Icon

Description automatically generated

v1.1.1-04.30.2023

Architecture and Design

**Project ROME**

Rasool Abbas, Onel Jimenez, Michael McFarlin, Ethan Wharton

Table of Contents

[Revision History 2](#_Toc133697638)

[Introduction 2](#_Toc133697639)

[Design Goals 2](#_Toc133697640)

[Process View 3](#_Toc133697641)

[System Behavior 4](#_Toc133697642)

# Revision History

|  |  |  |
| --- | --- | --- |
| **Version** | **Revising Author** | **Description** |
| v1.1.1-04.30.2023 | Onel Jimenez | Small design edits and revisions |
| v1.1-04.25.2023 | Ethan Wharton | Revisions, finished behavior diagrams and process diagrams. |
| v1.0-03.20.2023 | Ethan Wharton | Introduction and beginning on views. |
| v0.1-03.07.2023 | Onel Jimenez | Initialize document – begin formatting and process. |

# Introduction

This document presents the architecture and design of the three-tier software for Project ROME, a fundraising website that provides a platform for hosting, viewing, and donating to various fundraisers and events. The three tiers, consisting of ROME UI, ROME API, and ROME Database, work together seamlessly to provide a secure and efficient user experience.

The purpose of this document is to provide an overview of the software architecture and design, including the technologies and tools used to develop each tier, the data flow between tiers, and the user interface design. Additionally, this document will illustrate the technical considerations and decisions that have been made to ensure the reliability, scalability, and security of the platform.

By documenting the architecture and design of the three-tier software, this document serves as a reference for developers and stakeholders to understand how the platform works and how it has been designed to meet the needs of users.

# Design Goals

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.

# Process View

Graphical user interface

Description automatically generated

Project ROME is a fundraising website that uses a three-tier software architecture consisting of ROME UI, ROME API, and ROME Database. The UI layer is responsible for displaying the website's content and features to the user, while the API layer acts as an intermediary between the user and the database, processing user requests and returning responses. The database layer stores all the data associated with the website, including user profiles, donation records, and fundraising goals.

When a user visits the ROME UI, they can interact with the website's features and submit requests. These requests are sent to the ROME API layer, which authenticates the user and processes the request. The API layer then retrieves or updates the necessary data from the ROME Database, and sends a response back to the UI layer, which displays the result to the user. This process enables the website to handle high volumes of traffic, maintain data integrity, and provide a user-friendly experience.

Overall, the three-tier software architecture of Project ROME enables the website to handle user requests and interact with the database in a secure and efficient way, providing a seamless user experience.

# System Behavior



