



Bike Store and Blog

Submitted by

Rasel Ahamed

Id: 171-35-1958

Department of Software Engineering
Daffodil International University

Supervised by

Khalid Been Md. Badruzzaman Biplob

Lecturer

Faculty of Science and Information Technology
Department of Software Engineering
Daffodil International University

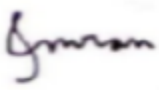
FALL- 2020

A project (SWE 431) submitted in fulfillment of the requirements for the degree of BSc in Software Engineering

APPROVAL

This **thesis/project/internship** titled on “**Bike Store and Blog**”, submitted by Student Name: Rasel Ahamed (ID: 171-35-1958) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



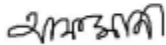
Chairman

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



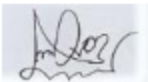
Internal Examiner 1

Dr. Md. Mostafijur Rahman
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



Internal Examiner 2

Afsana Begum
Lecturer (Senior)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



External Examiner

Professor Dr. Mohammed Nasir Uddin
Professor and Chairman
Department of Computer Science and Engineering
Jagannath University

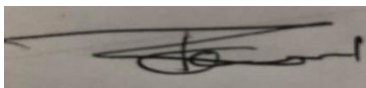
PROJECT DECLARATION

Declaration of Project:

Author Full Name : Rasel Ahamed
Date of Birth : 01-01-1998
Title of Project : Bike Store and Blog

I, the author, hereby declare that the project becomes the property of Daffodil International University (DIU) and to be placed at the department of Software Engineering. I also give permission to the DIU to reproduce this project in whole or in part for the purpose of research or academic exchange only.

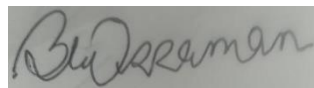
Certified by:



SIGNATURE

Rasel Ahamed (171-35-1958)

STUDENT NAME AND ID
Date: 03-02-2021



SIGNATURE OF SUPERVISOR

Khalid Been Md. Badruzzaman Biplob

NAME OF SUPERVISOR
Date: 03-02-2021

ACKNOWLEDGEMENT

At first, I express my heartiest thanks and gratefulness to almighty Allah as he has given the ability to find the concept and divine blessing made it possible to complete this project successfully. I am so thanked to Daffodil International University for giving me that chance to prove myself. I also feel grateful to Khalid Been Md. Badruzzaman Biplob, Lecturer, Department of Software Engineering, and Daffodil International University. Deep knowledge keen interest of my supervisor in the field of technology influenced me to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this report. I would like to express my heartiest gratitude to Dr. Imran Mahmud, Associate Professor & Head In- Charge, Department of Software Engineering, Daffodil International University, for his kind help to finish my project and to other faculty member and all other staffs. Finally, I must acknowledge with due respect the constant support and patience of my family.

Table of Contents

APPROVAL.....	ii
INTERNSHIP DECLARATION	iii
ACKNOWLEDGEMENT	iv
TABLE OF CONTENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER 1: INTRODUCTION.....	01
1.1 Project Overview.....	01
1.2 Project Purpose.....	01
1.2.1 Background.....	01
1.2.2 Benefits & Beneficiaries.....	01
1.2.3 Goals.....	01
1.3 Stakeholders.....	01
1.4 Proposed System Model.....	02
1.5 Project Schedule.....	02
1.5.1 Gantt Chart	02
CHAPTER 2: SOFTWARE REQUIREMENT SPECIFICATION	
2.1 Functional Requirements.....	04
2.1 Non-Functional Requirements.....	05
2.3 Performance Requirements.....	06
2.3.1 Speed and Latency Requirements.....	06
2.3.2 Precision or Accuracy Requirements.....	06
2.3.3 Capacity Requirements.....	06
2.4 Dependability Requirements	
2.4.1 Reliability Requirements	06
2.4.2 Availability Requirements.....	06
2.4.3 Robustness or Fault-Tolerance Requirements.....	06
2.4.4 Safety-Critical Requirements.....	06
2.5 Maintainability and Supportability Requirements.....	07
2.5.1 Maintenance Requirements.....	07

2.5.2 Supportability Requirements.....	07
2.5.3 Adaptability Requirements.....	07
2.5.4 Scalability or Extensibility Requirements.....	07
2.6 Security Requirements.....	07
2.6.1 Access Requirements.....	07
2.6.2 Integrity Requirements.....	07
2.6.3 Privacy Requirements.....	08
2.7 Usability and Human-Interaction Requirements.....	08
2.7.1 Ease of Use Requirements.....	08
2.7.2 Personalization and Internationalization Requirements.....	08
2.7.3 Understandability and Politeness Requirements.....	08
2.7.4 Accessibility Requirements.....	09
2.7.5 User Documentation Requirements.....	09
2.7.6 Training Requirements.....	10
2.8 Look and Feel Requirements.....	10
2.8.1 Appearance Requirements.....	10
2.8.2 Style Requirements.....	10
2.9 Operational and Environmental Requirements.....	11
2.9.1 Expected Physical Environment.....	11
2.9.2 Requirements for Interfacing with Adjacent Systems.....	11
2.9.3 Projectization Requirements.....	11
2.9.4 Release Requirements.....	11
2.10. Legal Requirements.....	11
2.10.1 Compliance Requirements.....	11
2.10.2 Standards Requirements.....	11

CHAPTER 3: SYSTEM ANALYSIS

3.1 Use Case Diagram.....	12
3.2 Use Case Description (for each use case).....	13
3.3 Activity Diagram (for each use case)	23
3.4 Sequence Diagram (for each use case).....	33

Chapter 4: System Design Specification

4.1 ER Diagram.....	43
4.2 Class Diagram.....	44
4.3 Development Tools & Technology.....	45

4.3.1 User Interface Technology.....	45
4.3.1 Implementation Tools & Platforms.....	45
CHAPTER 5: SYSTEM TESTING	
5.1 Testing Features.....	46
5.1.1 Features to be tested.....	46
5.1.2 Features not to be tested.....	46
5.2 Testing Strategies.....	46
5.2.1 Test Approach.....	46
5.2.2 Pass/Fail Criteria.....	47
5.2.3 Suspension and Resumption.....	47
5.2.4 Testing Schedule.....	48
5.3 Testing Environment.....	35
5.4 Test Case.....	49
CHAPTER 6: USER MANUAL	
61 User Manual.....	51
CHAPTER 7: PROJECT SUMMARY	
7.1 GitHub Link.....	55
7.2 Limitations.....	55
7.3 Obstacles & Achievements.....	55
7.4 Future Scope.....	55
CHAPTER 8: APPENDICS.....	56
REFERENCES	56

List of Tables

Point	Name of figures	Page no:
Table 2.1	Functional Requirements	4
Table 2.2	Non-Functional Requirements	5
Table 5.3	Testing Schedule	48
Table 5.4	Test Case-1	49
Table 5.6	Test Case-2	50

List of Figures

Point	Name of figures	Page no:
Figure 1.1	Agile Model	2
Figure 1.2	Gant Chart	2
Figure 1.3	Work Distribution	3
Figure 3.1	Use Case	12
Figure 4.1	ER Diagram	43
Figure 4.2	Class Diagram	44

CHAPTER-01

Introduction

1.1 Project Overview

Bike Store and Blog is a web-based system where biker or any other consumer can buy bike, buy bike parts. User also rent bike from this website. They can service their bike also.

If user want they can also post a review which thing they can buy from that website, or read review which other user post.

1.2 Project Purpose

1.2.1 Background

Based on the project overview, we have seen that the bike user faces many problems. They cannot find all product in one platform. So we want to decide develop a user friendly software where biker find everything in one platform. And who want to buy bike, parts or servicing their bike they can also see the review.

1.2.2 Benefits & Beneficiaries

- User can see the update bike price.
- User can rent bike from the website.
- User can service their bike.
- User can read the review.

1.2.3 Goals

- The goal of my project is to ensure save time and effort.
- User can rent bike for his emergency work.
- Before buy bike, parts etc they can read the review.

1.3 Stakeholders

- Admin
- User

1.4 Proposed System Model

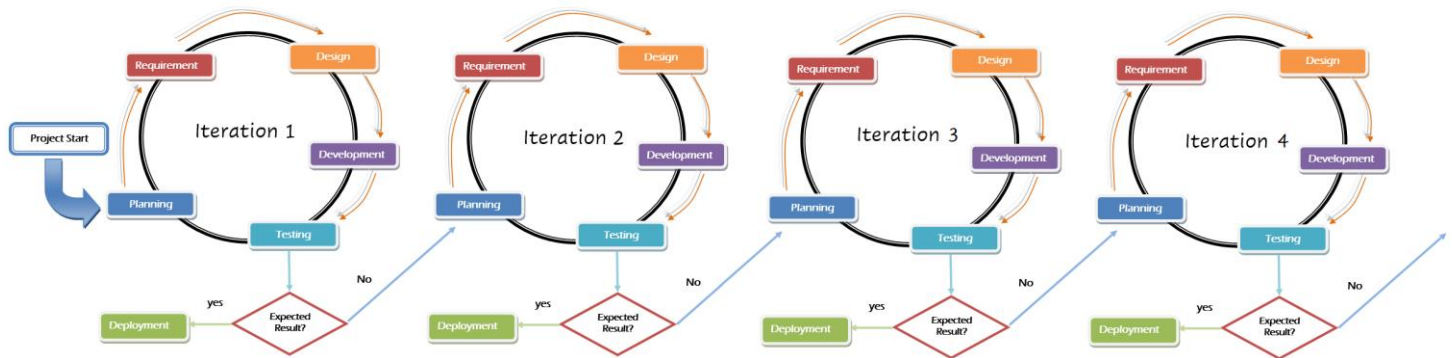


Figure 1.1: Agile Model

1.5 Project Schedule

1.5.1 Gant Chart

Duration

Task	25 – 30 June.	20 - 27 Oct.	02 – 22 Nov.	25Nov–25 Dec.	26 – 31 Dec	1 – 5 Jan.
Project Proposal						
SRS						
Designing						
Coding						
Testing						
Finalization						

Figure 1.2: Gant Chart

Project Proposal	Rasel Ahamed	5 Days
Software Requirement Specification	Rasel Ahamed	7 Days
Software Design	Rasel Ahamed	20 Days
Coding	Rasel Ahamed	30 Days
Software Testing	Rasel Ahamed	5 Days
Project Finalization	Rasel Ahamed	5 Days

Figure 1.3: Work Distribution

CHAPTER-02

Software Requirement Specification

2.1 Functional Requirements

Functional requirements for my project is given in the below table.

ID	Name	Description	Functional Requirements	Priority
FR-01	Registration	User input their valid ingredient for registration.	Functional	High
FR-02	Login	User can login with their valid email id and password.	Functional	High
FR-03	Buy Bike	User can buy any company or any model bike which they choose.	Functional	High
FR-04	Bike Rent	User can rent a bike which one is available.	Functional	High
FR-05	Bike Servicing	User can service their bike which service they need.	Functional	High
FR-06	Buy Parts	User can buy different type of parts for their bike.	Functional	High
FR-07	Review	User can post a review if they interest, and read review.	Functional	Medium
FR-08	Add Bike	Admin can add new model bike on behalf of customer demand.	Functional	High
FR-09	Add Rent Bike	Admin can add more rent bike to give more opportunity for customer.	Functional	High
FR-10	Add Parts	Admin can add more update bike parts.	Functional	High

Table 2.1: Functional Requirements

2.2 Non Functional requirements

Functional requirements for my project is given in the below table.

