Icon

Description automatically generatedv0.1-03.07. 2023

Software Project Management Plan

**Project ROME**

Rasool Abbas, Onel Jimenez, Michael Mcfarlin, Ethan Wharton

Table of Contents

[Revision History 1](#_Toc133666901)

[1 Overview 2](#_Toc133666902)

[1.1 Project Title and Objectives 2](#_Toc133666903)

[1.2 Project Deliverables 2](#_Toc133666904)

[1.2 Timeline 3](#_Toc133666905)

[1.4 Assumptions and Constraints 3](#_Toc133666906)

[1.6 Success Criteria 4](#_Toc133666907)

[2 Startup Plan 5](#_Toc133666908)

[2.1 Team Organization 5](#_Toc133666909)

[2.2 Project Communications 5](#_Toc133666910)

[2.3 Tools and Technologies 5](#_Toc133666911)

[3 Work Plan 6](#_Toc133666912)

[3.1 Activities and Tasks 6](#_Toc133666913)

[3.2 Release Plan 7](#_Toc133666914)

[4 Control Plan 8](#_Toc133666915)

[4.1 Monitoring and Control 8](#_Toc133666916)

[4.2 Metrics Collection 8](#_Toc133666917)

[5 Supporting Process Plans 9](#_Toc133666918)

[5.1 Risk Management Plan 9](#_Toc133666919)

# Revision History

|  |  |  |
| --- | --- | --- |
| Version | Revising Author | Description |
| v0.1-03.07.2023 | Onel Jimenez | Initialize document – begin formatting and process. |

# 1 Overview

## Project Title and Objectives

The main objectives of Project ROME are to develop a platform that is both secure and aesthetically pleasing, providing users with a seamless experience when viewing and donating to various fundraisers and events. During the design phase, particular attention was paid to the selection of color schemes that complement each other, in order to create a visually appealing UI. Moreover, the elements and artifacts that are interactable across different pages of the application were designed to be visually appealing, while also being easy to understand.

Furthermore, the development team has focused on selecting a modern development stack that would enable the creation of an application that is accessible to all users, regardless of their technical knowledge. The user interface has been designed to incorporate dynamic pages that can adapt to various screen sizes, without compromising on aesthetics or functionality. By prioritizing these aspects, Project ROME aims to create an enjoyable user experience that is both visually attractive and user-friendly.

## 1.2 Project Deliverables

The main deliverables for Project ROME include a functioning and deployed website that can be accessed by stakeholders on their personal computers as stated above.

Additionally, documentation outlining the design and development process should be provided. This documentation should provide a comprehensive overview of the project, including the selection of color schemes, design elements, and technical specifications. It should also include detailed documentation of the development process, including any challenges faced, strategies used to overcome these challenges, and lessons learned throughout the project.

Finally, well-documented and commented readable source code. This code should be clean and well-structured, with comments and documentation that provide a clear understanding of the code's functionality and purpose. The code should be easy to read and understand, even for developers who are not familiar with the project. By providing these deliverables, we can ensure that Project ROME is successful and meets the needs of our stakeholders.

## Timeline

Sprint 1:

* Finalize decision on project to complete.
* Plan and design the website's architecture and interaction between pages.
* Create Trello board for task delegation.
* Begin planning UI design.

Sprint 2-3:

* Finalize UI design plans.
* Begin building the database, API, and coding UI.
* Start implementing basic functionality.

Sprint 4:

* Prepare for live demo/prototype showcase for stakeholders.
* Website should be mostly functional.

Sprint 5-6:

* Continue polishing existing code.
* Implement any remaining features.
* Attempt to complete stretch goals if possible

Sprint 7:

* Finish all remaining tasks.
* Fix any remaining bugs.
* Finalize documentation and source code.

## 1.4 Assumptions and Constraints

Assumptions:

* The development team has the necessary technical skills to create a functional fundraising website.
* Stakeholders will provide clear requirements for the website's functionality.
* The development team will have consistent access to the required technology and software throughout the project.
* Stakeholders will be available for regular check-ins and feedback throughout the project.

