Software Requirements Specification

for

PedalPal

Version 1.1

Prepared by

Group 4		Group Name: Bit Brewers
Raghav Manglik	220854	raghavkmanglik@gmail.com
Amogh Bhagwat	220288	amogh.2004b@gmail.com
Srishti Chandra	221088	chandra.srishti2403@gmail.com
Wadkar Srujan Nitin	221212	srujanwadkar@gmail.com
Anaswar K B	220138	anaswarkb013@gmail.com
Khushi Gupta	220531	khushi07g@gmail.com
Ananya Singh Baghel	220136	ananyabaghel2004@gmail.com
Pathe Nevish Ashok	220757	nevu.pathe1234@gmail.com
Debraj Kamakar	220329	debraj2003jsr@gmail.com
Kaneez Fatima	220496	kaneezfatimamehdi7@gmail.com

Course:	CS253
Mentor TA:	Mr. Bharat
Instructor:	Prof. Indranil Saha
Date:	January 25, 2024

Contents

1. Revisions

2. Introduction

2.1. Product Scope

Many individuals, from students to faculties, have embraced the lifestyle of cycling at the IITK campus. These cycles require timely maintenance and often get lost. A large number of lost cycles are never found by the owners and remain stacked like waste. Considering the temporary stay of students on campus, buying new cycles is not very cost-effective in such cases. In addition to that, during fests and other campus events, visitors from outside generally face problems in roaming around our huge campus.

This calls for a need for public cycle stands, from where the cycles can be rented using our software application. These cycle stands will be located at some of the most visited locations on campus, which can be accessed through our portal, making the transport easier. Specifying the location will inform them about the nearest stands along with the number of available cycles in it. It will also schedule maintenance on a regular basis, taking a record of the client's feedback. The app will keep track of the duration for which the cycle is used by the client and charge accordingly. We will also provide cycle booking facilities

2.2. Intended Audience and Document Overview

Software Developers who will design the software as per the requirements given in the document, in this case, the group members. **Project Managers** who will supervise the planning and execution of the software development procedure, in this case, the TAs and the course instructor. **Testers and approvers** who will perform a quality check of the designed software and give their feedback on the interface, areas of improvement, etc. **Users** will be the customers of the software, in this case, our entire campus residents and visitors.

Document Overview

Section 1: Revisions

This section contains information about the various versions that this document has gone through.

Section 2: Introduction

In this section, we provide some basic information that would be useful in reading the SRS, such as document conventions, abbreviations, etc. The reader may choose to skip the section if they are familiar with the basic terminologies. In any case, this section will serve as a helpful collection of information to clarify any confusion that may occur while reading the document.

Section 3: Overall Description

This section offers an overall view of the software system and its functionalities, assumptions, and dependencies. This will be a useful read for those seeking to familiarize themselves with the system at a quick glance. A reader is encouraged to read this part as it provides a good basis for understanding the next section of the SRS.

Section 4: Specific Requirements

This section contains detailed information about the software and explains its functions in detail through the use of numerous tree diagrams. This proves indispensable for end-users, clients, and developers alike, serving as a roadmap during the development phase and a user manual for end-users.

Section 5: Other Non-Functional Requirements

Important non-functional requirements are expounded here. This is of special importance to the developers of the software.

2.3. Definitions, Acronyms, and Abbreviations

SRS	Software Requirements Specification
DBMS	Database Management System
UI	User Interface
API	Application Programming Interface
GPS	Global Positioning System
CSS	Cascading Style Sheets
SQL	Structured Query Language
OTP	One Time Password
HTTPS	Hypertext Transfer Protocol (Secure)
Subscribed Users	Users who avail services as per a prepaid subscription agreement
Guest Users	Users who avail services as per a postpaid agreement
Hubs / Stands	Places where the cycles will be present
Ride time	Total time between unlocking and locking a cycle

2.4. Document Conventions

2.5. References and Acknowledgments

- We would also like to acknowledge the help of our TA, Mr. Bharat, and our course instructor, Prof. Indranil Saha for guiding us through the document, and providing a template for the Software Requirements Specification document.
- We utilised Figma to craft visually compelling graphs, effectively translating our ideas into a concise and impactful pictorial representation.
- We used the tool Circuito.io to capture the electronic circuit of the hardware subsystem.

3. Overall Description

3.1. Product Overview

Our product, PedalPal, is designed to enhance the cycling experience for IITK students through a convenient and efficient bicycle-sharing system on campus. Invaluable for students who have lost their bicycles or are facing cycling issues, PedalPal serves as a self-contained product, allowing easy bicycle issuance from strategically installed hubs across the university. Moreover, it caters to campus visitors, providing a seamless means to explore the campus on wheels.

