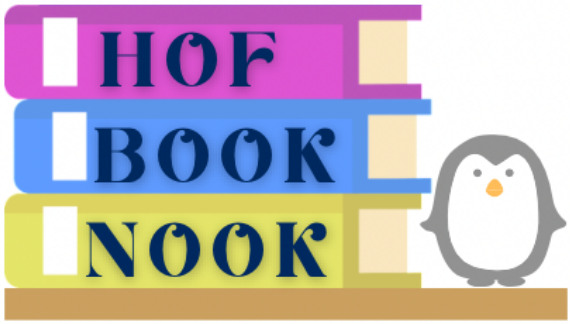
The Hof Book Nook

Project Planning



Version: 2.0

Date: 11/29/22

Team:

Brian Baptista, Christopher Rios, Michael Salomone, Jasmin Varela

Table of Contents

I. Introduction

The purpose of this document is to give us some structure in planning the creation of the Hof Book Nook. It details how we are splitting up the requirements into tasks, their length, and how we will allocate resources to each task. Additionally, this document describes the risks we have for each task and the mitigation strategies we have in case we run into problems. This project plan will help us accomplish our assignment before the deadline of December 1st, 2022. Flutter will be used to create the Hof Book Nook app and Firebase will be used for the database. Github will be used in order to manage all of our code. We will also be applying the tasks and deadlines from this project plan into Jira to keep track of our progress throughout the semester.

II. Configuration Management Strategy

We will be using Github in order to manage all the functions for the Hof Book Nook. The link for the original Github Repository is https://github.com/crios586/hof_book_nook. The Github with with our final project is https://github.com/JVarela02/the_hof_book_nook.

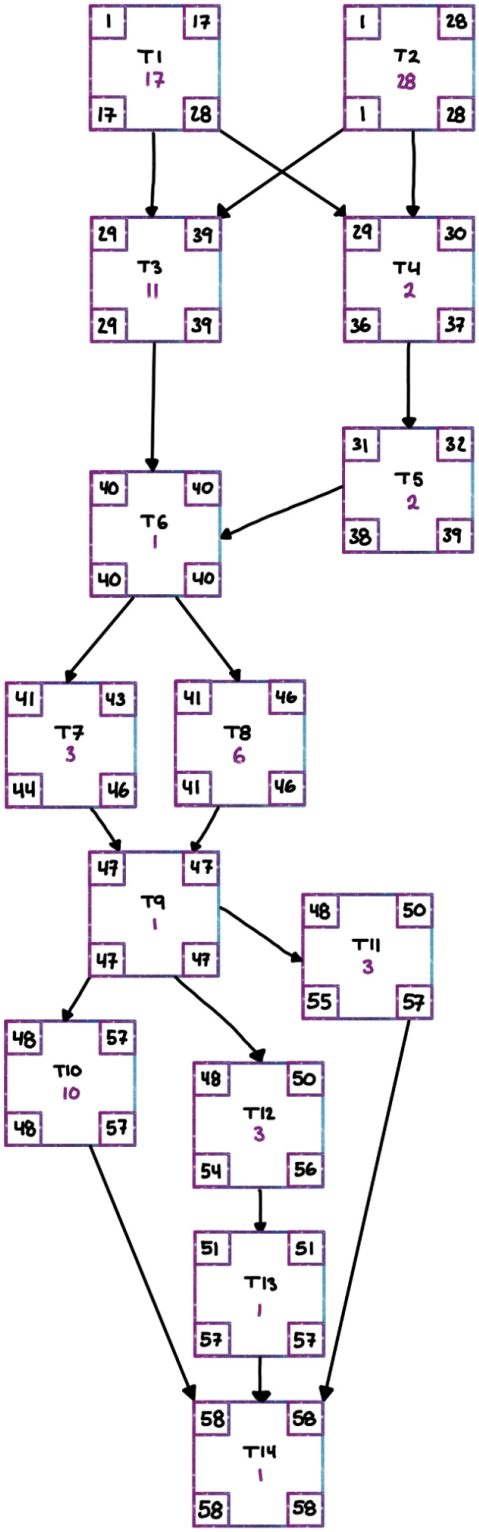
III. Hof Book Nook Team

MEMBER	GITHUB ACCOUNT	EMAIL
Brian Baptista	https://github.com/BrianBaptista	bbaptista1@pride.hofstra.edu
Christopher Rios	https://github.com/crios586	crios1@pride.hofstra.edu
Jasmin Varela	https://github.com/JVarela02	jvarela1@pride.hofstra.edu
Michael Salomone	https://github.com/mikes01	msalomone1@pride.hofstra.edu

