

---

# Software Requirements Specification

for

## IITK Cycling App

Version 1.0

Prepared by

**Group #: 4**

Avi kumar  
Het Hitendrakumar Patel  
Arpit Kumar Rai  
Mohd Umam  
Shubham kumar  
Nishi Mehta  
Abhishek Bansod  
Pratham Jain  
Divyansh Manakchand  
Kankariya  
Maurya Jadav

200229  
200440  
200190  
200595  
200966  
200645  
200022  
200712  
200352  
200567

**Group Name: INF\_LOOP**

kumara20@iitk.ac.in  
hetpatel20@iitk.ac.in  
arpitkr20@iitk.ac.in  
umam20@iitk.ac.in  
shuhamk20@iitk.ac.in  
nishim20@iitk.ac.in  
abansod20@iitk.ac.in  
pratham20@iitk.ac.in  
divyanshmk20@iitk.ac.in  
mauryaj20@iitk.ac.in

**Course:** CS253A

**Mentor TA:** Pinaki Chakraborty

**Date:** 30<sup>th</sup> January 2022

<b>CONTENTS</b>	<b>II</b>
<b>REVISIONS</b>	<b>II</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 PRODUCT SCOPE	1
1.2 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	1
1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	1
1.4 DOCUMENT CONVENTIONS	1
1.5 REFERENCES AND ACKNOWLEDGMENTS	2
<b>2 OVERALL DESCRIPTION</b>	<b>2</b>
2.1 PRODUCT OVERVIEW	2
2.2 PRODUCT FUNCTIONALITY	3
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS	3
2.4 ASSUMPTIONS AND DEPENDENCIES	3
<b>3 SPECIFIC REQUIREMENTS</b>	<b>4</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS	4
3.2 FUNCTIONAL REQUIREMENTS	4
3.3 USE CASE MODEL	5
<b>4 OTHER NON-FUNCTIONAL REQUIREMENTS</b>	<b>6</b>
4.1 PERFORMANCE REQUIREMENTS	6
4.2 SAFETY AND SECURITY REQUIREMENTS	6
4.3 SOFTWARE QUALITY ATTRIBUTES	6
<b>APPENDIX A – DATA DICTIONARY</b>	<b>8</b>
<b>APPENDIX B - GROUP LOG</b>	<b>9</b>

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Version 1.0	INF_LOOP	First version of IITK Cycling App.	01/02/2022

# 1 Introduction

## 1.1 Product Scope

IITK Cycling App is a web-based application to accelerate and ease the pre-existing cycle-renting environment for students on campus. It provides tools through which customers can reserve available cycles online prior to their expected pick-up date or time and purchase cycles online from the already existing cycle dealers at the campus.

Such an online facility will facilitate the transport requirements of students and reduce the problems faced while availing of such facilities in offline modes, such as inconsistent prices, knowledge of the availability of cycles at different locations and offline payment hassles.

## 1.2 Intended Audience and Document Overview

While the software requirement specification (SRS) document is written for a more general audience, this document in whole is intended for individuals directly involved in the development of this project. This includes software developers, project consultants, and team managers. This document need not be read sequentially; A general user is encouraged to jump to any section they find relevant.

The first part of the document offers a summary of this project- including goals and objectives, project scope, general system details, and some major constraints associated with the intended platform. Furthermore, Marketing staff may want to jump to sections 2.1-2.2; User manual writers and testers may jump to sections 2.1-2.4, 3.1, 4.1-4.2. Interested investors and end-users are recommended to jump to sections 2.1-2.2.

## 1.3 Definitions, Acronyms and Abbreviations

1. Admin: Superuser who may be the IITK administration in our case or some other firm which requests the software in case of scaling.
2. Campus: Campus refers to the Area under the maintenance of IIT Kanpur.
3. Dealer: Dealer refers to the Customer who shall be interacting with the Dealer interface to give out their bicycles for rent/sale.
4. Stand: Stand refers to a Bicycle stand from where a User can purchase or rent
5. User: User refers to the Customer who shall be interacting with the User interface to rent or purchase a bicycle.

