

Mobile App for Books and Electronic Devices Review and Look-up using OCR

Group: 2018959

Supervisor: Hamid Timorabadi

Administrator: Ross Gillett

Group Members: Wei Cao, Menglu Li, Jiehao Liu, Zengnan Liu

Submission Date: October 25th, 2018

Table of Contents

Executive Summary

1.0 Project Description	1
1.1 Background and Motivation	1
1.2 Project Goal	2
1.3 Project Requirements	2
1.3.1 Primary Functional Requirements	2
1.3.2 Subfunctional Requirements	3
1.3.3 Constraints	3
1.3.4 Objectives	4
1.4 Validation and Acceptance Tests	5
1.5 Validation and Acceptance Tests Summary	5
2.0 Technical Design	9
2.1 Possible Solutions and Design Alternatives	9
2.1.1 Alternatives of approaches to mobile application development	9
2.1.2 Smartphone operating systems selection between Android and iOS	9
2.1.3 Local server or a third-party service for user account services	10
2.1.4 Database Selection: relational or non-relational	11
2.2 System-level Overview	11
2.3 Module-level Descriptions	13
2.4 Assessment of Proposed Solution	18
3.0 Work Plan	20
3.1 Work breakdown Structure and Gantt Chart	20
3.1.1 Work Breakdown Structure	20
3.1.2 Gantt Chart	23
3.2 Financial Plan	24
3.3 Feasibility Assessment (resources, risks)	25
3.3.1 Skill and resources	25
3.3.2 Risks	26
4.0 Conclusion	27
References	28
Appendices	33

Executive Summary

During a shopping experience, consumers often found it difficult to make a decision whether they should make the purchase. Sometimes they might be doubtful about the price, or perhaps they would like to compare the product or see some user reviews about it, but simply could not find any online in time. This problem worsens for those who are not fast enough or incapable of typing on their smartphones, such as the elderly and visually impaired persons, as typing is usually the only input method to perform a web search. There are some existing smartphone applications that address parts of the issue, but none of them provides “one-stop service”. Therefore, the project goal is to develop an integrated solution on a smartphone which help shoppers make more informed decisions.

The proposed mobile application would first identify the product by taking a photo of a book cover or the price tag of an electronic device and scan it using Optical Character Recognition (OCR) technology. The application would then retrieve user reviews and price comparison from online selling sites such as Amazon or Best Buy. Further features include user account registration, which enables users to store their favorite items in a wishlist and to post their own reviews of products directly on the app.

The high-level architecture of the application consists of the frontend, backend, and database sections. The frontend is further divided into five layers, from the “user input” layer which offers three different input methods, to the “functional menu” layer in which users can store items for further research.

Each member will work towards separate design milestones according to the work breakdown structure and schedule depicted by a Gantt chart. Each of the proposed functions, constraints and objectives are verifiable by a series of acceptance tests. The team also assessed both the cost of capital and the work time required for each task in the design process. Should any complications emerge that would considerably delay the progress of the project, the design team would adjust to predetermined alternative design approaches to resolve the issue.

1.0 Project Description

1.1 Background and Motivation

During shopping, people often find themselves too busy to search for a particular item they saw in the store by typing it on their phone. Typing letter for letter on touch-screen smartphones is also physically difficult or cumbersome for many users, especially the elderly and visually impaired persons [1]. Thus, easier methods for input, internet searches, and comparisons of products become a growing interest among smartphone users [2].

There are some scanner applications already existing in the market, such as Seeing AI [3]. Seeing AI is a special application to help people who are blind or visually impaired read any scanned item out loud. It is capable of reading the item and providing basic information about the product. However, in many cases, customers are looking for a more detailed explanation as well as other users' experiences with the searched item. Thus, user reviews or price comparison services are becoming a necessity on top of the regular searching services.

Other state-of-the-art applications such as Trivago [4] and Vivino [5], provide customer reviews and price comparison about searched products directly, but their search is limited to only hotels and wines, respectively. These apps offer a well-rounded service in their own fields, but not on books or electronic devices.

Our motivation is to provide a platform for users to look at item prices from different online selling sites directly, without searching multiple times in each respective site. This will greatly simplify the procedure for users during their shopping experience. The application will offer an integrated suite of services including price comparison, user reviews, item price sorting, tracking and storing.

Our application supports the search services of books and electronics because items in both categories are often sold on various platforms. All these platforms often have discounts and sale seasons for diverse products they offer. Other merchandise, such as clothing, does not need price comparison since there are often exclusive to one seller. For example, it is impossible to buy new

Zara T-shirts from anywhere else other than the Zara store or its own online website. There is little need for a price comparison since these items are hardly found elsewhere.

1.2 Project Goal

The goal of this project is to develop a mobile application that scans a book cover or a price tag of an electronic device and instantly get reviews and price comparison from online selling sites.

1.3 Project Requirements

In this section, functional requirements will be introduced according to three aspects: Input Method, Searching and Price Comparison, and User Functions. The Input Method field lists different ways users can input data for the app to parse. The Searching and Price Comparison field summarizes the core services that the app shall offer; User Functions field specifies the UI/UX in terms of exactly the tasks users can accomplish in the app.

1.3.1 Primary Functional Requirements

Table 1 - List of primary functional requirement

ID	Field	Primary Project Requirement
1	Input method	Identify and parse 5 or more commonly-used English fonts with OCR [6]
2	Input method	Identify books by scanning ISBN barcode
3	Input method	Identify books or electronic devices from users' manual input (by typing)
4	Searching & Price Comparison	List average rating from 3 or more online-selling platforms (if applicable) [7]
5	Searching & Price Comparison	Highlight top customer reviews from 3 or more online-selling platforms (if applicable)

6	Searching & Price Comparison	Retrieve price information from 3 or more online-selling platforms (if applicable)
---	------------------------------	--

1.3.2 Subfunctional Requirements

Table 2 - List of sub-functional requirement

ID	Field	Subfunctional Project Requirement
7	Input method	Allow users to manually correct the scanned result
8	Searching & Price Comparison	Rank platforms by price of the searched book or electronic device
9	User Functions	Provide a “Purchased” tab which stores purchased items
10	User Functions	Provide a “Favorite” tab which saves users’ highlighted items
11	User Functions	Provide a “Recent Scans” tab which keeps the last 20 scanned items
12	User Functions	Support book links sharing between users and external social networks

1.3.3 Constraints

The table below describes regulations that the design shall not violate.

Table 3 - List of project constraint

ID	Field	Project Constraint
13	Security	The camera sensor data must not disclose outside the app. And must explain how the data will be used [8].

14	User Functions	No spam or repetitive information. Send notification for wishlist items price change only when necessary, i.e. maximum one notification per day with a collection of price change messages [9].
15		Require permission to access device's photo library, or other users data, or even using the camera [10].

1.3.4 Objectives

The table below lists the aspects that the team desire to complete in the design process. The design process should meet these specifications.

