**Software Requirements Specification**

**for**

UrbanKart

ONLINE SHOPPING WEBSITE

**Version 1.0**

**Prepared by Project Team 15**

|  |  |
| --- | --- |
| **Name** | **PRN** |
| Venkatesh Pujari | 230940520098 |
| Pushpak Gulkari | 230940320086 |
| Rushikesh Nimje | 230940320093 |
| Sabiha Naikawadi | 230940320094 |
| Aniket Pardeshi | 230940520007 |

**Centre for Development of Advanced Computing**

Raintree Marg, Near Bharati Vidyapeeth, Opp. Kharghar Railway Station, Sector 7, CBD Belapur, Navi Mumbai - 400 614 - Maharashtra (India)

Phone: +91-22-27565303

Fax: +91-22-2756-0004

# Table of Contents

[Table of Contents i](#_TOC_250035)

[Revision History ii](#_TOC_250034)

1. [Introduction 3](#_TOC_250033)
   1. [Purpose 3](#_TOC_250032)
   2. [Intended Audience and Reading Suggestions 3](#_TOC_250031)
   3. [Project Scope 3](#_TOC_250030)
   4. [References 4](#_TOC_250029)
2. [Overall Description](#_TOC_250028) 5
   1. [Product Perspective](#_TOC_250027) 5
   2. [Product Features](#_TOC_250026) 5
   3. [User Classes and Characteristics](#_TOC_250025) 6
   4. [Operating Environment](#_TOC_250024) 6
   5. [Design and Implementation Constraints](#_TOC_250023) 7
   6. [User Documentation](#_TOC_250022) 7
   7. [Assumptions and Dependencies](#_TOC_250021) 8
3. [System Features](#_TOC_250020) 9
   1. [User Login](#_TOC_250019) 9
   2. [Dashboard with kanban board](#_TOC_250018) 9
   3. [Admin Dashboard](#_TOC_250017) 10
   4. [Admin Role](#_TOC_250016) 10
4. [External Interface Requirements](#_TOC_250011) 12
   1. [User Interfaces 1](#_TOC_250010)2
   2. [Hardware Interfaces 1](#_TOC_250009)2
   3. [Software Interfaces 1](#_TOC_250008)2
   4. [Communications Interfaces 1](#_TOC_250007)3
5. [Other Nonfunctional Requirements 1](#_TOC_250006)4
   1. [Performance Requirements 1](#_TOC_250005)4
   2. [Safety Requirements 1](#_TOC_250004)4
   3. [Security Requirements 1](#_TOC_250003)4
   4. [Software Quality Attributes 1](#_TOC_250002)4

[Appendix A: Glossary 1](#_TOC_250001)5

[Appendix B: Analysis Models 1](#_TOC_250000)6

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to delineate the specifications for UrbanKart, an E-commerce platform facilitating online shopping. UrbanKart aims to provide a seamless shopping experience for both users and administrators. This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements of the UrbanKart system. It encompasses the functionalities such as user and admin login procedures, product browsing, cart management, checkout process, feedback mechanism, and administrative tasks such as product and category management, user management, and feedback handling. Additionally, the document delineates the performance benchmarks and security measures necessary for the effective operation of the system.

## Intended Audience and Reading Suggestions

The UrbanKart SRS document is intended for a diverse audience, encompassing developers, users, administrators, and stakeholders involved in the project.

**For Developers:**

The SRS document serves as a foundational guide for developers involved in the design, implementation, and maintenance of UrbanKart. It provides a detailed overview of functional and non-functional requirements, performance benchmarks, and security considerations. Developers should thoroughly review the document to ensure alignment with project goals and stakeholder expectations.

**For Users:**

Users, including shoppers and potential customers, can refer to this document to understand the features and functionalities offered by UrbanKart. It provides insights into the user experience, including account management, product browsing, shopping cart operations, and feedback submission. Users are encouraged to familiarize themselves with the document to optimize their interaction with the platform.

**For Administrators:**

Administrators responsible for managing the UrbanKart platform should refer to this document to comprehend administrative functionalities, including product and category management, user administration, and feedback handling. The document offers guidance on system configuration and management, ensuring efficient operation and user satisfaction.