## 1.4 Document Conventions

- Formatting Standards
  - o This document follows the Standard IEEE formatting requirements<sup>1</sup> for the Formatting front. Some of the often used are:
    - Usage of Arial font, with font size 11 throughout the document for text purposes.
    - Usage of Bold feature to emphasize upon the keywords mentioned in the Specification.
    - Usage of Italics feature for commenting wherever applicable.
    - Superscripts for redirecting to footnotes.
- Reference Conventions
  - o All references used in the document are reported as footnotes on the respective footer sections and also in the Section 1.5 References and Acknowledgements.

## 1.5 References and Acknowledgments

Link to the SRS document that we referred while preparing our document -  
<https://fdocuments.in/document/car-rental-srs-document.html>

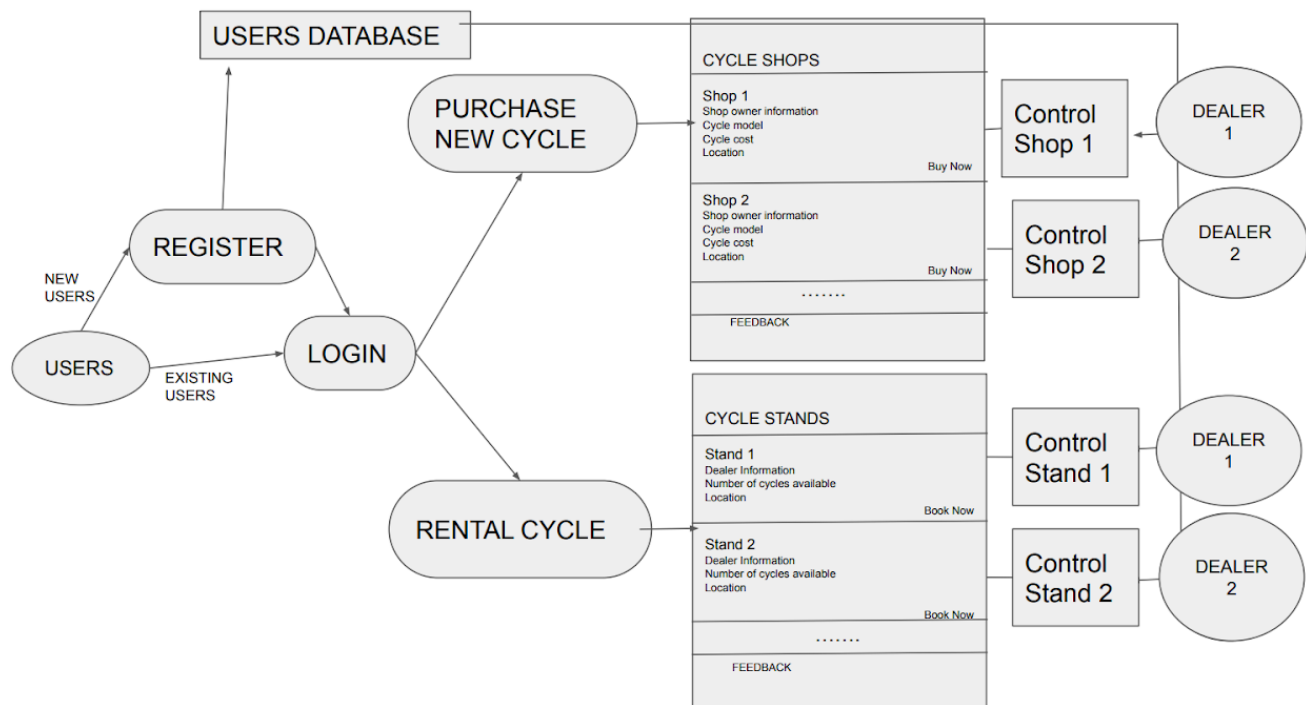
We express our gratitude to Prof. Indranil Saha and our teaching assistant Pinaki Chakraborty for their continuous guidance and assistance for helping us prepare this document.