Table 4 - List of project objective

ID	Field	Project Objectives
16	Input method	Identify and parse 10 or more commonly-used English fonts with OCR [6]
17	Input method	Parse existing images from device's photo library
18	User Functions	Enable user registration through email
19	User Functions	Provide a "Wishlist" tab which notifies users whenever an item in it goes on sale
20	User Functions	Recommend books of the same author or a similar genre based on user preferences
21	User Functions	Apply machine learning technologies to recommend relevant products to users, based on interests, purchased items, and wishlist.

1.4 Validation and Acceptance Tests

The verification matrix associates each of the project requirements, constraints and objectives with a specific verification method. There are two verification methods defined below.

1. **Test:** conduct several practical test procedures for the App to run in real-time. By selecting random test cases covering as many aspects and circumstances as possible, the test result will indicate whether the App meets the tested functionality.
2. **Review of design:** verify the design by examining if the tested feature or service exists in the app. Also, this method checks if the design meets any regulations set by the publisher platform on a pass/fail basis.

1.5 Validation and Acceptance Tests Summary

Each of the project requirements can be verified in the verification matrix via one or both methods defined above.

Table 5 - Project requirements verification matrix

ID	Method	Detailed description of Validation
1	Test	Books: Open the camera from the application, scan a book and check if the OCR successfully identifies characters from the fonts and finds the correct book. Then, check if the search results of the customer reviews and prices match this book. Test the OCR ability by scanning more than 5 books each with a different cover font. If at least 5 books with different fonts are identified, the test is considered pass. Electronic Devices: Repeat the same steps by scanning price tags from Best Buy, Canada Computer, The Source, and Canadian Tire.
2	Test	Books: Open the camera from the application, scan a book and check if the OCR reads the barcode and finds the correct book. Then, check if the search

		results of the customer reviews and prices are for this book. Prepare 10 books from different publishers. The test is considered pass if 9 books are successfully identified. (Electronic devices are not applicable for this test)
3	Test	<p>Books: Open the manual input interface on the application. Enter a book name, such as “The Great Gatsby”. Check whether the app returns the correct book. Repeat this test 4 times with different books to confirm the validity of the test result.</p> <p>Electronic devices: Open the manual input interface on the application. Enter a electronic device product name, such as iPhoneXs. Check whether the app returns the correct item. Repeat the same step with 5 different brand names and models.</p>
4	Test	Scan and search a book or price tag. Check if the returning search result contains ratings from more than 3 online-selling platforms. Scan 5 more books or price tags to validate the result.
5	Test	Scan and search a book or price tag. Check if the returning search result contains user reviews from more than 3 online-selling platforms. Scan 5 more books or price tags to validate the result.
6	Test	Scan and search a book or price tag. Check if the returning search result contains prices from more than 3 online-selling platforms. Go to the listed web platform respectively and check if the prices match. Scan 5 more books or price tags to validate the result.
7	Review of design	Scan a book or price tag, and wait for the OCR scanning result text to appear. Click the text and check if a keyboard is shown to enable the user to modify the text. When finished modification, click search again to see if the app displays books or electronic devices according to the modified text.

8	Test	Scan and search a book or price tag. Under the price section, check if the websites are ranked (low to high) according to the prices. Scan 5 more books or price tags to validate the result.
9	Test	Click the “Purchased” tab and clear all items inside. Search a few items and mark them as Purchased. Click the “Purchased” tab again and check if the marked items are saved inside.
10	Test	Click the “Favorite” tab and clear all items inside. Search a few items and mark them as Favorite. Click the “Favorite” tab and check if the marked items are saved inside.
11	Test	Click the “Recent Scans” tab and clear all items inside. Search a few items. Click the “Recent Scans” tab again and check if the searched items appear inside.
12	Review of design	Search an item ¹ , under each price provided by different platforms, check if there is a hyperlink that redirects to the selling platform.
13	Review of design	Check if the camera sensor data and images are saved only locally. Limit upload traffic to prevent any data leakage.
14	Review of design	Save a few items in the Favorite list. Check if the app sends more than the maximum number of notifications allowed in one day. Ensure that the notification content is about legit price change only. Provide options for users to disable notifications about price change.
15	Review of design	Install the application on a device. For the first time using the application, turn on the camera in the app, check if a prompt appears asking the permission to access the device’s camera. Check if a second prompt appears asking for permission to access the photo library (for image saving).

16	Test	<p>Books: Open the camera from the application, scan a book and check if the OCR successfully identifies characters from the fonts and finds the correct book. Then, check if the search results of the customer reviews and prices match this book. Test the OCR ability by scanning more than 10 books each with a different cover font. If at least 10 books with different fonts are identified, the test is considered pass.</p> <p>Electronic Devices: Repeat the same steps by scanning price tags from Best Buy, Canada Computer, The Source, and Canadian Tire.</p>
17	Review of design	<p>On the scanner interface, tap the button to access device photo library. Choose a photo² from the library, check if the OCR scanner identify correct text content.</p>
18	Test	<p>Tap the registration button. Enter an email as account username. Fill all the required fields to complete the registration. Check if an email about registration success is received.</p>
19	Review of design	<p>Tap the “Wishlist” tab, clean all the items there. Search a few items and mark them as Wishlisted. Tap the “Wishlist” tab and check if the marked items are stored inside.</p>
20	Test	<p>Scan and search a fiction book. Check if the result matches other fiction books and/or books of the same author. Repeat above steps for other genres. The test is considered pass if 5 genres are successfully parsed and matched.</p>
21	Test	<p>Save 20 random books in any of the tabs in the app. (Favorite, Purchased, Wishlist) Select any of the books after saving, and examine the book recommendations section contains books with related genre and/or the same author with the selected book.</p>

1. An “item” indicates a book cover or an electronic device price tag

2. A “photo” of a book cover or an electronic device price tag

2.0 Technical Design

2.1 Possible Solutions and Design Alternatives

The following are the four possible alternatives and their solutions according to application type, operating system, user identity services, and database options.

2.1.1 Alternatives of approaches to mobile application development

There are usually two approaches to develop a mobile application, which are native mobile apps or web apps [11].

Native mobile apps are built for using in mobile devices. After developers select application development platform, either Android or iOS, the app can only be run on this specific system. Native mobile apps always provide fast and responsive performance, which also bring high cost of implementation and maintenance [12].

Web apps are the internet-based application that can be accessible by the web browser of mobile devices, which indicates that web apps can be compatible with multiple platforms [11,13]. They are easier to build and maintain. However, web apps do not have some functionality compared to native apps, such as push notifications, and also the apps need internet connection to access [11].

2.1.2 Smartphone operating systems selection between Android and iOS

Both Android apps and iOS apps are possible solutions to our project goal. The following table compares Android and iOS operating platform in several aspects.