Constraints:

* The development team has limited professional experience and virtually no budget.
* The project must be completed within a 3 to 4 month timeline (late January – early May).
* The development team will be unable to hire additional team members or outsource any work.
* The website must be secure and adhere to relevant privacy laws.
* The website must be accessible to all users, regardless of their technical knowledge or physical abilities.
* The website must be able to handle high volumes of traffic.
* The development team must work within the constraints of the chosen technology stack.
* The website must be designed and developed with a focus on usability and user experience.

## 1.6 Success Criteria

* A functional and deployed website that meets all the requirements listed above.
* Code written in either .Net C# or Java, with 10% code coverage using the xUnit testing framework.
* Use of SQL Server 2012 or above as the database management system.
* Front-end design must incorporate at least one CSS framework, such as Bootstrap, and must be usable and accessible for all users.
* The framework or library used in the project must be properly documented and included with the project.
* Pages included in the project must meet the requirements outlined above, including a login page, home page/dashboard, view other users' fundraisers, individual fundraiser pages, view fundraiser details and donations, donation form, and user profile and settings.
* Proper attention paid to negative space, text readability, and alignment of all elements.
* Documentation outlining the design and development process, including any challenges faced and how they were overcome.
* Well-documented and commented source code that is easily readable and understandable by other developers.

# 2 Startup Plan

## 2.1 Team Organization

1. Development Lead: This team member will take the lead on development and ensure that the project is progressing according to the development plan. They will work closely with the other members of the team to ensure that development is aligned with the project's goals.
2. Documentation Lead: This team member will be responsible for documenting the project's progress and keeping the team up to date on any changes. They will work closely with the other members of the team to ensure that all updates and changes are properly recorded and communicated.
3. Design Lead: This team member will take the lead on design and ensure that the project has a cohesive and user-friendly design. They will work closely with the other members of the team to ensure that the design aligns with the project's goals and requirements.
4. Database Lead: This team member will be responsible for implementing and deploying the database. They will work closely with the other members of the team to ensure that the database meets the project's requirements and is properly integrated with the rest of the project.

While each team member will have their primary responsibilities, they will also work collaboratively with each other and provide support as needed. Communication between team members will be key to ensure that the project is progressing smoothly and that everyone is aware of any changes or updates. Additionally, regular meetings and check-ins will be scheduled to discuss progress and address any issues that may arise.

## 2.2 Project Communications

Our group will have 7 sprints, each lasting 2 weeks. We will start each sprint with a meeting on the first Monday and end each sprint with a meeting on the last Friday. Throughout the sprint, we can communicate freely and asynchronously through SharePoint, Discord, and Trello. These tools will allow us to share updates on our progress, delegate tasks, and discuss any issues or questions that arise. It will be important for us to check these platforms regularly to stay up to date with the team's progress and ensure that we are meeting our goals for each sprint.

## 2.3 Tools and Technologies

* Development Framework: .NET Core with C#
* Database: PostgreSQL/Docker
* Unit Testing Framework: nUnit for backend, jest for frontend, and playwright for end-to-end testing
* Front-End Framework: React
* Version Control: Git
* Project Management: Trello, SharePoint
* Integrated Development Environment (IDE): Visual Studio, Visual Studio Code, Docker

# 3 Work Plan

## 3.1 Activities and Tasks

Design and Planning:

* Finalize project selection.
* Determine project scope and requirements.
* Develop UI/UX design plans.
* Create architecture design.
* Determine necessary frameworks and libraries.
* Set up Trello board for task delegation.

Backend Development:

* Build database.
* Build API.
* Code backend functionality.
* Ensure 10% code coverage.
* Implement xUnit framework.

Frontend Development:

* Code front-end functionality.
* Implement Bootstrap framework.
* Ensure text readability and alignment.
* Ensure negative space is properly used.

User Authentication and Profile Management:

* Design and develop login page.
* Implement user authentication and authorization.
* Create user profile and settings page.

Fundraiser Pages:

* Develop home page/dashboard.
* Create individual fundraiser pages.
* Design and develop pages for viewing other users' fundraisers.
* Design and develop pages for viewing fundraiser details and donations.

Donation Functionality:

* Develop donation form functionality.
* Implement secure payment processing.
* Test donation functionality

Testing and Debugging:

* Conduct unit testing and debugging.
* Ensure website functionality and security.
* Squash any remaining bugs.