ID	Name	Description	Non-Functional Requirement	Priority
NFR-01	Security	Using token-based confirmation, meeting, and approval 2FA it will be secure from unapproved access.	Non-Functional	High
NFR-02	Availability	The system should work 24/7 as user can get access and service.	Non-Functional	High
NFR-03	Accuracy	Information or process requirements concern about characterizing the accuracy which the arrangement will record or deliver information.	Non-Functional	High
NFR-04	Maintenance	It's the way how simple to help, change, and improve the system.	Non-Functional	Medium

Table 2.2: Non-Functional Requirements

2.3 Performance Requirements

2.3.1 Speed and Latency Requirements

- Dataset would be embedded in MySQL Databases using python manage.py craftsman Make (Command).
- UI design expands on the user satisfies table informational collection and show. The Database we need to use python craftsman move this order.

2.3.2 Precisions or Accuracy Requirements

- All user are capable to show accurate page where he or she can show detail of bike, rent bike, parts. Admin can see what need customer.
- Admin can update bike, rent bike and parts.

2.3.3 Capacity Requirements

- At a time multiple users can use this system.
- Admin can add multiple product at a time.

2.4 Dependability Requirements

2.4.1 Reliability Requirements

- Admin must be log in to the system using their valid ingredient.
- Admin can easily update the product details.
- Admin can add bike, add rent bike, add parts etc. when need.

2.4.2 Availability Requirements

- At the point when we are using this system need to run xampp or wamp server.
- Need to know order for run appropriately and without any problem.
- Though this is a web application that should run on a web browser. (Ideal browser is Chrome, Firefox, Internet Explorer,).

2.4.3 Robustness or Fault-Tolerance Requirements

After service provide is very important to the end users.

2.4.4 Safety-Critical Requirements

- When the system detected any virus it notify the operator.
- This system detecting risk.

2.5 Maintainability and Supportability Requirements

2.5.1 Maintenance Requirements

- Web application didn't adjust or change.
- We need to keep up all security and different works.

2.5.2 Supportability Requirements

- When the system did not work perfectly then we test the system.
- We can configure everything which you need.
- Update security patch and others system which you need.

2.5.3 Adaptability Requirements

- This system is efficiently and first to change.
- This system is fit its environment according to changes.

2.5.4 Scalability or Extensibility Requirements

- Web application is good for user to get more information what the need.
- Every user can saw update bike information and price.

2.6 Security Requirements

2.6.1 Access Requirements

- Only the authorized user can buy things from the site.
- Guest allow this system.

2.6.2 Integrity Requirements

- This system does not expose to malicious modification.

2.6.3 Privacy Requirements

- User didn't see the admin data they see only which data create for them.
- Database should be protected by anonymous

2.7 Usability and Human Interaction Requirements

The system can be failed for ease of use. That is the reason I use this application very easy to understand, and easy to use.

2.7.1 Ease of Use Requirements

This system is easy to use and can easily be understood.

EUR-01	This system is very usable for Admin and User
Description	This system is indicates the several possible for users.
Stakeholders	Admin, User

2.7.2 Personalization and Internationalization Requirements

- This system is configurable.
- Multiple locales are supported at the same time.

2.7.3 Understand Ability and Politeness Requirements

UAR-01	The features of the Bike Store and Blog system.
Description	This system is more efficient and ease of use. This system is understand ability for both users. This system does not use any term and condition, that's why this is very easy to use.
Stakeholders	Admin, user

2.7.4 Accessibility Requirements

There is no access requirements beside those that has been outline in the below:

AR-01: Log in as Admin

AR-02: Log in as a User

AR-03: Log out as Admin

AR-04: Log out as a User

2.7.5 User Documentation Requirements

UDR-01	The system developer documentation
Description	Develop bike store and blog web application project we have specified requirements of user's documentation.
Stakeholders	System Developer
Priority	High

2.7.6 Training Requirements

Don't need any training to use this system. Only general knowledge to browse internet.
And need to know how to use a computer.

2.8 Look and Feel Requirements

2.8.1 Appearance Requirements

I make it simply justifiable with the goal that user can easily understand the system. For example, on the off chance that a user saw an updated product and price.

AR-01	Recently Added Product
Description	By seeing recent product user can buy recent product.
Stakeholders	User.

2.8.2 Style Requirements

SR-01	All content must be appear in a format.
Description	Input field and other view result show a specific format.
Stakeholders	Admin

2.9 Operational and Environmental Requirements

2.9.1 Expected Physical Requirements

There is no specific expected physical requirements in this system.

2.9.2 Requirements for Interfacing with Adjacent Systems

There is no specific interfacing with adjacent systems in this system.

2.9.3 Projectization Requirements

There is no specific projectization requirements in this system.

2.9.4 Release Requirements

There is no specific release requirements in this system.

2.10 Legal Requirements

These requirements consider any violence of rules and regulation and which rules should be followers to maintainers these systems.

2.10.1 Compliance Requirements

There is no specific compliance requirements in this system.

2.10.2 Standards Requirements

There is no specific standards requirements in this system.

CHAPTER-03

SYSTEM ANALYSIS

3.1 Use case Diagram

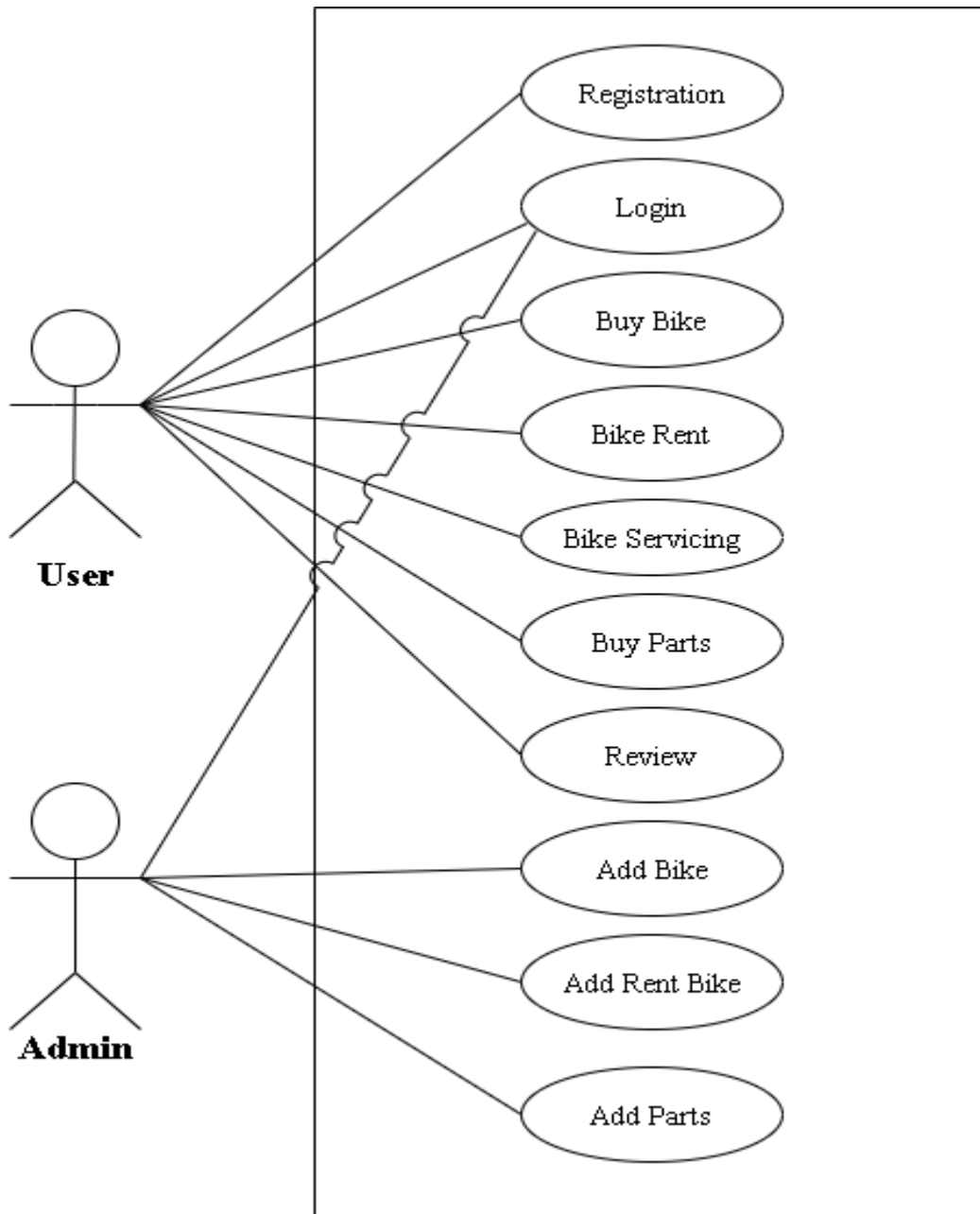


Figure 3.1: Use Case

3.2 Use Case Description

3.2.1 Registration

Use Case	Registration	
Goal	User registration the system with valid ingredient.	
Preconditions	Given data must be original.	
Success End Condition	User Registration complete.	
Failed End Condition	Registration failed by the user.	
Primary Actors:	User.	
Secondary Actors	N/A	
Trigger	After click the registration button fields shown by the system.	
Description/main Success Scenario	Step	Action
	1	Register data must be solid.
	2	User must be valid by the system.
	3	Unformatted data rejected by the system.
Alternative Flows	Step	Branching Action
	1a	Input unformatted data.
	2a	User cannot be registered.
Quality Requirements	Step	Requirement
	1	Registration with valid email and password.

Table 3.1: Registration

3.2.2 Login

Use Case	Login	
Goal	User and Admin must be login into to system.	
Preconditions	Without registration user cannot be login the system	
Success End Condition	User and Admin access the system	
Failed End Condition	Cannot get access the system?	
Primary Actors:	Admin, User	
Secondary Actors	N/A	
Trigger	Click the login button and get access the system.	
Description/main Success Scenario	Step	Action
	1	After complete the registration then login the system.
	2	After login then user can buy and rent bike
Alternative Flows	Step	Branching Action
	1a	User cannot login the system.
	2a	Unformatted data input by user.
Quality Requirements	Step	Requirement
	1	Login with valid email id and password.

Table 3.2: Login

3.2.3 Buy Bike

Use Case	Buy Bike	
Goal	User can buy bike in posted bike.	
Preconditions	Must be logged in to the system.	
Success End Condition	Bike buy successfully.	
Failed End Condition	Does not able to buy bike.	
Primary Actors	User	
Secondary Actors	Admin	
Trigger	After click on add to cart to button, admin see that.	
Description/main Success Scenario	Step	Action
	1	User can see the every model bike and bike details.
	2	Know the order product delivery date.
Alternative Flows	Step	Branching Action
	1a	User cannot buy bike.
Quality Requirements	Step	Requirement
	1	Always update bike list.