Table 6 - Comparison between Android and iOS operating systems

Consideration Aspects	Android	iOS
Development environment	Java is the programming language for Android, which is also the most commonly used programming language.	Swift is used as the programming language for iOS, which is invented by Apple.
Development Cost	Developing an Android mobile application takes more time because of fragmentation, which indicates a higher initial cost [14,15].	The average time to develop an Android app is 2-3 times shorter than developing an iOS app [14].
App Store Submission and Publishing	Easier procedure and shorter time to submit and publish an app.	Every submitted iOS app is tested by staffs of Apple company, which takes longer time to get reviewed and publish [16].
Return on Investment and Profit	Android apps has larger global share than iOS. However, Android targets customer in low-income areas, which leads lower revenues [14,15].	iOS apps have more engaged users, who are willing to spend more money in-apps [14].

2.1.3 Establishing a local server or using a third-party service such as Google Identity for user account services

In later stages of development, the application will be offering user account registration services. Implementing a dedicated server to handle user email registration requests gives the app flexibility and full control over any actions regarding user credentials [17]. However, the level of security required by the local server may prove to be difficult to achieve, even more so if compared with third-party services such as Google Sign-in [18]. Maintenance of the local server may raise unexpected problems, since the local server would have to remain online 24/7. Regarding third-party services, the team may need to integrate multiple sign-in methods as not all users would choose Google as their preferred account service provider [19].

2.1.4 Database Selection: relational or non-relational

The application will use MySQL as the relational database to manage user account information. Account information will be received in a well-constructed format since the user will enter those information according to what the registration of the application asks for. The advantage is that the structured data can be put into the database directly, and it is more efficient while retrieving the data [20].

In case that relational database is not feasible for this application, the app will use mongoDB as the non-relational database alternative. Because mongoDB is efficient at managing unstructured data and its performance increases as the database size grows larger, it may be more suitable for the program later in the project when we introduce customer reviews in the database [21].

2.2 System-level Overview

There are three components in the design of the mobile application, which are frontend, backend and the database (Figure 1).

The frontend subsystem uses camera to receives the request from users. Then, the data is transferred to the backend subsystem and is recognized by OCR module. The web crawler module performs the actions of browsing multiple online selling sites and collecting relevant production information based on user requests. After the backend subsystem gathers the information, it sends the searching result back to the frontend, which shows on the user interface.

Also, users can send their requests through control panel tabs on graphical user interface. These requests interact with the database component to store and retrieve data of application usage.

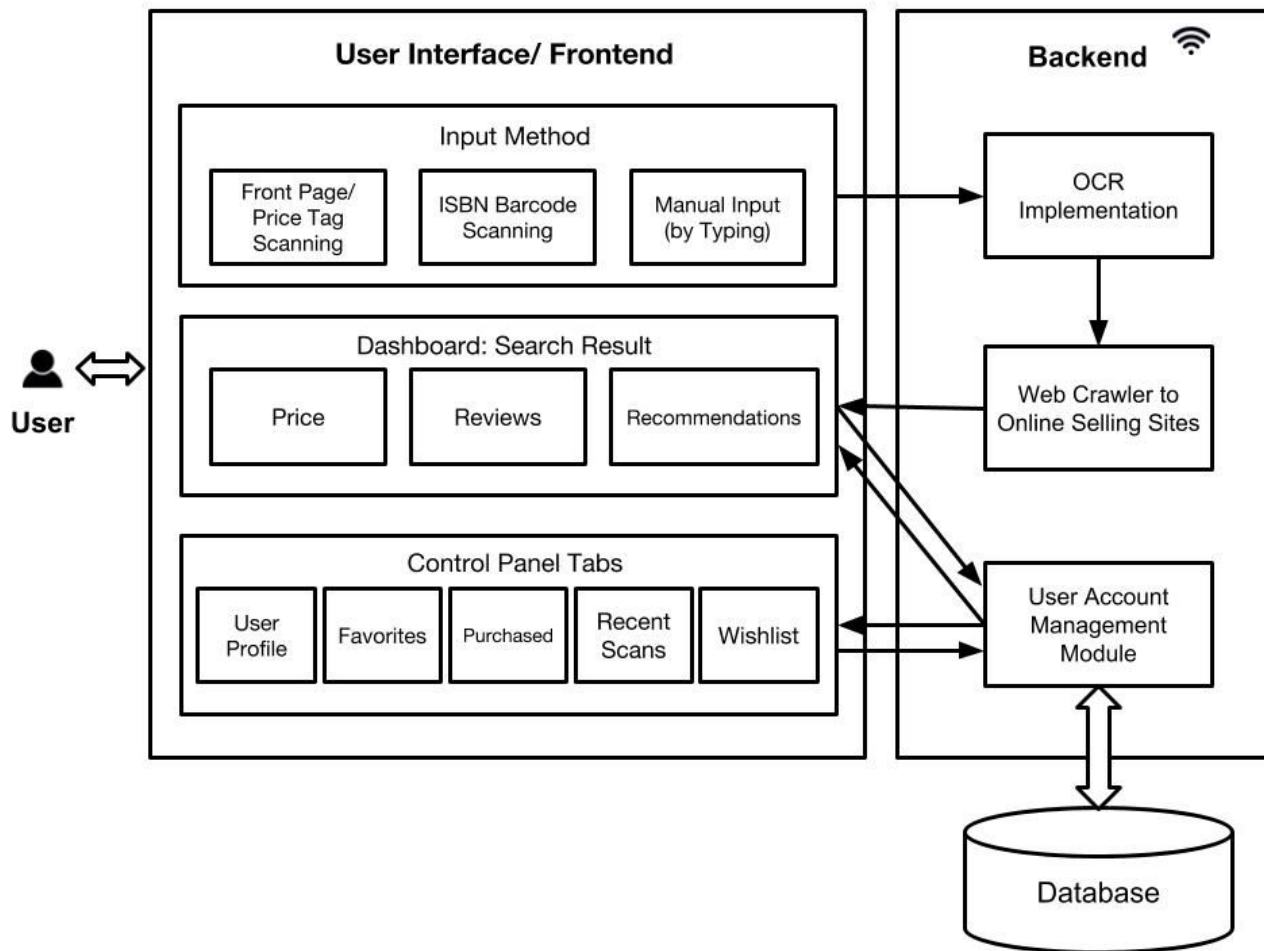


Figure 1. System level block diagram of the overall design

2.3 Module-level Descriptions

In this section, we divide the design into three main branches: User Interface, Backend, and Database. A detailed description for each module in the block diagram is provided.

2.3.1 User Interface/ Frontend

This user interface category is composed of three major subcategories: Input Methods, Dashboard Search Results, and control panel tabs. The following table describes the modules within each of the subcategories with a detailed explanation of their functionality, as well as input and output. A demonstration of the transition of user interface components is shown in figure 2.