Documentation:

* Document design and development process.
* Provide documentation on how to use and maintain the website.
* Include well-documented and readable source code.

## 3.2 Release Plan

1. Mid-term Prototype (End of Sprint 4, mid – late March):

* A functional website that implements the minimum viable product features as defined by the success criteria.
* Deploy the website to a staging environment (if possible).
* Conduct user testing and get feedback from stakeholders to ensure the project is on track to meet the desired outcomes.

1. Final Deployment (End of Sprint 7, beginning of May):

* A fully functional website that meets all the success criteria and has been thoroughly tested.
* Deploy the website to the production environment.
* Ensure all documentation and source code is properly archived and accessible to the stakeholders.

During the release phase, it's important to ensure that the website is deployed in a way that is accessible to all users and that it can handle the expected traffic. It's also important to ensure that all stakeholders are informed about the release plan and any potential downtime or disruptions that may occur. Any necessary training or support should be provided to stakeholders to ensure a smooth transition to the new website.

# 4 Control Plan

## 4.1 Monitoring and Control

1. Progress Tracking: The team will track progress towards the project goals by utilizing the Trello board. Each task will be assigned to an individual team member and will be tracked from start to completion. This will ensure that progress is being made in a timely and efficient manner and that the project stays on track.
2. Quality Assurance: The team will use automated testing tools such as nUnit for backend testing, Jest for frontend testing, and Playwright for end-to-end testing to ensure that the code is of high quality and that any issues or bugs are caught and addressed in a timely manner.
3. Code Reviews: The team will conduct regular code reviews to ensure that the code is being written according to best practices, and that any errors or inefficiencies are addressed in a timely manner.
4. Risk Management: The team will identify and manage any risks associated with the project, such as potential delays or issues with technology, and develop contingency plans to mitigate these risks.
5. Communication: The team will communicate regularly and effectively through Discord and SharePoint. Team members will be encouraged to ask questions and provide feedback, and regular meetings will be held to ensure that everyone is on the same page and that the project is progressing as planned.

By implementing these monitoring and control strategies, the team can ensure that the project is completed on time, and to the satisfaction of all stakeholders.

## 4.2 Metrics Collection

1. Task Completion: This metric would track the number of tasks completed by each team member during a sprint. It would provide an indication of each member's productivity and contribution to the project. This will be tracked through Trello.
2. Estimated vs Actual Effort: This metric would compare the estimated effort required to complete a task with the actual effort expended by the team member. It would help to identify areas where team members may be overestimating or underestimating their workload.
3. Code Coverage: This metric would measure the percentage of the code that is covered by automated unit tests. It would provide an indication of the quality and completeness of the codebase.
4. Code Quality: This metric would track the overall quality of the codebase, including factors such as adherence to coding standards, maintainability, and scalability.

These metrics would be collected regularly throughout the project and used to monitor the team's progress and identify areas where improvements could be made. The Trello board and the Estimated Effort spreadsheet would be used to track these metrics and ensure that they are easily accessible to the team. Regular team meetings and reviews would be held to discuss the metrics and make any necessary adjustments to the project plan.

# 5 Supporting Process Plans

## 5.1 Risk Management Plan

From the team's perspective, risk management for Project ROME will involve identifying and communicating potential risks to the project manager and other stakeholders. We will need to work together to develop mitigation strategies, such as testing plans, contingency plans, and risk response plans, to minimize the impact of these risks on the project.

As developers, we should take steps to mitigate technical risks by writing clean and well-documented code, conducting thorough testing, and debugging, and staying up to date with the latest technologies and best practices. To address resource risks, we would communicate our needs for resources such as time, equipment, and training, and work with the project manager to ensure that these needs are met.

We would also be responsible for addressing schedule risks by working collaboratively and efficiently to meet project milestones and deadlines. To mitigate business risks, we should stay up to date with market trends and user needs and communicate any concerns or challenges to the project manager and other stakeholders.

Throughout the project, we should prioritize effective communication and collaboration to ensure that risks are identified and addressed in a timely and effective manner. By taking a proactive approach to risk management, we can ensure that Project ROME is successful and meets the needs of our users.