The software streamlines the user interaction process by allowing users to book cycles for desired durations and receive real-time information about available cycles. PedalPal further enhances the experience by offering personalized details on the closest hub to the user's current location, ensuring convenient access. The user interface incorporates a history feature, allowing users to review past cycling sessions and track usage patterns. Thus providing a user-friendly solution and optimizing the cycling journey for both students and visitors on campus.

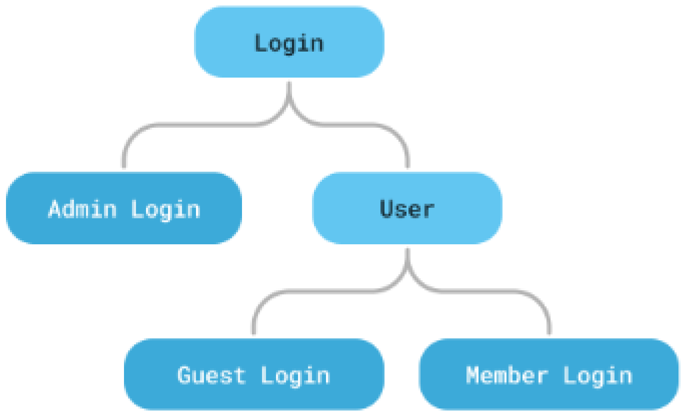
3.2. Product Functionality

- Administrative access using designated usernames and passwords
- Provision for users to register to avail services and subscribe for enhanced features
- Availability of an advance cycle booking system exclusively for subscribed users
- Real-time visibility of available cycles at each hub
- Visibility of hubs in proximity to any specified location
- User location tracking for personalized service

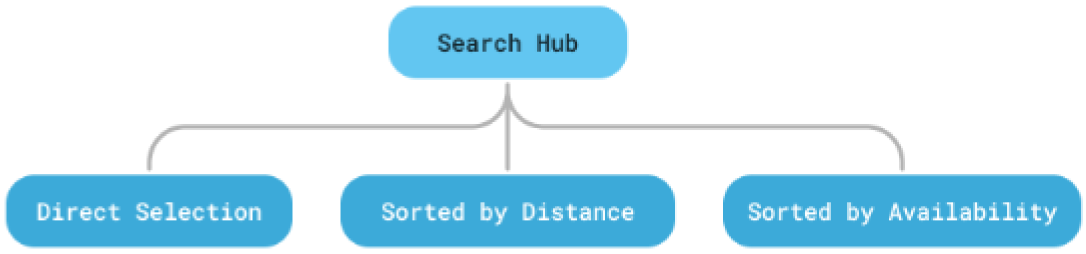
4. Specific Requirements

4.1. External Interface Requirements

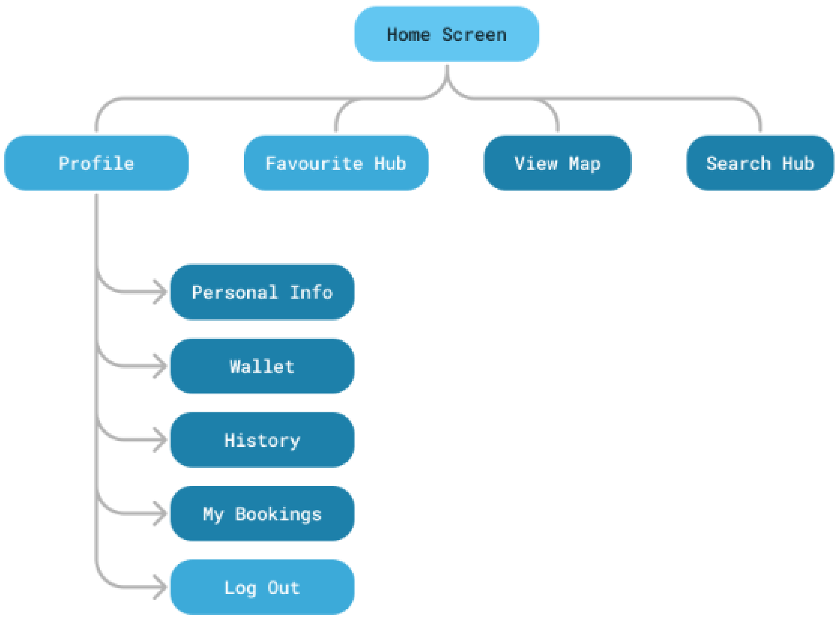
4.1.1. User Interfaces



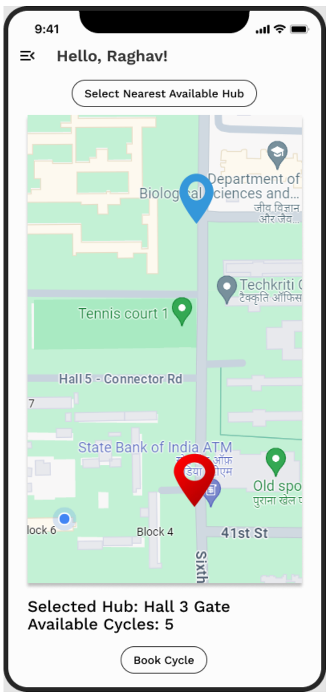
The application offers a login interface catering to both administrators and users, including guests and members. Users are required to register themselves and further undergo additional authentication processes.