IV. Work Schedule

T#	TASK	DURATION	DEPENDENCY	RESOURCES	RISK EVALUATION
T1	Create & connect Database	17 Days	N/A	Brian & Mike	5
T2	Create GUI	28 Days	N/A	Chris & Jasmin	3
T3	Create functions for Sign-Up and Log-in	11 Days	T1 & T2	Jasmin & Chris	2
T4	Create function for Add Books	2 Days	T1 & T2	Jasmin	1
T5	Create Function to Remove Books	2 Days	T4	Brian & Mike	1
T6	Implement Functions to GUI (I) / Testing	1 Day	T5 & T3	Brian, Chris, Jasmin, and Mike	1
T7	Create My Listings Display Function	3 Days	T6	Brian & Jasmin	2
T8	Create Search Function	6 Days	T6	Jasmin	4
T9	Implement Functions to GUI (II) / Testing	1 Day	T7 & T8	Brian, Chris, Jasmin, and Mike	1
T10	Create Notify Seller Function	10 Days	T9	Brian	4
T11	Create Mark “In Negotiations” Function	3 Days	T9	Brian & Mike	1
T12	Connect API	3 Days	T9	Jasmin	4
T13	Create Textbook Display Function	2 Days	T12	Brian & Jasmin	2
T14	Final Testing / Quality Assurance	1 Days	T10, T11, T12, T14	Brian, Chris, Jasmin, and Mike	1

V. Activity Graph & Gantt Chart:



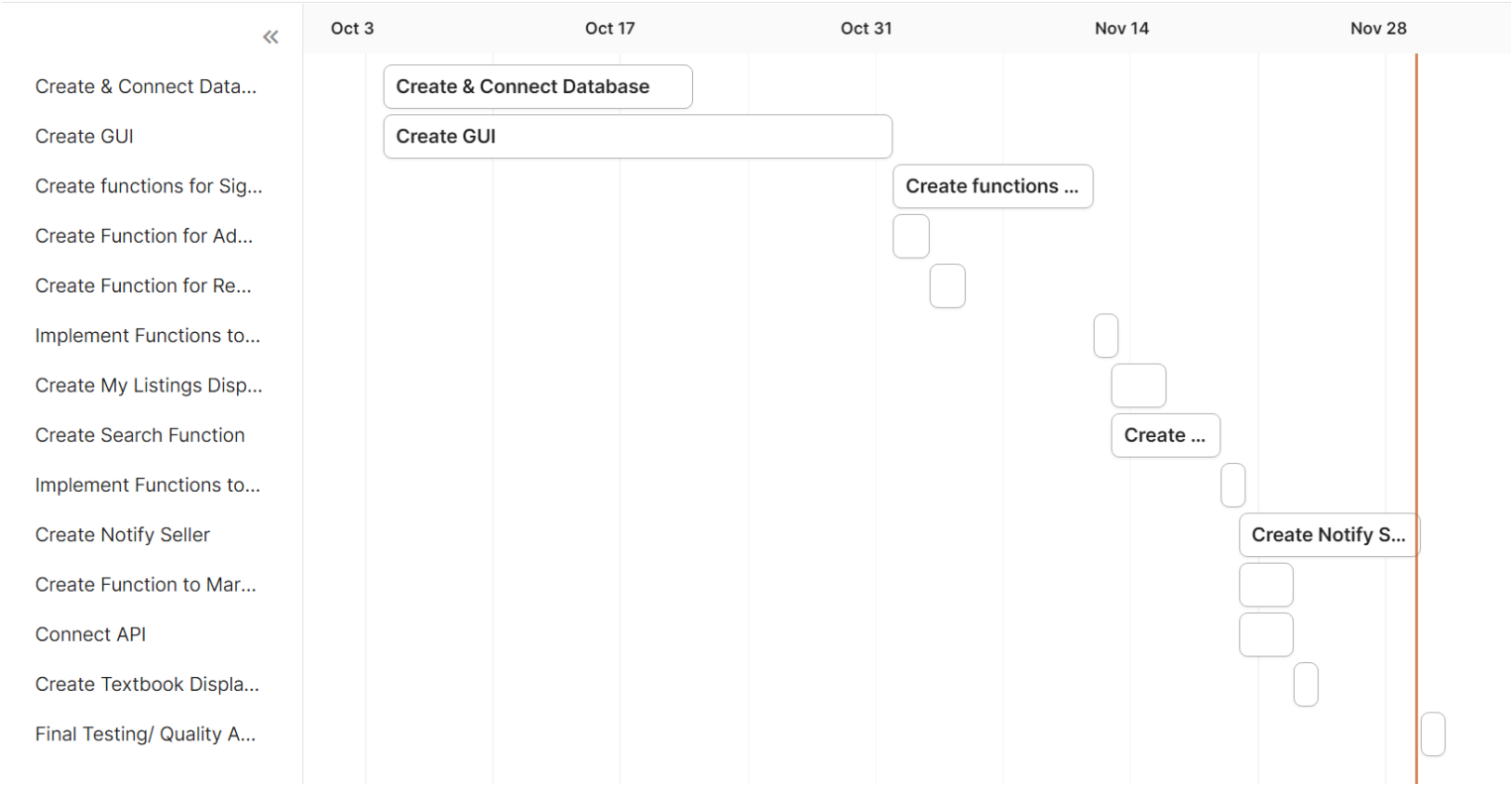
In Version 1.0 of our project plan we had multiple paths that were critical. In Version 2.0 we only have one critical path which goes from …

