



islington college

Acknowledgement

My final year project has helped me to gain technical skills and ideas while working for the time period. The completion of project would not be possible without the guidance of external supervisor Mr. Nishon Tandukar, who inspired me throughout the project and helped me solve problems faced during the implementation and development of my application.

I am also grateful to Islington College that gave me opportunity and support to enhance my skill and knowledge which will be helpful for my future. I would like to thank my internal supervisor Mr. Sugat Man Shakya who helped with the documentation part of Final Year Project.

Abstract

Online platforms, social media, websites, mobile applications has created a new platform for entrepreneurs to establish their business as well as made shopping, booking, renting easier for customers. Online booking, renting, selling has created a virtual market for customer where they can find different products from all around world. There are many people who needs to travel a lot for their business, work, meetings etc. and lot of people cannot afford vehicles so they are force to travel in public transportation. People often hire vehicles if they want to go for outing but it is more like booking a package. Also online vehicle rental web application existing in Nepal does not offer to drive on our own and also does not rent vehicles online. People wants to travel in comfort driving on their own, enjoy and feel like they are driving their own vehicle even they cannot afford to own a vehicle. Acknowledging the requirement of the people this project has been developed. My application provides the feature of booking and renting vehicles online. Also people can drive on their own or in case people don't know how to drive driver is provided. This project mainly focuses on renting vehicles to people who wants to travel with comfort and cannot afford vehicles and are worried about their high maintenance.

Developing a web application or any applications includes many phases, a lot of research, focus, patience and hard work. This project gives brief introduction to current topics, scenarios of both the world and Nepal. Suitable methodology for the project will be chose and task will be carried based on methodology for development of this application and are described in this report. Along with this all the technical terms, programming languages, frameworks which will be used to develop this application are discussed Thus this report contains the requirement gathering, designing, implementation phases and future plans for the project.

Table of Contents

Chapter 1: Introduction	1
 1.1 Introduction to Topic	1
 1.1.1 Current Scenario:.....	2
 World:.....	2
 Nepal:	3
 1.2 Problem Domain:.....	4
 1.3 Scope:	4
 1.4 Aims and Objectives:.....	5
 1.5 Structure of the Report:.....	6
 1.5.1 Background:.....	6
 1.5.2 Development:.....	6
 1.5.3 Testing and Analysis:	6
 1.5.4 Conclusion:	6
Chapter 2: Background	6
 2.1 End User:.....	6
 2.2 System Architecture.....	7
 2.3 Functions and Features.....	8
 2.4 Similar Systems:	9
 2.4.1 Rent Rabbit.....	9
 2.4.2 Sixt.....	10
 2.4.3 Vehicles Hire Nepal	10
 2.4.4 Hotwire.....	11
 2.4.5 Transports in Nepal.....	12
 2.4.6 Comparison	13
 2.4.7 Conclusion.....	14
 2.5 Review of Technical Aspects.....	16
Chapter 3: Development	17
 3.1 Considered Methodology.....	17
 3.2 Selected Methodology	18
 3.3 Phases of methodology.....	19

3.3.1 Requirement gathering.....	19
3.3.2 Analysis and Design	19
3.3.3 Implementation.....	19
3.3.4 Testing	19
3.3.5 Evaluation	19
3.4 System Requirement Specification.....	19
 3.4.1 Purpose	19
 3.4.2 Intended Audience and Reading Suggestions.....	20
 3.4.3 Project Scope.....	20
 3.4.4 Overall Description.....	20
 3.4.5 Features.....	21
 3.4.6 Functional Requirements	21
 3.4.7 External Interface Requirements.....	25
 3.4.8 Nonfunctional Requirements.....	25
3.5 Implementation of Methodology	27
 3.5.1 Iteration 1:.....	27
 3.5.1.1 Requirement Gathering:	27
 3.5.1.2 Analysis and Design:.....	27
 3.5.1.3 Implementation:	33
 3.5.1.4 Testing:	34
 3.5.1.5 Evaluation:	34
 3.5.2 Iteration 2:.....	35
 3.5.2.1 Requirement Gathering:	35
 3.5.2.2 Analysis and Design:.....	35
 3.5.2.3 Implementation:	51
 3.5.2.4 Testing:	56
 3.5.2.5 Evaluation:	56
 3.5.3 Iteration 3:.....	57
 3.5.3.1 Requirements:	57
 3.5.3.2 Analysis and Design:.....	57
 3.5.3.3 Implementation:	82

3.5.3.4 Testing:	86
3.5.3.5 Evaluation:	87
3.5.4 Iteration 4:	88
3.5.4.1 Requirements:	88
3.5.4.2 Analysis and Design:	88
3.5.4.3 Implementation:	97
3.5.4.4 Testing:	101
3.5.4.5 Evaluation:	101
3.6 Survey Results	102
3.6.1 Pre-survey results	102
3.6.2 Post-survey results	103
Chapter 4: Testing and Analysis of progress	104
4.1 Iteration 1 Testing	104
4.2 Iteration 2 Testing	105
4.3 Iteration 3 Testing	126
4.4 Iteration 4 Testing	164
4.5 Critical Analysis	173
Chapter 5: Conclusion	174
5.1 Legal, Social and Ethical issues	174
5.1.1 Legal issues	174
5.1.2 Social issues	175
5.1.3 Ethical issues	175
5.2 Advantages	175
5.3 Limitations	177
5.4 Future work	178
Chapter 6: References	179
Chapter 7: Bibliography	181
Chapter 8: Appendix	184
8.1 Appendix A: Pre-survey	184
8.1.1 Pre-survey Form	184
8.1.2 Sample of filled Pre-survey forms	184

8.1.3 Pre-survey result.....	185
8.2 Appendix B: Post-survey	190
8.2.1 Post- Survey Form.....	190
8.2.2 Sample of filled Post-survey forms.....	191
8.2.3 Post-survey result	192
8.3 Appendix C: Sample Codes	196
8.3.1 Sample Code of the UI.....	196
8.3.2 Sample code for the Automation Script	201
8.4 Appendix D: Designs	206
8.4.1 Gantt Chart	206
8.4.2 Work breakdown structure.....	208
8.4.3 Algorithms and flowcharts	215
8.4.4 3D Modelling	217
8.4.5 Use case.....	219
Iteration 2 High level Use case.....	219
Expanded Use Case	220
Iteration 3 High level Use case.....	223
Expanded Use Case	225
Iteration 4 High Level Use case	233
Expanded Use case.....	233
8.4.6 Wireframe.....	235
8.4.7 Designs.....	239
8.5 Appendix E: Screenshots of the system	245
8.6 Appendix F: User Feedback	253
8.6.1 Client Approval Letter.....	253
8.6.3 Sample of filled User feedback forms	254
8.7 Appendix G: Development.....	256
8.7.1 Considered Methodology	256
8.7.2 Selected Methodology	259
8.8 Appendix H: Initial Software Requirement Specification	260
8.9 Appendix I: Proposal.....	260

List of figures:

Figure 1: System Architecture	9
Figure 2: Views of Rent Rabbit	11
Figure 3: View of Sixt	12
Figure 4: Views of Vehicle Hire Nepal	13
Figure 5: Views of Hotwire	14
Figure 6: Views of Transports in Nepal	15
Figure 7: Class Diagram	33
Figure 8: ER Diagram	34
Figure 9: Wireframe of application	35
Figure 10: Wireframe for admin panel	36
Figure 11: Wireframe for customer panel	36
Figure 12: Wireframe of Login Form	37
Figure 13: Wireframe of registration form	38
<i>Figure 14: Final Use case of whole system.</i>	39
Figure 15: Setup database connection	40
Figure 16: Database created	40
Figure 17: Wire frame of admin profile	43
Figure 18: Wireframe of adding vehicles	44
Figure 19: Wireframe of List of Added Vehicle	45
Figure 20: Wireframe for Admin forgot password	46
Figure 21: Use case diagram for admin.....	47
Figure 22: Activity Diagram for Admin login	48
Figure 23: Activity Diagram for Admin Forgot password	49
Figure 24: Activity Diagram for Admin CRUD of vehicles	50
Figure 25: Activity Diagram for Admin change password.....	51
Figure 26: Sequence Diagram for Admin Login	52
Figure 27: Sequence diagram for admin crud of vehicle	53
Figure 28: Sequence diagram for admin change password	54
Figure 29: Sequence diagram for admin forgot password	55
Figure 30: Collaboration Diagram for Admin Login	56
Figure 31: Collaboration Diagram for Admin CRUD of vehicle.....	57
Figure 32: Collaboration diagram for admin change password	58
Figure 33: Collaboration diagram for admin forgot password	59
Figure 34: Login UI	60
Figure 35: Login method	60
Figure 36: Authenticate user api	61
Figure 37: Vehicle CRUD UI	62

Figure 38: Vehicle add and update method	62
Figure 39: Delete vehicle method.....	63
Figure 40: Wireframe of Available Vehicles for customer	66
Figure 41: Wireframe of Reserved Vehicle for customer	67
Figure 42: Wireframe of feedback for customer panel	67
Figure 43: Wireframe of profile for customer panel	68
Figure 44: Wireframe for Customer Forgot Password	69
Figure 45: Use case diagram for customer	70
Figure 46: Activity Diagram for Customer registration	71
Figure 47: Activity Diagram for Customer Login	72
Figure 48: Activity Diagram for Customer Forgot password	73
Figure 49: Activity diagram for customer password change	74
Figure 50: Activity Diagram for Book vehicles.....	75
Figure 51: Activity Diagram for Rent vehicles	76
Figure 52: Activity Diagram for Sending feedbacks	77
Figure 53: Activity Diagram Search vehicles	78
Figure 54: Sequence diagram for customer registration	79
Figure 55: Sequence diagram for customer login	80
Figure 56: Sequence diagram for customer password change	81
Figure 57: Sequence diagram for customer forgot password	82
Figure 58: Sequence diagram for booking vehicle	83
Figure 59: Sequence diagram for rent vehicle	84
Figure 60: Sequence diagram for sending feedback	85
Figure 61: Collaboration diagram for customer registration	86
Figure 62: Collaboration diagram for customer login.....	87
Figure 63: Collaboration diagram for customer password change	88
Figure 64: Collaboration diagram for customer forgot password	89
Figure 65: Collaboration diagram for booking vehicle	90
Figure 66: Collaboration diagram for rent vehicle	91
Figure 67 : Collaboration diagram for feedback	91
Figure 68: Register form UI	92
Figure 69: User register method	92
Figure 70: Book vehicle UI	93
Figure 71: Book vehicle method	93
Figure 72: Rent vehicle UI	94
Figure 73: Rent Vehicle method	94
Figure 74: Search vehicle UI	95
Figure 75: Search Vehicle method	95
Figure 76: Wireframe of view feedback for admin	98
Figure 77: Wireframe of Vehicle Report for admin panel	99
Figure 78: Wireframe for Email Template.....	100

Figure 79: Use case diagram for Admin -2	101
Figure 80: Activity Diagram for Admin view feedback	102
Figure 81: Activity Diagram for Admin view vehicle reserve report	103
Figure 82: Sequence diagram for view feedback	104
Figure 83: Sequence diagram for view vehicle report	105
Figure 84: Collaboration diagram for view feedback	106
Figure 85: Collaboration diagram for view vehicle report	106
Figure 86: Email Template for sending email	107
Figure 87: API to send email	108
Figure 88: Vehicle reserved by client UI	109
Figure 89: Vehicle Approve and Reject method	109
Figure 90: Pre-survey result 1	111
Figure 91: Pre-survey result 2	111
Figure 92: Post-survey result -1	112
Figure 93: Post-survey result -2	112
Figure 94: Frontend project setup	113
Figure 95: Backend project setup	114
Figure 96: Admin login UI	115
Figure 97: Admin credentials	116
Figure 98: Redirect to admin dashboard	116
Figure 99: Click 'Profile'	117
Figure 100 : Change admin password	118
Figure 101: Display updating credentials	118
Figure 102: Successful login after changing password	119
Figure 103: Click vehicle action	120
Figure 104 : Vehicle adding form	121
Figure 105: Adding vehicle details	121
Figure 106: Displaying added vehicles	122
Figure 107: Editing vehicles by clicking edit button	123
Figure 108: Editing highlighted text field	124
Figure 109: After updating vehicle	124
Figure 110 : Deleting vehicles by clicking delete button	125
Figure 111: Displaying vehicle deleted message	126
Figure 112: List of vehicles after deleting one vehicle.....	127
Figure 113: Logging out from admin panel	128
Figure 114: Redirecting to login page after log out	128
Figure 115: Click on 'Forgot Password?'	129
Figure 116: Enter Email on the given field	130
Figure 117: Email sent with OTP code success message	131
<i>Figure 118: OTP sent via email to reset password</i>	131
Figure 119: Message displayed in case of wrong token entered.	132

Figure 120: Resetting password with valid OTP	133
Figure 121: Reset password success message	134
Figure 122: Login success after resetting password	134
Figure 123: Displaying message when login with invalid username and password	135
Figure 124: Displaying error message for empty text field login	136
Figure 125: Displaying error for empty text field while adding vehicle	137
Figure 126: User registration form	138
Figure 127: Verifying email address	139
Figure 128: Verification email received by customer	140
Figure 129: Verifying email while registering user	141
Figure 130: user details stored in database after registration.....	141
Figure 131: Displaying error for empty text field while registering as new user	142
Figure 132: User login	143
Figure 133: Redirected to customer dashboard.....	144
Figure 134: Logging out from user panel	145
Figure 135: Redirecting to login page after log out	145
Figure 136: Click on 'Forgot Password?'	146
Figure 137: Enter Email on the given field	147
Figure 138: Email sent with OTP code success message	148
<i>Figure 139: OTP sent via email to reset password</i>	149
Figure 140: Message displayed in case of wrong token entered.	150
Figure 141: Resetting password with valid OTP	151
Figure 142: Reset password success message	152
<i>Figure 143: Login success after resetting password</i>	153
Figure 144: Click User 'Profile'	154
Figure 145 : Change user password	155
Figure 146: Display updating credentials	155
Figure 147: Successful login after changing password	156
Figure 148: Click on book button to book vehicle	157
Figure 149: Select destination coordinates via map	158
Figure 150: Fill up booking details and click button book.....	159
Figure 151: Vehicle reserved successful message	160
Figure 152: Displaying vehicle reserved by user	160
Figure 153: Click on rent button to rent vehicle	161
Figure 154: Displaying success message.	162
Figure 155: Displaying rented vehicle	162
Figure 156: Automatic price calculation for rent and book -1	163
Figure 157: Automatic price calculation for rent and book -2	164
Figure 158: Click on Available Vehicles.....	165
Figure 159: Displaying available vehicles.....	166
Figure 160: Before search	167

Figure 161 : Displaying vehicles according to search by Vehicle Name	168
Figure 162: Displaying vehicles according to search by Vehicle Type Two-Wheeler	169
Figure 163: Displaying vehicles according to search by Vehicle Name Four-Wheeler	170
Figure 164: Displaying feedback form on button click	171
Figure 165: Sending feedback	172
Figure 166: Feedback stored in database	172
Figure 167: Displaying error for empty text field while booking vehicle	173
Figure 168: Displaying error for empty text field while renting vehicle	174
Figure 169: Before cancelling rent or book	175
Figure 170: Available vehicles before rent or book cancel	176
Figure 171: Rent Request from clients	179
Figure 172: Press Approve button to rent vehicle	180
Figure 173: Request deleted after button click	180
Figure 174: Displaying approval confirmation email	181
Figure 175: Press Reject button to rent vehicle	182
Figure 176: Displaying rejection success message	182
<i>Figure 177: Displaying rejection email</i>	183
Figure 178: View vehicles that are on rent	184
Figure 179: Enter vehicle name or number	185
Figure 180: Search by vehicle name	185
Figure 181: Search by vehicle number	186
Figure 182: Before Sorting request details	187
Figure 183: Sorting by Vehicle Number	187
Figure 184: Sorting by Price	188
Figure 185: Sorting by Rent Duration	188
Figure 186: View feedback sent by users.....	189
Figure 187: Pre-Survey Form	201
Figure 188: Sample of filled Pre-Survey form	202
Figure 189: Pre-survey results 1 & 2	203
Figure 190: Pre-survey results 3 & 4	204
Figure 191: Pre-survey results 5 & 6	205
Figure 192: Pre-survey results 7 & 8	206
Figure 193: Post- survey form	207
Figure 194: Sample of filled Post-Survey form	208
Figure 195: Post-Survey result 1 & 2	209
Figure 196: Post-Survey results of 3 & 4	210
Figure 197: Post-Survey results of 5 & 6	211
Figure 198: Post-Survey results of 7 & 8	212
Figure 199: Post-Survey results of 9	213
Figure 200: Post-Survey results of 10.....	213
Figure 201: code of UI vehicle available	214

Figure 202: Vehicle Available layout	214
Figure 203: code of UI login	215
Figure 204: Login Form layout	215
Figure 205: code of UI vehicles reserved by client	216
Figure 206: Vehicle Reserved by client layout	216
Figure 207: code of UI vehicle action.....	217
Figure 208: Vehicle action layout	217
Figure 209: code of UI register user	218
Figure 210: Registration form layout	218
Figure 211: Automation script of table user.....	219
<i>Figure 212: Data populated into table user</i>	220
Figure 213: Automation script of table vehicle	221
Figure 214: Data populated into table vehicle	222
Figure 215: Automation script of table feedback.....	223
Figure 216: Data populated into table feedback	223
Figure 217: Initial Gantt chart	224
Figure 218: Final Gantt chart	225
Figure 219: Initial Work breakdown structure	226
Figure 220: Final Work breakdown structure	227
Figure 221: Final WBS for Iteration 1.....	228
Figure 222: Final WBS for Iteration 2.....	229
Figure 223: Final WBS for Iteration 3.....	230
Figure 224: Final WBS for Iteration 4.....	231
Figure 225: Flowchart of whole system	233
Figure 226: 3D modelling of Admin panel	234
Figure 227: 3D modeling of registration form	235
Figure 228: 3D modelling of rent vehicle	235
Figure 229: Initial Wireframe of Admin panel	255
Figure 230: Initial Wireframe of Customer panel	256
Figure 231: Initial Wireframe of Login form	257
Figure 232: Initial Wireframe of Registration form	258
<i>Figure 233: Initial Sequence Diagram for Admin</i>	259
Figure 234: Initial Sequence Diagram for Customer	260
<i>Figure 235: Initial Collaboration Diagram for Admin</i>	261
Figure 236: Initial Collaboration Diagram for Customer	262
Figure 237: Initial ER Diagram	263
Figure 238: Initial Activity Diagram for Admin	264
<i>Figure 239: Initial Activity Diagram for Custom</i>	265
Figure 240: Home page	266
Figure 241: Login Page	266
Figure 242: User registration form page	267

Figure 243: Customer dashboard.....	268
Figure 244: Admin dashboard	268
Figure 245: Vehicle action page	269
Figure 246: Vehicles reserved by client	269
Figure 247: Vehicle rented by client	270
Figure 248: Admin Profile page	270
Figure 249: View Feedback page	271
Figure 250: Reserved vehicles	271
Figure 251: Feedback form	272
Figure 252: Available vehicle page	272
Figure 253: About page	273
Figure 254: Client approval letter	274
Figure 255: User feedback form	275
Figure 256: Filled User feedback form	277
Figure 257: Phases of waterfall methodology	278
Figure 258: Phases of prototype methodology	280
Figure 259 : Phases of Iterative and incremental methodology	281

List of Tables:

Table 1: Comparison between similar systems	16
Table 2: Frontend compile testing	113
Table 3: Backend compile testing	114
Table 4: Test case for Admin login	115
<i>Table 5: Test case for password change</i>	117
<i>Table 6 : Test case for adding vehicles by admin</i>	120
Table 7: Test case for updating vehicles	123
Table 8: Test case for deleting vehicles	125
Table 9: Test case for Admin logout	128
Table 10: Test case for Forget password	129
Table 11: Test case for invalid username or password	135
Table 12: Test case for empty login fields	136
Table 13: Test case for empty fields while adding vehicle	137
Table 14: Test case for user registration	138
Table 15: Test case for empty registration form	142
Table 16: Test case for registered user login	143
Table 17: Test case for User logout	145
Table 18: Test case for User Forgot password	146
Table 19: Test case for user password change	154
Table 20: Test case for booking vehicles	157
Table 21: Test case for renting vehicles	161
Table 22: Test case for automatic price calculation	163
Table 23: Test case for viewing available vehicles	165
Table 24: Test case for viewing search vehicles	167
Table 25: Test case for sending feedback	171
Table 26: Test case for empty text field while booking and renting vehicle	173
Table 27: Test case for cancelling rent and book request	175
Table 28: Test case for rent approval confirmation email	179
Table 29: Test case for rent request rejected email	182
Table 30: Test Case for viewing report of vehicle on rent	184
Table 31: Test case for searching requests	185
Table 32: Test case for sorting request details	187
Table 33: Test case for viewing feedback	189

Chapter 1: Introduction

1.1 Introduction to Topic

With the growing technology and digitalization there are approximately 4.57 billion people who are active internet users as of April 2020. Internet has not only helped people to come closer with the family and friends all around the world but also helped many people start their own business. Internet has generated online business platform as well as making the world digital. Those online business includes web applications like vehicle rental which has market value of \$78.7 billion by now and is expected to be more in the coming years. (GlobeNewswire, oct 14,2019)

Vehicle rental web application helps people with comfortable travelling as well as gives the feeling of owning and driving their own vehicles. Users can rent vehicle of their own choice and travel in comfort. Online vehicle rental is quite popular in western countries and has a huge market value. However there are some rental companies in Nepal which are providing vehicle renting services but are still practicing traditional way of renting with lots of paperwork.

This project was developed addressing the state of the rental system in context of Nepal. This application will help the user to rent and book vehicles for various purposes like business, work, meeting, tourism etc. It not only benefits the vehicle seeker and business owner but also contributes in the economy of the country. Vehicle rental system can generate large economy as well as employment for people like drivers and mechanics etc. It also helps to encourage local people to travel in different parts of Nepal and promotes domestic or internal tourism.

Since all most all the industry of the world has been touched by digitalization and advance technology vehicle rental industry is not left out. In context of Nepal online vehicle rental system enhances the business process by giving a bigger platform for exposure of their business along with quality customer services like online registration, online booking and renting services.

1.1.1 Current Scenario:

World:

According to market research on global car and vehicle rental market published by P&S intelligence the size of this market value was estimated to be \$78.7 billion in 2018 which was predicted to reach around \$122.6 billion by 2024. Luxury cars category has been witnessing the fastest growth as increased in usage by business professionals. The vehicle rental market is globally recognized due to its convenient feature to rent a car and enjoy all the facilities without actually owning it also private or personal vehicles are high investment which includes vehicle cost, fuel cost, parking, servicing and maintenance charges and insurance cost. Not only luxury cars are covering the rental markets but also electric vehicles demands has been raised worldwide. In India Zoom India Pvt. Ltd in 2018 has been deploying electric cars by associating with Mahindra & Mahindra Ltd. Countries like U.S, China , Germany , UK, Canada has highest market for car rental system globally.

(GlobeNewswire, oct 14,2019)

Nepal:

In current scenario Nepal automobile market has grown drastically after 2015 earthquake and blockade however the vehicle rental market has not been progressing compared to the vehicle rental market of US, Canada, China etc. However there are some Rental Companies providing the service of renting vehicles for tourism purposes with option of different packages. These companies allows to rent and book vehicles and also provides experience drivers. However, these companies are more related to booking a vehicle based on selection of package tours. (Vehicle Hire Nepal, 2018)

Some of the sites based in Nepal are: Himalayan Car Rental Nepal, Vehicle Rental Nepal etc.

1.2 Problem Domain:

Nepal is way behind in the field of the technology when compared to other developed countries. The online vehicle rental system in the world has successfully marked it's placed with \$78.7 million worth market value in the online business industry. But in our country online vehicle rental industry has not been able to grow and progress. In the current context of Nepal there are limited number of vehicle rental web application in the market that allows them to rent vehicles and drive on their own.

The trend of renting vehicles has changed a lot. Most of the countries have made renting vehicles process completely digital but this trend has still not completely taken over Nepal. One the biggest problem is lack of transparency and trust issues in online services which made it difficult for this kind of industry to grow. Most of the people in Nepal cannot afford their personal vehicle and are compelled to travel with discomfort. Even if they want to hire vehicles they need to visit rental office and need to submit a lot of documents and paperwork which makes renting process long, time consuming and frustrating. Also there only few web application which gives online vehicle rental services but are does not give good online renting experience and services. In this situation this project can be considered valuable for fulfilling the need of online vehicle rental application in the current Nepali market.

1.3 Scope:

The world has been progressing drastically in the field of technology. People are constantly working on different technology that would help humankind to ease their work and doing research to benefit human and their lifestyle. We want everything thing digital and every work to be carried out in a second for example cashless transaction. Development of online payment platform, mobile banking, ATM cards has made cashless

tractions possible which is not only easy, and quick but also can track our transaction's and expenses. We also want online booking, renting and selling of various products.

The global market of Vehicle rental is worth more than \$78.7 billion today which has not only made travelling of people easier but has also increased tourism. And due to lack of proper vehicle rental system in Nepal, there is a huge scope for this kind of application as Nepal is visited by many tourists every year. Online vehicle rental system gives business owner a good exposure about their business not only to the local areas but also globally.

This application focuses on normal people who wants to travel with comfort but cannot afford their own vehicle .This will help people to travel in comfort, save time without waiting for public vehicle to arrive, avoid the crowd in public vehicles and can enjoy all the facilities like a vehicle owner even without owning a vehicle and without worrying about high maintenance cost, vehicle cost and parking costs etc.

1.4 Aims and Objectives:

This project is aimed to build a web application that will allow people to rent and book vehicle for tourism, business, meeting, and work without owning a vehicle. The idea is to let the users rent vehicles and feel like they are traveling in their own vehicles.

In order to achieve the aim, the following objectives will be met:

- To conduct research and survey on the topic which will help to understand the requirements of the application.
- To select a suitable methodology and implement it to develop the system.
- To learn the frameworks for developing the application and selecting suitable theme and modify it as per the requirements collected from research and surveys.
- To develop web based application that's allows online customer registration, booking and renting of vehicles and for the company to manage the vehicles and customers effectively.

1.5 Structure of the Report:

1.5.1 Background:

Background includes research on the similar types of systems all around the world and the country as well as comparisons among the system. Information about the end users has also been gathered. A brief explanation about the system architecture as well as list of functions and features need to be added in the system are included in this section. Also several tools that will be used in the development phase are also included in the background.

1.5.2 Development:

Development includes discussion of several methodologies considered and selected methodology with justification. Phases of the selected methodology are discussed with the implementation of the methodology. The UML diagrams and wireframes are also included in this section of the report.

1.5.3 Testing and Analysis:

Testing section of the report consist of test cases and screenshots of the different features being tested. The testing is conducted in different iterations as per the selected methodology.

1.5.4 Conclusion:

Overall report is concluded in this part. Legal, social and ethical issues are discussed in this part along with the advantages, limitations and future work of the application.

Chapter 2: Background

2.1 End User:

This application will have multiple end users. The end users of this application will be the general public who are willing to change their travel experiences. It will not only be useful for people who has license and knows how to drive but also for people who don't know how to drive as drivers are provided. The government can also be benefitted by this application as it generates an additional economy as the global market in the world is worth more than \$78.7 billion. These vehicles are added and managed by admin.

End user 1: Vehicle Seeker

Problem

Many people cannot afford their own vehicles so they often use crowded public vehicles to travel which consumes lot of time and makes travelling frustrating and boring. Sometimes people even cancel their plans because of transportation issue.

Solution

This project will have capacity of having more vehicles available to public who wants vehicle on rent. To search and rent vehicles a user has to register to website. They can book and view available vehicles details also can give their feedbacks.

End user 2 Admin

Administrates the web application

2.2 System Architecture

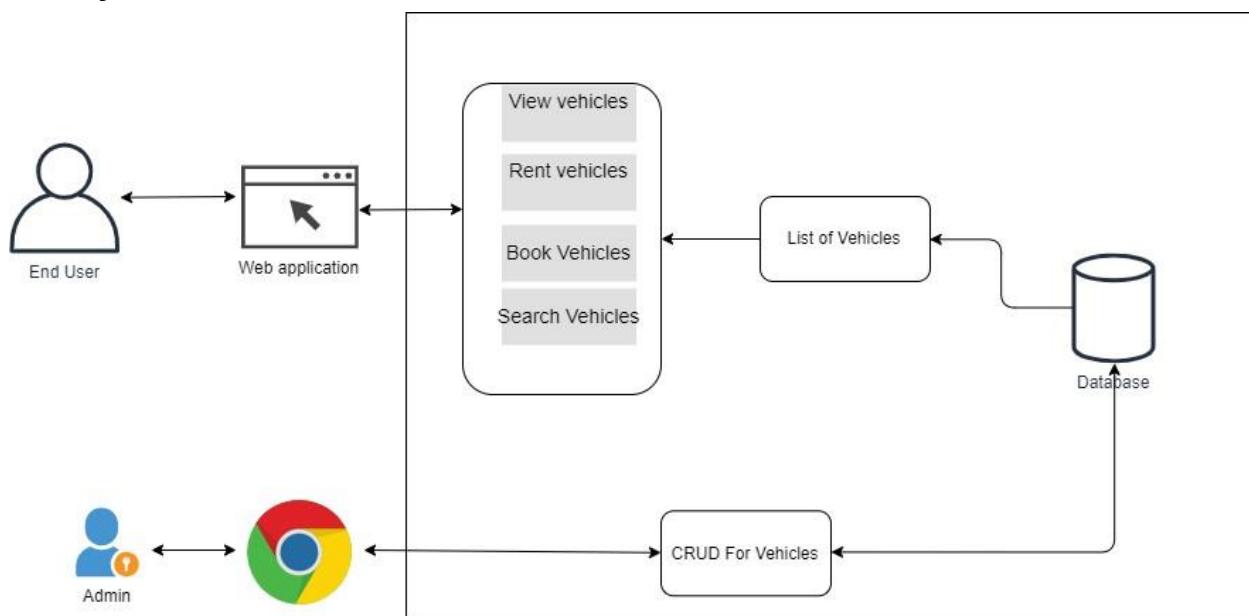


Figure 1: System Architecture

The system consists of a web application which will have list of vehicles available for rent. Users can view the price and details of vehicles as well as give their feedback. Details of vehicles will come from the database. Every time a customer rents a vehicle customer details will be stored in database along with the vehicle details they have rented.

The system will also have an admin page for admin to login into a dashboard to add new vehicles, check feedback and view vehicle status.

2.3 Functions and Features

The primary feature of the application is to rent the vehicles .Besides this, the system will also have the following features:

- Booking vehicles
- View available Vehicles
- Give feedback
- Add vehicles
- Manage vehicle □ View Feedback

2.4 Similar Systems:

2.4.1 Rent Rabbit

Rent rabbit is a web applications that allows user to rent vehicles like car, bike, boat and equipment which includes features like online payments, bookings, searching vehicles. (Rent Rabbit, 2019)

The figure consists of two screenshots of the Rent Rabbit website. The top screenshot shows the search interface for Chicago, IL, USA. It features a search bar with 'Chicago, IL, USA', a 'Day' button, and a 'Hour' button. Below the search bar are filters for 'Price' (radio buttons for '\$0 - \$0', 'Sports', 'Adventure', 'Race', 'Standard', 'Luxury', 'SUV', 'Sedan', 'Hatchback', 'yamaha', 'Hyundai', 'Renault', 'Porsche', 'Audi', 'BMW', 'Benz', 'Ferrari', 'Jaguar'), 'Car Type' (dropdown menu), 'Brands' (checkboxes for various car brands), 'Dates' (input fields for 'Pick Up Date' and 'Drop Date'), and 'Instant Pay' (checkbox). A 'Cancel', 'Reset Filters', and 'Apply' button are at the bottom right. To the right of the search interface is a map of Chicago with price zones marked. The bottom screenshot shows the search results for Los Angeles, CA, USA. It displays three vehicle options: a blue BMW 3 (price \$15.00 / Day), a red Jaguar XJ (price \$30.00 / Day), and an orange Duster (price \$40.00 / Day). Each vehicle has a 'RENT IT' button. To the right of the vehicle cards is a map of Los Angeles with price zones marked (\$15, \$30, \$40, \$10). Navigation buttons for 'Map' and 'Satellite' are also present.

Figure 2: Views of Rent Rabbit

(Rent Rabbit, 2019)

2.4.2 Sixt

Sixt is a web applications that allows user to rent cars of different brands and models which includes features like online payments, bookings, searching vehicles as well as has travel cost calculator. In addition it has also a mobile application. (sixt, 2020)

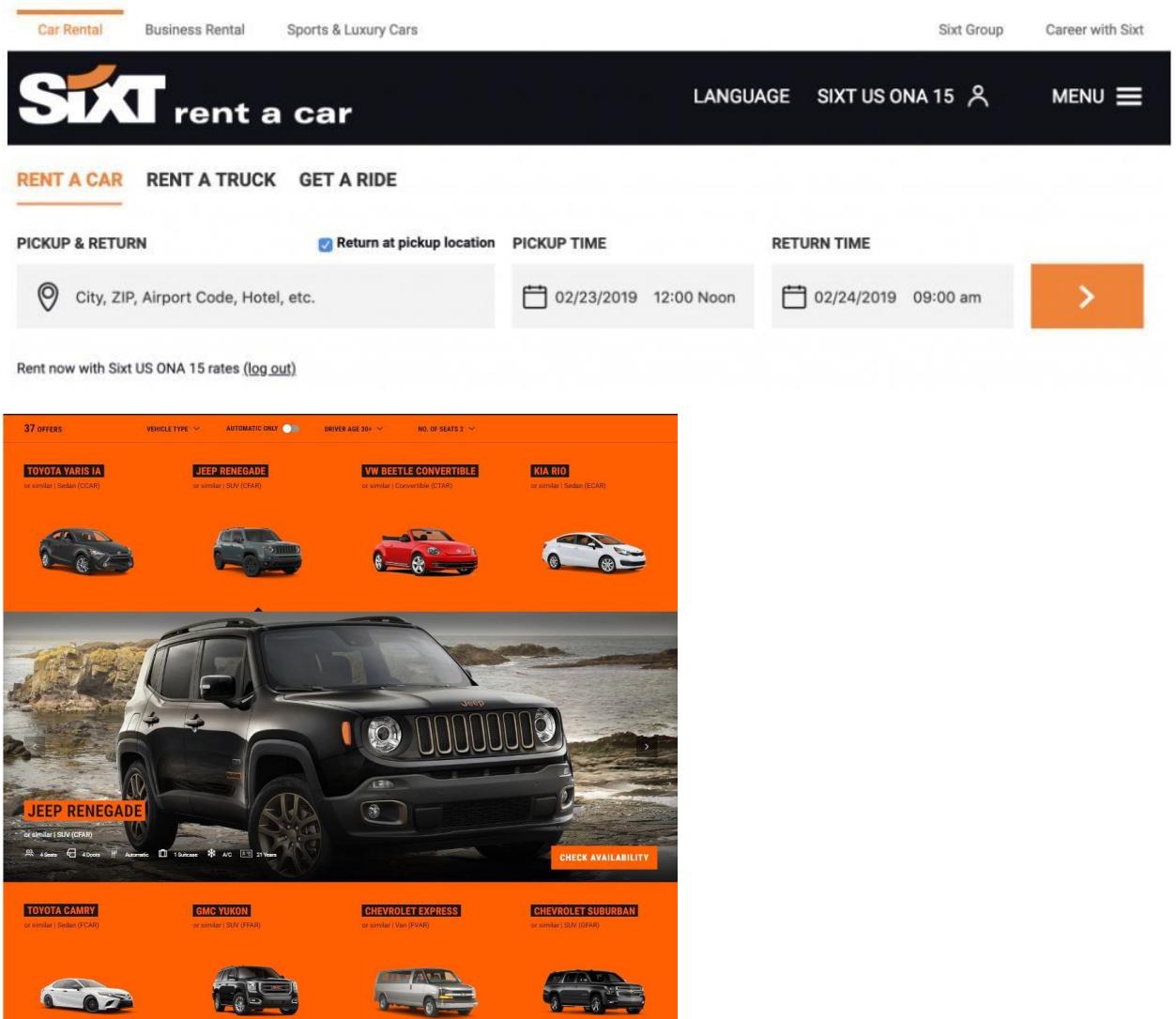


Figure 3: View of Sixt

(sixt, 2020)

2.4.3 Vehicles Hire Nepal

Vehicles Hire Nepal is a web applications that allows user to rent vehicles like car, van, bus which includes features like online bookings and has a details of destinations and their travel cost. (Vehicle Hire Nepal, 2018)

The screenshot shows the homepage of Vehicle Hire Nepal. At the top, there's a navigation bar with links to Home, About Us, Services, Packages, Tariff, Testimonials, Blog, and Contact Us. Below the navigation is a large banner featuring two cars, a red and a blue one, parked in a modern parking lot at night. The word "SERVICES" is prominently displayed in the center of the banner. Underneath the banner, the main content area has a heading "We believe in Quality Service" and a sub-section "Our services". A paragraph of text follows, mentioning travel experience and vehicle load capacities. To the right of this text is a form titled "CHECK VEHICLES AVAILABILITY FOR YOUR TRIP" with fields for "PICK-UP DATE" (mm/dd/yyyy), "DROP-OFF DATE" (mm/dd/yyyy), "TRIP LOCATION" (Location), and "SELECT VEHICLES" (a dropdown menu). A "Check Now" button is at the bottom of the form.

Tariff

S.N.	PARTULARS	KM	CAR	VAN	HIACE	COASTER	BUS	
1	Airport arrival/departure (2 hrs. only)	605	1059	1210	1815	2118		<button>Book Now</button>
2	Mountain Flight	1100	1925	2200	3300	3850		<button>Book Now</button>
3	Disposal charge per hour (min. 4 hrs.)	385	674	770	1155	1348		<button>Book Now</button>
Kathmandu half day sight seeing (3 hrs only) If more than 3 hrs will be charge the disposal charge hour								
5	Dinner	1100	1925	2200	3300	3850		<button>Book Now</button>
6	Ring road inside	1100	1925	2200	3300	3850		<button>Book Now</button>
7		<button>Book Now</button>

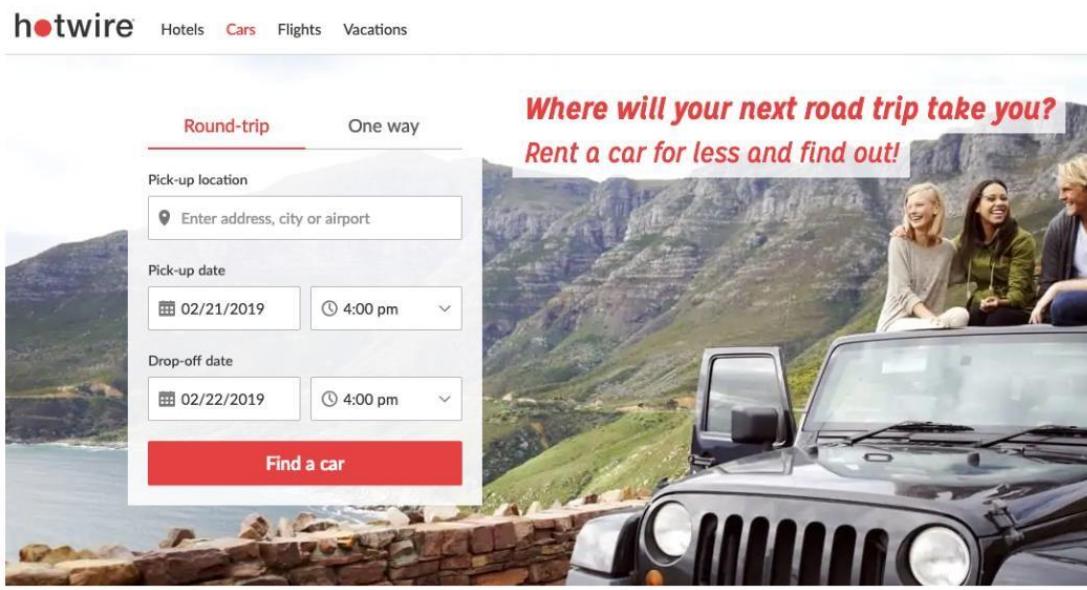
Figure 4: Views of Vehicle Hire Nepal

(Vehicle Hire Nepal, 2018)

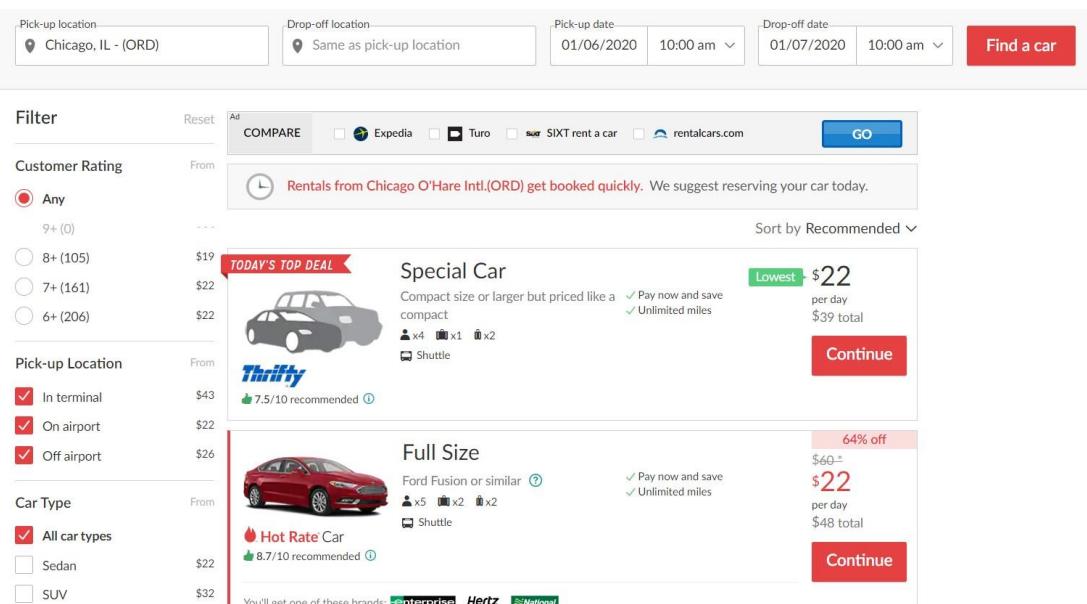
2.4.4 Hotwire

Hotwire is a web applications that allows user to rent cars, book flights, tour packages and hotels of different brands and models which includes features like online payments, bookings, searching vehicles as well as has travel cost calculator.

(Hotwire, Inc., an Expedia Group company, 2020)



The top section of the Hotwire homepage features a search form for car rentals. It includes fields for 'Pick-up location' (with a placeholder 'Enter address, city or airport'), 'Pick-up date' (set to 02/21/2019 at 4:00 pm), 'Drop-off date' (set to 02/22/2019 at 4:00 pm), and a large red 'Find a car' button. To the right is a promotional banner with the text 'Where will your next road trip take you? Rent a car for less and find out!' and an image of three people sitting on the hood of a dark SUV parked on a scenic coastal road.



The search results page shows filters applied: 'Chicago, IL - (ORD)' as the pick-up location, 'Same as pick-up location' as the drop-off location, and specific dates from 01/06/2020 at 10:00 am to 01/07/2020 at 10:00 am. A 'Find a car' button is present. The results are sorted by 'Recommended'. Two car options are listed:

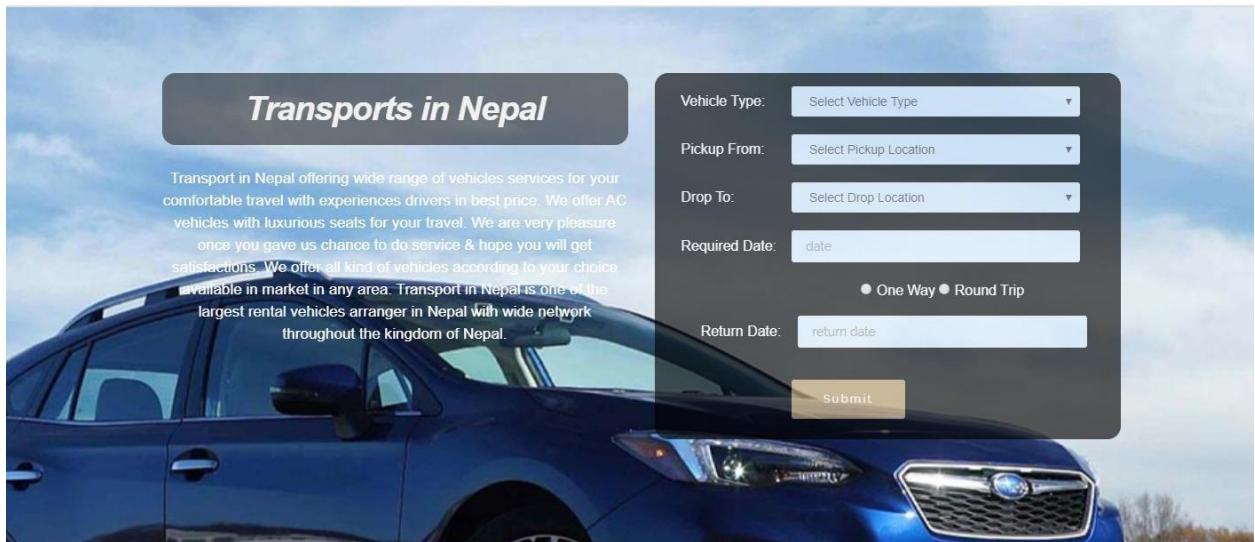
- Special Car**: Compact size or larger but priced like a compact. From \$19 per day (\$39 total). Options: Pay now and save, Unlimited miles. Features: x4 passengers, x1 luggage, x2 fuel. Brand: Thrifty. Rating: 7.5/10 recommended. Price: \$22 per day. Continue button.
- Full Size**: Ford Fusion or similar. From \$26 per day (\$48 total). Options: Pay now and save, Unlimited miles. Features: x5 passengers, x2 luggage, x2 fuel. Brand: Enterprise, Hertz, National. Rating: 8.7/10 recommended. Price: \$22 per day. Continue button.

Figure 5: Views of Hotwire

(Hotwire, Inc., an Expedia Group company, 2020)

2.4.5 Transports in Nepal

Transports in Nepal is a web applications that allows user to rent vehicles like car, bike, scooter, van which includes features like online bookings and has a travel details with travel cost and time durations. (Vehicle Hire Nepal, 2018)



Available Vehicles for Booking						
From	To	Vehicle	Cost	Duration		
Kathmandu	Airport Arrival / Departure	Car	Rs.550	2 hrs	Book it	
Kathmandu	Airport : Mountain Flight	Car	Rs 1000	3 hrs	Book it	
Kathmandu	Airport : Overnight Charge	Car	Rs.1800		Book it	
Kathmandu	Half day city Swoyambhu Stupa (Monkey Temple), Patan durbar squire.	Car	Rs.1250	4 hrs	Book it	
Kathmandu	Half day Bhaktapur tour only	Car	Rs.1250	4 hrs	Book it	
Kathmandu	Pashupati Nath and Boudhanath Stupa tour	Car	Rs.1250	4 hrs	Book it	
Kathmandu	Kritipur/Chovar tour	Car	Rs.1250	4 hrs	Book it	
Kathmandu	Bunamati/Khokana tour	Car	Rs.1250	4 hrs	Book it	

Figure 6: Views of Transports in Nepal

(Transports in Nepal, 2020)

2.4.6 Comparison

Features	Rent Rabbit	Sixt rent a car	Vehicles Nepal	Hire Hotwire	Transports in Nepal	My Application

OnlineRent Vehicles (like cars, motorbike, van etc.)	Yes	Yes (only cars)	No (only four wheeler)	Yes (only car)	No	Yes
Book Vehicles	No	No	Yes	Yes	Yes	Yes
Search Vehicles	Yes	Yes	Yes	Yes	Yes	Yes
Travelcost calculator	No	Yes	No	Yes	No	Yes
Payment gateway integration	Yes	Yes	No	Yes	No	No
Driver arrangement	No	No	Yes	No	Yes	Yes
Multi-language support	Yes	Yes	No	Yes	No	No
Free Cancellation of Booking	Yes	Yes	No	Yes	Yes	Yes
Feedbacks	Yes	Yes	Yes	Yes	Yes	Yes

Table 1: Comparison between similar systems

2.4.7 Conclusion

All the above similar application has their own strong points. Most of the application of foreign countries have many features however compare to application of Nepal there are not many features like online vehicle rent, cost calculator, self-driving permit. Therefore, these features are included in this project.

2.5 Review of Technical Aspects

Java

Java is free and open source object oriented programming language. It is one of the most popular and widely used programming language. One can create any type of applications like web, applications, web services, mobile applications in Java that's why Java programming language was taken into consideration.

(JournalDev, 2020)

Spring Boot

Spring Boot is a framework for Java used to create a micro service which contains all the features of the spring. Due to its production ready environment applications requires minimum configurations. Therefore Spring Boot is chosen for backend development in FYP.

(GeeksforGeeks, 2020)

Angular

Angular is a JavaScript framework used to create reactive Single Page Applications. It is used to create a web applications or a website for both mobile and desktop. Therefore Angular is chosen for frontend development of FYP project.

(java t point, 2018)

MySQL

MySQL is a free and open source Relational Database Management System which is flexible, quick processing and reliable that uses Structured Query Language. Therefore, MySQL database is considered to be used for storing data of the application in database.

(SiteGround, 2019)

Chapter 3: Development

3.1 Considered Methodology

Waterfall:

Justification for consideration:

- It is one of the simplest and easy methodology to use for less experienced developers as in the case of this project.
- The methodology is also appropriate when the requirements and objectives are clear and well understood as the main aim of functions of this project was to rent vehicles by customer. (Tutorialspoint, 2020)

Justification for Rejection:

- It does not allow for incrementing features and functions.
- There might be a situation where I need to go back to development phase but waterfall methodology doesn't allow going back to phases.
- Waterfall does not allow testing until the entire system is completed and there are many independent module that are completed or are being developed.

[View brief explanation of waterfall methodology](#)

Prototype Methodology:

Justification for consideration:

- This methodology allows to change functions and features.
- Useful when the requirements of end users/ client are not clear and are frequently changing. (GeeksforGeeks, 2020)

Justification for Rejection:

- Prototype methodology requires high involvement and interactions of clients/ end-users which is not possible.
- It is not ideal to use prototype since the primary requirements are already gathered.

- It is difficult to develop prototypes quickly as per its nature since each modules might requires different knowledge to be built and some are dependent on other modules too. (Tutorialspoint, 2020)

[View brief explanation of prototype methodology](#)

Iterative Incremental Methodology:

Justification for consideration and selection:

- It is ideal as each modules or components can be developed individually and at the end of each iterations improved version of product is achieved.
- Several iterations can be made to improve the functions and features also addition in functions can be made in iterations.
- Versioning of the software can be done in each iterations.
- Easier to test and debug the software in each iteration. □ Risk analysis is better and can be resolved during each iteration □ Testing can be carried out separately for each iteration.

[View brief explanation of iterative incremental methodology](#)

3.2 Selected Methodology

This project is developed following iterative and incremental approach i.e. gradual increase in addition of functions and features with a new version of product at the end of the cycle. Since there are separate components there is a need for iterative framework because all the modules cannot be developed at the same time in single phase. Hence Iterative incremental method is selected as we can separate one task for one iterations and progress development of each different modules in different iterations.

3.3 Phases of methodology

3.3.1 Requirement gathering

In this phase all the requirements are gathered for the system through research from journals, books and web pages, surveys and interviews with the end users.

3.3.2 Analysis and Design

Once all the requirements are gathered they are analyzed and the main functions and features that are to be added in the application are finalized. In this phase risks are identified and tools to be used are also identified. Also in this phase wireframes, use case diagram, collaboration diagram, sequence diagram and class diagram are designed.

3.3.3 Implementation

Implementation phase is the phase where all the development of the system starts after gathering all the requirements and completing the designing phase. Coding and making the features and the system functional are done in this phase.

3.3.4 Testing

After implementation phase functionality of the system is tested in this phase to make sure that developed system is fully functional without any error.

3.3.5 Evaluation

In this phase the whole system developed is evaluated along with the features and changes needed to be done in the future iteration is analyzed and discussed.

3.4 System Requirement Specification

3.4.1 Purpose

The purpose of SRS document is to describe the requirements of web application for Namaste Nepal Travels and Tours. This document will give all the necessary information to develop this application. This SRS document will cover all the necessary functions and features as well as provides some glimpse of the software development. This document will also cover hardware, software and other technical dependencies.

3.4.2 Intended Audience and Reading Suggestions

The intended audience for this document is Namaste Nepal Travels and Tours owners and the users. In this document the project will be divided into various sections like : Project Description, System Features, External Interface Requirements, and Non Functional Requirements.

3.4.3 Project Scope

This project targets the traditional way of renting vehicle and convert them to web application. The benefits of this project are to be able to book and rent vehicle online. The goal of this project is to make the application easy to use also implements features and requirements gathered from client.

3.4.4 Overall Description

3.4.4.1 Product Features

This program will allow users to book, search, rent vehicles and give feedback online online as well as allows admin to view vehicle reserved by client, add, update and delete vehicles. This app will allow users to have a seamless experience.

3.4.4.2 Operating Environment

This application is a web based application so it will operate on browsers like Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox etc.

3.4.4.3 Design and Implementation Constraints

The app must run on the browser. The information of all users, vehicles, feedbacks must be stored in a database and should be accessible by the website. MySQL will be used as database. The vehicle rental system will be running 24 hrs a day. Users can access the website from any computers that has internet browsing capabilities and internet connection. Users must login with valid credentials to use the application.

3.4.4.4 Assumptions and Dependencies

The project needs database connection, npm , JDK kit to develop the application. Customer will have to register themselves then login with valid credentials to get access to other features of the application. For Admin it can directly login with valid username and password.

3.4.5 Features

The following list provides a brief outline and description of the main features and functionalities of the system.

3.4.5.1 Core Features

3.4.5.1.1 Admin

- Must be able to login into system □ Must be able to add new vehicle.
- Must be able to update and delete vehicle.

3.4.5.1.2 Clients/Users

- Must be able to login into application. □ Must be able to register if is a new user □ Must be able to view all vehicles.
- Must be able to search vehicles.
- Must be able to book/rent vehicles.
- Must be able to change password or edit profile.

3.4.6 Functional Requirements

3.4.6.1 Admin should be able to login into the system

Priority: High (Must Have)

- Stimulus/Response

Step 1: Admin enters username and password

Step 2: System checks if username exists or not in the database.

Step 3: Admin is redirected to Dashboard if credentials are correct else error message is displayed.

3.4.6.2 Admin should be able to can add store/vendor

Priority: High (Must Have)

- Stimulus/Response

Step 1: Admin enters vehicle details like name, price, description, vehicle number, image etc.

Step 2: System add vehicle details into the database as new vehicle.

Step 3: Admin is redirected to Vehicle Action page.

3.4.6.3 Admin should be able to delete and edit vehicle

Priority: High (Must Have)

- Stimulus/Response

Step 1: Admin presses on the delete/ edit icon of the particular vehicle.

Step 2: System deletes/ edit the particular vehicle.

Step 3: Admin is redirected to Vehicle Action page.

3.4.6.4 Clients should be able to register in the app

Priority: High (Must Have)

- Stimulus/Response

Step 1: Client input their details like phone number, password, and email.

Step 2: System adds the user information into user database.

Step 3: Client is redirected towards login page for login.

3.4.6.5 Client should be able to login into the system

Priority: High (Must Have)

- Stimulus/Response

Step 1: Client enters username and password

Step 2: System checks if username exists or not in the database.

Step 3: Client is redirected to Dashboard if credentials are correct else error message is displayed.

3.4.6.6 Client should be able to book/rent vehicles

Priority: High (Must Have)

- Stimulus/Response

Step 1: Client selects vehicles to book/rent them.

Step 2: System sends requests to admin panel and reserves the vehicle for user.

Step 3: Client is redirected to available vehicles.

3.4.6.7 Client should be able to search vehicles Priority:

High (Must Have)

- Stimulus/Response

Step 1: Client selects search method like search by name or vehicle type

Step 2: System filters according to search.

Step 3: Client is redirected to available vehicles after filtering vehicle details.

3.4.6.8 Client should be able to send feedback

Priority: High (Must Have)

- Stimulus/Response

Step 1: Client enter comments and sends it.

Step 2: System stores the comment in the database.

Step 3: Displays success message.

3.4.6.10 Admin should be able to approve/reject reserved vehicle by client Priority:

High (Must Have)

- Stimulus/Response

Step 1: Admin clicks button approve/reject.

Step 2: System sends message to client according to button clicked.

Step 3: System deletes the request after approving or rejecting the requests.

Step 3: Admin is redirected to vehicle reserved by client page.

3.4.6.11 Admin should be able to approve/reject reserved vehicle by client Priority:

High (Must Have)

- Stimulus/Response

Step 1: Admin clicks button approve/reject.

Step 2: System sends message to client according to button clicked.

Step 3: System deletes the request after approving or rejecting the requests.

Step 3: Admin is redirected to vehicle reserved by client page.

3.4.6.12 Admin should be able to view vehicle on rent report Priority:

High (Must Have)

- Stimulus/Response

Step 1: Admin vehicle rented by client

Step 2: System displays report with vehicle and customer details.

Step 3: Admin is redirected to vehicle rented by client.

3.4.6.13 Admin should be able to view feedback

Priority: High (Must Have)

- Stimulus/Response

Step 1: Admin select feedback

Step 2: System displays report with feedback and username.

Step 3: Admin is redirected to feedback page.

3.4.6.14 Users should be able to logout Priority:

High (Must Have)

- Stimulus/Response

Step 1: User click button logout Step

2: System clears the session

Step 3: Admin is redirected to login page.

3.4.7 External Interface Requirements

3.4.7.1 User Interface

The user interface of the project will be simple and easy to use and understand. Simple designs and terminology will be used in the development of the application with standard interface as per the requirement of the client. No training or user guides will be required to operate this application for new users. Consistency in the application will be maintained throughout the system through the use of same of buttons and components.

3.4.7.2 Hardware Interfaces

No any extra hardware interfaces are required for the project. Only basic requirements of the computer system are needed for the project. The application will use hardware and database connection resource provided by Islington College.

3.4.7.3 Software Interfaces

The project will use software resources required for the application development and database connection which includes IntelliJ Idea software platform(libraries, plugins), MySQL database, npm, JRE etc.

3.4.7.4 Communications Interfaces

The application will use Gmail mail client as a communication interface between the admin/developer and user for approval and rejection of vehicle requests.

3.4.8 Nonfunctional Requirements

3.4.8.1 Performance Requirements

The application should not crash while opening. The application will display the contents of the page within 5-10 seconds. All the features of the application will be functional to the users.

3.4.8.2 Safety Requirements

Vehicle Rental System application cannot harm, affect, damage ant property of user, client or any organization. User's details and privacy of data will be maintained and cannot be accessible to others by any means or process. Hence, the system is safe to use and all the safety requirements are identified.

[Click to view initial SRS document](#)

3.5 Implementation of Methodology

3.5.1 Iteration 1:

3.5.1.1 Requirement Gathering:

This is the first phase of the methodology where requirements are gathered to understand the concept of rental system. The requirements for the vehicle rental system was gathered through research on different websites as well as also using them to understand the general work flow of the application. A survey was conducted to understand the view of people related to vehicle rental system.

Requirements Gathered:

- Admin/Client: User Login and authentication
- Client: User registration
- Admin: Add, Update and delete vehicles
- Client : Book, Rent, View Available features, Send Feedback

3.5.1.2 Analysis and Design:

The requirements were analyzed thoroughly. Then the tools to be used for the development was finalized. In this phase wireframes was designed for overall system, use case diagram, sequence diagram, collaboration diagram, activity diagram and entity relationship diagram was also designed for overall system.

