

**Computer Science and Engineering**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project Webucation**

**Project Management Plan**

**Version 2.0**

Document Number: SPMP-002

Project Team Number: B40

Project Team Members: Ivan Lin il2149

Calvin Chu cc6923

Sabahat Sami ss14243

Tanvi Rahman tr2128

**REVIEW AND APPROVALS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Members** | **Function** | **Date** | **Signature** |
| Ivan Lin | Author | 10/17/23 | Ivan Lin |
| Calvin Chu | Author | 10/17/23 | Calvin Chu |
| Sabahat Sami | Author | 10/17/23 | Sabahat Sami |
| Tanvi Rahman | Author, Poster | 10/17/23 | Tanvi Rahman |
|  |  |  |  |
|  |  |  |  |

**REVISION LEVEL**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision Number** | **Purpose** |
| 4/6/23 | Version 1.0 | Initial Release |
| 10/17/23 | Version 2.0 | Updated Release |
|  |  |  |
|  |  |  |
|  |  |  |

**TABLE OF CONTENTS**

[**1. OVERVIEW 5**](#_upt1n9gtq4kq)

[1.1 Project Summary 5](#_7nc8ysdlvgv7)

[1.2 Purpose, Scope, and Objectives 5](#_pikmwlj4aehu)

[1.3 Assumptions and Constraints 5](#_x7g01yd4ok11)

[1.4 Project Deliverables 6](#_b28m75k1b5un)

[1.5 Schedule and Budget Summary 6](#_mo1ycpspm5uu)

[1.6 Evolution of the Plan 6](#_db3a1ijgjzym)

[**2. REFERENCES 7**](#_2eydah3c2j2h)

[**3. DEFINITIONS 7**](#_ihz6jdmlvu59)

[**4. PROJECT ORGANIZATION 7**](#_31v2u9mwre0)

[4.1 External Interfaces 7](#_u6glw6r5tx8j)

[4.2 Internal Structure 7](#_gxt1bsshgn8y)

[4.3 Roles and Responsibilities 8](#_ijjh9hsnch72)

[**5. MANAGEMENT PROCESSES 8**](#_sk6awrivoq9o)

[5.1 Start-Up Plan 8](#_7bsln3st6cco)

[5.1.1 Estimation Plan 8](#_ooo4h340tw3f)

[5.1.2 Staffing Plan 8](#_pfxkffdc8v22)

[5.1.3 Resource Acquisition Plan 8](#_1zb98bq9h8th)

[5.1.4 Training Plan 9](#_lj7rf3cyo0zr)

[5.2 Work Plan 9](#_4k0dqzyhhp9h)

[5.2.1 Work Activities 9](#_kqanjt25u4ic)

[5.2.2 Schedule Allocation 9](#_1gu4vtftbrna)

[5.2.3 Resource Allocation 9](#_wxfuq3nuhoi)

[5.2.4 Budget Allocation 9](#_fdo7oeq5kxfk)

[5.3 Control Plan 9](#_n1ju77nun0q4)

[5.3.1 Requirement Control and Traceability 9](#_nd11jxo8t2sc)

[5.3.2 Schedule Tracking and Adjustment 10](#_jx36hin3pbfj)

[5.3.3 Budget Tracking and Adjustment 10](#_1meibhtjo9nl)

[5.3.4 Quality Control 10](#_yugd2z28meg9)

[5.3.5 Reporting Mechanisms 10](#_po8pccq0qgna)

[5.3.6 Metrics Collection Plan 10](#_krxurr3b0zsm)

[5.4 Risk Management Plan 10](#_ts70m9inao47)

[5.5 Post Implementation Plan 11](#_1xwkbvqu8wmv)

[**6. TECHNICAL PROCESSES 12**](#_kx9plwycbckm)

[6.1 Process Model 12](#_gpcsnffb93zi)

[6.2 Methods, Tools, and Techniques 12](#_wg6s2n9kgmyh)

[6.3 Infrastructure Plan 12](#_ialjq9cc9ly8)

[6.4 Product Acceptance and Migration Plan 12](#_16vns77gjmi)