Table 3.3: Buy Bike

3.2.4 Bike Rent

Use Case	Bike Rent	
Goal	User can rent any bike which are available.	
Preconditions	Must be logged in the system.	
Success End Condition	Successfully rent bike.	
Failed End Condition	Does not able to rent any bike.	
Primary Actors:	User	
Secondary Actors	Admin	
Trigger	When choose rent bike he or she can rent that bike.	
Description/main Success Scenario	Step	Action
	1	Successfully rent a bike.
Alternative Flows	Step	Branching Action
	1a	Cannot rent any bike
Quality Requirements	Step	Requirement
	1	Good quality bike and tested by the mechanics.

Table 3.4: Bike Rent

3.2.5 Bike Servicing

Use Case	Bike Servicing	
Goal	User can service his or her bike.	
Preconditions	Must be logged in the system.	
Success End Condition	Successfully service the bike.	
Failed End Condition	Cannot service the bike.	
Primary Actors	User	
Secondary Actors	Admin	
Trigger	After confirm the order, admin see the service details.	
Description/main Success Scenario	Step	Action
	1	Successfully complete the servicing.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	Committed to give good quality service.

Table 3.5: Bike Servicing

3.2.6 Buy Parts

Use Case	Buy Parts		
Goal	User can buy any parts which are available.		
Preconditions	Must be logged in the system.		
Success End Condition	Successfully buy parts		
Failed End Condition	Cannot buy any parts.		
Primary Actors	User		
Secondary Actors	Admin		
Trigger	After confirm order admin can deliver that product..		
Description/main Success Scenario	Step	Action	
	1	Successfully buy a parts.	
Alternative Flows	Step	Branching Action	
	1a	N/A	
Quality Requirements	Step	Requirement	
	1	Always update the parts list and details.	

Table 3.6: Buy Parts

3.2.7 Review

Use Case	Review	
Goal	User can post a review and also read product review.	
Preconditions	Must be logged in the system.	
Success End Condition	Successfully post a review.	
Failed End Condition	Cannot post any review.	
Primary Actors:	User	
Secondary Actors	N/A	
Trigger	Index review.	
Description/main Success Scenario	Step	Action
	1	Successfully index review.
Alternative Flows	Step	Branching Action
	1a	Cannot post any product review.
Quality Requirements	Step	Requirement
	1	N/A

Table 3.7: Review

3.2.8 Add Bike

Use Case	Add Bike	
Goal	Add more and new model bike.	
Preconditions	Must be logged in the system.	
Success End Condition	Successfully add bike.	
Failed End Condition	Cannot add any bike.	
Primary Actors:	Admin	
Secondary Actors	N/A	
Trigger	Add more bike.	
Description/main Success Scenario	Step	Action
	1	Successfully add new model bike.
Alternative Flows	Step	Branching Action
	1a	N/A
Quality Requirements	Step	Requirement
	1	Add authentic product.

Table 3.8: Add Bike

3.2.9 Add Rent Bike

Use Case	Add Rent Bike		
Goal	Add more and new model rentable bike.		
Preconditions	Must be logged in the system.		
Success End Condition	Successfully add rent bike.		
Failed End Condition	Cannot add any rent bike.		
Primary Actors:	Admin		
Secondary Actors	N/A		
Trigger	Add more bike.		
Description/main Success Scenario	Step	Action	
	1	Successfully add new more rent bike.	
Alternative Flows	Step	Branching Action	
	1a	N/A	
Quality Requirements	Step	Requirement	
	1	Add good quality product.	

Table 3.9: Add Rent Bike

3.2.10 Add Parts

Use Case	Add Parts		
Goal	Add more and new model bike parts.		
Preconditions	Must be logged in the system.		
Success End Condition	Successfully add parts.		
Failed End Condition	Cannot add any parts.		
Primary Actors:	Admin		
Secondary Actors	N/A		
Trigger	Add more bike.		
Description/main Success Scenario	Step	Action	
	1	Successfully add new model parts.	
Alternative Flows	Step	Branching Action	
	1a	N/A	
Quality Requirements	Step	Requirement	
	1	Add authentic product.	

