Software Rquirements Specification

College Merchandise

November 3, 2016

Devin Foulger
Hector Trujillo
Bryan Liauw

Contents

1 INTRODUCTION

This section will provide an overview of what is included in this Software Requirements Document (SRS).

1.1 Purpose

This SRS document will serve several purposes. First it will define the requirements for the ”College Merchandising” application that team ”DHB Solutions” will be developing for their CS Capstone project for eBay. The document will detailing the constraints, features, dependencies and performance of the application in a manner that clearly demonstrates what the client will be receiving.The intended audience for this Software Requirements Specification (SRS) will be the team, eBay, and the instructors of the computer science senior capstone class at Oregon State University.

1.2 Scope

The Android Application ”College Merchandising” will allow users to browse and shop for college merchandise that are for sale through eBay. The application will be published to the Google Play Store and will be optimized for Android’s latest operating system ’N’.

This application will also demonstrate the use and functionality of eBay’s public APIs by allowing users to search eBay’s inventory of items. The application will allow users to filter out their search results by different criteria. The application will also allow users to purchase items from eBay.

1.3 Definitions, acronums, and abbreviations

Below are definitions for terms that will allow for proper interpretation of the requirements document.

TABLE 1
Definitions

Term	Definition
API	Application Programming Interface, allows developers to develop applications that connect to eBay's large inventory of items.
Shop	Search for and view college related items on sale through eBay.
Vendors	Individuals or companies that sell or will sell items on eBay
Filters	Search criteria that will allow the user to more easily find the item that they are looking for.
Third-Party Developers	Individuals outside of eBay that will be using eBay's public APIs to create their own applications.
Application	An Android application that uses eBay's public APIs.
Interface	The User Interface.
Bugs	Errors in the API code that may be discovered during use.
College Merchandise	Items such as school supplies for Higher Ed, fan gear, etc...
User	Any one who will be using the Android Application utilizing eBay's APIs.
Client	eBay
BIN	Buy It Now
N	Android's new Nougat operating system
M	Android's Marshmallow operating system

1.4 Overview

The remainder of this document will provide an overview of the functionality that the ”College Merchandising” application will offer along with the interaction that it will have with other systems. The next chapter will identify any constraints that may prevent success for this project.

2 OVERALL DESCRIPTION

This section will provide the reader with an overview of the entire application. The application will be explained in a way that the functionality is easily identified along with any constraints.

2.1 Product Perspective

The android application will communicate with eBay to provide the user with the opportunity to shop through their inventory of college related merchandise. The communication will be accomplished with the use of eBay's APIs. This communication is important to provide the user with a seamless shopping experience without having to switch between different applications.

Since no data regarding the inventory of items that eBay provides will be stored by this application, there will be a constant need for an Internet connection to be able to pull search results from eBay's inventory.

2.1.1 User Interfaces

The user interface will mimic what people already know about eBay and other similar shopping experiences. This will include standard things such as a place to sign in, a place to search for items, and a place to view items. Signing in could consist of multiple pages for existing users or for new users. Searching could also vary depending on search settings like a basic search or advanced searching. When viewing items, there could be several kinds of list views. We would also have specific college-related images to help users to find the item that they are looking for. Since our application is limited to college merchandises, we will have some visual appeals that reflect what the application is about.

2.1.2 Hardware Interfaces

The devices that we use must be compatible with M and N os.

2.1.3 Software Interfaces

We will be using the new Android operating system, N. We will also be making use of a wide variety of eBay's new APIs. Some of these are apart of their new Buy APIs. These ones include the Browse and Order APIs. The Browse API will allow us to search for items and find different item feeds. The Order API will allow us to have some form of item checkout and order tracking. Will we also make use of their Finding API, which will allow us to search for items. Together, these APIs will allow us to connect with eBay's large collection of items.

2.1.4 Communication Interfaces

The device should always have some form of connection to the internet. This means the device can connect in one of two ways. The first being through the devices WLAN. The second being that it could connect using its cellular data.

2.1.5 Operations

There will be many user initiated operations. These include searching and purchasing. Searching means that the user will be finding items with specific search criteria. The items that they find could be related to different schools, or have specific colors. Purchasing means that the user is able to select an item that they want and pay for it.