**For Stakeholders:**

Stakeholders, including educational institutions and decision-makers, should review the SRS document to assess the suitability of UrbanKart for their requirements. It provides a comprehensive overview of the system's capabilities, goals, and objectives, enabling informed decision-making regarding project adoption and implementation.

## Project Scope

The scope of the UrbanKart project delineates the boundaries and objectives of the system, detailing its intended functionalities and limitations. The scope encompasses the following:

* Development of an E-commerce platform enabling users to browse, search, and purchase products online.
* Implementation of user authentication and authorization mechanisms to ensure secure access to the platform, with distinct roles for users and administrators.
* Integration of a product management system allowing administrators to add, update, and delete products and categories.
* Implementation of a user-friendly interface for seamless navigation and interaction with the platform, facilitating smooth shopping experiences.
* Incorporation of a cart management system enabling users to add products to their carts, view cart contents, and proceed to checkout for purchases.
* Deployment of a feedback mechanism allowing users to provide comments and ratings for products and overall website experience.
* Integration of automatic notification systems to keep users and administrators informed about order statuses, account activities, and system updates.
* Additional features such as search functionality, product recommendations, and order history tracking to enhance user experience and satisfaction.
* Development of a secure system architecture to safeguard user data, including personal information, payment details, and order history.
* Deployment of the platform on cloud infrastructure, ensuring scalability, availability, and easy access for users.

## References

Following is the list of websites we referred during the course of our project:

* + - https://getbootstrap.com/docs/5.1/getting-started/introduction/
    - https://reactjs.org/docs/getting-started.html
    - https://www.baeldung.com/
    - <https://www.w3schools.com/>
    - <https://docs.spring.io/springdata/jpa/docs/current/reference/html/#reference>
    - <https://javaee.github.io/javaee-spec/javadocs/>
    - https://javadoc.io/doc/org.springframework.data/spring-datajpa/latest/index.html

# Overall Description

## Product Perspective

UrbanKart is designed to address the evolving landscape of online shopping, catering to the needs and preferences of modern consumers. With the proliferation of E-commerce platforms and the growing demand for convenient and secure shopping experiences, UrbanKart seeks to provide a comprehensive solution for users seeking to shop online.

UrbanKart aims to streamline the online shopping process by offering a centralized platform where users can browse, search, and purchase products with ease. By leveraging modern technology and user-friendly interfaces, UrbanKart aims to enhance the overall shopping experience for users while also providing robust administrative tools for platform management.

The product perspective of UrbanKart is rooted in the following principles:

Addressing the challenges faced by traditional brick-and-mortar retail methods by providing a convenient and accessible online shopping platform.

Catering to the increasing demand for efficient and streamlined E-commerce solutions in today's digital landscape.

Providing a user-centric platform that prioritizes ease of use, security, and transparency throughout the shopping process.

Enhancing the overall shopping experience for users by offering a diverse range of products, personalized recommendations, and seamless checkout options.

Empowering administrators with robust tools for product management, user administration, and feedback handling to ensure smooth operation and customer satisfaction.

By aligning with these principles, UrbanKart aims to establish itself as a leading E-commerce platform, catering to the needs of modern consumers and providing a seamless shopping experience for all users.

## Product Features

Key Features and Requirements

1. User Authentication and Authorization:

* Enable users to log into the system with their credentials.
* Implement role-based access control to determine user permissions.

1. Product Browsing and Search:

* Allow users to browse products by category, brand, or keyword search.
* Implement filters and sorting options for enhanced product discovery.

1. Product Management:

* Enable administrators to add, edit, and delete products and categories.
* Support batch updates and bulk operations for efficient management.

1. Cart Management and Checkout:

* Allow users to add products to their carts, view cart contents, and proceed to checkout.
* Implement a secure checkout process with multiple payment options.

1. Feedback Mechanism:

* Provide users with the ability to leave feedback and ratings for products and overall shopping experience.
* Ensure real-time updates and notifications for submitted feedback.