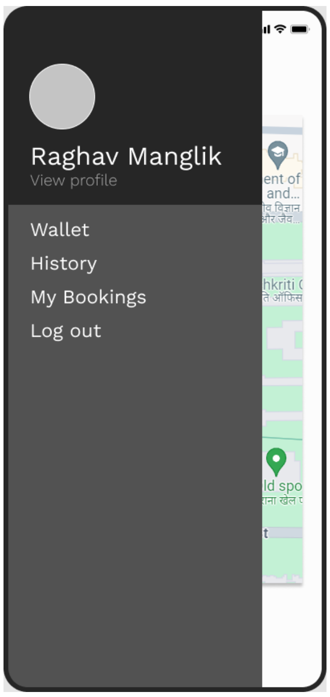


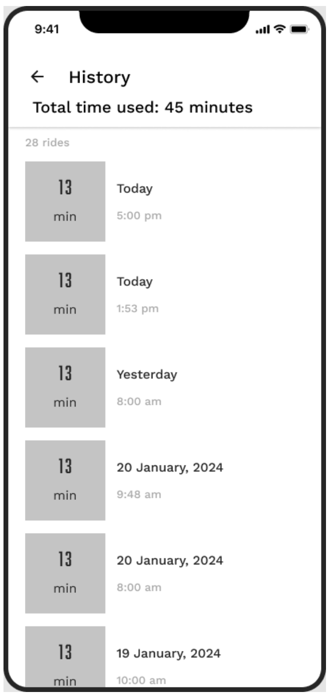
Hubs can be directly searched or sorted based on either distance or availability. The interface also provides a map view of the hubs, allowing users to select the most convenient option.



The home screen presents several options, such as a profile sidebar that features additional choices like Personal Info, Wallet, History, My Bookings, and a Log Out button. Additionally, the home screen displays the user's preferred hub along with functions for viewing the map and searching for hubs.






9:41
History
Total time used: 45 minutes
28 rides
13 min Today 5:00 pm
13 min Today 1:53 pm
13 min Yesterday 8:00 am
13 min 20 January, 2024 9:48 am
13 min 20 January, 2024 8:00 am
13 min 19 January, 2024 10:00 am

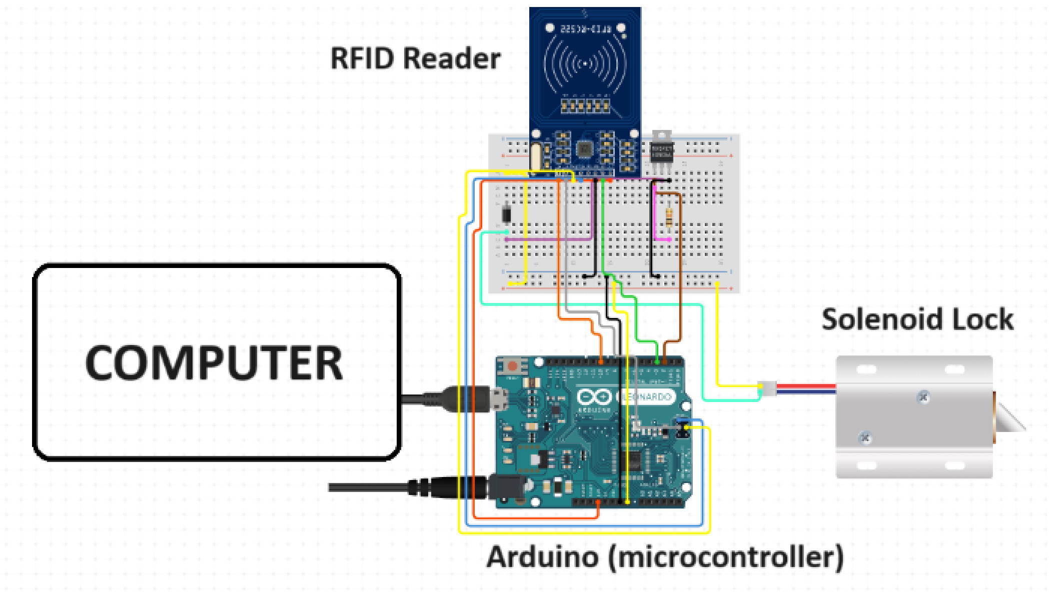
The main screen of the app will present a map view of the campus, with the hubs marked on it. The user can select a hub to view the number of available cycles at that hub. The user can also search for the nearest hub from their location.

