

PROJECT PLAN FOR FUNDING REQUESTS MANAGEMENT SYSTEM

Project Title: FundDocker – Funding Requests Management System

Date: 14 April 2024

Team:

Amaan Hanslod (2541305)

Anist Mampuru (449541)

Muhammed Ebrahim (2341063)

Nthabiseng Ramalepe (2429446)

Thabiso Mahloala (2426592)

Zahra Kotwal (2628128)

1. Introduction

- Project Overview

The aim of this project is to develop a web-based application to assist organisations in managing and advertising their funding opportunities. This application will cater to Educational, Business and Event funding – providing a central platform for funding applications, managing applications, reviews, budgets and reporting.

- Purpose and Objectives

Purpose:

To equip organisations with a central platform for advertising funding opportunities and managing applications efficiently.

Objectives:

- Develop a responsive web application using Agile Methodology.
- Incorporate Continuous Integration and Continuous Deployment (CI/CD) principles.
- Ensure usability and accessibility.
- Utilize a test-driven development approach.

- Scope

In Scope:

- **Design and Development:**

Creation of responsive web application to manage funding requests.

Development of user interfaces for different user roles.

Integration of CI/CD pipelines to automate testing and deployment processes.

- **Features and Functionalities:**

Capability for organisations to post detailed funding opportunities in categories – educational, business, event funding.

Functionality for users to submit applications for funding and fund managers to track and manage these applications.

Management of funding budgets, including allocation tracking.

Generation of detailed reports on funding allocations, changes in budget.

- **Testing and Quality Assurance:**

Implementation of a test-driven development (TDD) approach.

Unit testing, integration testing and user acceptance testing (UAT) to ensure the system meets all the requirements.

Performance and security testing to ensure the applications is robust and secure.

Out of Scope:

- **Long-Term Maintenance:**

Ongoing support and maintenance activities beyond initial post-launch period.

Continuous updates and feature enhancements not included in initial project plan.

- Stakeholders

- **Project Owner:** Lucky Nkosi and Shakeel Malagas

- **Development team:**

- Amaan Hanslod

- Anist Mampuru

- Thabiso Mahloala

- **Design Team:**

- Muhammed Ebrahim

- Nthabiseng Ramalepe

- Zahra Kotwal

- **End Users:**

- Organisations offering funding and applicants.

2. Project Requirements

- Functional Requirements

User Verification: Use 3rd party identity provider to verify users, assign roles – Applicants, Fund Manager, Platform Admin.

User Management: Admins should be able to manage users – review fund managers, block users, manage access and change permissions.

Funding Opportunities Adverts: Allow organisations to post and manage funding opportunities.

Applications: Enable users to create accounts, apply for different types of funding, upload and view attachments and track application status.

Funding Review: Provide tools for organisations to review and manage applications and approve or reject applications.

Budgeting: Allow organisations to view how much money is in the system and manage it accordingly.

Reporting: Generate export reports on usage of funds in the system over time, applications data as PDFs.

- Non-Functional Requirements

Performance: Ensure fast load times and efficient processing.

Security: Implement security measures to protect users' data.

Scalability: Design system to handle increasing numbers of users and applications.

Usability: Ensure the system is user-friendly and accessible.

3. Project Timeline

- Milestones and Deadlines

Initiation Phase:

- Duration: 2-3 days
- Deliverables: Requirements gathering, initial research and project planning.

Sprint 1:

- Duration: 1 week
- Deliverables: Wireframes, design mock-ups, UI/UX design, user verification and management, code coverage tool installed, software tracking and app deployed.

Sprint 2:

- Duration: 1 week
- Deliverables: Post ads, approve/reject applications for fund managers; Post applications and track status for users, unit tests; UML diagrams.

Sprint 3:

- Duration: 1 week
- Deliverables: Upload attachments for users, total money budget for fund managers, view and export reports (graphs).

Sprint 4:

- Duration: 1 week
- Deliverables: Notifications to user when application status changes and to fund managers when they receive a new application, product polish.

4. Project Resources

- Team Roles and Responsibilities

Developers: Code the website, integrate CMS, ensure functionality, set up CI/CD pipeline.

Designers: Create design mock-ups, UI/UX design, ensure responsive layout.

- Tools and Technologies

Project Management: Notion (Scrum Board)

Design: Creately

Development: HTML, CSS, JavaScript, GitHub Actions

Testing: Code cov

5. Budget

- Cost estimates per resource allocation:

- Development Team: Majority of Effort
- Design Team: Significant Effort
- Testing: Moderate Effort

6. Risk Management

- Risk Identification

- Delays due to academic workload.
- Technical issues with CI/CD setup
- Miscommunication among team members

- Risk Mitigation Strategies

- Regular progress reviews
- Allocate extra time for unforeseen issues.
- Use collaborative tools for clear communication – communication channels like WhatsApp and Discord and task tracking on Notion.

7. Communication Plan

- Meeting Schedule:

- Sprint Planning: At the start of every sprint.
- Daily Scrum meetings: Brief daily meetings over Discord to discuss progress and impediments.
- Sprint Reviews: At the end of each sprint to review progress with stakeholders.
- Sprint Retrospective: Post-sprint meeting to reflect on the process and improve.

- Reporting:
 - Status reports via WhatsApp/Discord.
 - Progress updates through project management tool, Notion.

8.Quality Assurance

- Testing Plan
 - Unit and integration testing.
 - User Acceptance testing by peers/team members.
 - Accessibility and performance testing.
- Acceptance Criteria
 - All functional and non-functional requirements met.
 - No critical bugs or issues.
 - Approval from client.

9.Launch Plan

- Pre-Launch:
 - Finalise all content.
 - Conduct full site testing.
 - Backup project files.
- Launch Strategy:
 - Launch for internal review.
 - Final public launch for assessment.
- Post-Launch:
 - Monitor website performance.
 - Address any initial issues promptly.