Table 3.10: Add Parts

3.3 Activity Diagram

3.3.1 Registration

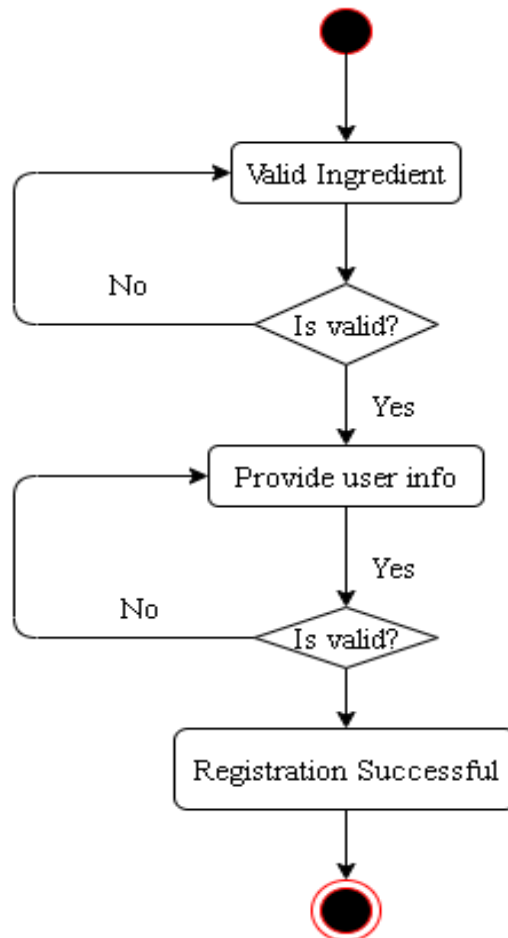


Figure: 3.2: Registration

3.3.2 Login

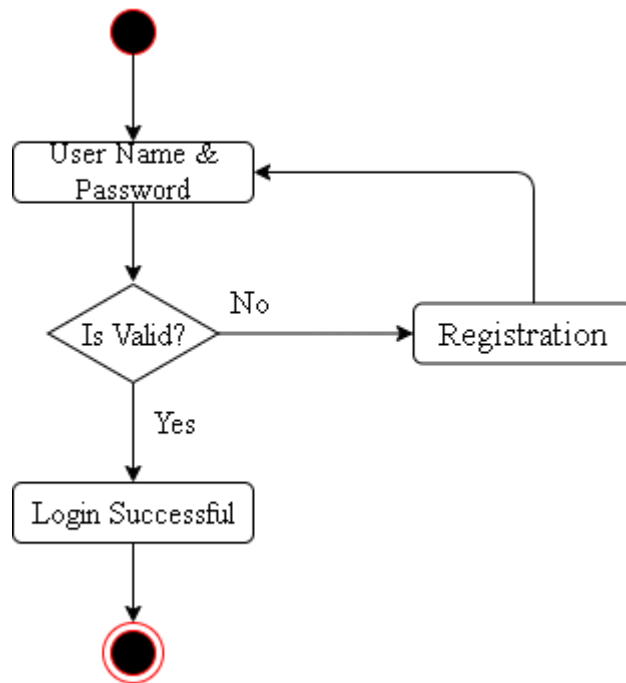


Figure: 3.3: Login

3.3.3 Buy Bike

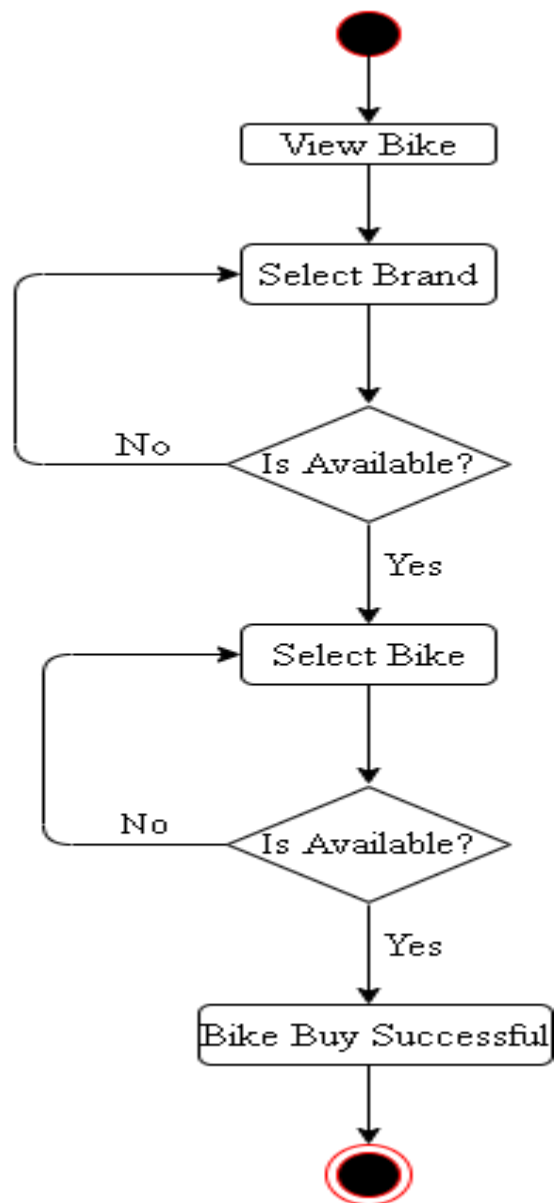


Figure: 3.4: Buy Bike

3.3.4 Bike Rent

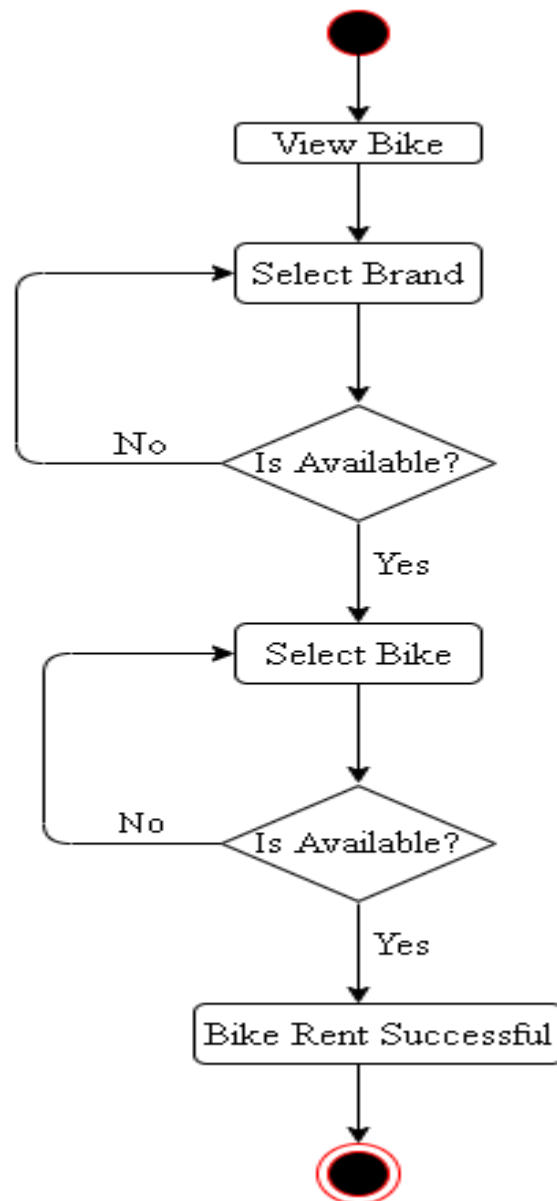


Figure: 3.5: Bike Rent

3.3.5 Bike Servicing

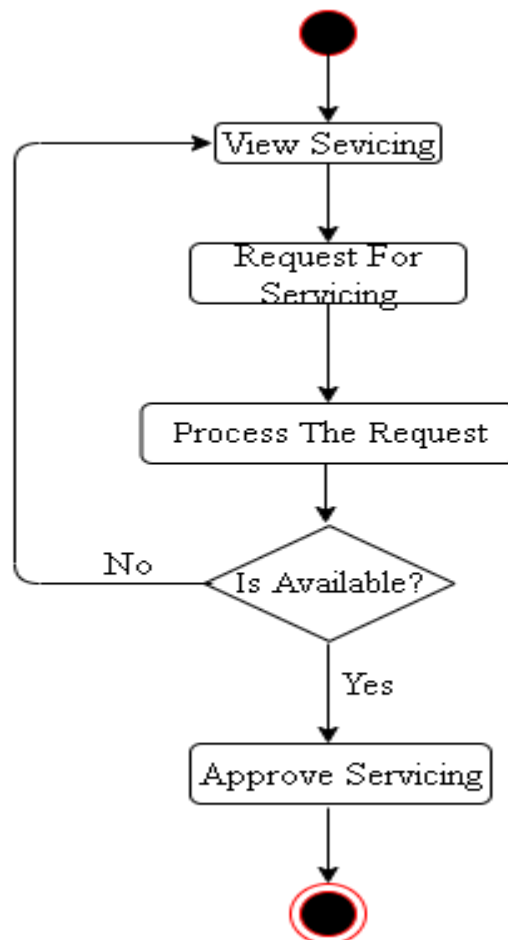


Figure: 3.6: Bike Servicing

3.3.6 Buy Parts

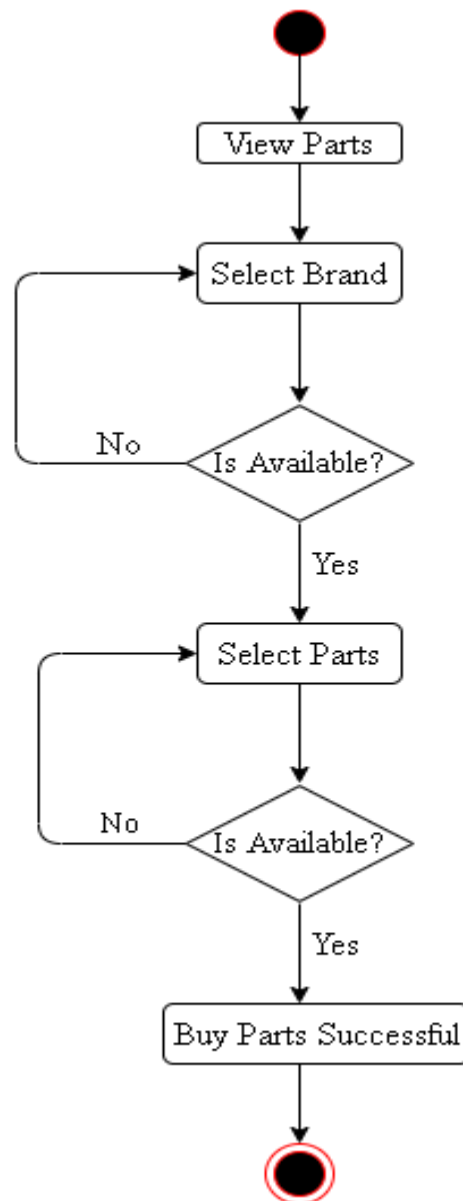


Figure: 3.7: Buy Parts

3.3.7 Review

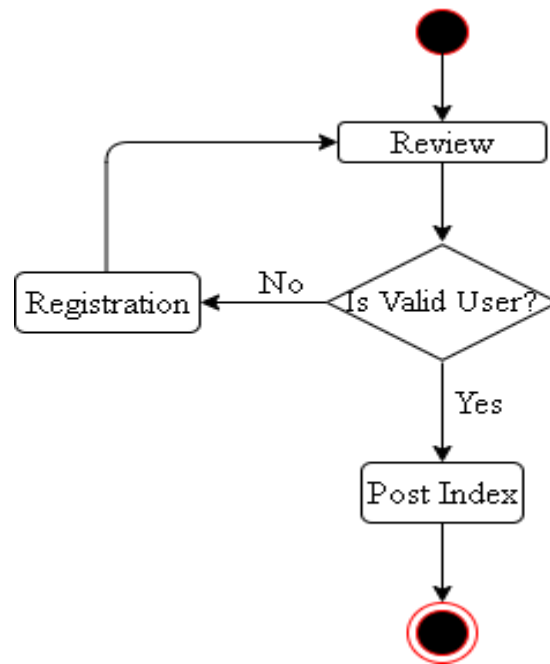


Figure: 3.8: Review

3.3.8 Add Bike

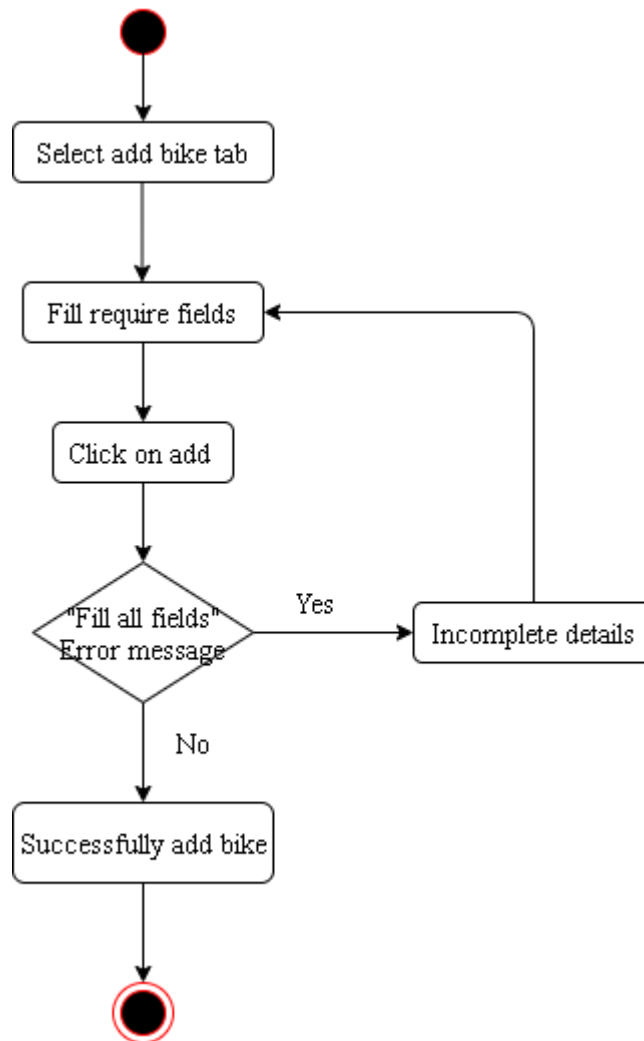


Figure: 3.9: Add Bike

3.3.9 Add Rent Bike

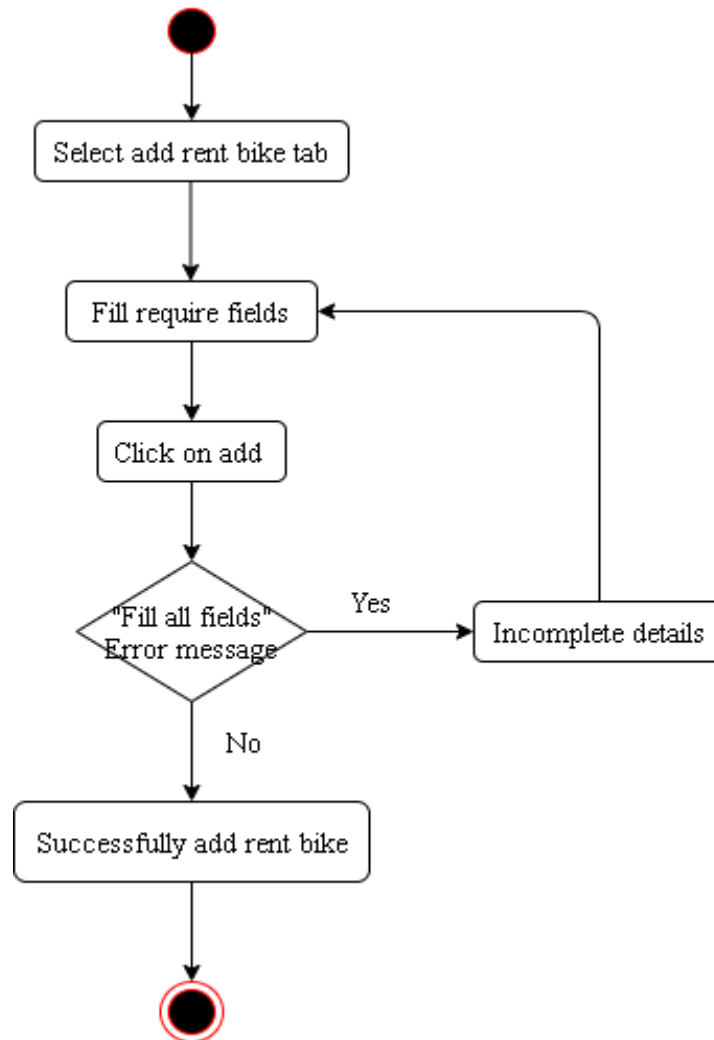


Figure: 3.10: Add Rent Bike

3.3.10 Add parts

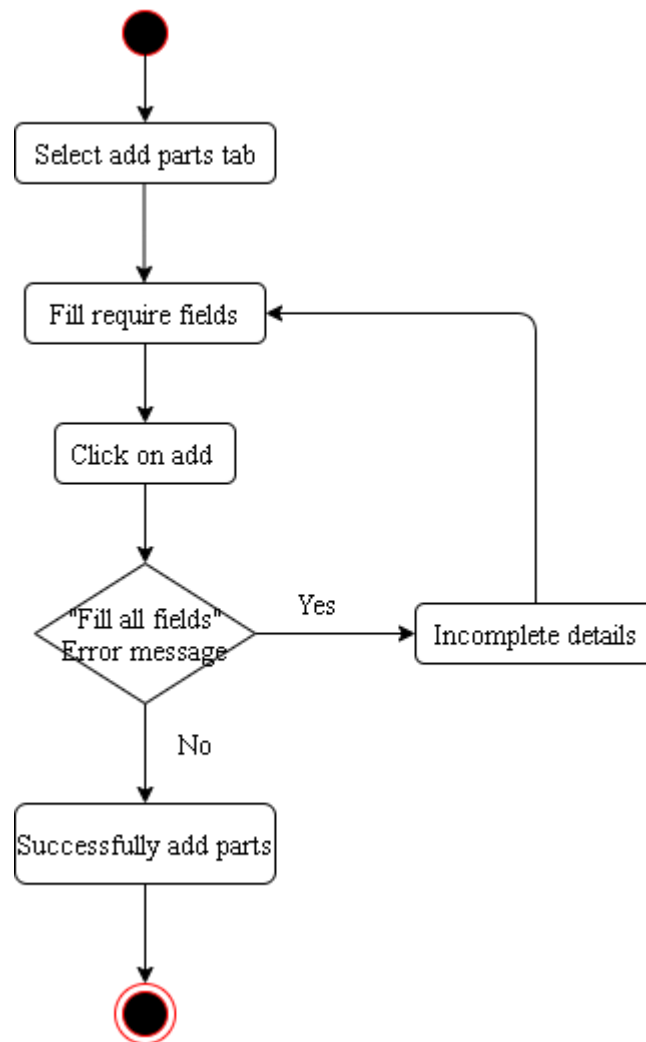


Figure: 3.11: Add Parts

3.4 System Sequence Diagram

3.4.1 Registration

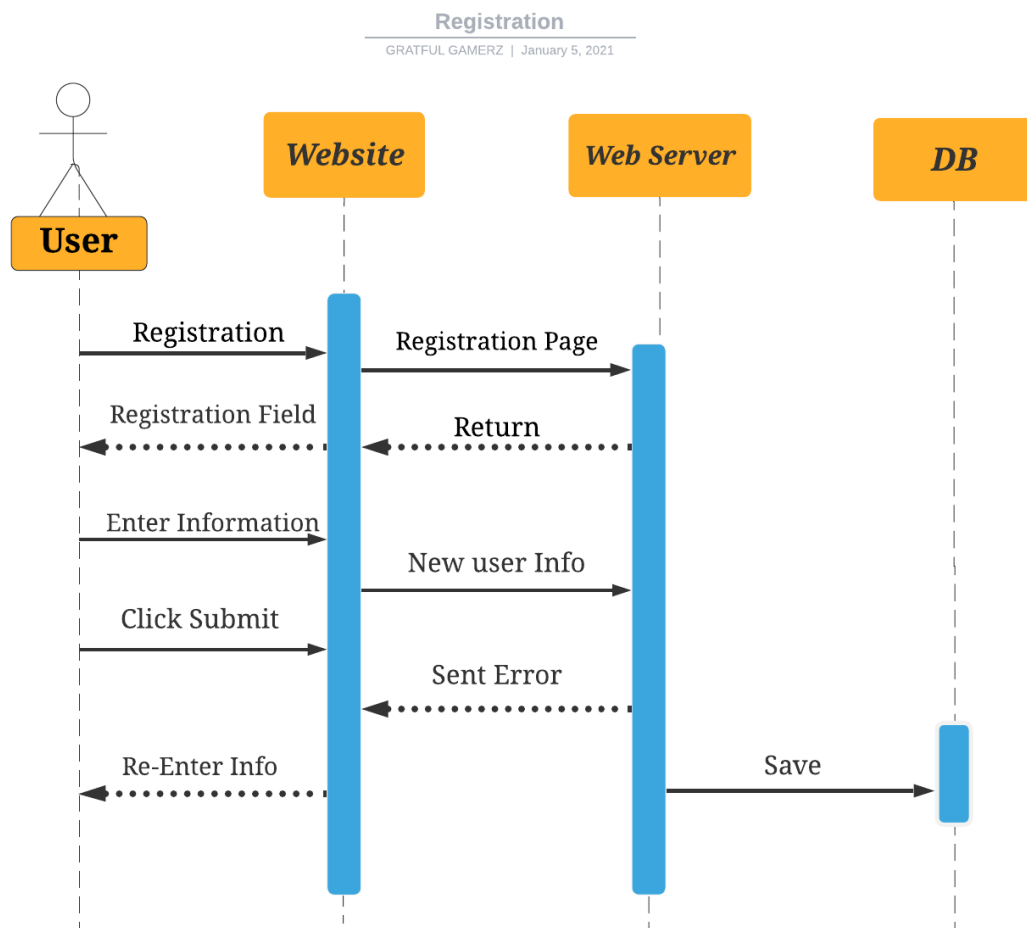


Figure: 3.12: Registration

3.4.2 Login

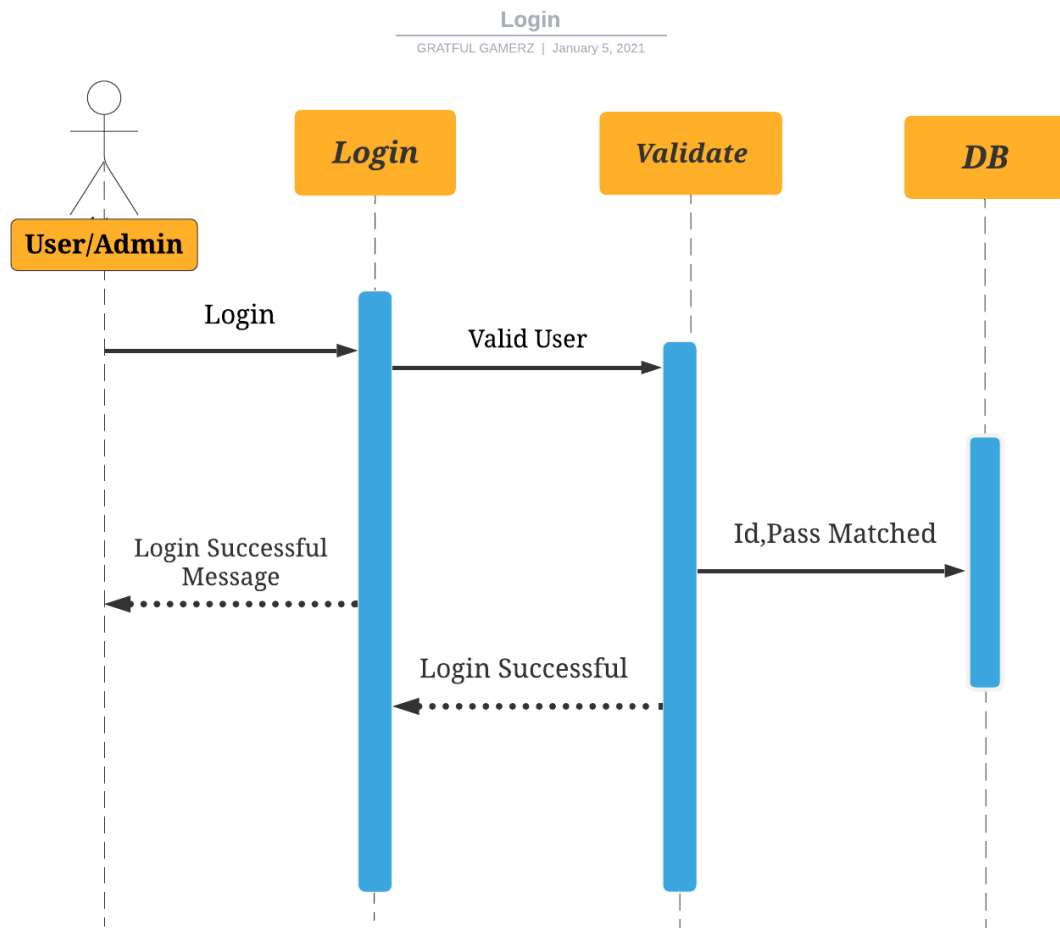


Figure: 3.13: Login

3.4.3 Buy Bike

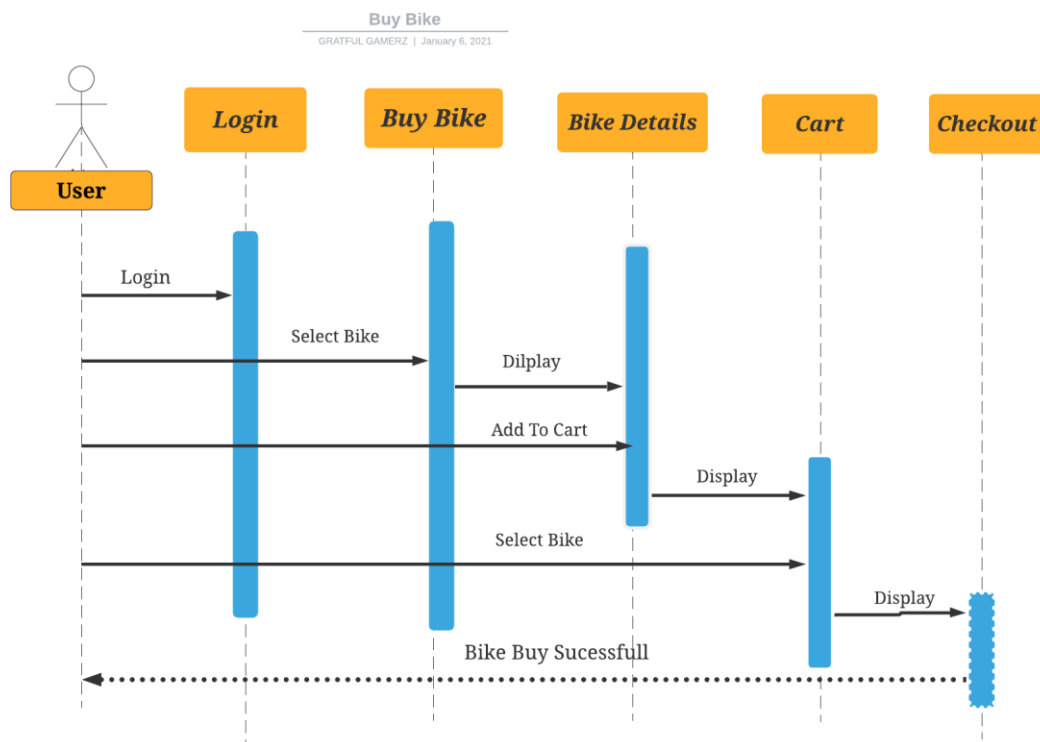


Figure: 3.14: Buy Bike

3.4.4 Bike Rent

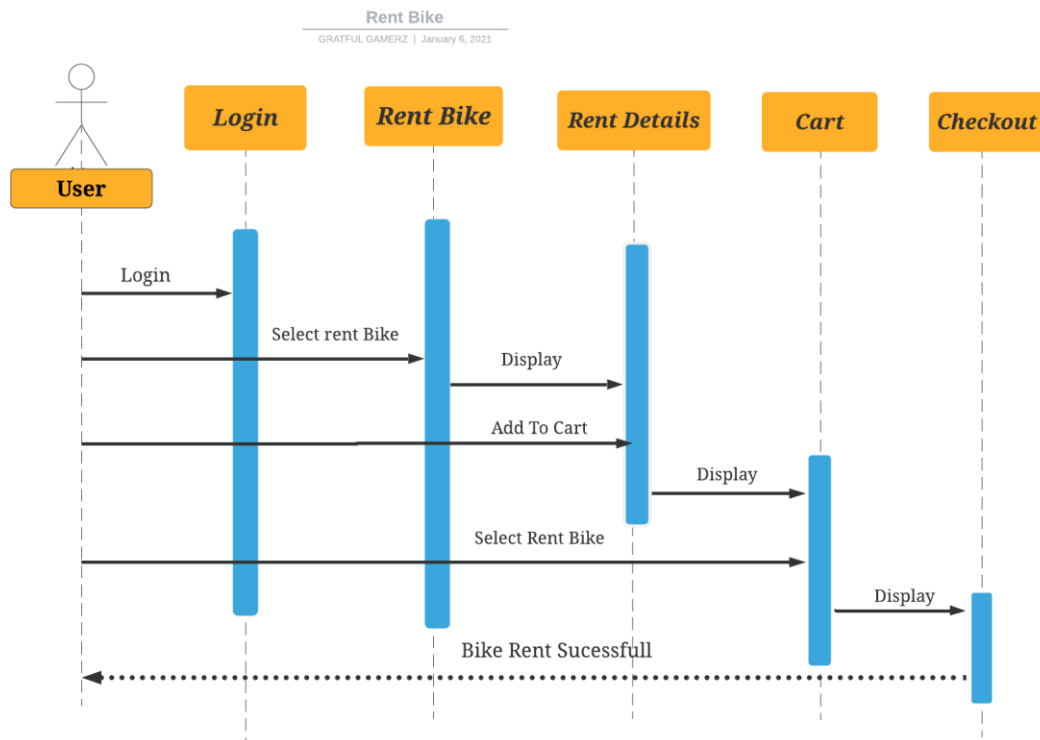


Figure: 3.15: Bike Rent

3.4.5 Bike Servicing

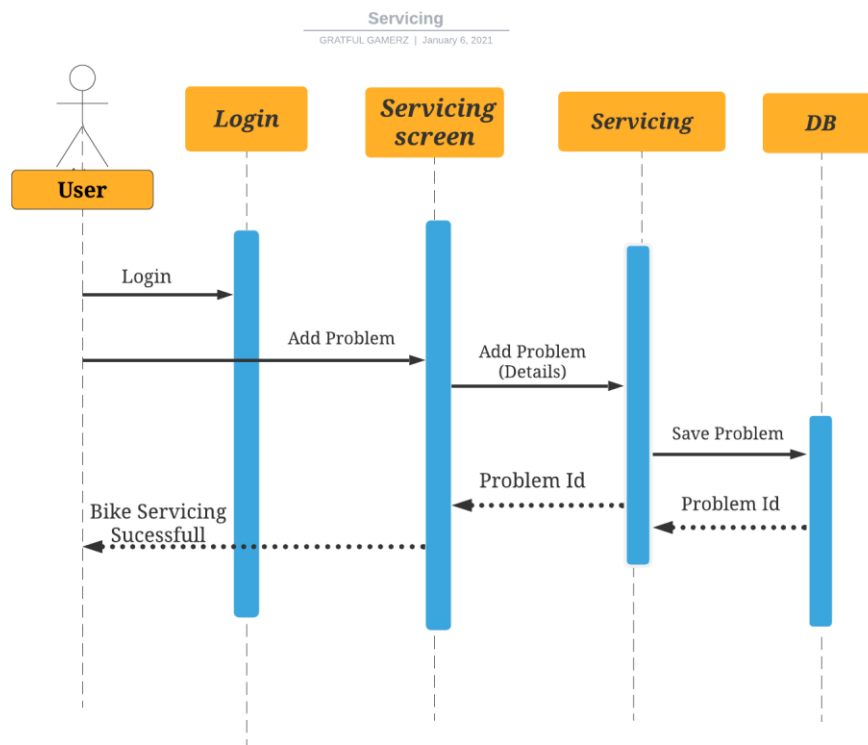


Figure: 3.16: Bike Servicing

3.4.6 Buy Parts

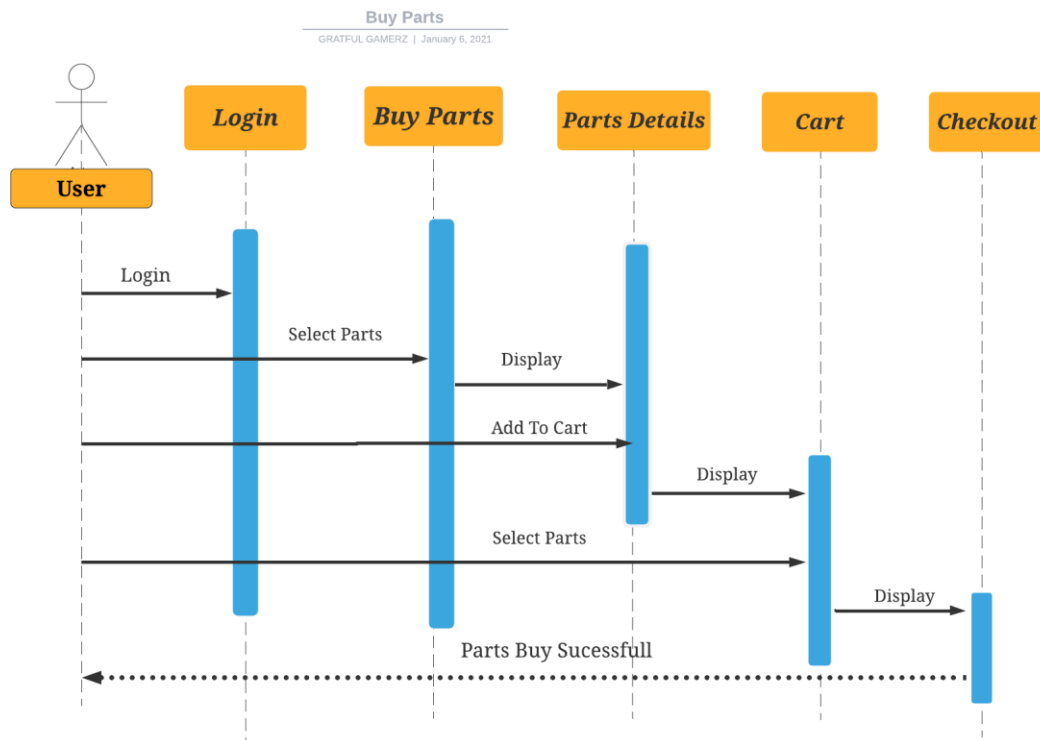


Figure: 3.17: Buy Parts

3.4.7 Review

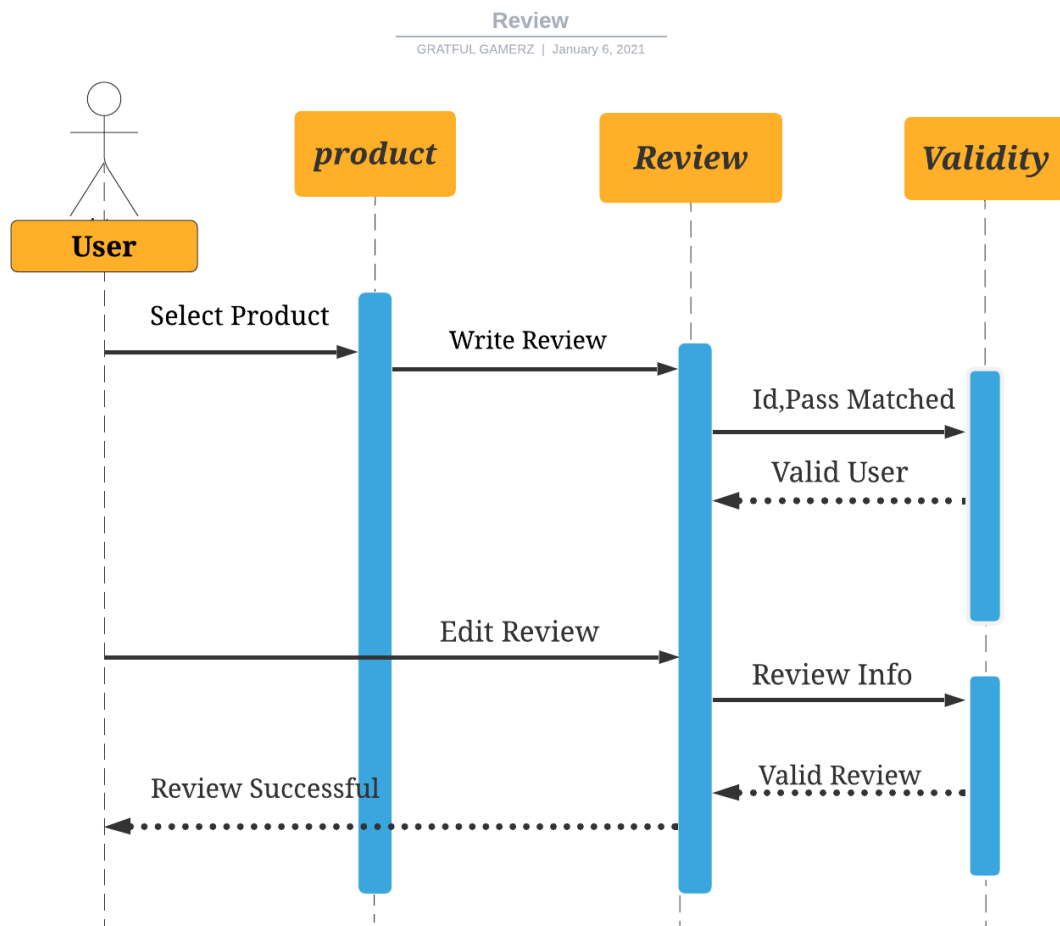


Figure: 3.18: Review

3.4.8 Add Bike

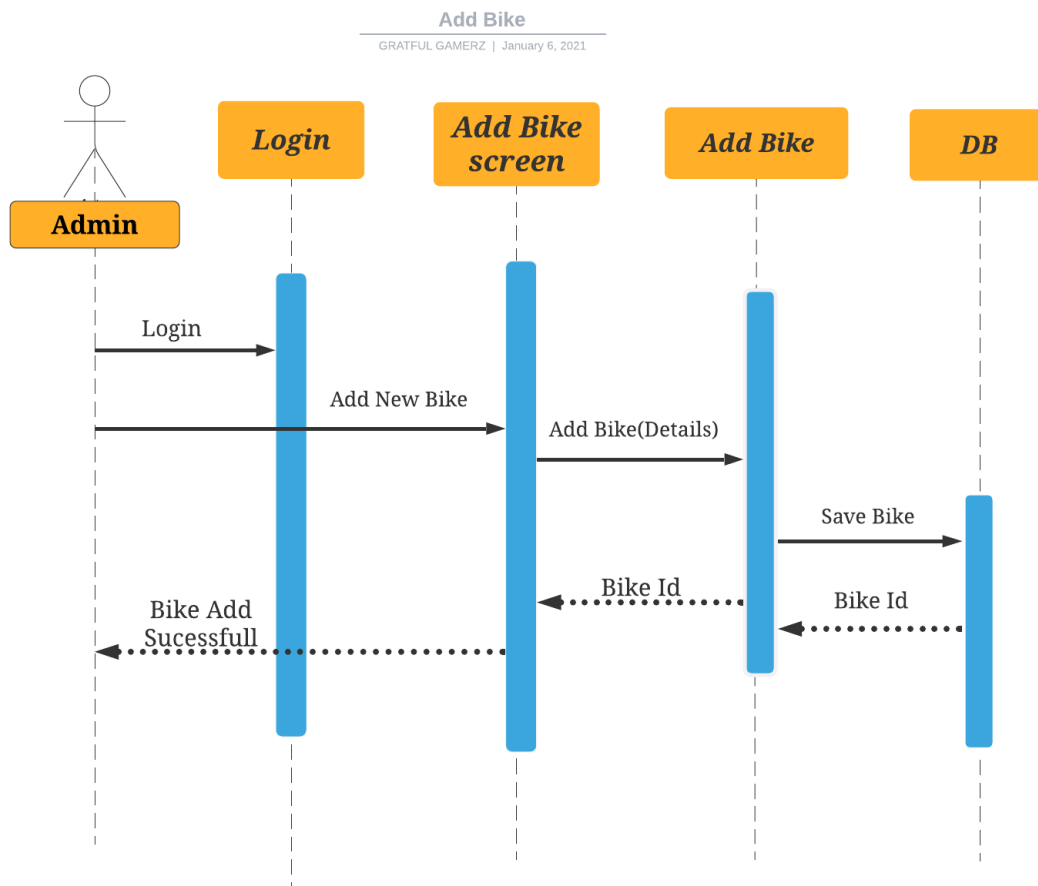


Figure: 3.19: Add Bike

3.4.9 Add Rent Bike

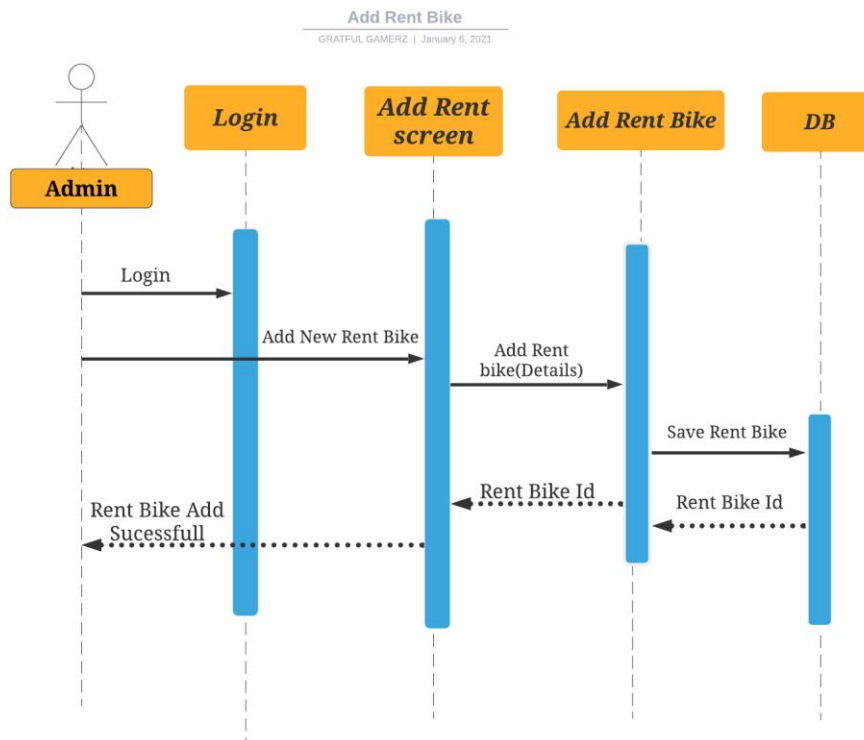


Figure: 3.20 : Add Rent Bike

3.4.10 Add Parts

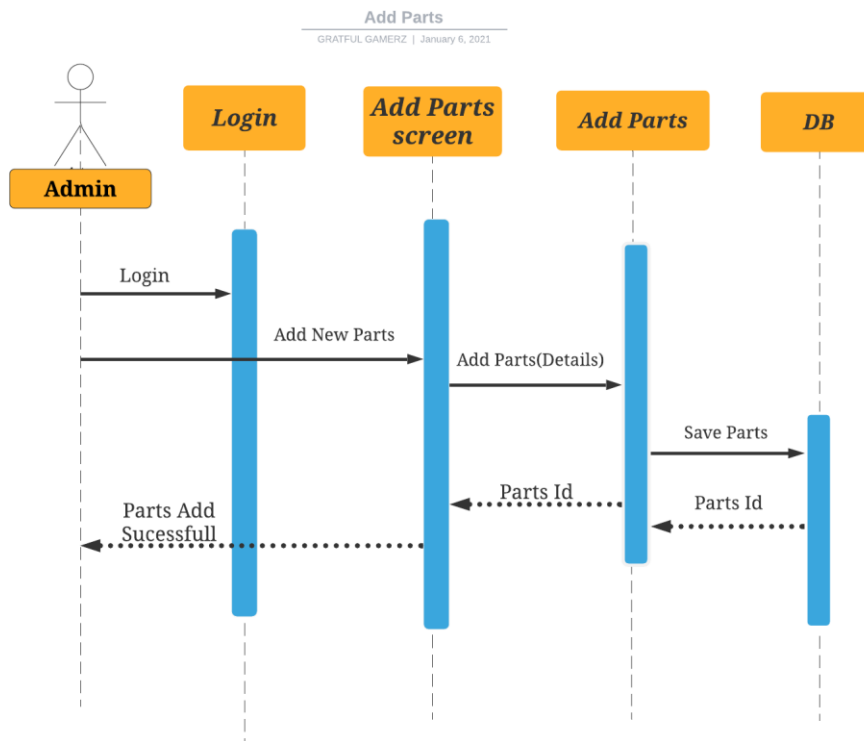


Figure: 3.21: Add Parts

CHAPTER-04

System Design Specification

4.1 Entity Relationship Diagram:

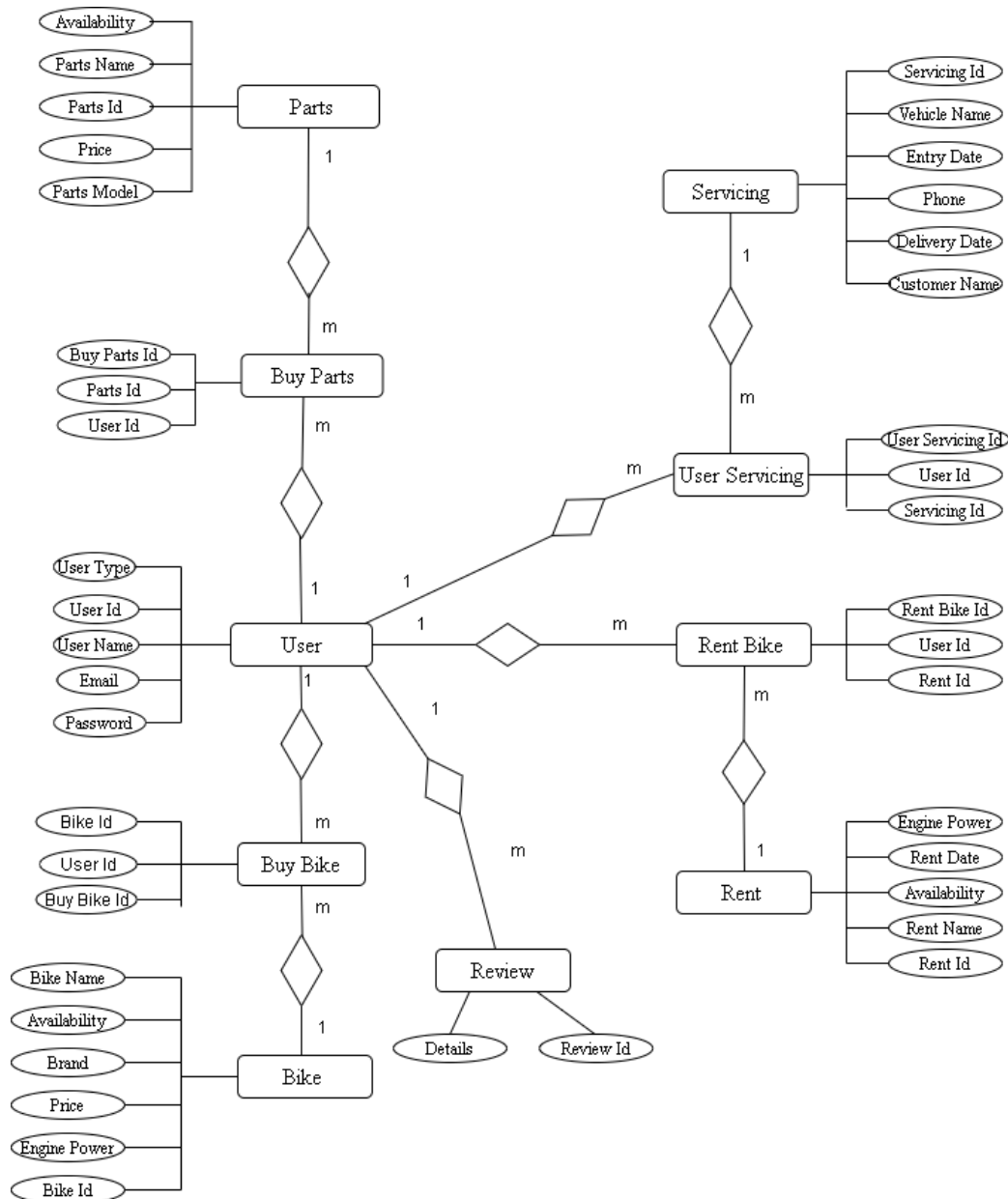


Figure 4.1: ER Diagram

4.2 Class Diagram

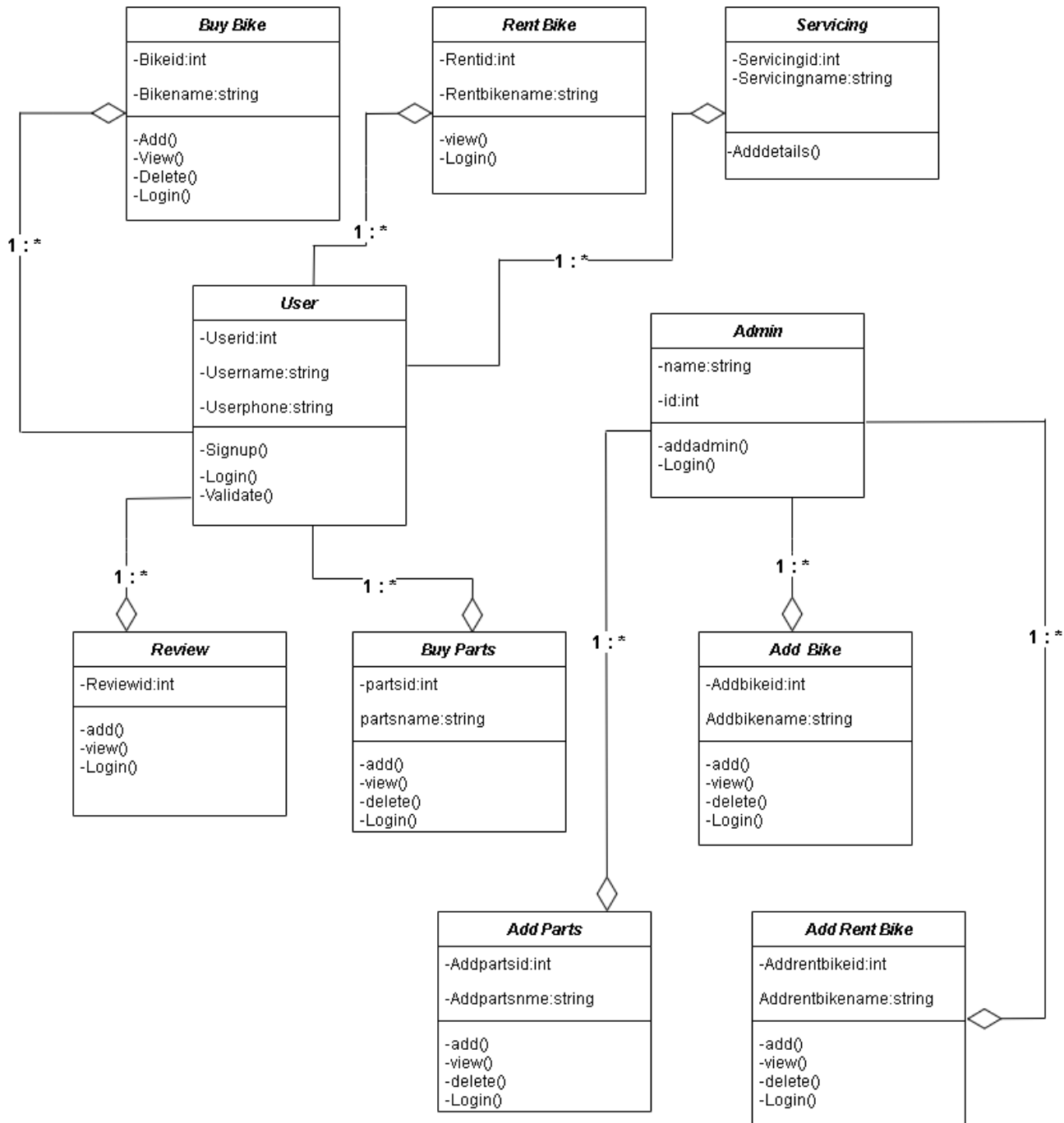