[**7. SUPPORTING PROCESSES PLANS 12**](#_ddop7kqj4bub)

[7.1 Configuration Management Plan 12](#_wqm9rfpb91bi)

[7.2 Qualification (Verification and Validation) Plan 13](#_y6pis0tzlyy9)

[7.3 Documentation (library) Plan 13](#_1lmk3e6gifhm)

[7.4 Quality Assurance Plan 13](#_sjvyc7vvo9gu)

[7.5 Reviews and Audits 13](#_c6x2j55ippxq)

[7.6 Problem Resolution Plans 13](#_kfki2012trz6)

[7.7 Environment Management Plans 14](#_lmyukrhfa5xo)

[7.8 Process Improvement Plan 14](#_jubxwyvm2c9w)

[**8. ADDITIONAL PLANS 14**](#_rhz258n42zdt)

[**9. INDEX 14**](#_47mnlha8f4dn)

[**10. RATIONALE 14**](#_eqgimc6zyqsj)

[**11. NOTES 15**](#_hwtgiqamk40)

[**12. APPENDICES 15**](#_to4fewqqkkqm)

[12.1 Schedule Tracking 15](#_y15zwl1kbubf)

[12.2 Defect Tracking 16](#_37ijtsmlbh3h)

[12.3 Gantt Chart 18](#_xvcvsa4g5l3x)

## 1. OVERVIEW

### 1.1 Project Summary

The motivation for this project is to provide an application that improves the student experience in studying and preparing for exams and other school assignments. The purpose of the SPMP is to provide an outline of the processes and project management organization. The target audience of this document is the software development team, software quality assurance team, project management team, clients, and any interested parties.

### 1.2 Purpose, Scope, and Objectives

The purpose of this document is to cover the organization of this project, the processes and technologies used for this project, and an outline for the overall project management. The target audience of this document will be the client, software quality group, development team, project management team, and anyone else who is interested in reading this document. The client in this case would be the New York University (NYU) student body.

This project will be a web application that will allow students to organize and share exam preparation materials with friends and classmates. A user will be able to consolidate all of his notes and materials for each class into this single application and share those documents with other users on the app. Users will be able to customize a profile to connect with other students of similar backgrounds and enrolled courses. Users will also be able to upload, share, edit, and delete any exam preparation materials they wish to associate with their profiles. This will allow students to connect with each other and help each other get through the classes that might be too much to handle alone. This will help students prepare for their exams more effectively, organize their notes more neatly, and connect with each other more smoothly.

The document will explain how the project is being managed and help keep the development of the project on track. It will also include how tasks are assigned and verified. The document will also include a list of the resource/budget requirements needed for the project.

### 1.3 Assumptions and Constraints

**Assumptions**

* The project is achievable
* The project will have a positive impact on student life
* The application will be able to interface with databases, allowing for the retrieval and storing of data

**Constraints**

* The project is to be completed within one school semester
* Entire project to be completed by only members of the development team
* The budget is limited

### 1.4 Project Deliverables

|  |  |
| --- | --- |
| Deliverable | Deadline |
| Project Proposal | 9/20/23 |
| Requirements and Analysis Documentation (SRS) | 9/25/23 |
| Project Description | 10/10/23 |
| Design Description (SDD) – Initial | 10/17/23 |
| Project Management Plan (SPMP) | 10/17/23 |
| Design Document – Final | 12/6 – 12/13/23 |
| Formal Oral Presentation / Demonstrations | 12/6 – 12/13/23 |

### 1.5 Schedule and Budget Summary

**Schedule**

The deliverables are on track to be delivered on time. The Project Proposal, System Requirements Specification, and Project and Design Description have been completed.

**Budget**

The calculated budget includes the cost of salary for each of our employees, as well as a potential the cost of hosting a website online using Firebase.

### 1.6 Evolution of the Plan

When there is a change to the system that impacts the schedule and management of the project, the change will be documented. Afterwards the plan will be updated and inspected. Finally, the updated plan will be distributed/posted for the targeted audience to see. Potential changes that could impact the schedule and management of the project would include changes in requirements or any potential defects.

