

Software Requirements Specification

for

Wanderer

Version: 0.1

Prepared by: Mariya Hossan Shayle

Organization: Developic

Date Created: 23 November 2024

Table of Contents

| | |
|---|-----------|
| Revision History..... | 3 |
| 1. Introduction..... | 4 |
| 1.1 Document Purpose..... | 4 |
| 1.2 Product Scope..... | 4 |
| 1.3 Definitions, Acronyms, and Abbreviations..... | 4 |
| 1.4 References..... | 4 |
| 1.5 Document Overview..... | 4 |
| 2. Product Overview..... | 4 |
| 2.1 Product Perspective..... | 4 |
| 2.2 Product Functions..... | 5 |
| 2.3 Product Constraints..... | 5 |
| 2.4 User Characteristics..... | 6 |
| 2.5 Assumptions and Dependencies..... | 6 |
| 2.6 Apportioning of Requirements..... | 6 |
| 3. Requirements..... | 6 |
| 3.1 External Interfaces..... | 6 |
| 3.1.1 User Interfaces..... | 6 |
| 3.1.2 Hardware Interfaces..... | 8 |
| 3.1.3 Software Interfaces..... | 8 |
| 3.2 Functional Requirements..... | 10 |
| 3.3.1 Performance..... | 12 |
| 3.3.2 Security..... | 12 |
| 3.3.3 Reliability..... | 12 |
| 3.3.4 Availability..... | 12 |
| 3.4 Compliance..... | 13 |
| 3.5 Design and Implementation..... | 13 |
| 3.5.1 Installation..... | 13 |
| 3.5.6 Cost..... | 13 |
| 3.5.7 Deadline..... | 13 |
| 4. Verification..... | 14 |

Revision History

| Version | Date | Description | Author |
|----------------|------------------|----------------------|----------------------|
| 0.1 | 23 November 2024 | Initial draft of SRS | Mariya Hossan Shayle |

1. Introduction

1.1 Document Purpose

This Software Requirements Specification (SRS) defines the functionalities, constraints, and design requirements for the Wanderer travel website. It serves as the foundational reference for stakeholders, developers, testers, and maintenance teams.

1.2 Product Scope

Wanderer is a web-based platform aimed at simplifying travel planning and management by offering:

- User registration, login, and role-based dashboards.
- Travel package browsing and secure trip booking.
- Blogging functionality for travel experiences.
- Administrative tools for content management and analytics.

1.3 Definitions, Acronyms, and Abbreviations

- **Traveler:** A general user exploring destinations and booking trips.
- **Blogger:** A user sharing travel experiences via blogs.
- **Admin:** A staff member managing roles, content, and system operations.
- **PCI DSS:** Payment Card Industry Data Security Standard.

1.4 References

1. Wanderer Project Proposal Document.
2. IEEE SRS Standard (ISO/IEC/IEEE 29148:2018).
3. PCI DSS Compliance Guidelines.

1.5 Document Overview

This SRS outlines the system's functionality, architecture, design constraints, and verification strategies. It includes visual aids, such as flowcharts and diagrams, for clarity.

2. Product Overview

2.1 Product Perspective

Wanderer is an independent, responsive platform integrating third-party APIs for travel data and social media logins. Payment gateways such as Stripe and PayPal ensure secure transactions.