Figure 4.2: Class Diagram

4.3 Development Tools and Technology

4.3.1 User Interface Technology

4.3.1.1: HTML, HTML5

4.3.1.2: CSS3, CSS

4.3.1.3: Bootstraps-4

4.3.1.4: JavaScript, Font Awesome

4.3.2 Implementation Tools and Platforms

4.3.2.1 : Django V2.1

4.3.2.2 : Xampp V3.2.4

4.3.2.3 : Pycharm professional

4.3.2.4 : Windows CMD

Chapter - 05

System Testing

5.1 Testing Features

5.1.1 Feature to be tested

Title	Description	Priority
Registration	User must be sign up to the system if buy anything.	High
Login	Must be login into the system.	High
Buy Bike	User can buy any bike which are listed in system	High
Rent Bike	User can rent any bike, any time	High
Servicing	User can service their bike.	High
Buy Parts	User can buy any bike parts which they need.	High
Add Bike	Admin can update his bike list and details.	High
Add Parts	Admin can add parts with customer demand.	High
Add Rent Bike	Admin can add more rentable for more customer.	High

Table 5.1: Feature to be tested

5.1.2 Feature not to be listed

Title	Description	Priority
Review	User can read any product review which are posted by other user	

Table 5.2: Feature not to be tested

5.2 Testing Strategies

5.2.1 Test Approach

1. The whole system tested manually.
2. System testing based on Project/Thesis Committee and Faculty.
3. This system test on user activities and admin activities.

5.2.2 Pass or Fail Criteria

1. Component Pass/Fail Criteria: - The test will be passing the cases meet the item plan necessities.
2. Integration Pass/Fail Criteria: - The test will be passing cases meets the item design necessity or flopped if not.

5.2.3 Suspension and Resumption

1. Regression Testing: - The system will be work appropriately after each difference in the system.
2. Database Change: - This system doesn't work appropriately in the event that you change the information base name.
3. System Design Changes: - If you change the design of your system will work properly.