## 2 Overall Description

### 2.1 Product Overview

The proposed Software is the First Edition of itself and shall function as a self-contained product. For the current version, the software shall comprise two interfaces, one for Users and one for Dealers. Each customer shall register with a username and password gateway to create an account with us.

- **User Interface:** Each of the user shall be provided with two options:
  - **Purchasing New Bicycle:** Through this option, the user shall be introduced to the available stocks of bicycles out for sale at different stands in the Campus and the respective features available for the same.
  - **Renting a Bicycle:** Via this option, the user shall be introduced to the available stocks of bicycles out for rental rides at different stands in the Campus and the respective features available for the same.
- **Dealer Interface:** Each of the Dealer shall be provided with the following features:
  - Dealers shall be given the feature to maintain the live status of bicycles at their stand i.e. they can add/delete cycles as the count changes after rental rides or selling/arrival of stock of bicycles meant for sale.



## 2.2 Product Functionality

1. Provide a bulleted list of all the major functions of the system

- Campus communities can take cycles on rent through their registered ID.
- Users can look for new cycle models and can buy them.
- Users can see the number of cycles available in every cycle stand.
- Cycle dealers can keep track of people who have taken a cycle .

## 2.3 Design and Implementation Constraints

Third Party softwares used :-

### 1. Firebase

- i) Firebase is a platform developed by Google for creating mobile and web applications.
- ii) Firebase provides a lot of functionalities which includes authorization, cloud ,hosting the website ,Database support.
- iii) It is being maintained and updated by google.

- iv) It can be easily used using javascript.
- 1. **Nodejs**
  - i) Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser
- 1. **React library**
  - i) React is a free and open-source front-end JavaScript library for building user interfaces based on UI components.
  - ii) It is maintained by Meta and a community of individual developers and companies.
- 1. And some other basic libraries which are used using web development .

**Security considerations:**

- 1. We will have signup using otp sent on the institute mail id provided and the user is asked to enter a password and further logins will take place using that mail id and password.
- 2. This whole process will be done with help of firebase auth.

**Database:**

- 1. The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real time.

## 2.4 Assumptions and Dependencies

We have made the following **assumptions**-

- 1. We assume that the development team has all the softwares and tools that will be required to complete this project on time.
- 2. We assume that dealers and users have access to internet resources and devices to access web-app.
- 3. We assume that our project team has been finalized and no new member will be added nor a present member will leave the group in the middle of the project.
- 4. We assume that all the requirements we get from our sources are correct and achievable.

These are a few **dependencies** we have for the system

- 1. The system is dependent on the firebase.
- 2. The system has efficient run time on Linux platform.



## 3 Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

The GUI of the application provides menus, toolbars, buttons, panes, containers, grids making the software interactive on the other hand not making it too descriptive.

#### 3.1.2 Hardware Interfaces

Users can easily interact with the software through the web app either through phones or on a PC with the help of mouse and keyboard.

#### 3.1.3 System Interfaces

The web app should be compatible on latest versions of browsers like chrome, firefox and microsoft edge on operating systems such as windows, linux, mac on pc and android, ios on mobile phones.

#### 3.1.4 Software Interfaces

Software allows the dealers to rent their bicycles at their price.  
customers can hire bicycles which suits them the best.

### 3.2 Functional Requirements

The software provides two types of interfaces - customer and dealer. On each end, there are several functions which the software shall provide. Section 3.2.1 provides the functional requirements for the customer end and section 3.2.2 provides the functional requirements for the dealer end. At the starting page the software will provide a page for registering and signing in. Both these features will have options whether to log in or register as a customer or a dealer. Then signing in any of them will open the respective page.

#### 3.2.1 Functional Requirement for the Customer end

- The software shall provide the user with the information of the **cycles available**. The cycle page when further checked **must display the price, pictures, whatsapp number of dealer, and also the availability**.