Table 7 - A detailed descriptions of modules in frontend subsystem

Module	Input/Output	Function
Input Methods		
Front Page/ Price Tag Scanning	Input: <ul style="list-style-type: none">- An image of a book's cover or front page with a title- An image of an electronic device's price tag Output: <ul style="list-style-type: none">- Image data	<ul style="list-style-type: none">- Provides an input method that captures or receives input data in a form of image- Sends the image to backend OCR implementation Module- Requires permission to access the device's camera and photo library
ISBN Barcode Scanning	Input: <ul style="list-style-type: none">- An image of a book's ISBN barcode Output: <ul style="list-style-type: none">- Image data	<ul style="list-style-type: none">- Provides an input method that captures or receives input data in a form of image- Sends the image to backend OCR Implementation Module- Requires permission to access the device's camera and photo library
Manual Input (by Typing)	Input: <ul style="list-style-type: none">- Text data received through a device's keyboard Output: <ul style="list-style-type: none">- Text data	<ul style="list-style-type: none">- Provides an input method that receives input data in a form of text- Sends the text to backend OCR Implementation Module.- The text data is collected through a keyboard

Dashboard: Search Results		
Prices	Input: <ul style="list-style-type: none"> - Price information retrieved from Web Crawler Output: <ul style="list-style-type: none"> - Frontend: application price subsection 	<ul style="list-style-type: none"> - A frontend interface that shows prices of the searched product - Receives web crawling data from Web Crawler to Online Selling Sites Module - Display the three lowest prices in order
Reviews	Input: <ul style="list-style-type: none"> - Customer reviews retrieved from Web Crawler Output: <ul style="list-style-type: none"> - Frontend: application review subsection 	<ul style="list-style-type: none"> - A frontend interface that shows reviews of the searched product - Receives web crawling data from Web Crawler to Online Selling Sites Module - Display the top reviews from at least three sites
Recommendations	Input: <ul style="list-style-type: none"> - Product recommendations retrieved from User Account Management Module - Product recommendations retrieved from Web Crawler Output: <ul style="list-style-type: none"> - Frontend: application recommendations subsection 	<ul style="list-style-type: none"> - A frontend interface that provides recommendations of the searched product - Collects book recommendations from User Account Management Module - It requires Machine Learning technique to study the user's interests to produce results - Receives web crawling data from Web Crawler to Online Selling Sites Module - Display the recommendations from both modules
Control Panel Tabs		
User Profile	Input: <ul style="list-style-type: none"> - Text data of user's login credentials through a keyboard - User's data from the User Account Management Module Output: <ul style="list-style-type: none"> - User's profile interface 	<ul style="list-style-type: none"> - A frontend interface that provides a user's profile information - Receives the credentials and sends to User Account Management Module to validate information - Obtains validation results from the User Account Management Module - Grants access to the user if the credential is valid

	<ul style="list-style-type: none"> - Access to customized information 	
Favorites	Input: <ul style="list-style-type: none"> - Customized data from User Account Management Module Output: <ul style="list-style-type: none"> - Favorite items interface 	<ul style="list-style-type: none"> - Frontend interfaces that display some customized items - Collects the data from User Account Management Module - Requires the user login status to access the data
Purchased	Input: <ul style="list-style-type: none"> - Customized data from User Account Management Module Output: <ul style="list-style-type: none"> - Purchased items interface 	
Recent Scans	Input: <ul style="list-style-type: none"> - Customized data from User Account Management Module Output: <ul style="list-style-type: none"> - Recent Scan items interface 	
Wishlist	Input: <ul style="list-style-type: none"> - Customized data from User Account Management Module Output: <ul style="list-style-type: none"> - Wishlist interface 	

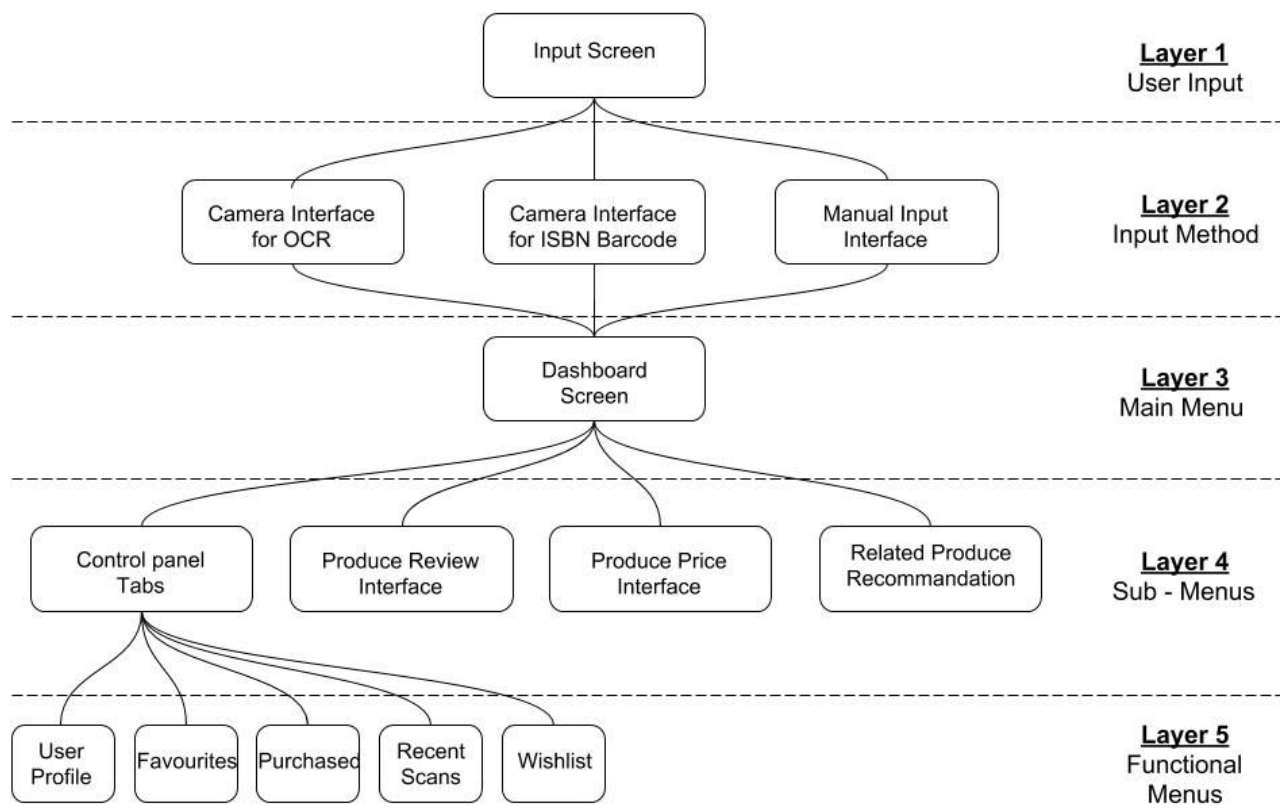


Figure 2. Breakdowns of User Interface Components

2.3.2 Backend

The backend subsystem consists of three modules: OCR Implementation, the Web Crawler and User Account Management module. The backend system interacts with both frontend and the database component.