1. Administrative Tools:

* Enable administrators to view and manage user accounts, including adding, editing, and deleting users.
* Implement role assignment and permission management for user accounts.

1. Order Management:

* Allow administrators to view and manage orders, including order status updates and order history tracking.
* Support order processing and fulfillment workflows for efficient order management.

1. Notification System:

* Implement a notification system to alert users about order updates, promotions, and other relevant information.
* Ensure timely delivery of notifications via email, SMS, or in-app notifications.

1. Reporting and Analytics:

* Provide administrators with tools to generate reports and analyze sales data, customer trends, and product performance.
* Support customizable reporting dashboards and data visualization tools.

1. Data Security and Privacy:

* Implement encryption protocols to secure user data during storage and transmission.
* Comply with relevant privacy regulations, such as GDPR or CCPA, to ensure user privacy and data protection.

## User Classes and Characteristics

UrbanKart caters to three main user classes, each with distinct characteristics and requirements:

**1. Customers (Users):**

- Characteristics: Customers are the primary users of the platform, utilizing it for browsing, purchasing, and managing their shopping activities.

- Requirements: Customers should be able to register and log into the system, browse products, add items to their cart, proceed to checkout, and manage their account settings. They should also have access to features such as order tracking, order history, and feedback submission.

**2. Administrators:**

- Characteristics: Administrators oversee the management and operation of the UrbanKart platform, ensuring its smooth functioning and adherence to business objectives.

- Requirements: Administrators should have privileged access to administrative tools and functionalities, including user management, product management, order management, and reporting and analytics. They should be able to monitor system activities, manage user accounts, and address any issues that may arise.

**3. Code Reviewers (Optional for UrbanKart):**

- Characteristics: Code reviewers play a vital role in assessing the quality and integrity of assignments submitted by students.

- Requirements: If applicable to UrbanKart's context, code reviewers should have access to assignments submitted by users, the ability to review and provide feedback on submissions, and the authority to approve or reject assignments based on predefined criteria.

Each user class has specific functionalities and permissions tailored to their roles within the UrbanKart ecosystem, ensuring a seamless and efficient user experience for all stakeholders..

## 2.4 Operating Environment

The operating environment for the project consists of the following hardware and software components:

### Hardware:

* + - A machine with at least 8GB of RAM and a fast processor, such as Intel Core i5 or higher, to ensure smooth and efficient execution of the project.

### Software:

* + - ReactJS for the frontend development.
    - Spring Boot for the backend development
    - MySQL for the database management.

### Other Applications:

* + - Code editor (such as Visual Studio Code, Eclipse)
    - Git version control software
    - Command line interface (CLI) or terminal
    - A browser for testing the application.
    - Postman for testing APIs.
    - MySQL Workbench or another database management tool

## Design and Implementation Constraints

UrbanKart operates within specific constraints that may impact its design and implementation:

**1. Language Limitation:**

The user interface is available only in English, with no support for other languages. Users who prefer languages other than English may encounter limitations in accessibility and usability.

**2. Frontend-Backend Compatibility:**

Compatibility issues may arise when integrating the frontend with two separate backend systems. Synchronization and data exchange between the frontend and backends may present challenges, potentially affecting the overall functionality and user experience.

**3. Parallel Operation Overload:**

Parallel operations, such as concurrent user activities or simultaneous backend processes, may strain system resources and lead to performance issues. Overloading the system with parallel operations may result in degraded performance and reduced system responsiveness.

**4. HTTPS Limitation:**

UrbanKart is limited to HTTPS (Hypertext Transfer Protocol Secure) for secure communication between clients and servers. This constraint ensures data security and privacy but may limit compatibility with older or unsupported HTTP-only protocols.

## User Documentation

User documentation for UrbanKart is available in the application's Help menu. It offers detailed guidance on how to use the application's features and modules. Additionally, this SRS document serves as a comprehensive guide, providing all necessary information about UrbanKart's functionalities, users, software, and hardware requirements.