- It should also provide the list of all the **cycle shops** that provide the rental service. The shop page when further checked **should provide the contact number of the shop and all the cycles that the shop is offering, their costs, pictures, and also the availability.**
- The software should provide the user, a feature to **manage his/her profile**, which would include **changing the profile picture, managing the password and profile, seeing the rented cycles - either current or in the past** and also **managing all the transactions.**
- The software shall also provide the feature to **book** and also **cancel the transaction.** The **my-profile page** provided by the software should display all the **information about the booked, returned and cancelled cycle.**
- The software should also keep a **record of the return of every cycle** rented by the user. In the record it must **store the duration of the rent** and also **the fare of the rented time.**
- The software at its end ensures that the user **can book the cycle for a maximum 2 hours in advance.**
- The software shall also provide the feature to the user that he/she can **add a particular cycle to Favourites** and **post the feedback** of personal experience with the cycle.

### 3.2.2 Functional Requirement for the Dealer end

- The software shall provide a feature so that the dealer can **upload the information(pictures, prices, contact number etc.) about the available cycles.**
- The software shall provide the dealer **information about the available stock of cycles.**
- The software must be flexible for the dealer to **update the number and rate of cycles** provided by his/her store.

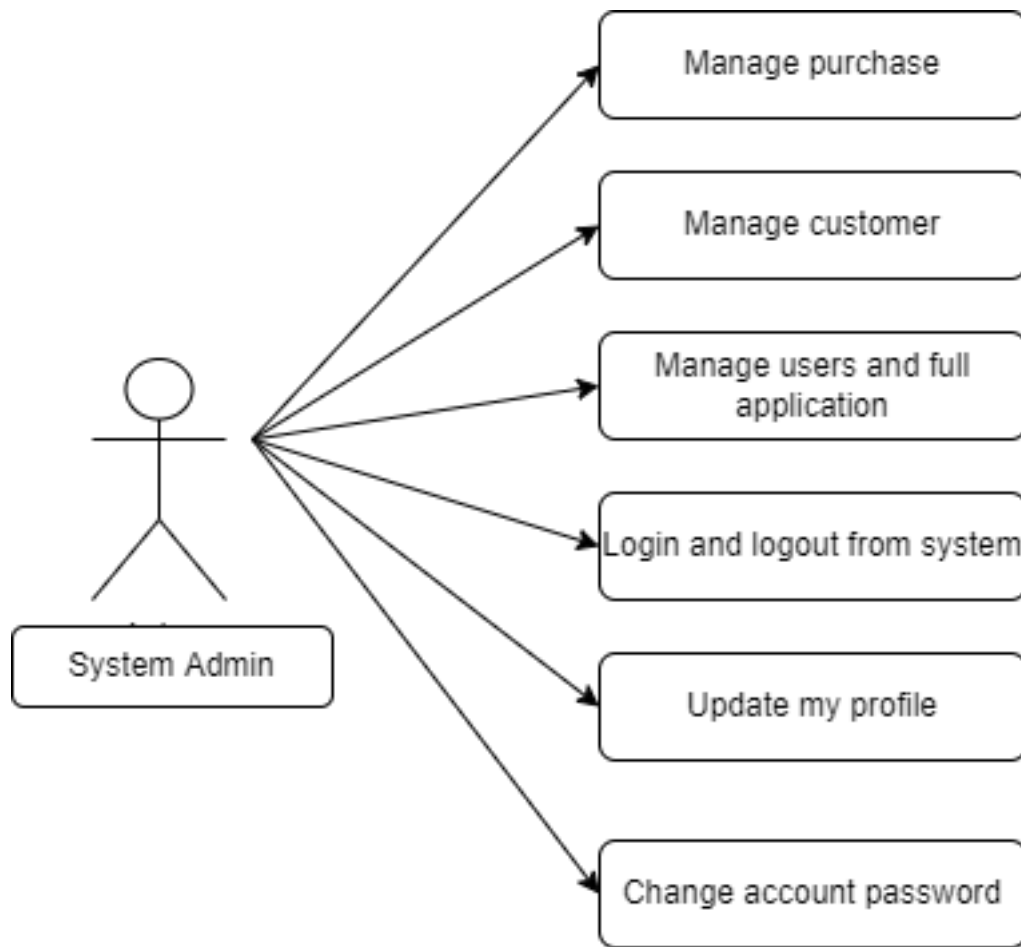
- The software shall ensure that the dealer gets a window which shall provide the dealer to **confirm the return of every cycle**.
- The software shall give the **information about how many cycles of any particular model are rented and to which user**. It should also keep a **record of every cycle being rented and also its fare**.