Table 8 - A detailed description of modules in backend subsystem

Module	Input/Output	Function
OCR Implementation	<p>Input:</p> <ul style="list-style-type: none"> - Gets the image data from the frontend subsystem <p>Output:</p> <ul style="list-style-type: none"> - If succeeded, sends the input data to the web crawler module - If failed, recognizes the image, send the error signal back to frontend 	<ul style="list-style-type: none"> - Recognizes the text in images and converts into text data structure that can be sent - Directs the user to take another clearer image if the OCR cannot get a desired result - The recognized result should be a book name or a name of the electronic device - Sends the product name to the Web Crawler module
The Web Crawler	<p>Input:</p> <ul style="list-style-type: none"> - Receives the product's name from OCR Implementation Module <p>Output:</p> <ul style="list-style-type: none"> - Sends searching result of the product to the frontend Dashboard Module 	<ul style="list-style-type: none"> - Browses multiple online selling platform for the product - Gathers the product information such as prices, reviews and recommendation products
User Account Management	<p>Input:</p> <ul style="list-style-type: none"> - Input information from Control Panel Tab module in the frontend subsystem. <p>Output:</p> <ul style="list-style-type: none"> - Returns the user account information from the database component 	<ul style="list-style-type: none"> - Records the users' operation, such as marking some certain products as favorites or purchased - Interacts with database and update the account or profile data for the particular user according to his/her operations - Generates product recommendation through Machine Learning Technique

2.3.3 Database

This section specifies the input, output and functionality of the database.

Table 9 - Descriptions of database module

Module	Input/Output	Function
Database	<p>Input:</p> <ul style="list-style-type: none">- Data gathered from User Account Management module in backend subsystem <p>Output:</p> <ul style="list-style-type: none">- Retrieves user account data from database, and sends back to the User Account Management module	<p>Data storage:</p> <ul style="list-style-type: none">- User profile: username and password- User preference: favorite products, purchased items and products in Wishlist- User history: images of recent scanned and product names of recent search.

2.4 Assessment of Proposed Solution

Choices of framework, database and server are the critical aspects influencing the performance of a mobile application [22]. After evaluation among each possible solutions and design alternatives, the team decides the proposed solution as the table shows below.

Table 10 - Assessment of proposed solution from four different aspects.

Aspects to be considered	Solution	Strengths	Weaknesses	Trade-offs
Approach of application development	Native mobile app	Native app can support most functionalities on mobile devices, like camera and GPS.	The development cycle for native app is longer. Native app will not be supported by	The mobile app only can be used in one specific platform, but gives more

		It also gives clean user-friendly UI design and high-speed performance [23].	different operating systems.	smooth performance.
Operating system	Android system	The average development time period for Android app are shorter. Android has 75% market share indicating a large amount of users [24].	Due to fragmentation of Android devices and operating system, it costs more to support older version of operating systems or devices.	The app may not support some older version of Android operating system, because of the high implementation cost.
Server	Using third-party service for user account functions	Using third-party service, such as Google, Facebook accounts, as sign-in method, is more secure and low-cost than setting up a local server.	Users need to log into third-party account first, then sign into the app. It is also possible that some users do not have account of any third-party social media.	Although redirecting to a third-party website for social login costs time, it creates a convenient way to link the app and social media [25]. Then users can share contents in the app to other social platforms easily.
Database	Relational database, such as SQLite	In relational database, data are managed in a well-constructed format, such as a table, which is easy to understand and implement [26]. The structure provides relationship between different data tables. Relational database, like SQLite, is serverless, self-contained and lightweight.	It costs more time to manage unstructured data in relational database.	Data, like account information and user preference are relational, which is more logical to store in a relational database, but it may need more time to access data from the tables.

3.0 Work Plan

3.1 Work breakdown Structure and Gantt Chart

The design team will focus on Environment Setup and Tesseract OCR Implementation together for the first part of the project. Then, the team will divide into two groups: front end and back end. These two subgroups will work in parallel for the second part of the project.

3.1.1 Work Breakdown Structure

R = Responsible, A = Assisting

Table 11 - Work Breakdown for each team member

Task #	Task	Zengnan	Jiehao	Menglu	Wei
	Environment Setup				
1	Setup Android Studio workstation	R	A	A	A
2	Purchase Google developer accounts	A	A	A	R
	Tesseract OCR Implementation				
2	Complete tutorials for Android-OCR API integration			A	R
3	Complete OCR demo by scanning simple line of text on an Android device	R	A		
4	Train Tesseract OCR data set	R	A	R	
5	OCR implementation: single-font support			R	
6	Multi-font support: 5 fonts	A		R	
7	Uni-language support: English		R		
8	Multi-language support: English and French		R		A
	Frontend: Home (Search) Tab UI Proposal & Implementation				

9	Dashboard & main page interface	R			
10	Camera interface (OCR)				R
11	Camera interface (ISBN barcode)				R
12	Manual input interface	R			
13	Sending manual input data to backend		R		
14	A drop-down menu to navigate in the application			R	
	Frontend: User Tab UI Proposal & Implementation				
15	Account registration page interface	R			A
	Account log in page interface				R
16	‘User profile’ interface		R	A	
	Fetch and send user account information to the database	A			R
17	‘Favorites’, ‘Purchased’ and ‘Recent Scans’ interface			R	A
18	‘Wishlist’ interface	R			
	Sending customized information to the database.	R			
19	Product review interface		A	R	
20	Product price comparison interface	A		R	A
21	Front end style design	R	A		
	Backend: Web Crawler & Database				
22	Search prices from multiple platforms. (Web Crawling)	A	R		
23	Validate and rank the prices		A		R
24	Fetch customer review from multiple platforms	R			A
25	Validate and rank the reviews		A	R	

26	Implement database for account information management		R	A	A
27	User account registration		R		
28	User account validation & security				R
29	Connect database to frontend GUI: fetch customized information for a specific user.	R	A	A	A
30	Generate product recommendation information for users using database	A	A	A	R

