

# **Software Project Management Plan**

**Group 8**

## **Expense Monitoring System**

**Software Engineering**

**Fall 2023**

**CS673 A1**



Abhishek Gupta  
Siddharth Krishnakumar  
Julia Peterson  
Jiuzhou Lu  
Haolong Yang

## 1. Overview

An Expense Monitoring System is a digital tool designed to help individuals from varying demographics and financial backgrounds to manage and track their financial expenditures efficiently. It provides users with a comprehensive view of their financial health by allowing them to record, categorize, and analyze their expenses. This system is especially valuable in today's fast-paced world, where individuals often juggle multiple income sources, expenses, and financial goals.

The goal of our project is to provide users with a comprehensive and interactive platform to monitor, track, categorize, and analyze their expenses by setting a budget limit monthly or yearly. This also helps the users study and track their spending patterns and revisit their spending on a monthly and yearly basis.

## 2. Purpose and Scope:

This Software Project Management plan outlines our approach toward managing, controlling, and tracking various components of configuration items in the project. Proper techniques will be followed so that they provide an efficient environment for team members working on the same project. These plans will ensure that project development is well-managed, and tracked and risks are mitigated as much as possible.

## 3. Definitions

- **EMS** – Abbreviation for Expense Monitoring System.
- **UI** - User Interface; how a user and a system interact.
- **GUI** - Graphical User Interface; Type of UI where interactions occur through elements like icons, buttons, and windows.
- **FQ** - Frequency, duration in which a wave repeats.
- **QA** - Quality Assurance.
- **UX** - User Experience.
- **CI/CD**: Continuous Integration and Continuous development; the process by which codes are integrated and a build is released.

## 4. Project Team Organization

### 4.1 Team Organization Graph

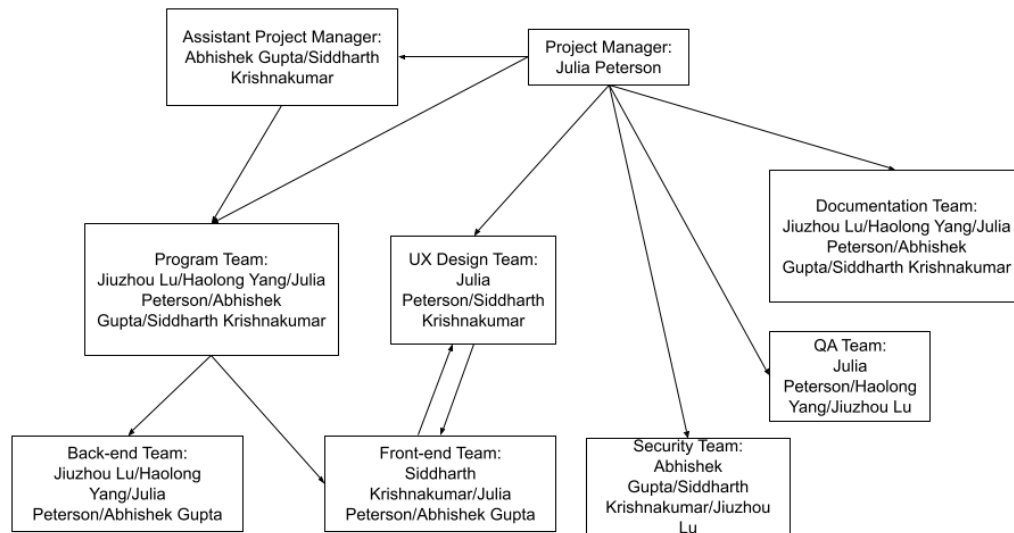


Fig: Team Structure.

### 4.2 Roles and Responsibilities:

We have split all the different tasks into teams to ensure that every team member is hands-on during the different stages of development. We are making sure that there is an equal division of labor, keeping in mind the different challenges each sub-team will face.