Class Diagram:

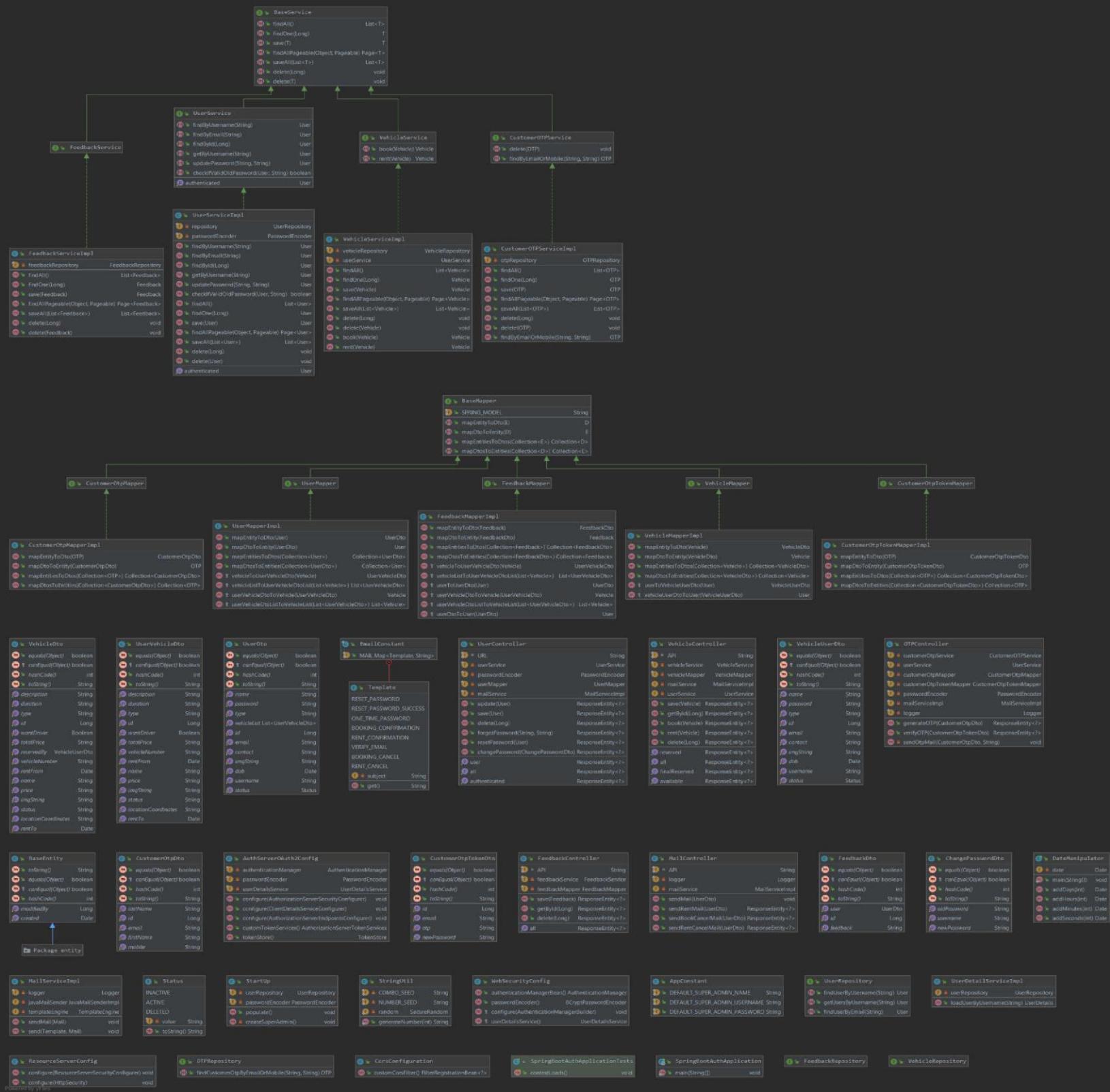


Figure 7: Class Diagram

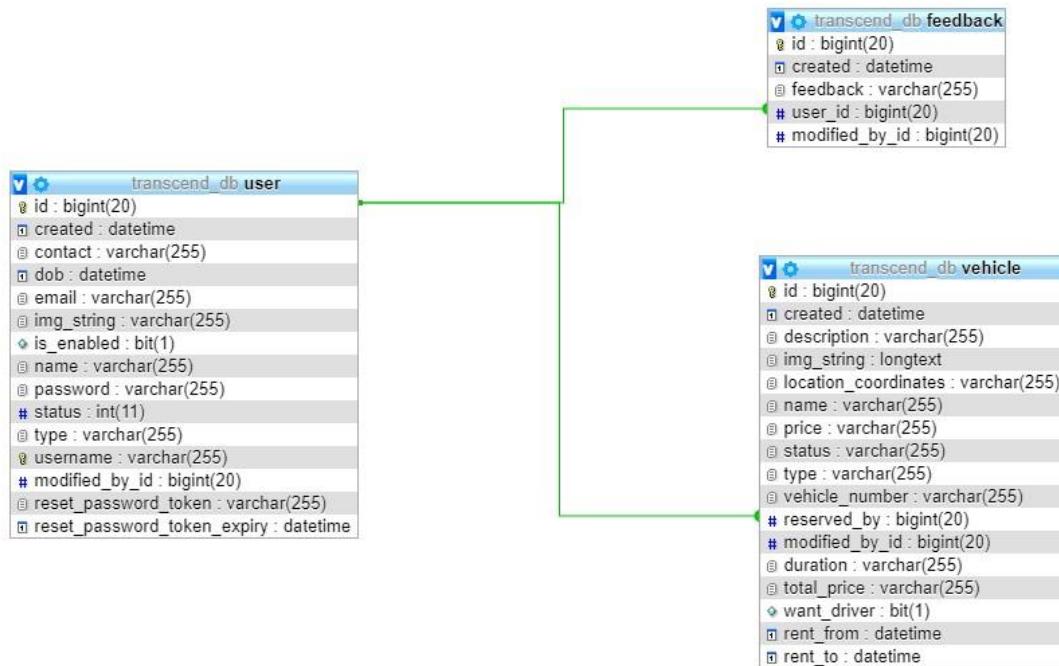
ERD:

Figure 8: ER Diagram

Wireframe:

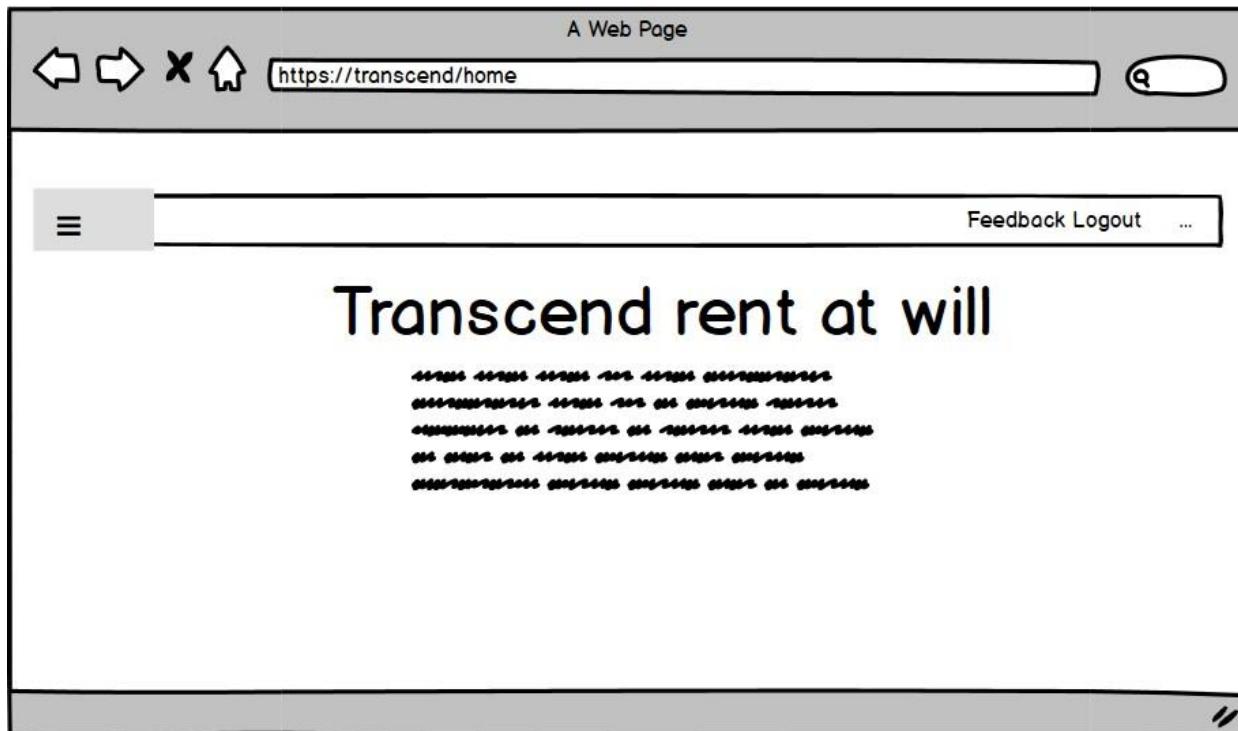


Figure 9: Wireframe of application

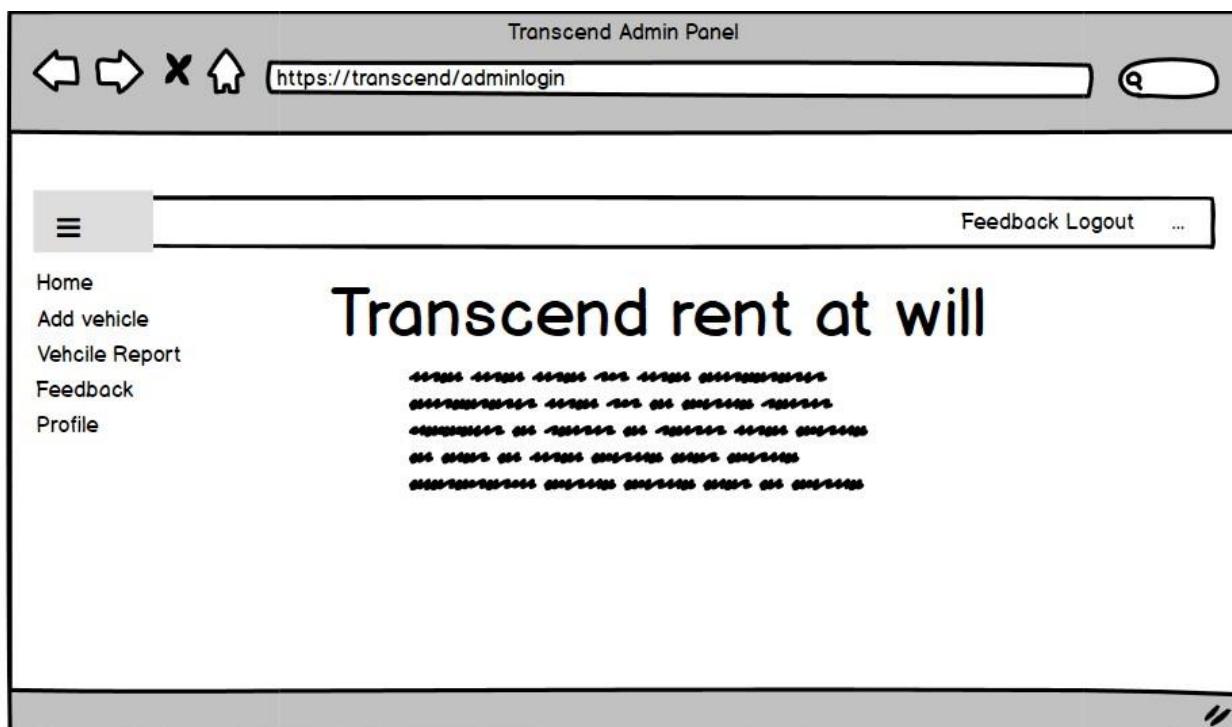


Figure 10: Wireframe for admin panel

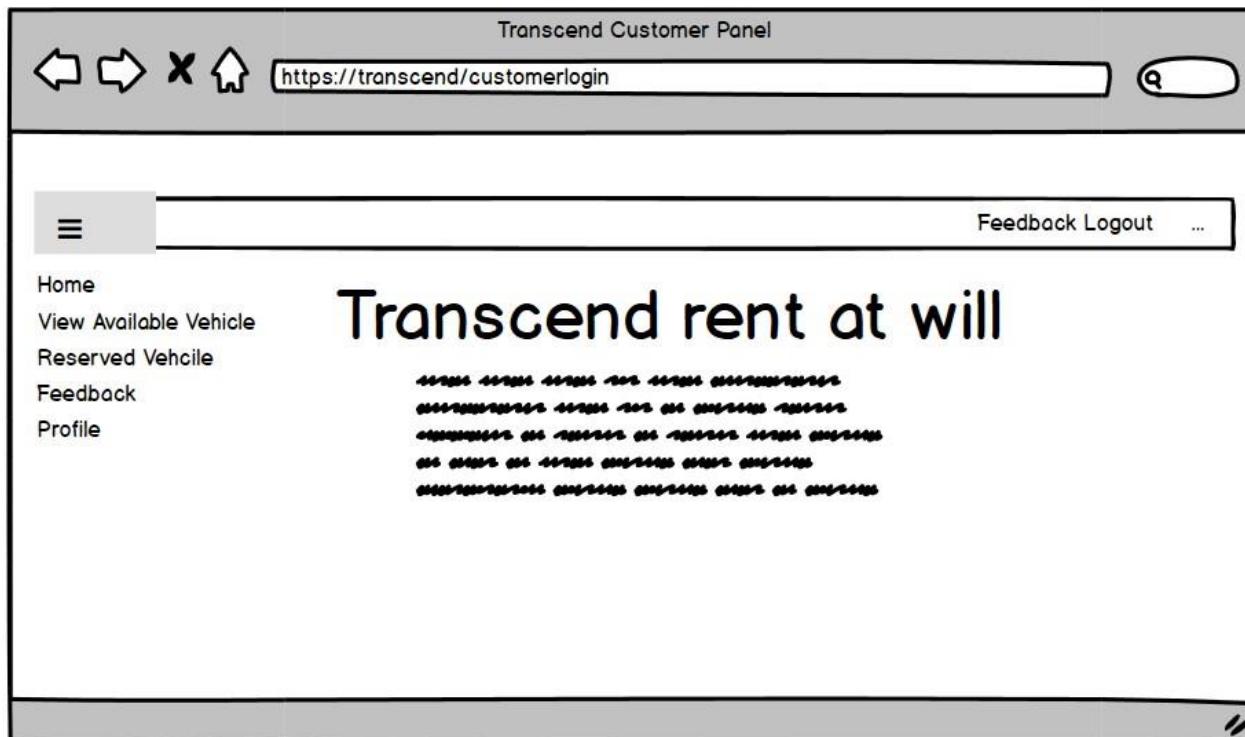


Figure 11: Wireframe for customer panel

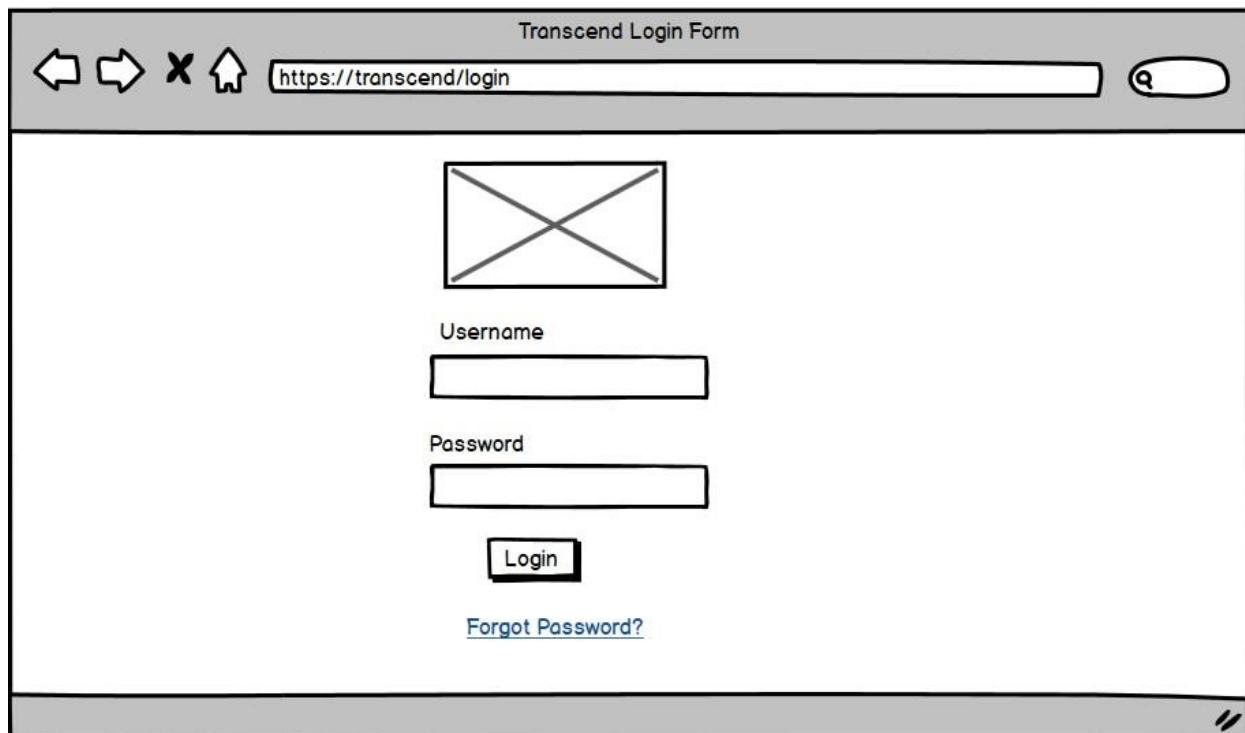
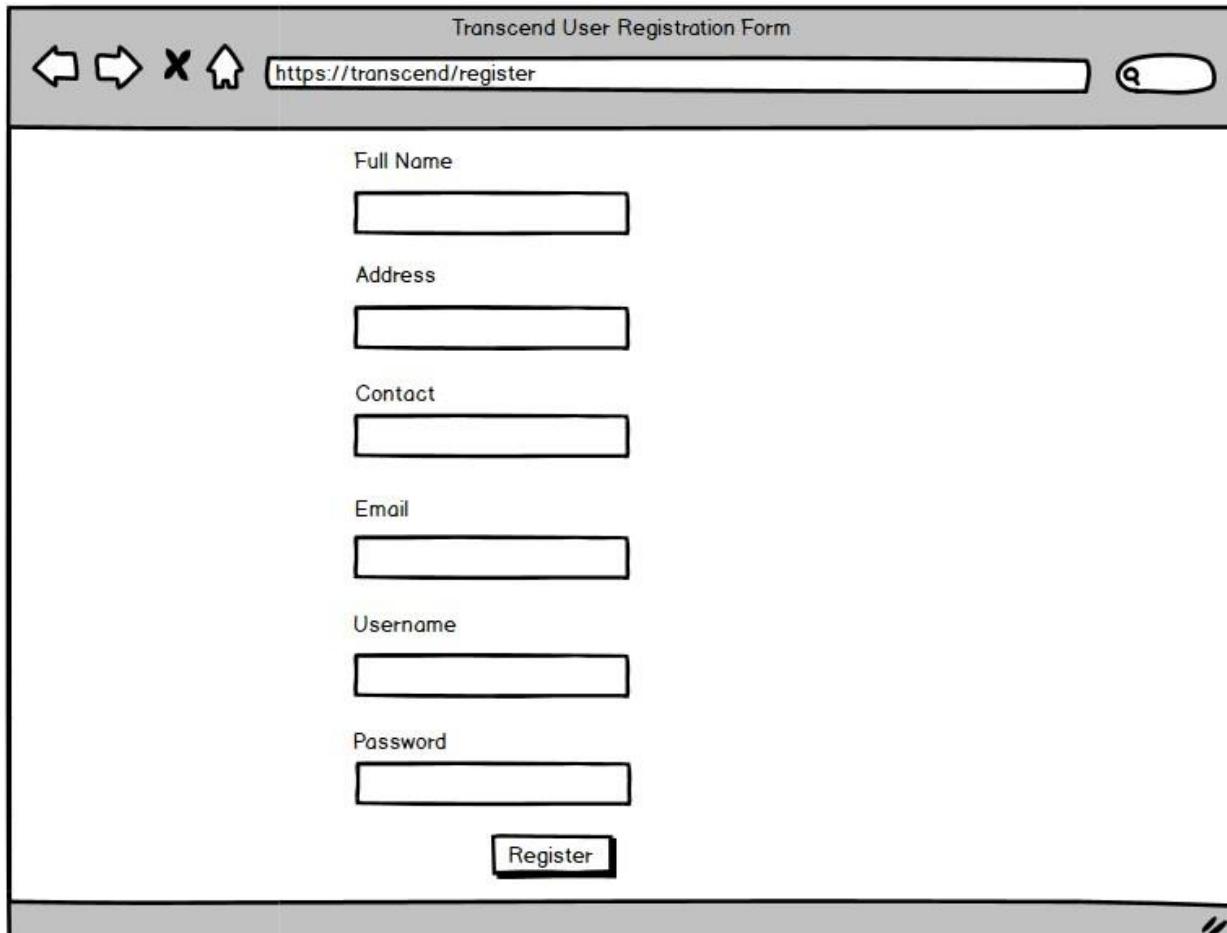


Figure 12: Wireframe of Login Form



The wireframe shows a registration form titled "Transcend User Registration Form" with the URL "https://transcend/register". The form includes fields for Full Name, Address, Contact, Email, Username, and Password, each with a corresponding input box. A "Register" button is located at the bottom.

Figure 13: Wireframe of registration form

Use Case Diagram:

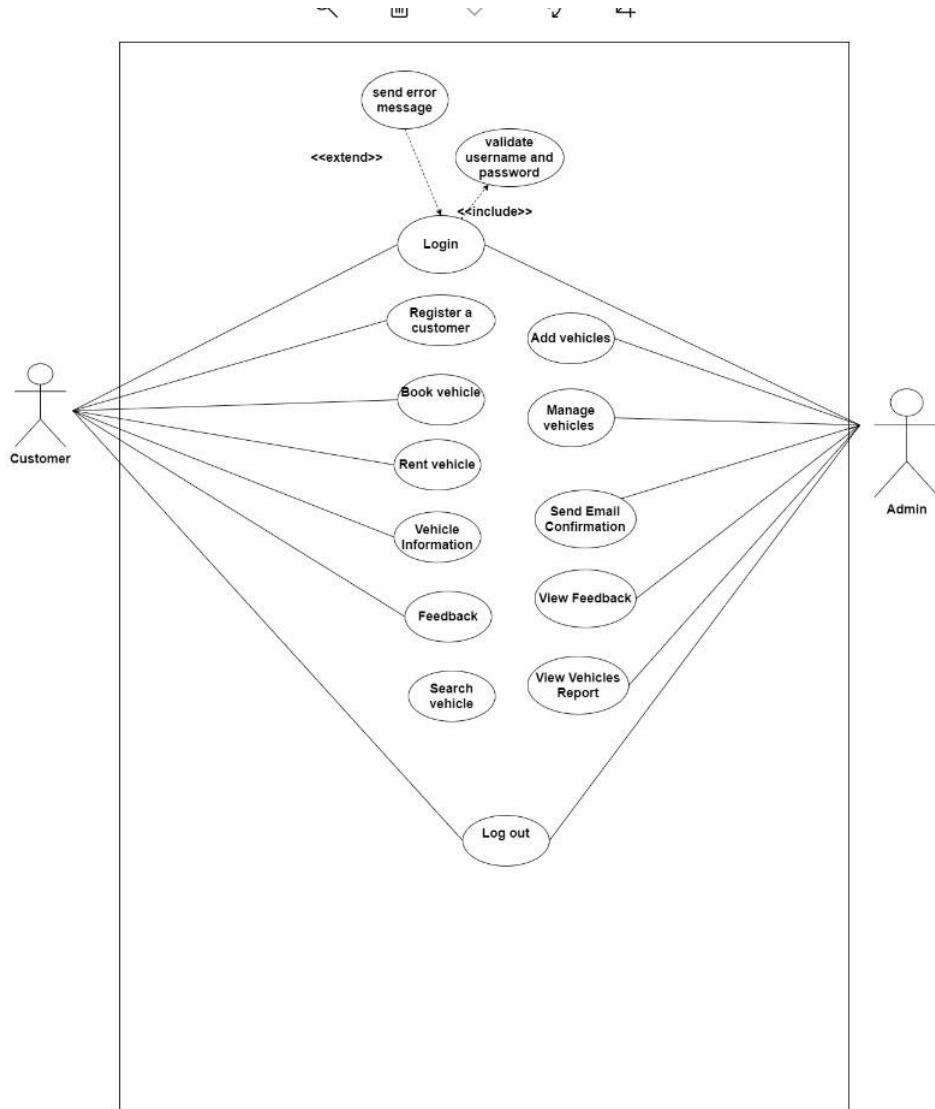


Figure 14: Final Use case of whole system

3.5.1.3 Implementation:

After completion of requirement gathering and designing phase the development of system was started on spring boot and angular. Frontend and backend for the system was setup in this phase. In memory authentication of spring security was implemented in backend for authentication of users along with creating MySQL database.

Code implementation:

Environment setup

```

## hibernate Properties
#The SQL dialect
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
## Hibernate Logging
logging.level.org.hibernate.SQL=debug

## MySQL Datasource
spring.datasource.url=jdbc:mysql://localhost:3306/transcend_db?useUnicode=true&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=

server.port=8081

```

Figure 15: Setup database connection

Table	Action	Rows	Type	Collation	Size	Overhead
customer_otp		0	InnoDB	latin1_swedish_ci	16 Kib	-
feedback		4	InnoDB	latin1_swedish_ci	32 Kib	-
hibernate_sequence		4	InnoDB	latin1_swedish_ci	16 Kib	-
mail		0	InnoDB	latin1_swedish_ci	16 Kib	-
user		9	InnoDB	latin1_swedish_ci	32 Kib	-
vehicle		5	InnoDB	latin1_swedish_ci	1.5 Mib	-
6 tables	Sum	22	InnoDB	latin1_swedish_ci	1.6 Mib	0 B

Figure 16: Database created

The above code consists of information that is required to connect the project with MySQL database. This code is present in the application properties the project once setup the project automatically connects the project to database

3.5.1.4 Testing:

[Click here to view testing for iteration 1](#)

3.5.1.5 Evaluation:

All the requirements mentioned above were completed successfully in this iteration. After the project was setup successfully the application was tested and the results were good as the program was running successfully.

3.5.2 Iteration 2:

3.5.2.1 Requirement Gathering:

Since primary requirements were already gathered in Iteration 1. This iteration mainly focuses on implementing the functionality of user type admin. Therefore no any other requirements were gathered in this iteration.

3.5.2.2 Analysis and Design:

The requirement gathered for the user type admin was analyzed and the features to be implemented was finalized. The wireframes for the admin page was designed. Then use case diagram along with collaboration, sequence, activity diagram was designed mainly focusing on the actor admin.

Wireframe

The wireframe depicts the 'Transcend Admin Panel' interface. At the top, there's a header bar with navigation icons (back, forward, search, etc.) and a URL bar showing <https://transcend/admin/profile>. Below the header is a main content area with a sidebar on the left containing links: Home, View Available Vehicle, Reserved Vehicle, Feedback, and Profile. The main content area features a large title 'Profile'. It includes several input fields for profile information: Name, Address, Contact, Username, and a 'Change' section with a 'New Password' field. A prominent 'Update' button is centered at the bottom of the form.

Figure 17: Wire frame of admin profile

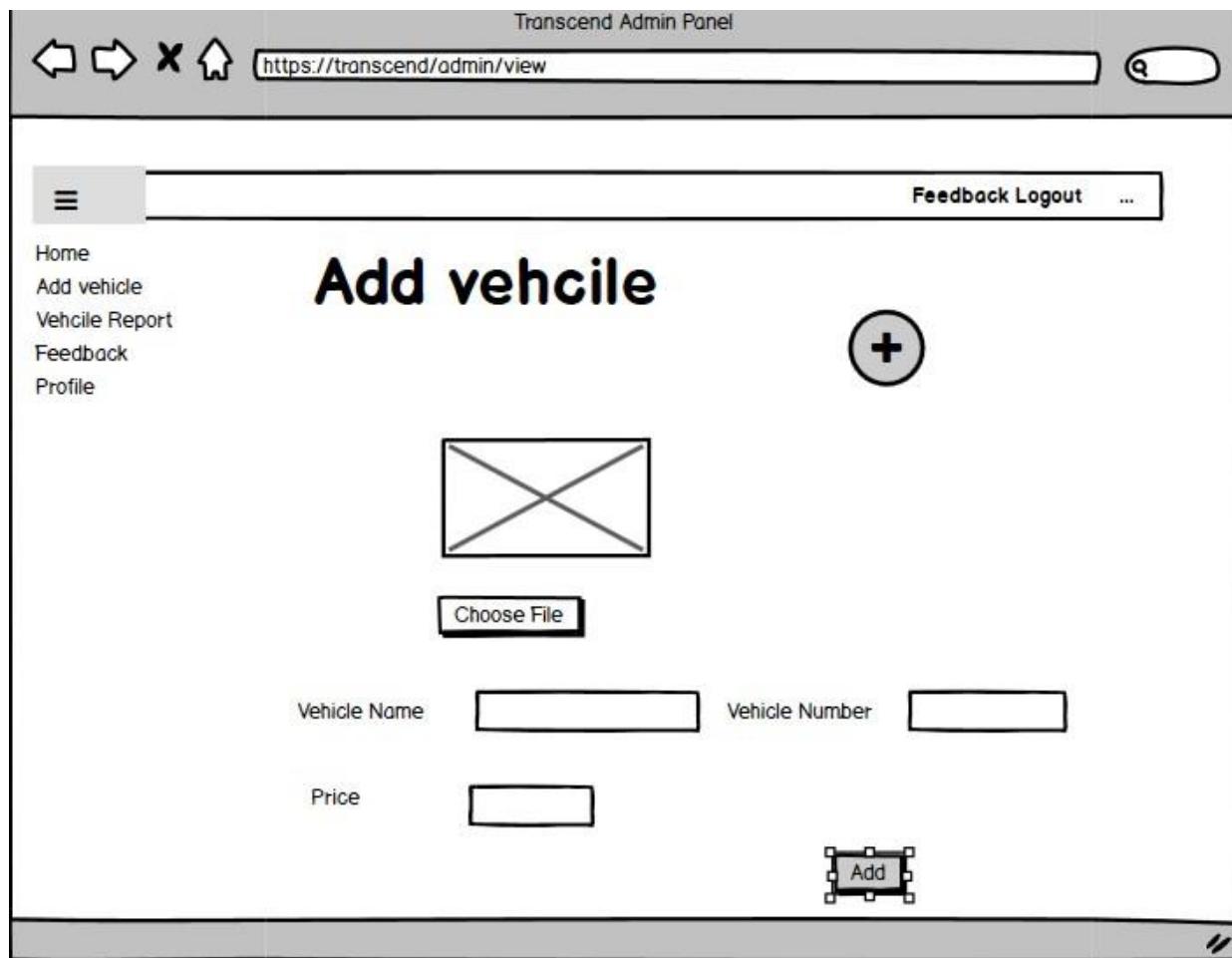


Figure 18: Wireframe of adding vehicles

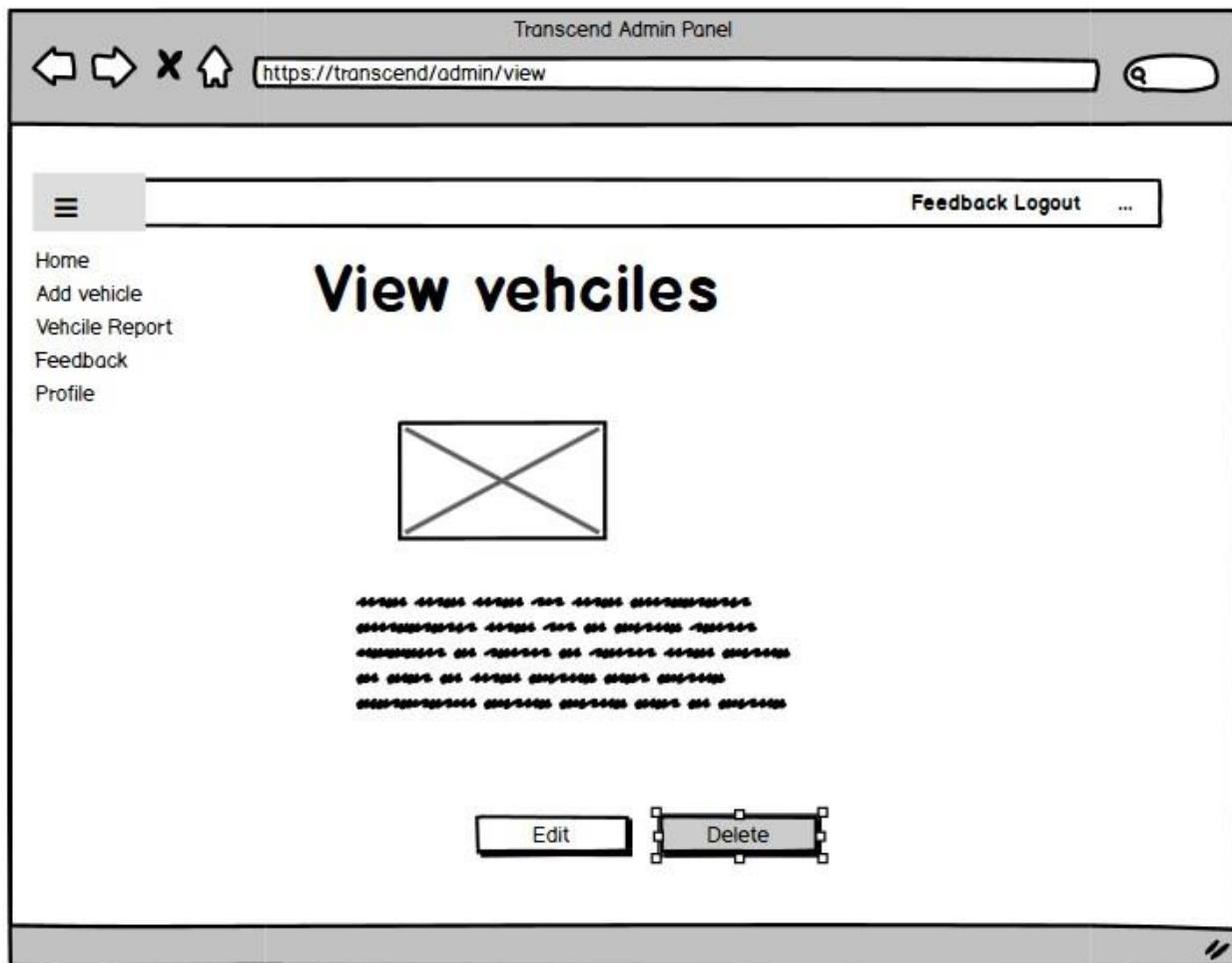


Figure 19: Wireframe of List of Added Vehicle



The wireframe depicts a web browser window with a title bar containing standard icons (back, forward, search, etc.). The main content area is titled "Forgot Password". It contains two input fields: one labeled "Verification code" and another labeled "New password", both represented by rectangular boxes. Below these fields is a single "Submit" button, also enclosed in a rectangular box.

Figure 20: Wireframe for Admin forgot password

Use case diagram

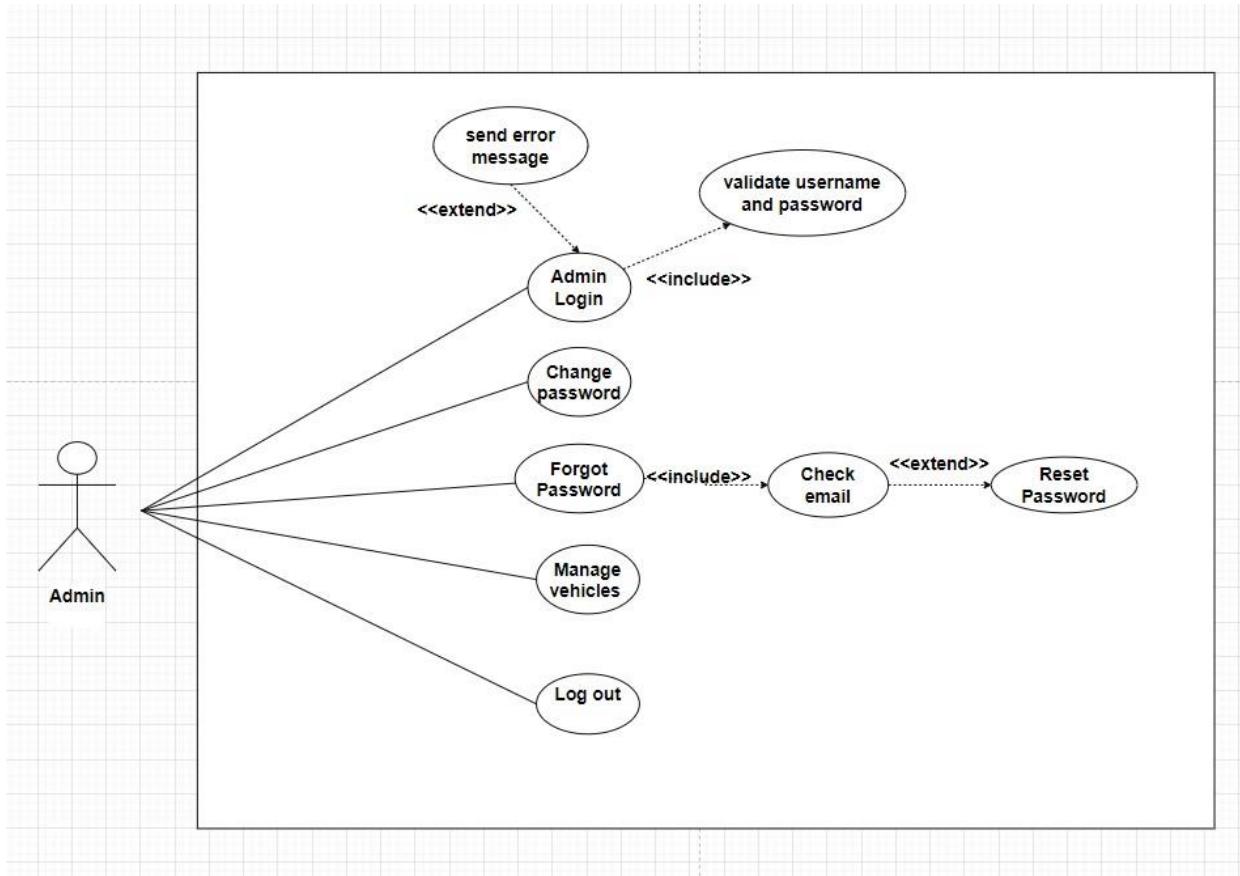


Figure 21: Use case diagram for admin

[High Level Use case diagram](#)

[Expanded Use case diagram](#)

Activity diagram

Admin Login

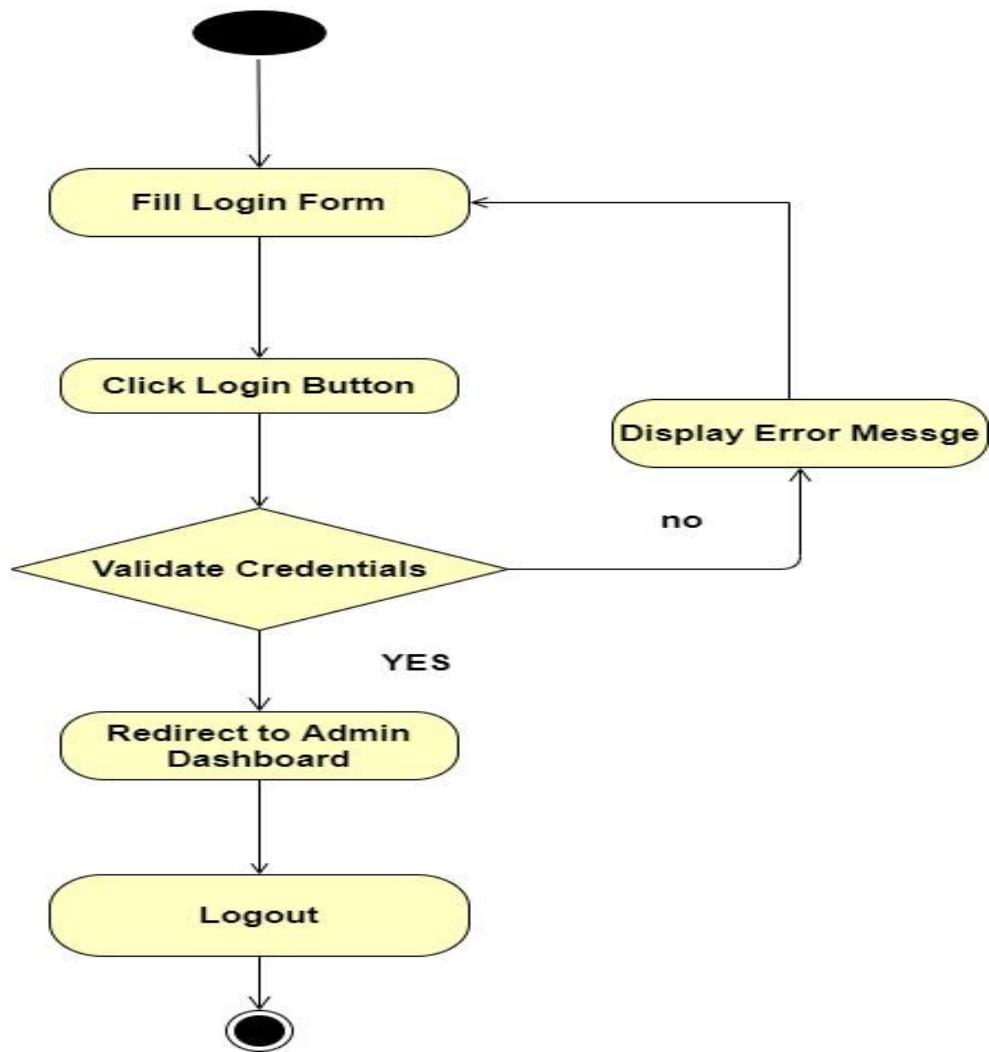


Figure 22: Activity Diagram for Admin login

Forgot Password

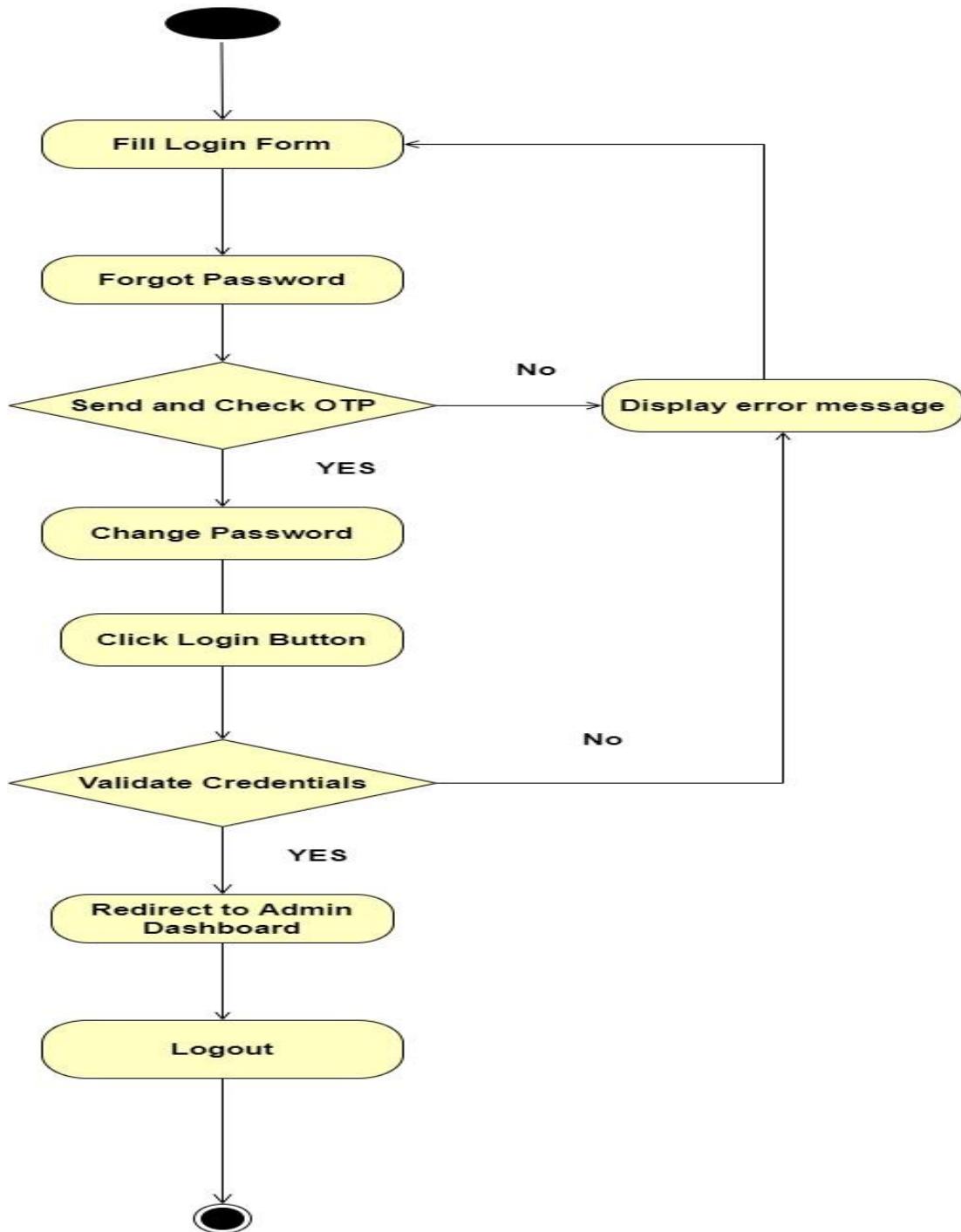


Figure 23: Activity Diagram for Admin Forgot password

CRUD operation

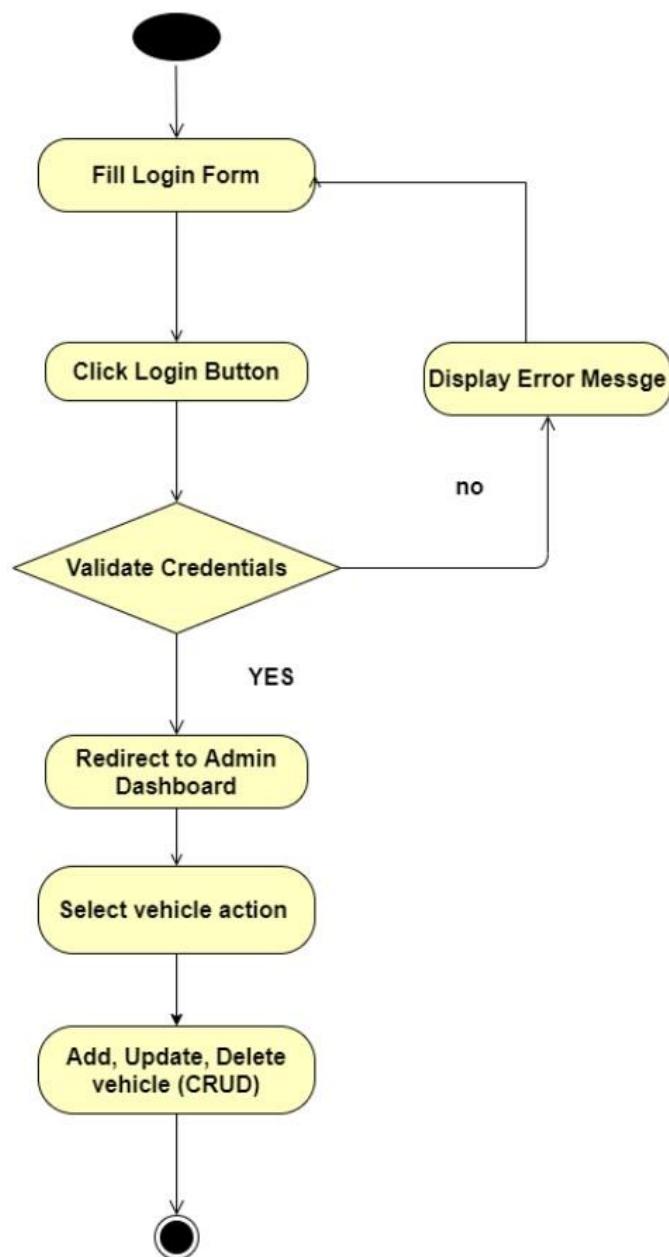


Figure 24: Activity Diagram for Admin CRUD of vehicles

Change Password

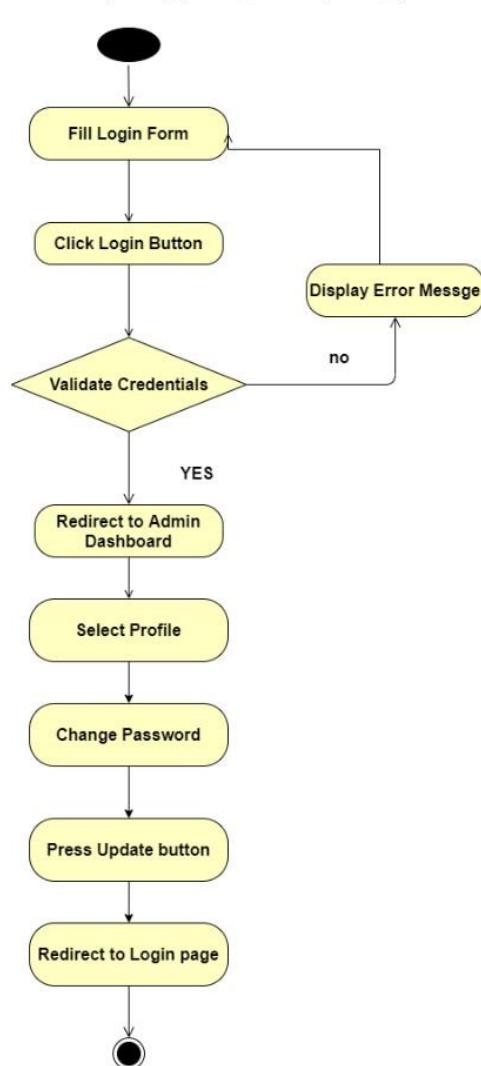


Figure 25: Activity Diagram for Admin change password

Sequence diagram

Admin Login

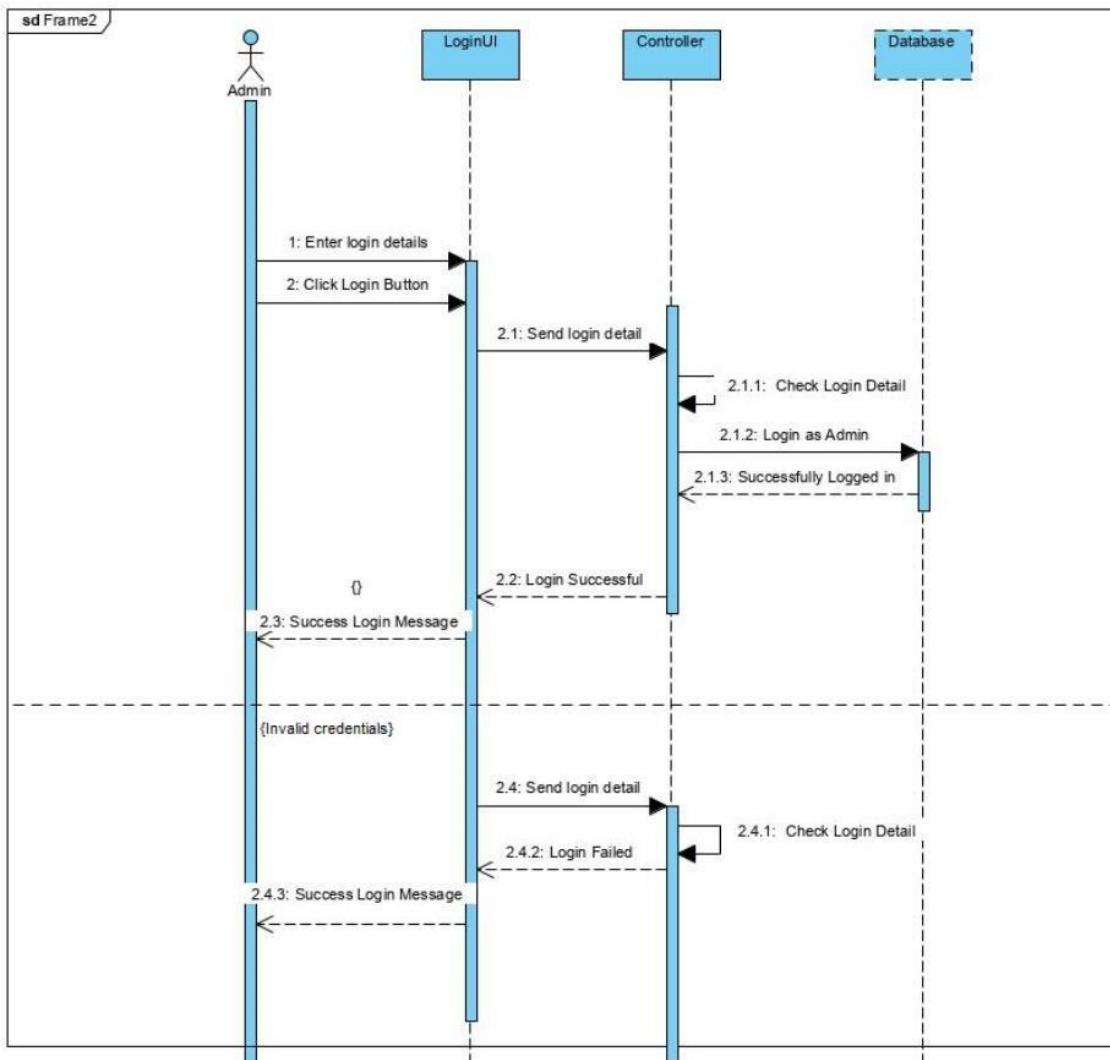


Figure 26: Sequence Diagram for Admin Login

Admin CRUD of vehicle

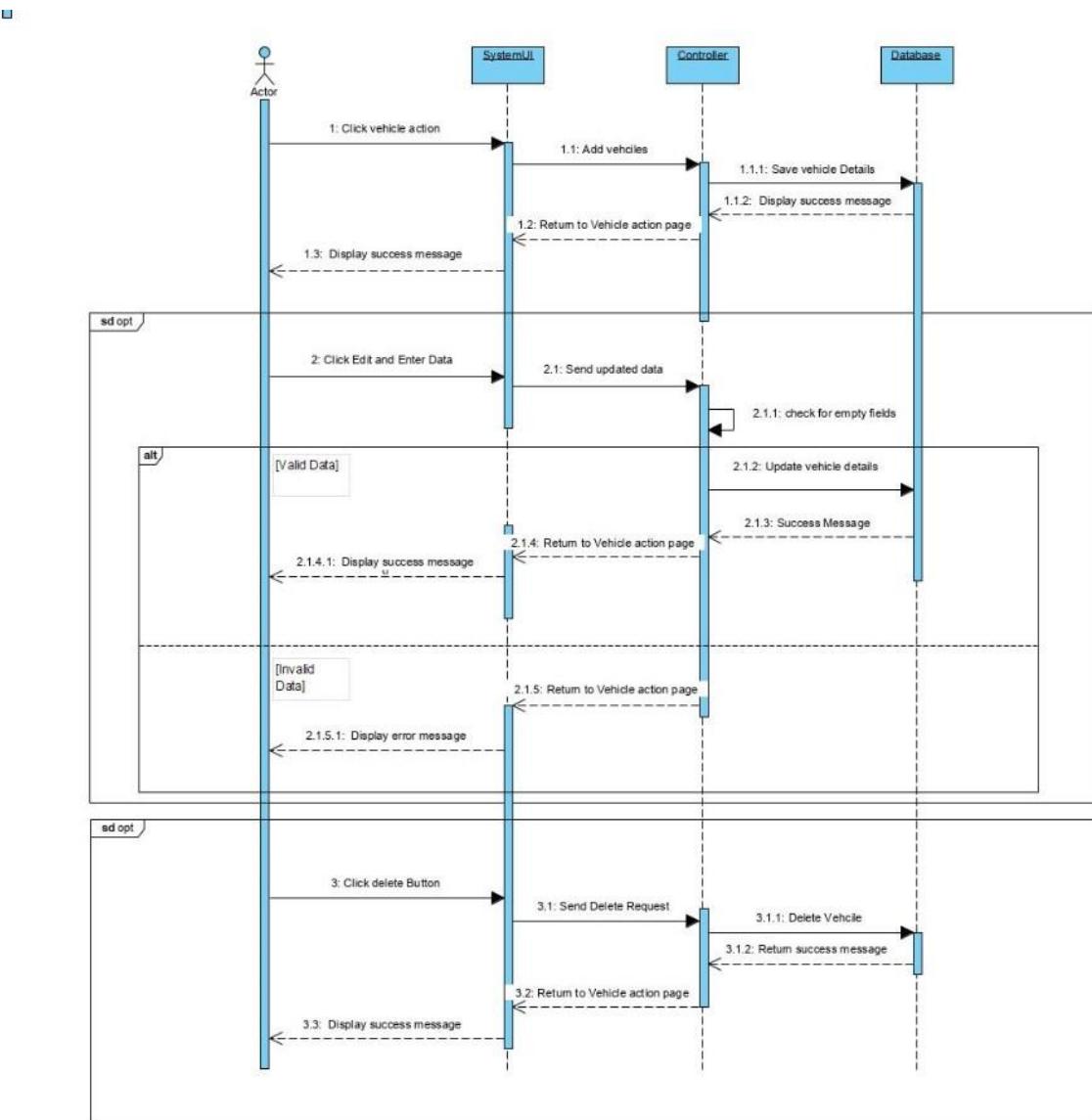


Figure 27: Sequence diagram for admin crud of vehicle

Change password

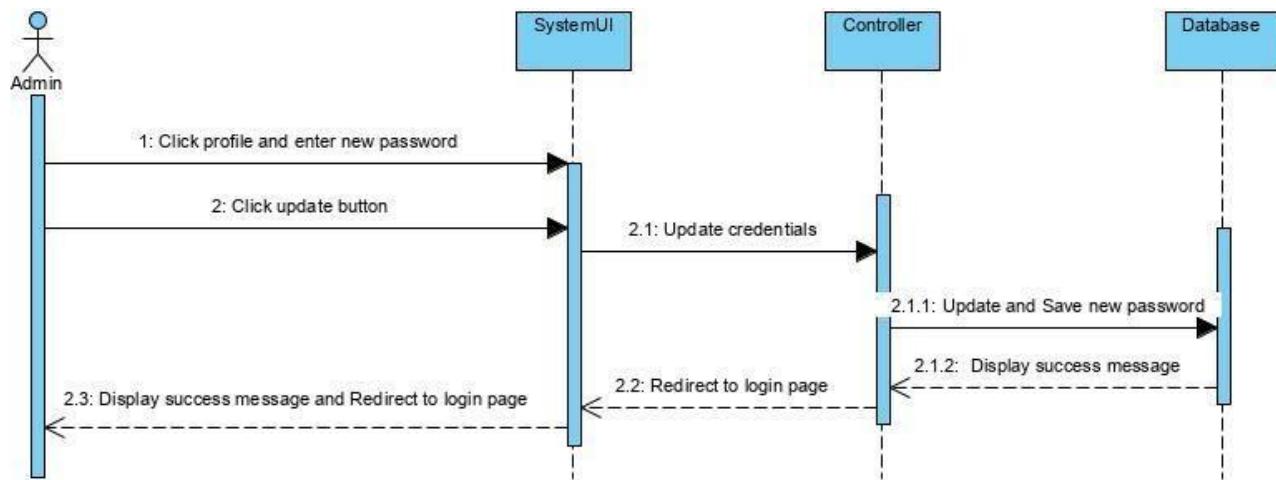


Figure 28: Sequence diagram for admin change password

Forgot Password

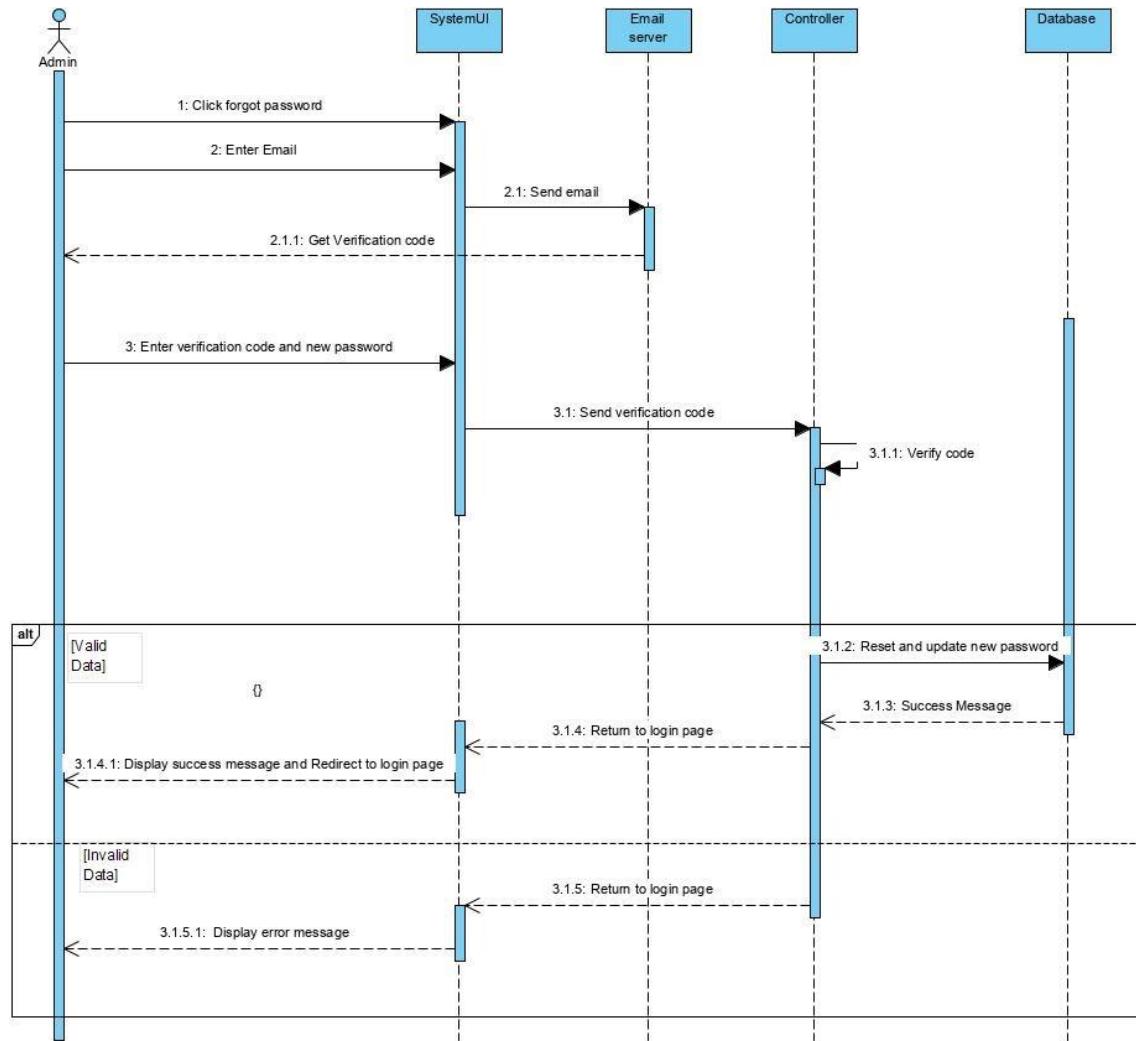


Figure 29: Sequence diagram for admin forgot password

Collaboration diagram

Admin Login

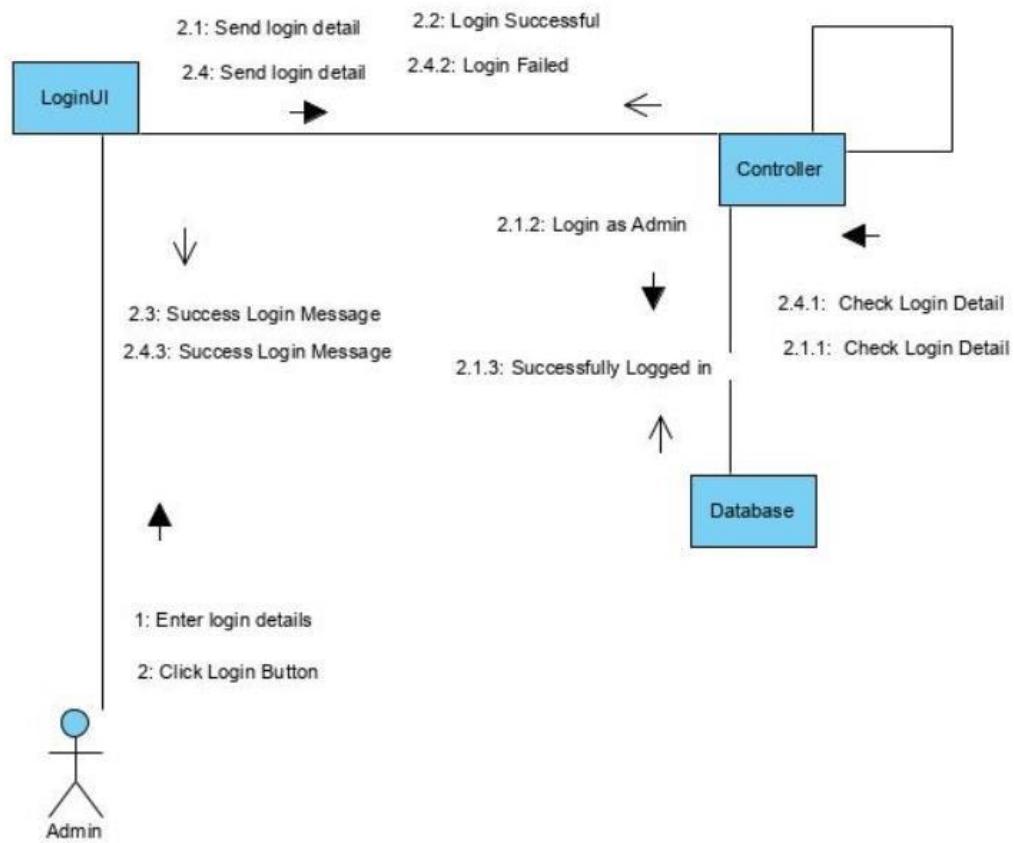


Figure 30: Collaboration Diagram for Admin Login

Admin CRUD of vehicle

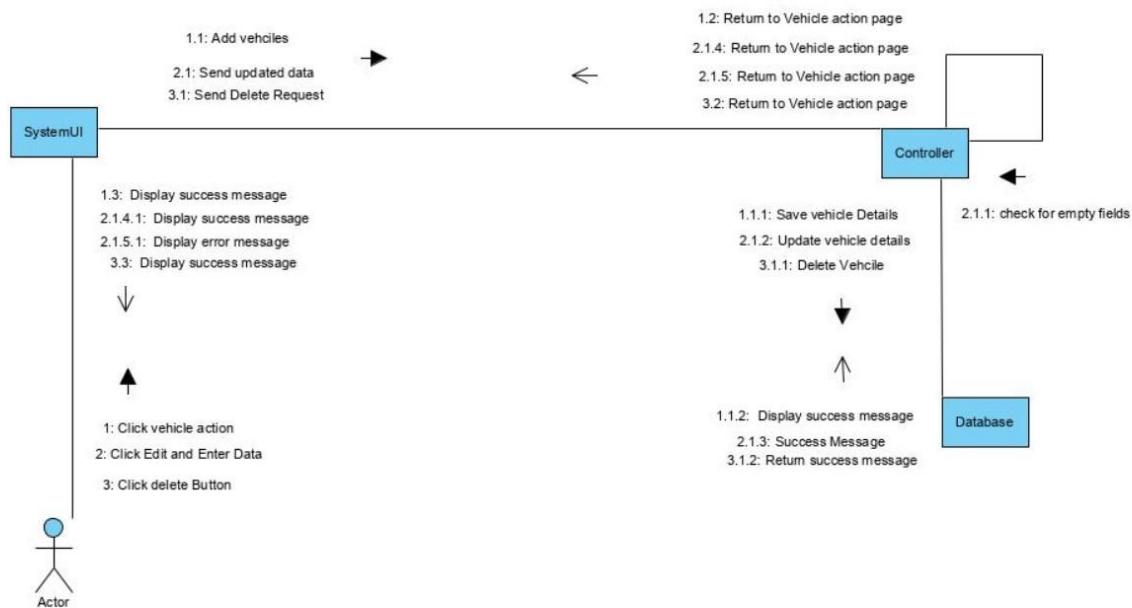


Figure 31: Collaboration Diagram for Admin CRUD of vehicle

Change password

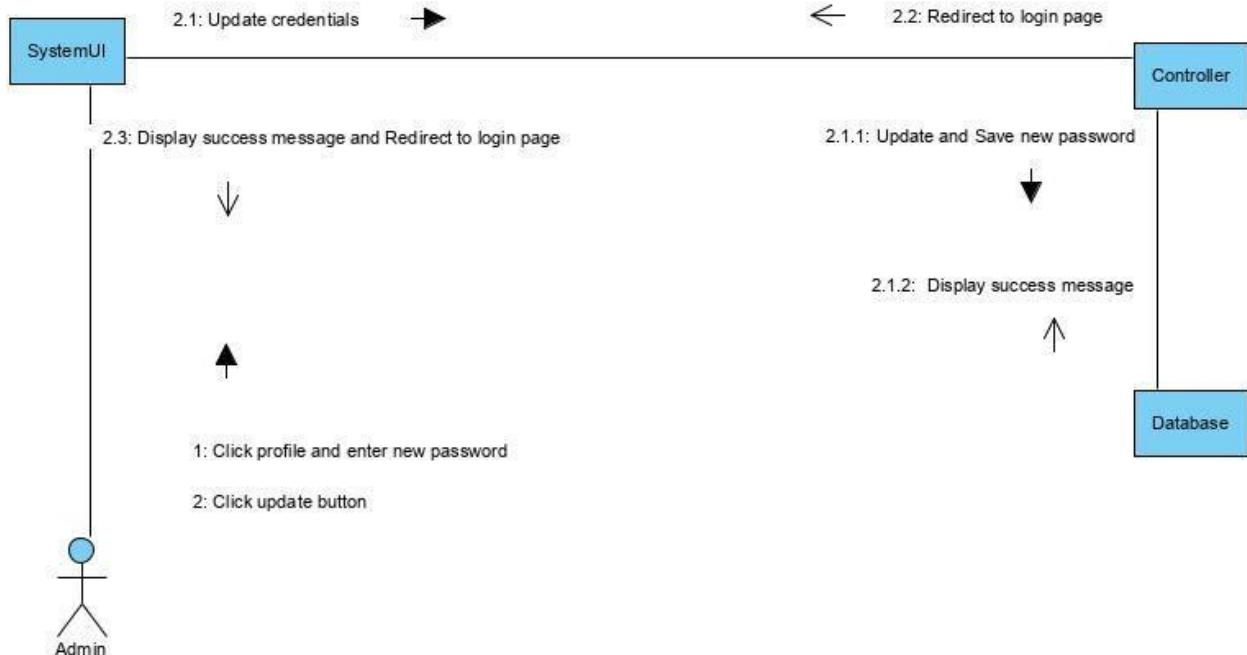


Figure 32: Collaboration diagram for admin change password

Forgot Password

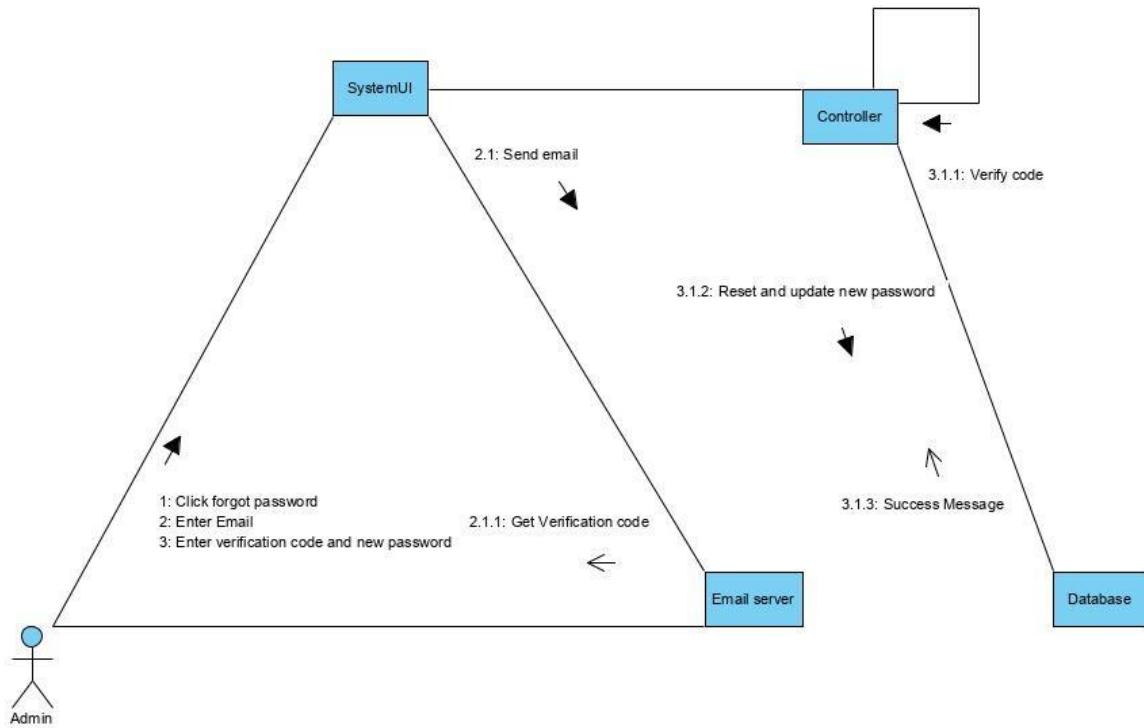


Figure 33: Collaboration diagram for admin forgot password

3.5.2.3 Implementation:

CRUD operation for admin panel was implemented in this phase.