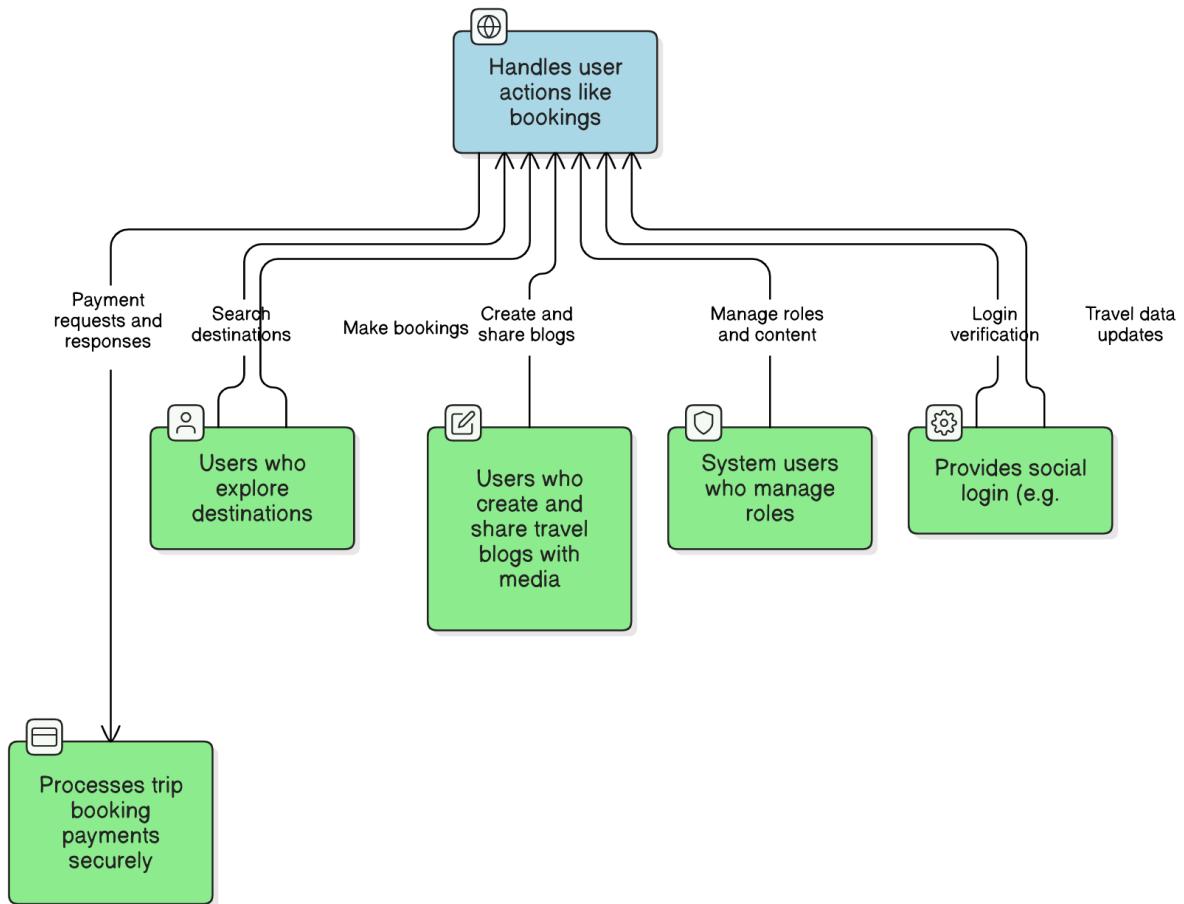


Figure 1: Context diagram for Wanderer website.

2.2 Product Functions

The platform supports the following core functionalities:

- **User Registration and Login:** Travelers and Bloggers can register and log in.
- **Travel Package Management:** Browse, book, and manage trips.
- **Blog Submission and Moderation:** Bloggers can create posts, and admins moderate them.
- **Administrative Controls:** Manage content, analytics, and user roles.

2.3 Product Constraints

- Compliance with PCI DSS for payment processing.

- Multi-language support is required for global accessibility.

2.4 User Characteristics

1. **Travelers:** Non-technical users who explore and book trips.
2. **Bloggers:** Content creators with basic technical skills.
3. **Admins:** Technically proficient users managing the system.

2.5 Assumptions and Dependencies

- Reliable internet connectivity is assumed.
- Payment gateways and third-party APIs will remain operational.

2.6 Apportioning of Requirements

Features such as advanced AI-based travel recommendations will be implemented in future iterations.