## 

## 2. REFERENCES

## 

Team A40: Project Proposal, Version 2.0, 2/25/23

Project Webucation System Requirements Specification, SRS-001, Version 1.0, 3/2/23

Project Webucation System Requirements Specification, SRS-002, Version 2.0, 3/23/23

Project Webucation Software Project Management Plan, SPMP-001, Version 1.0, 4/6/23

Project Webucation System Requirements Specification, SRS-003, Version 3.0, 4/20/23

Team B40: Project Proposal, Version 3.0, 9/20/23

Project Webucation System Requirements Specification, SRS-004, Version 4.0, 9/25/23

## 3. DEFINITIONS

*Software Quality Assurance* – process that assures that all software engineering processes, methods, activities, and work items are monitored and comply with the defined standards.

*Web Server* – a piece of software that displays website content through storing, processing and delivering web pages to users

## 4. PROJECT ORGANIZATION

### 4.1 External Interfaces

The external interfaces for the project organization includes the software quality group, software testing group, the operations group that controls the management of all projects in our organization, and the organization that will provide the development environment.

### 4.2 Internal Structure

The internal structure will follow that of a democratic structure where everyone has equal say in the development of the project.

### 4.3 Roles and Responsibilities

|  |  |
| --- | --- |
| **Role** | **Team member** |
| Author | Sabahat Sami, Ivan Lin, Tanvi Rahman, Calvin Chu |
| Inspector | Ivan Lin, Tanvi Rahman |
| Reviewer | Sabahat Sami, Ivan Lin, Tanvi Rahman, Calvin Chu |
| Poster | Tanvi Rahman |

## 5. MANAGEMENT PROCESSES

### 5.1 Start-Up Plan

To initiate the Start-Up Plan, we formed a team of our own choosing, and subsequently selected a project that intrigued our team. Once the project was decided, we put forward the proposal for it.

#### 5.1.1 Estimation Plan

A meeting will be held after each sprint to assess the progress of the project. These meetings will be used to create a better estimation of the team’s efficiency in completing tasks on time.

#### 5.1.2 Staffing Plan

The staff will be four members that will be on board for the entire project. Three individual members will focus on one aspect of the project. These aspects are the backend, frontend, and database. The fourth member will assist each member depending on the workload distribution for each sprint.

#### 5.1.3 Resource Acquisition Plan

We will use the React framework, Turbogears framework, and PostgreSQL for the project. Any technical issues will be handled by looking through the appropriate documentation as well as the internet. For administrative issues we will consult with the teaching assistant Richa Nitin Doshi.

#### 5.1.4 Training Plan

Required training and resources for project team members:

* Web Development
* Backend Development
* Database Management
* UI/UX Design

### 5.2 Work Plan

#### 5.2.1 Work Activities

See the Gantt chart in section 12.3.

#### 5.2.2 Schedule Allocation

See the Gantt chart in section 12.3.

#### 5.2.3 Resource Allocation

See the Gantt chart in section 12.3.

#### 5.2.4 Budget Allocation

Employees will be paid a yearly salary of $200,000. The total estimate per year for four employees will be $800,000. Outside of the scope of this class, we would plan to use Firebase for hosting our website and database server. The estimated cost of using Firebase would be $40 a month which over three months would total $120.

### 5.3 Control Plan

#### 5.3.1 Requirement Control and Traceability

The requirements outlined in the SRS will be assigned distinct numeric identifiers to enable seamless tracing throughout the project. This approach will facilitate both forward-tracing up and back-tracing down to each requirement during the development process.

#### 5.3.2 Schedule Tracking and Adjustment

The team will operate using the SCRUM framework, utilizing single-week sprints and daily 20-minute meetings. Upon the completion of each sprint, the team will have another meeting to present an agile report and distribute tasks for the subsequent sprint so that all members are updated on the current state of the project and are aware of what the next steps should be moving forward.