Code implementation

Login

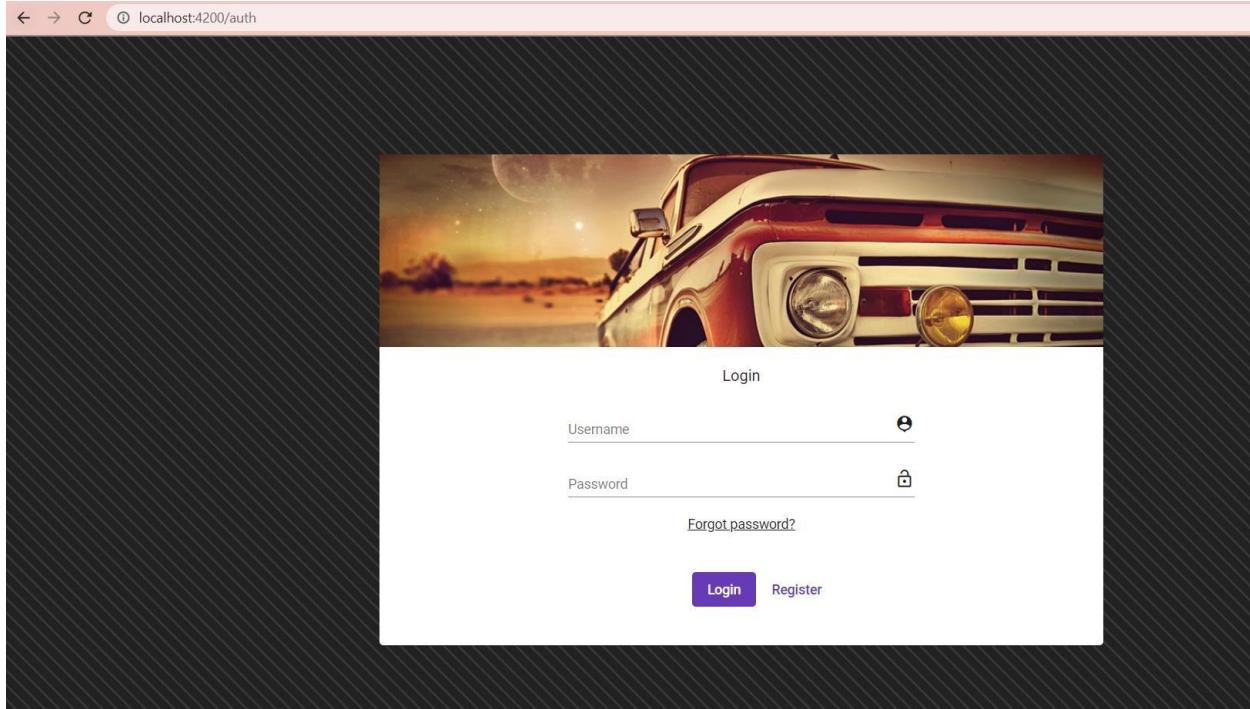


Figure 34: Login UI

```

    onSubmit() {
      if (this.loginForm.invalid) {
        return;
      }
      this.authService.login(this.loginForm.value).subscribe( res => {
        localStorage.clear();
        localStorage.setItem('token', res.access_token);
        this.authService.getActiveUser().subscribe( next: activeUser => {
          localStorage.setItem('user_type', activeUser.type);
          localStorage.setItem('user_id', activeUser.id);
          this.checkActiveUserService.changeCurrentUserId(activeUser.id);
          this.checkActiveUserService.changeCurrentUserToken(res.access_token);
          this.checkActiveUserService.changeCurrentUserType(activeUser.type);
          this.router.navigate( commands: ['/detail']).finally( () => {
            this.snackBar.open( message: 'Login successful!', action: 'Close', config: {duration: 2000, panelClass: ['success-snack-bar']});
          });
        });
      }, error: error => {
        this.snackBar.open( message: 'Login Failed!!', action: 'Close', config: {duration: 2000, panelClass: ['danger-snack-bar']});
        console.log(error);
      });
    }
  }
}

```

Figure 35: Login method



```
public login(user: User): Observable<any> {
  const headers = {
    Authorization: 'Basic c2FmaXNocmVzdGhhOnNwcmLuZw==',
    'Content-type': 'application/x-www-form-urlencoded'
  };
  const body = new HttpParams()
    .set('username', user.username)
    .set('password', user.password)
    .set('grant_type', 'password');
  return this.http.post( url: 'http://localhost:8081/oauth/token', body, options: {headers});
}
```

Figure 36: Authenticate user api

In the above picture the method **on submit** is used to authenticate valid user and generate access token for user to login into system. User details are stored in user password with user type and password are stored in encrypted form.

CRUD operation

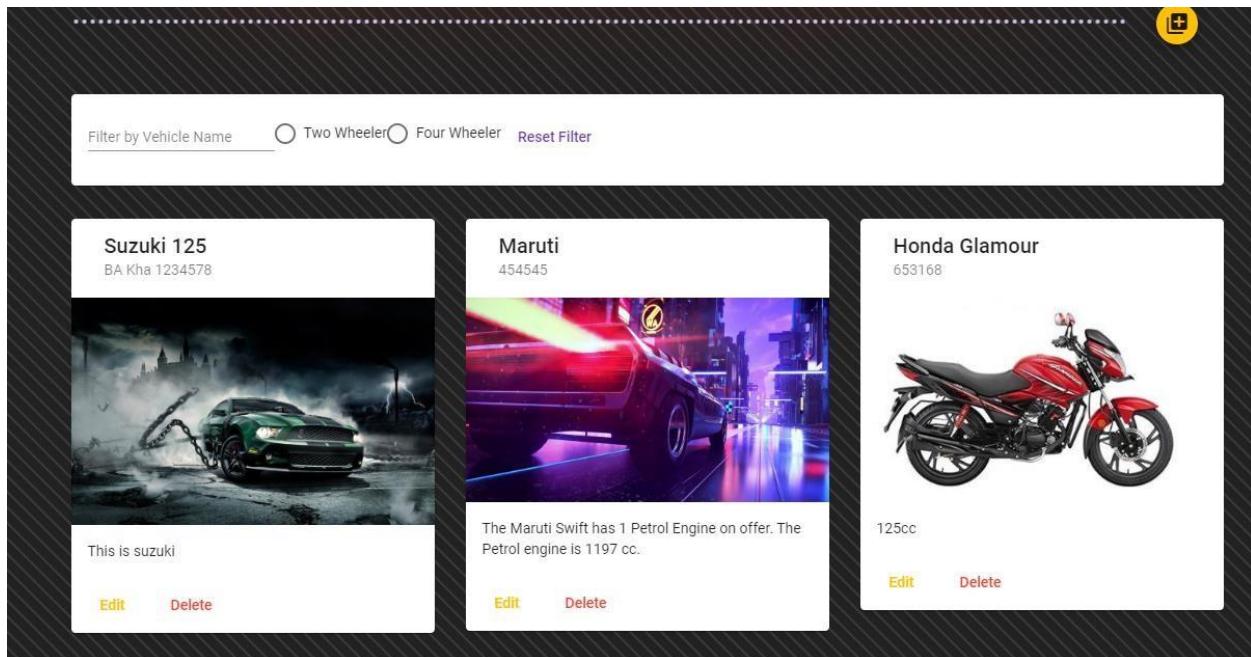


Figure 37: Vehicle CRUD UI

```

onSubmit() {
  if (this.forEdit) {
    this.data.name = this.VehicleActionForm.get('name').value;
    this.data.description = this.VehicleActionForm.get('description').value;
    this.data.imgString = this.VehicleActionForm.get('imgString').value;
    this.data.vehicleNumber = this.VehicleActionForm.get('vehicleNumber').value;
    this.data.type = this.VehicleActionForm.get('type').value;
    this.data.status = this.VehicleActionForm.get('status').value;
    this.data.price = this.VehicleActionForm.get('price').value;

    this.vehicleService.save(this.data).subscribe( next: res => {
      console.log(res);
      this.activeModal.close();
      this.snackBar.open( message: 'Successfully updated vehicle!', action: 'Close', config: {duration: 2000, panelClass: ['success-snack-bar']} );
    }, error: error => {
      this.snackBar.open(error.error, action: 'Close', config: {duration: 2000, panelClass: ['danger-snack-bar']} );
      console.log(error);
    });
  } else {
    const vehicle = this.VehicleActionForm.value as Vehicle;
    this.vehicleService.save(vehicle).subscribe( next: response => {
      console.log(response);
      this.activeModal.close();
      this.snackBar.open( message: 'Successfully added vehicle!', action: 'Close', config: {duration: 2000, panelClass: ['success-snack-bar']} );
    }, error: error => {
      this.snackBar.open(error.error, action: 'Close', config: {duration: 2000, panelClass: ['danger-snack-bar']} );
      console.log(error);
    });
  }
}
}

```

Figure 38: Vehicle add and update method

```
    deleteVehicle(vehicle: Vehicle) {
      if (confirm('Are you sure to delete this vehicle?')) {
        this.vehicleService.delete(vehicle.id).subscribe( res => {
          this.ngOnInit();
          this.snackBar.open( message: 'Deleted!', action: 'Close', config: {duration: 2000});
        }, error => {
          this.snackBar.open(error.error, action: 'Close', config: {duration: 2000, panelClass: ['danger-snack-bar']});
          console.log(error);
        });
      }
    }
```

Figure 39: Delete vehicle method

In above snippet code the function **onsubmit** and **deleteVehicle** is used to add vehicle, delete vehicle and update vehicle and details of vehicles are stored in the database in table vehicle.

3.5.2.4 Testing:

[Click here to view testing for iteration 2](#)

3.5.2.5 Evaluation:

All the requirements gathered to be completed in this iteration was completed on time. The features developed for admin page was fully functional and turned out to be satisfactory. However the features for customer or client side was not completed in this iteration so it was difficult to test and evaluate all the features for user admin. Thus, the feature for this user will be evaluated in the next iteration along with the other user client.

3.5.3 Iteration 3:

3.5.3.1 Requirements:

This iteration focuses on implementing the functionality of user type client or vehicle seeker. The primary requirements for customer or client side were already gathered in Iteration 1. However from the survey conducted in iteration1 provides a lot of information as well as survey reflected many new features that people want to see in a modern vehicle rental system. Requirements were gathered through survey, research conducted online as well as from client.

3.5.3.2 Analysis and Design:

The requirement gathered for the user type client or vehicle seeker was analyzed and the features to be implemented was finalized. The wireframes for the customer page was designed. Then use case diagram along with collaboration, sequence, activity diagram was designed mainly focusing on the actor customer or client.

Wireframe

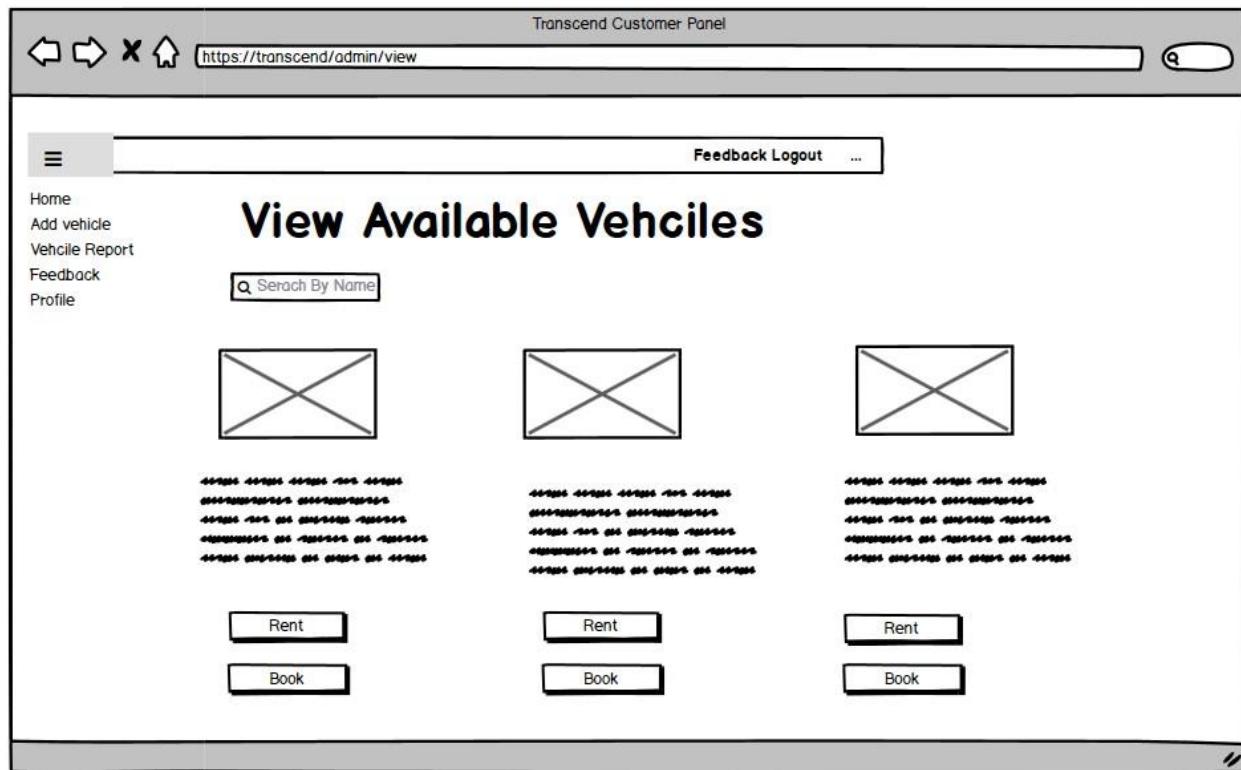


Figure 40: Wireframe of Available Vehicles for customer

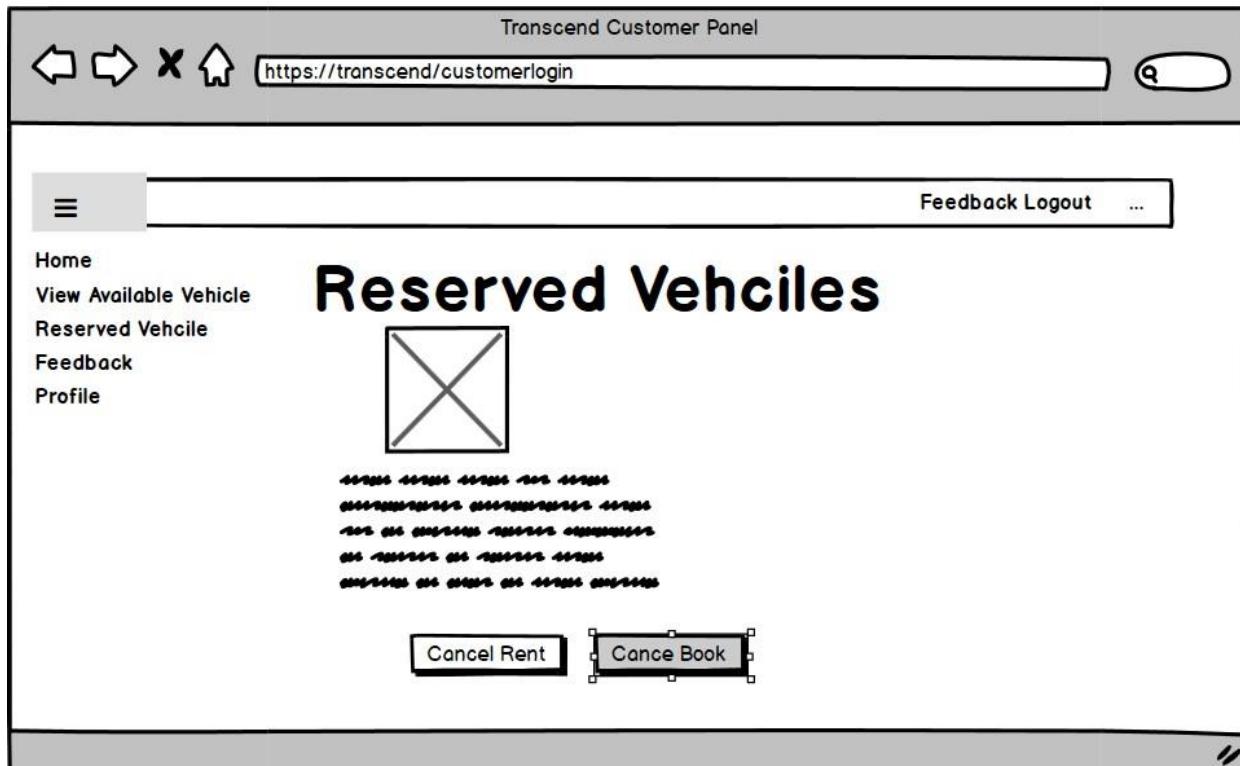


Figure 41: Wireframe of Reserved Vehicle for customer

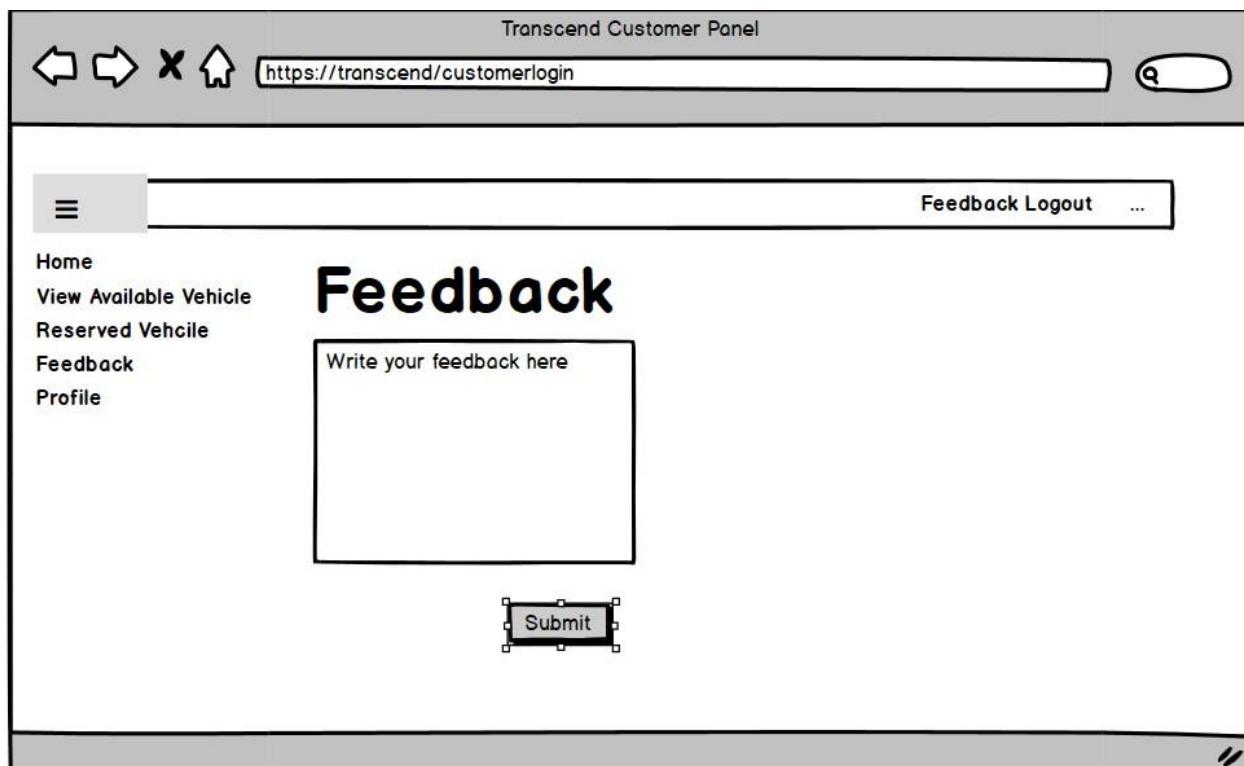


Figure 42: Wireframe of feedback for customer panel

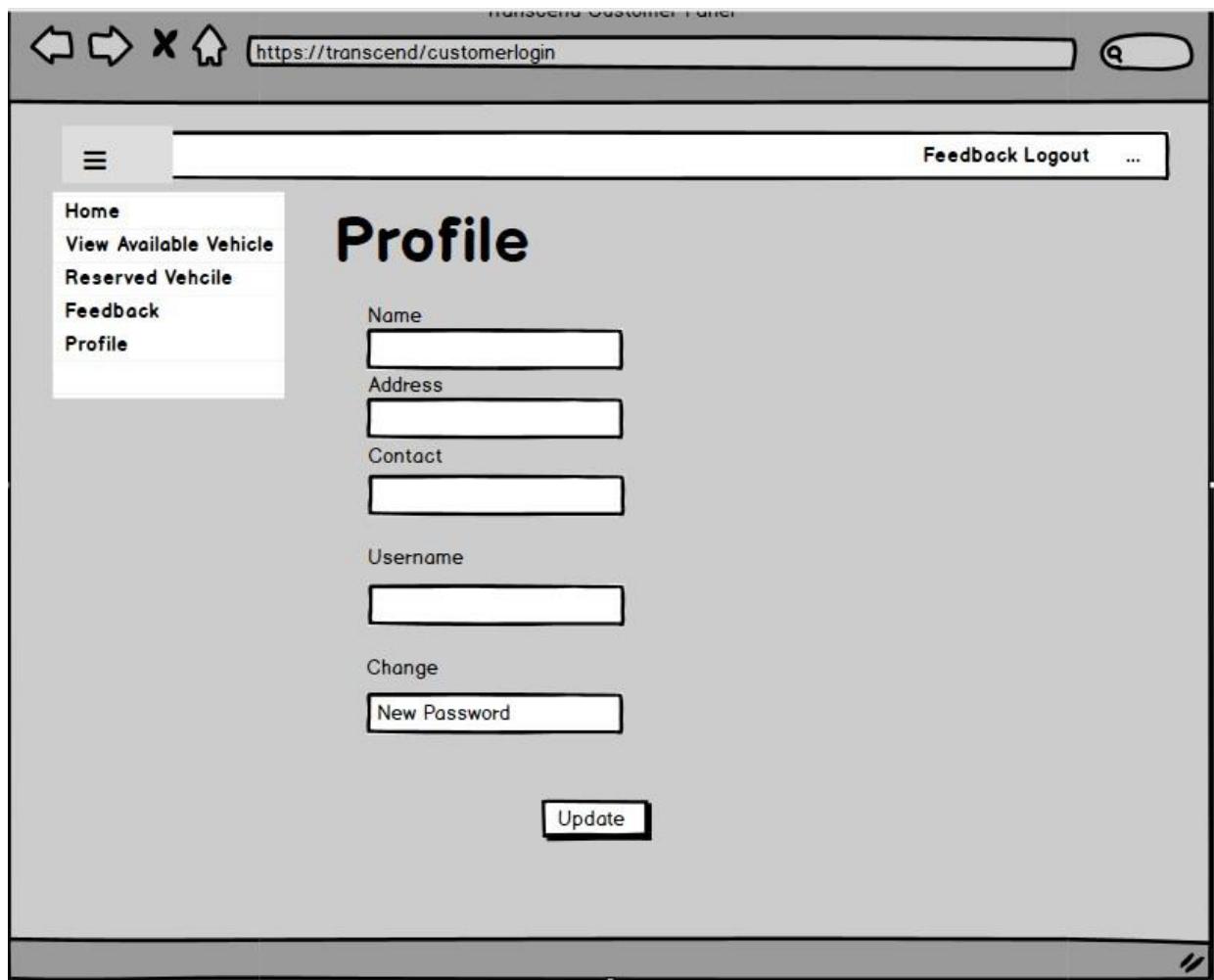


Figure 43: Wireframe of profile for customer panel



The wireframe depicts a web browser window with a light gray header bar containing standard navigation icons (back, forward, stop, home) and a search bar. The main content area has a white background. At the top center, the text "Forgot Password" is displayed in a large, bold, black font. Below this, there are two input fields: the first is labeled "Verification code" and the second is labeled "New password", both preceded by descriptive text. A single "Submit" button is centered below the input fields. The footer of the browser window is a dark gray bar.

Figure 44: Wireframe for Customer Forgot Password

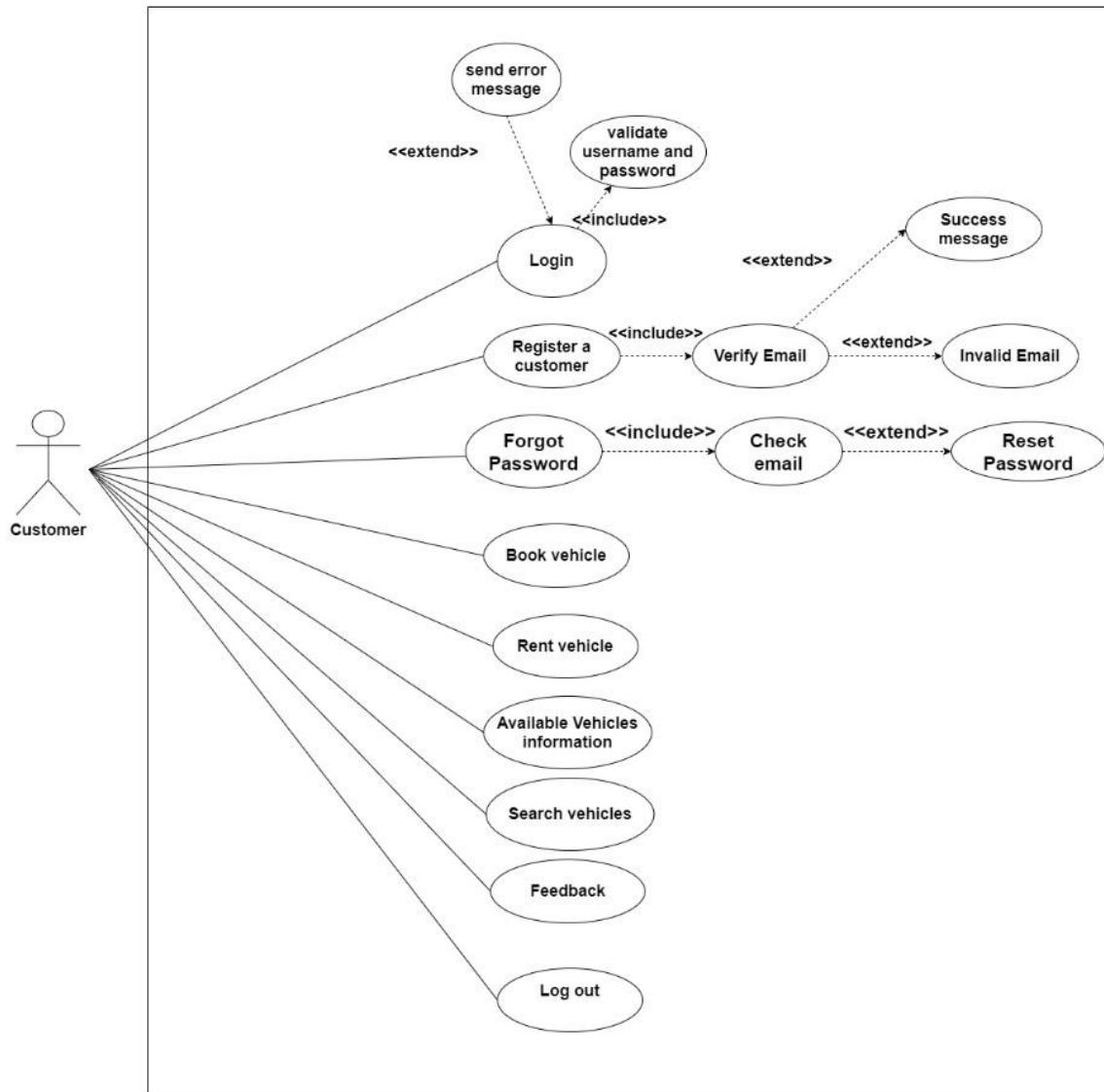
Use Case Diagram:

Figure 45: Use case diagram for customer

[High Level Use case diagram](#)

[Expanded Use case diagram](#)

Activity diagram

Register Customer

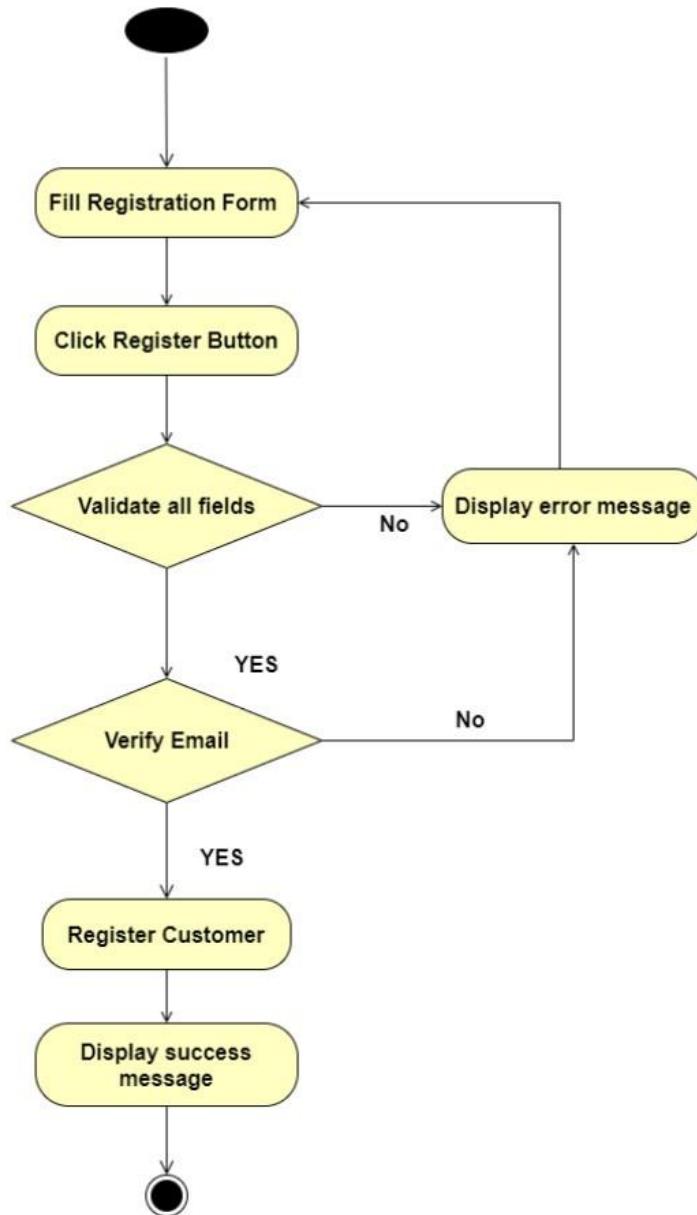


Figure 46: Activity Diagram for Customer registration

Customer Login

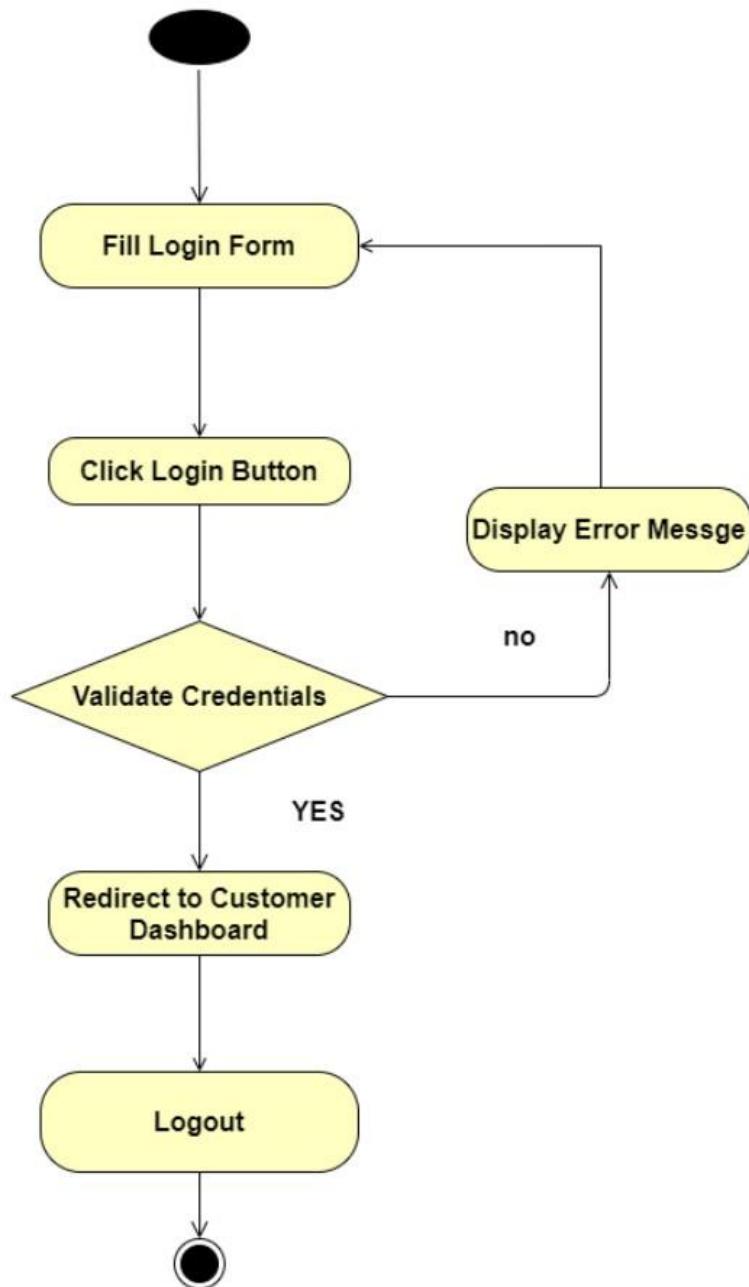


Figure 47: Activity Diagram for Customer Login

Forgot Password

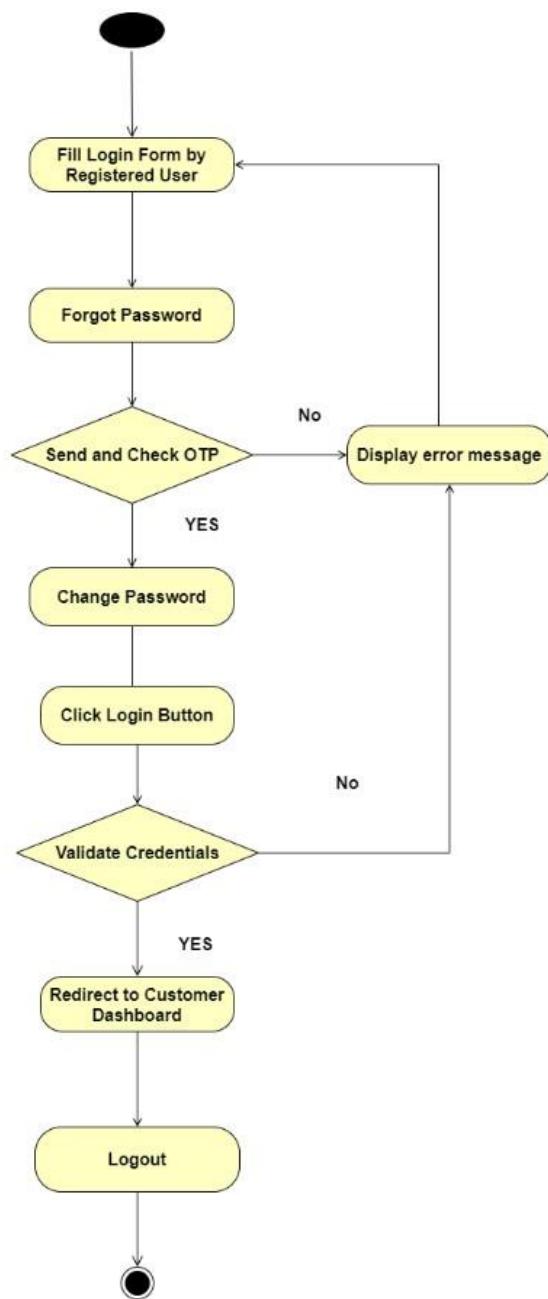


Figure 48: Activity Diagram for Customer Forgot password

Change Password

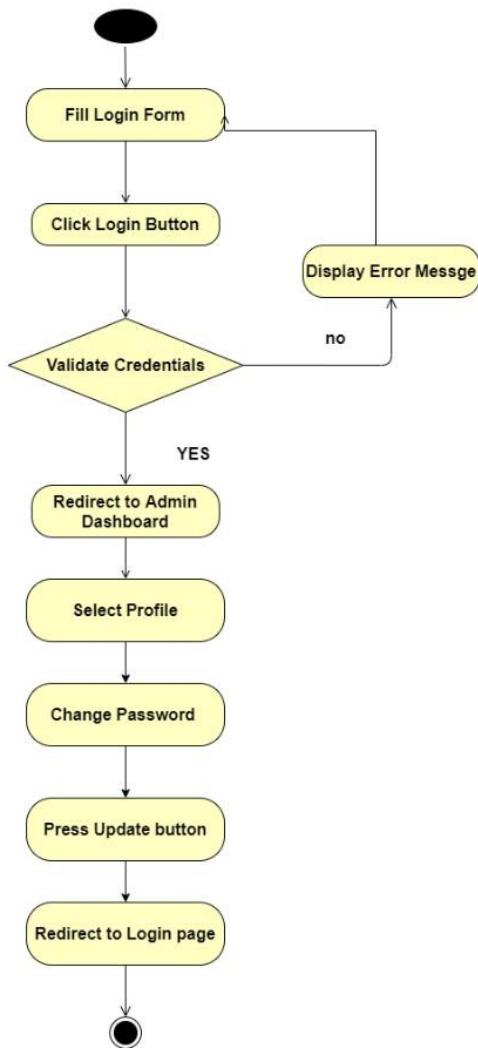
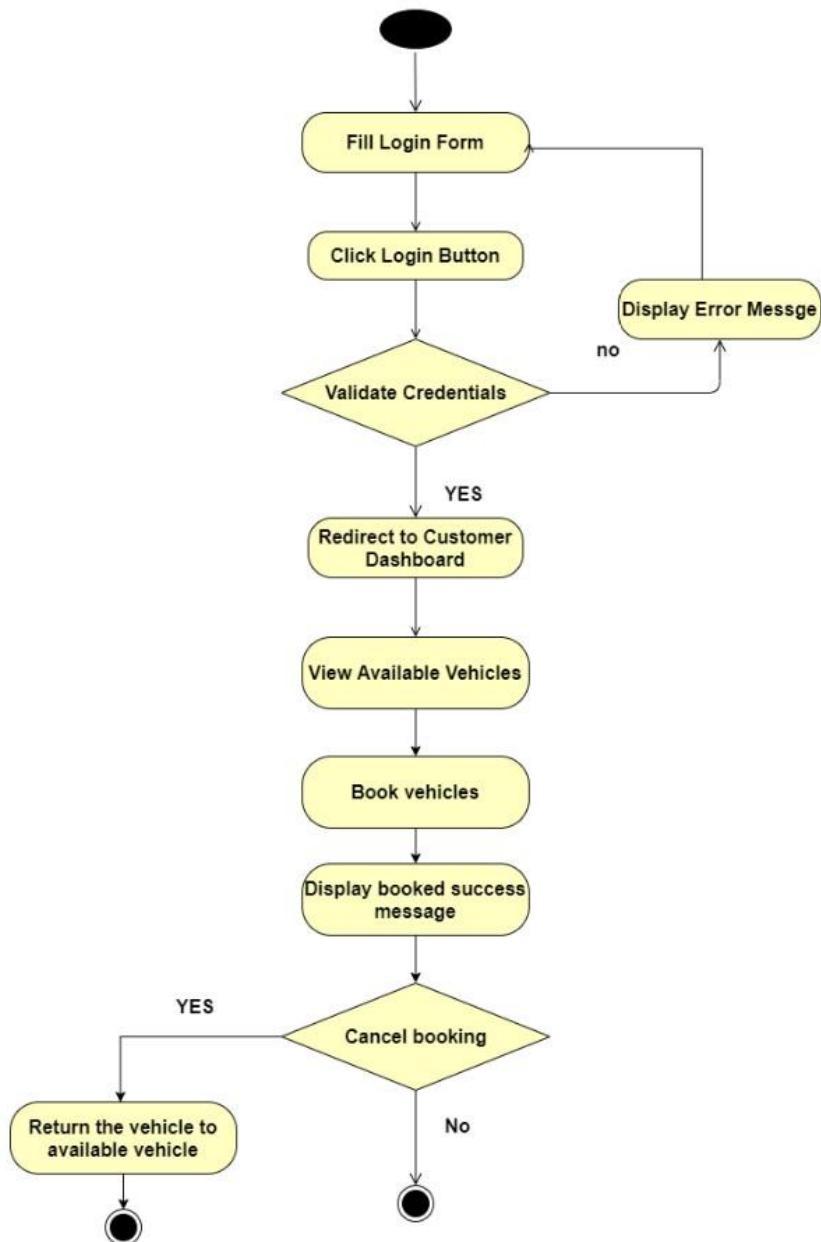