3.1.2 Gantt Chart

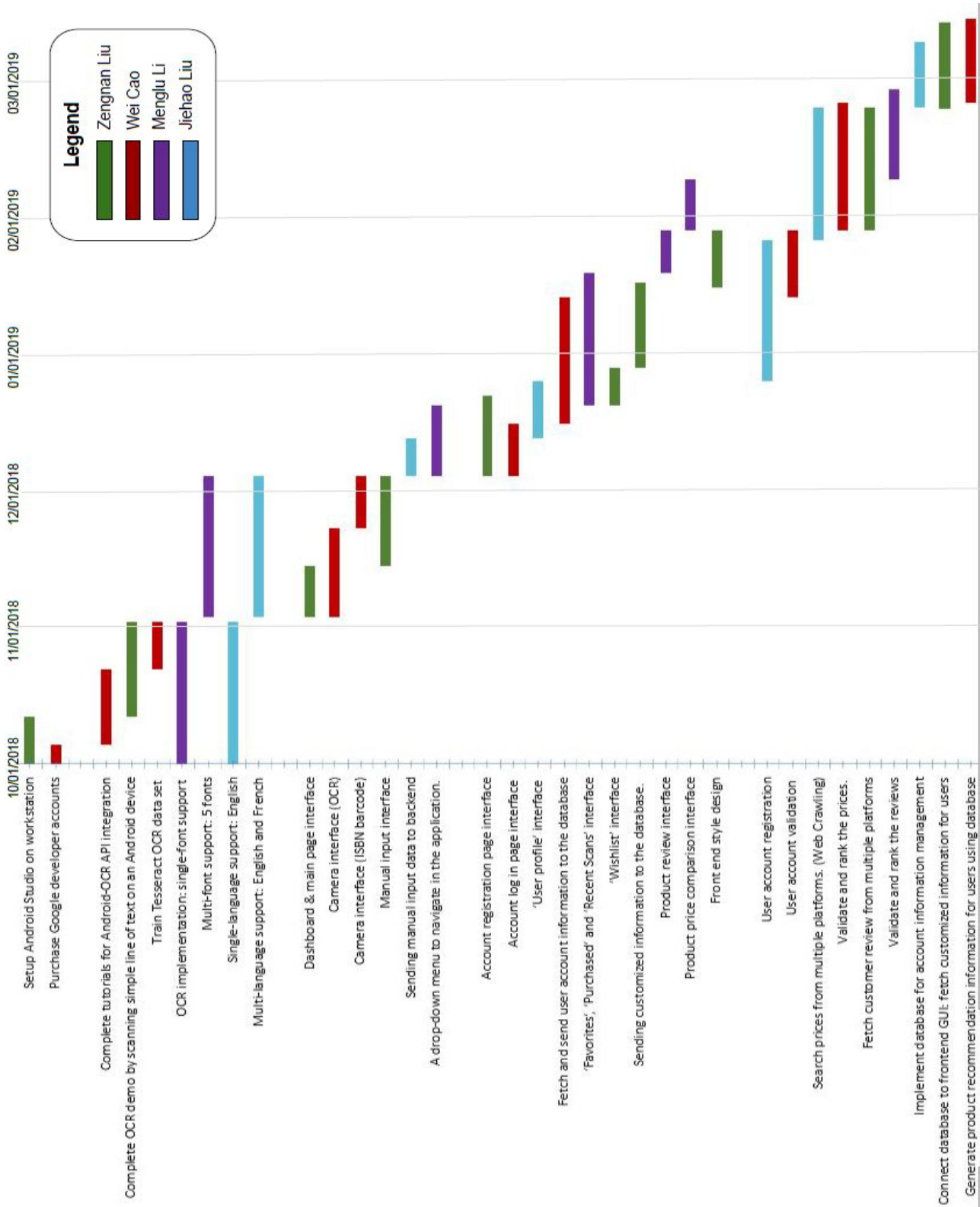


Figure 3 - Gantt Chart

3.2 Financial Plan

This section evaluates the financial cost of our project which consists of labour costs, equipment costs and software development costs.

3.2.1 Student labour cost

Table 12 - Detail of financial costs for labour

Labourer	Cost/unit	Quantity(# of hours)	Total Cost
Wei Cao	\$22.50 [27]	157 [Appendix C]	\$3,532.50
Menglu Li	\$22.50	142	\$3,195.00
Jiehao Liu	\$22.50	152	\$3,420.00
Zengnan Liu	\$22.50	159	\$3,577.50
Total Student Labour(unfunded)			\$13,725.00

3.2.2 Capital Equipment cost and software development cost

Table 13 - Detail of financial costs for capital equipment required for the design

Item	Cost/unit	Quantity	Total Cost	Requires Funding	Kept by students
Laptops owned by students	N/A	4	N/A	No	Yes
Google Developer Account	\$25[28]	4	\$100	Yes	No
Total E&D costs			\$100		

3.2.3 Funding

Table 14 - Detail of the funding resources required for the design

Funding Source	Amount
Students	\$25/each x 4 = \$100
Supervisor	\$0
Request from Design center	\$0
Total funding	\$100

3.2.4 Sum of three tables above

Total cost of project = Student Labour Cost + Equipment cost and software development cost
= \$13,725 + \$100
= \$13,825

Total cost requiring funding = Total funding = \$100

3.2.5 Priority

Our project is entirely software-based, and all development tools are available online for free except for Google Developer account, which is \$25 per person and we are able to fully cover the fees by ourselves.

3.3 Feasibility Assessment (resources, risks)

This section evaluates the feasibility of our proposed project from the following two aspects.

3.3.1 Skill and resources

Table 15 - Assessment of Resources and Skills Required for Project Development

Resources or skills required	Obtained	Plan of acquisition
Programming experience of JAVA language	Yes	Use online learning resources to get familiar with Java programming language [29].
Android mobile application development experience	No	Android provides online tutorials of application development for developers [30].

Database development and management experience	Yes	Members of our team have background knowledge of databases.
Skill of implementing OCR	No	Open source OCR software documents are available online [31].
Hardware equipment for Android mobile application development	Yes	Android mobile application can be developed at a laptop computer or a PC with Windows operating system.
Android mobile application development platform	Yes	Android Studio can be downloaded from official website, an integrated environment for Android application development [32].
Testing equipment for Android mobile application	Yes	An Android smartphone can be used to test mobile application.
Detailed information of each product which user searches	Yes	Electronic commerce companies provide large amount of product information, etc. price and item description, online [33].

3.3.2 Risks

Table 16 - Assessment of Possible Risks for Project Development

Possible risks	Effects of the risk	Plan of reducing risk
OCR fails to recognize the text on the scanned image	Users would not be able to scan tags or book covers, greatly reducing the usability and versatility of the app.	Take manual typing as the primary input method as a back up plan.
Application not running on different versions of Android smartphones due to compatibility issues	Application would freeze or crash when it runs on particular models that it is not compatible with.	Run compatibility debugging tests on the app prototype. Modify the app to support mainstream versions of Android [34].

4.0 Conclusion

The team notices that there is a need in the current market for instant and convenient item price comparison and review retrieval when people are shopping outside, and therefore come up with a mobile app as the solution to fulfill the gap. The app supports multiple input methods including manual input, ISBN barcode scanning and text scanning based on OCR technology to enter item information and retrieves results regarding item prices and reviews from various online-selling platforms. The categories of items that the app aims at are books and electronic devices for the purpose of accuracy. In terms of sufficient background research, the team decides to use Android Studio as the development platform, Java as the main programming language, SQLite as the database, and plans of reducing possible risks have been properly considered. All tasks have been reasonably divided and distributed, and the team has begun to work on them to finish the prototype by the design fair in April 2019.