#### 5.3.3 Budget Tracking and Adjustment

An Excel spreadsheet will be used to keep track of expenses and budget. This spreadsheet will be used to receive reimbursement from NYU. Any changes made to the budget must be approved by NYU, updated in the SPMP, and updated in the spreadsheet.

#### 5.3.4 Quality Control

A designated software quality assurance group will receive the requirements and be responsible for designing a comprehensive test plan that covers all functionalities of the project, along with the corresponding anticipated outcomes for each scenario. Once the test plan has been reviewed, the group will proceed to test the project and participate in inspections and walkthroughs of the code. Any defects discovered should be reported.

#### 5.3.5 Reporting Mechanisms

The team will have a weekly Tuesday meeting with the mentor for reports on the status of the project as well as any issues that could have arisen. The team will also hold weekly meetings on Facebook Messenger where they will discuss overall progress of the project and any potential issues. The team will also discuss any necessary adjustments to the project schedule. Each member will also be able to see the progress of development through the Git repository where they can see the commits made.

#### 5.3.6 Metrics Collection Plan

Metric collection will be done regularly, focusing on two key metrics: schedule adherence and defect rates. Both of these metrics will be tracked for each document and reported per team member.

### 5.4 Risk Management Plan

Business Risk: Low frequent users

Description: The number of people who utilize the application frequently are low

Probability: High

How discovered: Daily users report

Responsible Party Status: Users and designers

Mitigation Plan: Research and create easy-to-use UI and convenient features to encourage users to consistently use the application

Operational Risk: Server crash

Description: The server crashes or cannot handle the number of requests from different users

Probability: Medium

How discovered: Crash reports from users

Responsible Party Status: Server administrators and developers

Mitigation Plan: Frequent server analysis and testing to determine server needs, and expand if needed

Technology Risk: Application does not work or does not work well on some devices

Description: The application may not render well on different devices and input methods and may be more troublesome to use. For example, touch screen usage and mouse usage.

Probability: Low

How discovered: User feedback

Responsible Party Status: Designers and developers

Mitigation Plan: Frequent review of user feedback and conduct testing to determine how to improve the UI

Economic Risk: Low profitability

Description: The application may not be profitable enough to cover expenses.

Probability: High

How discovered: Financial report

Responsible Party Status: Designers and developers

Mitigation Plan: Conduct research on how to increase profits while retaining or increasing the number of users, such as ad revenue or a paywall for certain features.

### 5.5 Post Implementation Plan

The program files will be stored on Github. Different reports, including average user count, user feedback, and financial reports, will be gathered and consolidated for review.

## 6. TECHNICAL PROCESSES

### 6.1 Process Model

The process model for the project will be the object-oriented methodology, follow the waterfall lifecycle model.

### 6.2 Methods, Tools, and Techniques

The tools to be used include UML, Jira, Google Docs, Github, Discord, Figma, HTML/CSS, Javascript, and Python, Turbogears2, PostgreSQL, React.

### 6.3 Infrastructure Plan

The hardware for the project will include 4 computers for development, and one server to store and retrieve user information.

The software used will include Microsoft Windows OS, MacOS, Visual Studio Code, Figma, Discord, Github, and Jira.

### 6.4 Product Acceptance and Migration Plan

The product will be tested by the product testing group to verify that all functional and non-functional requirements have been met. The product will then be sent to the client for product acceptance testing and approval. The migration plan does not currently exist.

## 7. SUPPORTING PROCESSES PLANS

### 7.1 Configuration Management Plan

The documentation for the project will be stored on Google Drive. The code and other necessities for the project will be stored in a Github repository. Both will have team members accessing, creating, and controlling all pushes and changes.

### 7.2 Qualification (Verification and Validation) Plan

The qualification plan will include four different tests: Self test, peer review, walkthrough, and inspection. During self testing, programmers will individually test the components written by them to determine any defects. During the peer testing, programmers will test software written by others and themselves to determine any defects. Inspections will be performed by subject matter experts to find defects. Walkthroughs will be demonstrated by software engineers and managers for clients and users to find defects and to educate users.

### 7.3 Documentation (library) Plan

The documentation plan for the project entails that all documentation will be stored in Google Drive with a unique number for identification. Code and assets will be stored on Github.