The user can also view their booking history, which will contain details like the hub from which the cycle was booked, the time of booking, the duration of the ride, etc.

The home screen presents several options, such as a profile sidebar that features additional choices like Personal Info, Wallet, History, My Bookings, and a Log Out button. Additionally, the home screen displays the user's preferred hub along with functions for viewing the map and searching for hubs.

4.1.2. Hardware Interfaces

- **User devices**: The software would be available for use in devices with internet access, using which the users can directly book their cycles. This will include smartphones, tablets and laptops. Accessibility of the software on mobile devices like smartphones would make the process of cycle booking easy and handy for the user.
- **Cycle Locking mechanism**: The cycle locking mechanism would be made using solenoid locks present at the cycle hubs. The locks will be accessed by the server using an Arduino microcontroller, and its communication will be bridged using the Python `pyserial` library.
- **Keeping track of parked cycles using RFID**: Each bicycle will be equipped with an RFID tag, and RFID readers will be present at hubs along with each lock. This RFID-based system will keep track of which cycles are parked, enabling a quick and hassle-free cycle issuing and submission system. The unique identification of RFID tags will provide our users with a reliable and convenient solution for renting and returning cycles.



4.1.3. Software Interfaces

- The server-side components, including the database, must be hosted in a Linux-based operating system environment
- The client-side components must be functional on modern web browsers as well as their mobile versions, like Google Chrome, Safari, Mozilla Firefox, Microsoft Edge, and Brave
- The database management system used by us will be PostgreSQL. The following databases will be created:

  - The user database will store all users' data, including authentication details and personal information
  - The rides database will store all trips' data, including the duration and user associated with the trip
  - The hubs database will store all cycle hubs' data - number of available cycles and location of the hub
  - The cycles database will store all cycles' data - age, location, usability
  - The payments database will store all the information related to payments
  - The maintenance database will store the maintenance history of all cycles
  - The booking database will store all past and future bookings

4.2. Functional Requirements

4.2.1. Administrative access using designated usernames and passwords

- Usernames and passwords, known only to authorized users, authenticate their identity during login and grant them administrative access control for system management, including notifying users
- Administrative access allows users to authenticate or deactivate user accounts, including resetting passwords and managing user access.
- Administrative access allows users to update the inventory of available bicycles, add new bicycles to any operating hub, or remove bicycles that are no longer available
- Administrative access allows users to adjust rental and membership prices and see users with negative balances

4.2.2. Provision for users to avail services and subscribe for enhanced features

- The system allows users to create their profiles by providing essential information, password, and identification proof, IITK campus email for campus residents, and contact numbers for visitors
- After successful authentication, the system allows users to update their personal information, reset their passwords, and avail of bicycle rental services
- The system allows users to become members and avail of enhanced features by paying a monthly subscription fee

4.2.3. Availability of an advance cycle booking system for subscribed users

- Subscribed users gain exclusive access to a bicycle booking system, offering them early access to available bicycles
- Subscribed users can define their booking window by providing necessary information such as date, time, and location for their advanced bookings, allowing them to reserve bicycles for a specified period in advance by paying a booking fee charged according to the time for which the bicycle was booked

- The system offers flexibility in booking options, allowing subscribed users to modify or cancel their advanced bookings based on their changing plans

4.2.4. Real-time visibility of available cycles at each hub

- The system allows the display of real-time data on available bicycles at each hub
- Users can use the map interface integrated into the system, providing a visual representation of cycle hub locations and the number of available bicycles at each site
- The system automatically refreshes after a specific time period for real-time updates to ensure that users have the latest information on cycle availability

4.2.5. Visibility of hubs in proximity to any specified location

- The system allows users to input a specific address landmark or use their current location as the basis for finding nearby hubs
- The system, if given permission, utilizes location-based services (LBS), like GPS or Wi-Fi positioning, to determine the current location of the user or any specified location

4.2.6. Implementation of a scheduled maintenance system for the upkeep of cycles

- The system defines a regular maintenance schedule for each cycle, considering factors such as total usage time and customers' feedback
- The system automatically sends maintenance reminders to maintenance service providers
- The system provides customer support for handling emergency maintenance situations, such as breakdowns or safety-related issues

4.2.7. Customer feedback mechanism for maintenance or post-ride experiences

- The system allows users to report any issues with bicycles through the dedicated reporting system
- The system defines a feedback form where users can report any issues and give feedback by including fields for specific details, such as cycle ID, Hub location, and/or a description of the problem
- The system provides a rating system (5 stars rating system) for users to quickly express their overall satisfaction with the ride and the bicycle itself

4.2.8. Access to detailed analytics, including ride and payment history

- The system provides a user-friendly dashboard where users can access and review detailed analytics of their ride history, including
  - Total number of rides
  - Date and time of each ride
  - Source and destination of each ride
  - Duration of each ride