3. Requirements

3.1 External Interfaces

3.1.1 User Interfaces

1. **Traveler Interface:**
 - Personalized dashboards with booking history.
 - Search and browse travel packages.
2. **Blogger Interface:**
 - Blog submission forms with multimedia support.
3. **Admin Interface:**
 - Role management and analytics dashboards.

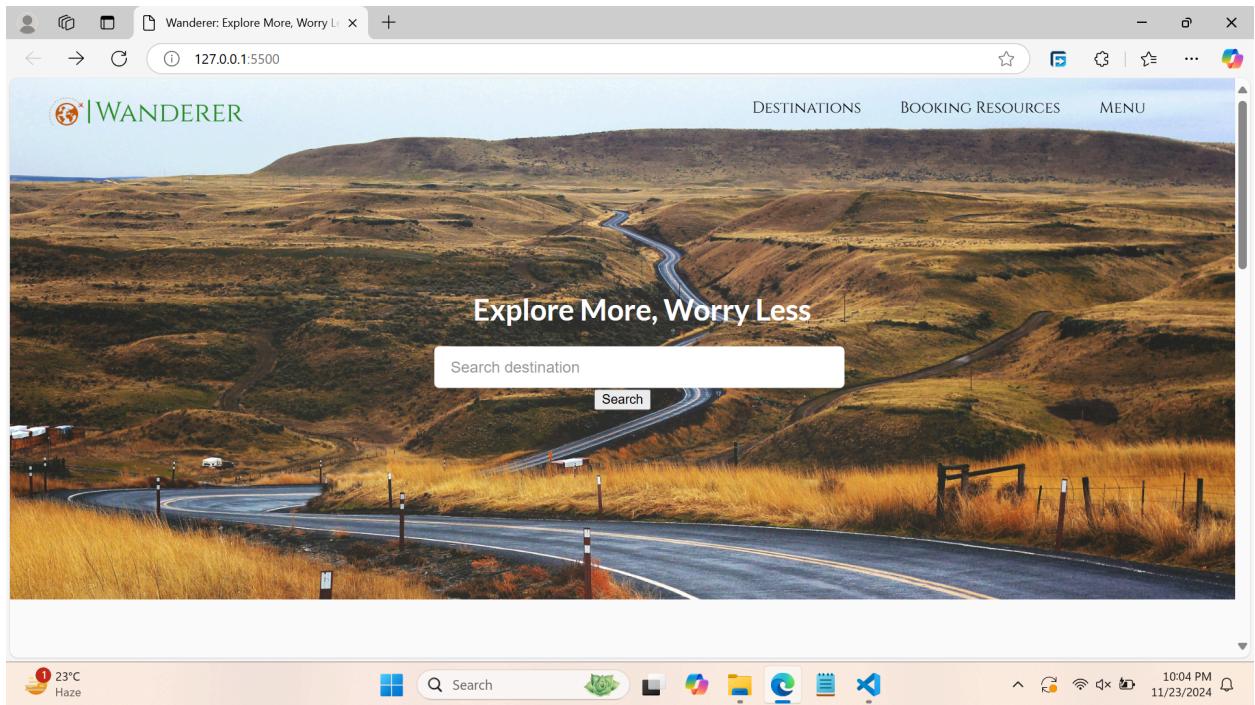


Figure 2: Homepage Interface - The main landing page for users to explore destinations and packages.