Figure 49: Activity diagram for customer password change

Book vehicles



Figure

50: Activity Diagram for Book vehicles

Rent vehicles

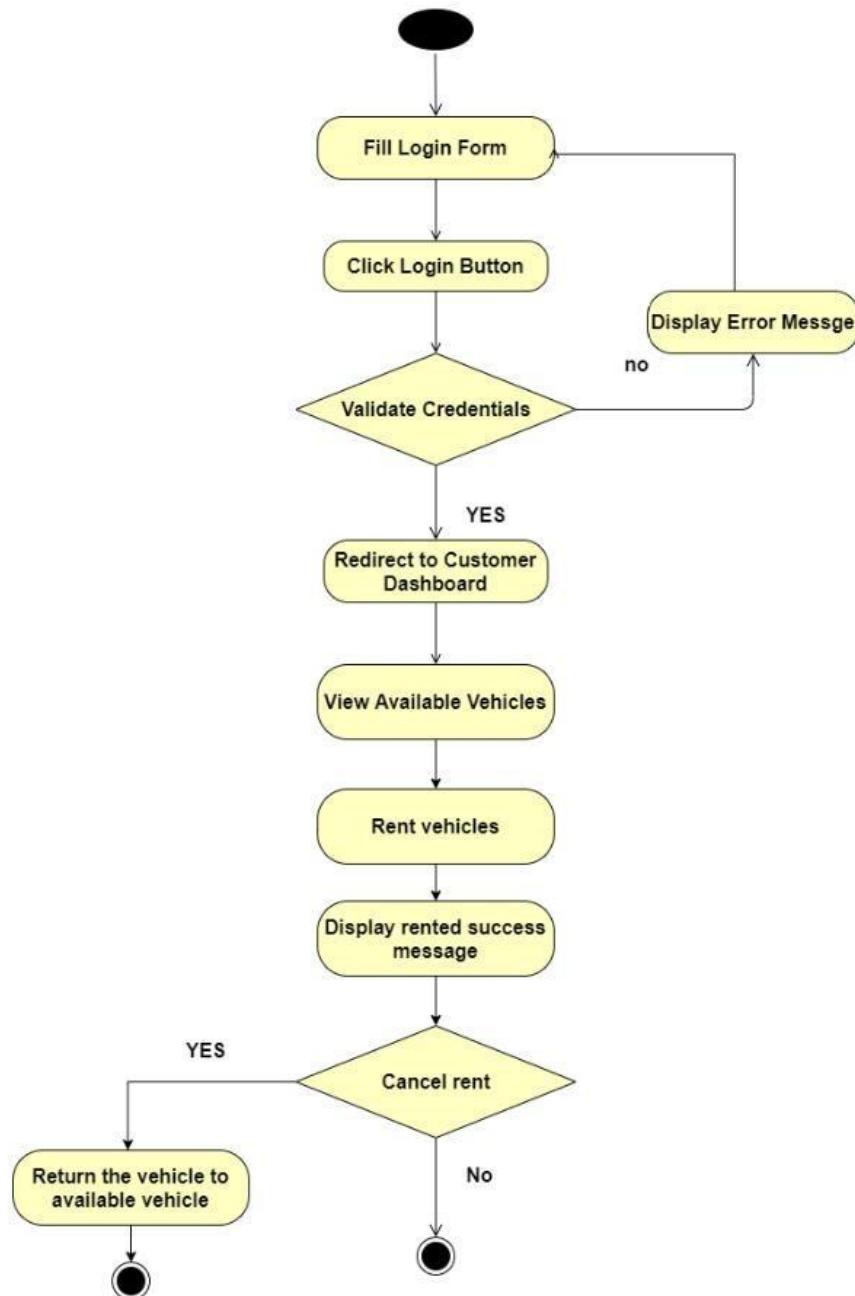


Figure 51: Activity Diagram for Rent vehicles

Send Feedback

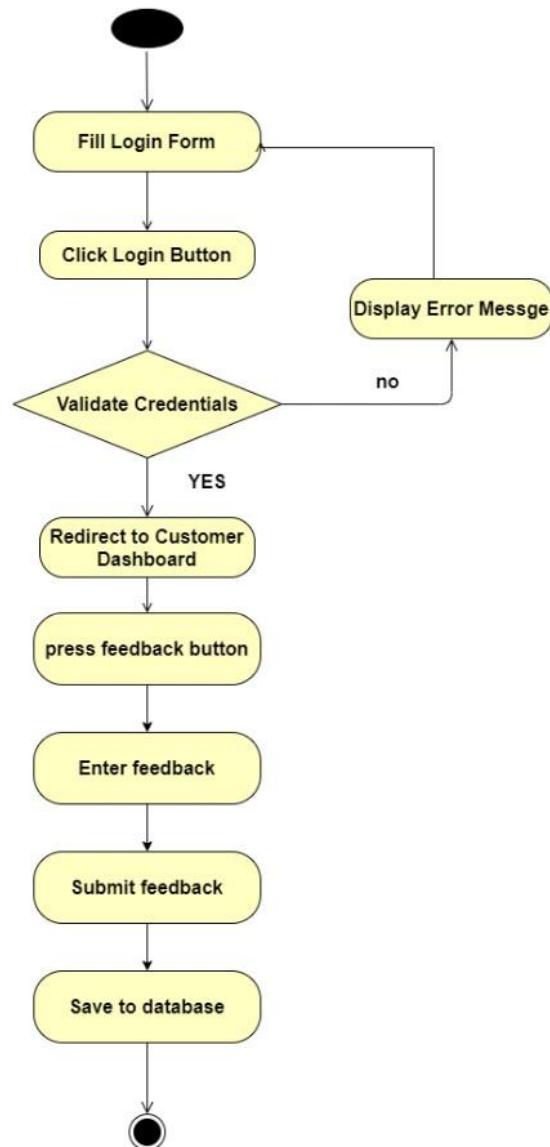


Figure 52: Activity Diagram for Sending feedbacks

Search Vehicles

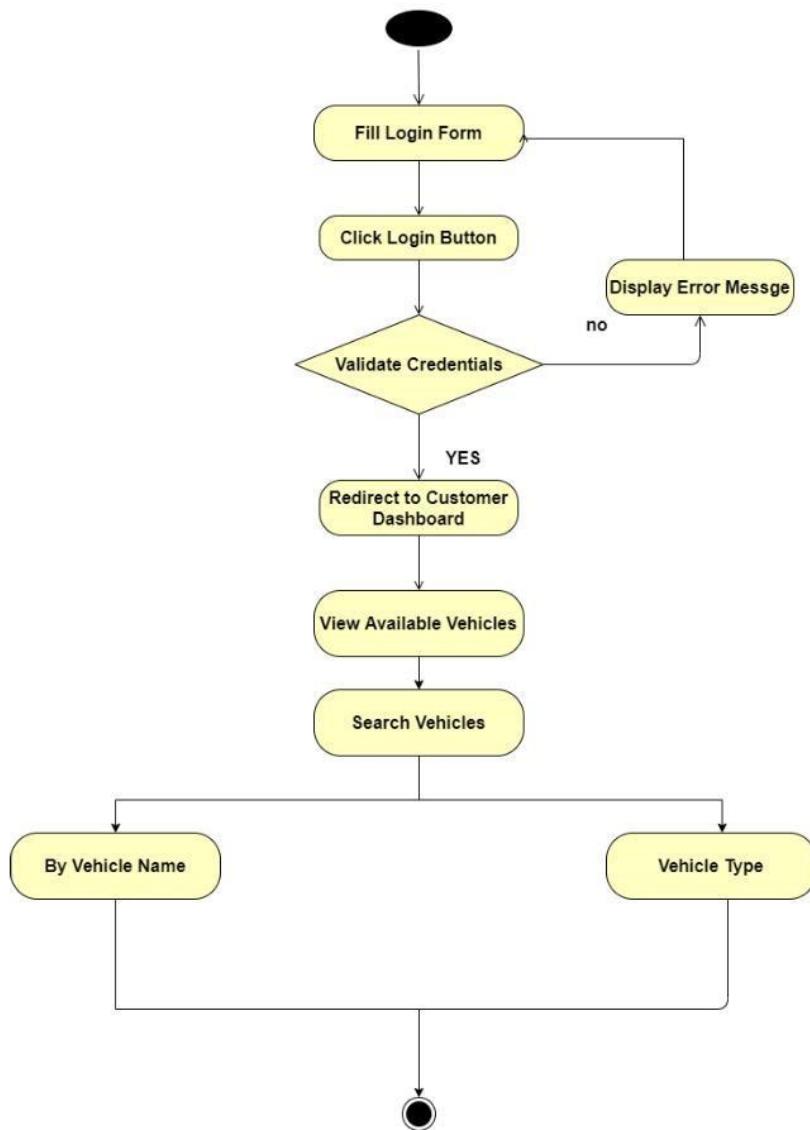


Figure 53: Activity Diagram Search vehicles

Sequence diagram

Register User

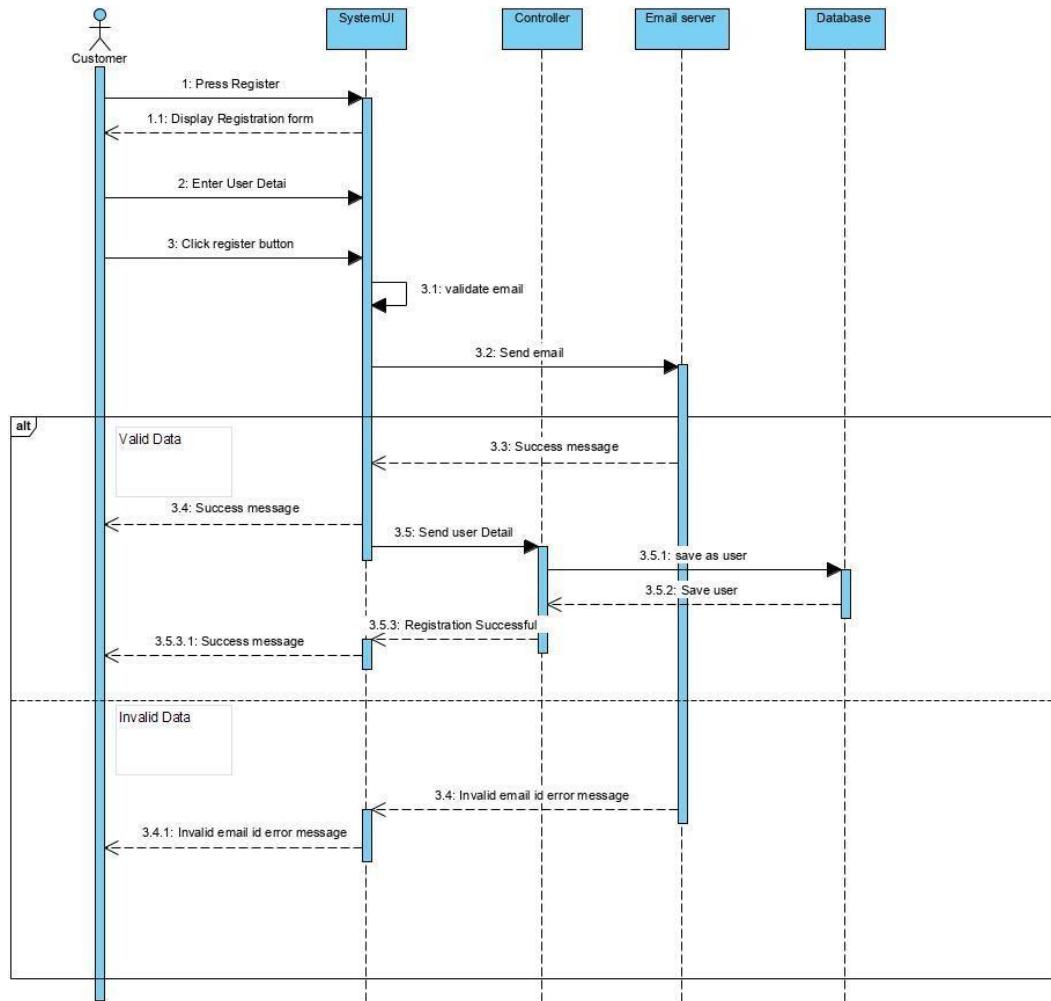


Figure 54: Sequence diagram for customer registration

Registered User Login

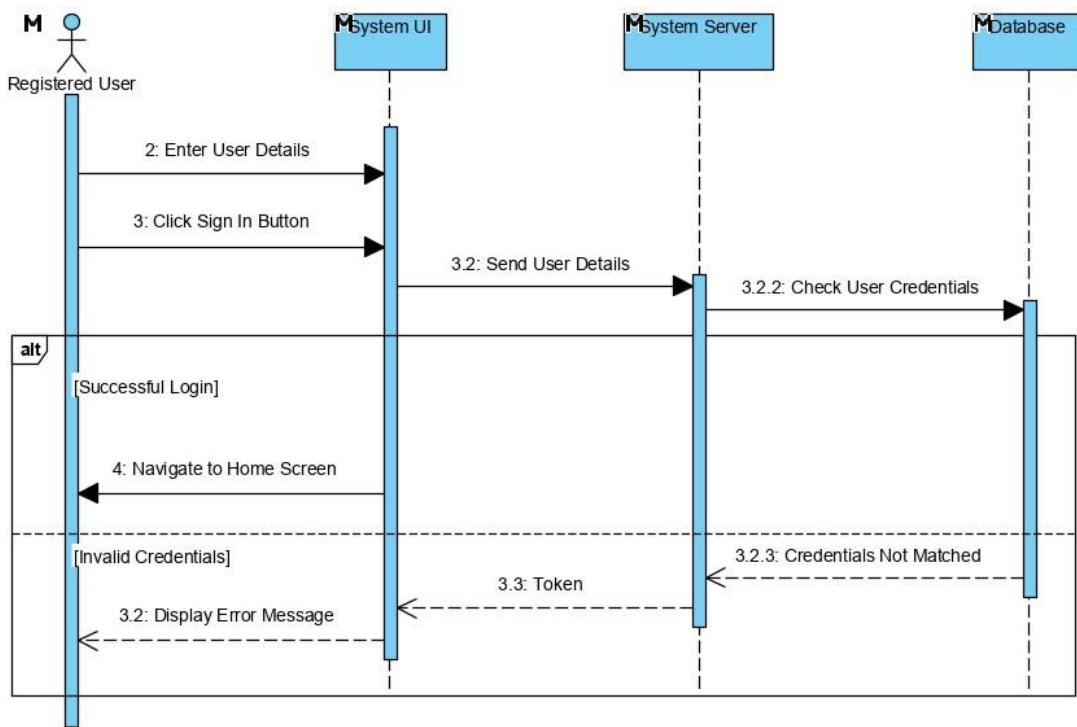


Figure 55: Sequence diagram for customer login

Change Password

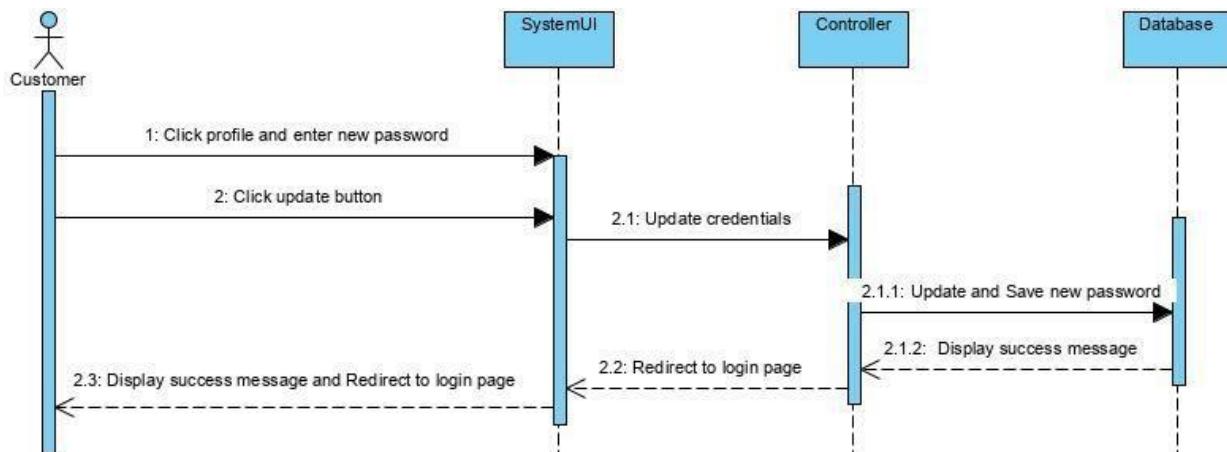


Figure 56: Sequence diagram for customer password change

Forgot Password

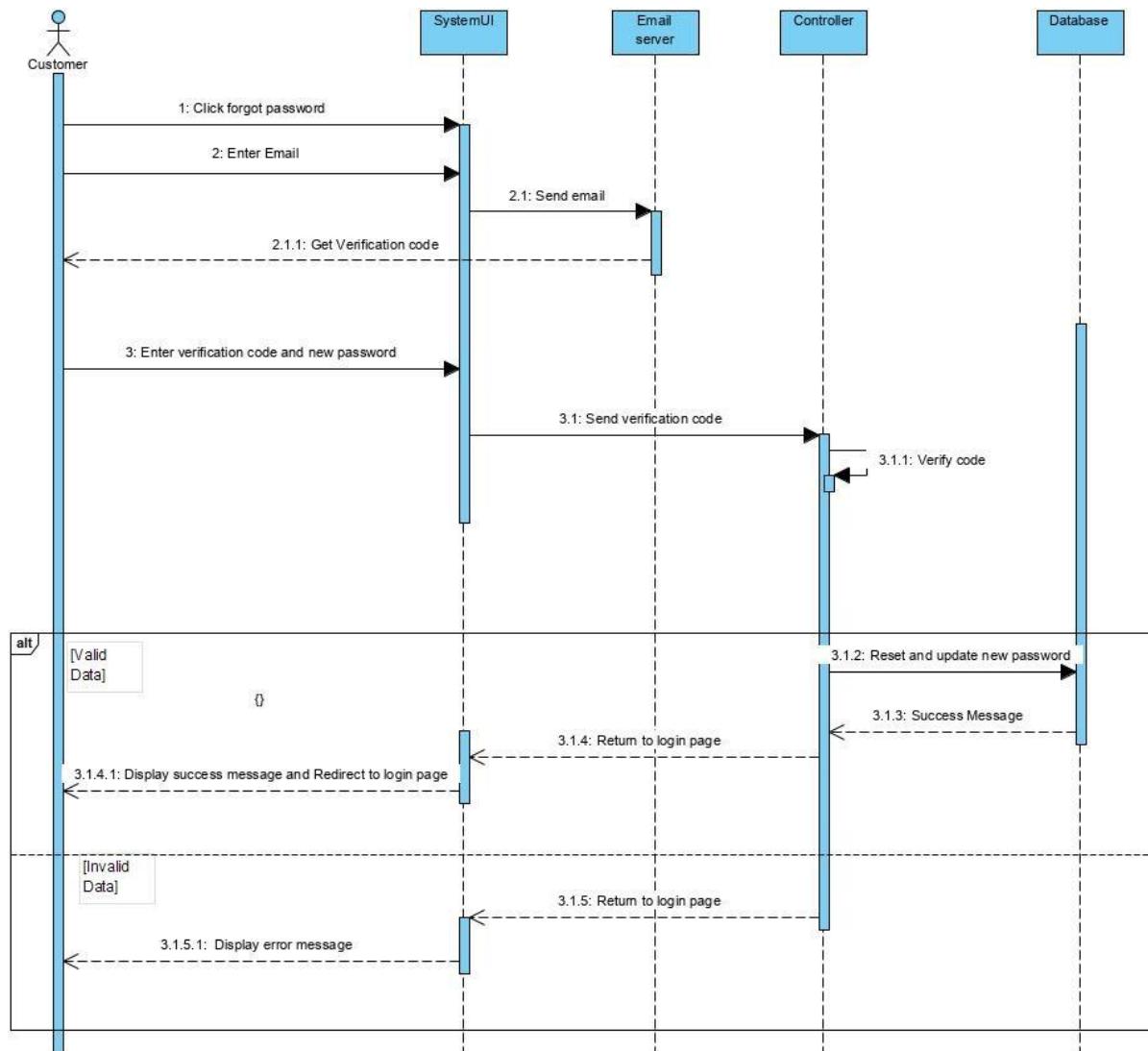


Figure 57: Sequence diagram for customer forgot password

Book vehicle

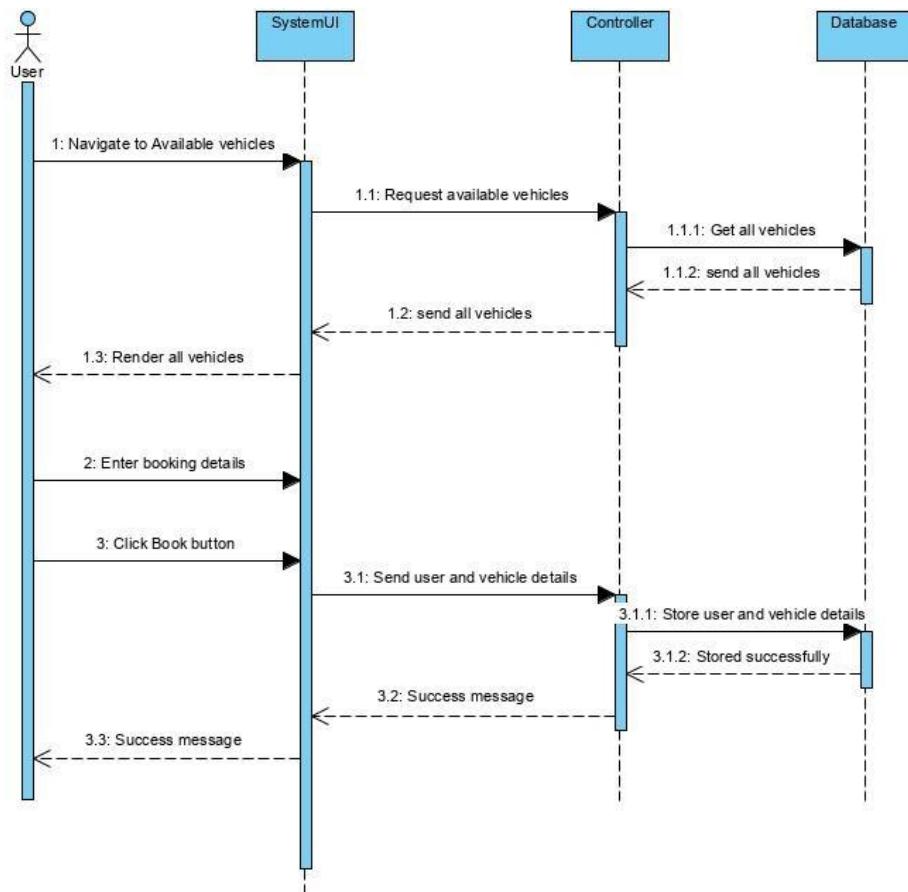


Figure 58: Sequence diagram for booking vehicle

Rent vehicle

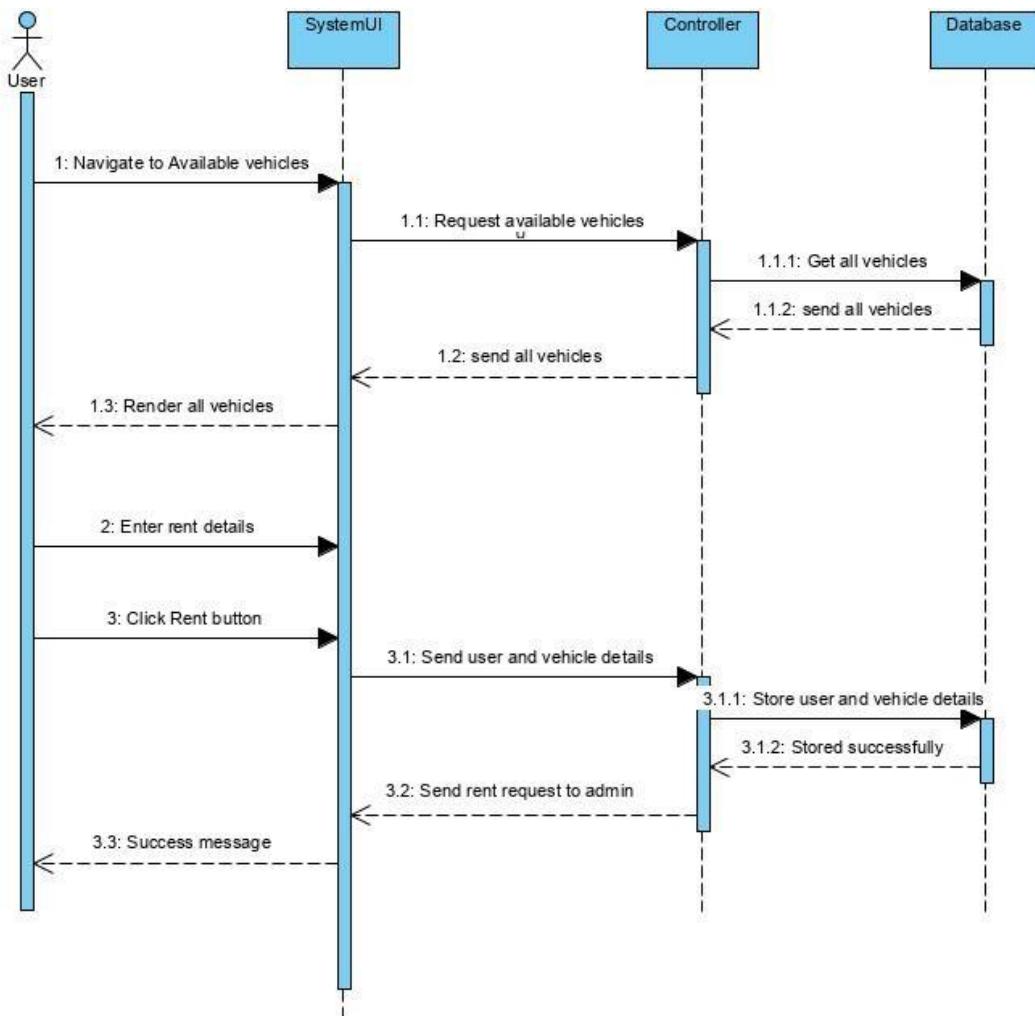


Figure 59: Sequence diagram for rent vehicle

Feedback

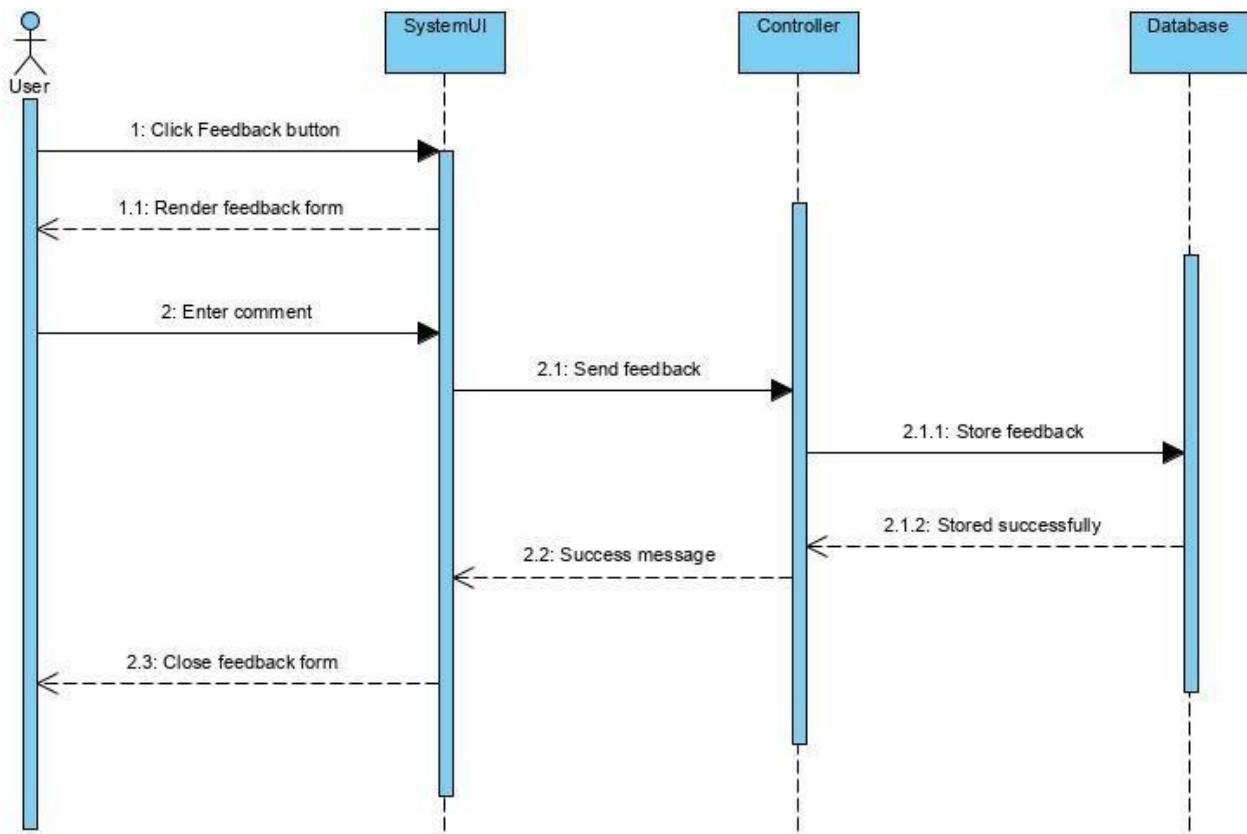


Figure 60: Sequence diagram for sending feedback

Collaboration diagram

Register Customer

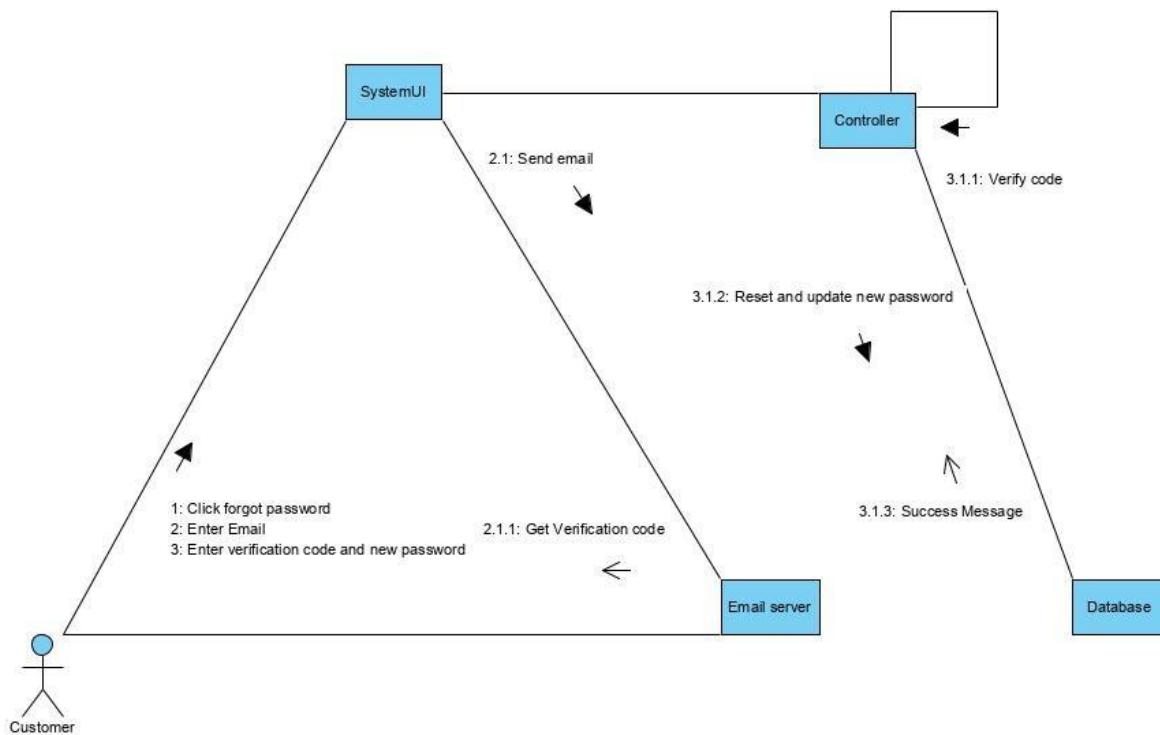


Figure 61: Collaboration diagram for customer registration

Customer Login

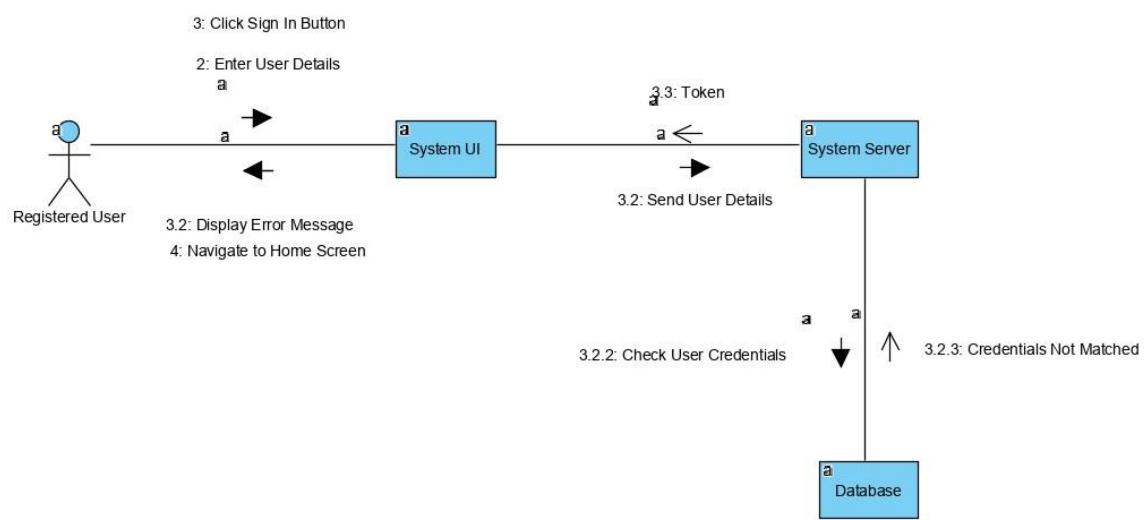


Figure 62: Collaboration diagram for customer login

Change Password

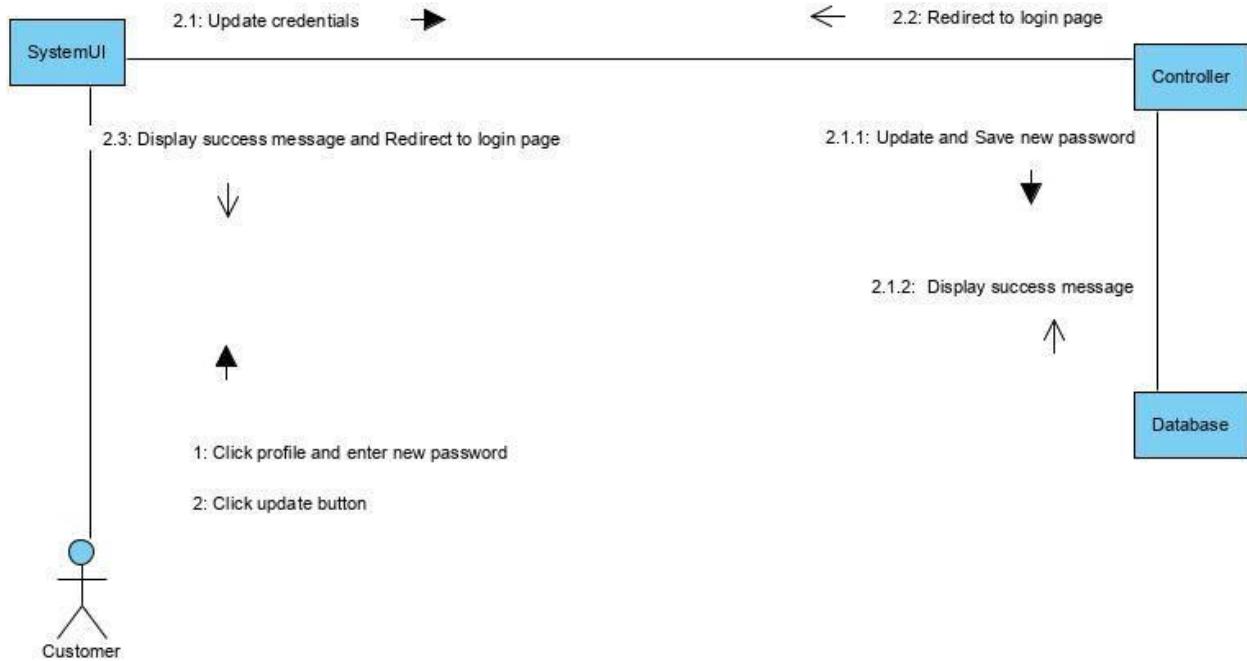


Figure 63: Collaboration diagram for customer password change

Forgot Password

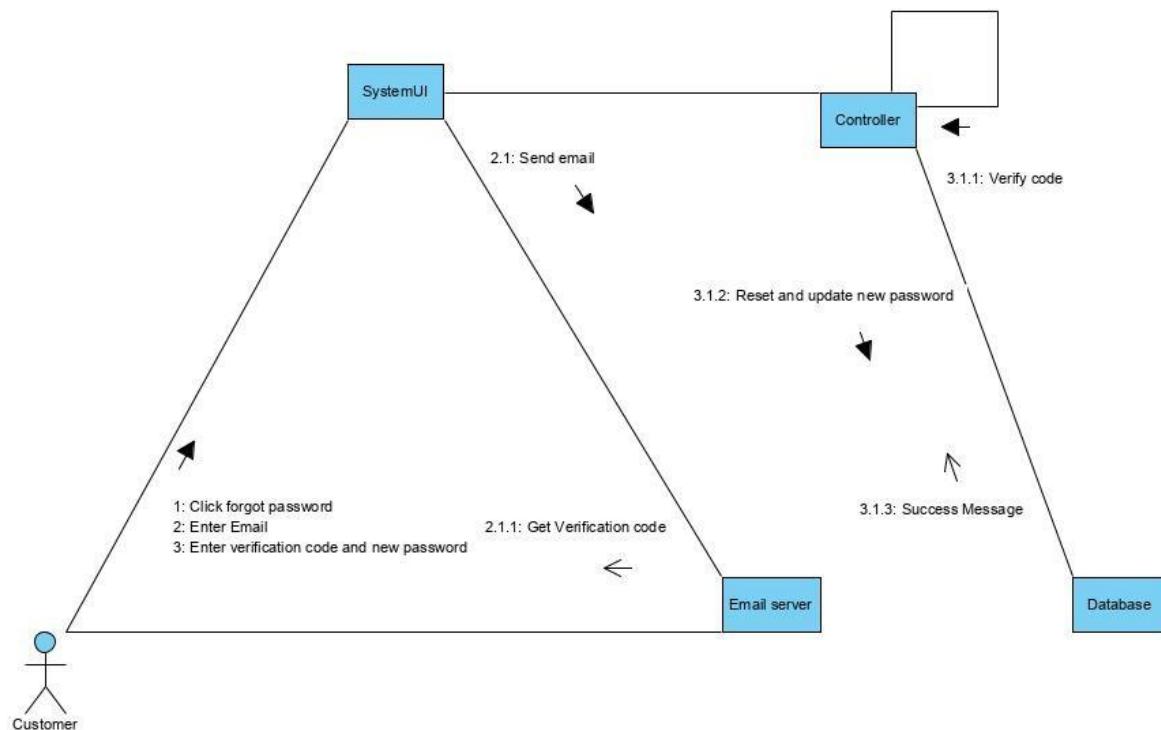


Figure 64: Collaboration diagram for customer forgot password

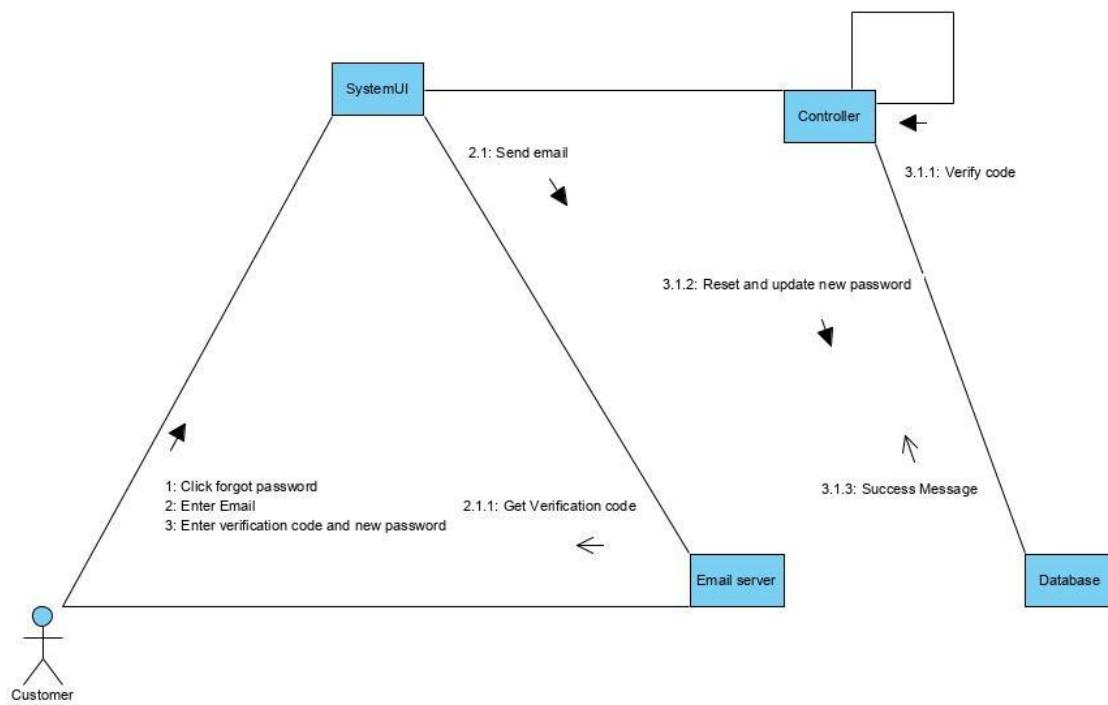
Book vehicle*Figure 65: Collaboration diagram for booking vehicle***Rent vehicle**



Figure 66: Collaboration diagram for rent vehicle

Feedback



Figure 67 : Collaboration diagram for feedback

3.5.3.3 Implementation:

Code Implementation:

Register:

Client

As a Client...

Full Name

Username

Email

Phone number

Date of Birth

Password

Figure 68: Register form UI

```
    onSubmit() {
      if (this.RegisterForm.invalid) {
        return;
      }
      this.dialog.open(VerifyingEmailDialogComponent).afterClosed().subscribe( next: () => {
        this.authService.register(this.RegisterForm.value).subscribe( next: res => {
          this.snackBar.open( message: 'Successfully Verified email!', action: 'Close', config: {duration: 2000, panelClass: ['success-snack-bar']});
          this.activeModal.dismiss();
        }, error: err => {
          this.snackBar.open( message: 'Verification Failed!', action: 'Close', config: {duration: 2000, panelClass: ['danger-snack-bar']});
        });
      });
    }
  }
}
```

Figure 69: User register method

In the above code register method **onSubmit** is used to register user after verifying email and user details which are then stored in the database in the table user with user type as “client”.

Book Vehicles:

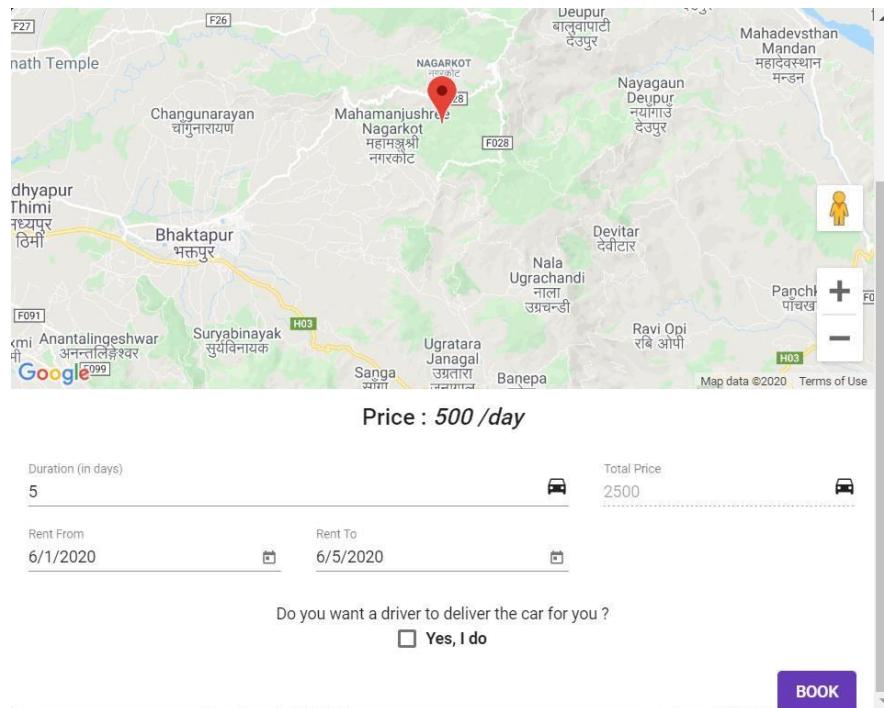


Figure 70: Book vehicle UI

```

onSubmit() {
  if (this.VehicleActionForm.invalid) {
    return;
  }
  this.data.locationCoordinates = this.VehicleActionForm.get('locationCoordinates').value;
  this.data.duration = this.VehicleActionForm.get('duration').value;
  this.data.rentFrom = this.VehicleActionForm.get('rentFrom').value;
  this.data.rentTo = this.VehicleActionForm.get('rentTo').value;
  this.data.wantDriver = this.VehicleActionForm.get('wantDriver').value;
  this.data.totalPrice = this.VehicleActionForm.get('totalPrice').value;
  this.vehicleService.rentAction(this.data).subscribe( next: res => {
    this.snackBar.open( message: 'Successfully reserved vehicle!', action: 'Close', config: {duration: 2000, panelClass: ['success-snack-bar']} );
    this.activeModal.close();
  }, error: err => {
    console.log(err);
  });
}
}

```

Figure 71: Book vehicle method

The above code method **onSubmit** is used to book vehicles. Vehicle along with booking details and user id of the customer is saved in the database. The status for vehicle is changed to book and is updated in the database.

Rent Vehicles:

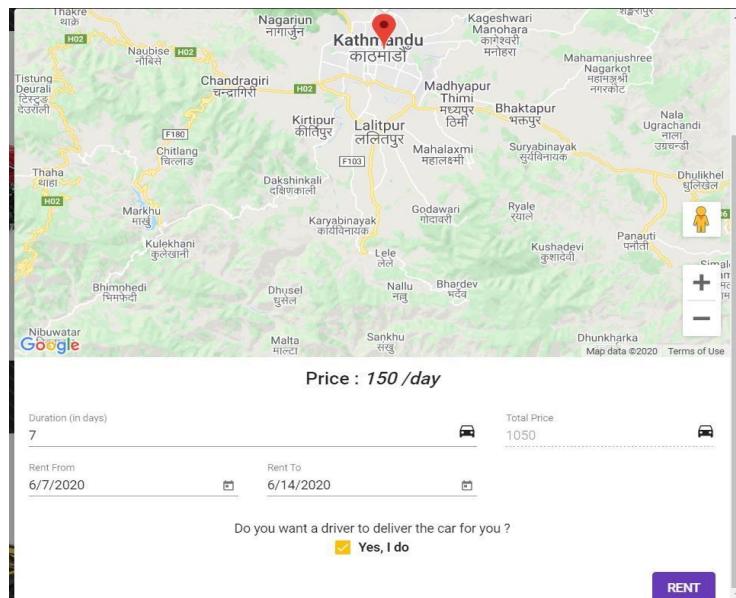


Figure 72: Rent vehicle UI

```

onSubmit() {
  if (this.VehicleActionForm.invalid) {
    return;
  }
  this.data.locationCoordinates = this.VehicleActionForm.get('locationCoordinates').value;
  this.data.duration = this.VehicleActionForm.get('duration').value;
  this.data.rentFrom = this.VehicleActionForm.get('rentFrom').value;
  this.data.rentTo = this.VehicleActionForm.get('rentTo').value;
  this.data.wantDriver = this.VehicleActionForm.get('wantDriver').value;
  this.data.totalPrice = this.VehicleActionForm.get('totalPrice').value;
  this.vehicleService.rentAction(this.data).subscribe( next: res => {
    this.snackBar.open( message: 'Successfully reserved vehicle!', action: 'Close', config: {duration: 2000, panelClass: [ 'success-snack-bar' ]});
    this.activeModal.close();
  }, error: err => {
    console.log(err);
  });
}
}

```

Figure 73: Rent Vehicle method

The above code method **onSubmit** is used to rent vehicles. Vehicle along with rent details and user id of the user is saved in the database. The status for vehicle is changed to rent and is updated in the database.

Search Vehicles:

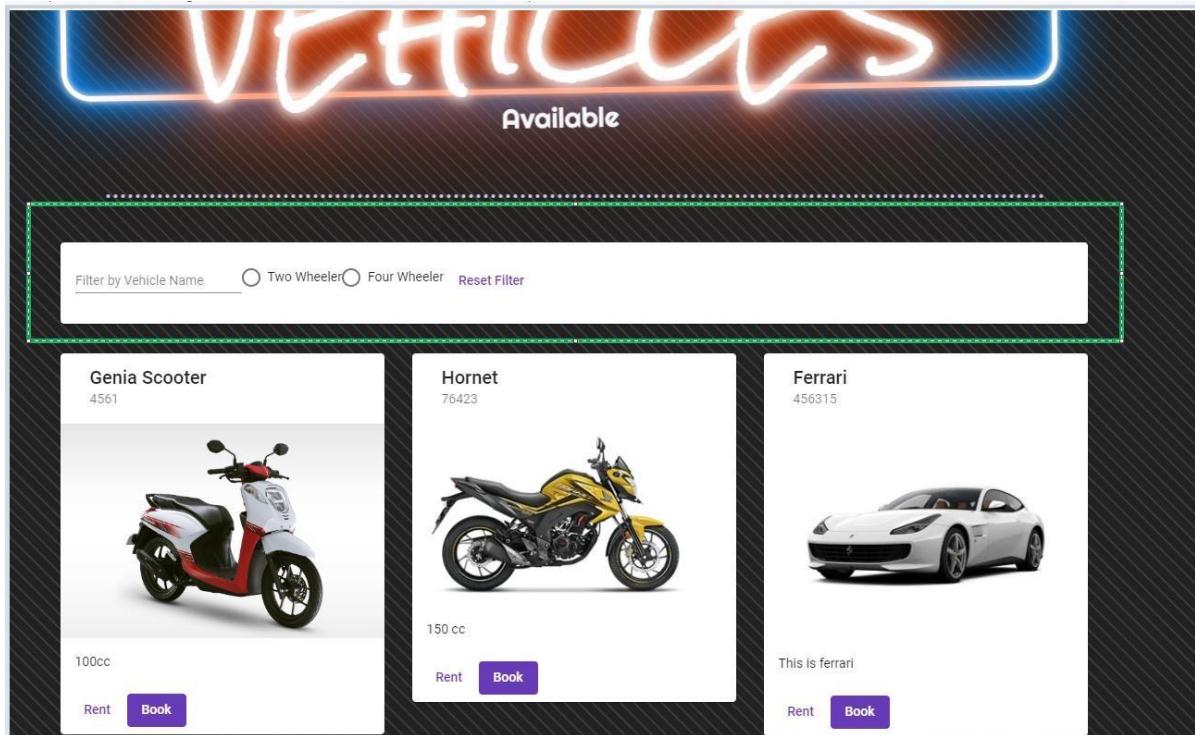


Figure 74: Search vehicle UI

```

searchFromVehicleName(name: string) {
    this.vehicleList = this.vehicleList.filter(vehicle => vehicle.name.toLowerCase().includes(name.toLowerCase()));
}

searchByType(type: string) {
    this.vehicleList = this.vehicleList.filter(vehicle => vehicle.type.toLocaleLowerCase() === type.toLowerCase());
}

```

Figure 75: Search Vehicle method

The above two methods are used to search the vehicle by vehicle name and vehicle type. These two methods filter the vehicle list according to vehicle name and vehicle type (two-wheeler, four-wheeler).

3.5.3.4 Testing:

[Click here to view testing for iteration 3](#)

3.5.3.5 Evaluation:

All the requirements gathered to be completed in this iteration was completed on time. The features developed for customer page was fully functional and turned out be satisfactory. Since admin as well as customer page was completed in this iteration overall functionality of the whole system was good. However some features like sending email confirmation to customer, view feedback and status of vehicles for admin were discussed which would be added in the next iteration.

3.5.4 Iteration 4:

3.5.4.1 Requirements:

All the main features of this application were completed before this iteration.

Requirements for this iteration was obtained through suggestion of external supervisor of FYP and client. Also feature like confirmation through email was one of the requested features obtained from the survey conducted on the beginning of the project.

Requirements Gathered:

- Admin: Approving and rejecting requests by client.
- Admin: Send email to client
- Admin: View vehicle report and feedback

3.5.4.2 Analysis and Design:

The requirement gathered for the user type admin was mainly an additional feature and confirmation of vehicle rent via email to customer. The wireframes for the admin page and related feature was designed. Then use case diagram along with collaboration, sequence, activity diagram was designed.

Wireframe

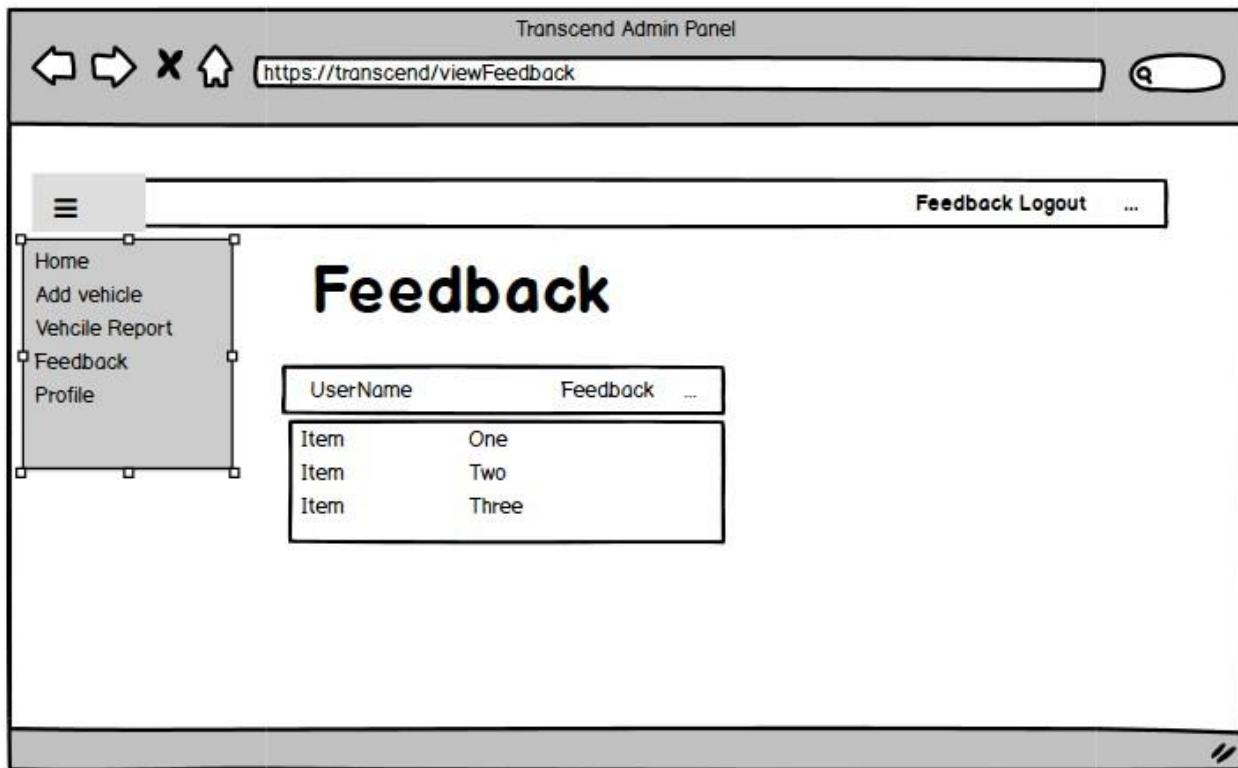


Figure 76: Wireframe of view feedback for admin

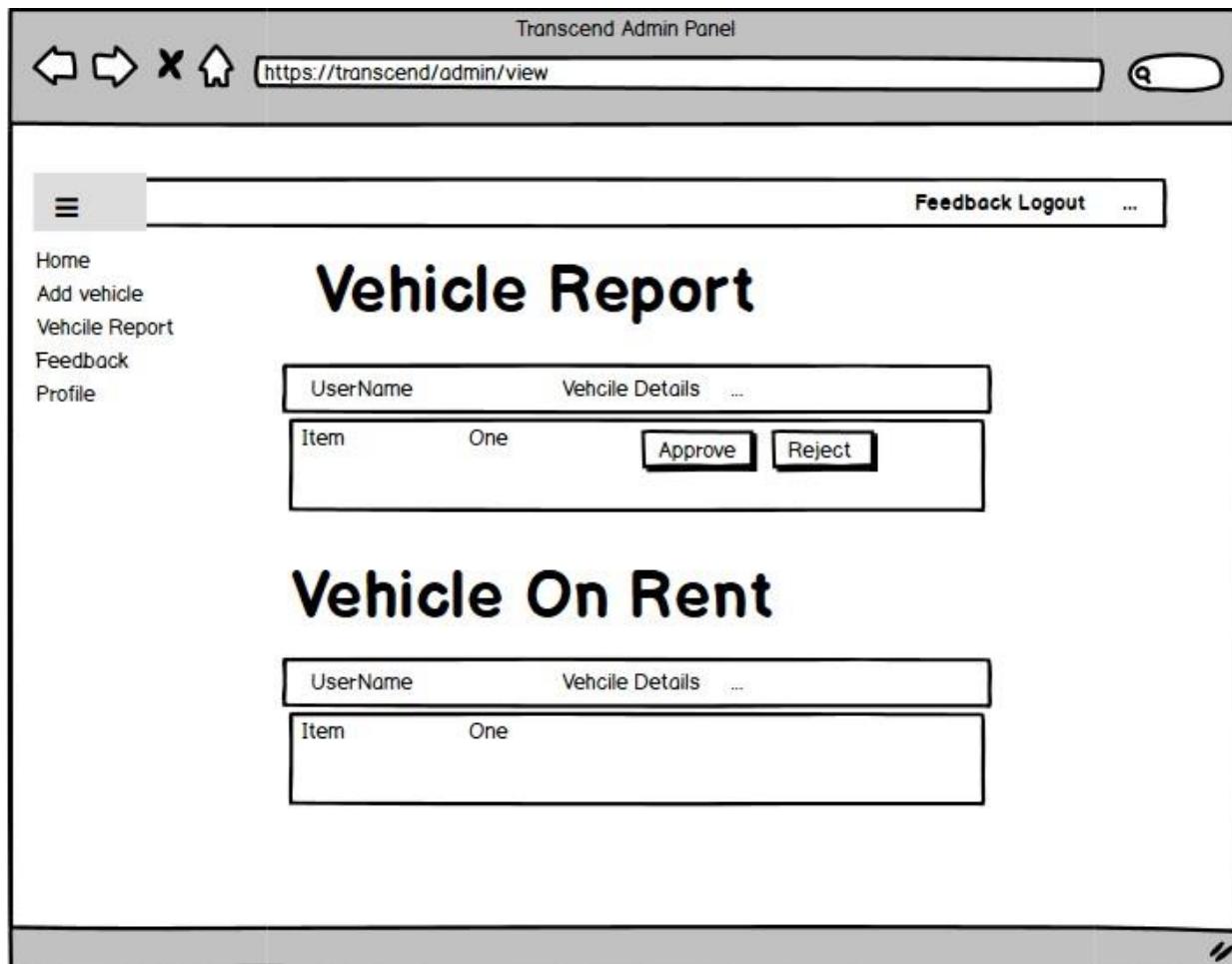


Figure 77: Wireframe of Vehicle Report for admin panel

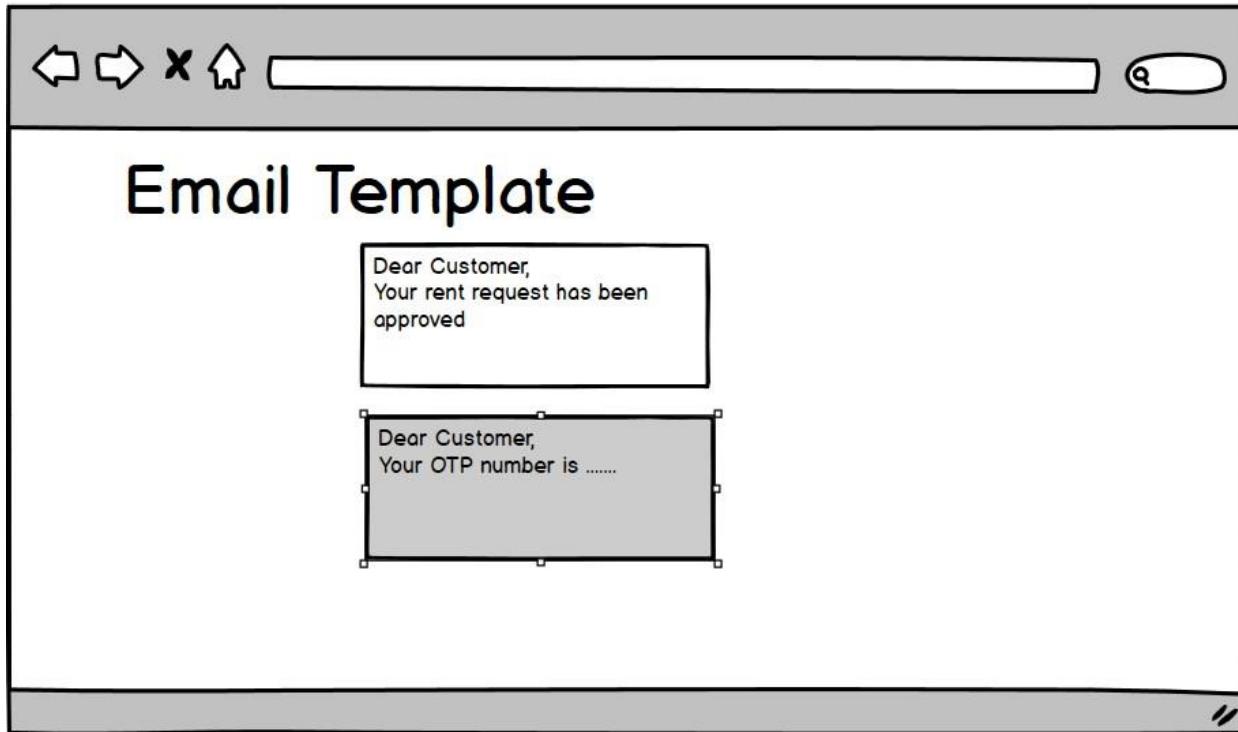


Figure 78: Wireframe for Email Template

Use case

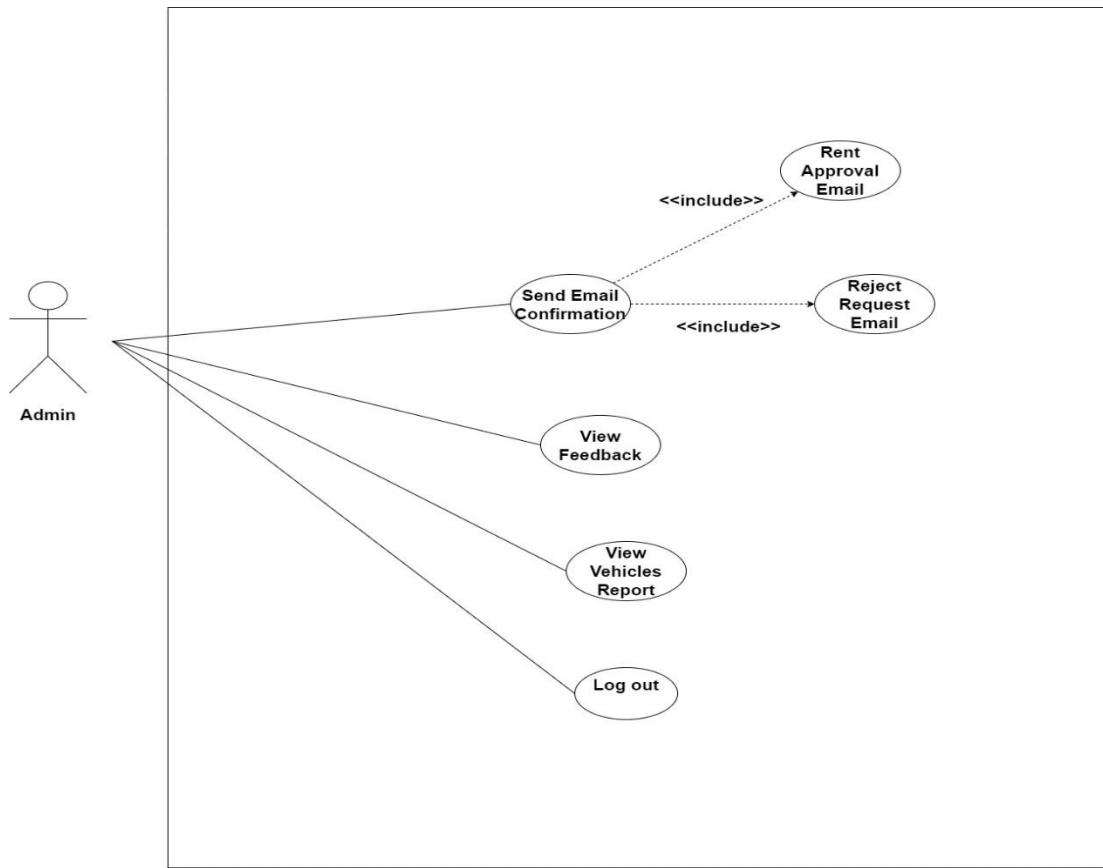


Figure 79: Use case diagram for Admin -2

[High Level Use case diagram](#)

[Expanded Use case diagram](#)

Activity diagram

View Feedback

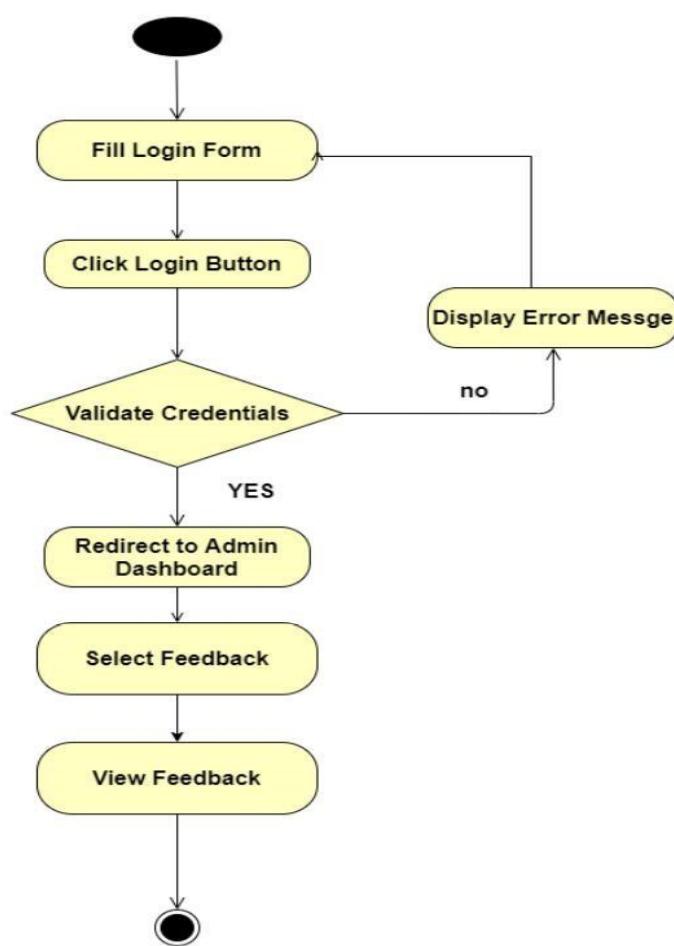


Figure 80: Activity Diagram for Admin view feedback

View Vehicle Report

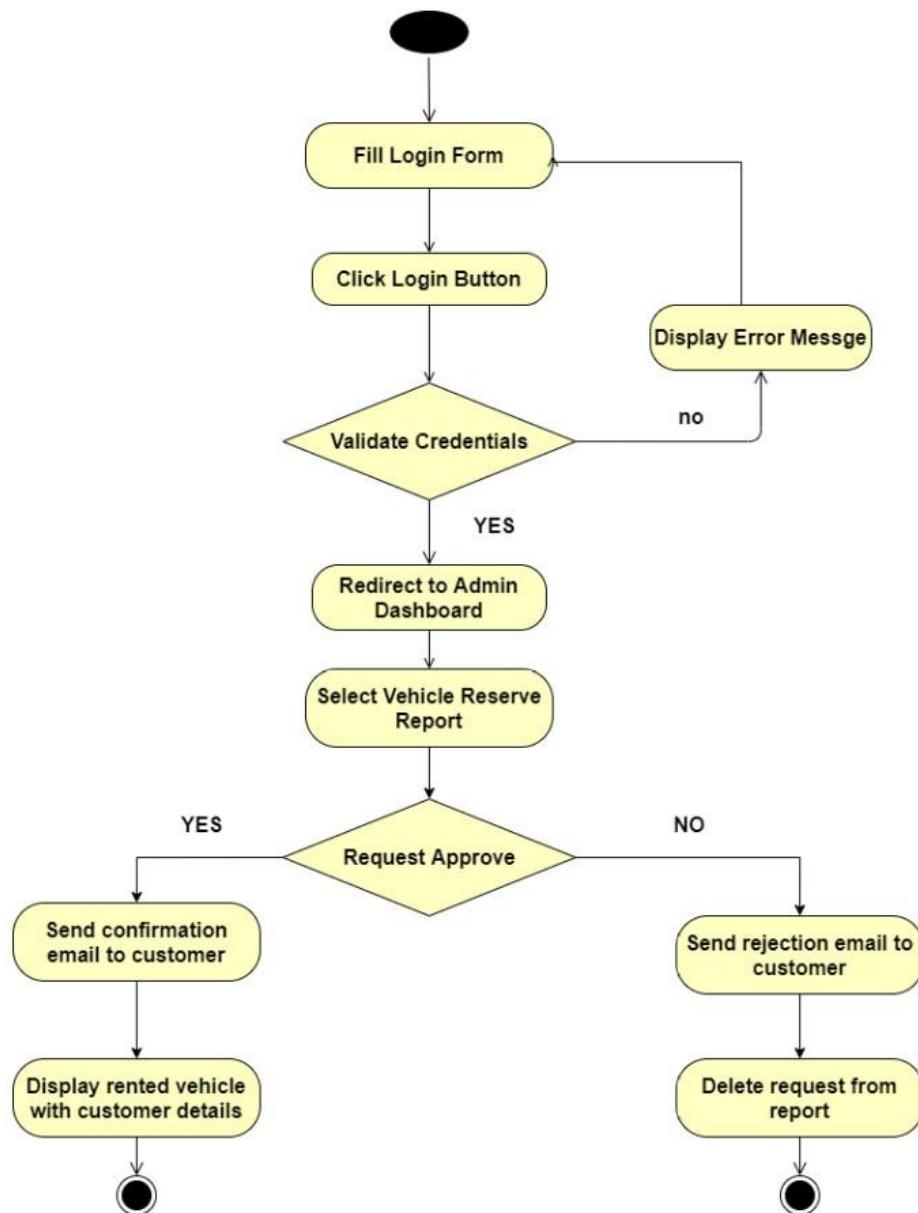


Figure 81: Activity Diagram for Admin view vehicle reserve report

Sequence diagram

View Feedback form

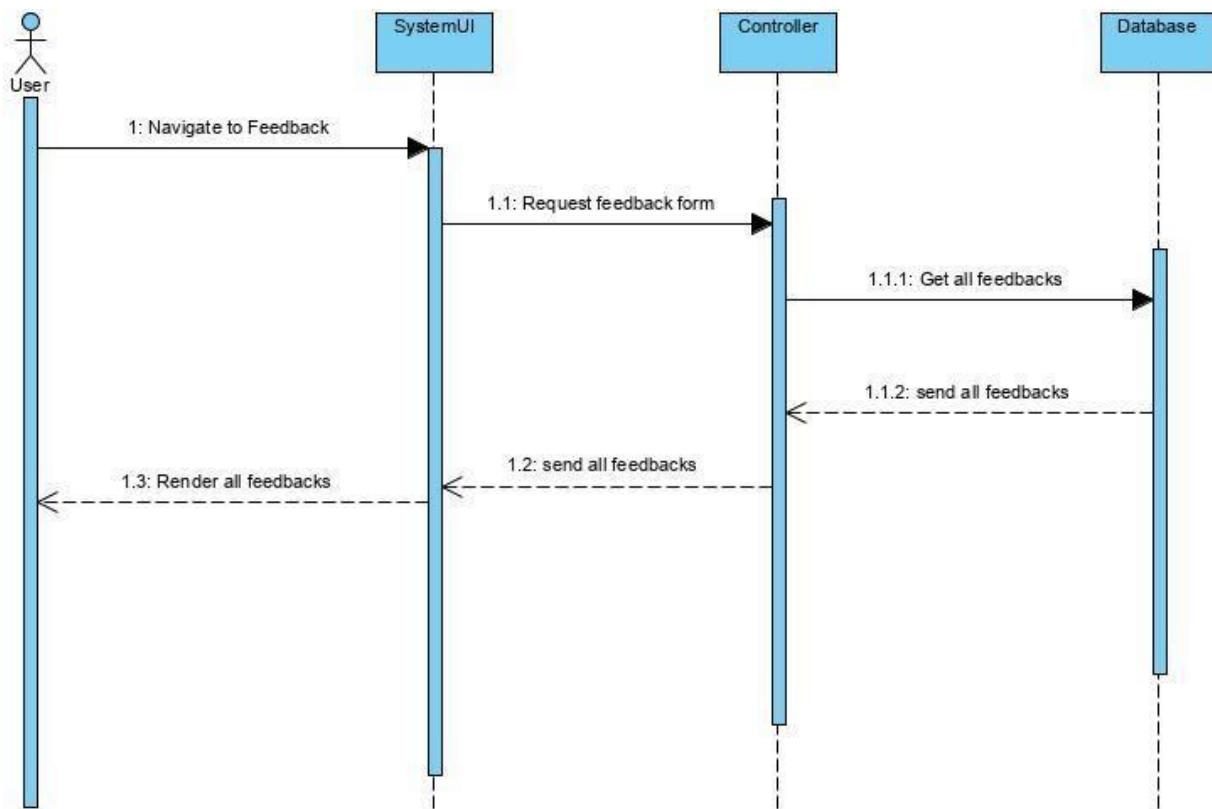


Figure 82: Sequence diagram for view feedback

View Vehicle Report

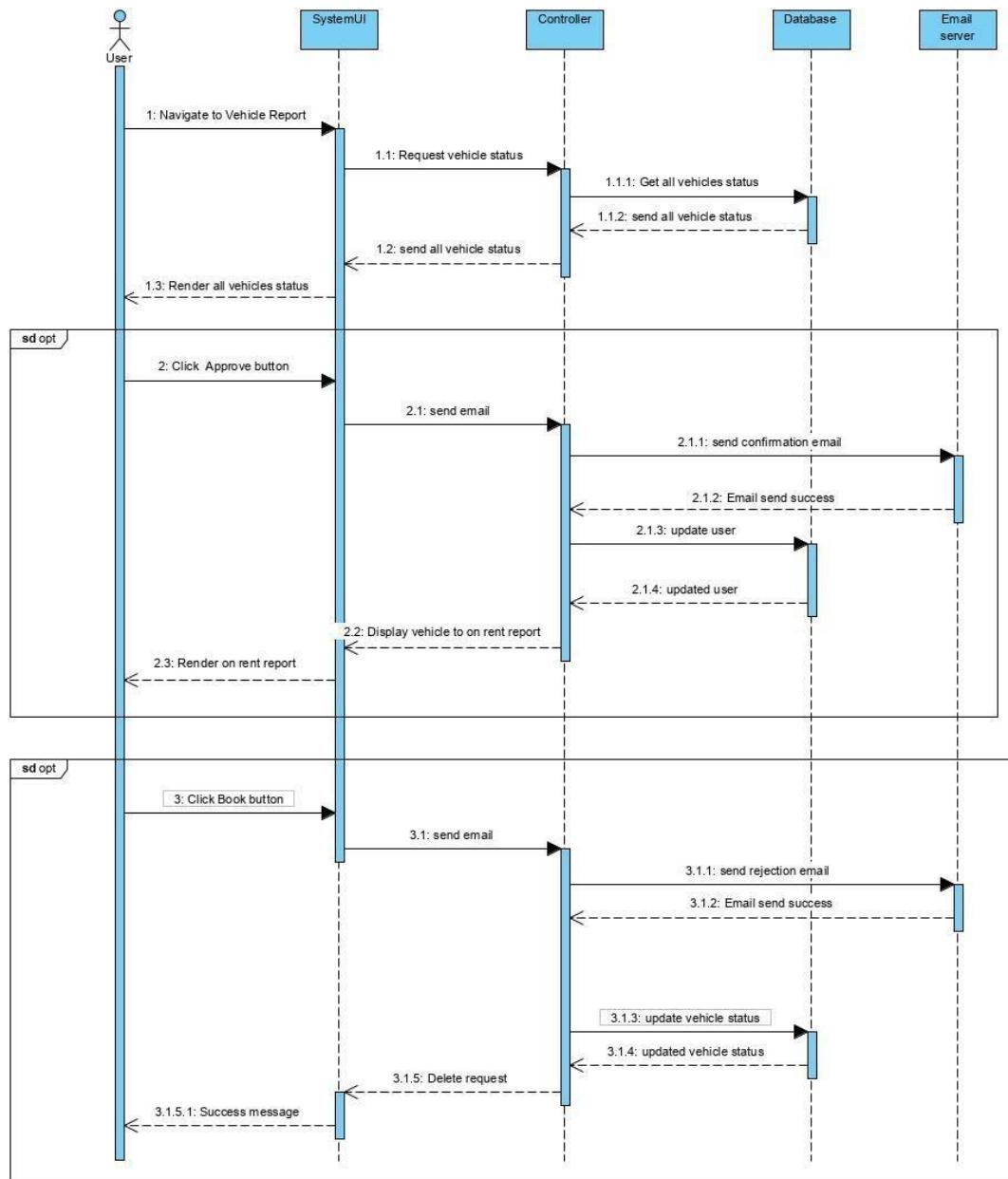


Figure 83: Sequence diagram for view vehicle report

Collaboration diagram

View Feedback form



Figure 84: Collaboration diagram for view feedback

View Vehicle Report

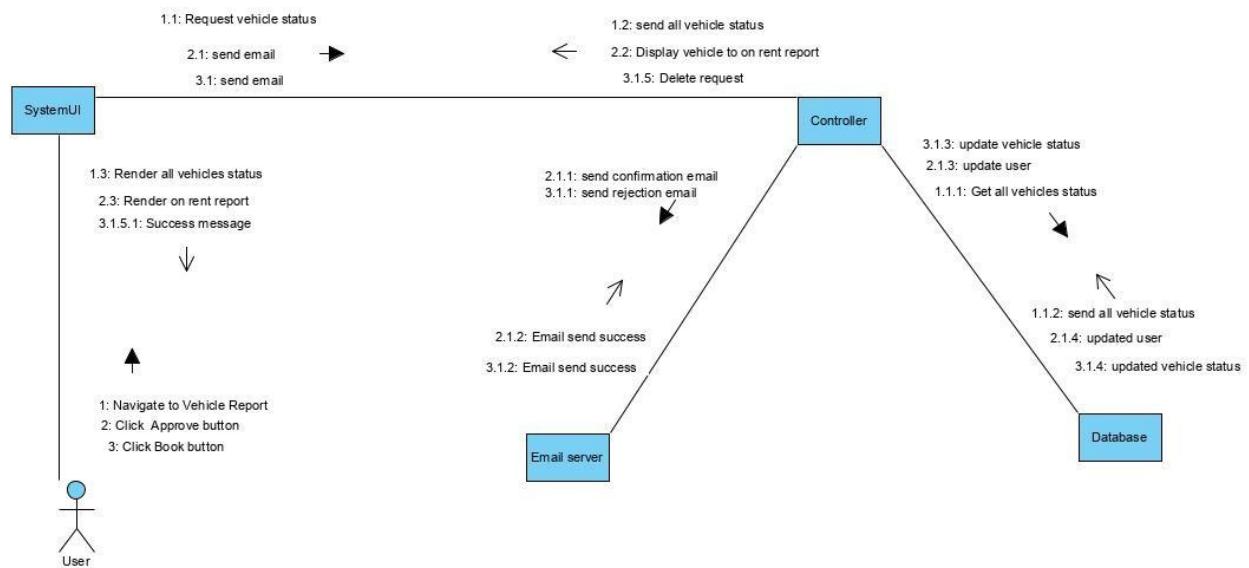
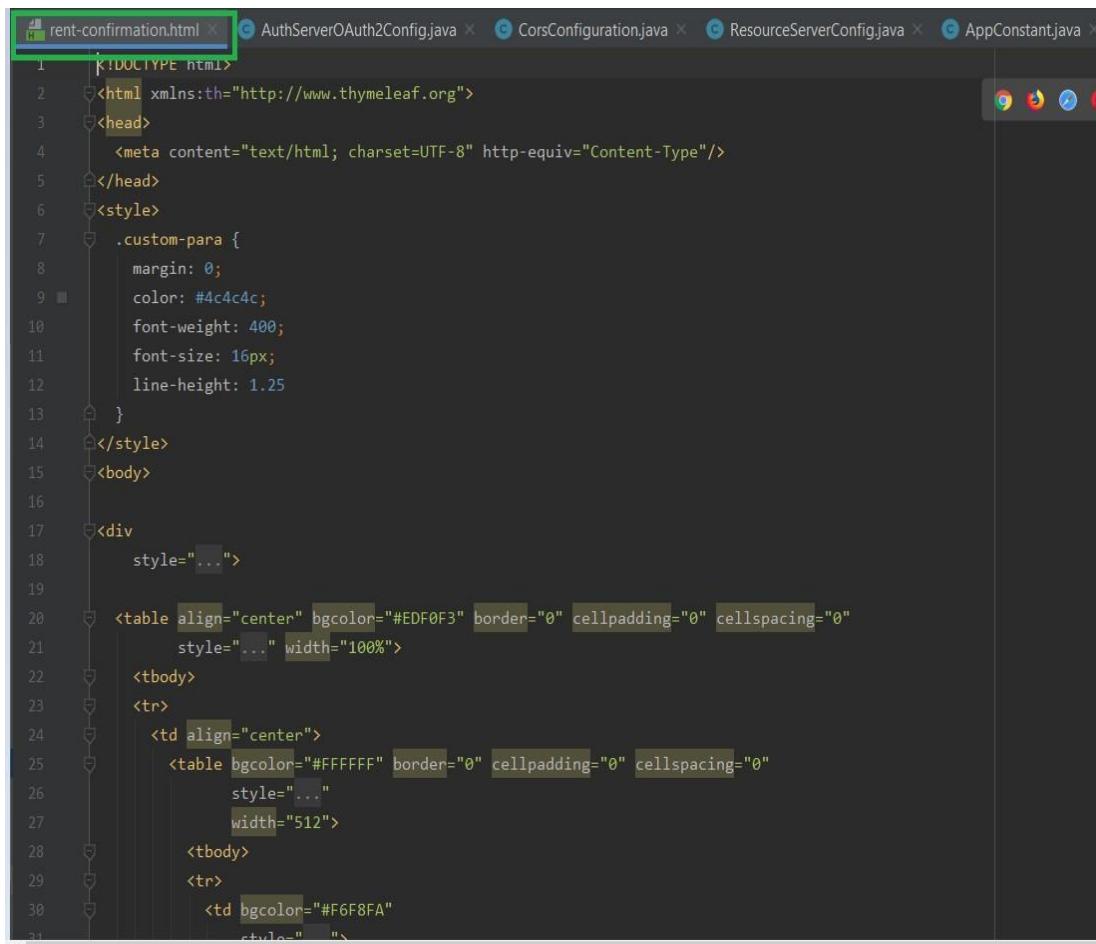


Figure 85: Collaboration diagram for view vehicle report

3.5.4.3 Implementation:

Code Implementation:

Send confirmation mail:



```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta content="text/html; charset=UTF-8" http-equiv="Content-Type"/>
5   </head>
6   <style>
7     .custom-para {
8       margin: 0;
9       color: #4c4c4c;
10      font-weight: 400;
11      font-size: 16px;
12      line-height: 1.25
13    }
14  </style>
15  <body>
16
17  <div
18    style="...">
19
20    <table align="center" bgcolor="#EDF0F3" border="0" cellpadding="0" cellspacing="0"
21      style="..." width="100%">
22      <tbody>
23        <tr>
24          <td align="center">
25            <table bgcolor="#FFFFFF" border="0" cellpadding="0" cellspacing="0"
26              style="..."
27              width="512">
28              <tbody>
29                <tr>
30                  <td bgcolor="#F6F8FA"
31                    style="...">

```

Figure 86: Email Template for sending email

```

43
44  @RequestMapping("/rentMail")
45  public ResponseEntity<?> sendRentMail(@RequestBody UserDto userDto) {
46    Mail mail = new Mail();
47    mail.setMailTo(userDto.getEmail());
48    mail.setToName(userDto.getUsername());
49    mailService.send(EmailConstant.Template.RENT_CONFIRMATION, mail);
50    logger.debug("Rent confirmation email sent.");
51    return new ResponseEntity<>(HttpStatus.OK);
52  }
53
54  @RequestMapping("/rentCancel")
55  public ResponseEntity<?> sendRentCancelMail(@RequestBody UserDto userDto) {
56    Mail mail = new Mail();
57    mail.setMailTo(userDto.getEmail());
58    mailService.send(EmailConstant.Template.RENT_CANCEL, mail);
59    return new ResponseEntity<>(HttpStatus.OK);
60  }
61

```

Figure 87: API to send email

These two http request or API is used to send email to user to notify them about approval and rejection of their requests.

Vehicle reserved by client:

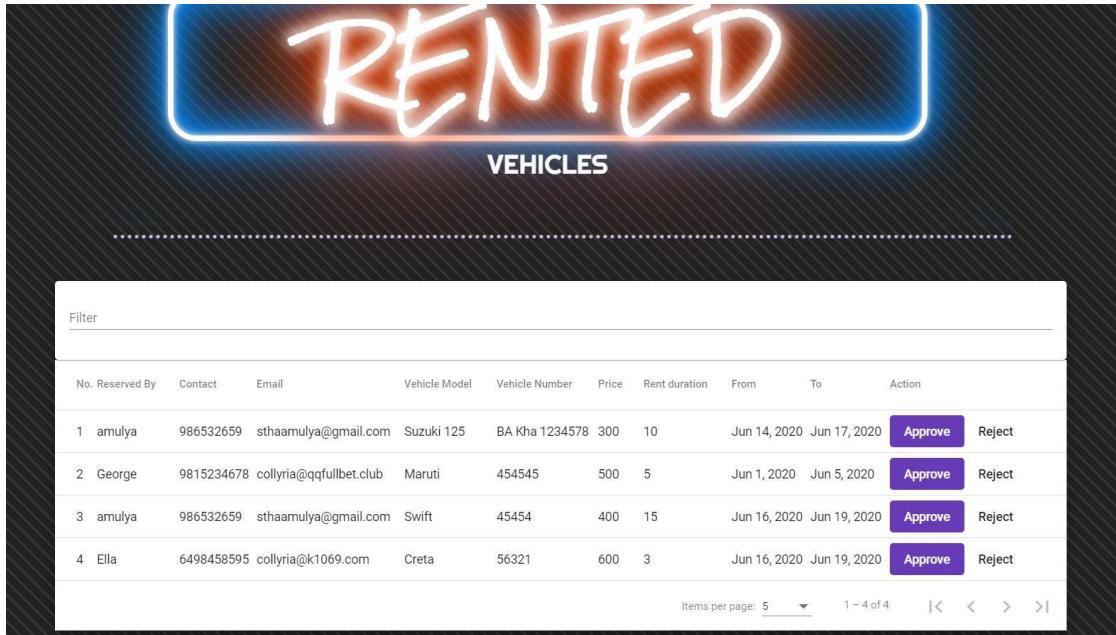


Figure 88: Vehicle reserved by client UI

```

applyFilter(event) {
  const filterValue = (event.target as HTMLInputElement).value;
  this.dataSource.filter = filterValue.trim().toLowerCase();
}

onActionApprove(vehicle) {
  this.spinner = true;
  this.vehicleService.finalRentAction(vehicle).subscribe( next: res => {
    this.snackBar.open( message: 'Successfully rented vehicle!', action: 'Close', config: {duration: 2000, panelClass: ['success-snack-bar']} );
    this.ngOnInit();
    this.spinner = false;
  }, error: error => {
    this.spinner = false;
    this.snackBar.open( message: 'Action failed!!', action: 'Close', config: {duration: 2000, panelClass: ['danger-snack-bar']} );
  });
}

onActionReject(vehicle) {
  this.spinner = true;
  this.vehicleService.rentAction(vehicle).subscribe( next: res => {
    this.ngOnInit();
    this.spinner = false;
    this.snackBar.open( message: 'Rejected successfully!', action: 'Close', config: {duration: 2000, panelClass: ['success-snack-bar']} );
  }, error: error => {
    this.spinner = false;
    this.snackBar.open( message: 'Action failed!!', action: 'Close', config: {duration: 2000, panelClass: ['danger-snack-bar']} );
  });
}
}
]

```

Figure 89: Vehicle Approve and Reject method

The above two method **onActionApprove** and **onActionReject** is used to approve and reject the request made by client.

3.5.4.4 Testing:

[Click here to view testing for iteration 4](#)

3.5.4.5 Evaluation:

All the features needed for the development for this project was completed after the completion of this iteration. This iteration was quite short compared to other three iterations as it did not require any new additional features and new requirements. Overall system and its features were analyzed along with the possible improvements.

3.6 Survey Results

3.6.1 Pre-survey results

Before starting the project a pre-survey was done to understand the requirements for the application and understand the market value for this application.

Some of the results of survey are as follows:

How do you prefer to hire your vehicle online?



40 responses

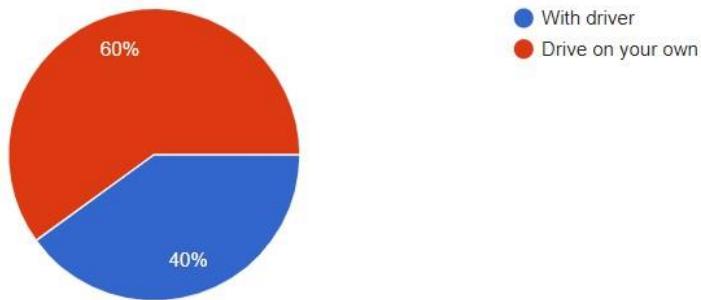


Figure 90: Pre-survey result 1

Have you heard about online vehicle renting application?

40 responses

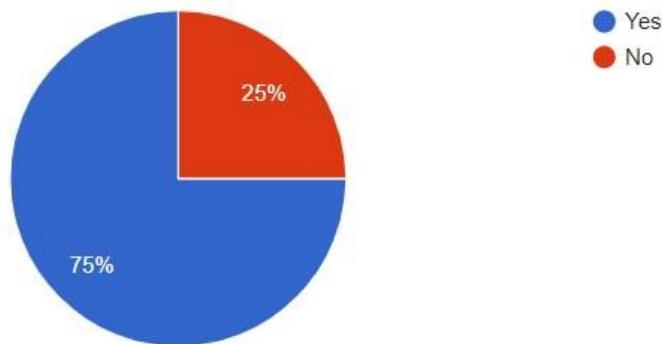


Figure 91: Pre-survey result 2

[Click here to view more results of Pre-Survey](#)

3.6.2 Post-survey results

After pre-survey was conducted successfully post-survey was also conducted to learn more about people reviews and their need. This survey was done to know what kind of features they want in an application.

Some of the results of survey are as follows:

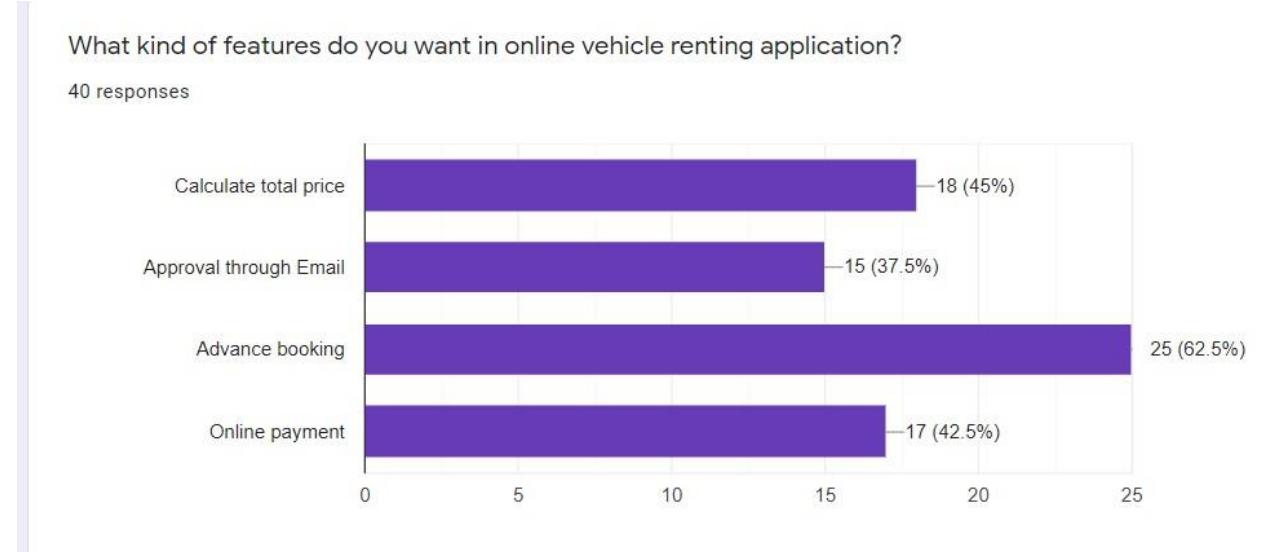


Figure 92: Post-survey result -1

Any other features you would like to add to this application?