5.2.4 Testing Schedule

The testing schedule of my project is given below:

Test Phase	Time
Test Plan	1 Week
Test Specification	2 Week
Test Specification Team	1 Week
Component Test	3 Week
Integration Test	2 Week
System Testing	4 Week

Table 5.3: Testing Schedule

5.3 Testing Environment

- Browser: - Firefox, Google Chrome
- Corei7, Ram# 8GB, HDD#1TB

5.4 Test Case

5.4.1 Test Case 1 (User Login)

Test Case: - 01	Test Case Name: Login
System: User Login	Subsystem: N/A
Designed By: - Rasel Ahamed	Design Date: 10-10-2020
Execute By: - Rasel Ahamed	Execution date: - 15-10-2020

Table 5.4: Test Case 1 (User Login)

Step	Action	Expected System response	Pass/Fail	Comment
1	At the point when the system user tops off the user login field and snap the login button	On the off chance that user don't enter email id, at that point show required email id.	Pass	Email Field are required
2	When user enters emails like a@gmail.com	The system shows require email id.	Pass	Need the valid email id.

Table 5.5: User Login

5.4.2 Test Case 2 (Buy Bike)

Table 5.2: Test Case 1 (User Login)

Test Case: - 02	Test Case Name: Submit Proposal
System: Buy Bike	Subsystem: N/A
Designed By: - Rasel Ahamed	Design Date: 17-10-2020
Execute By: - Rasel Ahamed	Execution date: - 22-10-2020

Table 5.6: Test Case (Buy Bike)

Step	Action	Expected System response	Pass/Fail	Comment
1	When user buy bike button then he fills up the field.	If user missing any field then this system shows required field.	Pass	All field require
2	When user add to card the product add his cart.	Show the how amount he pay.	Pass	Must be login.

Table 5.7: Buy Bike

CHAPTER – 06

User Manual

6.1 User Manual

[Ecom](#) [Store](#) [Service](#) [Buy_parts](#) [Login](#)



R15

[Add to Cart](#) [View](#) **\$50000.00**



yamma

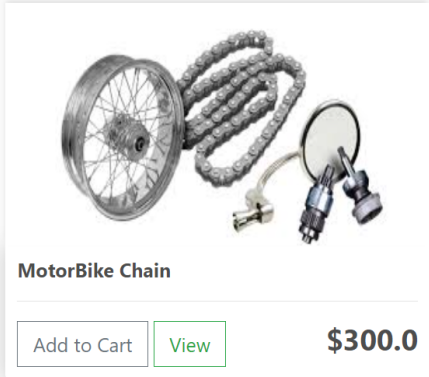
[Add to Cart](#) [View](#) **\$444444.00**



Suzuki

[Add to Cart](#) [View](#) **\$200000.00**

