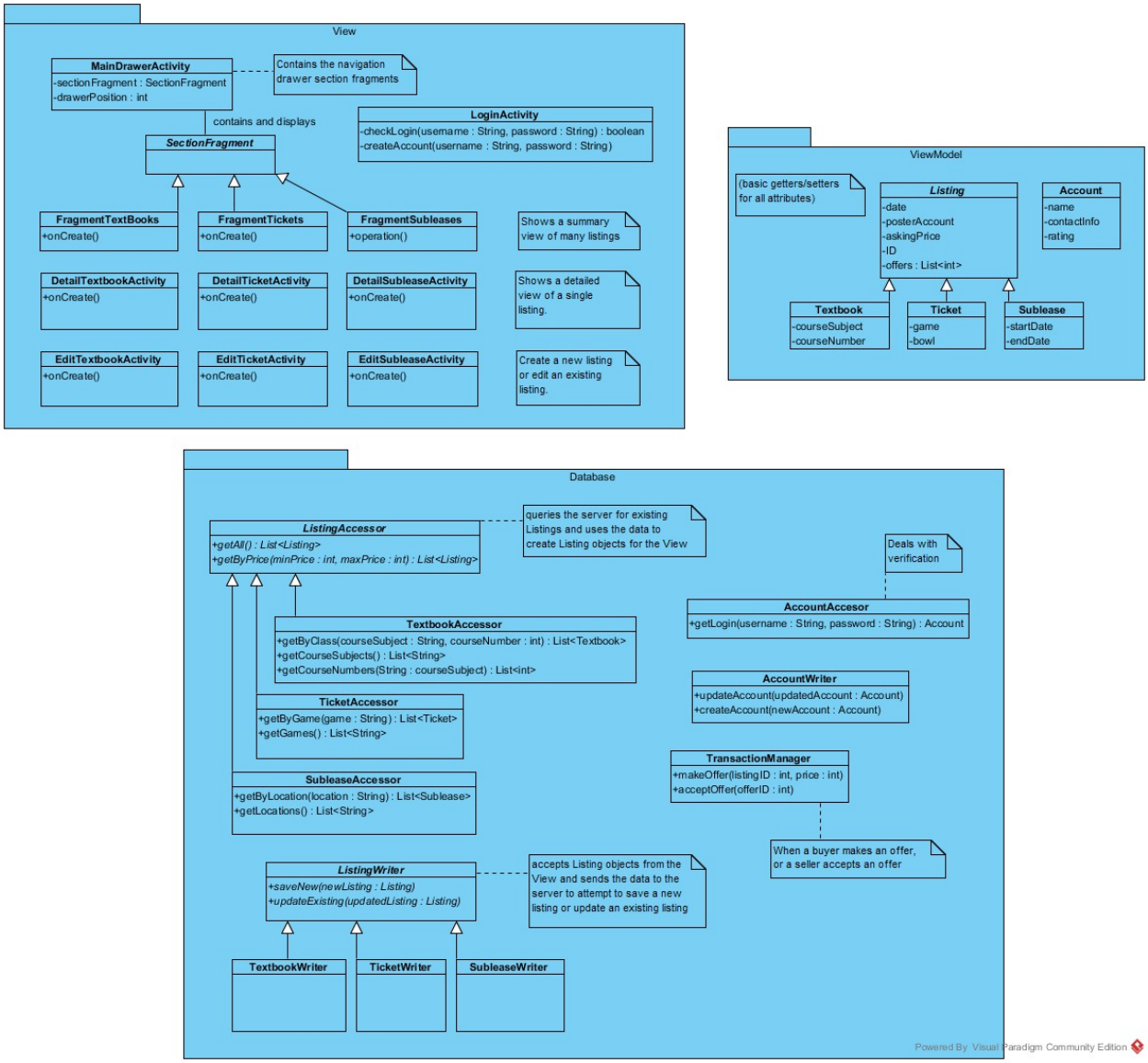
4.1 Use Cases



Use Case Name:	Post a new listing
Actors:	Seller
Summary:	This use case describes the basic way that a seller would interact with the application.
Basic Flow:	1. A seller logs into the application using their '.edu' email address. 2. The seller selects to create a new listing and fills in all the required information. 3. The seller selects to submit the posting. The new listing is then added and available for other users to see. 4. The use case terminates after the listing is submitted.

Use Case Name:	Find and purchase a listing
Actors:	Buyer
Summary:	This use case describes the basic way that a buyer would interact with the application.
Basic Flow:	1. A buyer logs into the application using their '.edu' email address. 2. If the buyer knows what they are looking for they can select to search for it specifically. It will then show all of the listings that match what the buyer searched for. Skip to step 4. 3. The buyer selects a specific category and all the listings for that category are displayed. 4. The buyer selects to view a specific listing. 5. The buyer selects to make an offer and a notification is sent to the seller. 6. The use case terminates at this point.

4.2 High-Level Class Diagram



Our class diagram consists of three packages- the View, ViewModel, and Database package. The View package consists of the classes Android needs to display activities. The ViewModel package consists of classes which provide the information needed by the View package. The Database package is responsible for generating and updating ViewModel objects for the View. We are trying to follow the Model-View-ViewModel architecture pattern, using Database as the Model.