22 responses

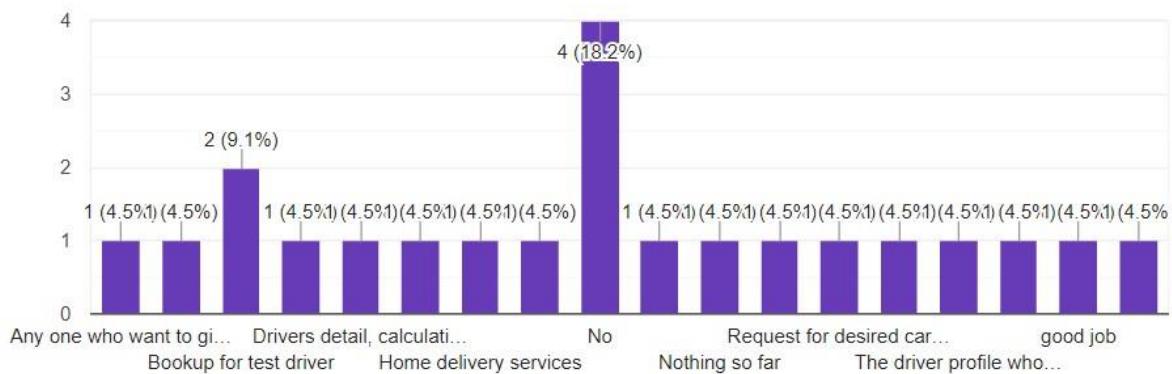


Figure 93: Post-survey result -2

[Click here to view more results of Post-Survey](#)

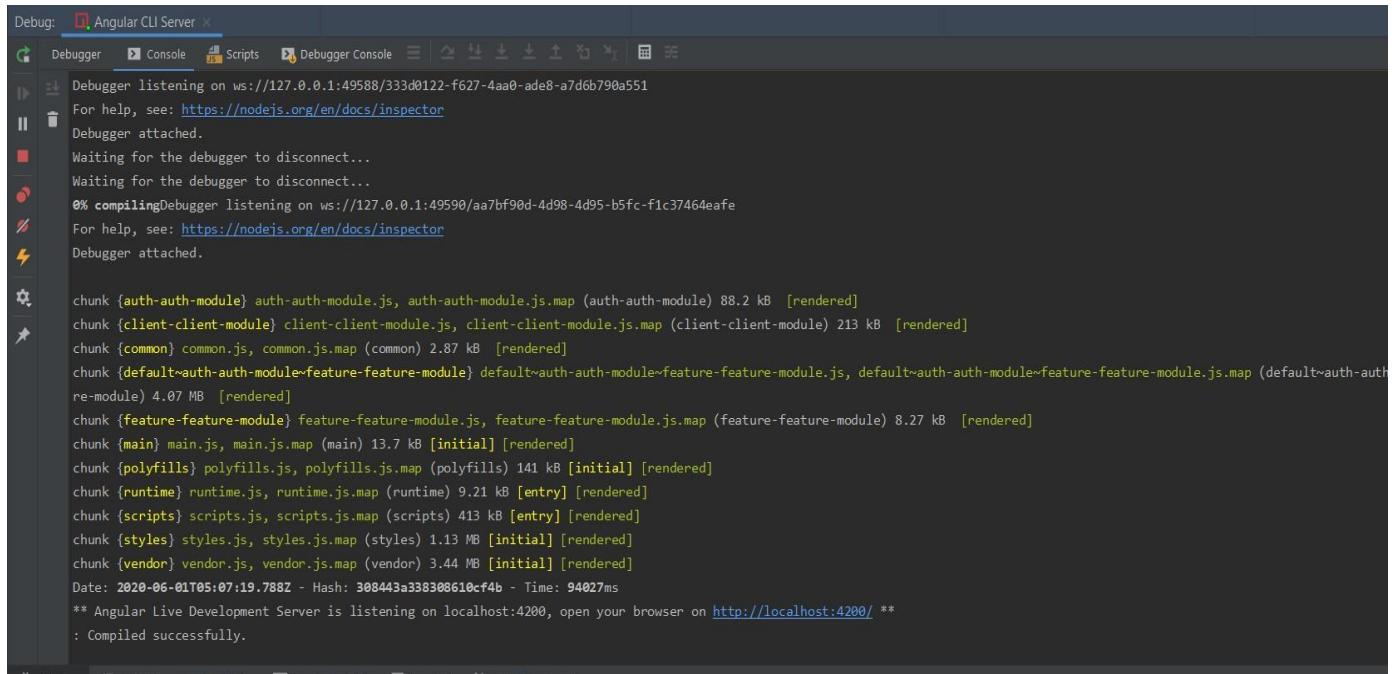
Chapter 4: Testing and Analysis of progress

4.1 Iteration 1 Testing

Frontend Setup

Objective	Compile project
Action	Run frontend project
Expected Result	Frontend project will compile and run successfully.
Actual Result	Frontend project compiled and run successfully.
Conclusion	Test Successful

Table 2: Frontend compile testing



```

Debug: Angular CLI Server
Debugger Console Scripts Debugger Console
Debugger listening on ws://127.0.0.1:49588/333d0122-f627-4aa0-ade8-a7d6b790a551
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Waiting for the debugger to disconnect...
Waiting for the debugger to disconnect...
0% compilingDebugger listening on ws://127.0.0.1:49590/aa7bf90d-4d98-4d95-b5fc-f1c37464eafe
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.

chunk {auth-auth-module} auth-auth-module.js, auth-auth-module.js.map (auth-auth-module) 88.2 kB [rendered]
chunk {client-client-module} client-client-module.js, client-client-module.js.map (client-client-module) 213 kB [rendered]
chunk {common} common.js, common.js.map (common) 2.87 kB [rendered]
chunk {default~auth-auth-module~feature-feature-module} default~auth-auth-module~feature-feature-module.js, default~auth-auth-module~feature-feature-module.js.map (default~auth-auth-module) 4.07 MB [rendered]
chunk {feature-feature-module} feature-feature-module.js, feature-feature-module.js.map (feature-feature-module) 8.27 kB [rendered]
chunk {main} main.js, main.js.map (main) 13.7 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 9.21 kB [entry] [rendered]
chunk {scripts} scripts.js, scripts.js.map (scripts) 413 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 1.13 MB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3.44 MB [initial] [rendered]
Date: 2020-06-01T05:07:19.788Z - Hash: 308443a338308610cf4b - Time: 94027ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.

```

Figure 94: Frontend project setup

Backend Setup

Objective	Compile project
Action	Run backend project
Expected Result	Backend project will compile and run successfully.
Actual Result	Backend project compiled and run successfully.
Conclusion	Test Successful

Table 3: Backend compile testing

```
Debug: SpringBootAuthApplication
Debugger Console Endpoints
[|] [x] [!]
[!]
:: Spring Boot :: (v2.2.5.RELEASE)

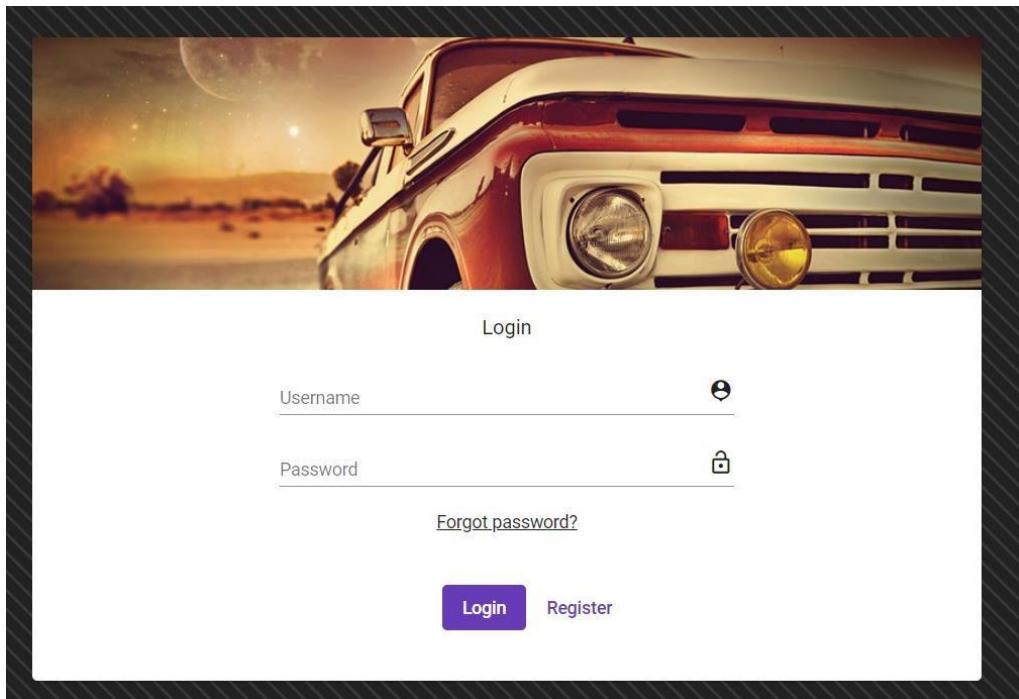
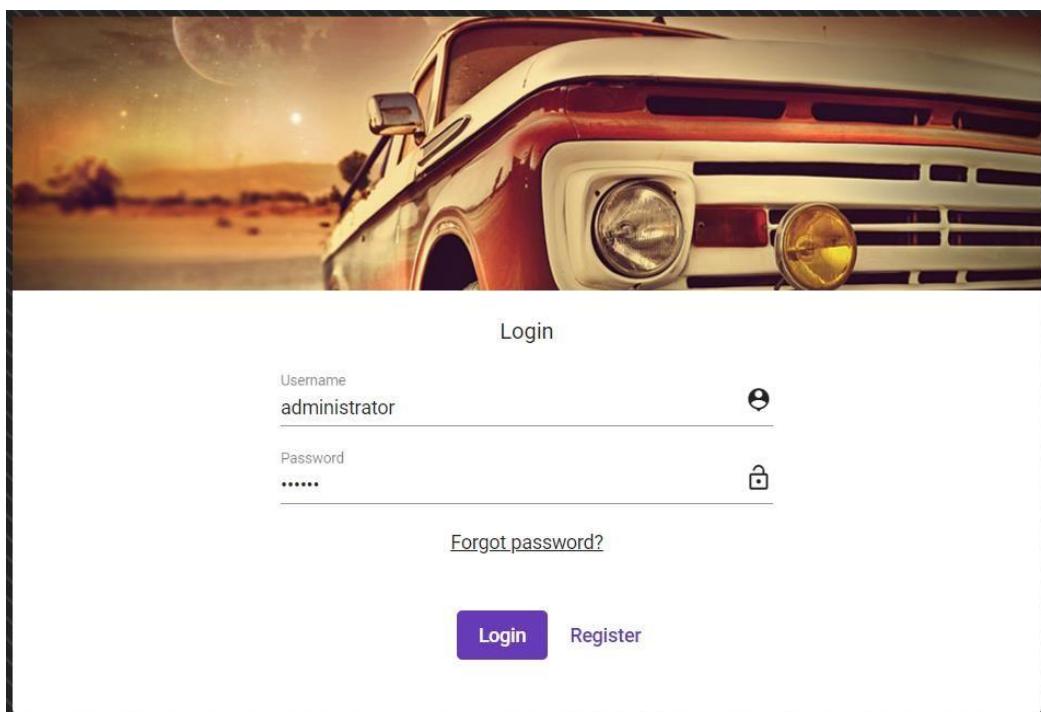
2020-06-01 12:56:19.807 INFO 13428 --- [main] c.g.s.s.SpringBootAuthApplication : Starting SpringBootAuthApplication on DESKTOP-T34HJCB with PID 13428 (started by ACER in D:\FYP)
2020-06-01 12:56:19.815 INFO 13428 --- [main] c.g.s.s.SpringBootAuthApplication : No active profile set, falling back to default profiles: default
2020-06-01 12:56:20.966 INFO 13428 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2020-06-01 12:56:21.076 INFO 13428 --- [main] s.d.r.c.RepositoryConfigurationDelegate : finished Spring Data repository scanning in 97ms. Found 4 JPA repository interfaces.
2020-06-01 12:56:21.310 INFO 13428 --- [main] o.s.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8081 (http)
2020-06-01 12:56:22.328 INFO 13428 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2020-06-01 12:56:22.329 INFO 13428 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.31]
2020-06-01 12:56:22.501 INFO 13428 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2020-06-01 12:56:22.502 INFO 13428 --- [main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 2544 ms
2020-06-01 12:56:22.799 INFO 13428 --- [main] o.h.license.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2020-06-01 12:56:22.930 INFO 13428 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.4.12.Final
2020-06-01 12:56:23.356 INFO 13428 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations (5.1.0.Final)
2020-06-01 12:56:23.582 INFO 13428 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2020-06-01 12:56:23.817 INFO 13428 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2020-06-01 12:56:23.860 INFO 13428 --- [main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQL55Dialect
2020-06-01 12:56:25.625 INFO 13428 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH0000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformImpl]
2020-06-01 12:56:25.637 INFO 13428 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized PTA EntityManagerFactory for persistence unit 'default'
2020-06-01 12:56:26.513 WARN 13428 --- [main] jpaBaseConfiguration3PawWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during a flush.
2020-06-01 12:56:26.785 DEBUG 13428 --- [main] org.hibernate.SQL : select user0_.id as id1_3_, user0_.created as created2_3_, user0_.modified_by_id as modified3_3_, user0_.contact as contact4_3_, user0_.dob as dob5_3_, user0_.email as email6_3_, user0_
Hibernate: select user0_.id as id1_3_, user0_.created as created2_3_, user0_.modified_by_id as modified3_3_, user0_.contact as contact4_3_, user0_.dob as dob5_3_, user0_.email as email6_3_, user0_
2020-06-01 12:56:27.342 INFO 13428 --- [main] org.springframework.context.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2020-06-01 12:56:28.314 INFO 13428 --- [main] o.s.s.web.DefaultSecurityFilterChain : Creating filter chain: OrgRequestMatcher@{Ant [patterns='/oauth/*]}, Ant [pathPatterns='/*'], [filters=DelegatingFilterProxy@53333333], [dispatcherNames=REQUEST, FORWARD, INCLUDE, ERROR], [order=1]
2020-06-01 12:56:28.336 INFO 13428 --- [main] o.s.s.web.DefaultSecurityFilterChain : Creating filter chain: org.springframework.security.oauth2.config.annotation.web.configuration.OAuth2WebSecurityFilterChain@53333333, [filters=DelegatingFilterProxy@53333333], [dispatcherNames=REQUEST, FORWARD, INCLUDE, ERROR], [order=2]
2020-06-01 12:56:28.345 INFO 13428 --- [main] o.s.s.web.DefaultSecurityFilterChain : Creating filter chain: any request, [org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFilter@53333333], [filters=DelegatingFilterProxy@53333333], [dispatcherNames=REQUEST, FORWARD, INCLUDE, ERROR], [order=3]
2020-06-01 12:56:28.456 INFO 13428 --- [main] o.s.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8081 (http) with context path ''
2020-06-01 12:56:28.461 INFO 13428 --- [main] c.g.s.s.SpringBootAuthApplication : Started SpringBootAuthApplication in 9.732 seconds (JVM running for 11.642)
```

Figure 95: Backend project setup

4.2 Iteration 2 Testing

Login

Objective	Login with admin credentials
Action	Enter admin login credentials
Expected Result	Login successfully and redirect to admin dashboard.
Actual Result	Logged in successfully and redirected to admin dashboard.
Conclusion	Test Successful

Table 4: Test case for Admin login*Figure 96: Admin login UI**Figure 97: Admin credentials*

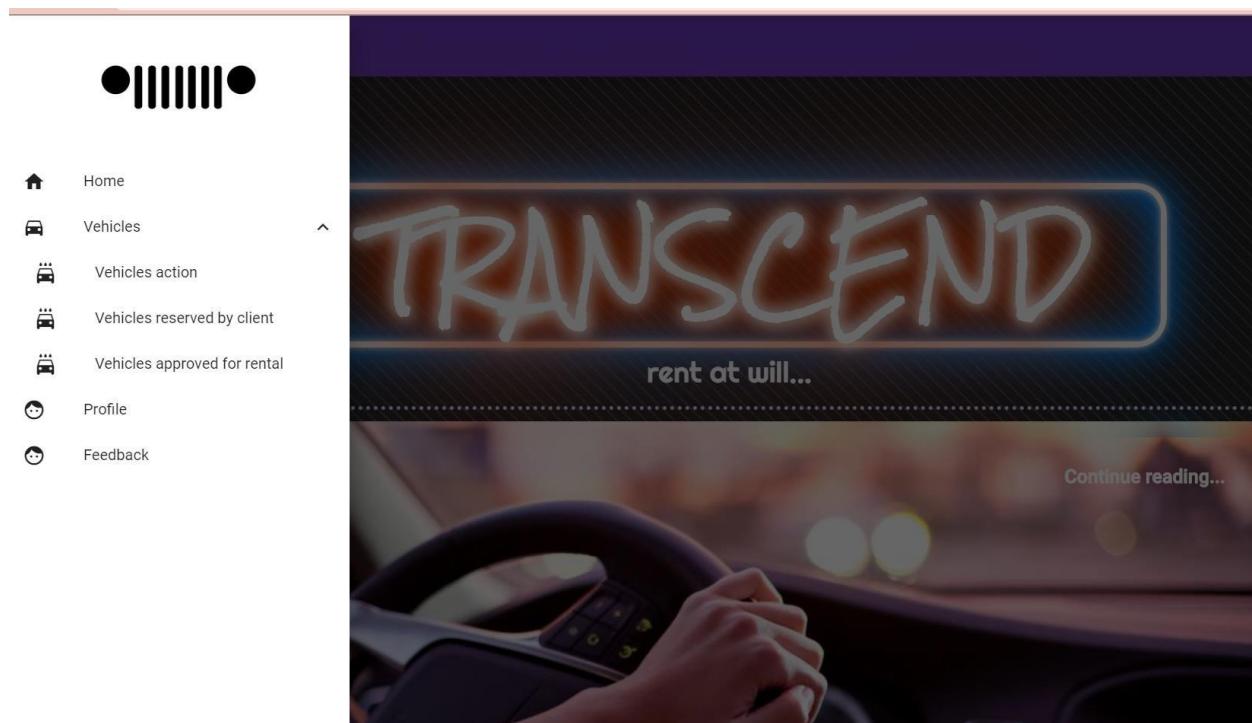


Figure 98: Redirect to admin dashboard

Change password

Objective	Change password for Admin
Action	Click profile and enter new password.
Expected Result	Password will change and credentials will be updated redirecting to login page.
Actual Result	Password changed with updated credentials and redirected to login page.
Conclusion	Test Successful

Table 5: Test case for password change

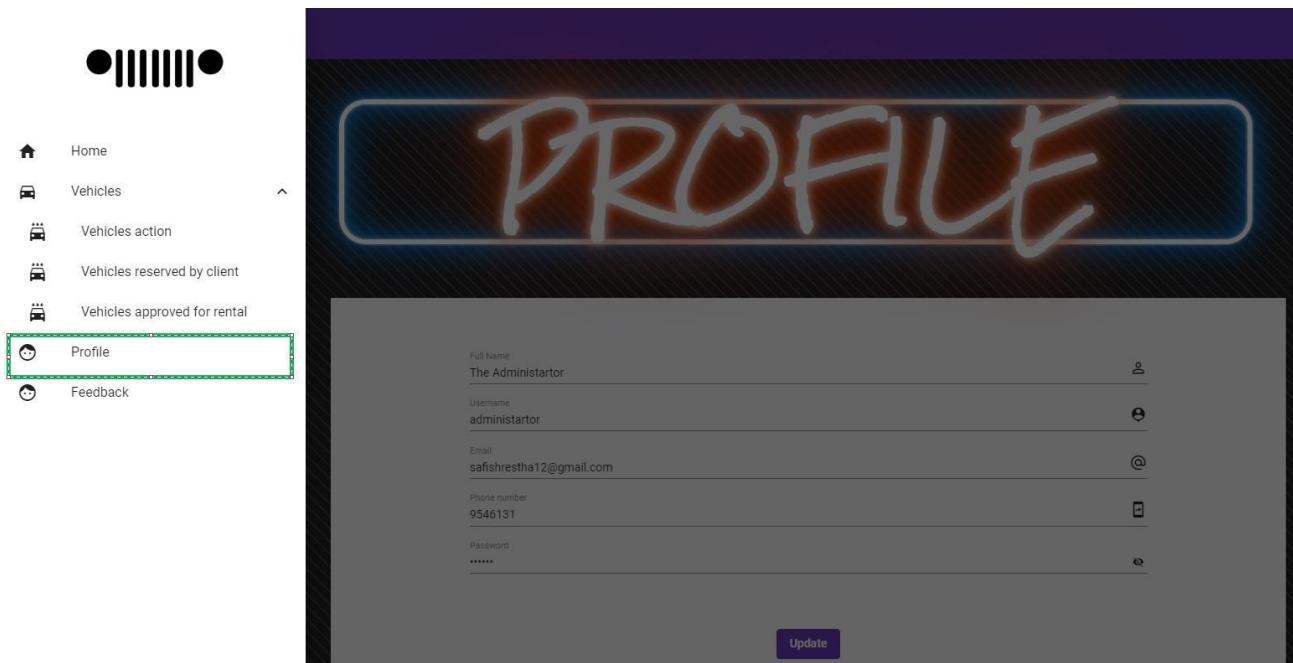


Figure 99: Click 'Profile'

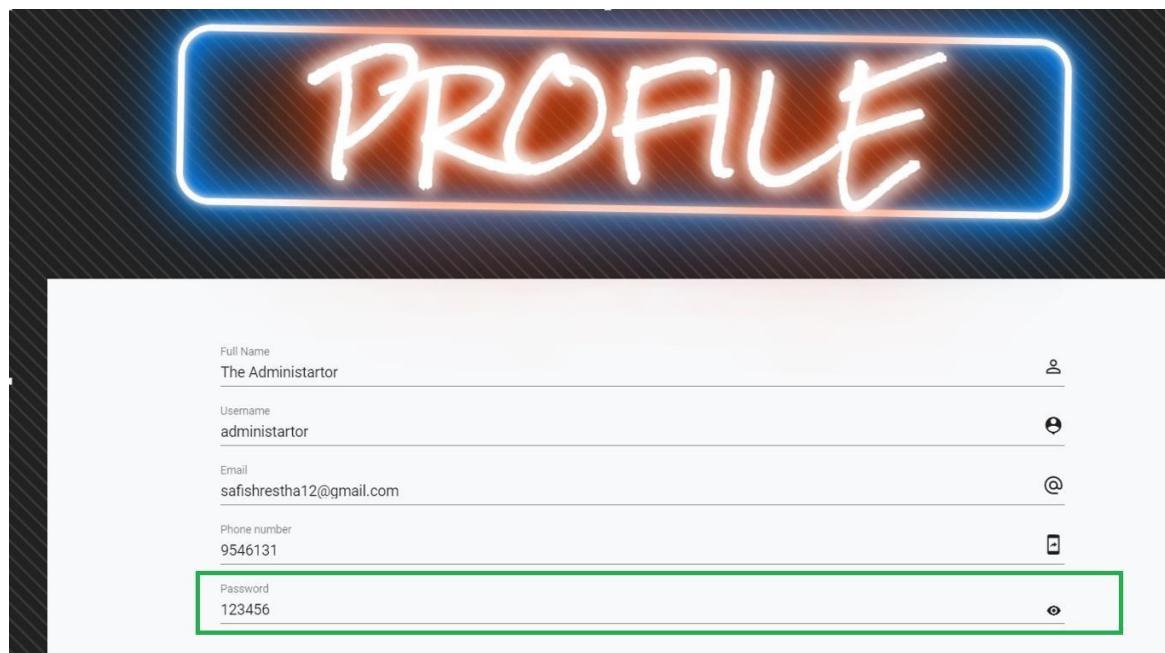


Figure 100 : Change admin password

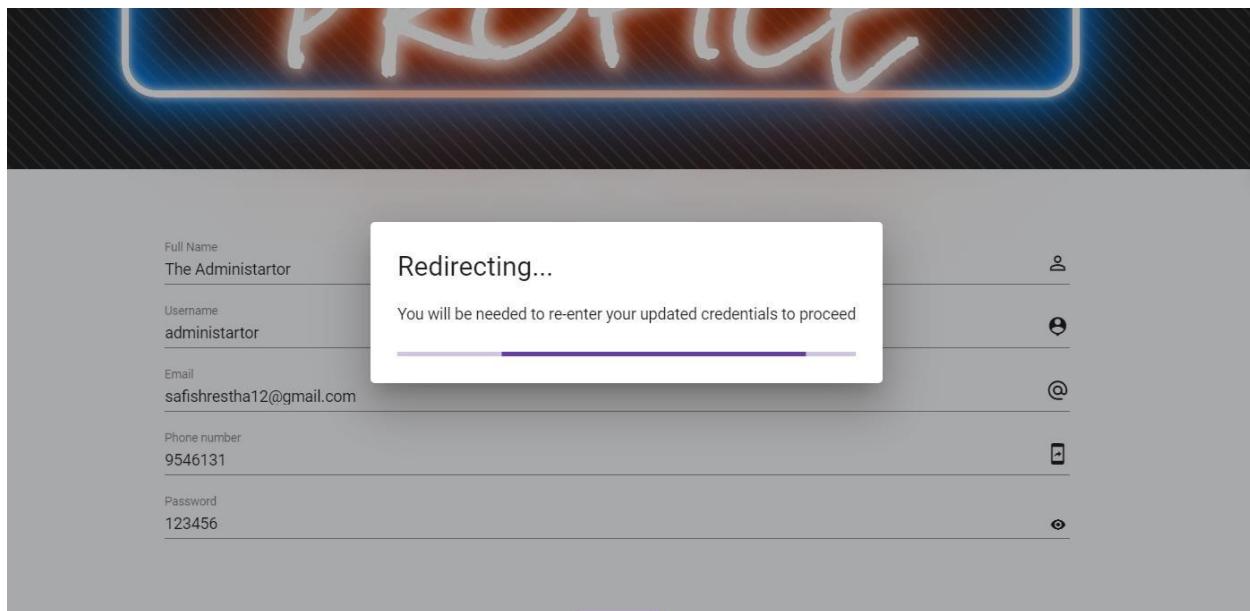


Figure 101: Display updating credentials



Figure 102: Successful login after changing password

Add Vehicles

Objective	To add new vehicles
Action	Fill the details of vehicles to add it.
Expected Result	Vehicles will be added after add button is clicked.
Actual Result	Vehicles were added after add button was clicked.
Conclusion	Test Successful

Table 6 : Test case for adding vehicles by admin

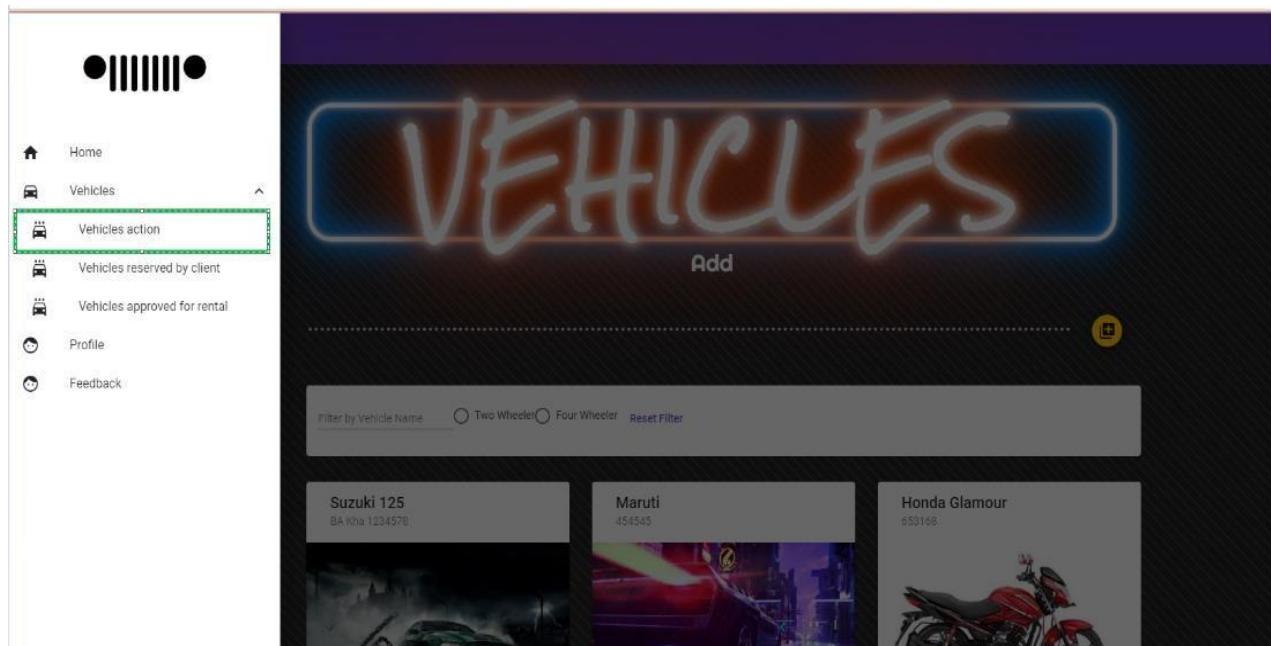


Figure 103: Click vehicle action

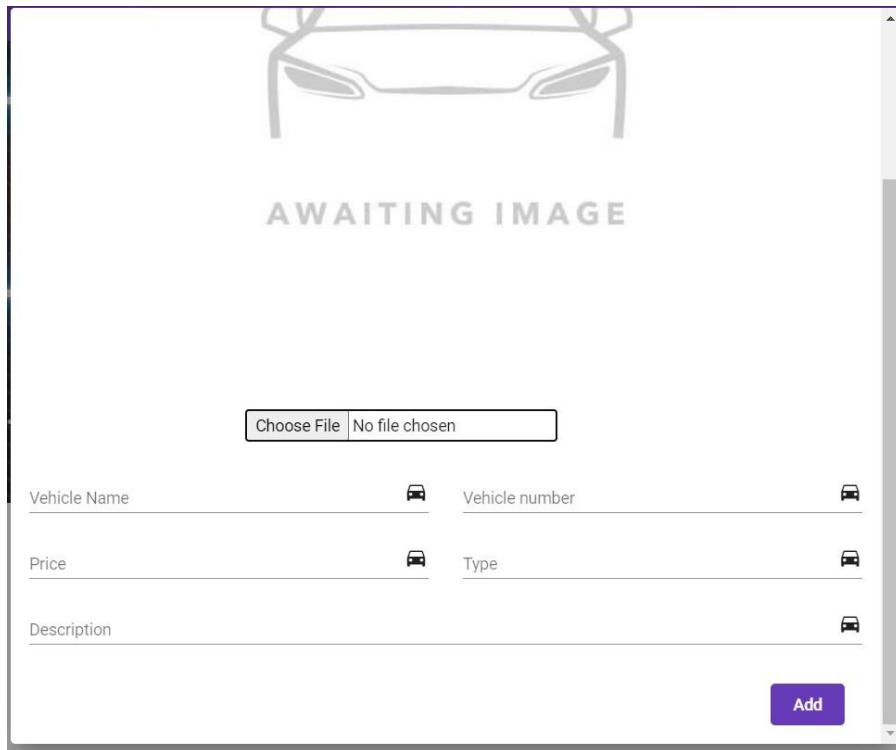


Figure 104 : Vehicle adding form

The screenshot shows the same vehicle addition form as Figure 104, but with a different background image: a green Ford Mustang driving through a dark, smoky industrial landscape with lightning in the background. The form fields are populated with data: "Vehicle Name" is "Suzuki", "Vehicle number" is "BA Kha 12345", "Price" is "300", "Type" is "Four wheeler", and "Description" is "This is suzuk!". The "Choose File" input field shows the path "9-Ford-Mustan...allpaper.jpg". The "Add" button is visible at the bottom right.

Figure 105: Adding vehicle details

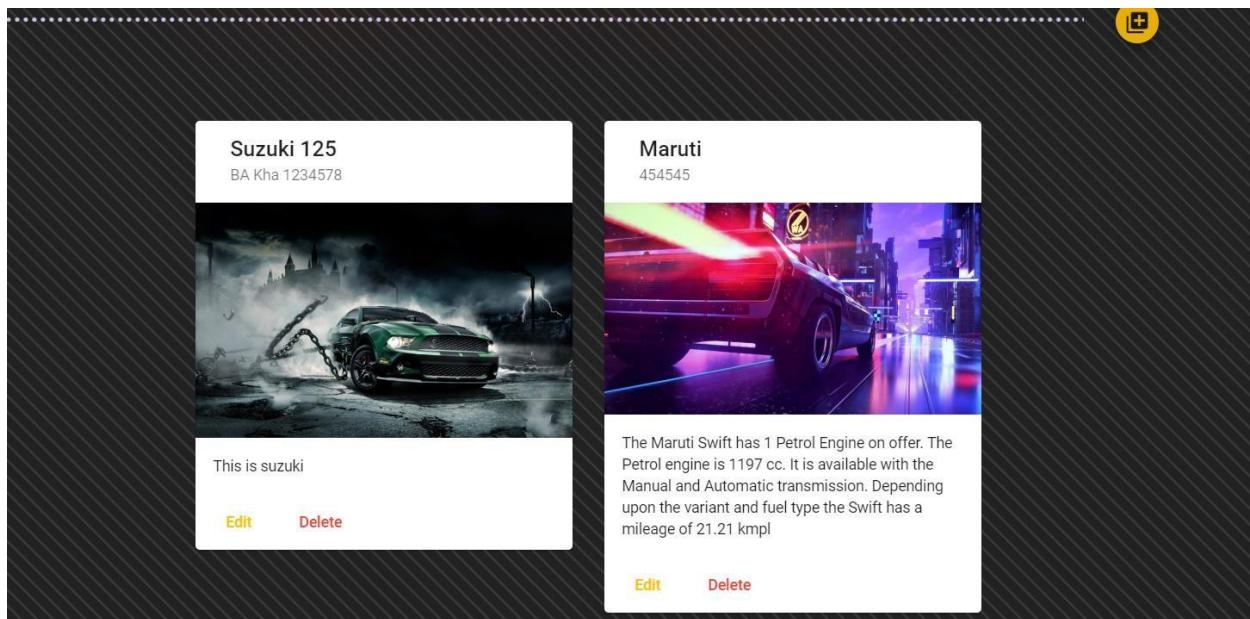


Figure 106: Displaying added vehicles

Update Vehicles

Objective	To update vehicles details
Action	Click on Edit button to update vehicles.
Expected Result	Vehicle details will be updated after update button is clicked.
Actual Result	Vehicle details were updated after update button was clicked.
Conclusion	Test Successful

Table 7: Test case for updating vehicles

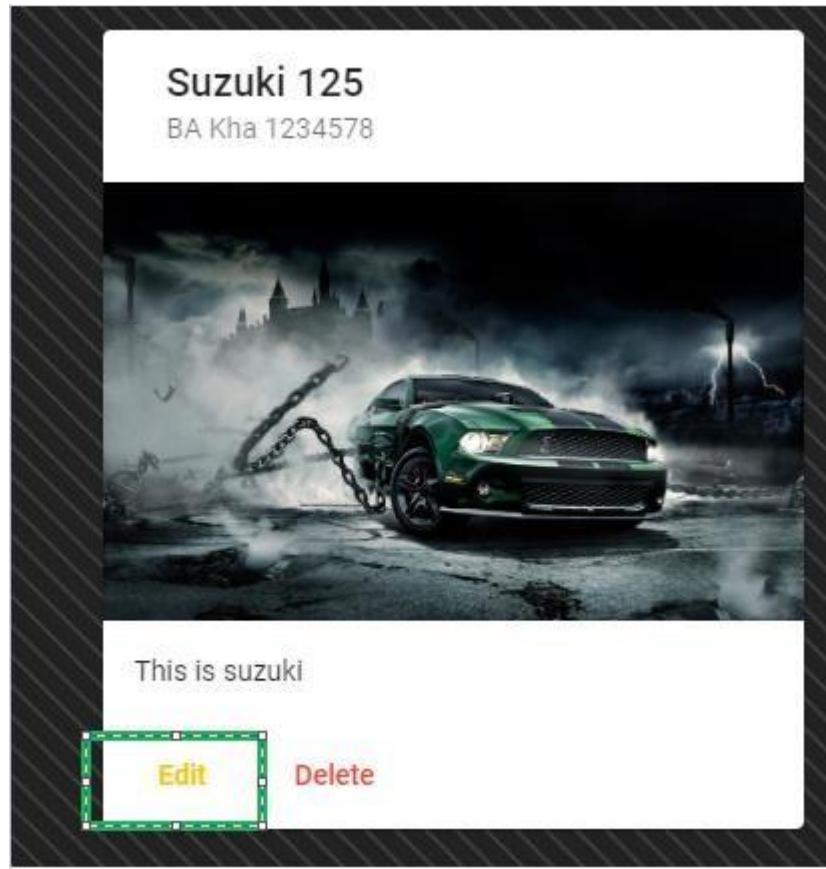


Figure 107: Editing vehicles by clicking edit button



Choose File No file chosen

Vehicle Name Suzuki 125	Vehicle number BA Kha 1234578
Price 300	Type Four wheeler
Description This is suzuki	

Update

Figure 108: Editing highlighted text field

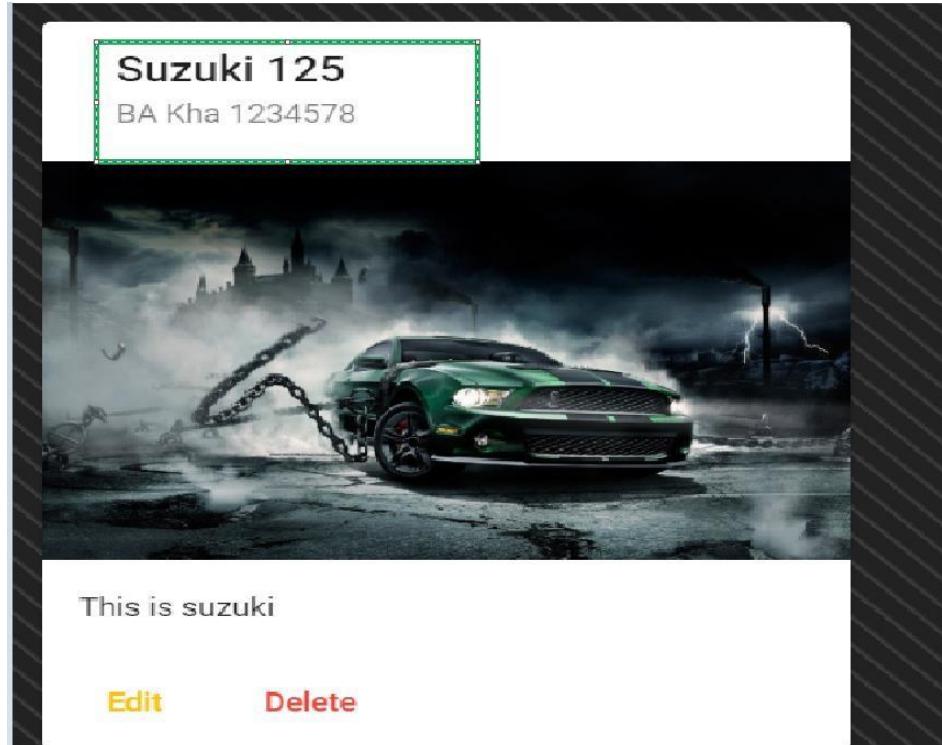


Figure 109: After updating vehicle

Delete Vehicles

Objective	To delete vehicles
Action	Click on Delete button to delete vehicles.
Expected Result	Vehicles will be deleted after delete button is clicked.
Actual Result	Vehicles were deleted after deleted button was clicked.
Conclusion	Test Successful

Table 8: Test case for deleting vehicles

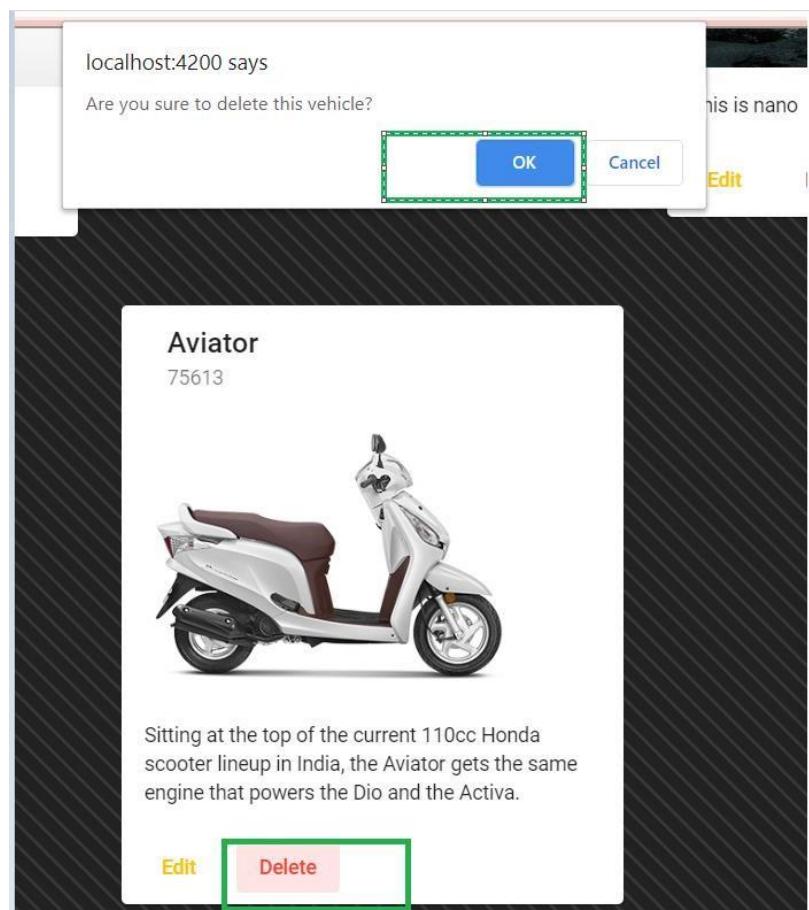


Figure 110 : Deleting vehicles by clicking delete button

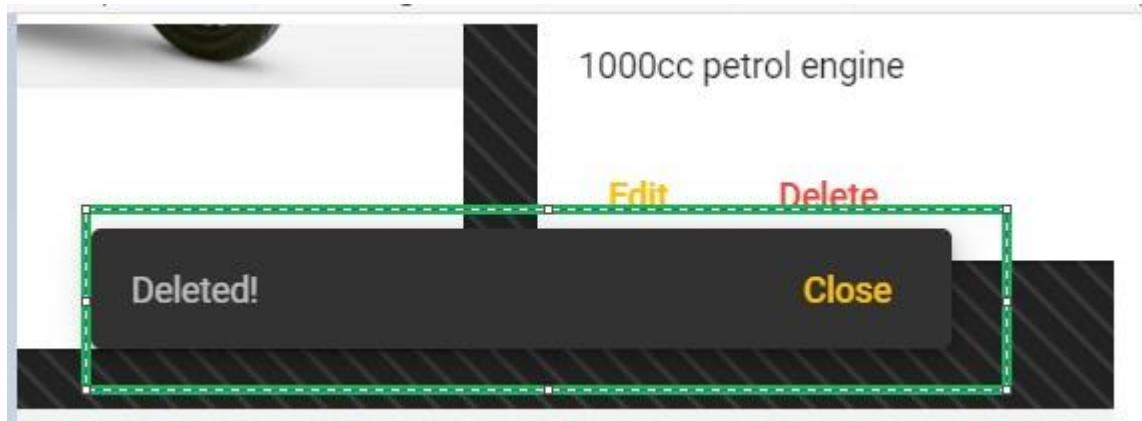


Figure 111: Displaying vehicle deleted message

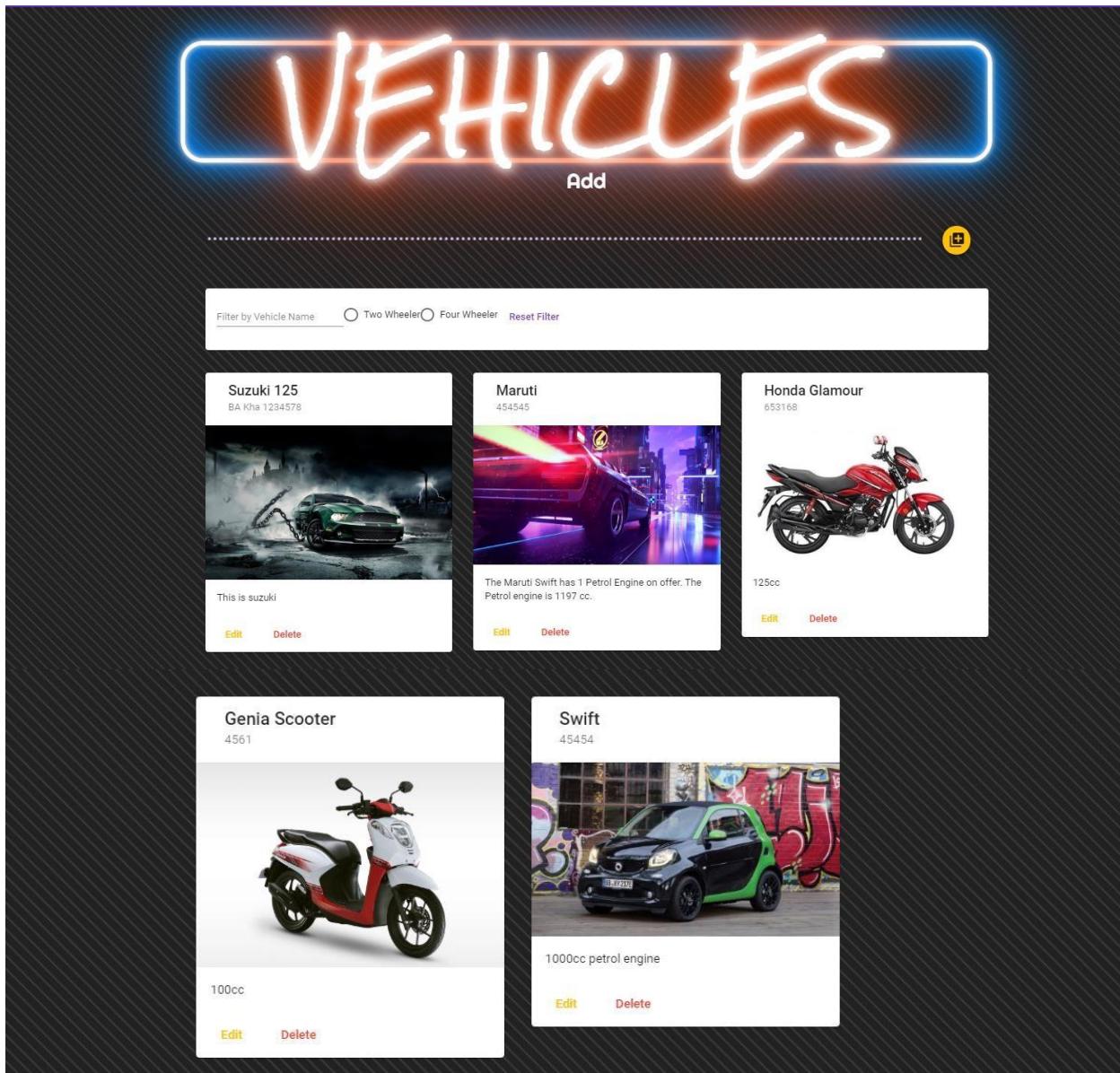


Figure 112: List of vehicles after deleting one vehicle

Admin Logout

Objective	To logout from admin panel and redirect to login page after clicking 'Log out' button
Action	Click on Log Out button
Expected Result	Application will log out and redirect to login page.
Actual Result	Application redirects to login page after logout.

Conclusion	Test Successful
-------------------	-----------------

Table 9: Test case for Admin logout

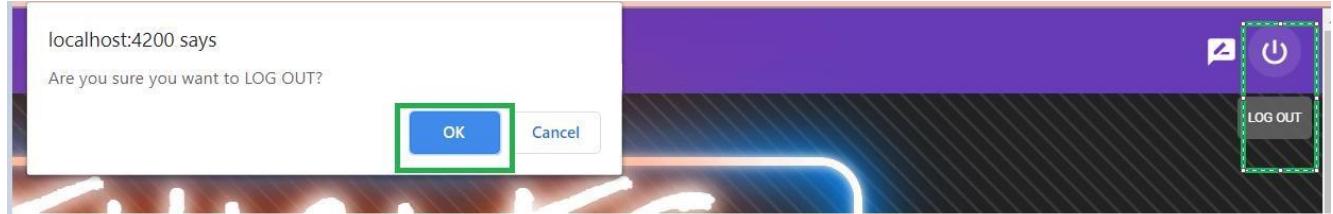


Figure 113: Logging out from admin panel

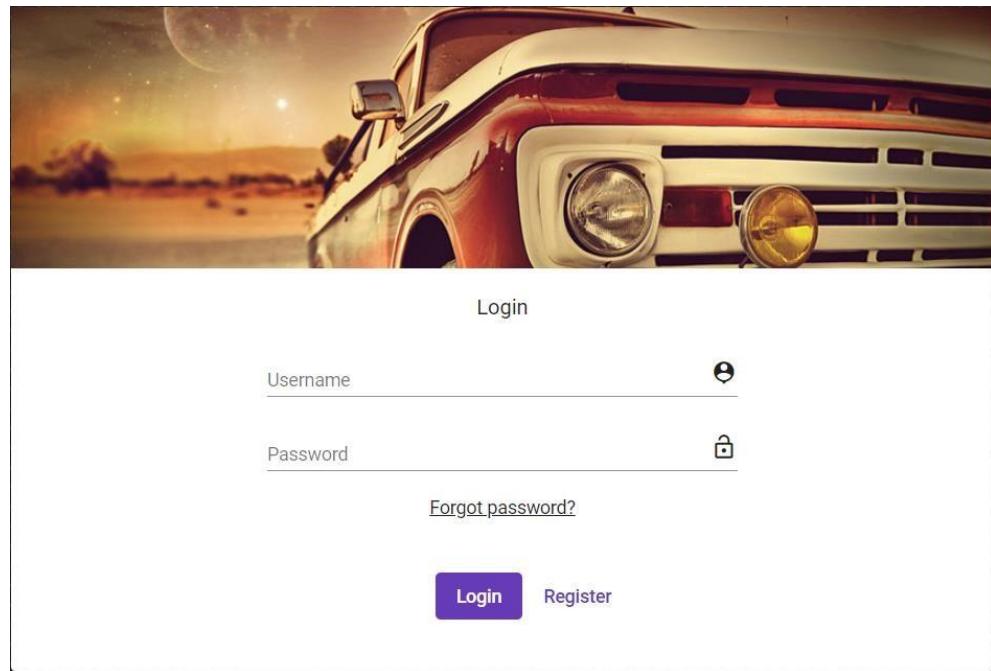


Figure 114: Redirecting to login page after log out

Forgot Password

Objective	To test if mail is sent to user for resetting password
Action	Click on Forget Password button and send email.
Expected Result	Application will send email with OTP to user with valid email and user will be able to reset password.
Actual Result	Application sent email with OTP to user with valid email and user will be able to reset password.

Conclusion	Test Successful
------------	-----------------

Table 10: Test case for Forget password

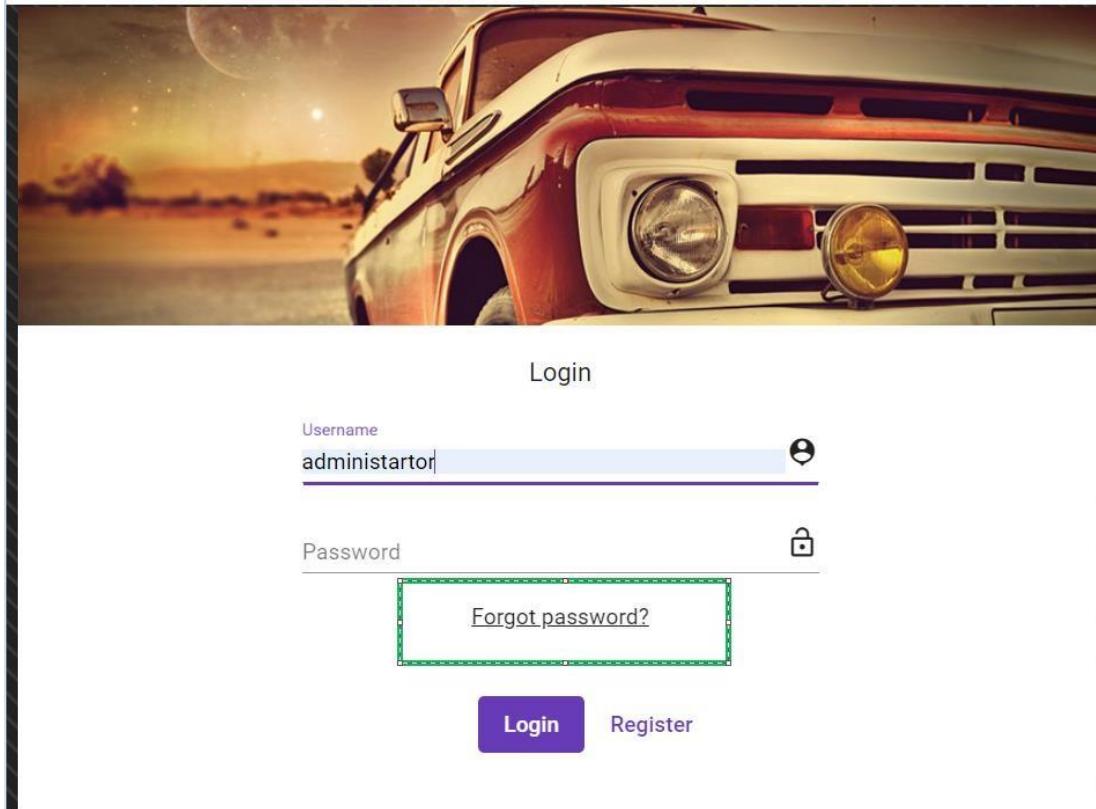


Figure 115: Click on 'Forgot Password?'

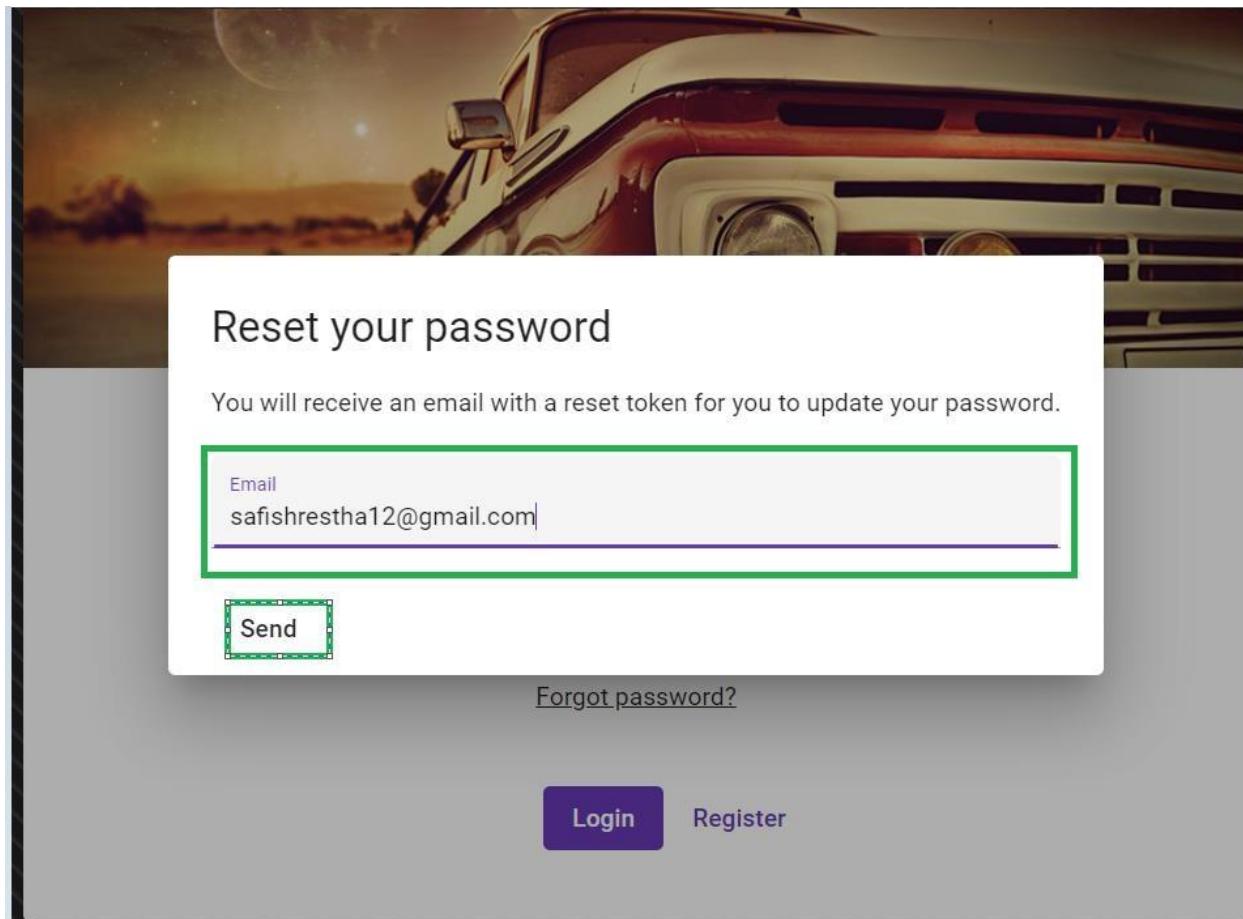


Figure 116: Enter Email on the given field

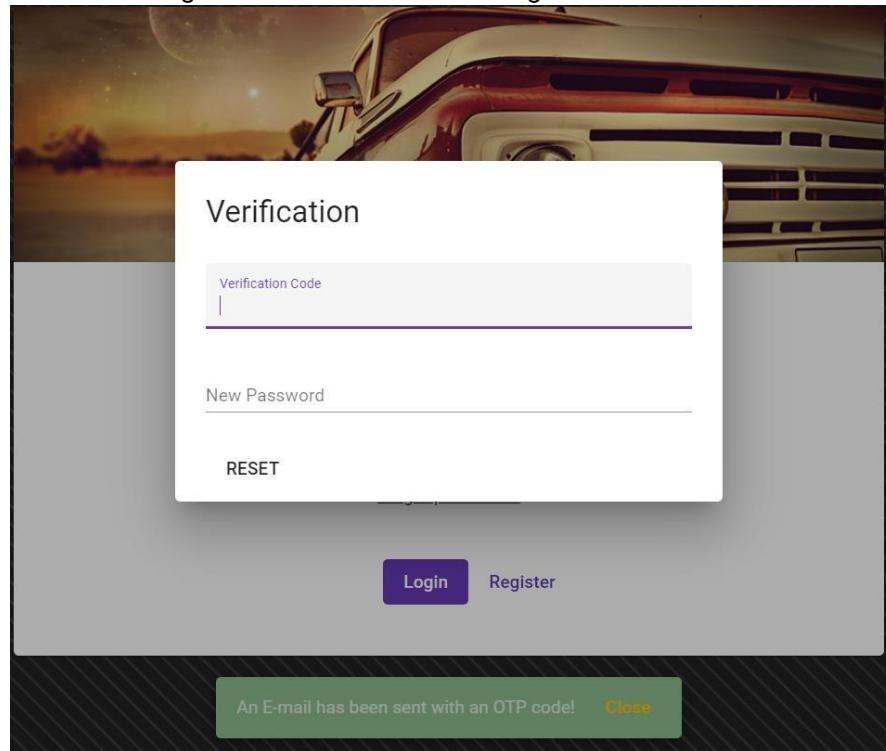


Figure 117: Email sent with OTP code success message

One Time Password

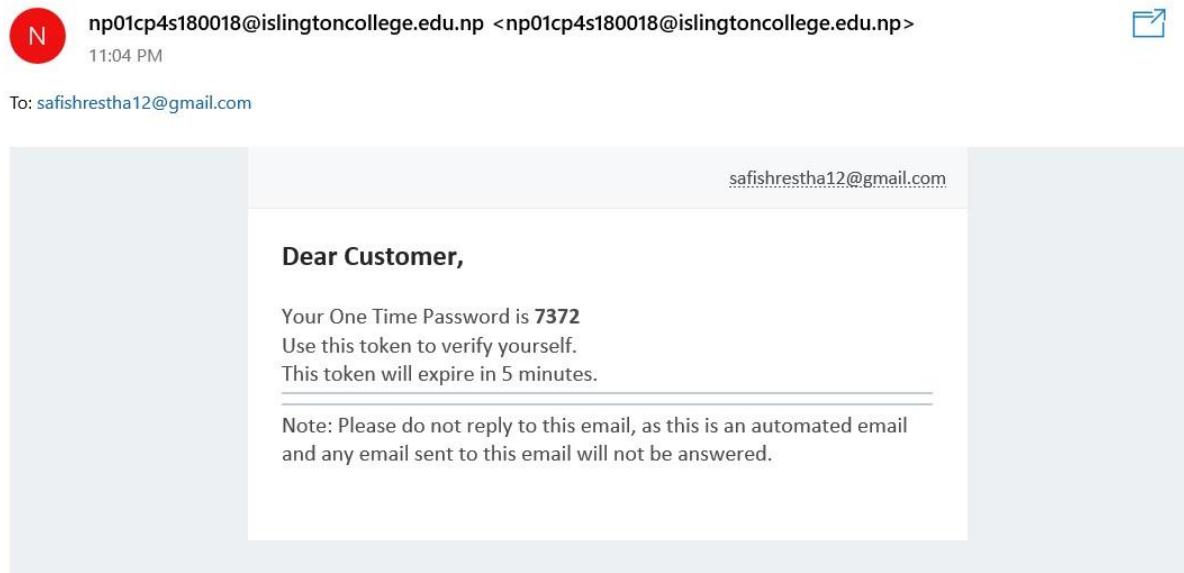


Figure 118: OTP sent via email to reset password

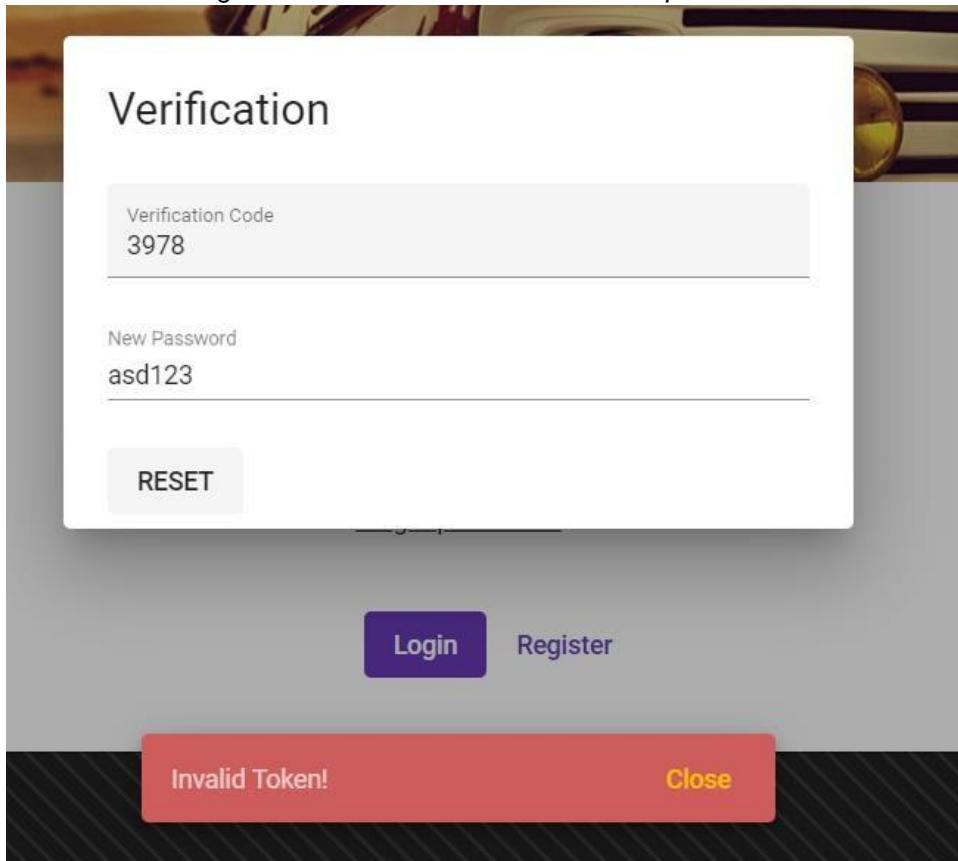


Figure 119: Message displayed in case of wrong token entered.