  - Cost charged for each ride
  - Average ride duration

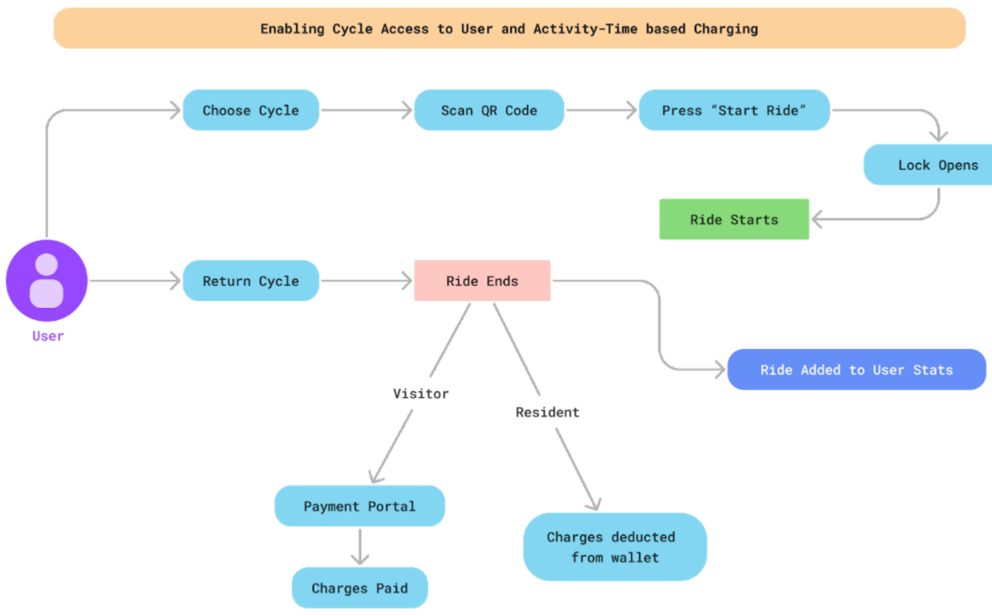
- The system provides a dashboard where users can access and review detailed analytics of wallet activities of users, including
  - Subscription charges (if any)
  - Current balance
  - Transcation history
  - Payment methods used

4.2.9. Implementation of a notification system

- The system sends regular notifications to users to inform them about upcoming maintenance activities or system updates that might affect bike availability
- The system sends reminders to users about upcoming subscription renewals