## 2.7 Assumptions and Dependencies

### Assumptions:

### • Users are proficient in the English language, as the user interface is available only in English.

### • Compatibility issues may arise when integrating the frontend with multiple backend systems.

### • Concurrent operations may overload the system and affect its smooth functioning.

### • UrbanKart operates exclusively over HTTPS for secure communication between clients and servers.

### Dependencies:

• UrbanKart relies on a functional email service to send notifications to users.

• Access to the GitHub API is necessary to verify project URLs and retrieve branch information.

• An active internet connection is required to access the UrbanKart application and receive notifications.

• Deployment on AWS infrastructure is necessary for hosting UrbanKart.

• UrbanKart relies on components such as ReactJS, Spring Boot, and MySQL for its functionality.

# System Features

* 1. **User Login**

**3.1.1 Description and Priority**

• Description: Users, including students and administrators, can log in to the UrbanKart application using their unique credentials.

• Priority: High

**3.1.2 Stimulus/Response Sequences**

• Users navigate to the app's homepage and click on the "Login" button.

• They enter their username and password and submit the form.

• If the credentials are valid, the user is redirected to their respective dashboard.

**3.1.3 Functional Requirements**

• User Authentication: Verify the user's credentials against the data stored in the application's database to authenticate their identity.

• Session Management: Maintain user sessions and implement session timeout mechanisms to ensure security and privacy.

• Data Encryption: Encrypt user passwords before storing them in the database to enhance data security and protect sensitive information.

* 1. **Dashboard**

**3.2.1 Description and Priority**

• Description: Users, particularly students, will have access to a dashboard where they can view a list of their current and past assignments, along with their due dates and statuses.

• Priority: High

**3.2.2 Stimulus/Response Sequences**

• Upon logging in, users are directed to their dashboard.

• The dashboard displays a comprehensive list of the user's assignments, including relevant details such as title, due date, and status.

**3.2.3 Functional Requirements**

• Display Assignments: Present users with a list of their assignments, showcasing essential information such as title, due date, and status.

• Update Assignment Status: Automatically update the status of assignments based on the submission status, ensuring accuracy and real-time tracking.

• Sort by Due Date: Allow users to sort assignments based on due date, facilitating better organization and prioritization.

**3.3 Admin Dashboard**

**3.3.1 Description and Priority**

• Description: The Admin Dashboard serves as the central control hub for system administrators. It offers a comprehensive overview of all projects, including their statuses, and enables administrators to create new projects and assign managers.

• Priority: High

**3.3.2 Stimulus/Response Sequences**

• Upon logging in and selecting the Admin role, the user is directed to the Admin Dashboard.

Admins can view a list of all projects, their current statuses, and associated managers.

Admins can click on a project to access detailed project information.

**3.3.3 Functional Requirements**

Display Project Information:

• Show a list of all projects with details such as project name, status, and associated manager.

• Provide the ability to click on a project for detailed information.

**3.4 Admin Role**

**3.4.1 Description and Priority**

• Description: The admin role is pivotal in managing the UrbanKart system. Admins are tasked with overseeing system operations, creating and managing user accounts, and ensuring the system's overall functionality.

• Priority: High

**3.4.2 Stimulus/Response Sequences**

• Admins log in to the UrbanKart application using their unique credentials.

• Upon successful login, admins can perform various tasks, including creating user accounts for new students, adding or removing code reviewers, and managing assignments and reviews.

• The system responds by presenting relevant information and providing admin privileges to execute necessary changes.

**3.4.3 Functional Requirements**

• Admin Login:

Admins must be able to log in using their unique credentials to access the admin dashboard.

• User Account Management:

Admins should have the capability to create, update, and delete user accounts, including those of students and code reviewers.

• Reports and Analytics:

Admins should be provided with access to reports and analytics generated by the system, allowing them to monitor system performance and user activity.

• System Monitoring and Adjustment:

Admins must be able to monitor the overall system functionality and make necessary adjustments to ensure smooth operation and optimal performance.