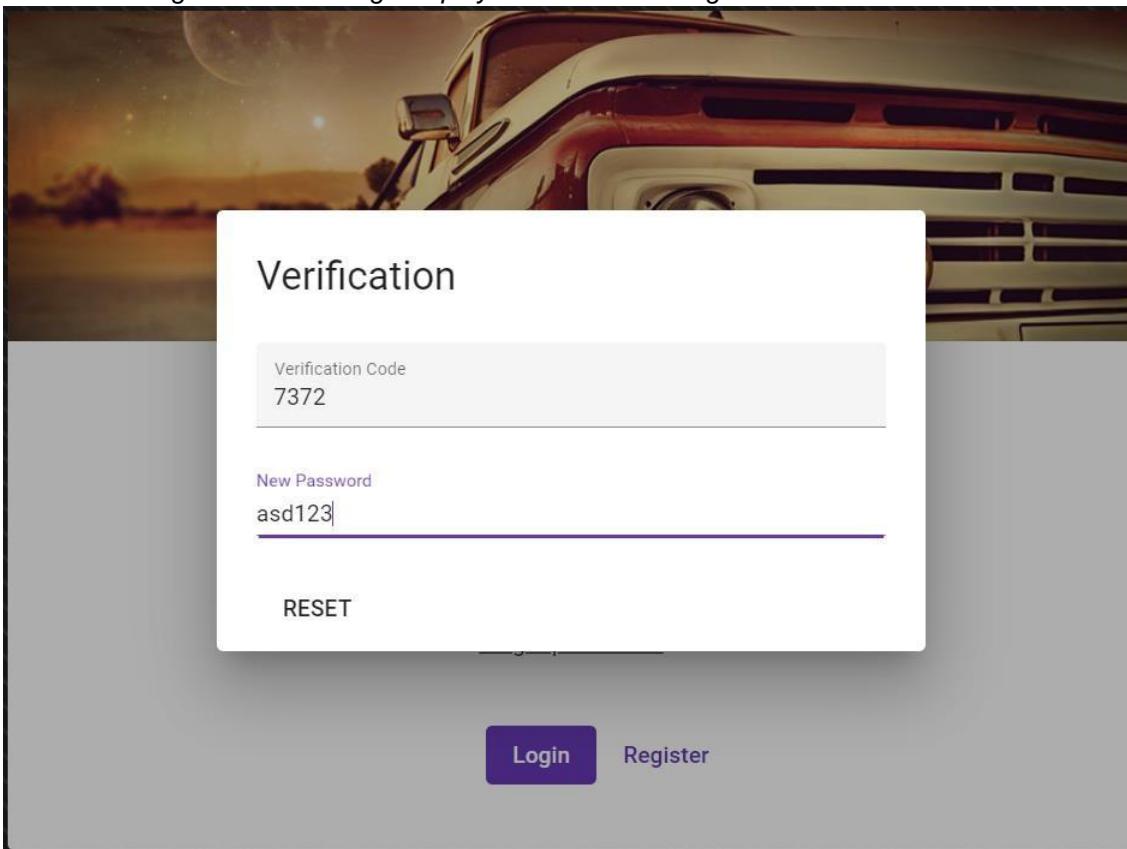


Figure 120: Resetting password with valid OTP

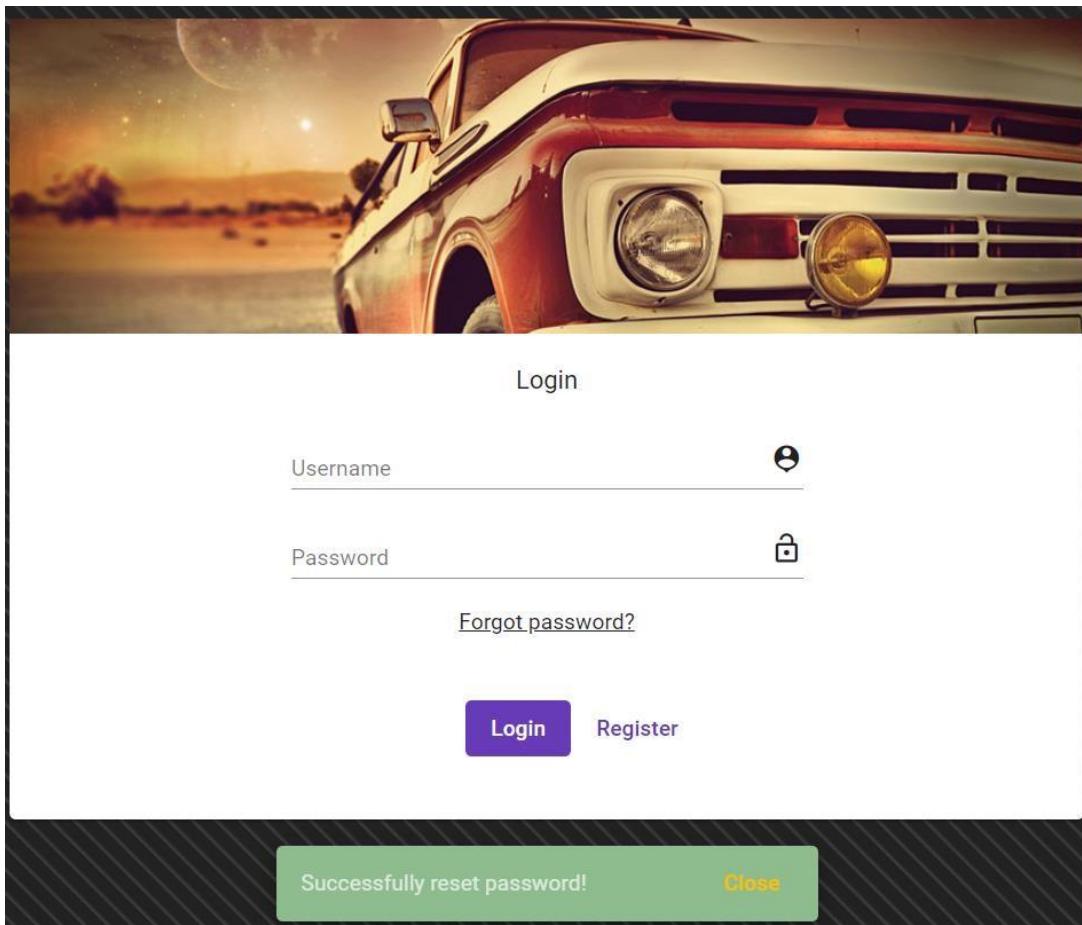


Figure 121: Reset password success message

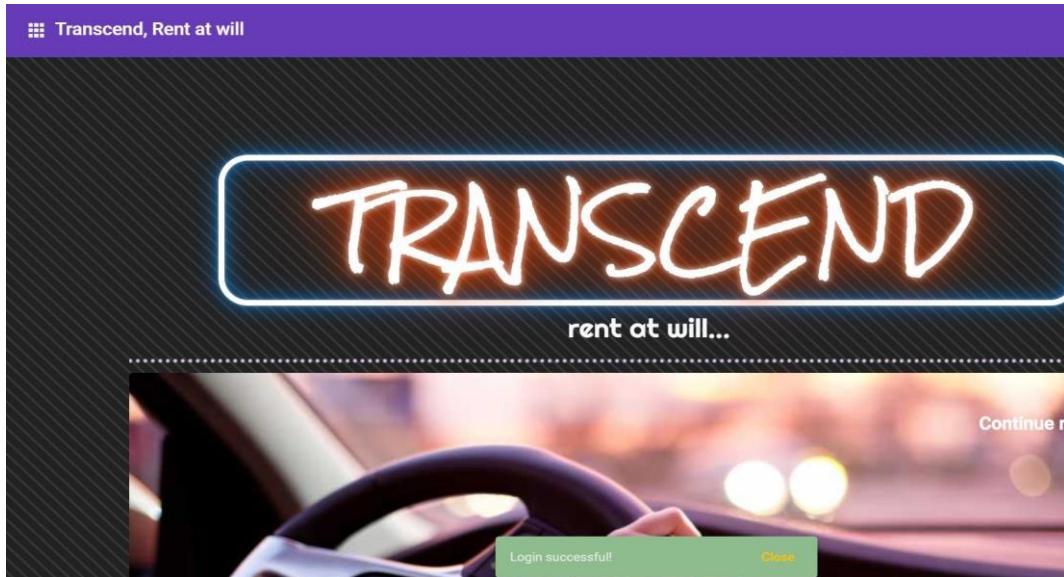


Figure 122: Login success after resetting password

Bad credentials

Objective	To test invalid username or password.
Action	Click on Login button.
Expected Result	Application will display message in case of invalid username or password entered.
Actual Result	Application displayed message in case of invalid username or password entered.
Conclusion	Test Successful

Table 11: Test case for invalid username or password

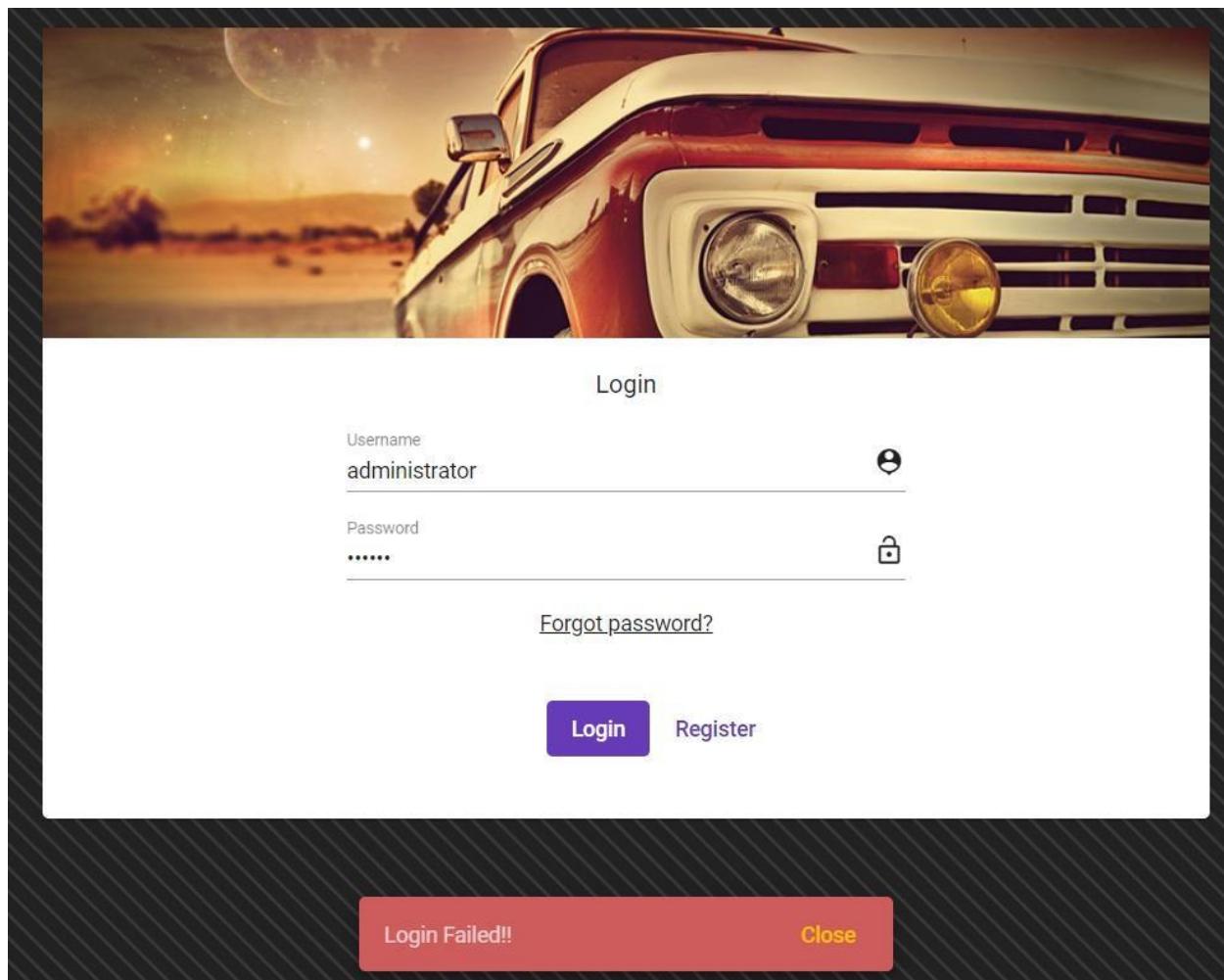
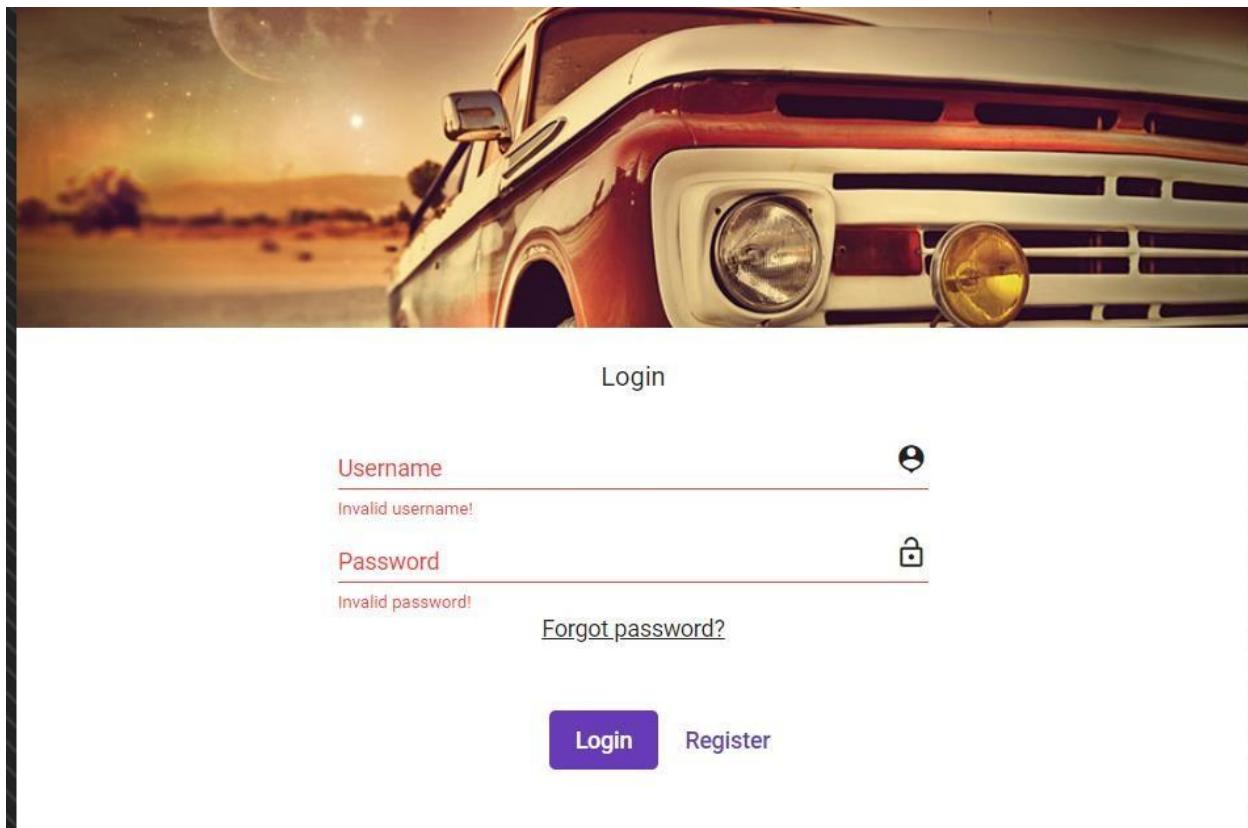


Figure 123: Displaying message when login with invalid username and password

Empty login fields

Objective	To test whether admin can login when text fields are empty.
Action	Click on Login button.
Expected Result	Application will display message in case of invalid username or password entered.
Actual Result	Application displayed message in case of invalid username or password entered.
Conclusion	Test Successful

Table 12: Test case for empty login fields



Figure

124: Displaying error message for empty text field login

Empty Description

Objective	To test whether admin can add vehicles when text fields are empty.
Action	Click on Add button.

Expected Result	Application will not add vehicles and show missing fields with red text fields.
Actual Result	Application did not add vehicles and shows missing fields with red text fields.
Conclusion	Test Successful

Table 13: Test case for empty fields while adding vehicle

The screenshot shows a web-based application interface for adding a vehicle. At the top, there is a file upload input labeled "Choose File" with the path "Hcc9c902d67...G.jpg_.webp". Below this, there are several input fields:

- "Vehicle Name": "Velvo scooter" (with a small car icon to its right)
- "Vehicle number": "454545" (with a small car icon to its right)
- "Price": An empty field highlighted with a green border and a red underline, indicating it is required.
- "Type": "Two wheeler" (with a small car icon to its right)
- "Description": A long empty text area highlighted with a dashed green border, indicating it is required.

At the bottom right of the form is a purple "Add" button.

Figure 125: Displaying error for empty text field while adding vehicle

4.3 Iteration 3 Testing

Register customer

Objective	To register new user.
Action	Click on Register button.
Expected Result	Application will verify email and register user.

Actual Result	Application verified email and registered user.
Conclusion	Test Successful

Table 14: Test case for user registration

Client

As a Client...

Full Name
Georgie Ruiz

Username
George

Email
collyria@qqfullbet.club

Phone number
9815234678

Date of Birth
7/9/1997

Password

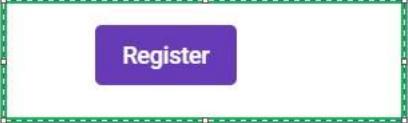


Figure 126: User registration form

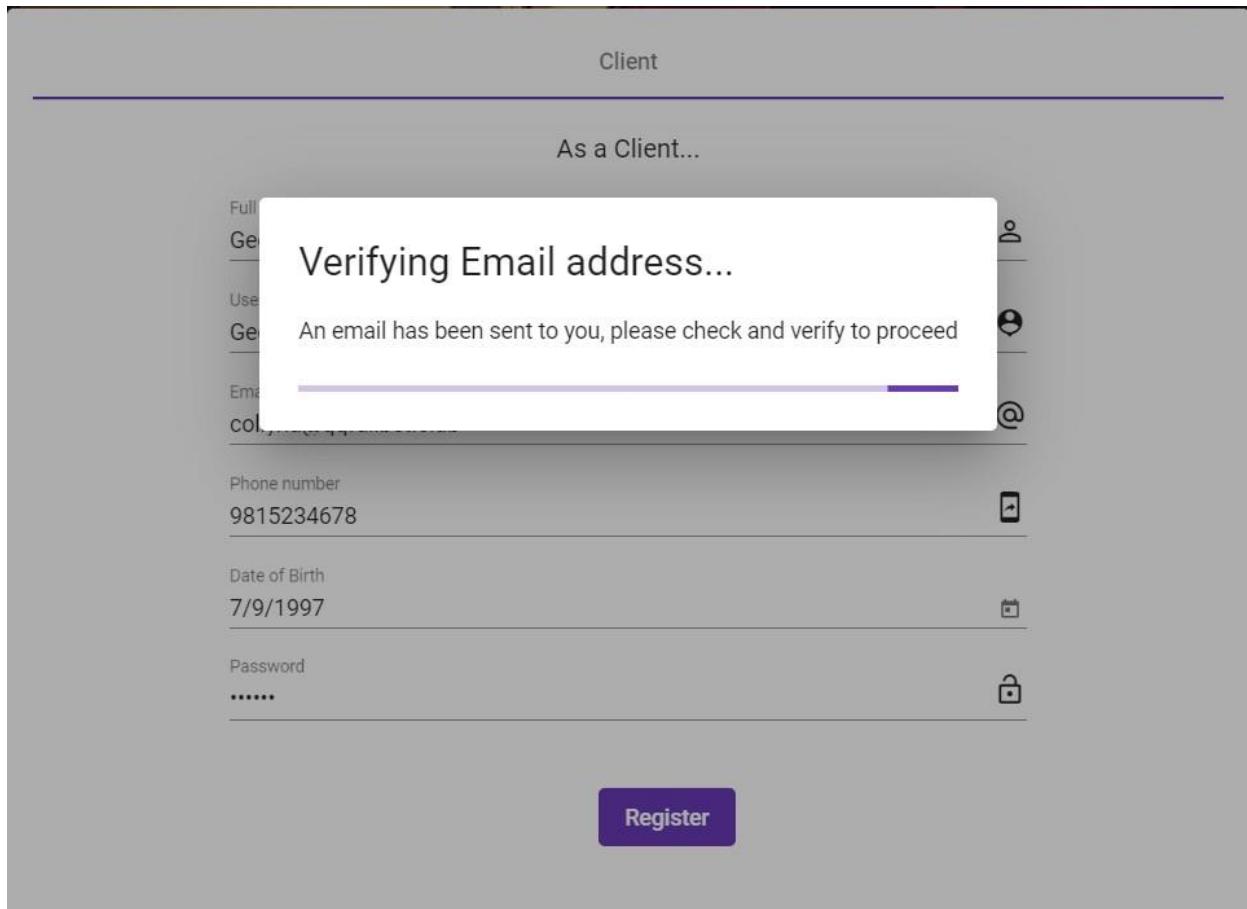


Figure 127: Verifying email address

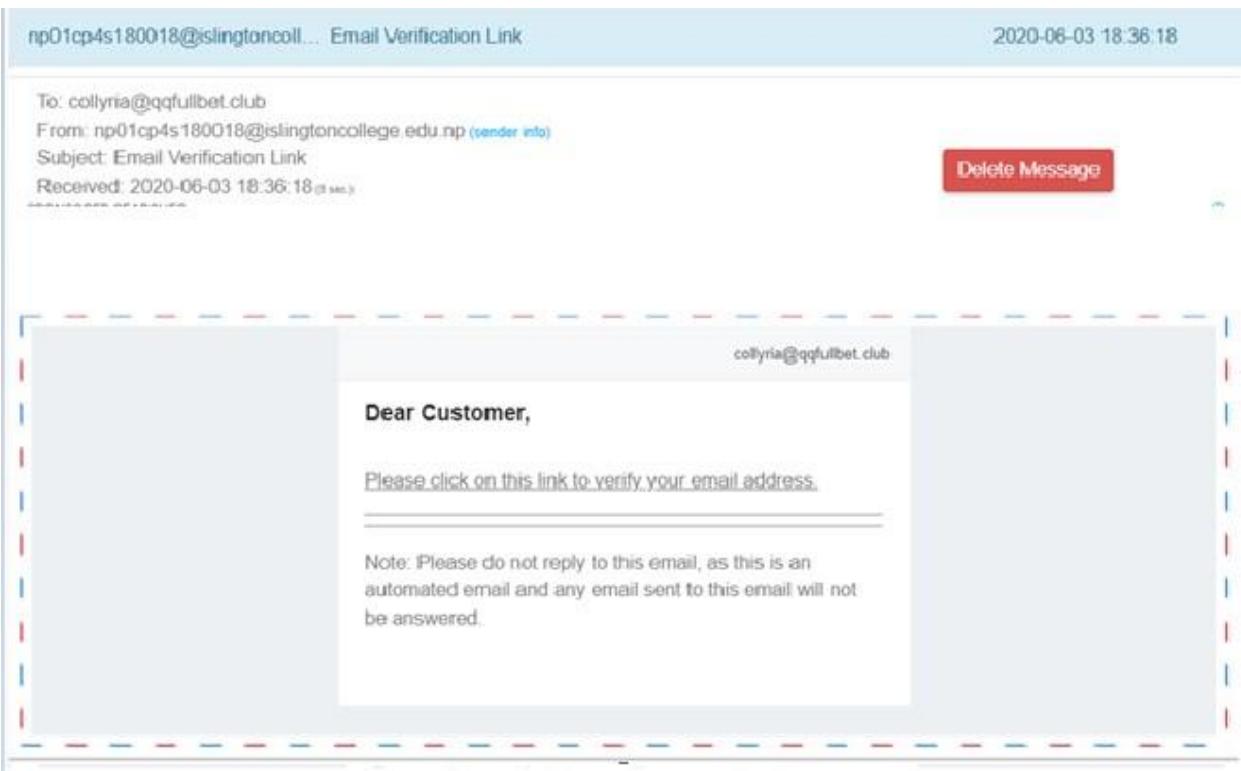


Figure 128: Verification email received by customer

contact	dob	email	img_string	is_enabled	name	password	status	type	username	
14:09:55	9546131	NULL	safishrestha12@gmail.com	NULL	0	The Administrator	\$2a\$10\$/97TLqet94FHpZBWWYsrS..A.2w5jFUyz2oLxRQ32al...	1	admin	administartor
10:10:57	9860027618	1999-11-19 18:15:00	safishrestha19@gmail.com	NULL	0	John Shrestha	\$2a\$10\$W2Q4/cj/GOyPntYdOh5EWeZzCx67BxGakl.PgE8tq7U...	1	client	John
13:09:12	986532659	2020-06-10 18:15:00	sthaamulya@gmail.com	NULL	0	amulyadon	\$2a\$10\$WIEfUyWks4YlcF.VCk62neAUNCkwgtTJEoDNfZ0u/IH...	1	client	amulya
18:37:17	2324932620	1993-11-17 18:15:00	collyria@qqfullbet.club	NULL	0	Bernard Chambers	\$2a\$10\$EUdXN3PLFmVm4pJxbixKQeUV7rAwanlsLDYoy3/aV73...	1	client	Nelle Doyle
18:40:37	9815234678	1997-07-08 18:15:00	collyria@qqfullbet.club	NULL	0	Georgie Ruiz	\$2a\$10\$3iOu296r6sRZscUkZYjftuisnXNTwJoe0GYs7XjLjbE...	1	client	George

Email Verification Link

From: <ee007f4383a2cc>
 To: <pehige2057@lerwfv.com>

Show Info

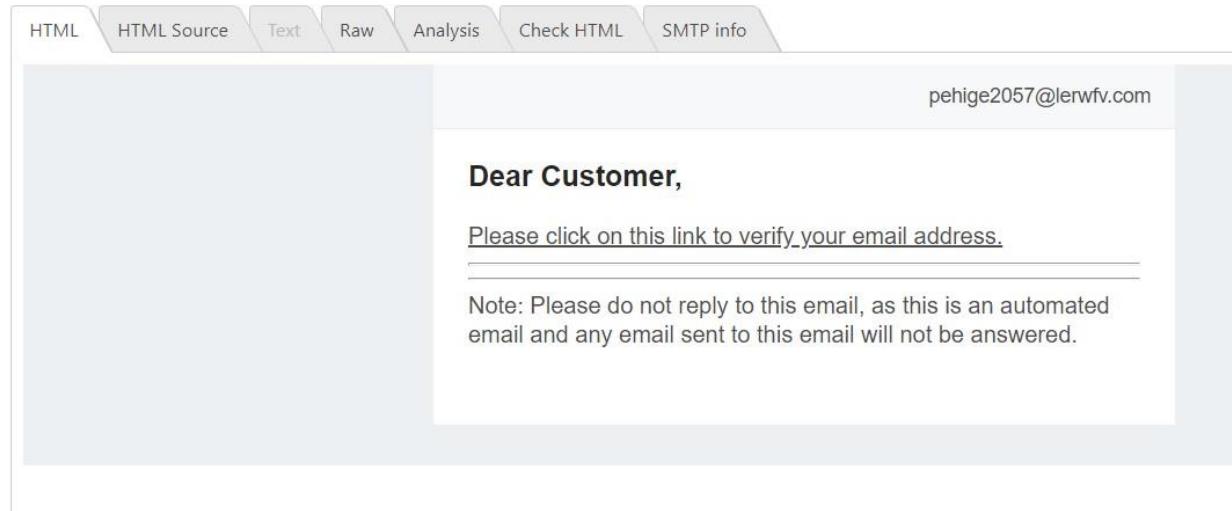


Figure 129: Verifying email while registering user

Figure 130: user details stored in database after registration

Empty Registration form

Objective	To test whether user can register themselves when text fields are empty.
Action	Click on Register button.
Expected Result	Application will not be able to register user and show missing fields with red text fields.
Actual Result	Application is not be able to register user and shows missing fields with red text fields.
Conclusion	Test Successful

Table 15: Test case for empty registration form

Client

As a Client...

Full Name	∅
Invalid Name!	
Username	∅
Invalid Username!	
Email	@
Invalid Email!	
Phone number	📞
Invalid Phone number!	
Date of Birth	📅
Invalid Date of Birth!	
Password	🔒
Invalid Password!	

Figure 131: Displaying error for empty text field while registering as new user

Registered User Login

Objective		Login with registered user credentials
Action	Enter registered user login credentials	
Expected Result	Login successfully and redirect to customer dashboard.	
Actual Result	Logged in successfully and redirected to customer dashboard.	
Conclusion	Test Successful	

Table 16: Test case for registered user login

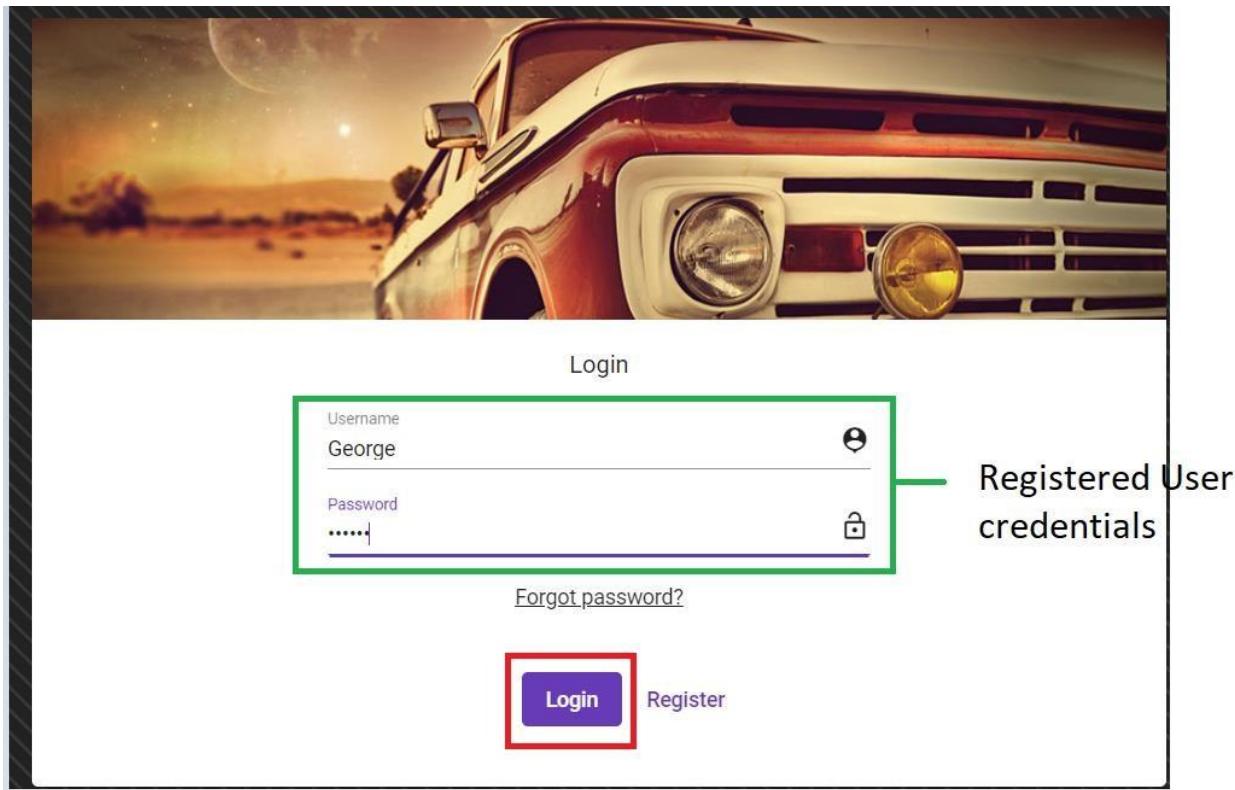


Figure 132: User login

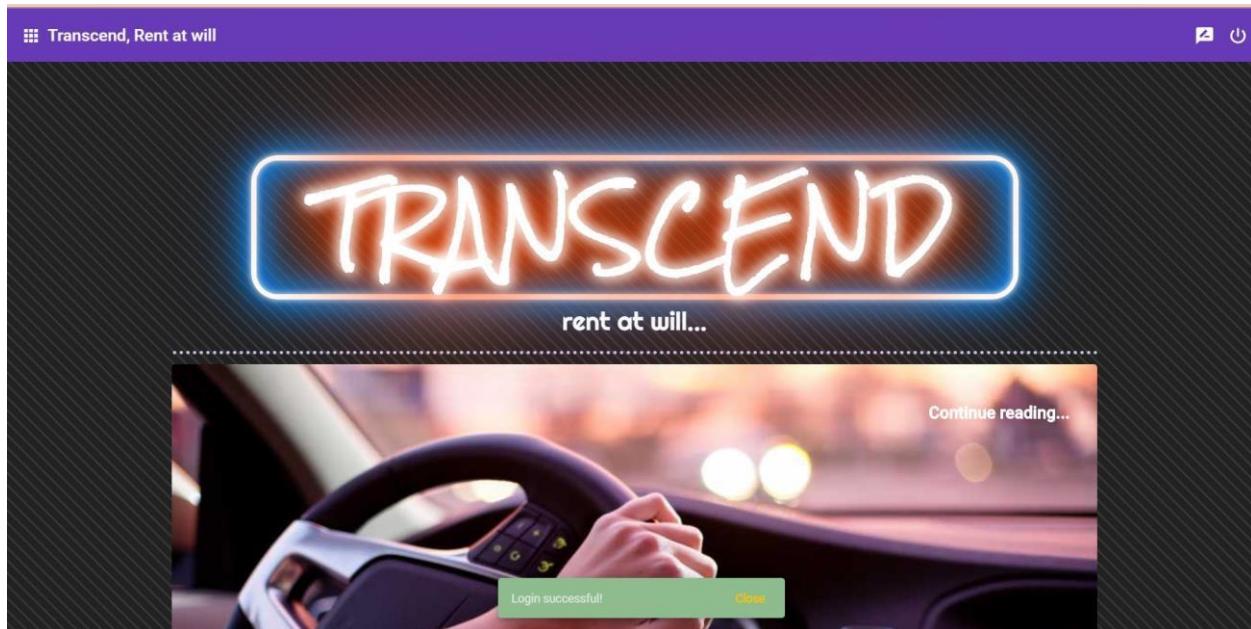


Figure 133: Redirected to customer dashboard

User Logout

Objective	To logout from customer panel and redirect to login page after clicking 'Log out' button
Action	Click on Log Out button
Expected Result	Application will log out and redirect to login page.
Actual Result	Application redirects to login page after logout.
Conclusion	Test Successful

Table 17: Test case for User logout

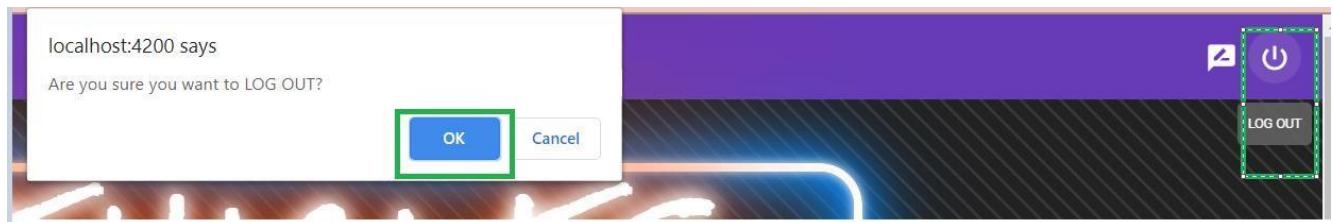


Figure 134: Logging out from user panel

Login

Username

Password

[Forgot password?](#)

Figure 135: Redirecting to login page after log out

Forgot Password

Objective	To test if mail is sent to user for resetting password
Action	Click on Forget Password button and send email.
Expected Result	Application will send email with OTP to user with valid email and user will be able to reset password.
Actual Result	Application sent email with OTP to user with valid email and user will be able to reset password.
Conclusion	Test Successful

Table 18: Test case for User Forgot password

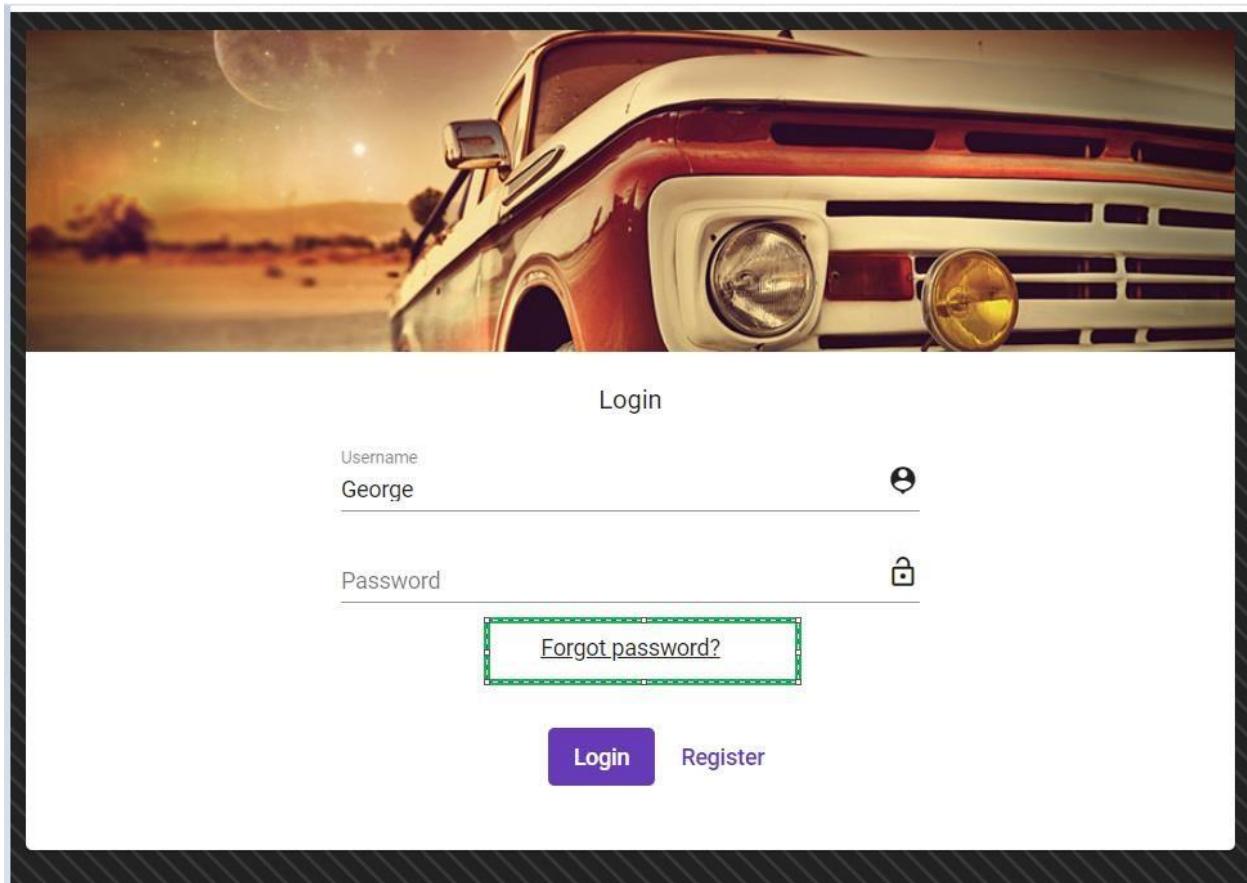


Figure 136: Click on 'Forgot Password?'

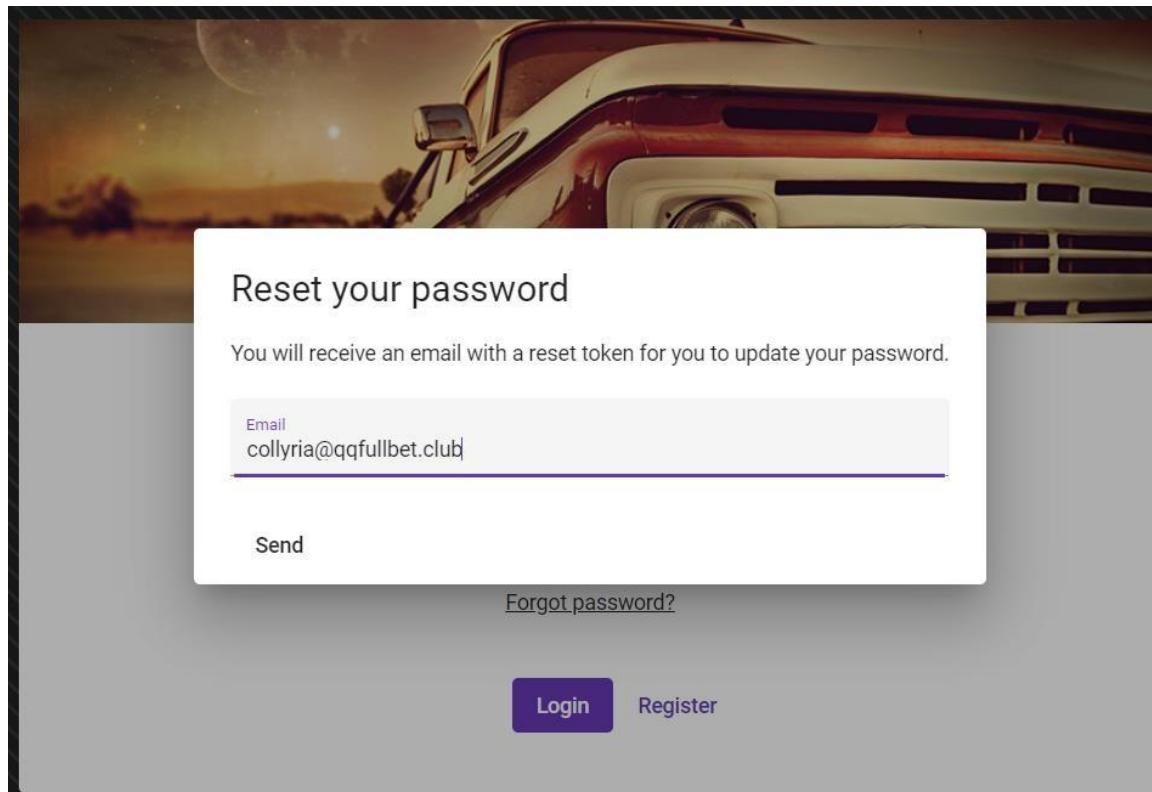


Figure 137: Enter Email on the given field

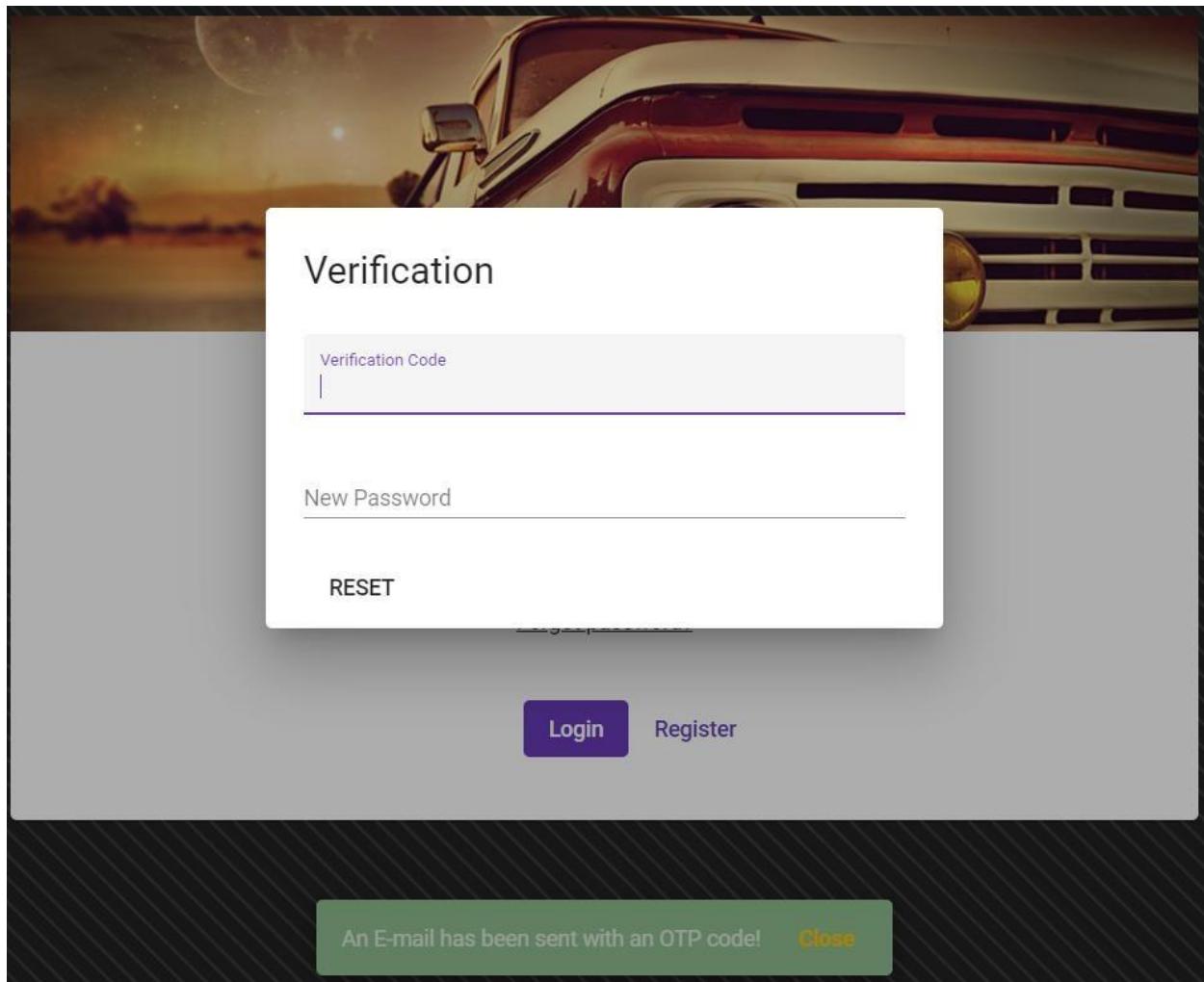


Figure 138: Email sent with OTP code success message

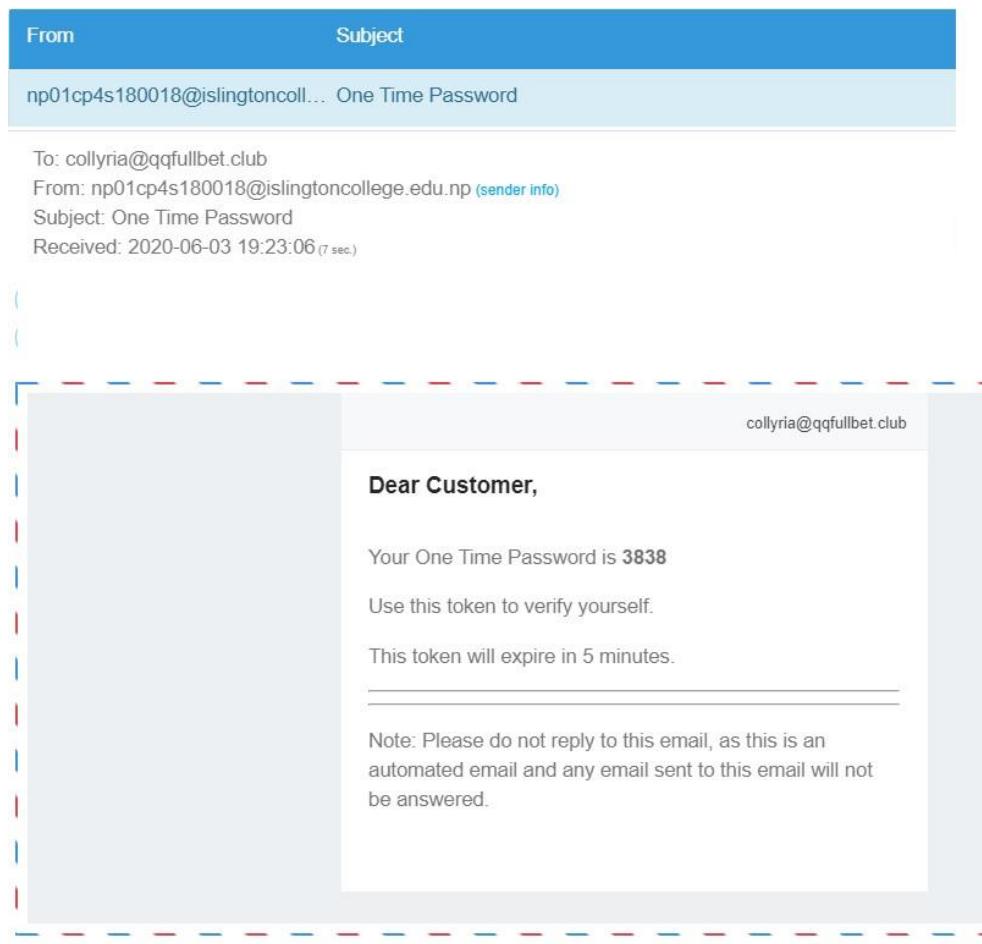


Figure 139: OTP sent via email to reset password

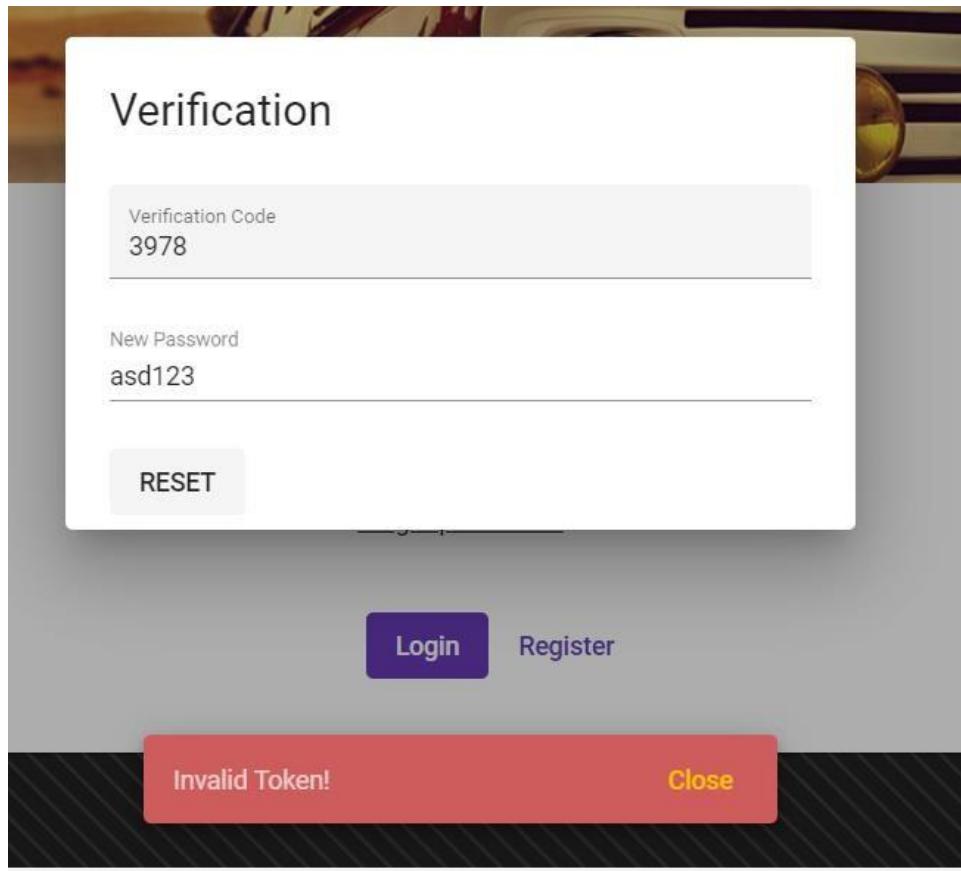


Figure 140: Message displayed in case of wrong token entered.

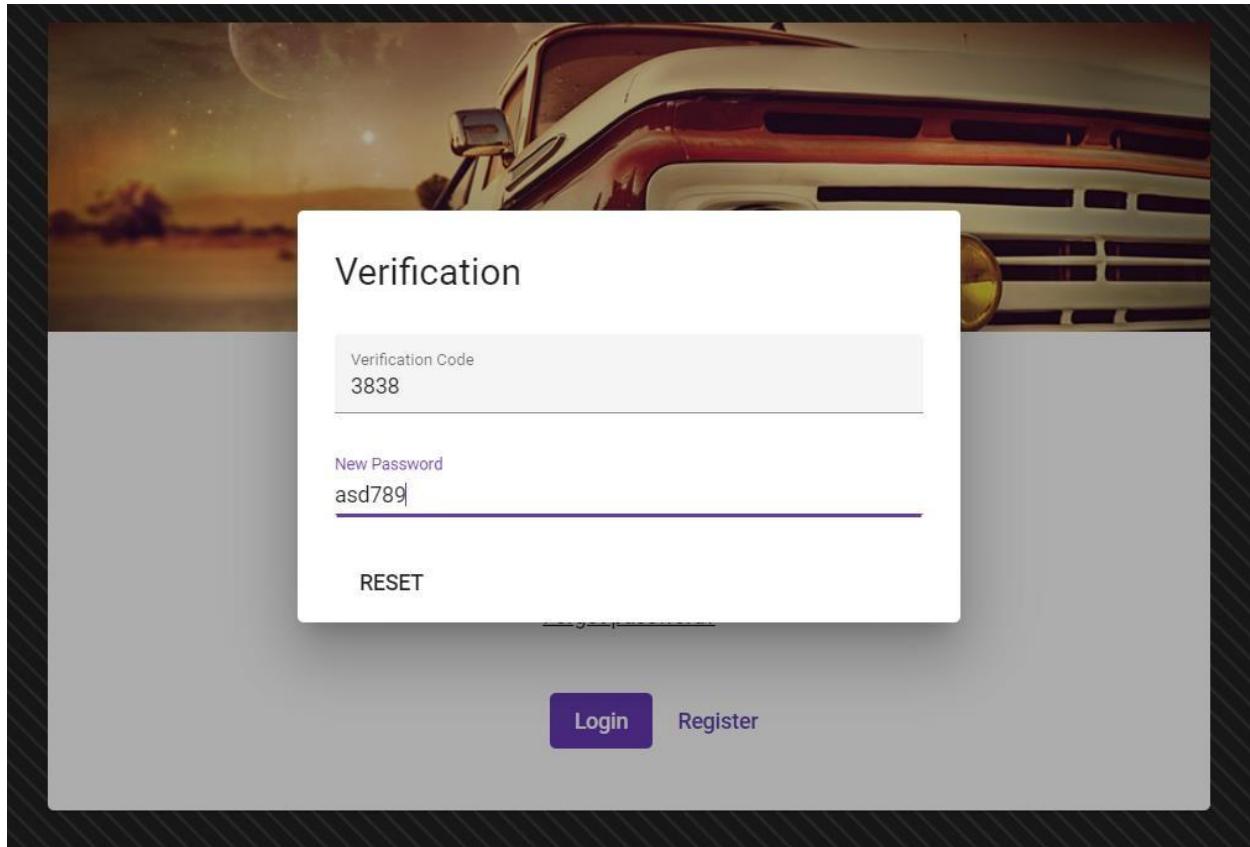


Figure 141: Resetting password with valid OTP

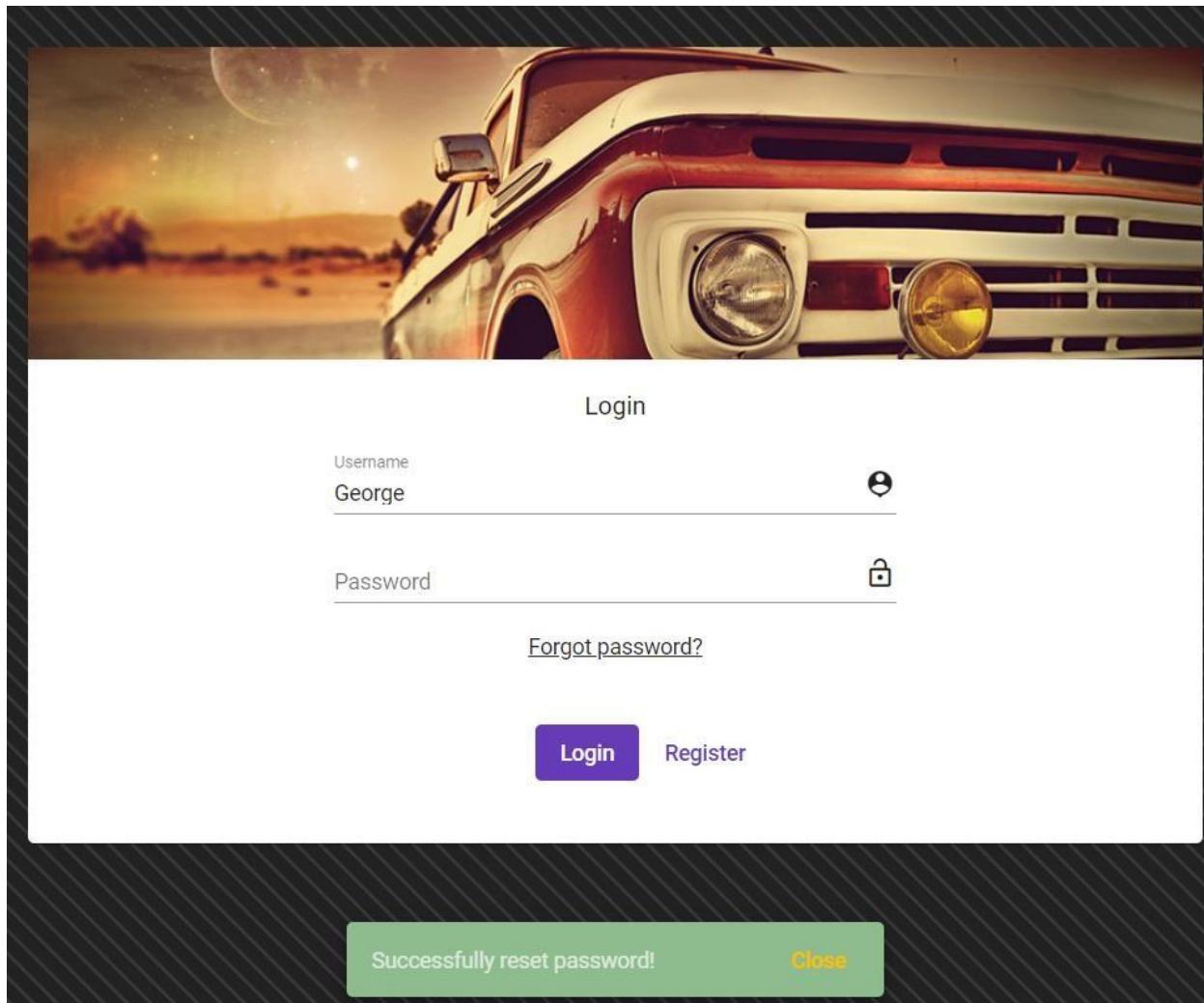


Figure 142: Reset password success message

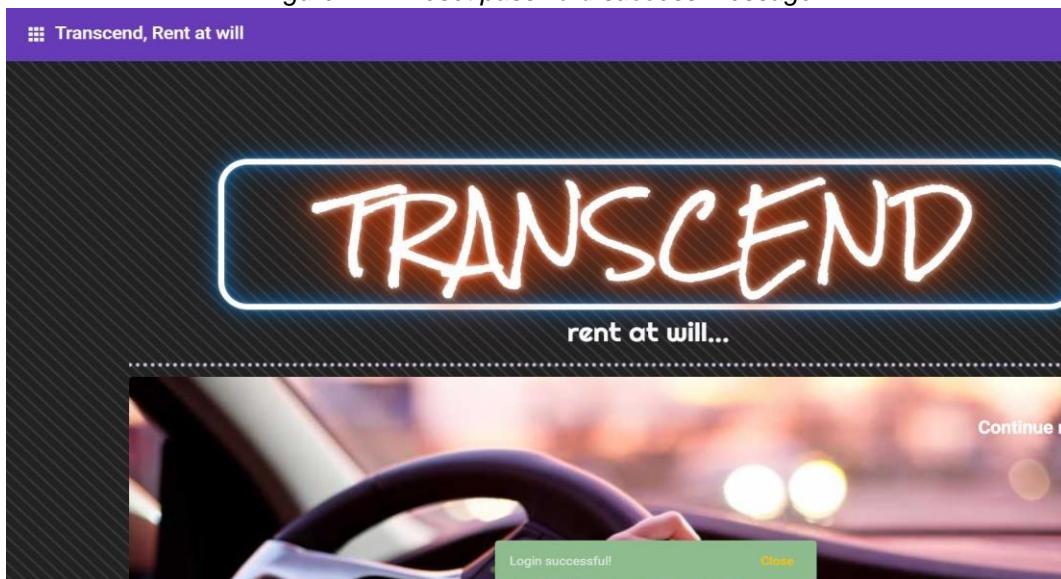


Figure 143: Login success after resetting password

Change password

Objective	Change password for User
Action	Click profile and enter new password.
Expected Result	Password will change and credentials will be updated redirecting to login page.
Actual Result	Password changed with updated credentials and redirected to login page.
Conclusion	Test Successful

Table 19: Test case for user password change

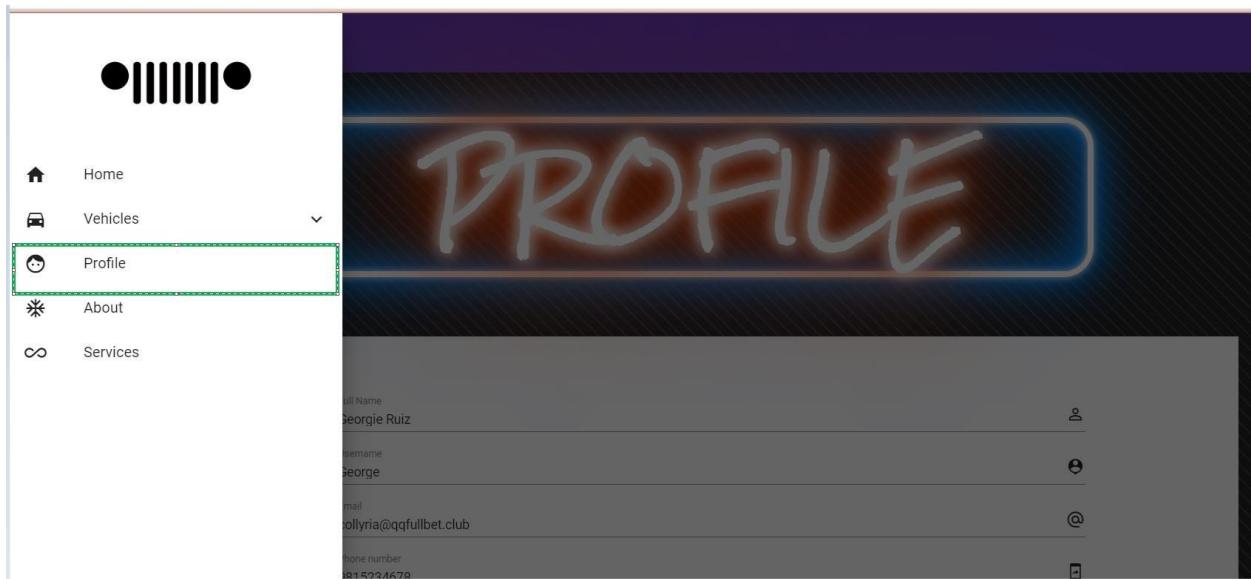


Figure 144: Click User 'Profile'

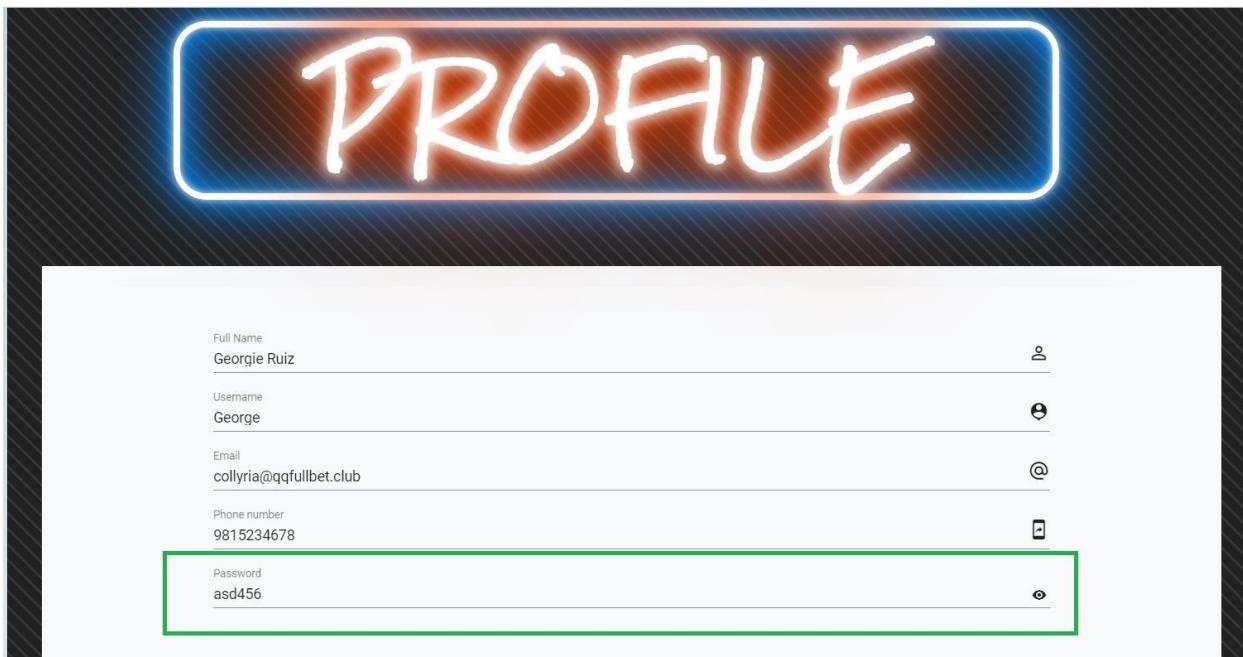


Figure 145 : Change user password

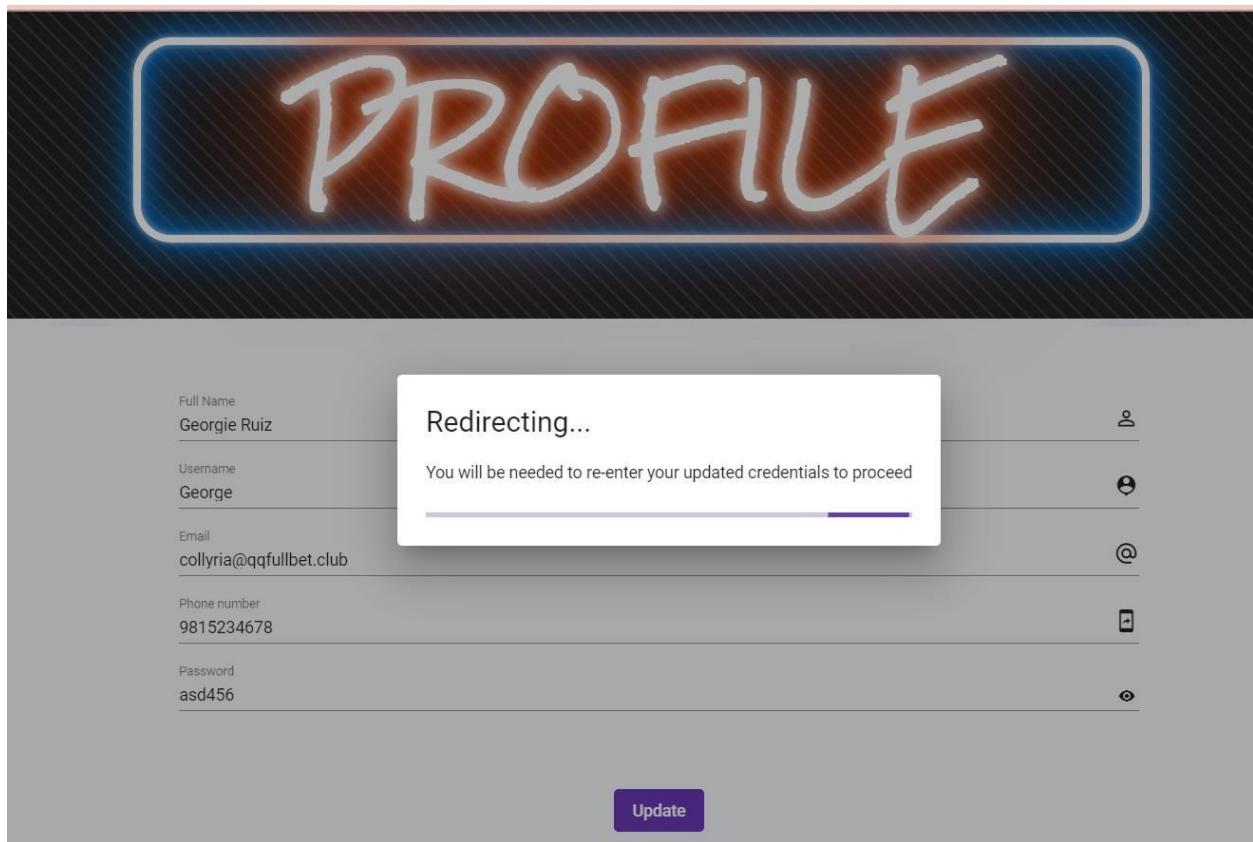


Figure 146: Display updating credentials

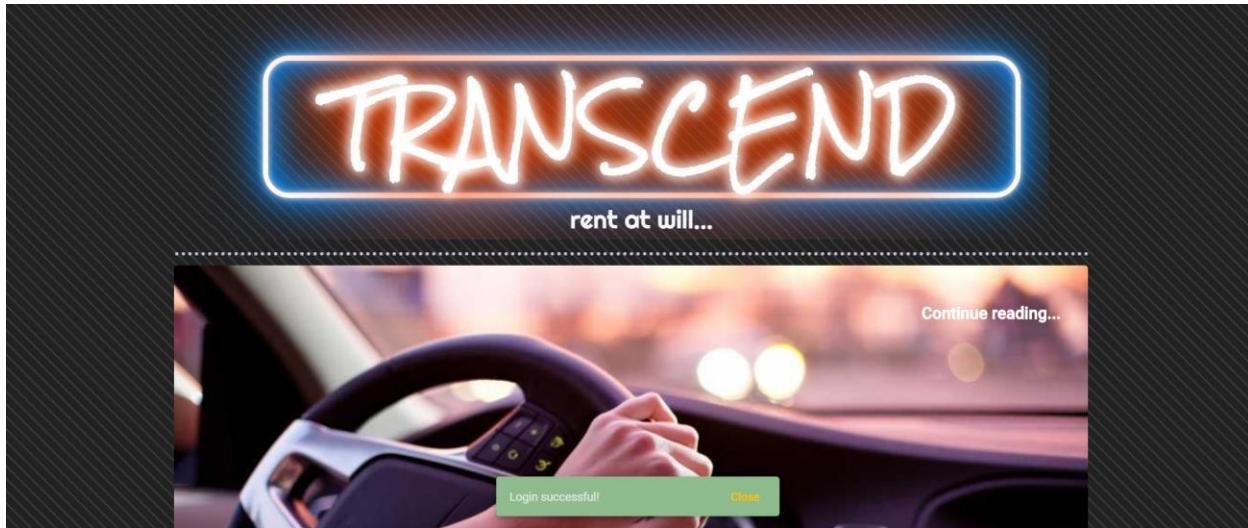


Figure 147: Successful login after changing password

Book vehicles

Objective	To test whether user can book vehicles.
Action	Click on Book button and enter details to book vehicle.
Expected Result	Application will display booking successful message and reserves the vehicle for user.
Actual Result	Application will display booking successful message and reserves the vehicle for user.
Conclusion	Test Successful