- T2 → T3 → T6 → T8 → T9 → T10 → T14

This critical path takes into account creating the GUI, the sign-in/sign-up function, the search function, the notify seller function, and all of our testing and implementations.

Our completion date for this project remained the same, December 1st, 2022.

As with Version 1.0, we still only had a slack of 3 days for our project.



VI. Requirements Specification Traceability

Task	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
Traceability	IV. A IV. B	VI	IV. A	IV. B	IV. B	N/A	IV. B	IV. C	N/A	IV. D V. D	IV. B	IV. B	IV.C	N/A

VII. Risk Classification

The Risk Classification is scored on a scale from 1 to 5. It estimates the relative level of risk that a given activity has at jeopardizing the successful and timely completion of the project. The factors that contribute to the risk classification score are: the complexity of the activity and the likelihood of encountering an issue during the activity that will require significant time to remedy.

A. High (5)

- Will absolutely derail the project if this kind of risk is realized. Having a contingency plan is a must. Ideally, avoid its realization at all costs.

B. Medium (3-4)

- Will cause a significant setback if this kind of risk is realized. The project can still recover if a risk of this kind is realized but equally so it puts the project at significant risk.

C. Low (1-2)

- Is a minimal level of risk. If this kind of risk is realized, it has a low chance of jeopardizing the project. But, many little problems can become one big problem. So stay vigilant.

VIII. Risk Mitigation:

1. Hardcode in a temporary username and password apart from the database to allow the website to flow for testing in case the team runs into database troubles. This will also help with detecting any bugs to fix early on in the development cycle.

2. Test software through regression testing after every new implementation of software to ensure the software still runs and does not crash. This will also help detect any bugs early on and reduce having to dig through source code to figure out what is causing the bug. If the team can identify bugs during testing, it is highly likely the bug occurs inside the most recent software update which will lead to higher turn around times for fixing bugs as the team will spend less time searching through older source code to find where the bug is occurring.

3. If bugs are constantly found at an alarming rate that can cause a bottleneck to development, the team should implement a developer log to help see how the code is interacting with one another. For example printing statements of functions being called, entered and exited. If the log fails to reach the end of a function before crashing it is highly likely that the broken code will reside inside the code block of that function. Sometimes buttons will call multiple functions at a time, which may call other functions in files as well. This will cause a hard time to find the bug if a log file is not implemented as the bug will not be as obvious to spot. Although a developer log is not needed, it may be a useful tool if the team is spending more time trying to fix bugs than expected.