### 7.4 Quality Assurance Plan

​​The software quality group will be given a set of requirements, around which the group will develop test cases. The test cases will then be run by the group, and any defects or suggested modifications will be reported. The software quality group will be involved in unit testing, component testing, and system testing.

### 7.5 Reviews and Audits

Each member of the group will be responsible for checking his own code and documentation before any part of the system or specification documents is deployed or distributed. The entire team will also peer review the code and documentation of their partners and make suggestions for any potential problems or improvements. A walkthrough of the entire document and application, as well as a final formal team inspection will be conducted. Finally, the document will be distributed to an external organization for a formal auditing process to be completed.

### 7.6 Problem Resolution Plans

Whenever a defect is found, it will be precisely documented in a defect tracking system and discussed with the rest of the team if needed. Each defect will be assigned a specific team member to handle, as well as a certain level of priority. Both of these will be used to schedule when the defect will be resolved by. It will be scheduled to be modified and included in a particular future release after being fixed and tested by the team.

Once the defect is resolved, all updates and modifications related to that defect will be documented and distributed in the anticipated release.

### 7.7 Environment Management Plans

The project environment will encompass React, Python (managed with pip), TurboGears, and PostgreSQL. The React environment will be self-contained, requiring no additional management. The Python environment will utilize pip for package management, while the TurboGears framework and PostgreSQL database will be seamlessly integrated to support the project's web application development needs. All components will be shared with the team for collaborative and efficient development.

### 7.8 Process Improvement Plan

Our team will be having a roughly 20-minute meeting once per week in which each member of the team will present the progress they have made over the course of the week. In these meetings, team members will get a chance to discuss what issues they’ve been dealing with, and check in with the potential difficulties their teammates have been experiencing. Members will also get a chance to reach out to one another to get assistance on any tasks they might need help with. From these discussions, the team will be able to understand how to help each other better and be more efficient in completing their tasks for the following week.

## 8. ADDITIONAL PLANS

To be updated as required.

## 9. INDEX

None at this time.

## 10. RATIONALE

None at this time.

## 11. NOTES

None at this time.

## 12. APPENDICES

### 12.1 Schedule Tracking

**Hours**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact/Deliverable | Team Member | Estimated | Actual | Difference |
| SRS – Domain | Ivan Lin | 2 hr | 1 hr | -1 hr |
| Calvin Chu | 1 hr | 2.5 hr | 1.5 hr |
| Sabahat Sami | 2 hr | 2 hr | 0 hr |
| Tanvi Rahman | 1.5 hr | 2.5 hr | 1 hr |
| Team Summary | 6.5 hr | 8 hr | 1.5 hr |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact/Deliverable | Team Member | Estimated | Actual | Difference |
| SRS – Requirements | Ivan Lin | 1 hr | 1.5 hr | 0.5 hr |
| Calvin Chu | 1.5 hr | 1.5 hr | 0 hr |
| Sabahat Sami | 1 hr | 1.5 hr | 0.5 hr |
| Tanvi Rahman | 2 hr | 2 hr | 0 hr |
| Team Summary | 5.5 hr | 6.5 hr | 1 hr |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Team Member | Estimated | Actual | Difference |
| SPMP - 001 | Ivan Lin | 3 hr | 4.25 hr | 1.25 hr |
| Calvin Chu | 3 hr | 3.75 hr | 0.75 hr |
| Sabahat Sami | 3 hr | 4 hr | 1 hr |
| Tanvi Rahman | 3 hr | 4.5 hr | 1.5 hr |
| Team Summary | 12 hr | 16.5 hr | 4.5 hr |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact/Deliverable | Team Member | Estimated | Actual | Difference |
| SRS – Analysis | Ivan Lin | 3 hr | 2.5 hr | -0.5 hr |
| Calvin Chu | 3 hr | 3 hr | 0 hr |
| Sabahat Sami | 2.5 hr | 3 hr | 0.5 hr |
| Tanvi Rahman | 2.5 hr | 2.5 hr | 0 hr |
| Team Summary | 11 hr | 11 hr | 0 hr |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact/Deliverable | Team Member | Estimated | Actual | Difference |
| SRS – Final | Ivan Lin | 2 hr | 1 hr | -1 hr |
| Calvin Chu | 2 hr | 1.5 hr | -0.5 hr |
| Sabahat Sami | 2 hr | 2 hr | 0 hr |
| Tanvi Rahman | 2 hr | 1.5 hr | -0.5 hr |
| Team Summary | 8 hr | 6 hr | -2 hr |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Team Member | Estimated | Actual | Difference |
| SPMP - 002 | Ivan Lin | 2 hr | 1.5 hr | -0.5 hr |
| Calvin Chu | 1 hr | 2 hr | 1 hr |
| Sabahat Sami | 1 hr | 0.5 hr | -0.5 hr |
| Tanvi Rahman | 2 hr | 2 hr | 0 hr |
| Team Summary | 6 hr | 6 hr | 0 hr |