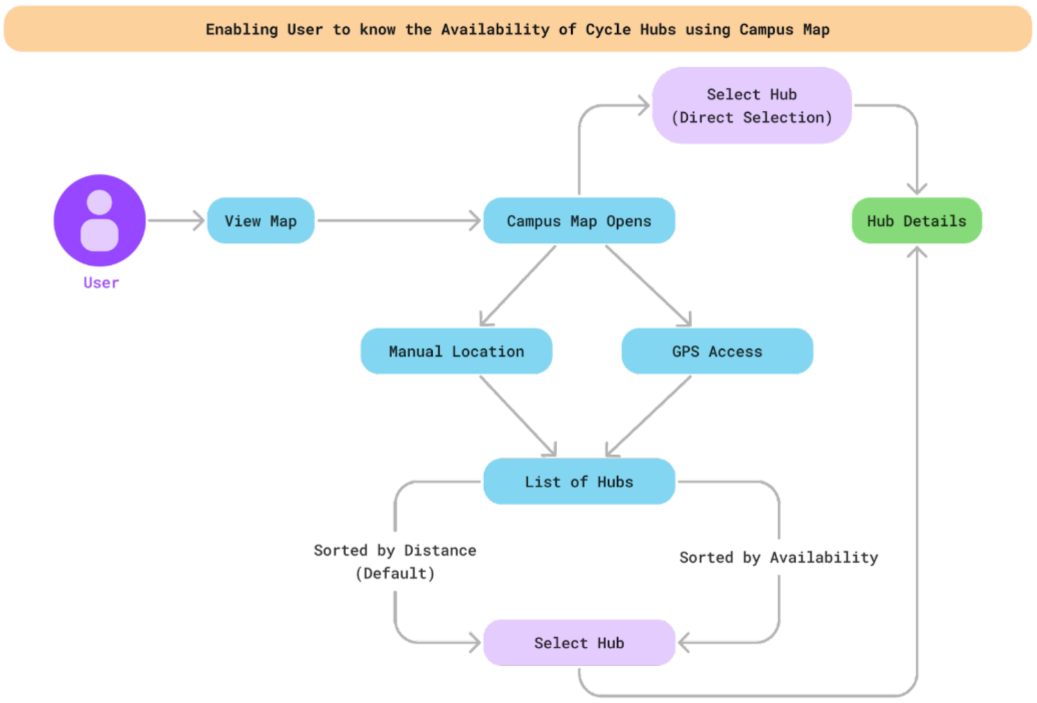
4.3. Use Case Model

4.3.1. Use Case 1



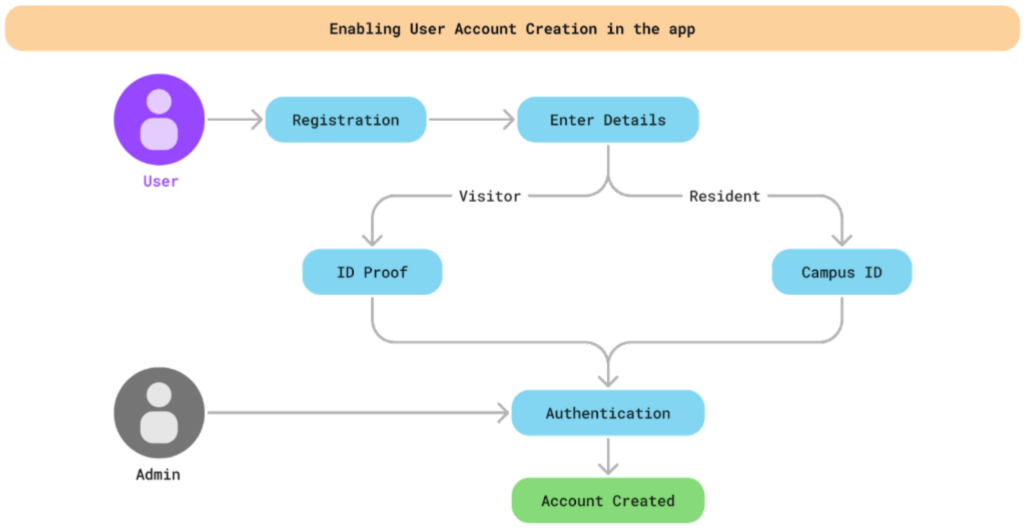
Author	Pathe Nevish Ashok
Purpose	To let the resident/visitor access cycles from a cycle hub and charge them based on their activity time
Requirements Traceability	User should be logged in to the app and should have access to the Internet
Priority	The priority of this use case is high as it is the feature of utmost importance and Is one of the major reasons for making this app
Pre Conditions	The user should be physically present at one of the cycle hubs on campus
Post Conditions	The user will get access to the desired cycle until they return it to another cycle hub, and they will be charged based on the time of the ride
Actors	The actors involved in this use case are the campus residents/visitors who want access to cycles to travel across the campus

4.3.2. Use Case 2


Enabling User to know the Availability of Cycle Hubs using Campus Map
User
View Map
Campus Map Opens
Select Hub (Direct Selection)
Hub Details
Manual Location
GPS Access
List of Hubs
Sorted by Distance (Default)
Sorted by Availability
Select Hub

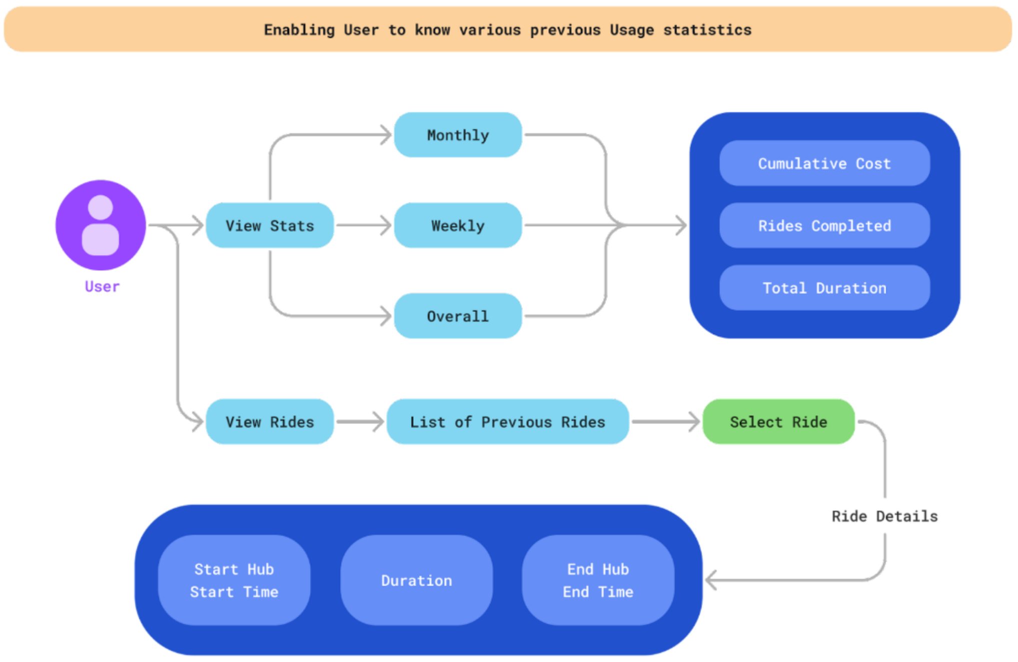
Author	Raghav Manglik, Srishti Chandra
Purpose	To let the user know the availability of cycles at strategically placed different cycle hubs with the help of a campus map
Requirements Traceability	The user should be logged in and should give access to the location if possible
Priority	The priority of this use case is medium as it provides a feature for increased convenience and is not highly important for campus residents in particular
Pre Conditions	The user should either give location access to the app or select a checkpoint manually from the app
Post Conditions	The user will get to know the availability of the cycles in the nearby/desired cycle hubs
Actors	The actors involved in this use case are the campus residents/visitors who want to know the availability of cycles on the campus