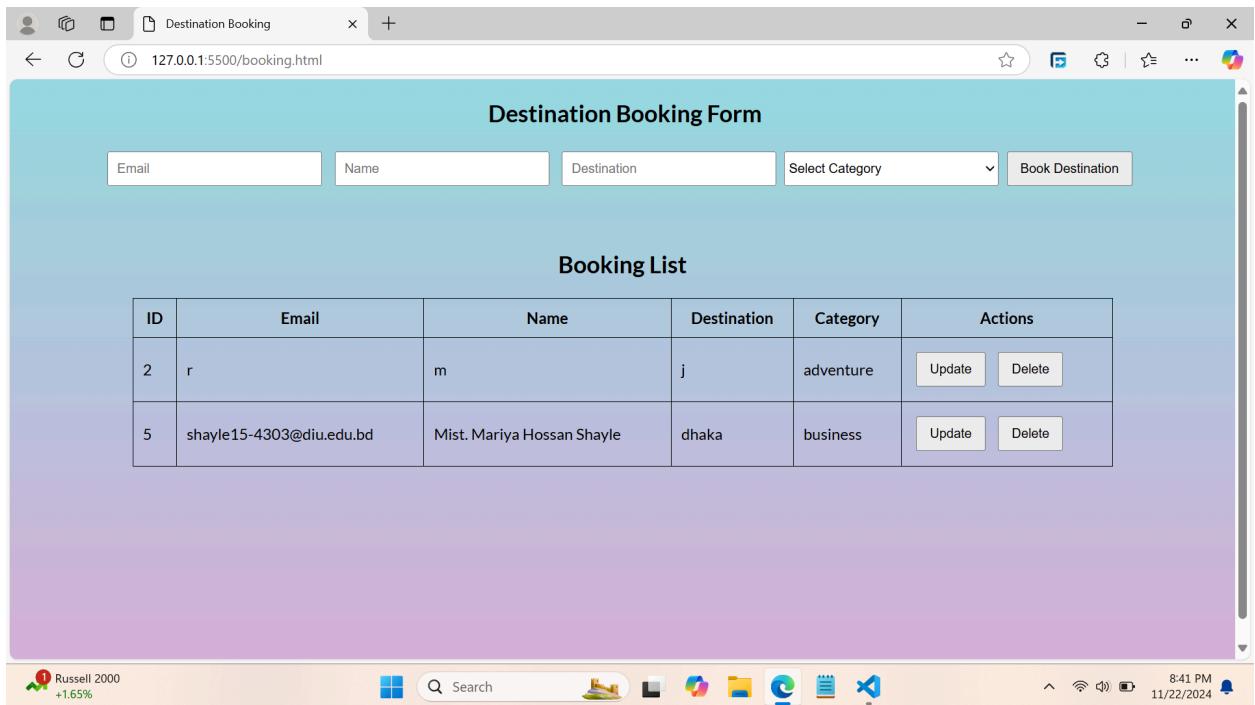


Figure 3: Booking Form - The interface for users to select travel packages and provide payment details.

3.1.2 Hardware Interfaces

- Compatible with desktops, laptops, and mobile devices.

3.1.3 Software Interfaces

Integration with:

- Laravel (backend framework).
- MySQL (database).
- React.js (frontend framework).