Table 20: Test case for booking vehicles

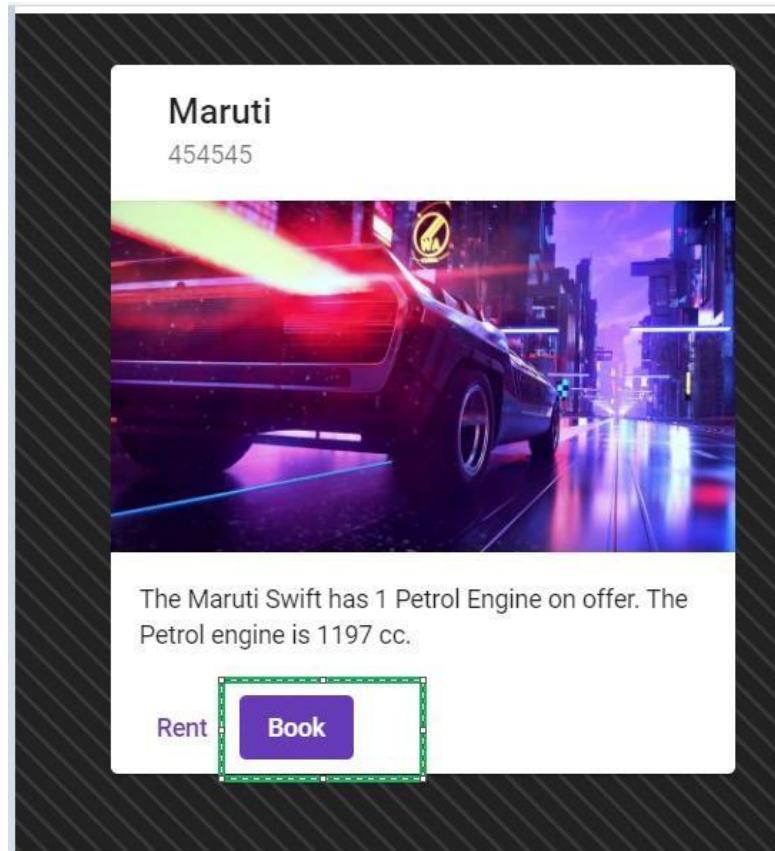


Figure 148: Click on book button to book vehicle

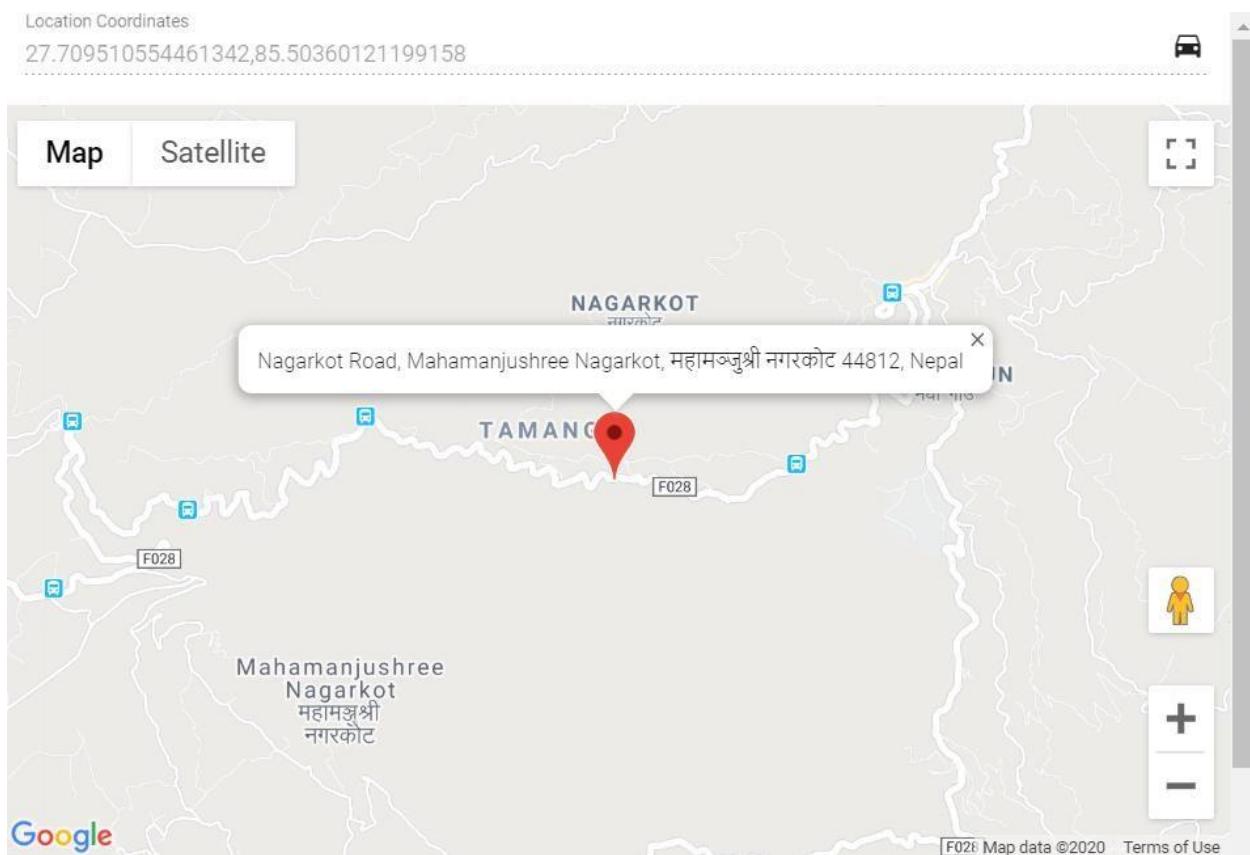


Figure 149: Select destination coordinates via map

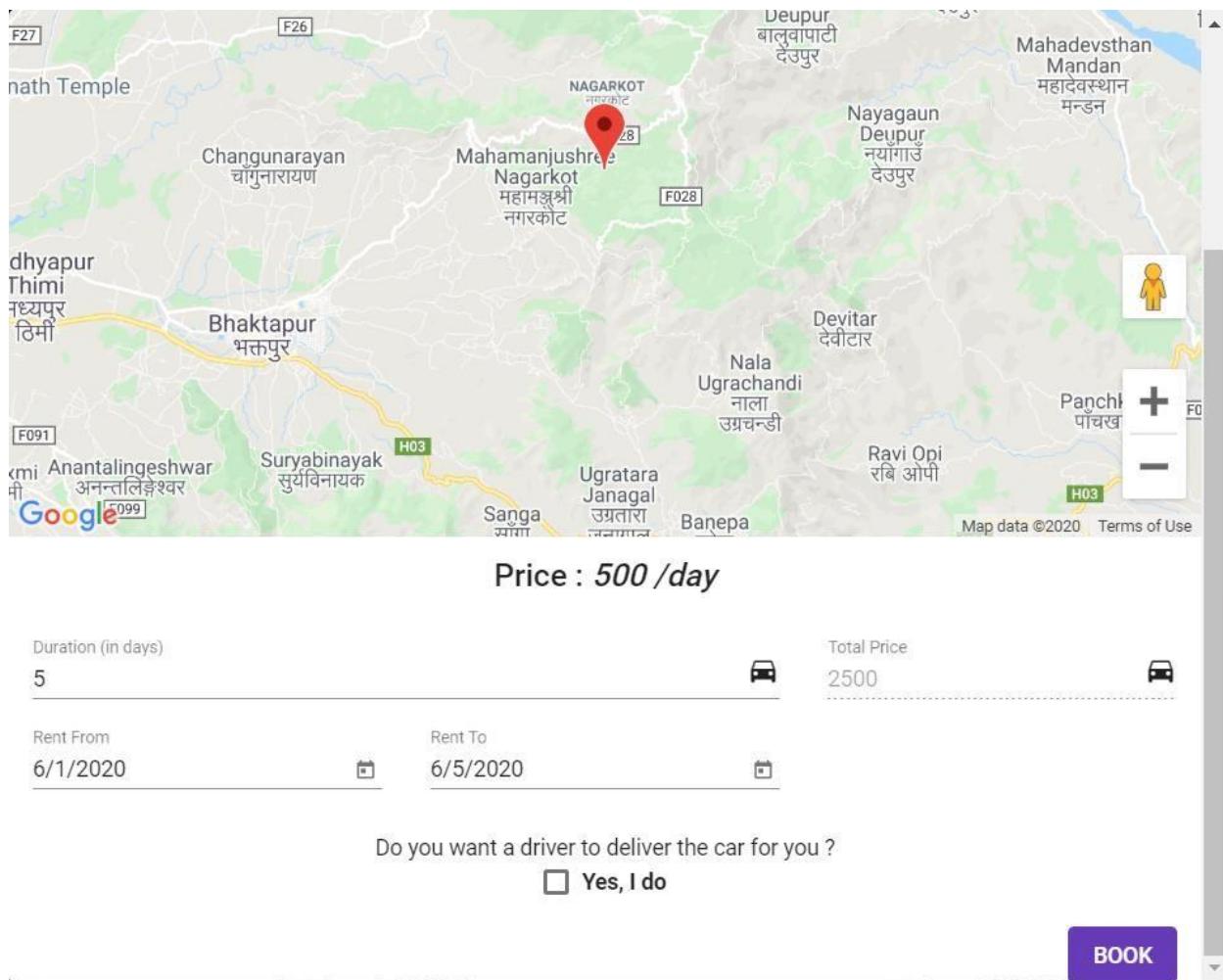


Figure 150: Fill up booking details and click button book

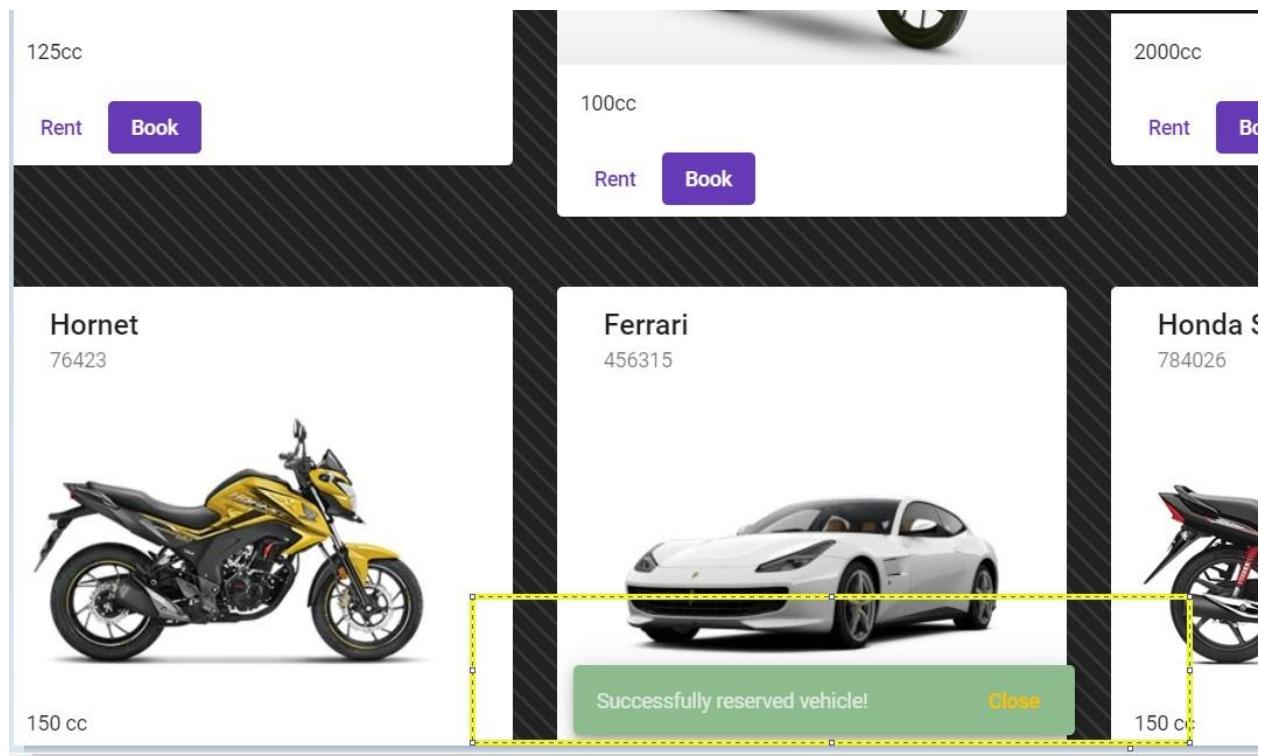


Figure 151: Vehicle reserved successful message

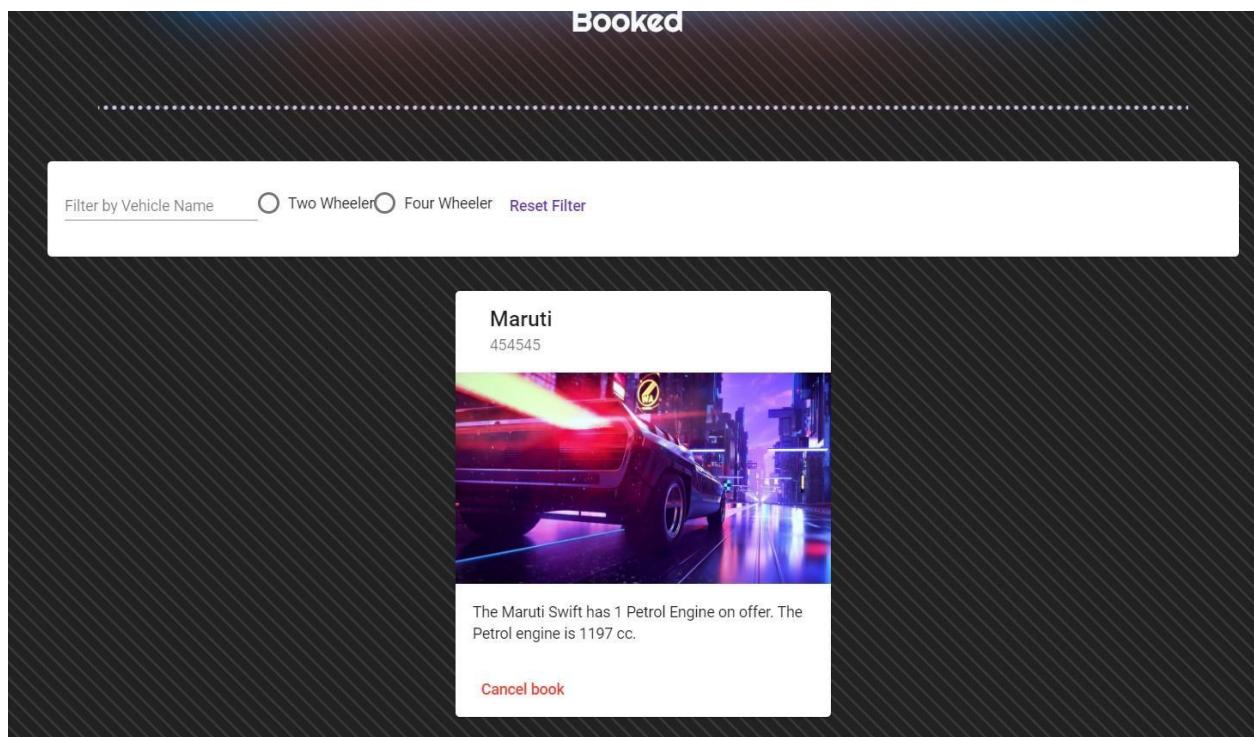


Figure 152: Displaying vehicle reserved by user

Rent vehicles

Objective	To test whether user can rent vehicles.
Action	Click on Rent button.
Expected Result	Application will display rented successfully message and sends rent request to admin.
Actual Result	Application will display rented successfully message and sends rent request to admin.
Conclusion	Test Successful

Table 21: Test case for renting vehicles

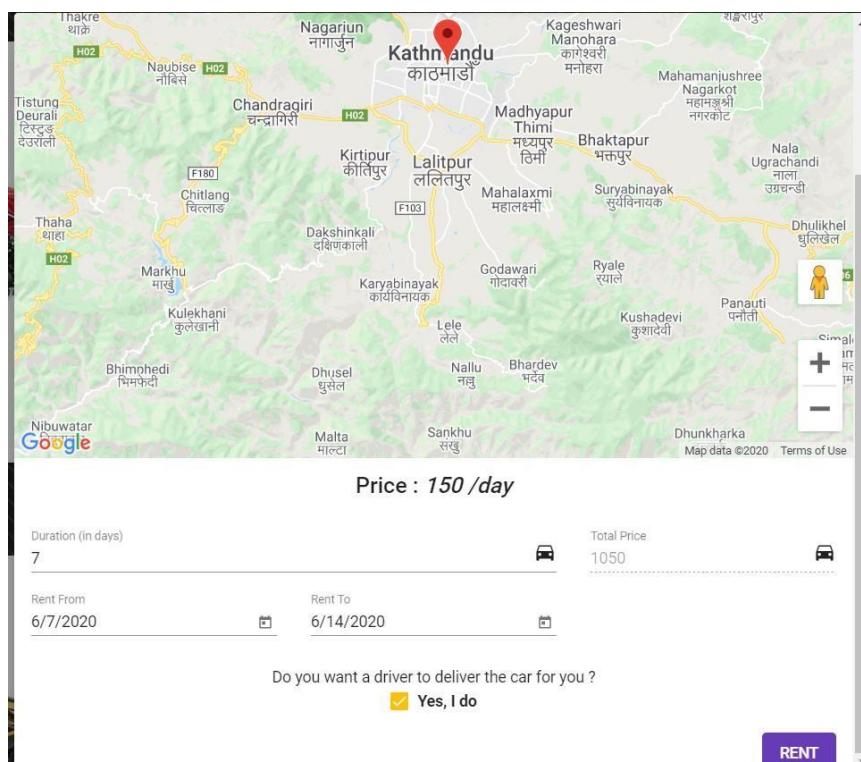


Figure 153: Click on rent button to rent vehicle

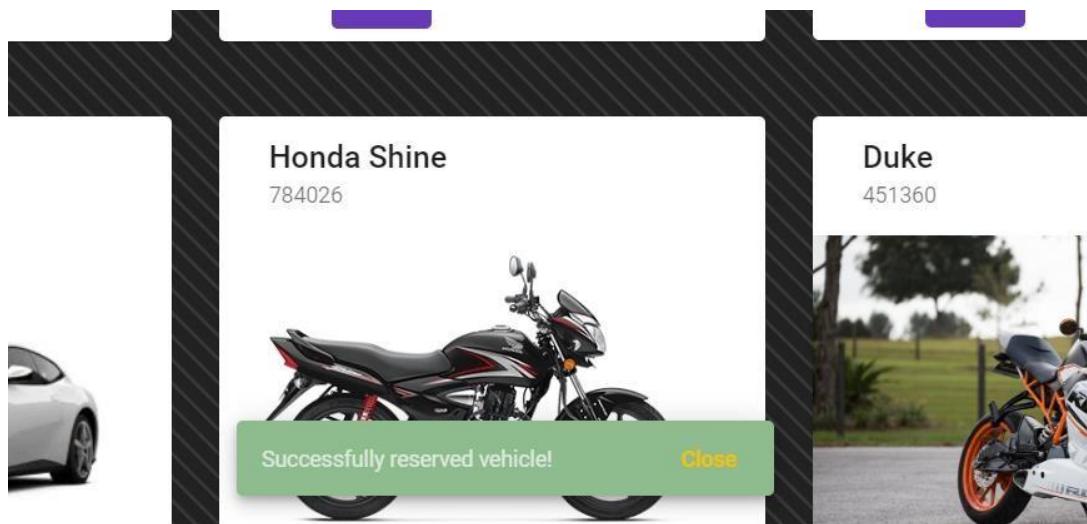


Figure 154: Displaying success message.

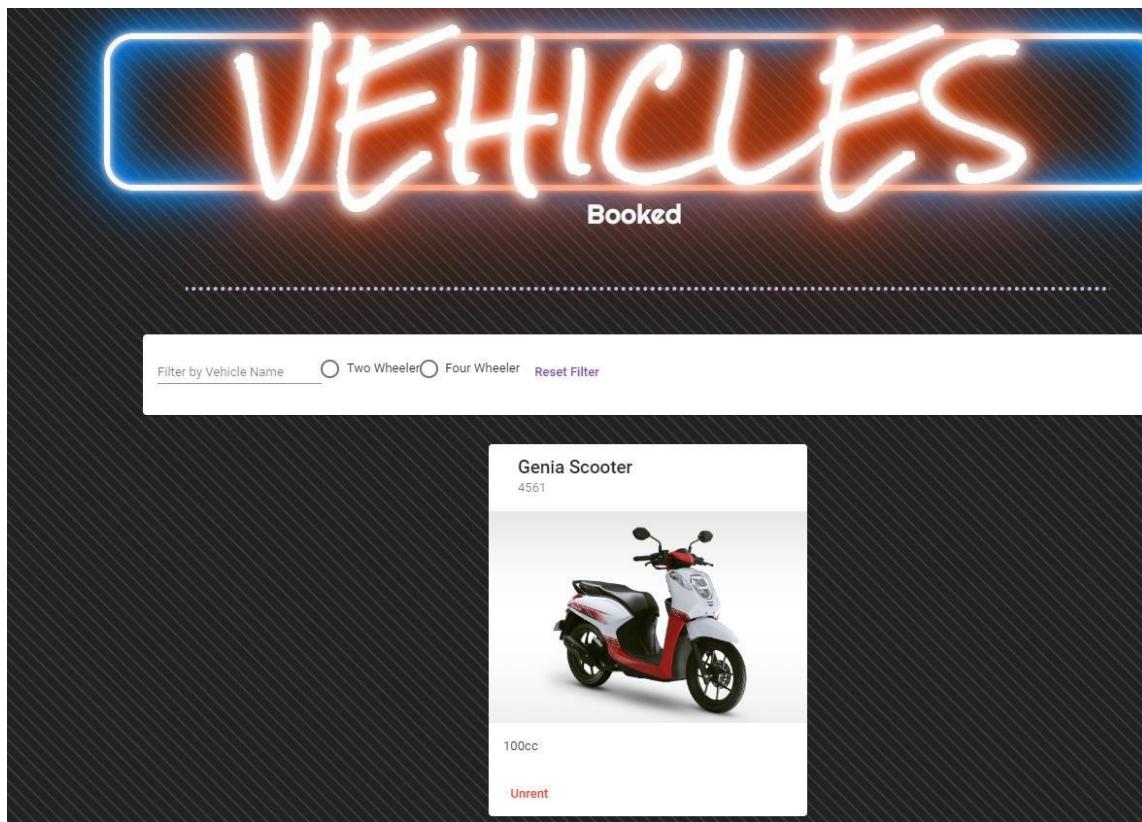


Figure 155: Displaying rented vehicle

Calculating Price

Objective

To test automatic price calculation.

Action	Enter total duration for rent and book in days.
Expected Result	Application will calculate and display total price for vehicle.
Actual Result	Application will calculate and display total price for vehicle.
Conclusion	Test Successful

Table 22: Test case for automatic price calculation

The screenshot shows a user interface for a car rental application. At the top, a green box displays the text "Price : 500 /day". Below this, there is a form field labeled "Duration (in days)" containing the value "5", with a small car icon to its right. To the right of this field is another green box labeled "Total Price" with the value "2500" and a car icon. Below these fields are two date inputs: "Rent From" set to "6/1/2020" and "Rent To" set to "6/5/2020", each accompanied by a calendar icon. Underneath the dates is a question: "Do you want a driver to deliver the car for you ?" followed by a checkbox labeled "Yes, I do". In the bottom right corner, there is a large purple button with the word "BOOK" in white capital letters.

Figure 156: Automatic price calculation for rent and book -1

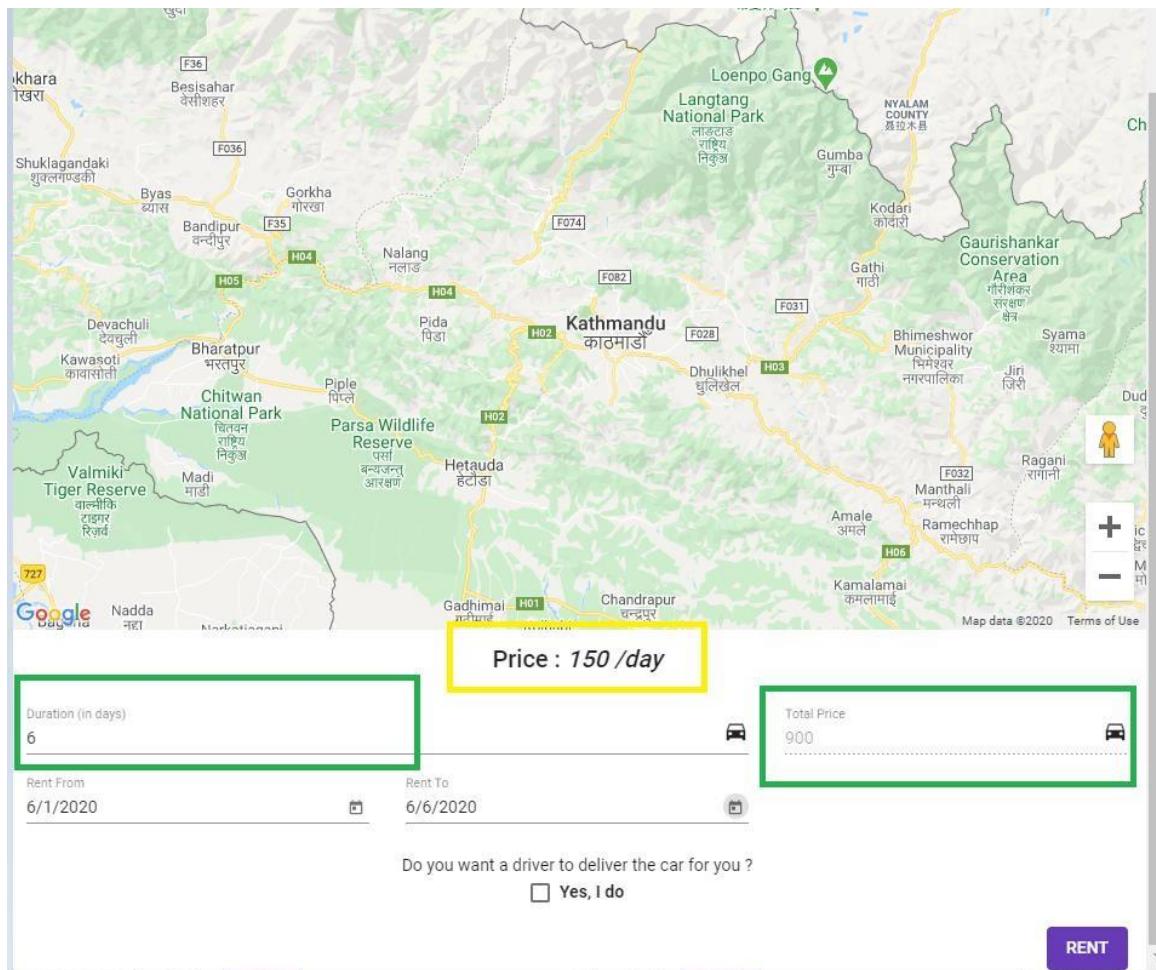


Figure 157: Automatic price calculation for rent and book -2

Available vehicles

Objective	To test whether user can view all available vehicles with their details.
Action	Click on Available Vehicle button.
Expected Result	View all available vehicles with their details to user.
Actual Result	Views all available vehicles with their details to user.
Conclusion	Test Successful

Table 23: Test case for viewing available vehicles

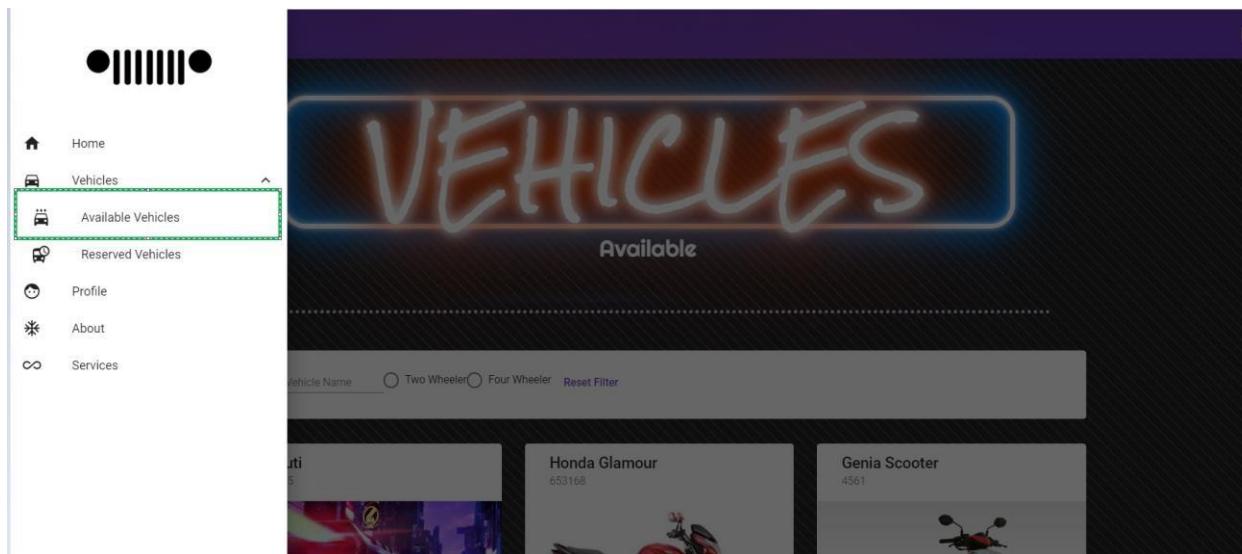


Figure 158: Click on Available Vehicles

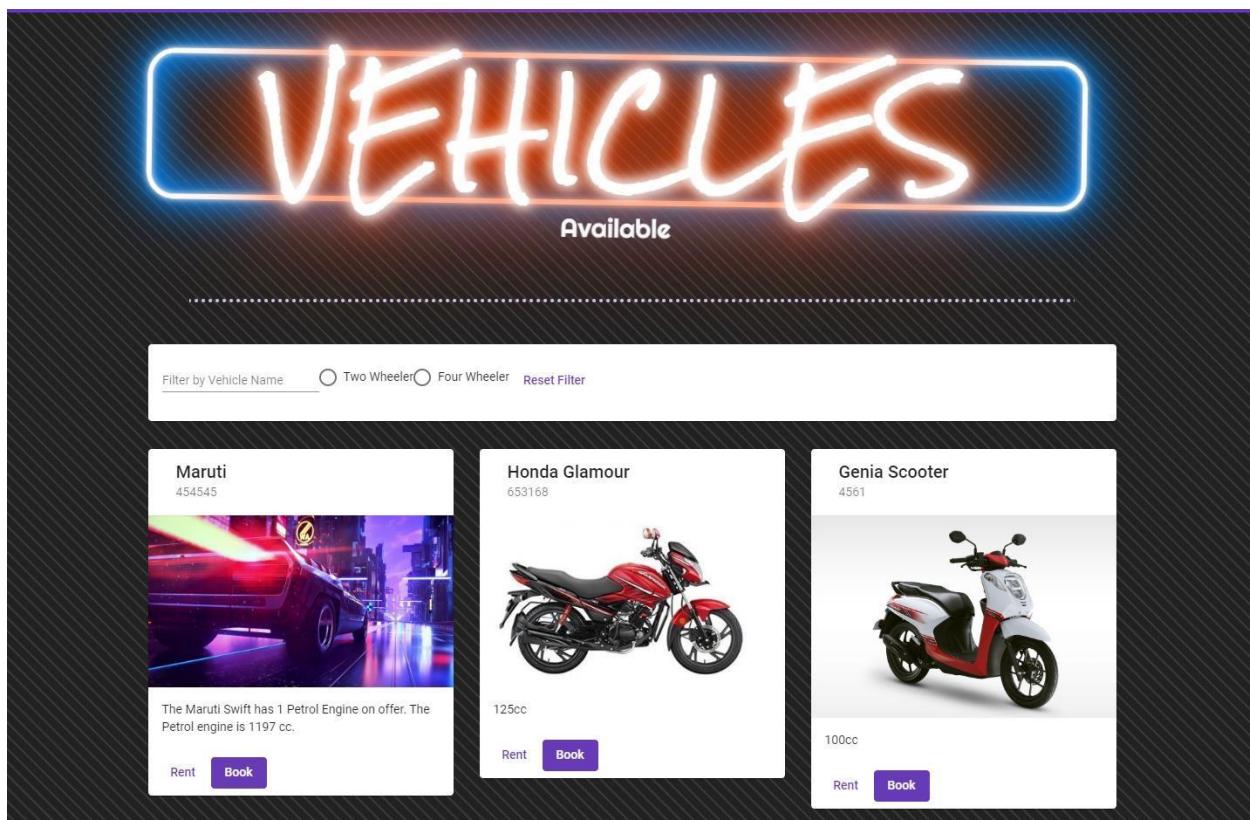


Figure 159: Displaying available vehicles

Search vehicles

Objective	To test whether user can search vehicles.
Action	Enter vehicle name or select radio button to search by vehicle type.

Expected Result	View vehicles according to search.
Actual Result	Vehicles viewed according to search.
Conclusion	Test Successful

Table 24: Test case for viewing search vehicles

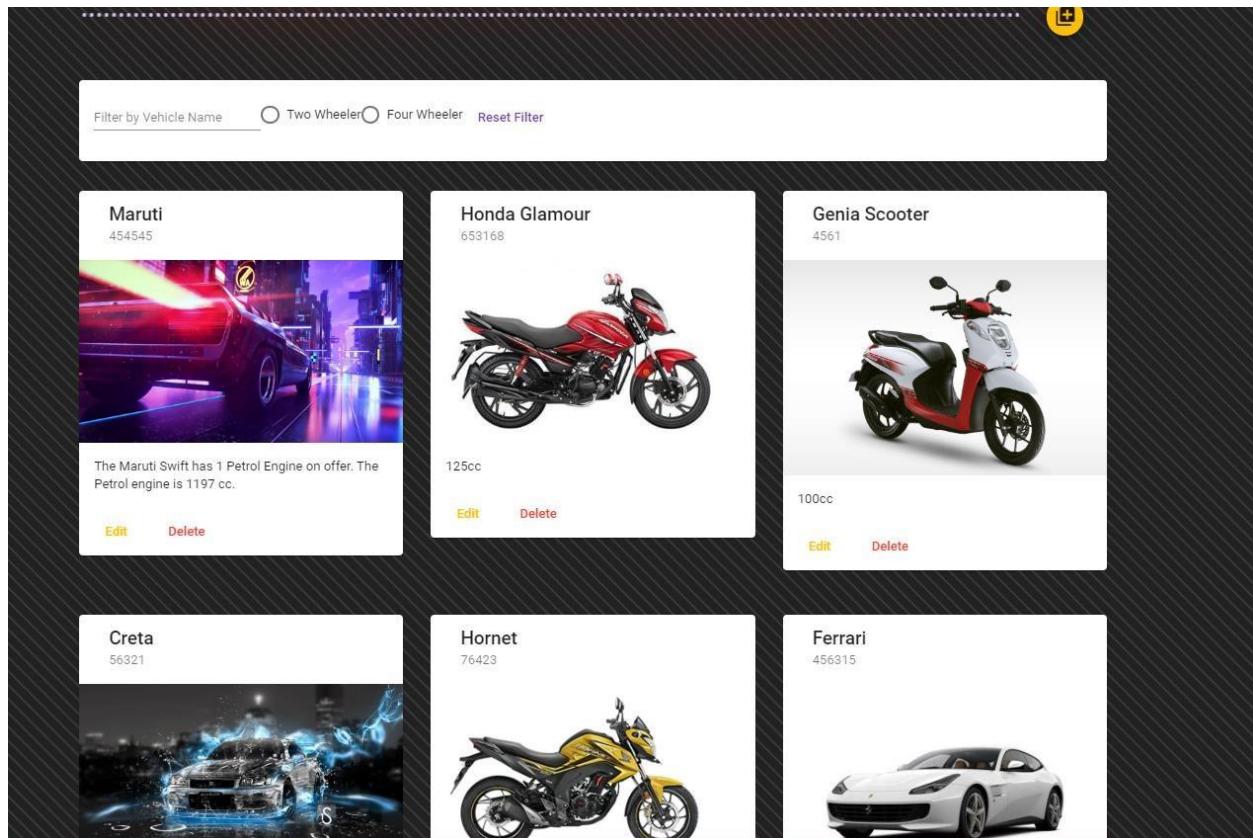


Figure 160: Before search

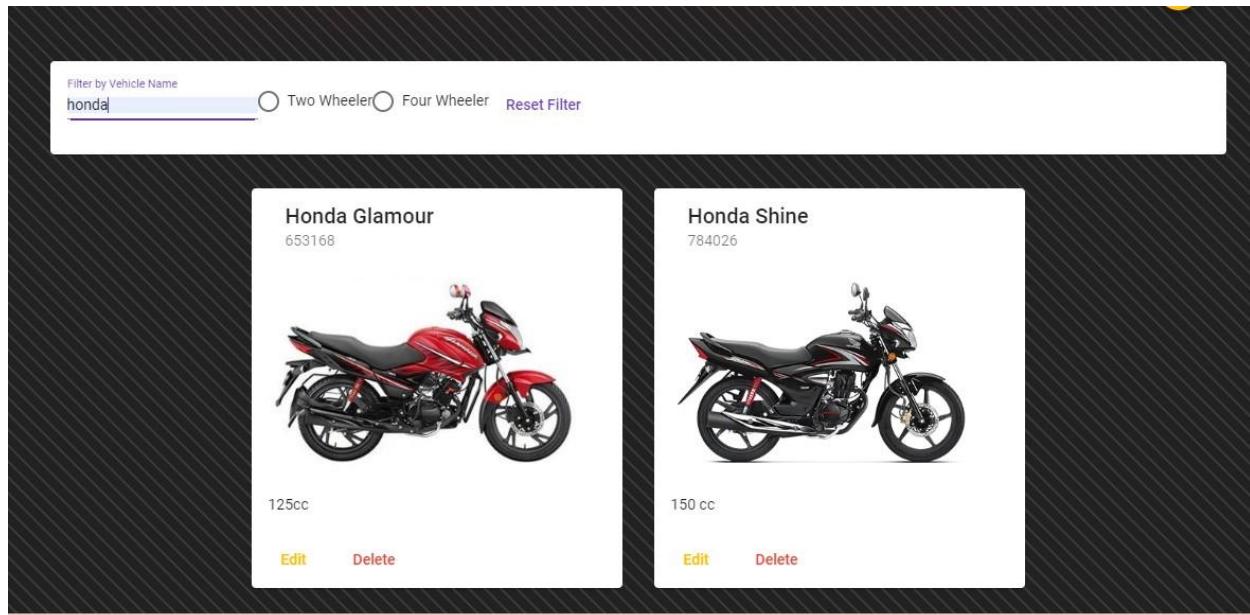


Figure 161 : Displaying vehicles according to search by Vehicle Name

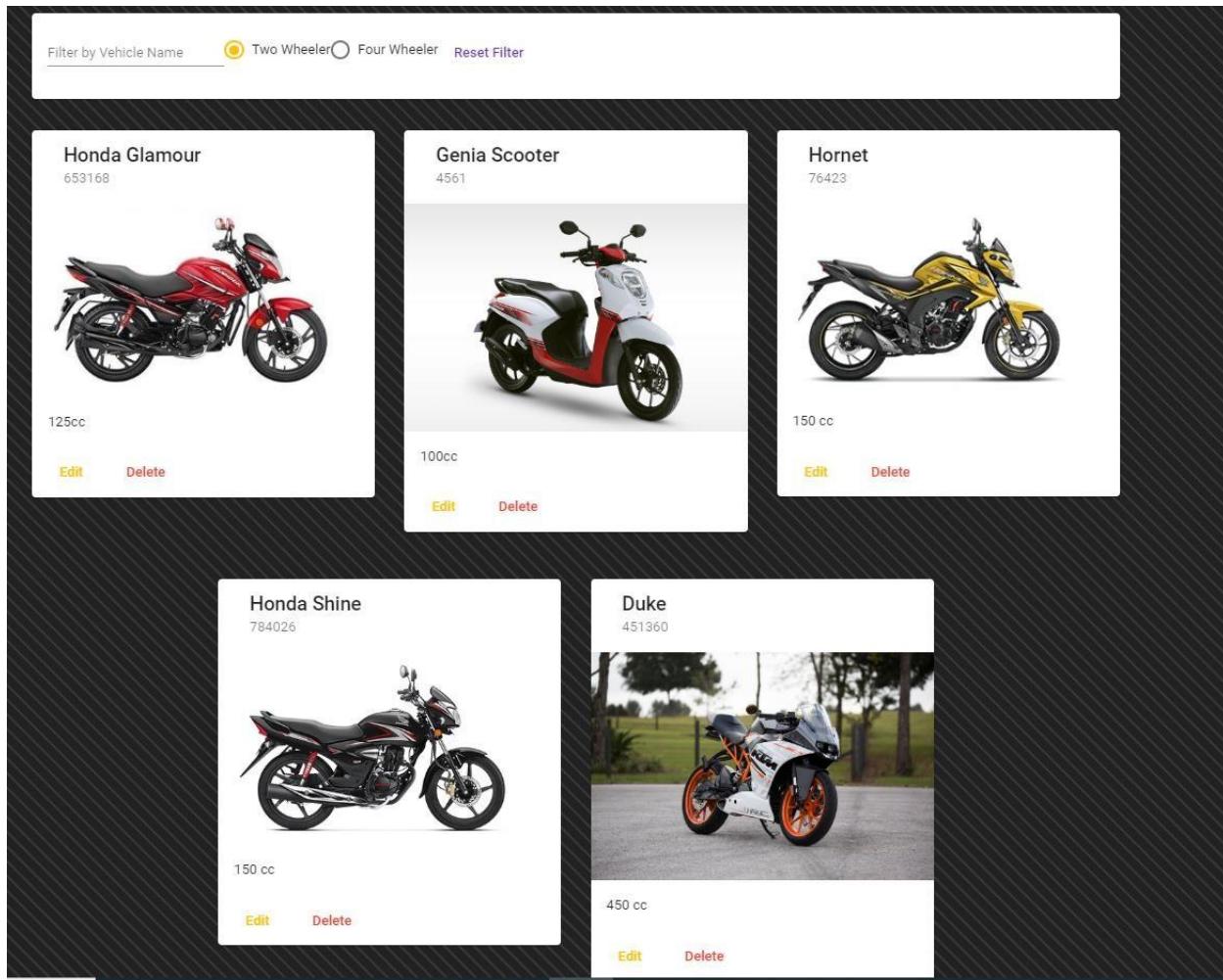


Figure 162: Displaying vehicles according to search by Vehicle Type Two-Wheeler

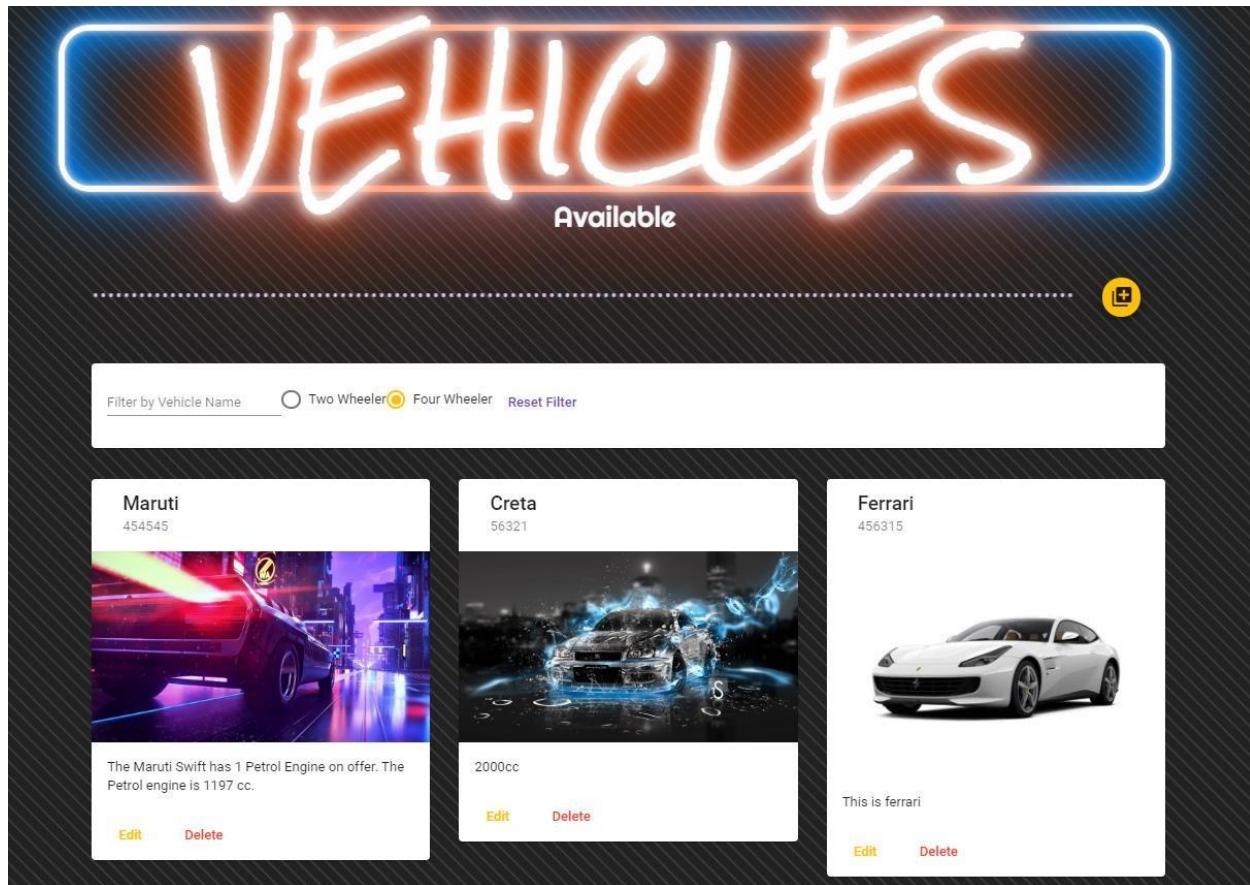


Figure 163: Displaying vehicles according to search by Vehicle Name Four-Wheeler

Feedback

Objective	To test send feedback feature.
Action	Click on Feedback button.
Expected Result	Send feedback and store in database.
Actual Result	Sends feedback and stores in database.
Conclusion	Test Successful

Table 25: Test case for sending feedback

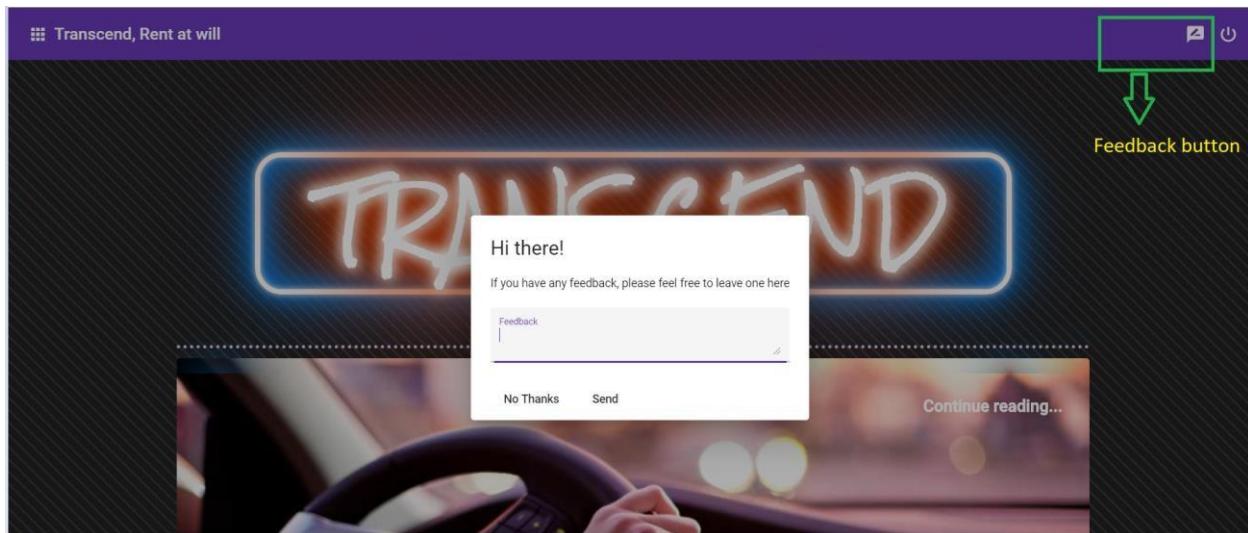


Figure 164: Displaying feedback form on button click

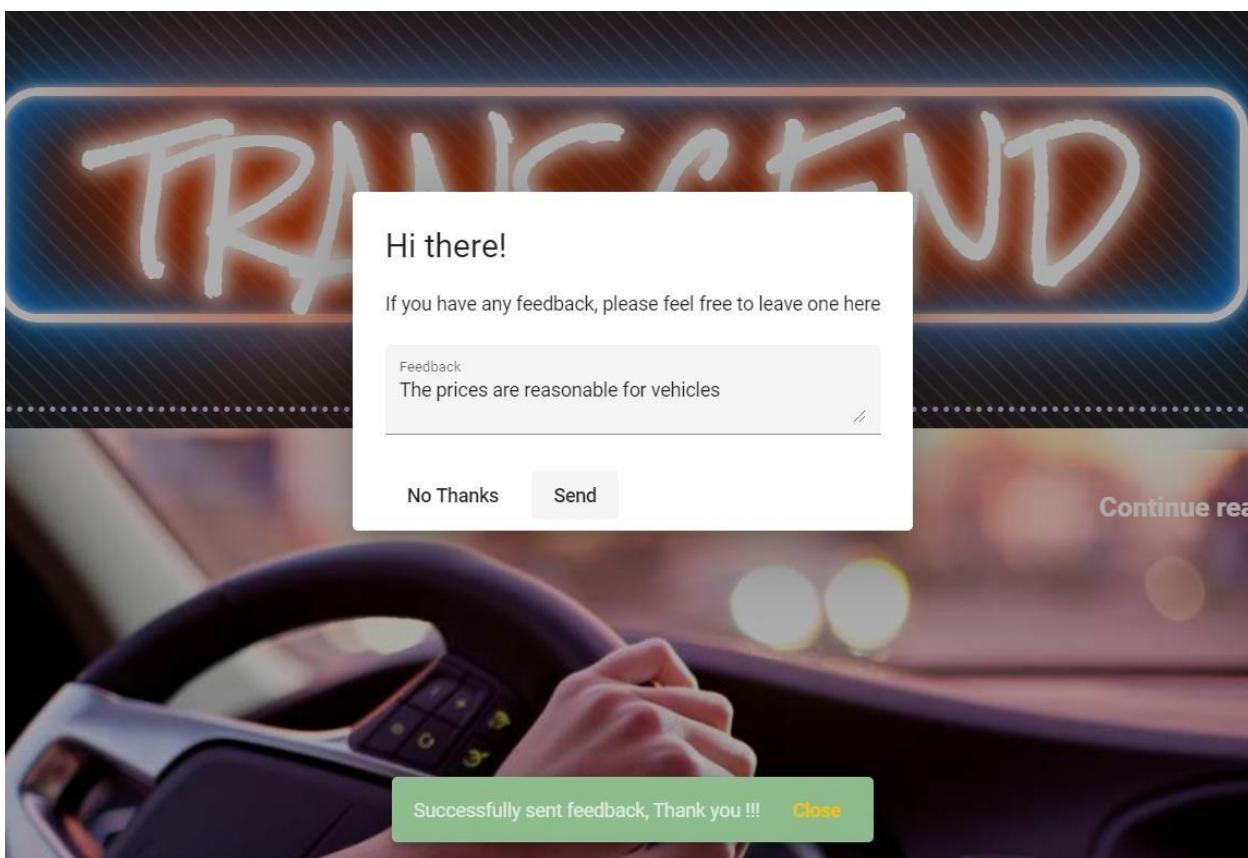


Figure 165: Sending feedback

<input type="checkbox"/> Show all Number of rows: 25 <input type="button" value="▼"/> Filter rows: <input type="text" value="Search this table"/> Sort by key: <input type="button" value="None"/>						
<input type="checkbox"/> + Options <input type="button" value="T"/> id created feedback user_id modified_by_id						
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	18	2020-05-28 10:11:31	this feedback
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	35	2020-06-03 14:06:25	Service is good
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	59	2020-06-03 19:56:31	The prices are reasonable for vehicles

Figure 166: Feedback stored in database

Empty fields

Objective	To test whether user can book and rent when text fields are empty.
Action	Click on Register button.
Expected Result	Application will not be able to register user and show missing fields with red text fields.
Actual Result	Application is not be able to register user and shows missing fields with red text fields.
Conclusion	Test Successful

Table 26: Test case for empty text field while booking and renting vehicle

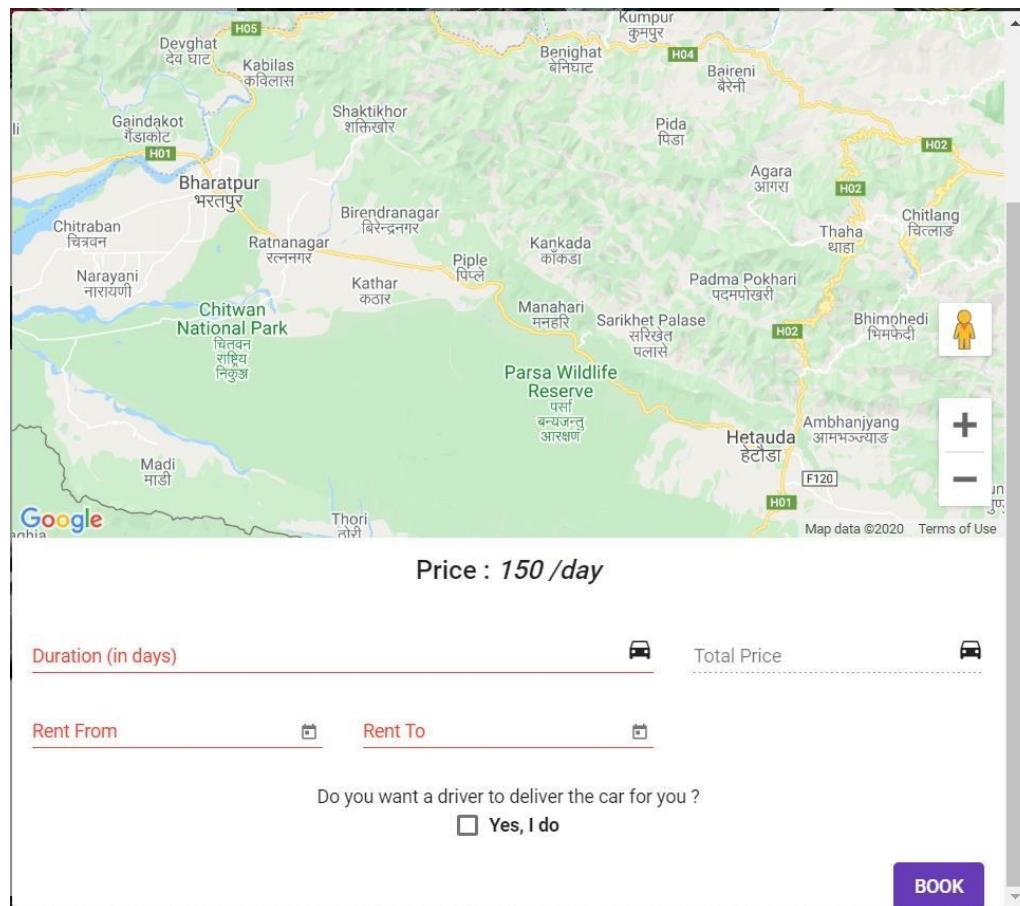


Figure 167: Displaying error for empty text field while booking vehicle

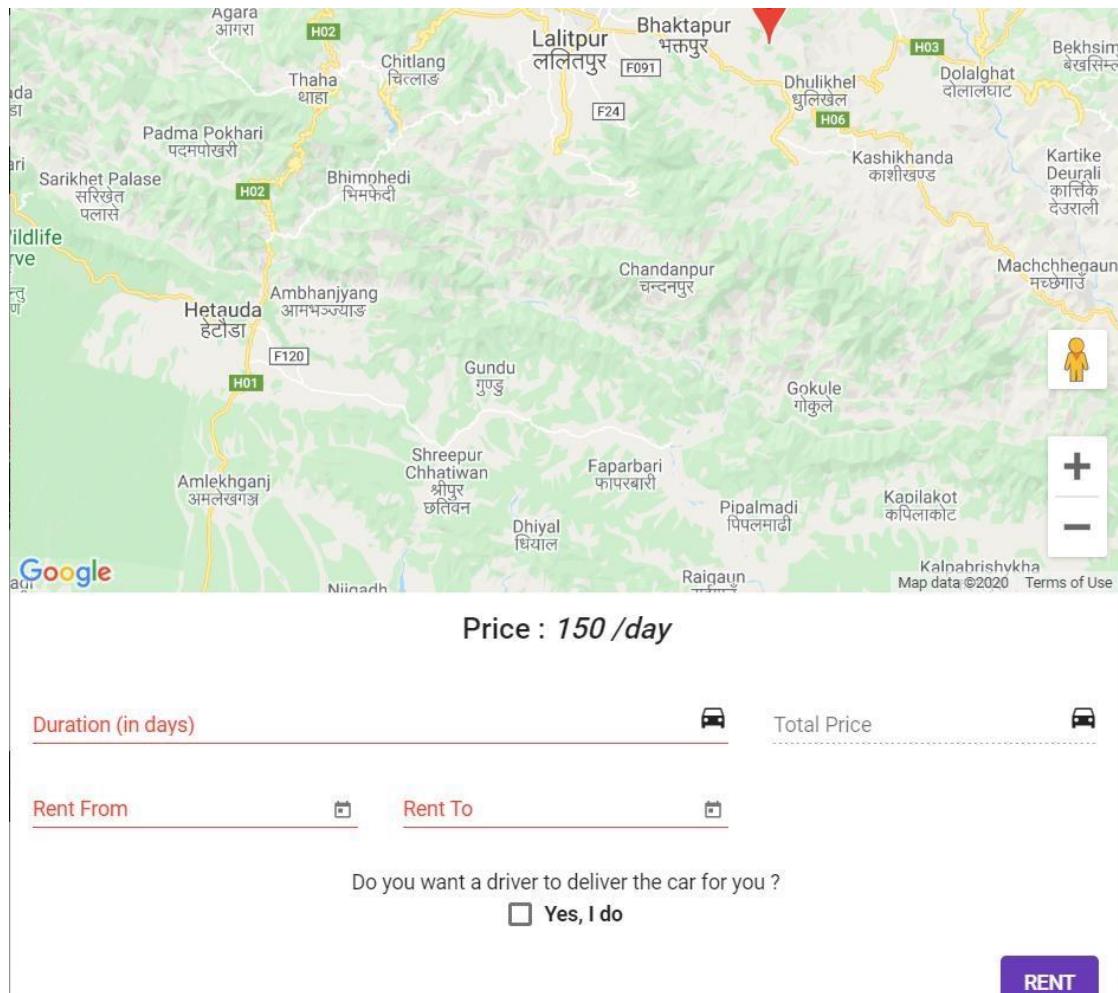


Figure 168: Displaying error for empty text field while renting vehicle

Cancelling rent and booking vehicle request

Objective	To test whether user can cancel book and rent vehicle request.
Action	Click on cancel rent and cancel book button.
Expected Result	Book and rent request will be cancelled and cancelled vehicle will be available to other users.
Actual Result	Book and rent request are cancelled and cancelled vehicle is available to other users.
Conclusion	Test Successful