Members	Roles	Responsibilities
Abhishek Gupta	Team Lead, Full Stack developer.	Design and develop the architecture of the product. Development of product features and functionalities. Reviews codes and configurations.

Julia Peterson	Configuration Lead, QA, UI-UX Design,	Quality assurance and testing. Creates user experience design. Manages the whole team's progress. Creates software documents like design, plan, requirements, and analysis.
Siddharth Krishnakumar	Scrum Master, UI-UX, Security, Documentation	Schedules team huddle. Manages Jira stories and tasks. Creates user experience design. Ensures a secure application development process. Creates software documents like design, plan, requirements, and analysis.
Haolong	Back-End developer,	Implements robust backend APIs and functionalities in Python.
Jiuzhou	Front-End developer	Implements a responsive web interface in React.js.

**Documentation** - The whole team would work on the documentation report based on the work done by them for the week. This includes all required software engineering documents such as design, plan, requirements, and analysis.

#### 4. Project Management and Tools

The following tools and resources will be used for the management and development of the project. These are reputed resources that makes the development and management process efficient.

- Scheduling: Jira
- Task management: Jira
- Documents: Google Drive
- Design: Figma
- Source code management: GitHub
- Development environment: Visual Studio Code
- CI/CD: Docker(Subject to requirement).
- Language: Python, JavaScript, HTML, CSS
- Libraries: React.js for front-end development
- Framework: Django for back-end development
- Database: MongoDB as NoSQL database and MySQL/Postgres as SQL database. (May vary depending on technical changes, performance, and complexities.)
- Cloud Hosting: AWS (using the AWS Educate plan) or Google Cloud Platforms. (Subject to change as the project progresses in varying needs and capabilities.

## 5. Configuration Identification, Control and Management:

Various configuration items have been identified. Its control and management plan has been set up. This will make sure that any update and request follows its due process to be incorporated into the project.

### 5.1 Configuration Items:

- **Code (Production and Test):**  
The source code will be managed via Git and will be organized in a Git repository. The project will follow various code management techniques like version control and branching strategies.
- **Specification Documentation:**  
All documents will be stored in Google Drive with appropriate names and revision history specifying changes done in case of any revisions. This includes all documents such as design, requirements, plan, and analysis to user-specific documents such as guidelines and tutorials.
- **User Documentation:**  
This includes all documents specific to the end user such as guidelines and tutorials. With proper updates based on changing products.
- **Supporting Software:**  
In case of any third-party software usage, we will document the versions of all the software used in the project. All the dependencies will be documented, and their versions will be tracked to maintain uniformity across the project.

### 5.2 Source code version control:

- We will use git version controlling architecture to maintain our code base. Appropriate branching techniques will be followed. The naming convention will be defined.
- Each change will be submitted in the form of a pull request which will be reviewed and merged if expectations are met.
- No one will be allowed to work at the main branch directly.
- The project owner/lead will have access to the repository and will be responsible for reviewing the pull request and merging it.

### 5.3 Change Management:

- Changes requested will follow a process of request, review, and approval.
- Each change request should be documented with revised plans taking into consideration possible delays.

### 5.4 Progress Tracking:

- Jira will be used to create epics and user stories which will then be assigned to a member to work on.
- Members can log the number of hours taken to complete a user story.

### 5.5 Build and Release Management:

- Build will be released at decided intervals which will go through approvals from various stakeholders.
- The build must be approved by the Quality assurance team, Product Owner, and higher management before the final send-off is done.
- The build from the testing phase will now be released to production.

### 5.6 Audits and Reviews:

- Before any release, various audits will be done on the configuration to maintain SCM compliance.

## 6. Risk Management

Risks are the major concern when it comes to any development process. We have foreseen various risks that might occur as we go through the process and maintenance. We plan to tackle these risks with defined protocol as defined below. We might encounter some unforeseen risks in the future which will be tackled by our team with regular audits and reviews.