The application should only be in use when a user is actually using it. There are no periods of unattended use. The only time that might be considered as unattended use, is when the application is gathering and displaying search results to the user.

2.2 Production Functions

There are a couple of major features that this application will support. It will allow for users to search for items and to purchase them. Users will be able to browse eBay's large collection of college merchandise that they have for sale. The application will allow users to narrow down what they are looking for by allowing them to use filters such as types of items, condition, cost, college specific, by terms, and so on.

The results of these searches will be displayed in either a listview that simply provides minor details regarding the item or thumbnail views that provide an image of the product along with a few details. The number of results displayed at once

will vary depending on how the user wants the results presented to them and by the screen size of the product that they will be using. The application will also allow the user to sort the order in which the items are displayed. The sort options will be similar to what eBay provides on their current website. This includes sorting by price and time (ending soon or recently posted).

2.3 User Characteristics

Users will include individuals who are looking to find great deals on college merchandise online. The typical user will be someone who is looking for college merchandise. The actual user group could range between many different people. The main audience should be individuals looking for college merchandise and could be incoming college students, or people who are already college students. Users could also be parents, grandparents or other individuals who are looking to help out a relative or friend in searching for any college merchandise that the individual might need.

Since are users are going to use this application on a mobile device, our main focus to the user would be to create an interface that is simple and straightforward. The user should be able to learn and familiarize with the interface in a short period of time. Furthermore, the processes should be fast; the window between processes should be short to keep the attention of the users.

2.4 Constraints

Android's new operating system, N OS, will be the biggest constraint. This is because the operating system is new and is not present on many of today's top Android devices. This also means that the documentation may not be up to date with the new operating system. Older APIs created by Google may also be deprecated.

Large loads on the application might put strain on lower quality machines. Example of large loads would be a lot of activities and threads and overly large image.Testing might also be a constraint since a virtual machine will take a huge amount of computer resources. For a better testing environment, the usage of an actual physical smartphone might be needed.

2.5 Assumptions and Dependancies

An assumption and dependency for this project will be the availability of the Android operating system N and a device that can run it. A secondary assumption is that the mobile phone which the application will be running on will have an active internet connection. A third assumption would be that users of the application will have an eBay account that they will be able to use to save searched items and even checkout and complete a purchase through the application. A dependency that the application will have is that we highly depends on the connection to the Ebay database. If the database server is down, we might not get the app to work.

3 SPECIFIC REQUIREMENTS

3.1 Functional Requirements

This section defines all the requirements that are fundamental actions to the application.

3.1.1 Functional Requirement 1

Name: Download application

Description: The user must be able to download the application from the Google Play store.

Input: None.

Output: Application will be installed on the user's device.

Dependencies: The user has an Android device that can support N os.

3.1.2 Functional Requirement 2

Name: Update application

Description: This should allow the user to update the application via the Google Play store.

Input: None.

Output: Update application.

Dependencies: The user has an Android device that can support N os.

3.1.3 Functional Requirement 3

Name: Sign in

Description: This should allow the user to sign in to our application using and eBay account.

Input: User's password and username for their eBay account.

Output: Confirmation that user signed in.

Dependencies: Functional requirement 1.

3.1.4 Functional Requirement 4

Name: Determine search criteria

Description: The user should be able to select what they want to search for. The user should be able to select between many options. These include search by BIN (Buy It Now), price, school, color, condition, and bidding. They should also be able to use different search criteria in combination.

Input: A combination of the above stated search criteria.

Output: None.

Dependencies: Functional requirement 1.

3.1.5 Functional Requirement 5

Name: Search by BIN

Description: Allow the user to search by items tagged as BIN (Buy It Now).

Input: Allow the user to select BIN as a search option.

Output: Search displays that the items displayed are only BIN items.

Dependencies: Functional requirement 4.

3.1.6 Functional Requirement 6

Name: Search by price

Description: The user should be able determine a price range in which their item should cost. There should be a minimum

and maximum price.

Input: A minimum price, a maximum price, or both.

Output: None.

Dependencies: Functional requirement 4.

