Project Planning
For
HofSwap

Prepared by:
1. Parhum Ebrahimian
2. Frank Martin
3. Aishik Mallick
4. Trey Jean-Baptiste
5. Joseph Gentile

CSC190 Project
Spring 2020

Table of Contents

Edits After the Fact

After having gone through most of the development for the app, we are adding this edit to update you on what has changed. The first change we had to make was to drop the choice of making a duel platform app. Originally our app was supposed to run on Android and iOS, but due to unforeseen circumstances we have had virtually no time to meet and work out certain development problems. Due to this we chose to stick strictly with iOS. We were also supposed to use the Amazon Affiliate program in order to have valid ISBNs and textbook images on the application. Unfortunately, Amazon's current policies do not allow the public to easily access their API at the moment. Although we did explore other APIs, many had other stipulations, like requiring payment, not working with our development platform, or not fitting with how our application worked. Because of this, we could not get the pictures of the textbook easily, so we opted to drop this feature in order to make our due date which gave us the capability to finish. Another change we made was the removal of the dedicated Detailed View page since the removal of images allowed us to display more information on the Search View listings. Since the extra page with the same information would have been redundant, we rolled the features into the Search View listings. This cut down the development time of T7.

Intro

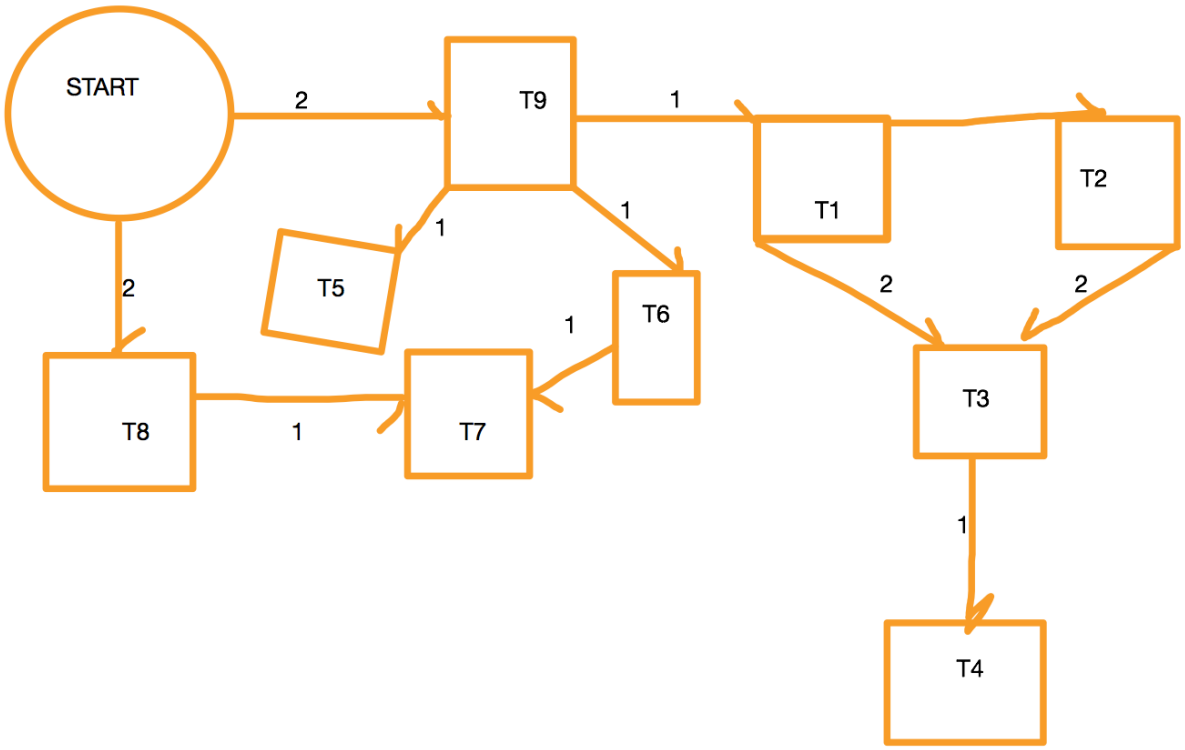
Our app is meant to be used by Hofstra Students. Ideally, every account will be able to buy and sell textbooks. When a buyer is ready to purchase, they will hit a button which will then reach out to a seller in order to make sure the seller is ready to sell and the book is still intact. Our app is NOT a marketplace, but more like an exchange platform for users to get in contact to

each other and work out pricing and other details. Below is our Task Chart and how we plan on finishing the app. Since we will be developing on iOS and Android OS, our team will be taking the tasks one by one, but split into two teams; one developing for iPhone, the other for Android.