4.3.3. Use Case 3



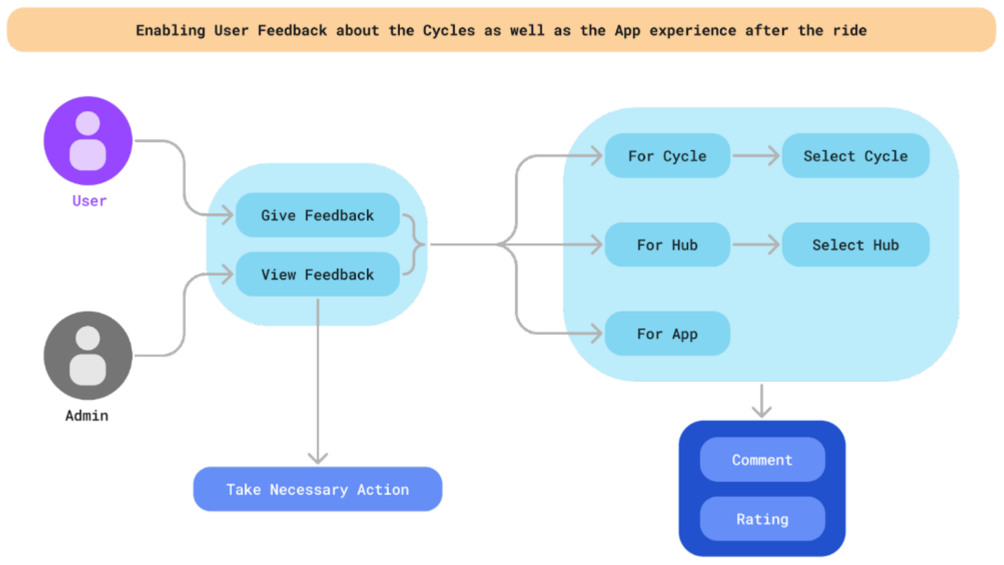
Author	Anaswar K B
Purpose	To let the user create an account on the portal
Requirements Traceability	The user should have a valid ID proof for the administrator to authenticate the user
Priority	The priority of this use case is high as if the user does not have an account, they can not use the app
Pre Conditions	The user should possess the required details along with an ID proof for the admin to authenticate
Post Conditions	An account will be created for the user on the app
Actors	The actors involved in this use case are the users who want to create an account on the app and the administrator who authenticates the user

4.3.4. Use Case 4



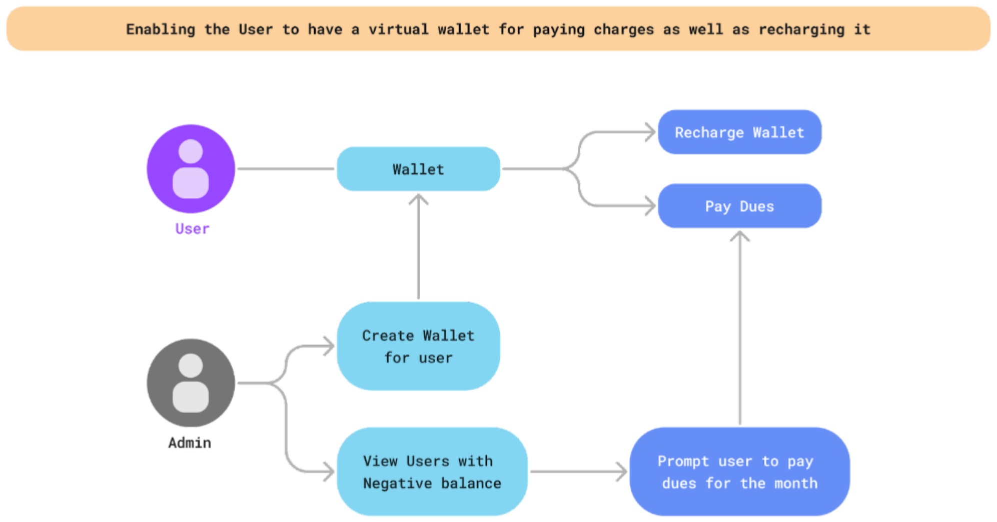
Author	Khushi Gupta, Ananya Singh Baghel, Kaneez Fatima
Purpose	To let the user know their previous usage statistics, such as the date of the trip, the duration for which the cycle was accessed, pickup hub, drop hub, etc.
Requirements Traceability	The user must have an account on the app
Priority	The priority of this use case is medium, as knowing the stats is also an additional feature of this app and not a major feature
Pre Conditions	The user must have completed at least one ride using their account on the application
Post Conditions	The user will receive the statistics of all the previous rides along with the option to get to know the statistics in a cumulative manner too
Actors	The actors involved in this use case are users who want to know their ride statistics