3.1.7 Functional Requirement 7

Name: Search by school

Description: The user should be able to search for a specific school or select from a pre-determined list of schools.

Input: The specific school the user is searching for.

Output: None.

Dependencies: Functional requirement 4.

3.1.8 Functional Requirement 8

Name: Search by condition

Description: The user should be able to search for items ranging from very used to new.

Input: The condition of the item the user is searching for.

Output: None.

Dependencies: Functional requirement 4.

3.1.9 Functional Requirement 9

Name: Purchase

Description: The user should be able to purchase an item that they have searched for.

Input: The item that is going to be purchased.

Output: Some receipt or confirmation that the item has been ordered.

Dependencies: Functional requirement 4.

3.1.10 Functional Requirement 10

Name: Display search results

Description: Search results should be displayed to the user given the user's search criteria. Search results should also be displayed according to how the user specifies. That should be able to display results from lowest price to largest price, or from BIN to bid.

Input: None.

Output: A list of items organized the way the user has specified.

Dependencies: Functional requirements 4.

3.1.11 Functional Requirement 11

Name: Display search results (no results found)

Description: This should display a list that doesn't include any items and will have a simple message stating that no items could be found.

Input: None.

Output: A simple message that states nothing could be found.

Dependencies: Functional requirement 4.

3.1.12 Functional Requirement 12

Name: Item view for a selected item

Description: When a user selects an item, it should be displayed in its own view with greater detail.

Input: Item from listview.

Output: A page with complete detail of the item.

Dependencies: Functional requirement 10.

3.1.13 Functional Requirement 13

Name: Shopping Cart

Description: This is where items would be saved before users want to purchase the items. A user should be able to view their shopping cart with the items they have selected.

Input: Items they have selected.

Output: A page with the items they want to purchase.

Dependencies: Functional requirement 4.

3.2 Performance Requirements

This section will document that the requirements provide a measurement of how the application will perform.

3.2.1 Performance Requirement 1

Name: Search feature

Measurement: The search feature should be easy to find and easy to use.

Dependencies: None.

3.2.2 Performance Requirement 2

Name: Usage of search feature

Measurement: The search criteria provided should be easy to understand and simple.

Dependencies: Performance requirement 1.

3.2.3 Performance Requirement 3

Name: Usage of search results in a listview

Measurement: The results displayed in a listview should provide the user with easy to understand results. Things like the item name, price, and school should be provided to the user.

Dependencies: Performance requirement 1.

3.2.4 Performance Requirement 4

Name: Purchase feature

Measurement: The purchase feature should mirror what users are already familiar to, such as a shopping cart with checkout buttons.

Dependencies: None.

3.2.5 Performance Requirement 5

Name: Usage of purchase feature

Measurement: Users should recieve a receipt that the feature worked.

Dependencies: Performance requirement 4.

3.2.6 Performance Requirement 6

Name: Sign in feature

Measurement: The user should be able to find the sign in feature easily. They should also be able to sign in to their eBay accounts in a similar way to the online experience.

Dependencies: None.

3.2.7 Performance Requirement 7

Name: Usage of sign in feature

Measurement: The user should know that they are signed in.

Dependencies: Performance requirement 6.

3.3 Software System Attributes

This section will define how are application will meet the software system attributes.

3.3.1 Reliability

The application should be able to search and purchase items according to what has been specified by the user.

3.3.2 Availability

The application will be available only to users who have Android devices. These users must also have devices that are compatible with the new operating system, N and the current operating system, M. Other than the two constraints above, any user who has a valid internet connection and access to the Google Play store, should be able to download and use our application.

3.3.3 Security

Security should not be a problem, as the APIs are using HTTPS protocols for recieving and sending data.