# External Interface Requirements

## User Interfaces

The UrbanKart system shall feature intuitive and straightforward user interfaces for both users and administrators. These interfaces aim to facilitate seamless navigation and efficient interaction with the system's functionalities. Key interface components include:

• User Interface:

The user interface for UrbanKart will be designed to be user-friendly and intuitive, allowing users to easily navigate through pages, browse products, add items to cart, and proceed with the checkout process. Additionally, users will be able to provide feedback on products and the website through a feedback form.

• Admin Interface:

The admin interface will provide administrators with easy access to manage user details, including user accounts, product categories, and feedback sections. Admins will have the capability to perform tasks such as adding, updating, and deleting user accounts, as well as viewing reports and analytics generated by the system.

### 4.2 Hardware Interfaces

### Server Side:

### • The UrbanKart web application will be hosted on a web server listening on the standard web port, port 80.

### Client Side:

### • Monitor Screen:

### The software will display information to users via the monitor screen, providing a visual interface for interaction.

### • Mouse:

### UrbanKart will interact with mouse movement and mouse buttons. The mouse will be used to activate areas for data input, command buttons, and select options from menus.

### • Keyboard:

### The software will interact with keyboard keystrokes. Users will input data into active areas of the interface using the keyboard.

### 4.3 Software Interfaces

### Server Side:

### • UrbanKart will utilize an Apache web server to handle all client requests and route them accordingly. Additionally, a centrally hosted MySQL database will manage and store system data.

### Client Side:

### • The UrbanKart application will be accessed through an operating system capable of running a modern web browser that supports JavaScript and HTML5. Users will interact with the application's frontend using a compatible web browser.

### 4.4 Communications Interfaces

### The UrbanKart system will utilize the HTTP or HTTPS protocols to facilitate communication between the client and server. These protocols will ensure secure and efficient data exchange between the UrbanKart application running on the client-side and the server hosting the application and database.

# Other Nonfunctional Requirements

## Performance Requirements

The performance requirements of the UrbanKart system will define the expected speed, reliability, scalability, and efficiency of the system. These requirements encompass various aspects, including:

• Response Time: The maximum response time for user actions, such as loading pages, adding items to the cart, and checking out, should be within acceptable limits to ensure a seamless user experience.

• Uptime Percentage: The expected uptime percentage of the system should be defined to ensure continuous availability for users. The system should strive for high availability to minimize downtime and disruptions to user access.

• Scalability: The system should be capable of handling increasing numbers of users and submissions without significant degradation in performance. It should scale efficiently to accommodate growing user demand and workload.

• Resource Usage: The resource usage of the system, including CPU, memory, and network bandwidth, should be optimized to ensure efficient utilization of hardware resources and minimize operational costs.

## Safety Requirements

To ensure a reliable and safe user experience, the UrbanKart system is designed with the following safety measures:

• Minimal Downtime: UrbanKart aims to minimize downtime to ensure timely access to project management activities. This ensures that users can rely on the system for their shopping needs without interruption.

• Intuitive and User-Friendly Interface: UrbanKart will feature an intuitive and user-friendly interface to reduce the likelihood of errors or misunderstandings during shopping activities. This helps in ensuring a smooth and hassle-free experience for users.

• Robust Error Handling Mechanisms: UrbanKart will implement robust error handling mechanisms to address unexpected situations effectively. This includes handling errors gracefully to prevent harm to users or loss of data, thereby enhancing the overall safety and reliability of the system.

## Security Requirements

## • Minimal Downtime:

## UrbanKart will strive to maintain minimal downtime to ensure that users, especially students, can submit assignments and receive feedback in a timely manner. This ensures uninterrupted access to the system's functionalities.

## • Intuitive and User-Friendly Interface:

## The UrbanKart user interface will be designed to be intuitive and user-friendly. This design approach aims to minimize errors or misunderstandings while users interact with the application, enhancing overall usability and safety.

## • Robust Error Handling Mechanisms:

## UrbanKart will incorporate robust error handling mechanisms to effectively manage unexpected situations. These mechanisms will prevent harm to users and mitigate the risk of data loss by handling errors gracefully and providing appropriate feedback to users.

## 5.4 Software Quality Attributes

To safeguard sensitive information such as student assignments, grades, and personal data, UrbanKart will implement the following security measures:

• Authentication: UrbanKart will employ a secure authentication system to ensure only authorized users can access the system.