### 6.1 Identify potential risks.

#### Highest Priority:

1. Unauthorized access: Unauthorized changes made to code scripts and other configuration data.
2. Data Loss: Loss of codebase scripts and other configuration data.

#### Moderate Risk:

1. Scope Creep: Any additional features or requirements added at the time of development might add to possible delays.
2. Developmental delays: Due to any unforeseen technical issues or complexities there can be possible delays.
3. Data Breach: Exposure of expense data due to weak security configurations.

**Lower Risk:**

1. Team member Unavailability: The team members' unavailability due to any circumstances might affect the progress of the project.
2. Inefficient deployment process: Issues that come from inconsistencies in deployment environments.

**6.2 Risk Mitigation Strategies**

- Developmental delays: Thorough research and understanding should be done to avoid any such risk.
  - Owner: Abhishek Gupta
- Scope Creep: Our design and requirements must be made in a way that accommodates as many details as possible. A thorough market study should be done to point out all the requirements in the first place. This will reduce the risk by many folds.
  - Owner: Abhishek Gupta
- Team member Unavailability: The Project Manager should make sure the team members are available and appropriate replacements should be made in human resources in case of any absence.
  - Owner: Julia Peterson
- Data Loss: Scheduling backups of the configuration data would reduce the occurrence of data loss.
  - Owner: Haolong Yang
- Data Breach: Sensitive configuration data such as the API keys and database credentials must be encrypted securely.
  - Owner: Jiuzhou Lu
- Unauthorized access: Regular review of access logs and implementation of strong access controls can reduce unauthorized access.
  - Owner: Siddharth Krishnakumar
- Inefficient deployment process: Usage of automated deployment scripts and tools can mitigate inefficient deployment processes.
  - Owner: Siddharth Krishnakumar

## 7. Estimation

The total project's estimation depends upon the size of the project and the level of hosting and deployment. Although the size of the project or application may vary. The components in which the estimation of cost is done remain the same. Assuming the project is medium size, The estimation is done for human resources, and the cost for the application development.

### 7.1 Human Resource Estimation:

Components	Hours worked	Hourly Cost	Total Cost
Project Management	100 hours	40\$/hr	4000\$
Requirement Analysis	75 hours	35\$/hr	2625\$
UI/UX Design	120 hours	30\$/hr	3600\$
Development	500 hours	50\$/hr	2500\$
Quality Assurance and Testing	150 hours	35\$/hr	5250\$
Deployment and Infrastructure	50 hours	60\$/hr	3000\$
Documentation	30 hours	40\$/hr	1200\$

### 7.2 Hosting and Licensing:

Components	Monthly cost	Yearly Cost
Cloud hosting services	\$ 350	\$ 3000
Database	\$100	\$ 1200
Miscellaneous	\$ 60	\$ 800

Costs include the software licenses, the third-party services, and miscellaneous costs summing up to a total of 5000\$. The total cost estimated for deploying the Expense Monitoring System in a medium size level comes up to **\$ 27,175**.

Note: Hypothetical simulation of cost estimation is done here. In reality, it is an open-source project.



## 8. Managerial Process Plans

Managerial processes and plans for project development are discussed below.

### 8.1 Staffing Plan

Julia Peterson, Siddharth Krishnakumar, Abhishek Gupta, Haolong Yang, Jiuzhou Lu

### 8.2 Resource Acquisition Plan

All the software required is open source and free to use with the community version.

### 8.3 Technical Process Plan

An Agile Software Development approach will be taken for developing this project. The process follows a cycle of iterations as depicted in the chart below.