The View package has largely been determined by the Android APIs. We will use a navigation drawer layout for switching between different types of listings. The navigation drawer layout needs a main activity to host the various sections (or pages), and MainDrawerActivity fulfills this purpose. The sections hosted by MainDrawerActivity are not Activities, but Fragments. We will call them SectionFragments. This allows Android to switch

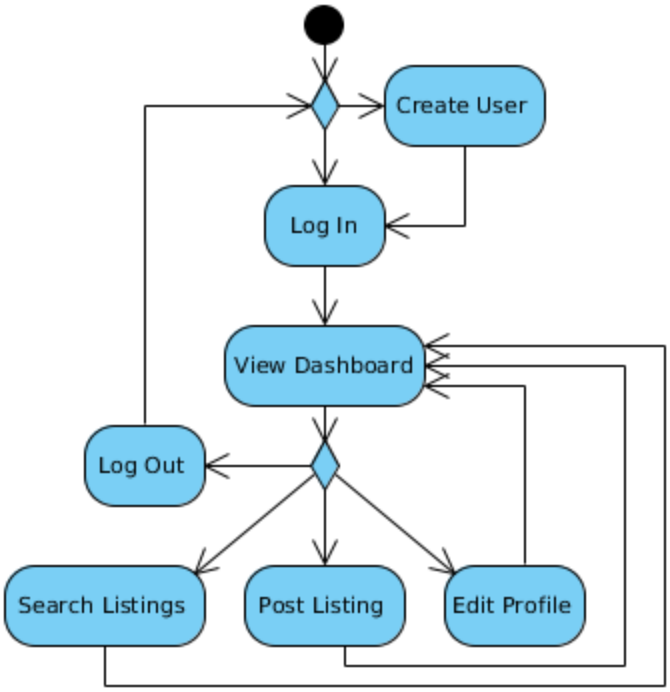
between sections without reloading the whole Activity. There are three subclasses of SectionFragments- FragmentTextbooks, FragmentTickets, and FragmentSubleases. These Fragments show a summary list view of many listings. They each have an onCreate() method like an Activity, allowing them to perform initial setup when loaded. For each type of listing, there is also a corresponding Activity to view the details of a single listing and to edit a single listing. The edit Activities will also allow us to create a new listing.

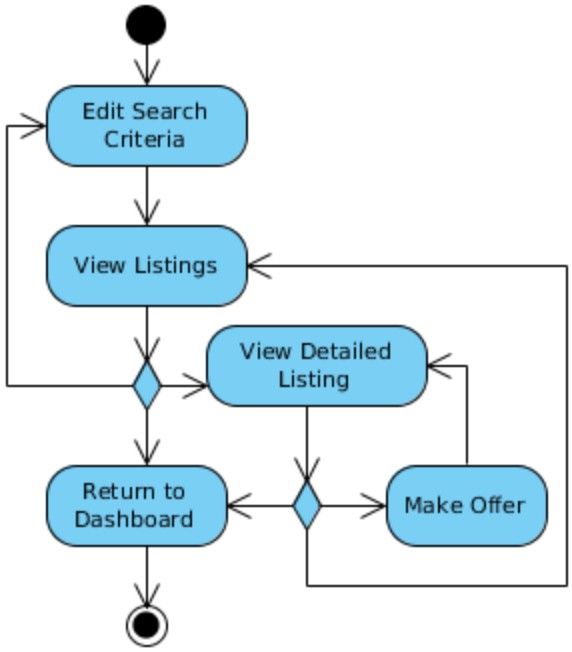
The ViewModel package consists of Plain Old Java Objects (POJOs). They don't extend any class, and they only provide getters and setters. These classes model various types of data, such as Listings and Accounts. They have fields to store any kind of data needed to display and store the listing. Listing subclasses have fields relevant to the specific type of Listing, such as course number or football game. Once the View has a reference to a ViewModel class, the View can simply call the ViewModel getters and do whatever Android requires to display the data.

The Database package is responsible for generating ViewModel objects containing previously stored data. Accessor classes allow accessing the data by generating ViewModel objects. Writer classes allow writing or storing the data by accepting ViewModel objects and saving their data to the database. There are subclasses of ListingAccessor and ListingWriter for the three types of Listings. The ListingAccessor subclasses provide methods to get Listings filtered by fields specific to that type of Listing. For example, the TicketAccessor provides a method to get Ticket listings for a certain game. The AccountAccessor class handles verification, and the AccountWriter class allows updating user preferences and creating new accounts. The TransactionManager handles making and accepting offers. By defining these database classes early in the development process, we can quickly define these methods to return ViewModel objects with mock data. This will allow us to simultaneously begin working on the Android View and the server side component. Once we are ready to integrate the app with the server, we can change the content of these methods without affecting the rest of the app.

4.3 Activity Diagrams

The first activity diagram shows the high level steps when the user starts the app. After creating an account (if needed) and logging in, the user can view the dashboard and access various features. The second activity diagram shows the steps involved in searching for listings and making an offer once a listing has been located.





In these activity diagrams, we have the activity to post a listing and to edit and view your profile. The first diagram shows you how to post your own listing, which first gives you the option to select TextBooks, Football Tickets, and Subleases. Then it shows you the process of selecting which item you want to purchase. The second activity diagram is the activity in which the user will view and edit their own profile. There are two options which the user can choose from, and within those options the user can edit their information as needed or just view their recent buying and selling history.