• Authorization: An authorization mechanism will be implemented to define user permissions based on roles, determining the actions they can perform within the system.

• Encryption: UrbanKart will encrypt data both in transit and at rest to protect sensitive information from unauthorized access or interception.

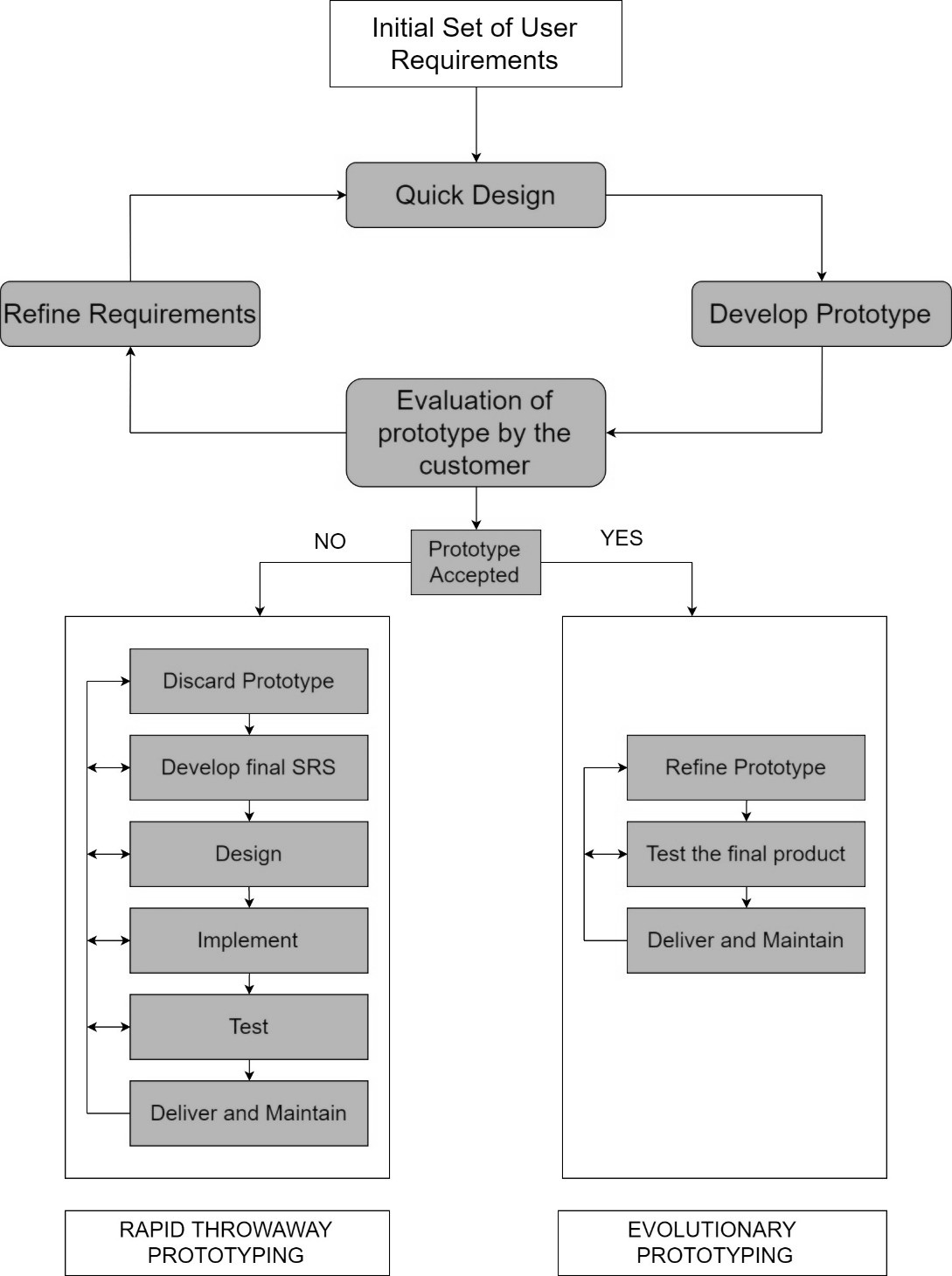
• Access Control: The system will incorporate access control mechanisms to regulate access to sensitive information, allowing administrators to set permissions for different users and roles.