**Cumulative**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Estimated | Actual | Difference |
| Ivan Lin | 13 hr | 11.75 hr | -1.25 hr |
| Calvin Chu | 11.5 hr | 14.75 hr | 3.25 hr |
| Sabahat Sami | 11.5 hr | 13 hr | 1.5 hr |
| Tanvi Rahman | 13 hr | 15 hr | 2 hr |
| Team Summary | 49 hr | 54.5 hr | 5.5 hr |

### 12.2 Defect Tracking

**Counts**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact/Deliverable | Team Member | Estimated | Actual | Difference |
| SRS – Domain | Ivan Lin | 4 | 6 | 2 |
| Calvin Chu | 2 | 7 | 5 |
| Sabahat Sami | 7 | 14 | 7 |
| Tanvi Rahman | 5 | 8 | 3 |
| Team Summary | 18 | 35 | 17 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact/Deliverable | Team Member | Estimated | Actual | Difference |
| SRS – Requirements | Ivan Lin | 5 | 7 | 2 |
| Calvin Chu | 3 | 8 | 5 |
| Sabahat Sami | 9 | 12 | 3 |
| Tanvi Rahman | 4 | 9 | 5 |
| Team Summary | 21 | 36 | 15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Team Member | Estimated | Actual | Difference |
| SPMP - 001 | Ivan Lin | 7 | 10 | 3 |
| Calvin Chu | 3 | 7 | 4 |
| Sabahat Sami | 5 | 3 | -2 |
| Tanvi Rahman | 10 | 2 | -8 |
| Team Summary | 25 | 22 | -3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact/Deliverable | Team Member | Estimated | Actual | Difference |
| SRS – Analysis | Ivan Lin | 6 | 9 | 3 |
| Calvin Chu | 5 | 8 | 3 |
| Sabahat Sami | 3 | 7 | 4 |
| Tanvi Rahman | 6 | 9 | 3 |
| Team Summary | 20 | 33 | 13 |