### 3.2.3 Functional Requirement for the Admin

- The software should provide a **contact us** feature where the **customer and dealer can raise their issues**.
- The users having admin privileges can **monitor each and every transaction** and also the **activities of other users**.
- The Admin can **remove or block the account of any user** when he/she encounters any malicious activity.

## 3.3 Use Case Model

### 3.3.1 Use Case #1 (use case name and unique identifier – e.g. U1)



**Author** -Shubham Kumar

**Purpose** - The Use-case model is a diagram which is used to show how users interact with the system in order to solve a problem. As such, the use case model defines the **user's objective**, **the interactions between the system and the user**, and the system's behavior required to meet these objectives

**Requirements Traceability** – This use case traces the requirements of system admin side.purchase history, login and account interface and customer information all done in this use case

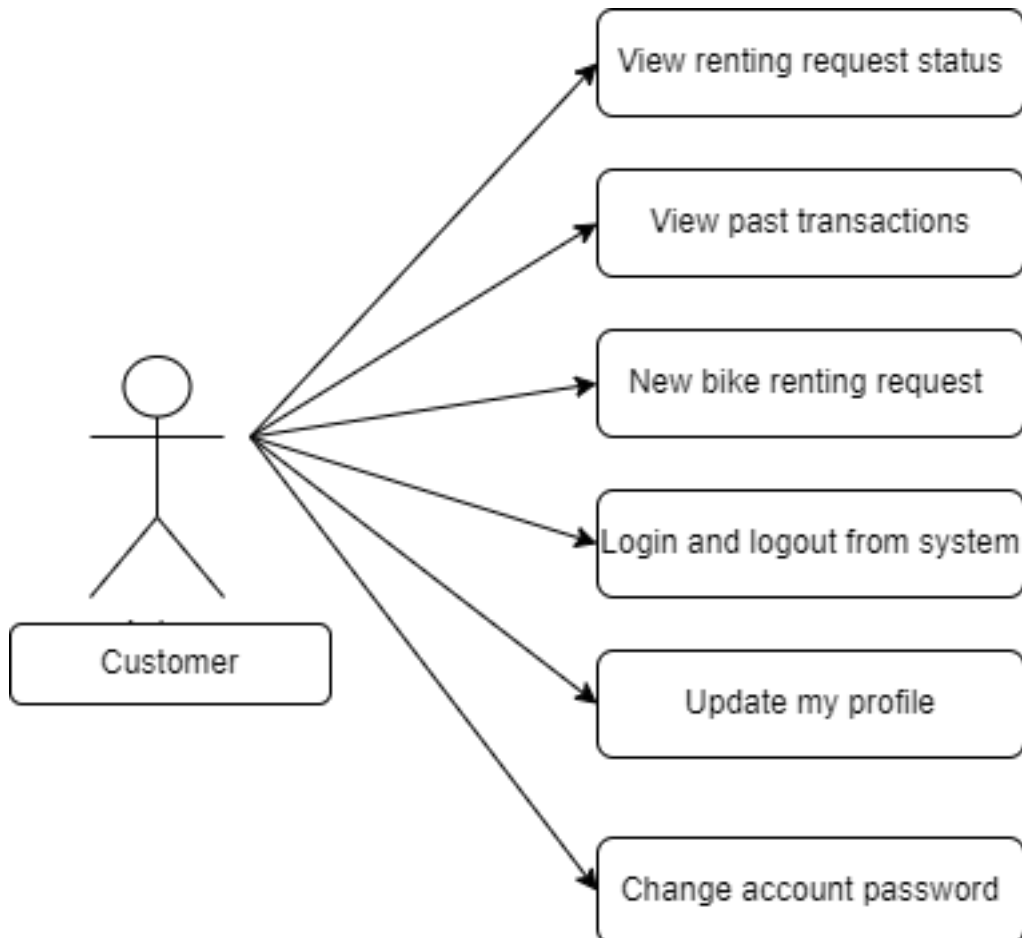
**Priority** - High

**Preconditions**- The system should check the status of all cycles.

**Post conditions** - The bike which is booked should be removed from the catalogue.

**Actors** – Dealers, Customer and system admin.

### 3.3.2 Use Case #2



**Author** – Shubham kumar

**Purpose** - The Use-case model is a diagram which is used to show how users interact with the system in order to solve a problem. As such, the use case model defines the **user's objective**, **the interactions between the system and the user**, and the system's behavior required to meet these objectives