# Appendix A: Glossary

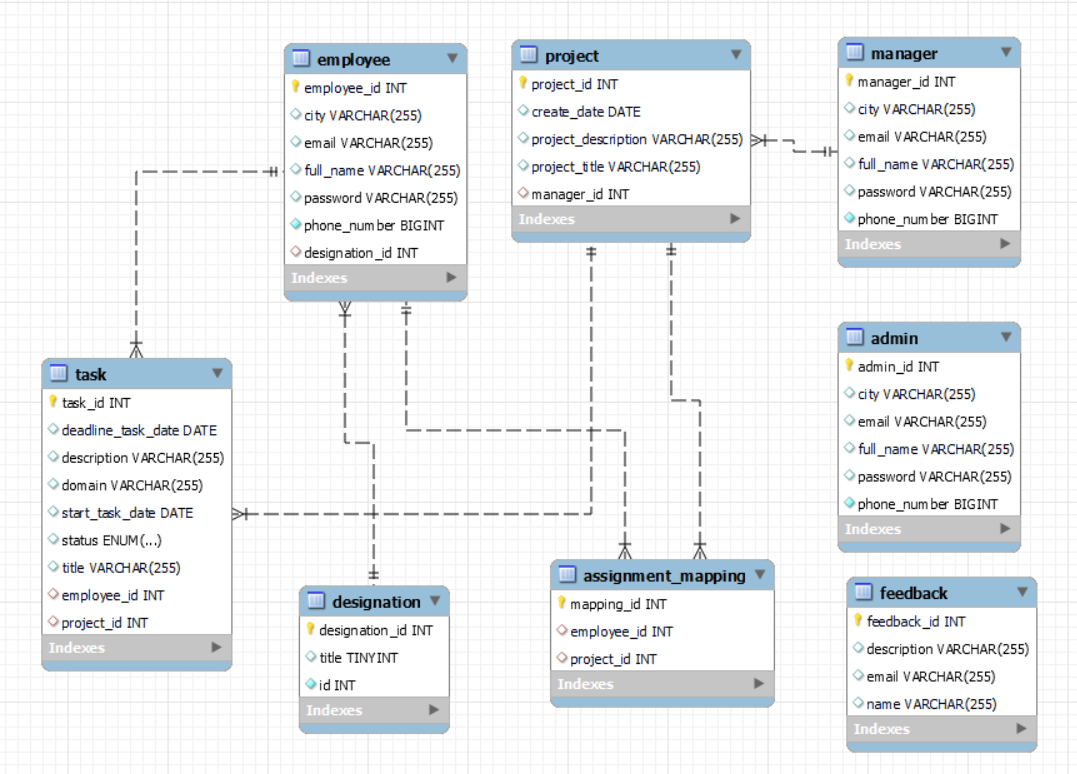
|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Abbreviation** | **Full Form** |
| 1. | API | Application Programming Interface |
| 2. | CLI | Command line Argument |
| 3. | GB | Gigabyte |
| 4. | HTML | Hypertext Markup Language |
| 5. | HTTP / HTTPS | Hypertext Transfer Protocol / Hypertext Transfer Protocol Secure |
| 6. | ID | Identification |
| 7. | JS | JavaScript |
| 8. | JWT | Java Web Token |
| 9. | OS | Operating System |
| 10. | RAM | Random Access Memory |
| 11. | SQL | Structured Query Language |
| 12. | SRS | Software Requirement Specification |
| 13. | URL | Uniform Resource Locator |
| 14. | ER | Entity Relationship |