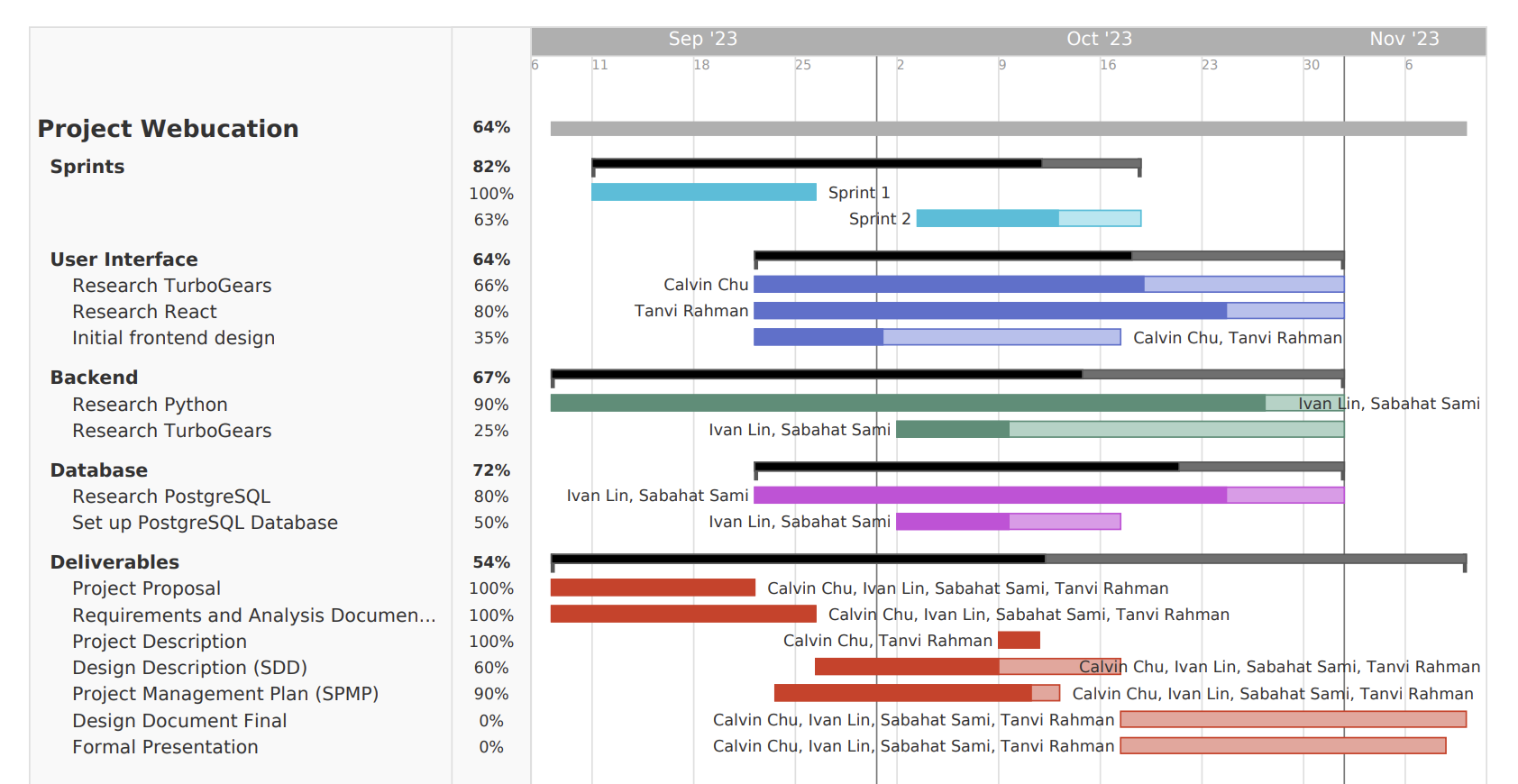
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact/Deliverable | Team Member | Estimated | Actual | Difference |
| SRS – Final | Ivan Lin | 5 | 6 | 1 |
| Calvin Chu | 5 | 2 | -3 |
| Sabahat Sami | 5 | 5 | 0 |
| Tanvi Rahman | 10 | 5 | -5 |
| Team Summary | 25 | 18 | -7 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Team Member | Estimated | Actual | Difference |
| SPMP - 002 | Ivan Lin | 10 | 15 | 5 |
| Calvin Chu | 10 | 0 | -10 |
| Sabahat Sami | 10 | 6 | -4 |
| Tanvi Rahman | 10 | 13 | 3 |
| Team Summary | 40 | 32 | -6 |

**Cumulative**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Estimated | Actual | Difference |
| Ivan Lin | 37 | 53 | 16 |
| Calvin Chu | 28 | 33 | 5 |
| Sabahat Sami | 39 | 47 | 8 |
| Tanvi Rahman | 45 | 46 | 1 |
| Team Summary | 149 | 179 | 30 |

### 12.3 Gantt Chart



Note: Condensing the overhead tasks was approved in class by Professor Strauss, as it is difficult to clearly show each individual subtask in an organized manner. Thus, we have condensed the dropdowns for each overarching task to show the main components of our Gantt Chart/Project Schedule.