References

- [1] Hugo Nicolau, Kyle Montague, Tiago Guerreiro, André Rodrigues, and Vicki L. Hanson, “Typing Performance of Blind Users: An Analysis of Touch Behaviors, Learning Effect, and In-Situ Usage,” *ASSETS '15 Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, p. 273-280, October 26, 2015. [Online]. Available: <https://dl.acm.org/citation.cfm?id=2809861>. [Accessed: September 18, 2018].
- [2] John D. Sutter, “Can Braille be faster than QWERTY? App developer thinks so,” *CNN*, February 20, 2012. [Online]. Available: <https://www.microsoft.com/en-us/seeing-ai>. [Accessed: September 18, 2018].
<https://www.cnn.com/2012/02/20/tech/can-braille-be-faster-than-qwerty-app-developer-thinks-so/index.html>. [Accessed: September 18, 2018].
- [3] Microsoft, *Seeing AI*. United States: Microsoft, 2017. [Online]. Available: [4] Trivago, “Find your ideal hotel and compare prices from different websites,” *trivago.ca*. [Online]. Available: <https://www.trivago.ca>. [Accessed: September 18, 2018].
- [5] Vivino, “Buy your next bottle of wine with the help of 32 Million friends,” *Vivino.com*. [Online]. Available: <https://www.vivino.com>. [Accessed: September 18, 2018].
- [6] Tesseract-ocr, “langdata/font_properties,” *GitHub.com*, Feb 21, 2018. [Online]. Available: https://github.com/tesseract-ocr/langdata/blob/master/font_properties. [Accessed: September 18, 2018].
- [7] GoodReads, “GET A COPY OF THE FIRST DAYS,” *GoodReads*. [Online]. Available: https://www.goodreads.com/book/9648068/get_a_copy. [Accessed: September 18, 2018].
- [8] Google Plays, “Personal and Sensitive Information,” *Google*. [Online]. Available:

<https://play.google.com/about/privacy-security-deception/personal-sensitive/>. [Accessed: October 3, 2018].

[9] Google Plays, “Webviews and Affiliate Spam,” *Google*. [Online]. Available: <https://play.google.com/about/spam-min-functionality/spam/webviews-affiliate/>. [Accessed: October 3, 2018].

[10] Google Plays, “Permissions,” *Google*. [Online]. Available: <https://play.google.com/about/privacy-security-deception/permissions/>. [Accessed: October 3, 2018].

[11] Kinjal Dua, “A Guide to Mobile App Development: Web vs. Native vs. Hybrid,” *Clear Bridge Mobile*, March 7, 2018. [Online]. Available: <https://clearbridgemobile.com/mobile-app-development-native-vs-web-vs-hybrid>. [Accessed: September 18, 2018].

[12] Priya Viswanathan, “The Pros and Cons of Native Apps and Mobile Web Apps,” *Lifewire*, April 19, 2017. [Online]. Available: <https://www.lifewire.com/pros-and-cons-of-native-apps-and-mobile-web-apps-2373173>. [Accessed: September 18, 2018].

[13] Priya Viswanathan, “Native Apps vs. Web Apps: What is the Better Choice?” *Lifewire*, April 30, 2018. [Online]. Available: <https://www.lifewire.com/native-apps-vs-web-apps-2373133>. [Accessed: September 18, 2018].

[14] Volodymyr Bilyk, “IOS VS ANDROID DEVELOPMENT,” *The App Solutions*. [Online]. Available: <https://theappsolutions.com/blog/development/ios-vs-android>. [Accessed: September 18, 2018].

- [15] Shrikant Srivastava, “Benefits of iOS vs. Android App Development: {A 2018 Lookout},” *Appinventiv Technologies*, December 22, 2017. [Online]. Available: <https://appinventiv.com/blog/ios-vs-android-app-development-a-2018-lookout>. [Accessed: September 18, 2018].
- [16] Adam Sinicki, “Developing for Android vs developing for iOS – in 5 rounds,” *Android Authority*, June 9, 2016. [Online]. Available: <https://www.androidauthority.com/developing-for-android-vs-ios-697304>. [Accessed: September 18, 2018].
- [17] Christopher Eri, “How to Pick a Dedicated Server for Your App,” *Inmotion Hosting*, January 19, 2018. [Online]. Available: <https://www.inmotionhosting.com/support/team/how-to-pick-a-dedicated-server-for-your-app>. [Accessed: September 18, 2018].
- [18] Google Developers, “Integrating Google Sign-In into your web app,” *Google*, March 15, 2018. [Online]. Available: <https://developers.google.com/identity/sign-in/web/sign-in>. [Accessed: September 18, 2018].
- [19] Priit Kallas, “Top 15 Most Popular Social Networking Sites and Apps,” *Dreamgrow*, August 2, 2018. [Online]. Available: <https://www.dreamgrow.com/top-15-most-popular-social-networking-sites>. [Accessed: September 18, 2018].
- [20] James Serra, “Relational databases vs Non-relational databases,” August 27, 2015. [Online]. Available: <https://www.jamesserra.com/archive/2015/08/relational-databases-vs-non-relational-databases>. [Accessed: September 18, 2018].

- [21] Jeff Hu, “Relational versus Non Relational Database An Easy Table For Better Understanding,” *Hacker Noon*, November 21, 2017. [Online]. Available: <https://hackernoon.com/relational-versus-non-relational-database-d5d1c439fb86>. [Accessed: September 18, 2018].
- [22] Shormistha Chatterjeem, “What are the main technical aspects of a mobile app?” *Quora*, August 30, 2018. [Online]. Available: <https://www.quora.com/What-are-the-main-technical-aspects-of-a-mobile-app>. [Accessed: October 3, 2018].
- [23] KATARZYNA LORENC, “KEY PROS AND CONS OF NATIVE AND HYBRID APPS,” *ready4s.com*. [Online]. Available: <https://www.ready4s.com/blog/key-pros-and-cons-of-native-and-hybrid-apps/>. [Accessed: October 3, 2018].
- [24] Daria R. “iOS vs Android Development: Which One is Best for Your App?” *RubyGarage.org*. [Online]. Available: <https://rubygarage.org/blog/ios-vs-android-development>. [Accessed: October 3, 2018].
- [25] Lonneke Spinhof, “10 Advantages and disadvantages of social logins,” *humix.be*, July 16, 2015. [Online]. Available: <https://www.humix.be/blog/10-advantages-disadvantages-social-logins/>. [Accessed: October 3, 2018].
- [26] Craig Borysowich, “Some Pros & Cons of Relational Databases,” *Toolbox.com*, 2008. [Online]. Available: <https://it.toolbox.com/blogs/craigborysowich/some-pros-cons-of-relational-databases-050108>. [Accessed: October 3, 2018].
- [27] PayScale, “Entry-Level Computer Programmer in Toronto, Ontario Salary,” *payscale.com*. [Online]. Available:

https://www.payscale.com/research/CA/Job=Computer_Programmer/Salary/ff38fbe1/Entry-Level-Toronto-ON. [Accessed: October 3, 2018].