[Ecom](#) [Store](#) [Service](#) [Buy_parts](#) [Login](#)



MotorBike Chain

[Add to Cart](#) [View](#) **\$300.0**

EcomStoreServiceBuy_parts

Login

Home>>service

service

Phone:

Type:

Bike type:

submit

Django administration

WELCOME, **RASEL** [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Site administration

AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add Change
Users	+ Add Change

STORE	
Buy_parts	+ Add Change
Customers	+ Add Change
Faqs	+ Add Change
Order items	+ Add Change
Orders	+ Add Change
Products	+ Add Change
Reviews	+ Add Change
Services	+ Add Change
Shipping addresss	+ Add Change

Recent actions

My actions

None available

Django administration

WELCOME, **RASEL** [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home / Store / Buy_parts

✔ The buy_parts "Chain Cover" was added successfully.

Select buy_parts to change

ADD BUY_PARTS +

Action: 0 of 1 selected

☐ BUY_PARTS

☐ Chain Cover

1 buy_parts

Add faq

Ordernumber:

Question:

Answer:

Status:

[Save and add another](#)[Save and continue editing](#)[SAVE](#)[← Continue Shopping](#)

Items: 0

Total: \$0.00

[Checkout](#)

Item

Price

Quantity

Total

[← Continue Shopping](#)

Items: 0

Total: \$0.00

[Checkout](#)

Item

Price

Quantity

Total

CHAPTER-07

Project Summary

GitHub Link: <https://github.com/Rasel2233/project/tree/master>

7.1 Limitations:

- Cannot send email to the user.
- There is no real time payment system in this system

7.2 Obstacles and Achievements:

In the new era of the world day by day the technologies are update so the Django framework is updated each day. Along these lines, I includes there are some technologic can be wiping out and the venture can be actualized with new innovation. There are too more mistakes while I am built up this system.

7.3 Future Work:

Every Project have some future scope so that in features the application adds some new feature

1. Send email every register user when update product list.