**Requirements Traceability** – Login and account managements and all past history is include in this use case

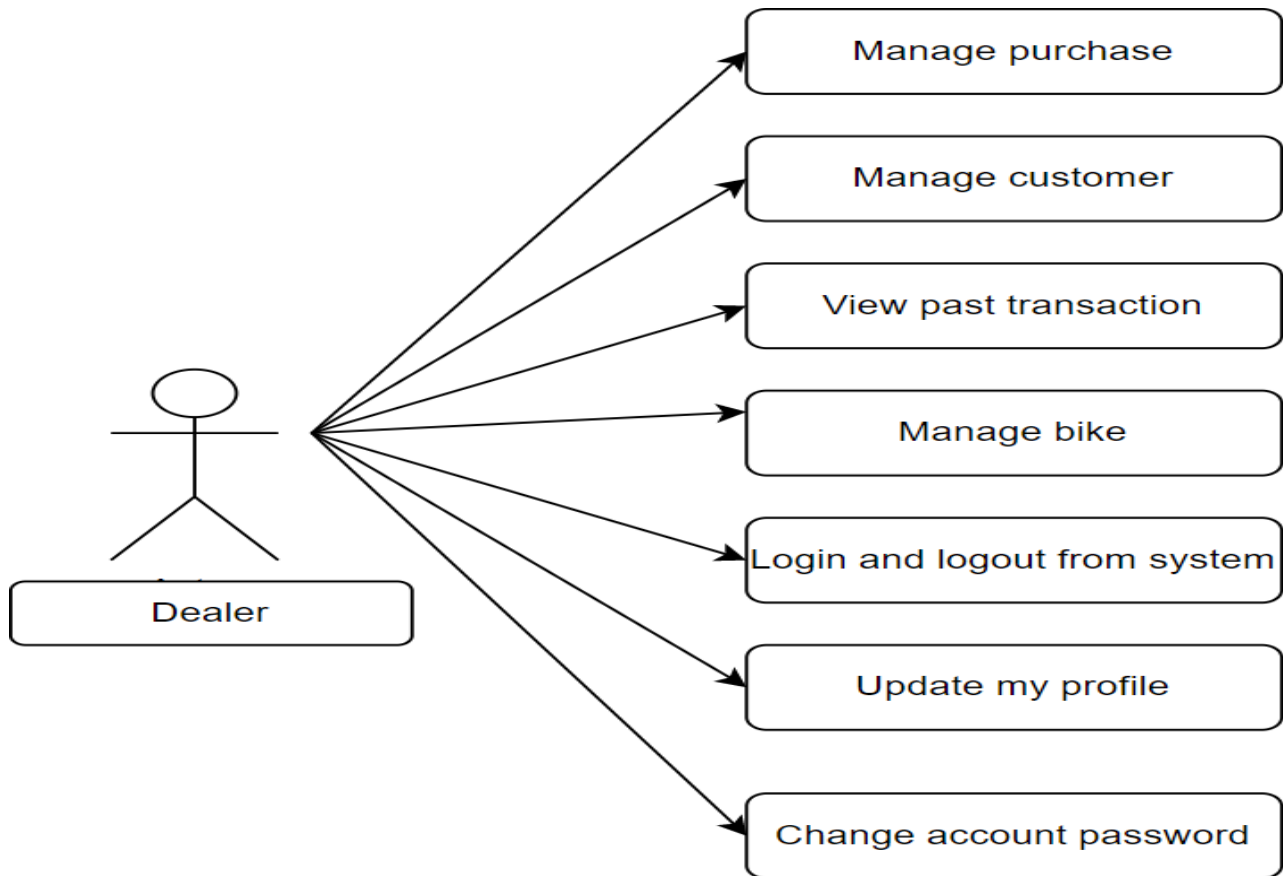
**Priority** - High

**Preconditions** - The system should check the status of all cycles.

**Post conditions** - The bike which is booked should be removed from the catalogue and put in customer account.

**Actors** – customer and system admin.

### 3.3.3 Use case #3



**Author** – Shubham Kumar

**Purpose** - The Use-case model is a diagram which is used to show how users interact with the system in order to solve a problem. As such, the use case model defines the **user's objective**, **the interactions between the system and the user**, and the system's behavior required to meet these objectives

**Requirements Traceability** – Bike managements and customer managements are requirements traced by this use case.

**Priority** - High

**Preconditions** - The system should check the status of all cycles.

**Post conditions** - system should check all bike status.

**Actors** – Dealer

## 4 Other Non-functional Requirements

### 4.1 Performance Requirements

The website application must load within the Industry Standard time.

The website application must support up to 10000 simultaneous cycle renting users (client).

Efficient algorithms should be used in database updation so that database can be updated in a short time.

### 4.2 Safety and Security Requirements

1. A user shall be required to create an account for the software using the institute email id. A verification mail shall be sent to verify the authenticity of the user.
2. A user can register on the software under two roles – one for the renter and one for the dealer. Each of these roles shall have some exclusive features facilitating the use of the software.
3. The application shall protect user data by using username and password for login.
4. The system shall embed security measures to ensure that the records present in the system are secure and no unauthorized personnel can access them.
5. The domain will get an SSL Certificate that protects the domain to be published by hackers and get it to some secure level.
6. On the database level of security, the system will encrypt the password of each user and determine the user who has access to the database.