# Appendix B: Analysis Models

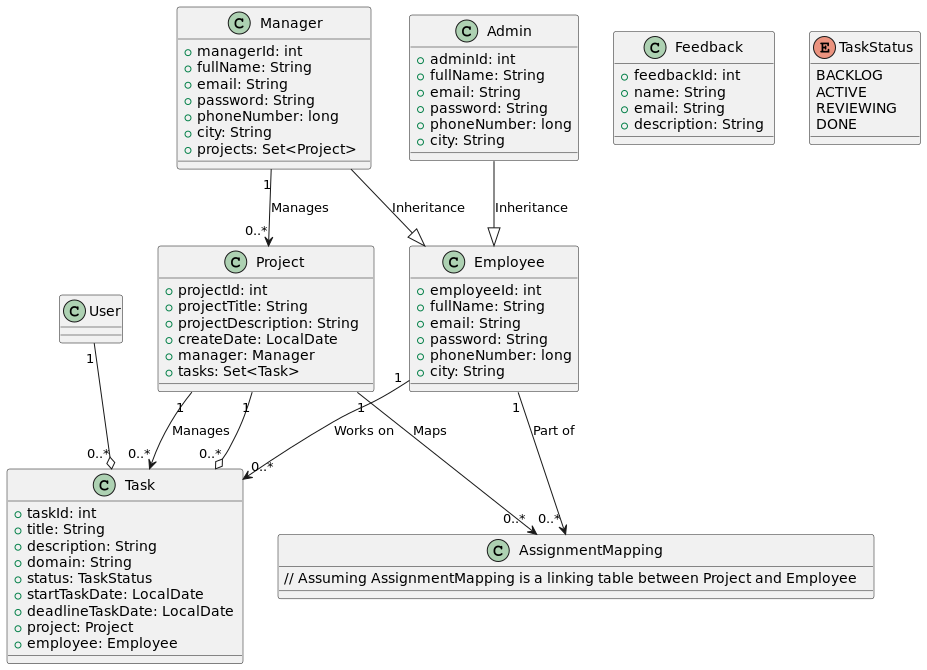
## Software Development Approach in Our System



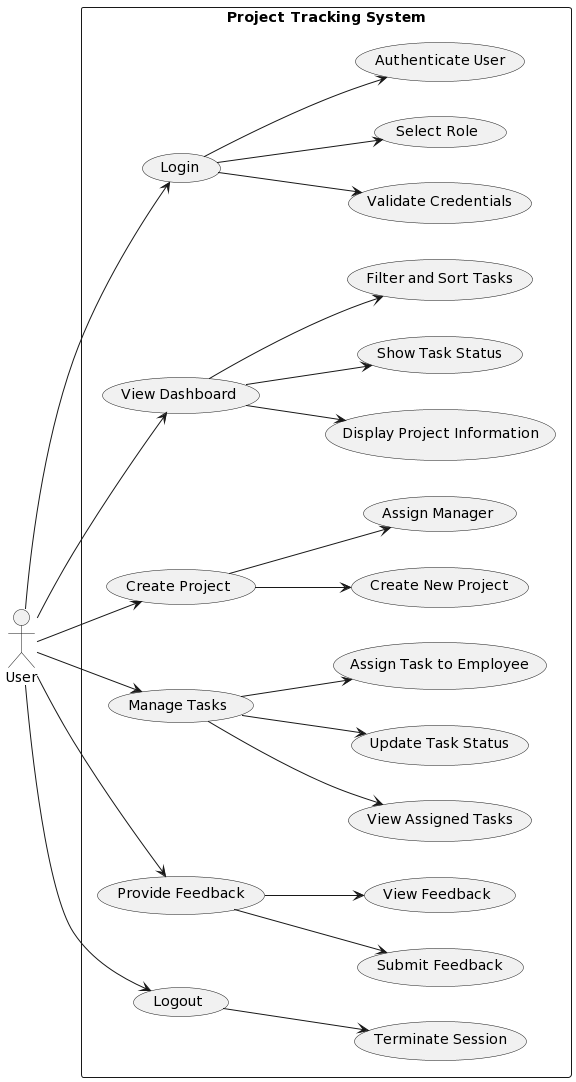
* **Class Diagram**

****

* **ER Diagram**

****

* **Use-Case Diagram**

****

* **Activity Diagram**