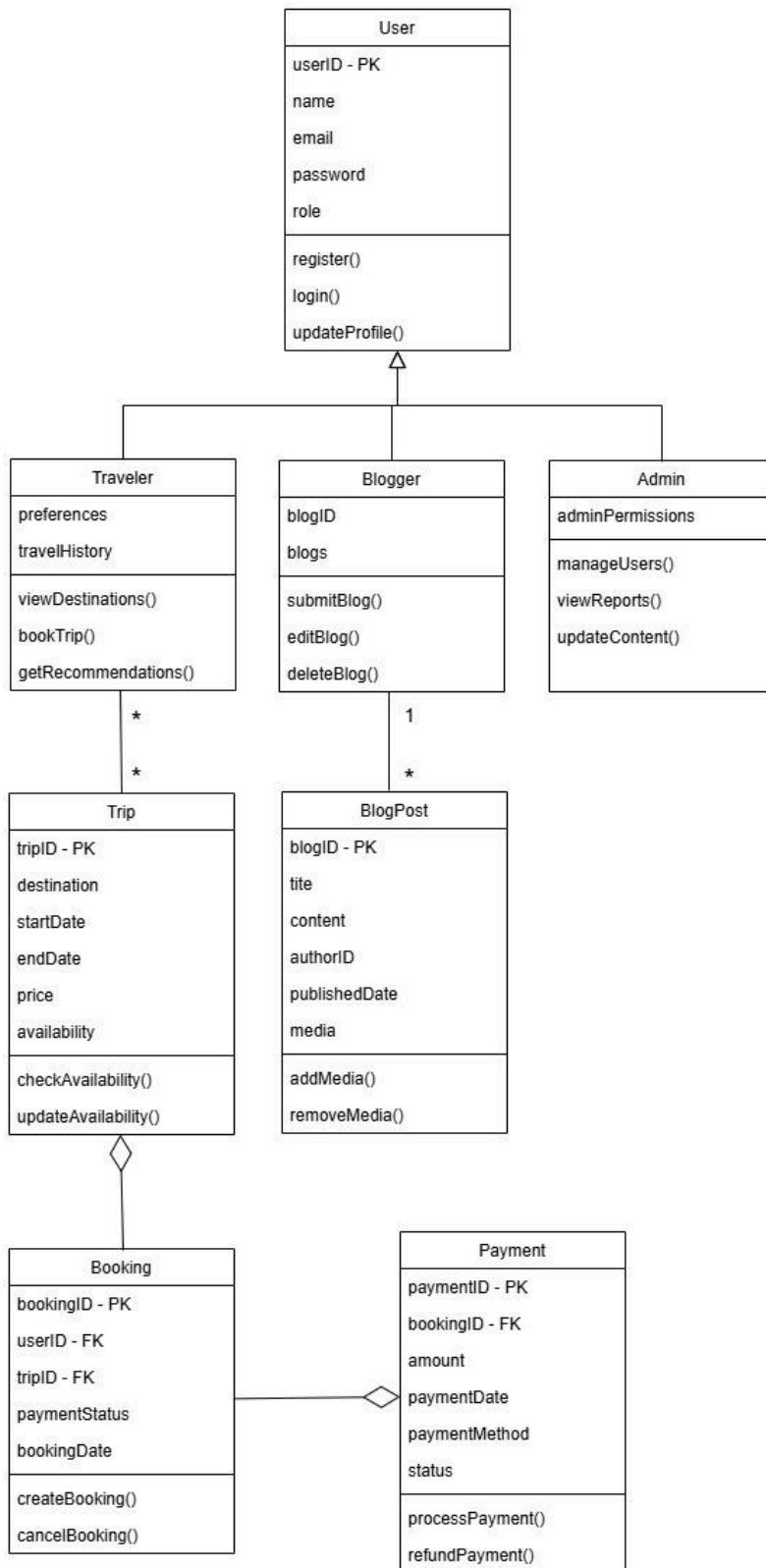


Figure 4: Class Diagram for Wanderer website.

This class diagram represents the object-oriented design of the Wanderer system. It captures the primary entities (User, Traveler, Admin, Booking, BlogPost, etc.) and their relationships, including inheritance and association. The diagram showcases how the system components interact to fulfill the system's functionalities, such as trip booking, blog submission, and payment processing.

3.2 Functional Requirements

Key functionalities include:

- Secure user authentication and role-based dashboards.
- End-to-end trip booking with payment processing.
- Blogging capabilities and content moderation.

Here are a few flow charts depicting the workflow of the system:

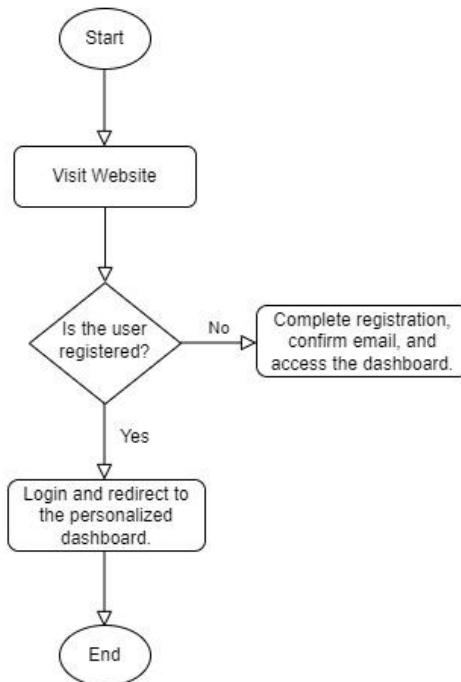


Figure 5: Login and registration workflow - illustrating the step-by-step process users follow to register and log in, including validations and system responses.