### 4.3 Software Quality Attributes

#### 4.3.1 Reliability

- Downtime after a failure shall not exceed 24 hours.
- The system will be available all the time. There are no page crashes and freezes. If there is an unexpected error, an error page will appear, and the reason will register on the error log, and the correction will be provided.
- The system shall be available 99.99% of the time.

#### 4.3.2 Availability

- The system should be available on the specified date and specified time as many customers are making a booking.
- The system shall protect against denial of service (DOS).

#### 4.3.3 Usability

- A user who already knows what product he is interested in should locate and view that page in 30 seconds.
- The facility to return to the home page from any page Should be available.



- Labels of all Objects in the entire system Must be in Understandable form (Meaningful form).
- Multiple flows (ways) are preferable to do any task.
- The home page Should be a Centralised System (Screen/Window) to go to any feature and to get any result.

#### **4.3.4 Flexibility**

- The system should accommodate new products and product lines without significant reengineering.
- The system website shall be viewable from Mozilla Firefox 90.0 or later, Internet Explorer 9.0 or later, Google Chrome 90.0 or later, Safari 13.0 or later, Brave 1.25 or later, and Microsoft Edge 95.0 or later.

#### **4.3.5 Portability**

- The system shall work on Windows, Linux, and macOS.
- The system shall open fine on iOS and Android phones.
- The system shall work on web browsers.

#### **4.3.6 Maintainability**

- The software should be easy to extend.
- The code should be written in a way that favors the implementation of new functions. For future operations to be implemented quickly to the software.

#### **4.3.7 Testability**

- Test environments should be built for the software to allow testing of the software's different functions.

#### **4.3.8 Interoperability**

- The system shall have a communications layer with only one interface for all services to follow.
- The system shall require all services to communicate only to the communications layer, not with other services.