4. The tasks have been designed to mitigate risks as much as possible. Although there are dependencies that must be met, most tasks can be worked on as long as the previous page has some functionality to it. For example, as long as the home page can direct to other pages such as “My Account” and “My Listings”, the development team can continue to work on these pages independently from another developer working on the functionality of the “Home Page”. This will help improve efficiency as the team will be working on multiple parts of the project at a time, rather than experiencing a bottleneck waiting for a page to be completely implemented before moving on to the next page. Additionally, many of the tasks have some leniency time included. This is if we finish a task early or if another task needs more resources we can work with some flexibility.

IX. Deviation from Requirements :

There are a few items that we will not be implementing that were stated in our original systems requirements sheet that were not included in the project plan. The three items that have been placed on the backburner are …

- ➢ Customizing / Editing Account Profile
    - In this sub category adding and changing profile pictures will also not be included
- ➢ Displaying list of Recently Viewed Textbooks

The reasoning for why we aren't implementing these functions is because of lack of time. Since we want to ensure that the main requirements of the app are functioning we want to have multiple testing phases through different implementations. As seen in the work schedule and activity graph we have split the function implementation into three groups each having 2-3 functions. In order to do this we had to sacrifice the two functions listed above.

Customizing and editing their profile was added to the functions in order to give users the opportunity to add their own touch to their account profile. But since the app relies primarily on their initial create account function eliminating this extra component will not affect the app.

Displaying list of recently viewed books was also added with the intention of providing more accessibility for users. With this function not being present, users will simply have to look for the textbook in the database to see it again.

If any functions that are currently in the work schedule take less days to complete then estimated then an effort will be made to incorporate these functions into the system.

X. Testing :

After each new piece of code is implemented into the build, the developer should test their work independently to see if there are any bugs in their code that they can quickly identify and fix. When they are satisfied with their work, the rest of the team will do tests on the software to validate the changes and give any feedback if needed. A set of tasks will be followed to test

the software to make sure the team tests full functionality of the build. After the test has been followed and feedback has been given, the task will either be completed or sent back for further revisions. A sample testing checklist is provided below:

Task (independent from dependency chart tasks)	Description	Feedback
Test 1	Click Sign-up page	Sign up page not popping up
Test 2	Create an account with your credentials. i.e username: John Smith 700 #: 70000000 or h700… Email: Jsmith@pride.hofstra.edu Password: Penguin	N/A (Working)
Test 3	Plug in login credentials for John Smith	N/A
Test 4	Navigate to “My Account Page” and edit profile	N/A
Test 5	Change Password and Press Logout Function	N/A
Test 6	Login with new credentials for John Smith	New Password not working, old password still saved in database for john smith
Test 7	Navigate to “Add Listing” and create a new listing for a book.	N/A
Test 8	Ensure that listing has been added to “My Listing Page” and is available through search	Listing Not appearing on “My Listings” page
Test 9	On the listing, press notify seller and ensure an email is	N/A

	sent to your hofstra email account	
Test 10	Click “Search” and search for the book listed in Test 7.	N/A
Test 11	Navigate to “My Account” and press “Remove Listing”	N/A
Test 12	Remove listing of the book listed in Test 7.	Removed from search list but still appearing on “My Listing”
Test 13	Navigate to “My Account” and press “Sign out” ensure that you cannot gain access to “Search” or “My Listings” from login screen	N/A

The goal is to have no feedback in the feedback column for the final demo. Using a system like this ensures that every aspect of the code is accounted for and will be helpful to identify bugs and fix issues before they become major problems in the development cycle. As the development life cycle nears its end, all tests can be run. Since these tests will be tested after every new implementation of the code, it will become easier to solidify older code as quality production code. Testing everytime new code is implemented will in turn take less time and turn up fewer bugs rather than doing one large test before the demonstration with the customer team. Furthermore, our usage of automated unit and integration testings on the source code repository will greatly simplify this aspect of the project.