2. Adding text conversation with client to Admin.

APPENDICES-A

REFERENCES

- <https://mogadalai.wordpress.com/2007/05/28/howto-write-a-project-proposal/>
- <https://docs.djangoproject.com/en/3.1/>
- <https://www.w3schools.com/>
- <https://colorlib.com/wp/templates/page/2/>
- <https://guides.lib.berkeley.edu/how-to-write-good-documentation>
- <https://stackoverflow.com/questions/29953463/python-django-user-object-not-found>
- <https://www.sitepoint.com/writing-software-documentation/>

Document Viewer

Turnitin Originality
Report

Processed on: 01-Feb-2021 22:30 +06

ID: 1499143273

Word Count: 4522

Submitted: 1

171-35-1958 By Rasel Ahamed

Similarity Index

24%

Similarity by Source

Internet Sources:	22%
Publications:	1%
Student Papers:	12%

[include quoted](#) [include bibliography](#) [excluding matches < 10 words](#)

mode:

quickview (classic) report

Change mode

[print](#)[refresh](#)[download](#)

5% match (student papers from 12-Apr-2018)

Class: April 2018 Project Report

Assignment: Student Project

Paper ID: [945516662](#)

3% match (Internet from 10-Jan-2020)

<http://dspace.daffodilvarsity.edu.bd:8080>

3% match (Internet from 01-Apr-2020)

<https://www.slideshare.net/RaihanMahmud5/remote-doctor-project-report>

2% match (Internet from 06-Jan-2020)

<http://dspace.daffodilvarsity.edu.bd:8080>

2% match (Internet from 20-Feb-2020)

<http://dspace.daffodilvarsity.edu.bd:8080>

2% match (student papers from 07-Apr-2018)

Class: Article 2018

Assignment: Journal Article

Paper ID: [942459358](#)

1% match (Internet from 28-Jul-2019)

<http://dspace.daffodilvarsity.edu.bd:8080>

1% match (student papers from 26-Nov-2015)

[Submitted to Higher Education Commission Pakistan on 2015-11-26](#)

1% match (student papers from 29-Oct-2008)

[Submitted to University of Greenwich on 2008-10-29](#)

1% match (student papers from 01-May-2016)

https://www.turnitin.com/newreport_classic.asp?lang=en_us&oid=1499143273&ft=1&bypass_cv=1

1/10