#### **4.3.9 Effectiveness**

- The system process of every function must be on a separate page so that the user can do the function directly, not through many pages.
- The system shall operate 100% of the records submitted.

#### **4.3.10 Data Integrity**

- To prevent malicious corruption of the system, the system shall retain its data for 90 days after a designated user authorizes the deletion of a record.
- The system shall maintain data integrity by keeping backups of all updates to the database for every record transaction.

#### **4.3.11 Fault Tolerance**

- The system shall have all functions implemented as services within a service-oriented architecture to allow the system to operate in the event of one or more services failing.
- The system must have Error handling features.

## Appendix A – Data Dictionary

### Class 1 : Cycles

Element s	Functions
Cycles	Cycles of a company given on rent by a particular dealer.

<b>Attributes</b>	number :int	Number of total cycles
	company: string	Which company's cycles are they
	dealer: dealer_profile *dealer_profile class is explained afterwards	Who is the dealer of this cycle
	current_available: int	Number of current available cycles
	booked_cycle :user_profile[]* *user_profile class is explained afterwards	array of profiles who have booked a cycle
	Id : string	A unique id is given to an object
	Rating : pair(double,int)	What is rating of this cycle and number of users who have given the rating
<b>Operation</b>	book_cycle(user_profile)	Decrease the number of current available cycles and append the booked_cycle.
	return_cycle(user_profile)	User returns a cycle back to dealer
	Increase_cycle(int)	dealer increases the cycle
	Decrease_cycle(int)	the dealer decreases the number of cycles.
	set_company(string)	sets the company of cycles
	set_dealer(dealer_profile)	sets the dealer of cycle

	rating(int)	change the rating
--	-------------	-------------------

**Class 2 : user\_profile**

Elements	Functions
user_profile	gives the profile of the user

Attributes	name: string	Name of user
	email: string	email of user
	roll_number : long int	Institute roll no. of user
	profile_photo : image	profil image of user
	Id	unique id of user
	Current_status_booked : {cycles[],number}	array of cycles and their number booked by user
	favourites: cycles[]	array of favourite cycles of user
Operation	add_favourite(cycles)	add a cycle to user favourites
	remove_favourite(cycles)	remove a cycle from favourites
	return_cycle(cycles,number)	User returns a cycle back to dealer
	book_cycle(cycles,number)	user books cycles
	set_image(image)	sets the profile photo of user
	set_name(string)	sets the name of user
	set_email()	sets the email of user
	rating(cycles,int)	give rating to cycle

**Class 3 : dealer\_profile**

Elements	Functions
----------	-----------

dealer_profile	gives the profile of the dealer
----------------	---------------------------------

<b>Attributes</b>	name: string	Name of dealer
	email: string	email of dealer
	insttute_id: long int	Institute id of dealer
	profile_photo : image	profil image of dealer
	Id	unique id of dealer
	dealer_cycles : {cycles[]}	array of cycles given on rent by dealer
	bookings(cycle[])	array of upcoming bookings of dealer
<b>Operation</b>	confirm_pick(cycles,user_profile,number)	confirms the pick of cycles.
	confirm_return(cycles,user_profile,number)	confirms the return of cycles
	set_image(image)	sets the profile photo of dealer
	set_name(string)	sets the name of dealer
	set_email()	sets the email of dealer

## Appendix B - Group Log

The total number of group meetings are 4-5 with each meeting of about 2 hours each. We also divided our group into 4 sub-groups of 2-3 students and these sub-groups interacted through meetings of about 5-6 hours.

Sections of SRS and respective contributors

1. Section 1: Abhishek Bansod, Divyansh Kankariya, Het Hitendrakumar Patel
2. Section 2: Avi Kumar, Pratham Jain
3. Section 3: Shubham Kumar, Nishi Mehta, Mohammad Umam
4. Section 4: Het Hitendrakumar Patel, Arpit Kumar Rai, Maurya Jadhav
5. Appendix A: Avi Kumar
6. Appendix B: Abhishek Bansod