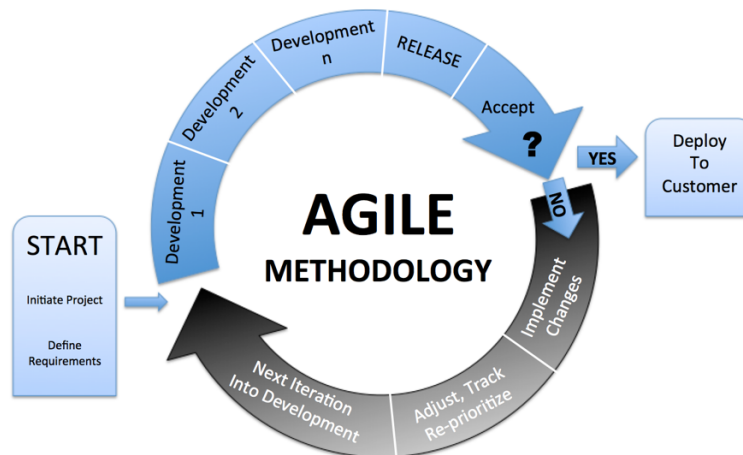


Fig: Agile process cycle  
[Reference](#)

### 8.4 Resource Allocation

Each member has access to the same resources as all the other members. The resources include codes from GitHub and all the documentation from Google Drive and Jira.

### 8.5 Budget Allocation

Budget allocation should be done considering the estimation provided in the estimation section. However, the budget that fits to be allocated to this project should be \$ 30,000.00 as the project should remain in positive cash flow in case additional resources are required.

## 9. Scheduling

Week number/Due date	To-do List
Week 4 (09/26)	<ul style="list-style-type: none"><li>• Project Status Report 1</li><li>• Software Project Management Plan</li><li>• Create GitHub repository</li></ul>
Week 5 (10/03)	<ul style="list-style-type: none"><li>• Project Status Report 2</li><li>• Prototype #1 Design</li></ul>
Week 6 (10/10)	<ul style="list-style-type: none"><li>• Project Status Report 3</li><li>• Prototype #1 Development and testing</li></ul>
Week 7 (10/17)	<ul style="list-style-type: none"><li>• Final testing of Prototype #1</li><li>• Presentation slides</li></ul>
Week 8 (10/24)	<ul style="list-style-type: none"><li>• Project Status Report 4</li><li>• Prototype #2 Development</li></ul>
Week 9 (10/31)	<ul style="list-style-type: none"><li>• Project Status Report 5</li><li>• Prototype #2 Development</li></ul>
Week 10 (11/07)	<ul style="list-style-type: none"><li>• Project Status Report 6</li><li>• Prototype #2 Development and testing</li></ul>
Week 11 (11/14)	<ul style="list-style-type: none"><li>• Project Status Report 7</li><li>• Prototype #3 Development</li></ul>
Week 12 (11/21)	<ul style="list-style-type: none"><li>• Project Status Report 8</li><li>• Prototype #3 Development</li></ul>
Week 13 (11/28)	<ul style="list-style-type: none"><li>• Project Status Report 9</li><li>• Prototype #3 Development and testing</li></ul>
Week 14 (12/05)	<ul style="list-style-type: none"><li>• Project Status Report 10</li><li>• Prototype #3 Development and testing</li></ul>
Week 15 (12/12)	<ul style="list-style-type: none"><li>• Presentation slides</li><li>• Final testing of Prototype #3</li><li>• Delivery of Final Product</li></ul>

## 10. Planning & Monitoring

Slack and Zoom will be used to ensure effective project monitoring. Daily updates and discussions will take place via Slack, where team members can share progress, discuss challenges, and seek immediate assistance when needed. In-depth project assessments will be conducted weekly using Zoom including reviewing project milestones, addressing any deviations from the plan, and making necessary adjustments. Reports provided by Jira will help track team progress as well.

## 11. References

- <https://www.cprime.com/resources/what-is-agile-what-is-scrum/>
- <https://hqsoftwarelab.com/blog/writing-a-project-management-plan-for-software-development/>
- [https://en.wikipedia.org/wiki/Agile\\_software\\_development](https://en.wikipedia.org/wiki/Agile_software_development)