[28] Appy Pie, “How much does a Google/Apple/Windows Developer Account cost?” *appypie.com*. [Online]. Available: <https://www.appypie.com/faqs/how-much-does-a-googleapple-developer-account-cost>. [Accessed: October 3, 2018].

[29] TutorialsPoint, “Java Tutorial,” *tutorialspoint.com*. [Online]. Available: <https://www.tutorialspoint.com/java/>. [Accessed: September 18, 2018].

[30] Android Developers, “Build your first app,” *developer.android.com*, April 22, 2018. [Online]. Available: <https://developer.android.com/training/basics/firstapp>. [Accessed: September 18, 2018].

[31] Tesseract-ocr, “Tesseract Home,” *GitHub.com*, September 12, 2018. [Online]. Available: <https://github.com/tesseract-ocr/tesseract/wiki>. [Accessed: September 18, 2018].

[32] Android Developers, “Meet Android Studio,” *developer.android.com*, June 5, 2018. [Online]. Available: <https://developer.android.com/studio/intro>. [Accessed: September 18, 2018].

[33] Amazon.ca. [Online]. Available: <https://www.amazon.ca>. [Accessed: September 18, 2018].

[34] Fossbytes, “Most Popular Android Versions In September 2018 (Always Updated List),” *fossbytes.com*, September 3, 2018. [Online]. Available: <https://fossbytes.com/most-popular-android-versions-always-updated>. [Accessed: September 18, 2018].

Appendices

Appendix A: Student-supervisor agreement form

ECE496 Design Project

Student – Supervisor Agreement

Our signatures below indicate that we have read and understood the following agreement, and that all parties will do their best to live up to the word as well as the spirit of it.

We agree to meet at least once every two weeks for at least half an hour to discuss progress, plans, and problems that have arisen. Before each meeting, the group will prepare a brief progress report that will form the basis for the discussions at the meeting.

If a meeting has to be cancelled by the supervisor, she/he should advise the group as early as possible. If a student cannot attend a meeting, she/he should advise members of the group as well as the supervisor as early as possible.

Both the supervisor and the students will:

- Inform themselves of the course expectations and grading procedure.

The supervisor will:

- Provide regular guidance, mentoring, and support for his/her design project group(s),
- Take an active role in evaluating the work and performance of the students' by completing the supervisor's portion of the grading forms for each course deliverable expediently.
- Return a photocopy of the completed grading evaluation forms to the appropriate section administrator in a timely fashion.
- Be aware of the aims and processes of the course as outlined in the Supervisor's Almanac.

We have read and understood this agreement. Date: Oct 25, 2018

Signature of supervisor: _____

Signature of student: _____

Signature of student: Jichao Liu

Signature of student: _____

Signature of student: _____

Last revision: 7/08

Team 2018959

Appendix B: Report Attribution Table

Project Proposal Document Attribution Table

Section	Student Initials			
	1.WC	2.ML	3.JL	4.ZL
Executive Summary	ET	ET	ET	RD,RS,MR
Background and Motivation	ET	ET	ET	RD,RS,MR
Project Goal	RD,RS,MR	RD,RS,MR	RD,RS,MR	RD,RS,MR
Validation and Acceptance Tests	MR	ET	ET	RD,RS
Validation and Acceptance Test Summary	RD,RS	ET	ET	MR
Possible Solutions and Design Alternatives	RD,RS	RD,RS	MR	RD,RS
System-level Overview	RD,MR	RD,MR	ET	ET
Module-level Descriptions	RD,MR	RD,MR	ET	ET
Assessment of Proposed Solution	ET	RD,RS,MR	ET	MR
Work breakdown Structure and Gantt Chart	RD,RS,MR	RD,RS,MR	RD,RS,MR	RD,RS,MR
Financial Plan	ET	ET	RD,RS,MR	ET
Feasibility Assessment (resources, risks)	MR	RD,RS,MR	RD,RS,MR	MR
Conclusion	ET	ET	RD,RS,MR	ET
All	CM	CM	FP,OR	FP

Abbreviation Codes:

Fill in abbreviations for roles for each of the required content elements. You do not have to fill in every cell. The "All" row refers to the complete document and should indicate who was responsible for the final compilation and final read through of the completed document.

RS – responsible for research of information

RD – wrote the first draft

MR – responsible for major revision

ET – edited for grammar, spelling, and expression

OR – other

"All" row abbreviations:


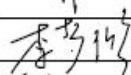
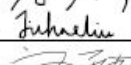

FP – final read through of complete document for flow and consistency

CM – responsible for compiling the elements into the complete document

OR – organizing references and appendices

Signatures

By signing below, you verify that you have read the attribution table and agree that it accurately reflects your contribution to this document.

Name	Wei Cao (WC)	Signature		Date:	2018.10.24
Name	Menglu Li (ML)	Signature		Date:	2018.10.24
Name	Jiehao Liu (JL)	Signature		Date:	2018.10.24
Name	Zengnan Liu (ZL)	Signature		Date:	2018.10.24

Appendix C: Team Work Hours Distribution per Task

Task Name	Work Hours Required	Team Member
Environment Setup		
Setup Android Studio on workstation	10	ZL
Purchase Google developer accounts	4	WC
Tesseract OCR Implementation		
Complete tutorials for Android-OCR API integration	16	WC
Complete OCR demo by scanning simple line of text on an Android device	20	ZL
Train Tesseract OCR data set	10	WC
OCR implementation: single-font support	30	ML
Multi-font support: 5 fonts	30	ML
Single-language support: English	30	JL
Multi-language support: English and French	30	JL
Frontend: Home (Search) UI Proposal & Implementation		
Dashboard & main page interface	11	ZL
Camera interface (OCR)	19	WC
Camera interface (ISBN barcode)	11	WC
Manual input interface	19	ZL
Sending manual input data to backend	8	JL
A drop-down menu to navigate in the application.	15	ML
Frontend: User Tab UI Proposal & Implementation		
Account registration page interface	17	ZL
Account log in page interface	11	WC
'User profile' interface	12	JL
Fetch and send user account information to the database	27	WC
'Favorites', 'Purchased' and 'Recent Scans' interface	28	ML
'Wishlist' interface	8	ZL

Sending customized information to the database.	18	ZL
Product review interface	9	ML
Product price comparison interface	11	ML
Front end style design	12	ZL
Backend: Web Crawler & Database		
User account registration	30	JL
User account validation	14	WC
Search prices from multiple platforms. (Web Crawling)	28	JL
Validate and rank the prices.	27	WC
Fetch customer review from multiple platforms	26	ZL
Validate and rank the reviews	19	ML
Implement database for account information management	14	JL
Connect database to frontend GUI: fetch customized information for users	18	ZL
Generate product recommendation information for users using database	18	WC