Requirements Traceability

#	TASK	Requirements Point	DURATION (weeks)	DEPENDENCIES
T1	Login Page (Authentication)	1.1	1	T9
T2	Sign up View	1.2	1	T1
T3	Profile Page	2, 2.1, 2.2	2	T1, T2
T4	Notification	6	1	T3
T5	Upload page	3	1	T9
T6	Search page	4, 4.1, 4.2,	1	T9
T7	Detailed View	5, 5.1, 5.2, 6	1	T8, T6
T8	Amazon functions	8	2	-
T9	Firebase implementation	7	2	-

CRITICAL PATH: T9, T1, T3, T4



Risk level	**Tasks**
High Risk	T9, T8,
Medium Risk	T6, T7, T3,
Low Risk	T5, T4, T2, T1

T9 Earliest Start: 0 Earliest Finish: 2 Latest Start: 0 Latest Finish: 2 Slack: 0	T1 Earliest Start: 2 Earliest Finish: 3 Latest Start: 2 Latest Finish: 3 Slack: 0
T2 Earliest Start: 3 Earliest Finish: 4 Latest Start: 3 Latest Finish: 4 Slack: 0	T3 Earliest Start: 4 Earliest Finish: 6 Latest Start: 4 Latest Finish: 6 Slack: 0
T4 Earliest Start: 6 Earliest Finish: 7 Latest Start: 6 Latest Finish: 7 Slack: 0	T8 Earliest Start: 0 Earliest Finish: 2 Latest Start: 4 Latest Finish: 6 Slack: 2
T7 Earliest Start: 3 Earliest Finish: 4 Latest Start: 6 Latest Finish: 7 Slack: 3	T5 Earliest Start: 2 Earliest Finish: 3 Latest Start: 6 Latest Finish: 7 Slack: 4
T6 Earliest Start: 2 Earliest Finish: 3 Latest Start: 5 Latest Finish: 6 Slack: 3	

TESTING/ Quality Assurance

1. Development Testing
    a. Each person will test their specific module
    b. Unit Testing
        i. Login Page-
        ii. Signup View-
            1. Create a new user; Check if log-in is successful using dummy data, access database to see if new user is there
        iii. Notification page-
            1. send seller message as a buyer to see if notifications show up
        iv. Upload Page-
            1. Upload textbooks using the database and check if it is there
        v. Search page
            1. Search for multiple books with the location known to us
        vi. Detailed View
            1. Check for the status of the book and its condition with our own data
        vii. Amazon and Firebase
            1. Check API calls and its compatibility with what code we have
    c. System Testing
        i. Once every component is working individually we will link everything together, it's important to see the system working together.
    d. Release Testing
        i. Make sure entire system is working flawlessly before release

SCHEDULE

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9
P	T9	T9	T2	T3	T3	T4	TEST	TEST	TEST
A	T9,T8	T9,T8	T6	T5	T3	T4	TEST	TEST	TEST
J	T8	T8	T2,T6	T3	T3	T4	TEST	TEST	TEST
F	T8	T1	T2	T5	T7	T4	TEST	TEST	TEST
T	T9	T1	T6	T3	T7	T4	TEST	TEST	TEST

	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9
TESTING	-	T9,T8	T1	T2	T6,T2	T3,T5	T4	SYSTEM	SYSTEM

CONFIGURATION MANAGEMENT

A. Version management:

   i. GitHub platform: will be following a GitHub branch hierarchy containing the following branches:

   ii. Consumer ready branch: main version ready for the consumer to use.

   iii. Updated branch: branch containing new updated versions of the software.

   iv. Creation branch: branch reserved for the sole purpose of development.

- Will be utilizing pull requests to merge branches

B. System management: Will be using Amazon firebase to assemble libraries, data, and program components.

C. C. Change management: any changes made will be recorded on Github. As a second alternative we will be listing all changes made on a separate file.