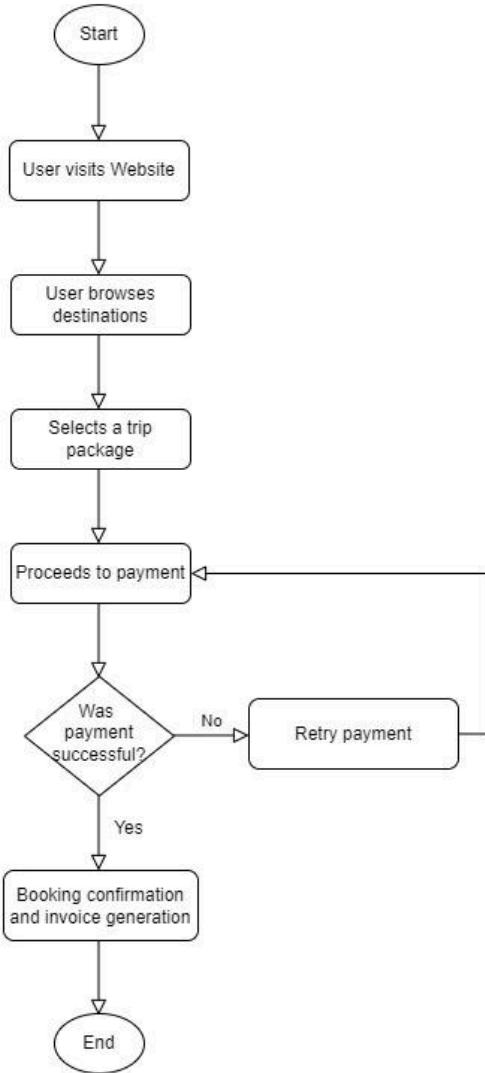


Figure 6: Booking and payment workflow - depicting the workflow for booking trips, covering destination selection, payment processing, and booking confirmation.

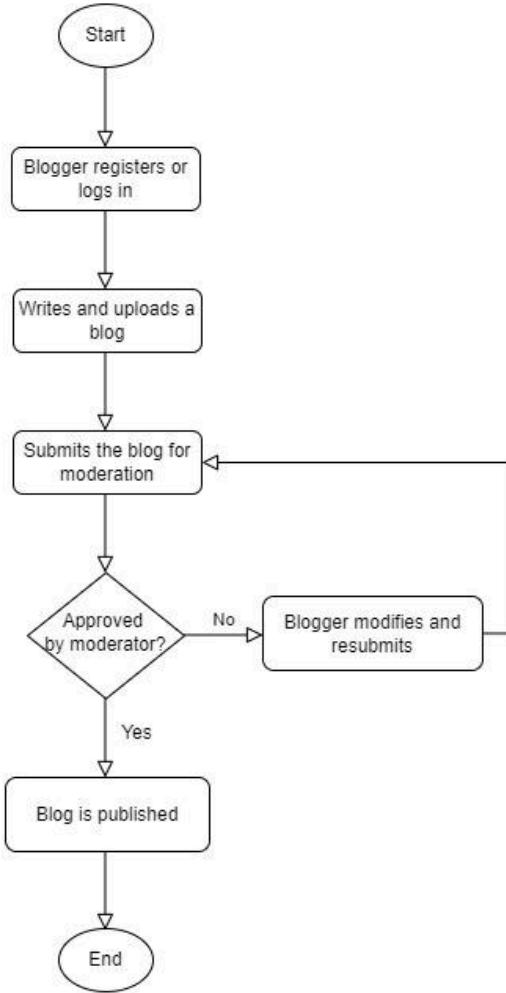


Figure 7: Blog publishing workflow - depicting the workflow for blog publishing, covering writing blog, submitting blog and getting approval.