3.3.4 Maintainabillity

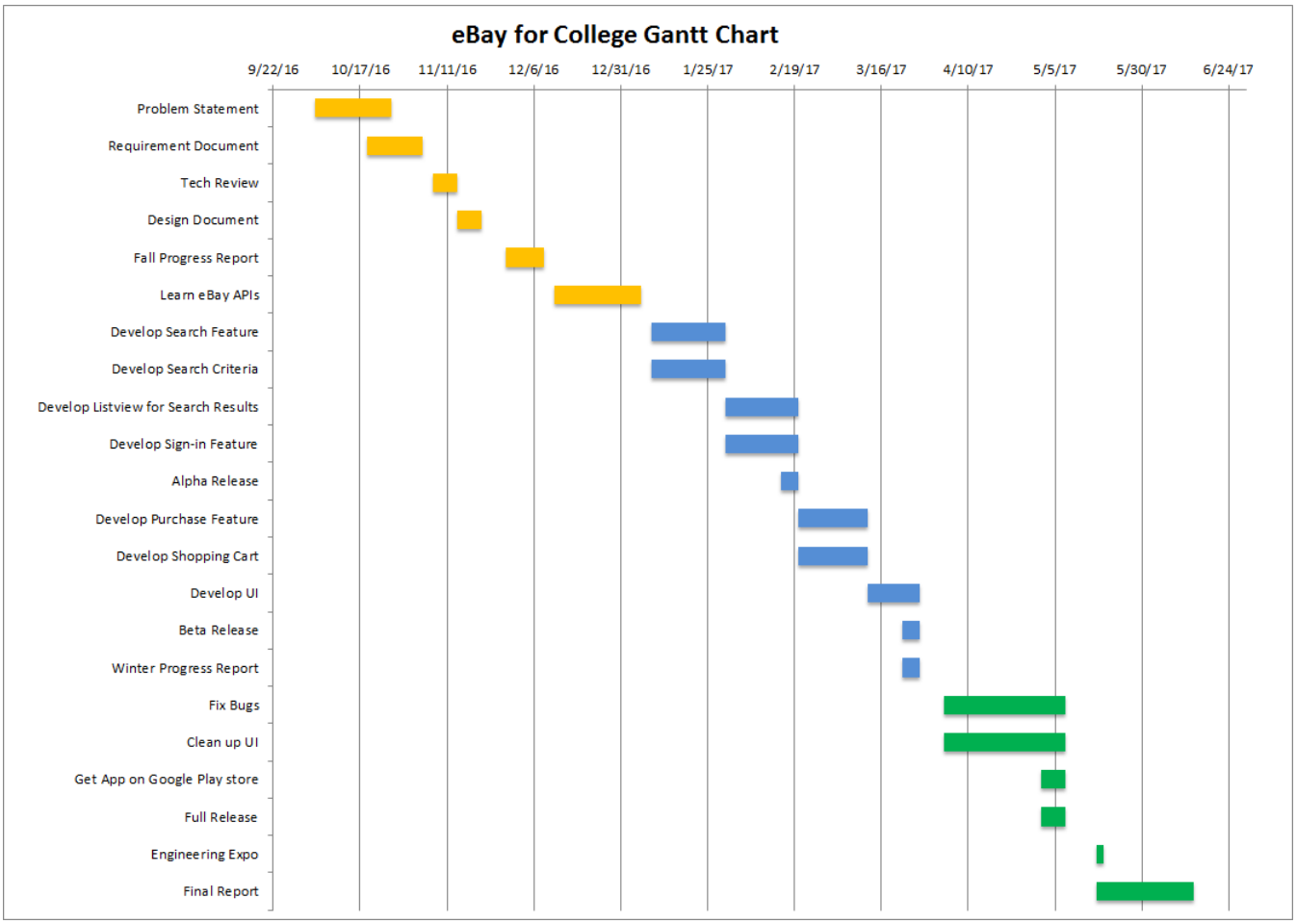
In order to create an application that is easily maintainable, we must be sure that all the parts are separated in an easy to manage fashion. We would avoid over-coupling objects and classes and use an MVC design pattern. Our code will also be testable, readable, and commented well.

3.3.5 Portability

We acknowledge the fact that our application will not be very portable. This is because we a specifically developing the application for Android systems. This means that our language is also going to be in Java.

4 Appendices

4.1 Appendix I: Gannt Chart for Application



4.2 Appendix II: Signatures

Luther Boorn

_____

Signature

_____

Date

Devin Foulger

_____

Signature

_____

Date

Hector Trujillo

_____

Signature

_____

Date

Brian Liauw

_____

Signature

_____

Date