4.3.5. Use Case 5



Author	Wadkar Srujan Nitin
Purpose	To let the user have a virtual wallet from which the charges will be deducted and allow them to recharge their wallet
Requirements Traceability	The user must have an account on the app and also have a UPI/bank account to recharge the wallet
Priority	The priority of this use case is high as it is necessary to maintain a digital wallet to pay for the rides
Pre Conditions	The user must have created a wallet on the app for the expenses to be deducted from the user's wallet
Post Conditions	The wallet will show the remaining balance or the amount required to pay when a ride is completed, when the wallet is recharged, or whenever the user demands
Actors	The actors involved in this use case are users for whom the wallet is maintained and the admin

4.3.6. Use Case 6


Enabling the User to have a virtual wallet for paying charges as well as recharging it
User
Wallet
Recharge Wallet
Pay Dues
Admin
Create Wallet for user
View Users with Negative balance
Prompt user to pay dues for the month

Author	Debraj Karmakar, Amogh Bhagwat
Purpose	To let the user give feedback about the cycles as well as the app experience after the ride
Requirements Traceability	The user must have an account on the app and must have used the app for a sufficient amount of time
Priority	The priority of this use case is medium, as maintenance of cycles will not be fully dependent on feedback and comprehensive testing of the app will be done to ensure no bugs are found by the user
Pre Conditions	The user should have checked the condition of the cycle at the time of giving feedback about it as well as they should have used the app for a sufficient amount of time
Post Conditions	The app will record the feedback of the user and the admin will take actions by viewing the feedback if necessary
Actors	The actors involved in this use case are the users who will provide the feedback to the app and the admin who will view it

5. Other Non-Functional Requirements

5.1. Performance Requirements

The software should maintain a 99.9% uptime, allowing users to access cycle rental services without significant disruptions, excluding scheduled maintenance windows.

The system should support a minimum of 500 concurrent rental transactions per minute to ensure efficient and timely processing of user requests.The system should respond to user requests for cycle rentals within 3 seconds under normal operating conditions.The software should be designed with built-in fault tolerance mechanisms to mitigate the impact of hardware or software failures, ensuring uninterrupted service availability.

5.2. Safety and Security Requirements

- **User Authentication**: The system must employ robust user authentication mechanisms, such as strong password policies or multi-factor authentication, to verify the identity of users before granting access to rental services. Proper login mechanisms should be used to avoid hacking.
- **Data Encryption**: All sensitive user information, including personal details and payment data, must be encrypted during transmission to prevent unauthorized access and data breaches. Information should be securely transmitted to the server without any changes in information. Secure storage of passwords will be done after salting and hashing the passwords. In other words, the system admin can only verify if the password is correct but won't know what the password submitted by the user was. On maintenance complaints, the cycle's information will be sent to the admin. The complainant users' information is not sent to the server.
- **Secure Transactions**: Payment transactions within the software must be conducted through secure channels. The API of a safe payment gateway, like Razorpay, will be used for the same. A notification will be sent to the concerned people whenever a transaction is added.
- **Physical Security**: Implement physical security measures for any hardware components of the system to prevent unauthorized access. The cycle locks will only be opened when the user scans the QR code on the cycle.

5.3. Software Quality Attributes

5.3.1. Usability

- **Intuitiveness**: The software should have a user-friendly interface, allowing users to navigate through the rental process without confusion easily.

- **Response Time**: Provide quick response times for user interactions, such as renting a cycle or viewing available options.

5.3.2. Availability

- In case of a server crash, the system state must be restored within two hours.
- **Performance Scalability**: The number of users will increase in large numbers during campus fests. The system availability needs to be enough for this purpose. Ensure the software can handle an increased load as the user base grows.

5.3.3. Reliability

- The MTTF (mean time to failure) shall be more than one week.
- The system must undergo extensive feature testing, load testing, and regression testing prior to release and/or deployment.
- The system should be reliable in giving correct results consistently.

5.3.4. Portability

We are using Django with bootstrap and inline CSS to design the front-end part of our application, thus our application is portable, responsive and can run on most of the modern web browsers.

6. Other Requirements

Appendices

A. Data Dictionary

B. Group Log