3.3 Quality of Service Requirements

3.3.1 Performance

- Support up to 1,000 concurrent users.

3.3.2 Security

- Two-factor authentication for user login.
- End-to-end encryption for sensitive data.

3.3.3 Reliability

- Ensure 99.9% system uptime.

3.3.4 Availability

- Accessible 24/7 via major browsers.

3.4 Compliance

- Adherence to PCI DSS standards for payment security.
- Compliance with local data protection regulations (e.g., GDPR).

3.5 Design and Implementation

3.5.1 Installation

- Cloud-hosted deployment for scalability.

3.5.6 Cost

- Total project cost: 100,000 BDT (inclusive of VAT).

3.5.7 Deadline

- Completion by October 2024.

3.5.8 Development Phases and Timeline

The development of Wanderer is structured into sequential phases to ensure smooth execution and timely delivery. The Gantt chart below provides a visual timeline of the project, highlighting task durations, dependencies, and key milestones.

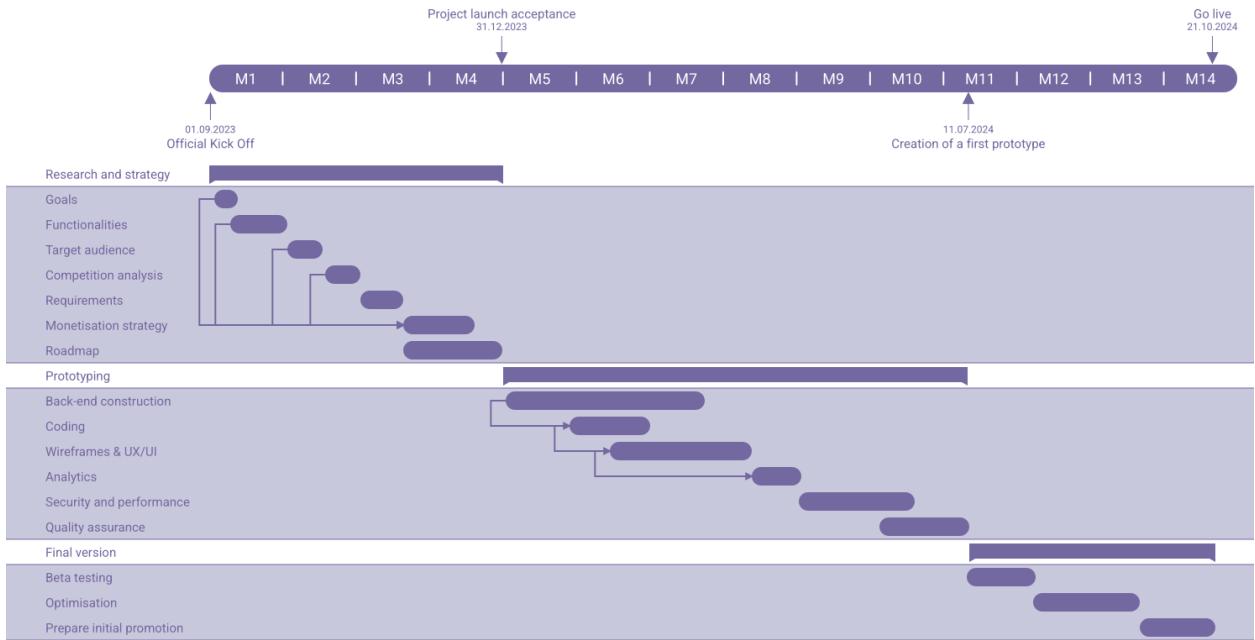


Figure 8: Gantt Chart of development process

4. Verification

Verification will include:

- Unit testing to validate individual modules.
- Integration testing to ensure smooth interaction between components.
- System testing for end-to-end functionality.