Table 27: Test case for cancelling rent and book request

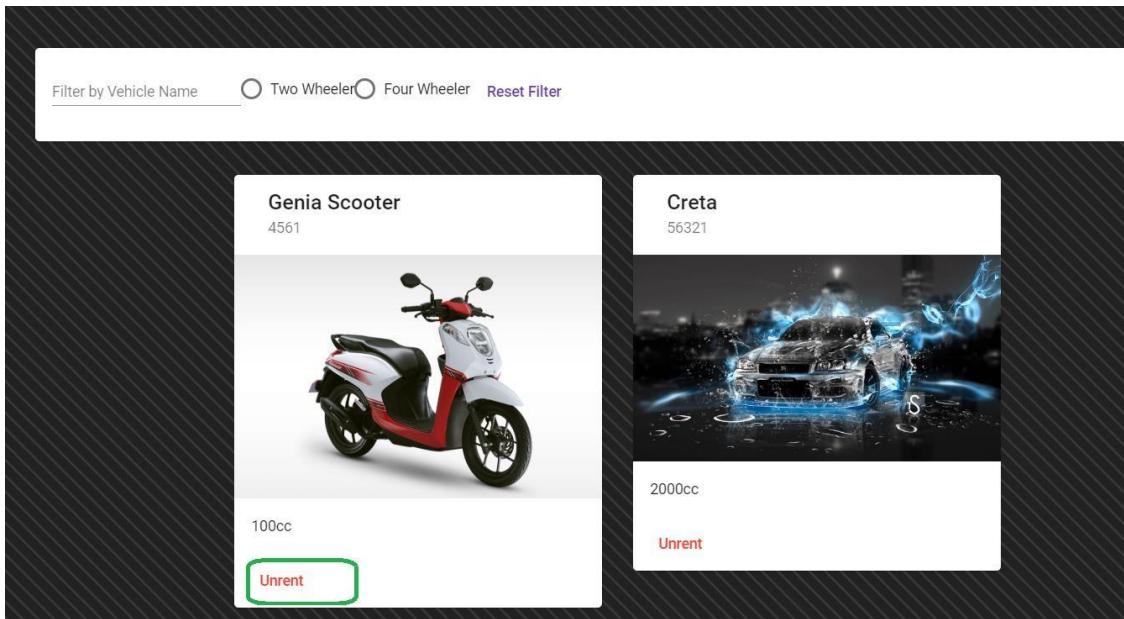


Figure 169: Before cancelling rent or book

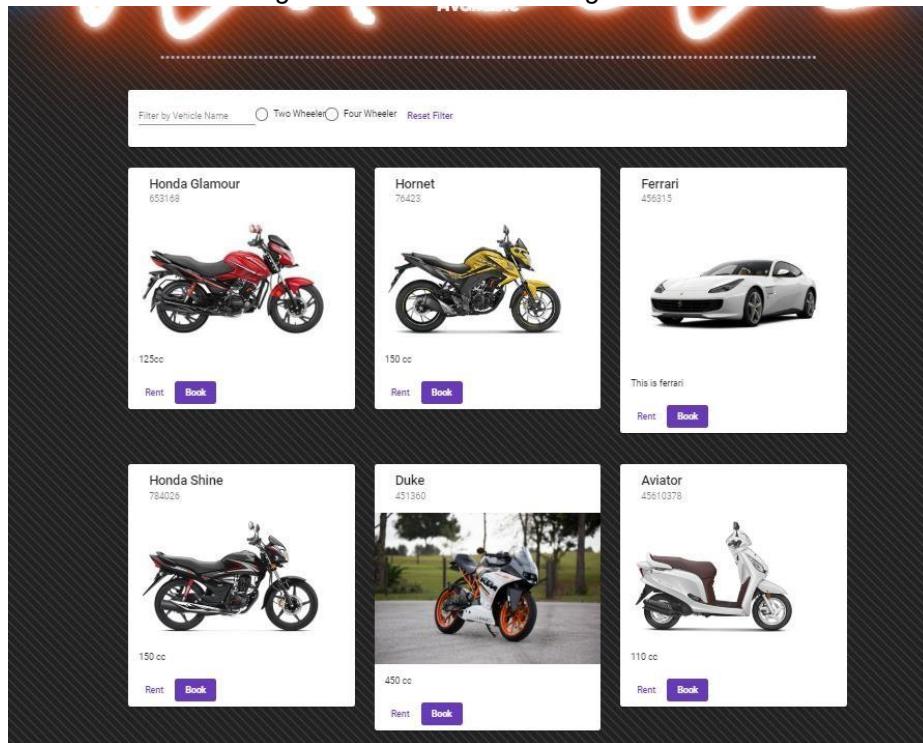


Figure 170: Available vehicles before rent or book cancel

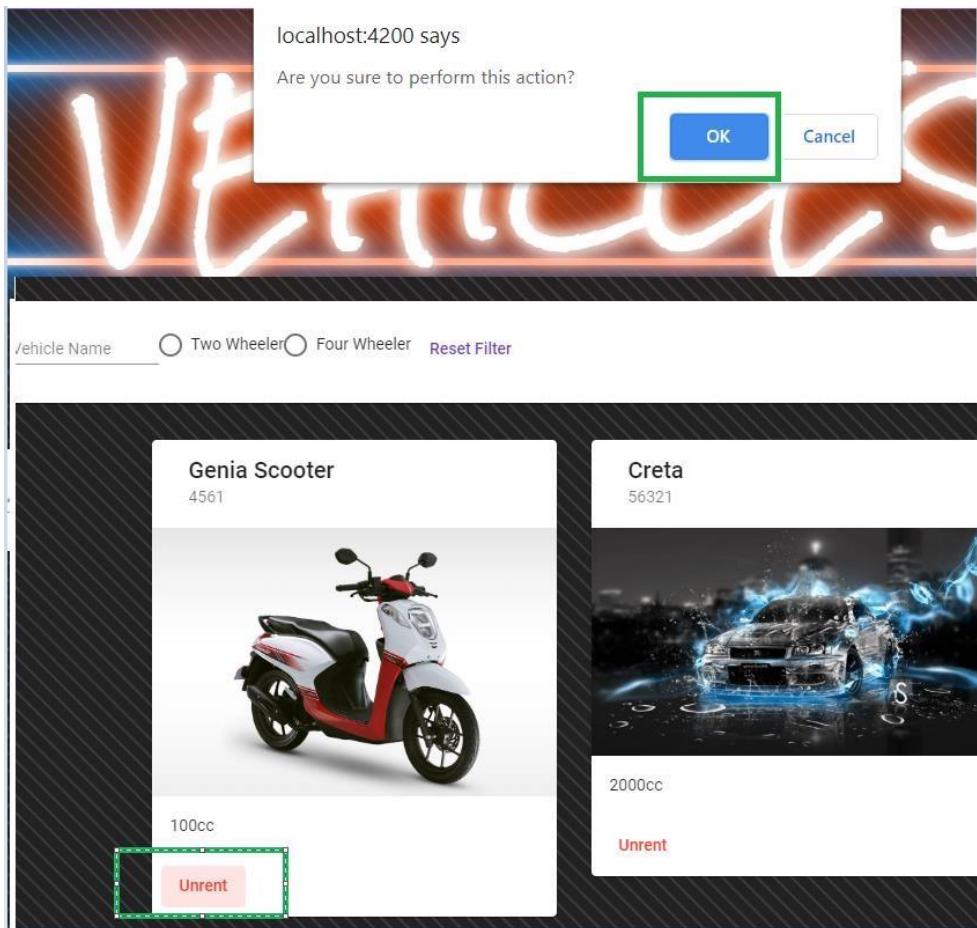


Figure 150: Cancelling rent or book request

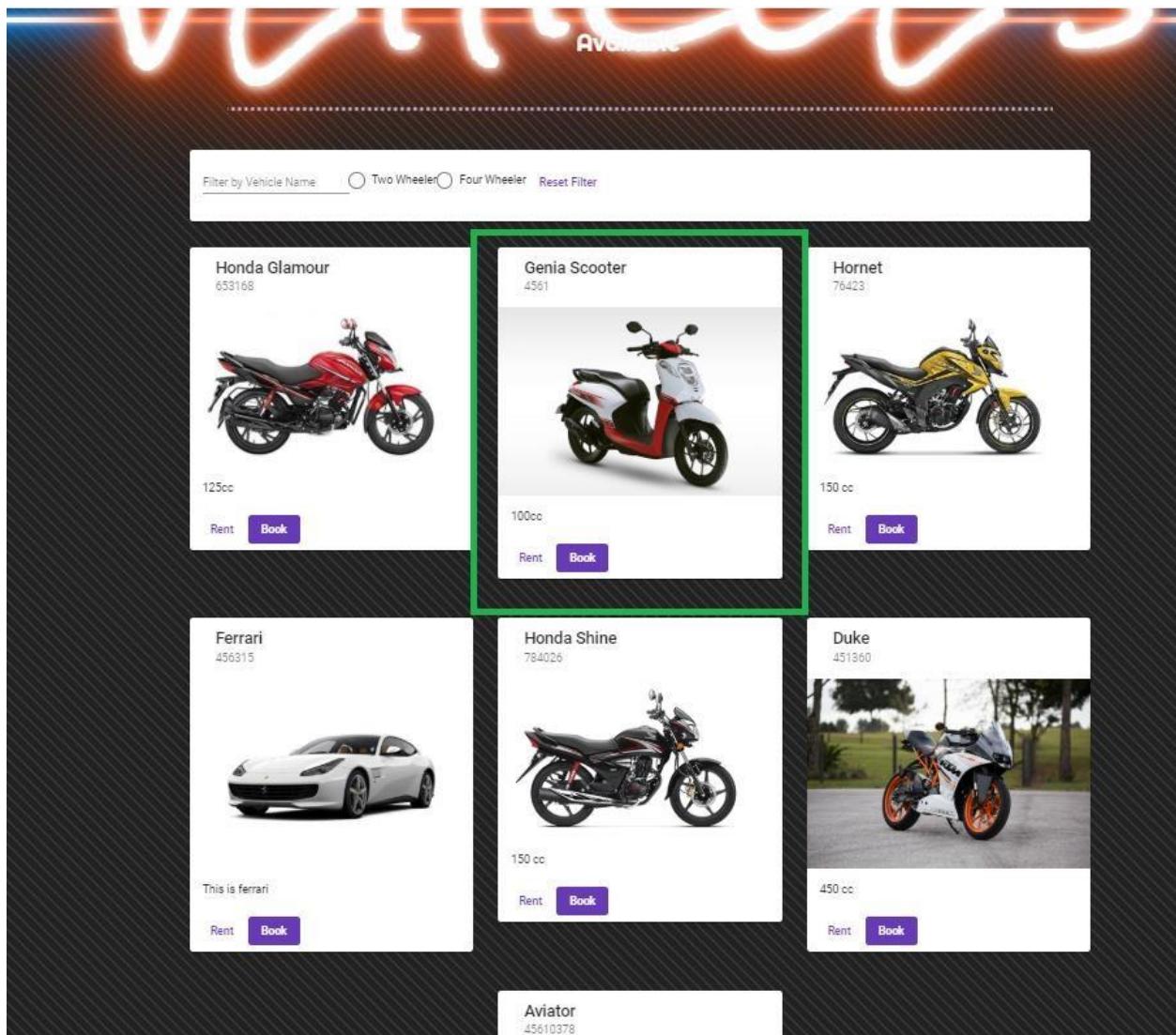
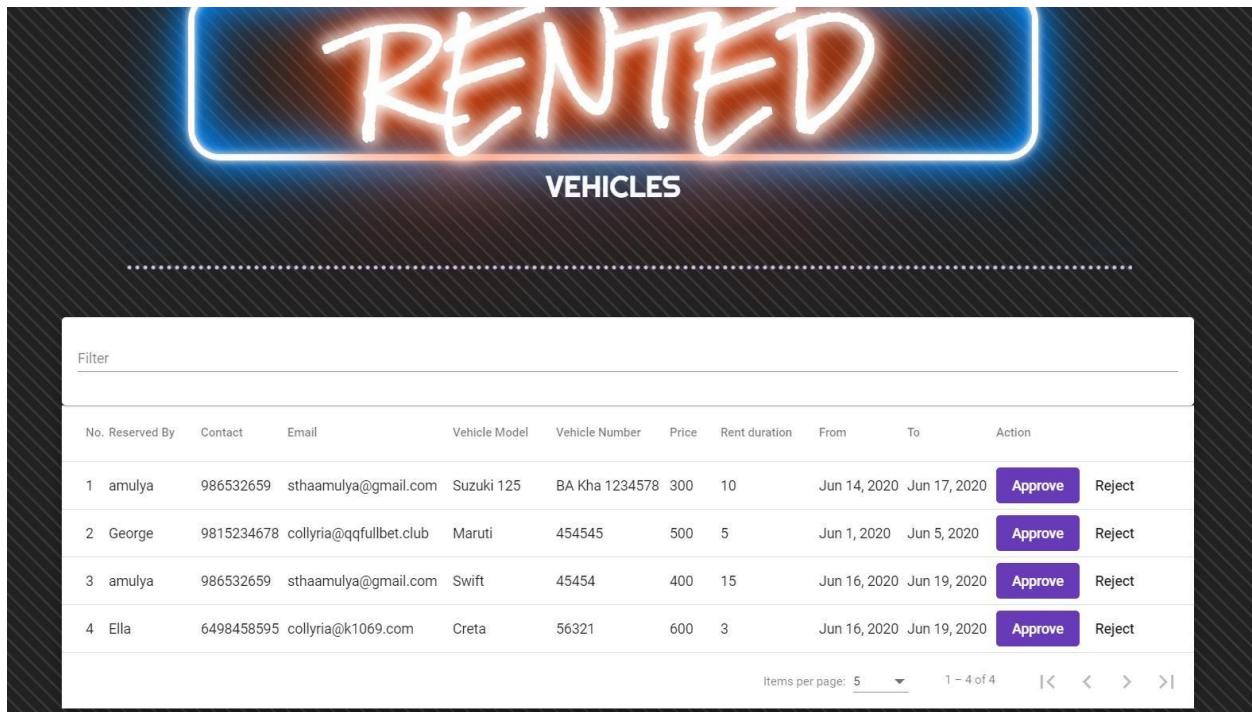
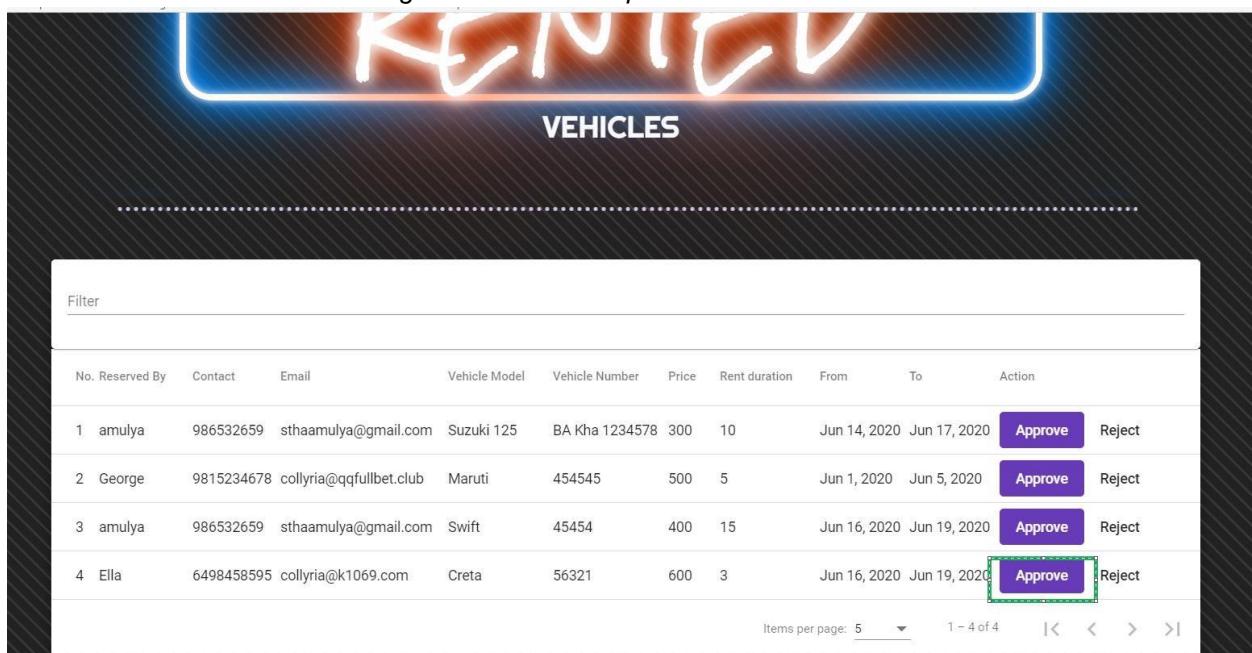


Figure 151: Available vehicles after rent or book cancel

4.4 Iteration 4 Testing Approve Rent

Objective	To approve rent request and send approval confirmation email.
Action	Click on approve button.
Expected Result	Approval confirmation email will be send to user.
Actual Result	Approval confirmation email is send to user.

Conclusion	Test Successful
-------------------	------------------------

Table 28: Test case for rent approval confirmation email*Figure 171: Rent Request from clients**Figure 172: Press Approve button to rent vehicle*

Filter										
No.	Reserved By	Contact	Email	Vehicle Model	Vehicle Number	Price	Rent duration	From	To	Action
1	amulya	986532659	sthaamulya@gmail.com	Suzuki 125	BA Kha 1234578	300	10	Jun 14, 2020	Jun 17, 2020	<button>Approve</button> <button>Reject</button>
2	George	9815234678	collyria@qqfullbet.club	Maruti	454545	500	5	Jun 1, 2020	Jun 5, 2020	<button>Approve</button> <button>Reject</button>
3	amulya	986532659	sthaamulya@gmail.com	Swift	45454	400	15	Jun 16, 2020	Jun 19, 2020	<button>Approve</button> <button>Reject</button>

Items per page: 5 | < < > > | 1 - 3 of 3

Figure 173: Request deleted after button click

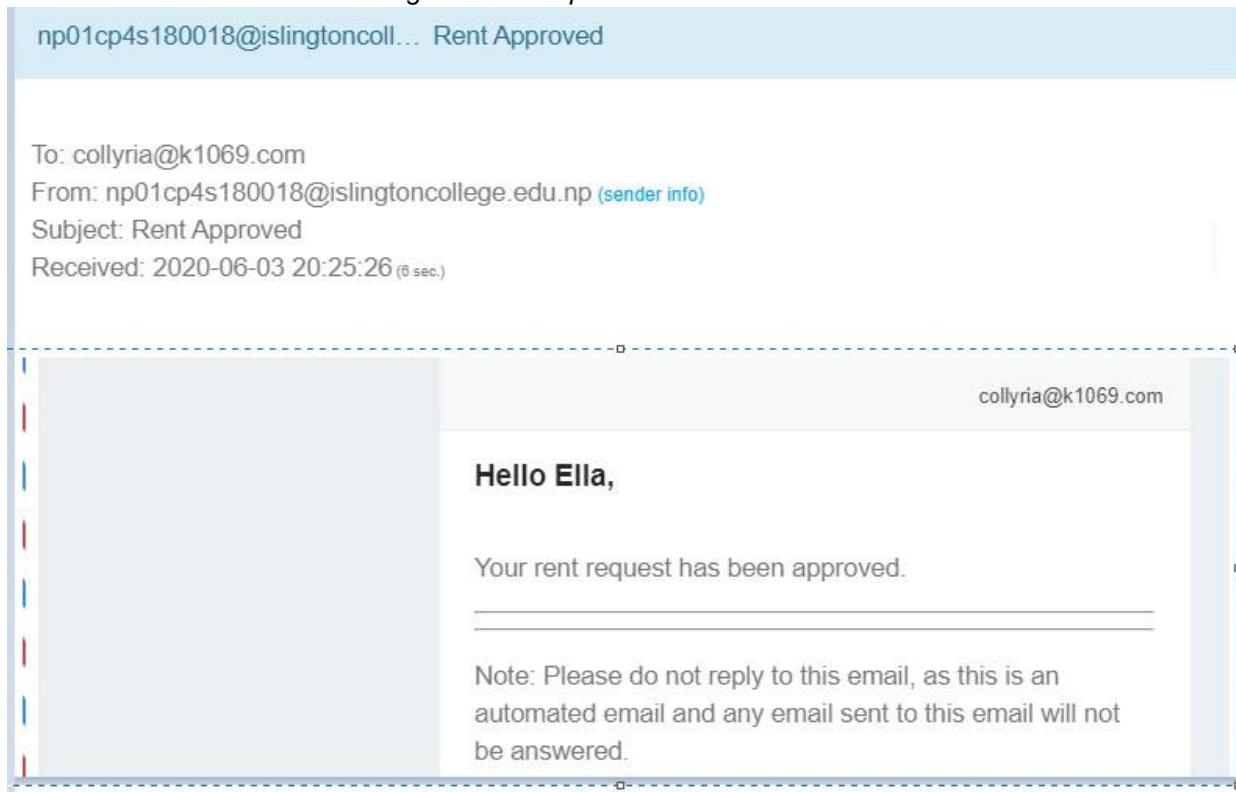


Figure 174: Displaying approval confirmation email

Reject Rent

Objective	To reject rent request and send rejection email.
Action	Click on Reject button.
Expected Result	Rejection email will be send to user and request will be deleted.
Actual Result	Rejection email is send to user and request is deleted.

Conclusion	Test Successful
-------------------	------------------------

Table 29: Test case for rent request rejected email

No.	Reserved By	Contact	Email	Vehicle Model	Vehicle Number	Price	Rent duration	From	To	Action
1	amulya	986532659	sthaamulya@gmail.com	Suzuki 125	BA Kha 1234578	300	10	Jun 14, 2020	Jun 17, 2020	Approve Reject
2	George	9815234678	collyria@qqfullbet.club	Maruti	454545	500	5	Jun 1, 2020	Jun 5, 2020	Approve Reject
3	amulya	986532659	sthaamulya@gmail.com	Swift	45454	400	15	Jun 16, 2020	Jun 19, 2020	Approve Reject

Items per page: 5 | < < > > | 1 - 3 of 3

Figure 175: Press Reject button to rent vehicle

VEHICLES										
No.	Reserved By	Contact	Email	Vehicle Model	Vehicle Number	Price	Rent duration	From	To	Action
1	amulya	986532659	sthaamulya@gmail.com	Suzuki 125	BA Kha 1234578	300	10	Jun 14, 2020	Jun 17, 2020	Approve Reject
2	amulya	986532659	sthaamulya@gmail.com	Swift	45454	400	15	Jun 16, 2020	Jun 19, 2020	Approve Reject

Items per page: 5 | < < > > | 1 - 2 of 2

Figure 176: Displaying rejection success message

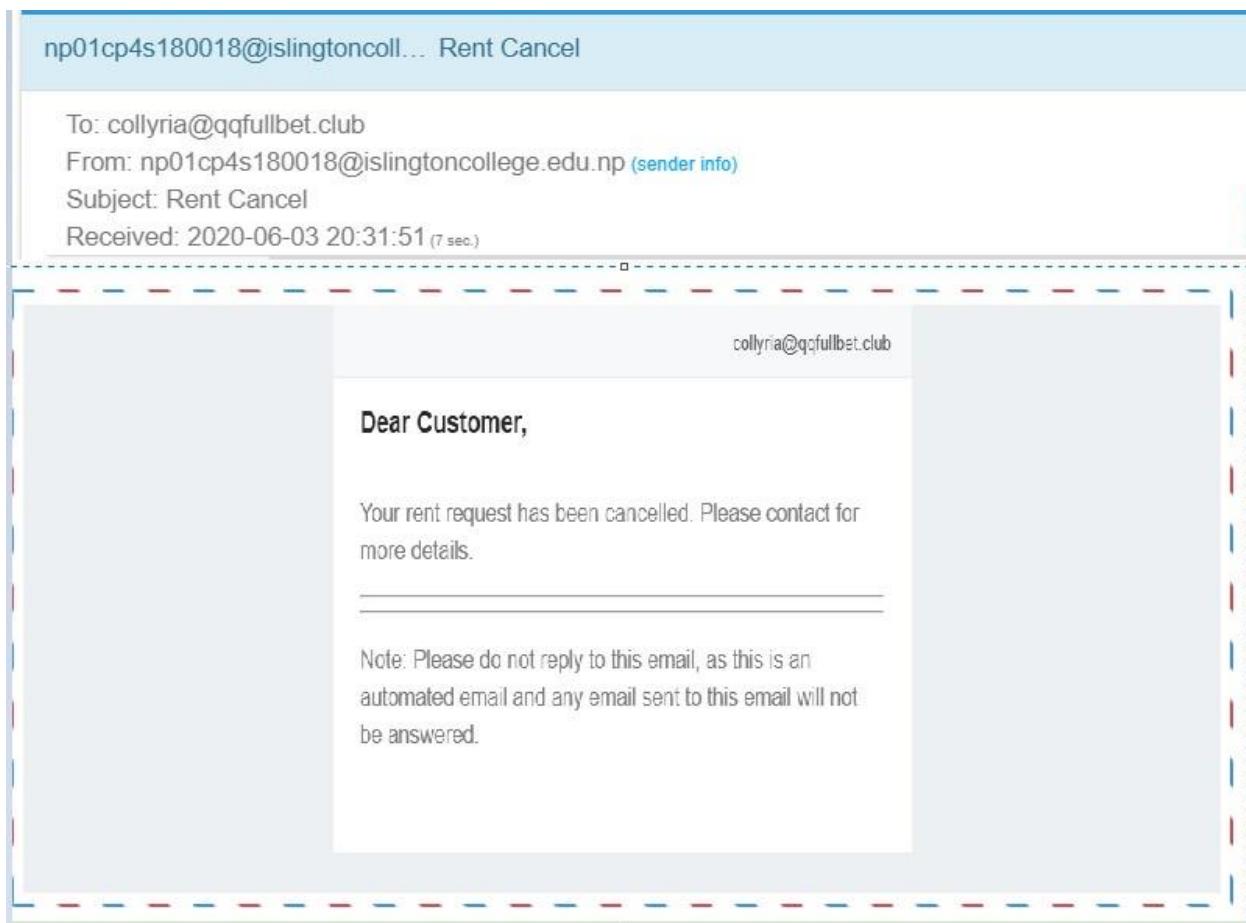


Figure 177: Displaying rejection email

View Report

Objective	To test whether user can view status of vehicles.
Action	Click on view vehicles button.
Expected Result	View report of vehicles that are on rent.
Actual Result	View reports of vehicles that are on rent.
Conclusion	Test Successful

Table 30: Test Case for viewing report of vehicle on rent

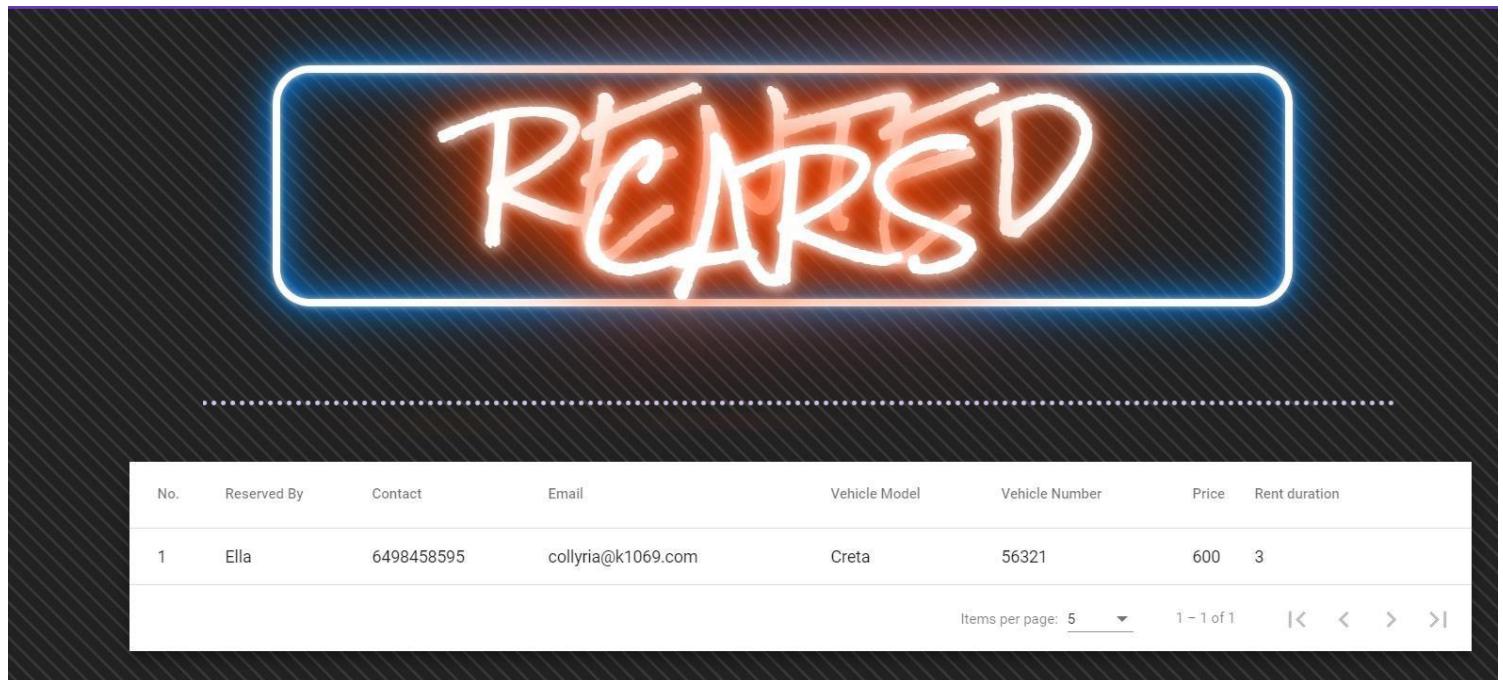


Figure 178: View vehicles that are on rent

Search requests:

Objective	To test whether admin can search by vehicle name and number.
Action	Enter by vehicle name or number.
Expected Result	View request details according to search.
Actual Result	View request details according to search.
Conclusion	Test Successful

Table 31: Test case for searching requests

Filter										
No.	Reserved By	Contact	Email	Vehicle Model	Vehicle Number	Price	Rent duration	From	To	Action
1	amulya	986532659	sthaamulya@gmail.com	Suzuki 125	BA Kha 1234578	300	10	Jun 14, 2020	Jun 17, 2020	<button>Approve</button> <button>Reject</button>
2	Cole Atkins	4138755104	collyria@rielunu.ml	Honda Glamour	653168	150	1	Jun 10, 2020	Jun 11, 2020	<button>Approve</button> <button>Reject</button>
3	amulya	986532659	sthaamulya@gmail.com	Swift	45454	400	15	Jun 16, 2020	Jun 19, 2020	<button>Approve</button> <button>Reject</button>
4	Ernest McDonald	5244584919	collyria@virlata.ga	Duke	451360	500	2	Jun 12, 2020	Jun 14, 2020	<button>Approve</button> <button>Reject</button>

Items per page: 10 | < < > > | 1 - 4 of 4

Figure 179: Enter vehicle name or number

Filter										
suzuki										
No.	Reserved By	Contact	Email	Vehicle Model	Vehicle Number	Price	Rent duration	From	To	Action
1	amulya	986532659	sthaamulya@gmail.com	Suzuki 125	BA Kha 1234578	300	10	Jun 14, 2020	Jun 17, 2020	<button>Approve</button> <button>Reject</button>

Items per page: 10 | < < > > | 1 - 1 of 1

Figure 180: Search by vehicle name

Filter										
653168										
No.	Reserved By	Contact	Email	Vehicle Model	Vehicle Number	Price	Rent duration	From	To	Action
1	Cole Atkins	4138755104	collyria@rielunu.ml	Honda Glamour	653168	150	1	Jun 10, 2020	Jun 11, 2020	<button>Approve</button> <button>Reject</button>

Items per page: 10 | < < > > | 1 - 1 of 1

Figure 181: Search by vehicle number

Sort requests:

Objective	To test whether admin can sort requests by vehicle number, price and rent duration.
Action	Click vehicle number or price or rent duration to sort.
Expected Result	View request details according to sort by.
Actual Result	View request details according to sort by.
Conclusion	Test Successful

Table 32: Test case for sorting request details

Filter										
No. Reserved By	Contact	Email	Vehicle Model	Vehicle Number	Price	Rent duration	From	To	Action	
1 amulya	986532659	sthaamulya@gmail.com	Suzuki 125	BA Kha 1234578	300	10	Jun 14, 2020	Jun 17, 2020	Approve	Reject
2 Cole Atkins	4138755104	collyria@rielunu.ml	Honda Glamour	653168	150	1	Jun 10, 2020	Jun 11, 2020	Approve	Reject
3 amulya	986532659	sthaamulya@gmail.com	Swift	45454	400	15	Jun 16, 2020	Jun 19, 2020	Approve	Reject
4 Ernest McDonald	5244584919	collyria@virlata.ga	Duke	451360	500	2	Jun 12, 2020	Jun 14, 2020	Approve	Reject

Items per page: 10 | < < > >| 1 - 4 of 4

Figure 182: Before Sorting request details

Filter										
No. Reserved By	Contact	Email	Vehicle Model	Vehicle Number ↑	Price	Rent duration	From	To	Action	
1 amulya	986532659	sthaamulya@gmail.com	Suzuki 125	BA Kha 1234578	300	10	Jun 14, 2020	Jun 17, 2020	Approve	Reject
2 amulya	986532659	sthaamulya@gmail.com	Swift	45454	400	15	Jun 16, 2020	Jun 19, 2020	Approve	Reject
3 Ernest McDonald	5244584919	collyria@virlata.ga	Duke	451360	500	2	Jun 12, 2020	Jun 14, 2020	Approve	Reject
4 Cole Atkins	4138755104	collyria@rielunu.ml	Honda Glamour	653168	150	1	Jun 10, 2020	Jun 11, 2020	Approve	Reject

Items per page: 10 | < < > >| 1 - 4 of 4

Figure 183: Sorting by Vehicle Number

No.	Reserved By	Contact	Email	Vehicle Model	Vehicle Number	Price ↑	Rent duration	From	To	Action
1	Cole Atkins	4138755104	collyria@rielunu.ml	Honda Glamour	653168	150	1	Jun 10, 2020	Jun 11, 2020	<button>Approve</button> Reject
2	amulya	986532659	sthaamulya@gmail.com	Suzuki 125	BA Kha 1234578	300	10	Jun 14, 2020	Jun 17, 2020	<button>Approve</button> Reject
3	amulya	986532659	sthaamulya@gmail.com	Swift	45454	400	15	Jun 16, 2020	Jun 19, 2020	<button>Approve</button> Reject
4	Ernest McDonald	5244584919	collyria@virlata.ga	Duke	451360	500	2	Jun 12, 2020	Jun 14, 2020	<button>Approve</button> Reject

Items per page: 10 | < < > > | 1 - 4 of 4

Figure 184: Sorting by Price

No.	Reserved By	Contact	Email	Vehicle Model	Vehicle Number	Price	Rent duration ↑	From	To	Action
1	Cole Atkins	4138755104	collyria@rielunu.ml	Honda Glamour	653168	150	1	Jun 10, 2020	Jun 11, 2020	<button>Approve</button> Reject
2	Ernest McDonald	5244584919	collyria@virlata.ga	Duke	451360	500	2	Jun 12, 2020	Jun 14, 2020	<button>Approve</button> Reject
3	amulya	986532659	sthaamulya@gmail.com	Suzuki 125	BA Kha 1234578	300	10	Jun 14, 2020	Jun 17, 2020	<button>Approve</button> Reject
4	amulya	986532659	sthaamulya@gmail.com	Swift	45454	400	15	Jun 16, 2020	Jun 19, 2020	<button>Approve</button> Reject

Items per page: 10 | < < > > | 1 - 4 of 4

Figure 185: Sorting by Rent Duration

View Feedback

Objective	To test whether user can view feedback
Action	Click on Feedback button.
Expected Result	View Feedback sent by user.
Actual Result	View Feedback sent by user.
Conclusion	Test Successful

Table 33: Test case for viewing feedback

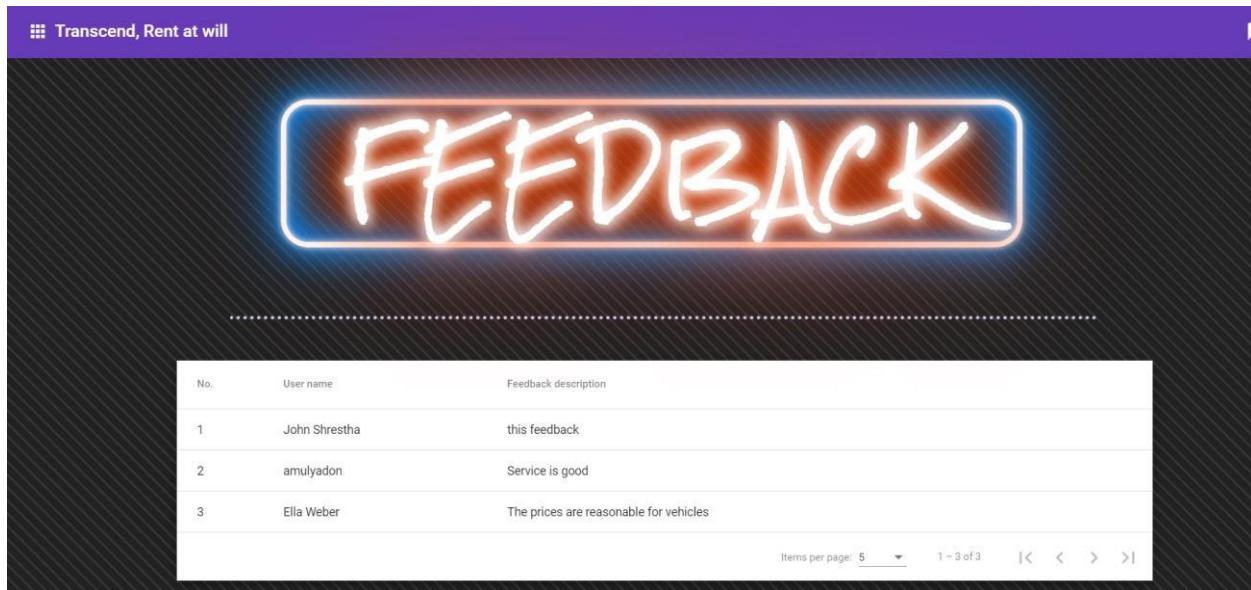


Figure 186: View feedback sent by users

4.5 Critical Analysis

All the features testing process was carried out according to the iterative incremental methodology. Each step of testing was explained with the help process elaboration through code snippets and graphical representation. The testing's are categorized according to iterations i.e. Iteration 1 testing, Iteration 2 testing, Iteration 3 testing and Iteration 4 testing. Hence all the testing process are successful and the application operates smoothly.

Chapter 5: Conclusion

This project is about replacing traditional way of booking and renting vehicles to systematic online booking and renting. This is final year project also one the biggest project I have done so far. This project gave me an opportunity to learn new framework and do more research.

The project was developed using Angular framework for front end and spring boot framework for backend in IntelliJ IDE. Iterative incremental methodology was used to develop the application which helped me to gather all the requirements as well as the iterative nature helped be to repeat iteration making it easier to develop the project. Also all the requirements were gathered through survey and from the client and implemented properly.

This project was developed after a lot of research. Since this project was developed for the client on real case scenario the requirements were provided directly by the client and features were added as per client recommendation. Overall with completion of this project I was able to gain knowledge about JavaScript and Java programming language also I gain experience in working with framework which will be very useful for my career and future projects. At the end of the project I was more familiar and gained more experience working with these language and frameworks.

5.1 Legal, Social and Ethical issues

5.1.1 Legal issues

Legal issues are issues faced after violation of Acts or rules and regulation .Some the legal issues faced by application nowadays are related to intellectual property ownership, intellectual property protection like trademarks, copyrights, patents etc., terms of use and privacy policies. (App Empire, 2018)

The Vehicle Rental System application is simple online booking and renting vehicle. Thus application hasn't violated any of these type of legal issues while developing the application. This application maintains the confidentiality and privacy of data and is legally accepted.

5.1.2 Social issues

This application targets the Nepalese audience and the Nepalese market as the number of people using internet has grown and people are going digital. In social point of view the app has positive side considering the number of people using internet and online services. Since the app is useful for both client as well as vehicle rental business owners. This app can benefit not only people but also government by increasing the economy of the country. Thus this project and whole system does not have negative impact to the society and is accessible by all people. (LinkedIn Corporation, 2020)

5.1.3 Ethical issues

One the ethical issues related to vehicle rental system is leaking of the personal details of the customers and misuse of the details as this web application is a server based application so personal details are stores in database. Privacy of customer and their details will be maintained. Personal details will not be accessible by everyone and will not be misused. Also some of the other ethical issues are hidden fare, not well maintained vehicles, ownership of copyright, license agreement and intellectual property. This application tries to be transparent as much as possible so there is no chance of additional or hidden fare as the prices and charges of the vehicles are calculated by the application and the maintenance of the vehicles are done time to time and are well checked before giving it on rent. Any of these ethical issues were not violated while developing this application. (Simple Programmer, 2018)

5.2 Advantages

Some of the advantages of this application are as follows:

- This application is fully functional and flexible and easy to use.
- It saves a lot of time, money and labor as most of the process is online.
- Use of this application helps to manage the customers as well as offers quality services to the customer.

- It helps company to provide it services without paper work.
- This application acts as office providing service 24/7 to its customer.
- This application can be run on any platforms as well as codes can be reused.
- Custom features can be added further or this application can be customized according to the requirements of the company.

5.3 Limitations

Since every application has its own features and limitations. Some of the limitations for this project are as follows:

- If client/user is new they must register themselves before logging in.
- Valid email should be entered to register otherwise the system will give error message.
- In order to perform the rent and book vehicle capability customer must login with valid credentials.
- Vehicles can go on rent only after the admin approves the requests made by client.

5.4 Future work

The functions and features of the project is all completed as per client recommendation and requirements made by client. However, this application can be improved further and new features can be added to application. Some of the features that can be added to enhance this application are as follows:

- Development of mobile application.
- Notification for both admin and client side.
- Online payment integration.
- Implementing Chabot's.

Chapter 6: References

AirBrake, 2020. *AirBrake*. [Online]

Available at: <https://airbrake.io/blog/sdlc/waterfall-model> [Accessed 6 April 2020].

App Empire, 2018. *App Empire*. [Online]

Available at: <https://appempire.com/top-5-legal-issues-facing-app-developers/> [Accessed 15 March 2020].

GeeksforGeeks, 2020. *GeeksforGeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/software-engineering-prototyping-model/> [Accessed 4 1 2020].

GeeksforGeeks, 2020. *GeeksforGeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/introduction-to-spring-boot/> [Accessed 5 1 2020].

GlobeNewswire, oct 14,2019. *Car Rental Market Predicted to Attain a Size of \$122.6 Billion By 2024: P&S Intelligence*, s.l.: P&S Intelligence.

guru99, 2020. *guru99*. [Online]

Available at: <https://www.guru99.com/software-engineering-prototyping-model.html> [Accessed 6 April 2020].

Hotwire, Inc., an Expedia Group company, 2020. *Hotwire*. [Online]

Available at: <https://www.hotwire.com/car-rentals/results/ORD/ORD/2020-01-06T10%253A00%253A00/2020-01-07T10%253A00%253A00> [Accessed 1 1 2020].

java t point, 2018. *java t point*. [Online]

Available at: <https://www.javatpoint.com/angular-7-introduction> [Accessed 4 1 2020].

Java T Point, 2020. *Java T Point*. [Online]

Available at: <https://www.javatpoint.com/software-engineering-prototype-model> [Accessed 6 April 2020].

JournalDev, 2020. *JournalDev*. [Online]

Available at: <https://www.journaldev.com/32975/what-is-java-programming-language> [Accessed 4 1 2020].

LinkedIn Corporation, 2020. *Slide Share*. [Online]

Available at: <https://www.slideshare.net/andresbaravalle/im2044-week-11-lecture> [Accessed 12 May 2020].

LinkedIn Corporation, 2020. *Slideshare*. [Online]

Available at: <https://www.slideshare.net/kronat/8-architetture-software-architecture-centric-processes> [Accessed 6 April 2020].

Rent Rabbit, 2019. *Rent Rabbit*. [Online] Available at: <https://www.rentrabbit.io/> [Accessed 1 1 2020].

searchcio, 2020. *searchcio*. [Online] Available at: <https://searchcio.techtarget.com/definition/Prototyping-Model#:~:text=The%20prototyping%20model%20is%20a,or%20product%20can%20be%20developed.> [Accessed 6 April 2020].

Simple Programmer, 2018. *Simple Programmer*. [Online] Available at: <https://simpleprogrammer.com/ethical-issues-software-engineering/> [Accessed 12 May 2020].

SiteGround, 2019. *SiteGround*. [Online] Available at: <https://www.siteground.com/tutorials/php-mysql/mysql/> [Accessed 5 1 2020].

sixt, 2020. *sixt*. [Online] Available at: <https://www.sixt.de/> [Accessed 1 1 2020].

Techopedia, 2020. *Techopedia*. [Online] Available at: <https://www.techopedia.com/definition/25895/iterative-and-incremental-development> [Accessed 6 April 2020].

Transports in Nepal, 2020. *Transports in Nepal*. [Online] Available at: <http://www.transportsinnepal.com> [Accessed 1 1 2020].

Try QA, 2020. *Try QA*. [Online] Available at: <http://tryqa.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/> [Accessed 6 April 2020].

Tutorialspoint, 2020. *Tutorialspoint*. [Online] Available at: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm [Accessed 2 1 2020].

Tutorialspoint, 2020. *Tutorialspoint*. [Online] Available at: https://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm [Accessed 4 1 2020].

UKEssays, 2020. *UKEssays*. [Online] Available at: <https://www.ukessays.com/essays/computer-science/waterfall-methodology-in-softwaredevelopment.php> [Accessed 6 April 2020].

Vehicle Hire Nepal, 2018. *Vehicle Hire Nepal*. [Online] Available at: <http://vehiclehirenepal.com/> [Accessed 1 1 2020].

Chapter 7: Bibliography

AirBrake, 2020. *AirBrake*. [Online]

Available at: <https://airbrake.io/blog/sdlc/waterfall-model> [Accessed 6 April 2020].

App Empire, 2018. *App Empire*. [Online]

Available at: <https://appempire.com/top-5-legal-issues-facing-app-developers/> [Accessed 15 March 2020].

GeeksforGeeks, 2020. *GeeksforGeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/software-engineering-prototyping-model/> [Accessed 4 1 2020].

GeeksforGeeks, 2020. *GeeksforGeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/introduction-to-spring-boot/> [Accessed 5 1 2020].

GlobeNewswire, oct 14,2019. *Car Rental Market Predicted to Attain a Size of \$122.6 Billion By 2024: P&S Intelligence*, s.l.: P&S Intelligence.

guru99, 2020. *guru99*. [Online]

Available at: <https://www.guru99.com/software-engineering-prototyping-model.html> [Accessed 6 April 2020].

Hotwire, Inc., an Expedia Group company, 2020. *Hotwire*. [Online]

Available at: <https://www.hotwire.com/car-rentals/results/ORD/ORD/2020-01-06T10%253A00%253A00/2020-01-07T10%253A00%253A00> [Accessed 1 1 2020].

java t point, 2018. *java t point*. [Online]

Available at: <https://www.javatpoint.com/angular-7-introduction> [Accessed 4 1 2020].

Java T Point, 2020. *Java T Point*. [Online]

Available at: <https://www.javatpoint.com/software-engineering-prototype-model> [Accessed 6 April 2020].

JournalDev, 2020. *JournalDev*. [Online]

Available at: <https://www.journaldev.com/32975/what-is-java-programming-language> [Accessed 4 1 2020].

LinkedIn Corporation, 2020. *Slide Share*. [Online]

Available at: <https://www.slideshare.net/andresbaravalle/im2044-week-11-lecture> [Accessed 12 May 2020].

LinkedIn Corporation, 2020. *Slideshare*. [Online]

Available at: <https://www.slideshare.net/kronat/8-architetture-software-architecture-centric-processes> [Accessed 6 April 2020].

Rent Rabbit, 2019. *Rent Rabbit*. [Online] Available at: <https://www.rentrabbit.io/> [Accessed 11 2020].

searchcio, 2020. *searchcio*. [Online] Available at: <https://searchcio.techtarget.com/definition/Prototyping-Model#:~:text=The%20prototyping%20model%20is%20a,or%20product%20can%20be%20developed.> [Accessed 6 April 2020].

Simple Programmer, 2018. *Simple Programmer*. [Online] Available at: <https://simpleprogrammer.com/ethical-issues-software-engineering/> [Accessed 12 May 2020].

SiteGround, 2019. *SiteGround*. [Online] Available at: <https://www.siteground.com/tutorials/php-mysql/mysql/> [Accessed 5 1 2020].

sixt, 2020. *sixt*. [Online] Available at: <https://www.sixt.de/> [Accessed 11 2020].

Techopedia, 2020. *Techopedia*. [Online] Available at: <https://www.techopedia.com/definition/25895/iterative-and-incremental-development> [Accessed 6 April 2020].

Transports in Nepal, 2020. *Transports in Nepal*. [Online] Available at: <http://www.transportsinnepal.com> [Accessed 11 2020].

Try QA, 2020. *Try QA*. [Online] Available at: <http://tryqa.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/> [Accessed 6 April 2020].

Tutorialspoint, 2020. *Tutorialspoint*. [Online] Available at: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm [Accessed 2 1 2020].

Tutorialspoint, 2020. *Tutorialspoint*. [Online] Available at: https://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm [Accessed 4 1 2020].

UKEssays, 2020. *UKEssays*. [Online] Available at: <https://www.ukessays.com/essays/computer-science/waterfall-methodology-in-softwaredevelopment.php> [Accessed 6 April 2020].

Vehicle Hire Nepal, 2018. *Vehicle Hire Nepal*. [Online]
Available at: <http://vehicleshirenepal.com/>
[Accessed 1 1 2020].

Chapter 8: Appendix

8.1 Appendix A: Pre-survey

8.1.1 Pre-survey Form

<p>Vehicle Rental Nepal</p> <p>Please, take some time to fill this form. Vehicle Rental is a web application that will allow people to hire vehicles of their choices giving people chance to travel with comfort even without owning a vehicle.</p> <p>*Required</p>		<p>If yes, which way?</p> <p><input type="radio"/> Online <input type="radio"/> Direct visit to office <input type="radio"/> Through phone call</p>
<p>Email address *</p> <p>Your email address</p>		<p>On the scale of 1-10 do you prefer to hire your own personal vehicle online? *</p> <p>1 2 3 4 5 6 7 8 9 10</p> <p><input type="radio"/> <input type="radio"/></p>
<p>Do you or your family have a two/four wheeler vehicle? *</p> <p><input type="radio"/> Yes <input type="radio"/> No</p>		<p>When planning for solo trip or with friends/ family . I am more likely to: *</p> <p><input type="radio"/> Use a public trasportation <input type="radio"/> Use a taxi service <input type="radio"/> Hire vehicle <input type="radio"/> Have a hotel pickup</p>
<p>Have you heard about online vehicle renting application? *</p> <p><input type="radio"/> Yes <input type="radio"/> No</p>		<p>If you have any suggestion/idea for application ,please feel free to send feedback.</p> <p>Your answer</p>
<p>Do you/your friends/colleagues/family often hire vehicles? *</p> <p><input type="radio"/> Yes <input type="radio"/> No</p>		

Figure 187: Pre-Survey Form

8.1.2 Sample of filled Pre-survey forms

0 of 0 points Score released 1 Jan 21:05 [Release score](#)

Vehicle Rental Nepal

Please, take some time to fill this form.
Vehicle Rental is a web application that will allow people to hire vehicles of their choices giving people chance to travel with comfort even without owning a vehicle.
*Required

Email address *
imichchha830@gmail.com

Do you or your family have a two/four wheeler vehicle? * / 0

Yes
 No

Add individual feedback

Do you have a driving license (local/international)? * / 0

Yes
 No

Add individual feedback

Have you heard about online vehicle renting application? * / 0

Yes
 No

Add individual feedback

Do you/your friends/colleagues/family often hire vehicles? * / 0

Yes
 No

Add individual feedback

If yes, which way? * / 0

Online
 Direct visit to office
 Through phone call

Add individual feedback

On the scale of 1-10 do you prefer to hire your own personal vehicle online? * / 0

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>						

Add individual feedback

When planning for solo trip or with friends/ family , I am more likely to: * / 0

Use a public trasportation
 Use a taxi service
 Hire vehicle
 Have a hotel pickup

Add individual feedback

If you have any suggestion/idea for application ,please feel free to send feedback. / 0

Nice

Add individual feedback

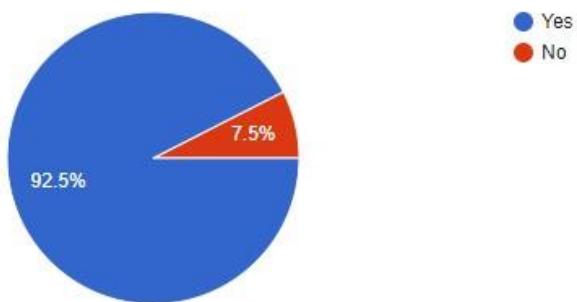
Figure 188: Sample of filled Pre-Survey form

8.1.3 Pre-survey result

Do you or your family have a two/four wheeler vehicle?



40 responses



Do you have a driving license (local/international)?

40 responses

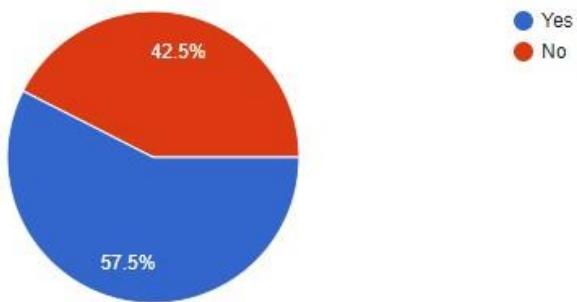
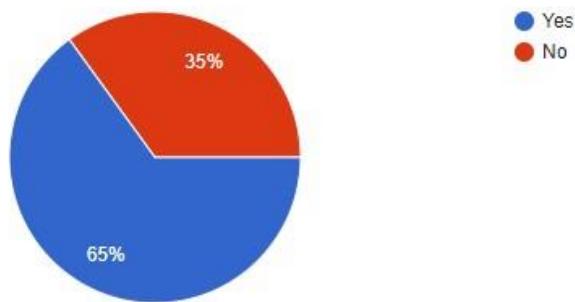


Figure 189: Pre-survey results 1 & 2

Have you heard about online vehicle renting application?

40 responses



Do you/your friends/colleagues/family often hire vehicles?

40 responses

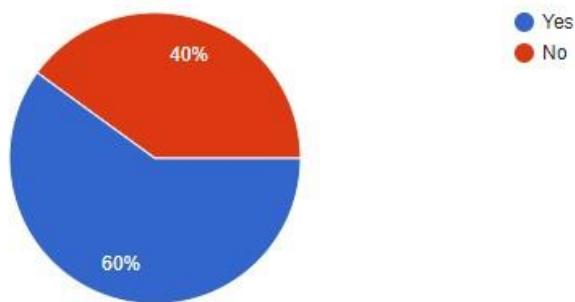
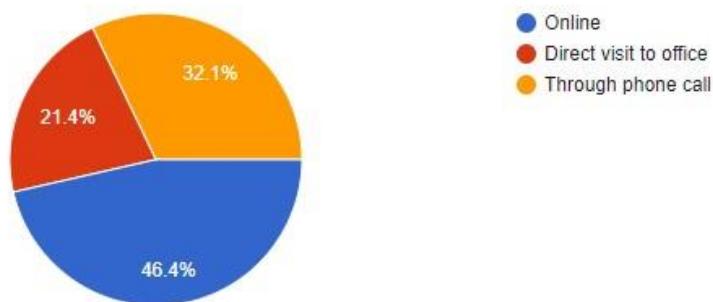


Figure 190: Pre-survey results 3 & 4

If yes, which way?

28 responses



On the scale of 1-10 do you prefer to hire your own personal vehicle online?

40 responses

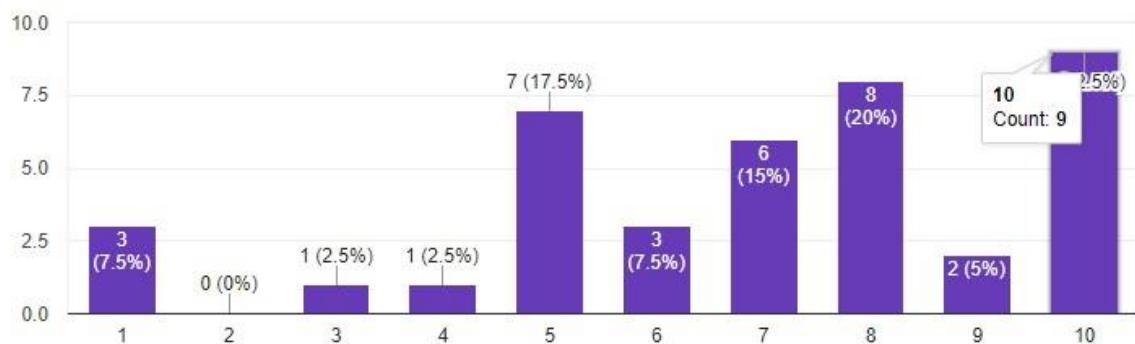


Figure 191: Pre-survey results 5 & 6



Figure 192: Pre-survey results 7 & 8

8.2 Appendix B: Post-survey

8.2.1 Post- Survey Form

Vehicle Rental Nepal

Please, take some time to fill this form.
Vehicle Rental is a web application that will allow people to hire vehicles of their choices giving people chance to travel with comfort even without owning a vehicle.

*Required

*Required

Email address *

Your email address _____

Do you or your family have a two/four wheeler vehicle? *

Yes

No

Do you have a driving license (local/international)? *

Yes

No

Have you heard about online vehicle renting application? *

Yes

No

When planning for solo trip or with friends/ family . I am more likely to: *

- Use a public trasportation
- Use a taxi service
- Hire vehicle
- Have a hotel pickup

Have you used online vehicle hiring services before?

- Yes
- No

If yes, rate their services on the scale of 1-10? *

1 2 3 4 5 6 7 8 9 10

How do you prefer to hire your vehicle online? *

- With driver
- Drive on your own

What kind of features do you want in online vehicle renting application? *

- Calculate total price
- Approval through Email
- Advance booking
- Online payment

Any other features you would like to add to this application?

Your answer _____

If you have any suggestion/idea for application ,please feel free to send feedback.

Your answer _____

Send me a copy of my responses.

Submit

Figure 193: Post- survey form

8.2.2 Sample of filled Post-survey forms

Responses cannot be edited

Vehicle Rental Nepal

Please, take some time to fill this form.
Vehicle Rental is a web application that will allow people to hire vehicles of their choices giving people chance to travel with comfort even without owning a vehicle.
*Required

*Required

Email address *

np01cp4s180012@islingtoncollege.edu.np

Do you or your family have a two/four wheeler vehicle? *

Yes
 No

Do you have a driving license (local/international)? *

Yes
 No

Have you heard about online vehicle renting application? *

Yes
 No

When planning for solo trip or with friends/ family . I am more likely to: *

Use a public trasportation
 Use a taxi service
 Hire vehicle
 Have a hotel pickup

Have you used online vehicle hiring services before?

Yes
 No

If yes, rate their services on the scale of 1-10? *

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>						

How do you prefer to hire your vehicle online? *

With driver
 Drive on your own

What kind of features do you want in online vehicle renting application? *

Calculate total price
 Approval through Email
 Advance booking
 Online payment

Any other features you would like to add to this application?

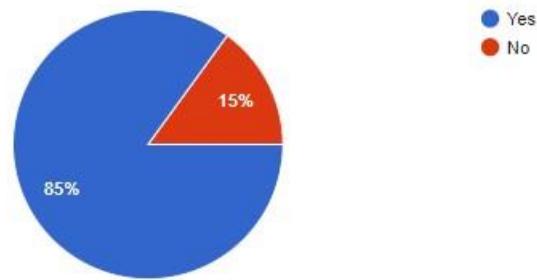
If you have any suggestion/idea for application ,please feel free to send feedback.

Figure 194: Sample of filled Post-Survey form

8.2.3 Post-survey result

Do you or your family have a two/four wheeler vehicle?

40 responses



Do you have a driving license (local/international)?

40 responses

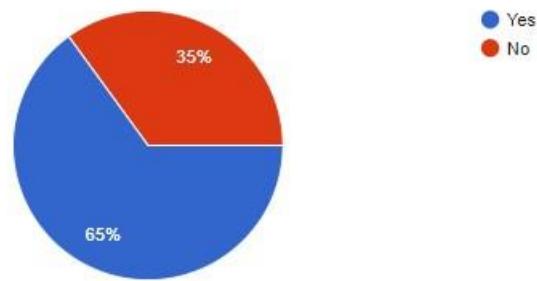
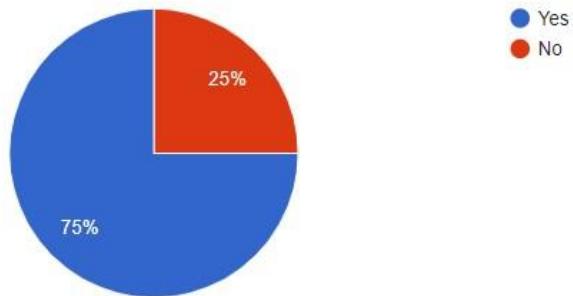


Figure 195: Post-Survey result 1 & 2

Have you heard about online vehicle renting application?

40 responses



When planning for solo trip or with friends/ family . I am more likely to:

40 responses

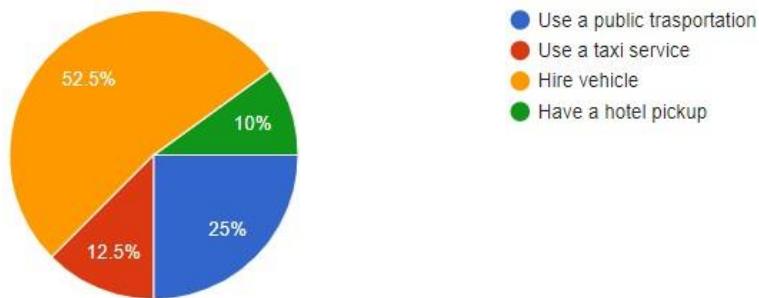
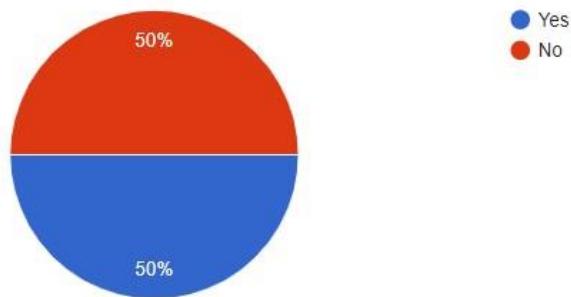


Figure 196: Post-Survey results of 3 & 4

Have you used online vehicle hiring services before?

40 responses



If yes, rate their services on the scale of 1-10?

40 responses

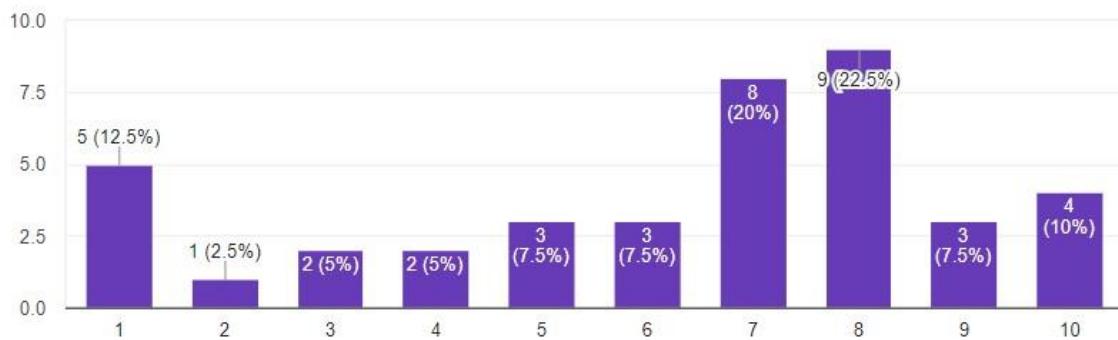
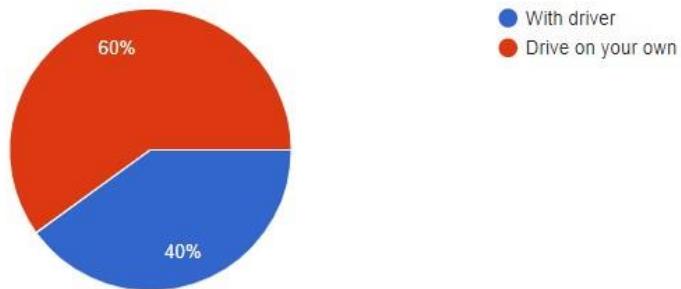


Figure 197: Post-Survey results of 5 & 6

How do you prefer to hire your vehicle online?

40 responses



What kind of features do you want in online vehicle renting application?

40 responses

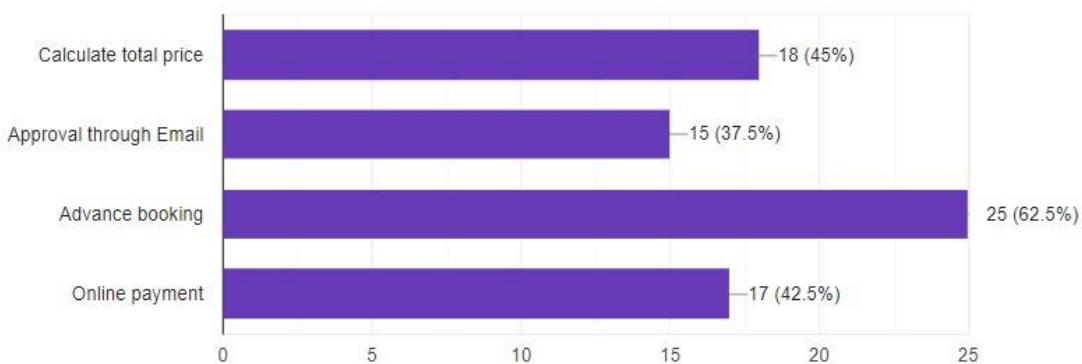


Figure 198: Post-Survey results of 7 & 8

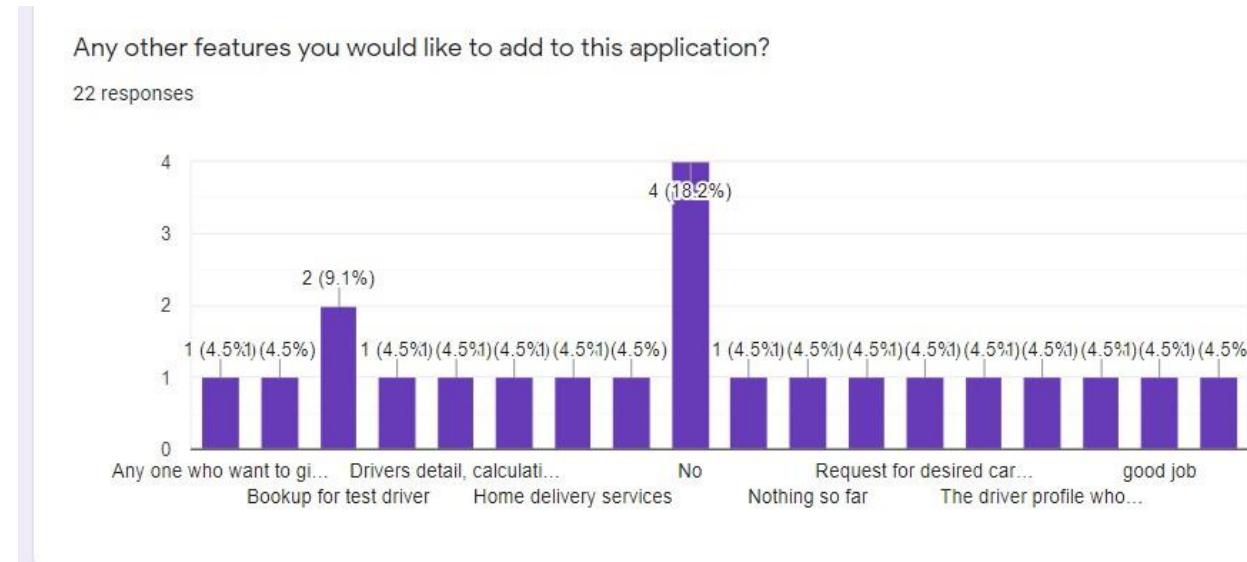


Figure 199: Post-Survey results of 9

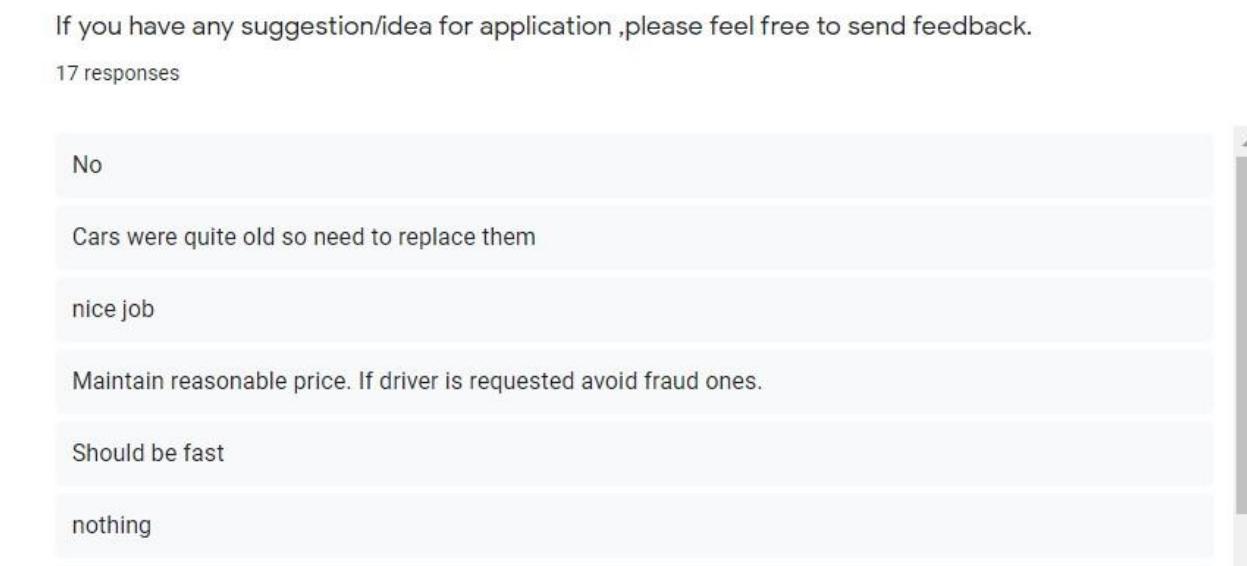


Figure 200: Post-Survey results of 10

8.3 Appendix C: Sample Codes

8.3.1 Sample Code of the UI

```
1 <div class="carbon-background w-100">
2   <div class="row flex-nwarp justify-content-center w-100 bg-transparent">
3     <div class="col-md-7 mt-5 text-center shaky">
4       <h1 class="blog-header-logo text-center monoton glow">Vehicles</h1>
5       <span class="righteous text-light" style="font-size: 30px">{[action]}</span>
6     </div>
7   </div>
8   <div class="container bg-transparent">
9     <div class="row my-5 justify-content-center">
10       <div class="col-md-11 pt-3">
11         <mat-progress-bar mode="buffer"></mat-progress-bar>
12       </div>
13       <div class="col-md-1" *ngIf="admin" (click)="openVehicleActionModel( vehicle: null)">
14         <button mat-mini-fab aria-label="Example icon-button with a heart icon">
15           <mat-icon>library_add</mat-icon>
16         </button>
17       </div>
18     </div>
19   <mat-card>
20     <mat-form-field>
21       <mat-label>Filter by Vehicle Name</mat-label>
22       <input #vName matInput (change)="searchFromVehicleName(vName.value)" placeholder="Filter keyword">
23     </mat-form-field>
24     <mat-radio-group aria-label="Select an option">
25       <mat-radio-button #tw (click)="searchByType(tw.value)" value="two-wheeler">Two Wheeler</mat-radio-button>
26       <mat-radio-button #fw (click)="searchByType(fw.value)" value="four-wheeler">Four Wheeler</mat-radio-button>
27     </mat-radio-group>
28     <b><button mat-button color="primary" (click)="ngOnInit()">Reset Filter</button></b>
29   </mat-card>
30   <div class="row mt-2 justify-content-center">
31     <div class="col-md-4 my-4" *ngFor="let vehicle of vehicleList">
32       <mat-card class="example-card">
33         <mat-card-header>
34           <img alt="Placeholder image for vehicle card" data-bbox="109 116 875 880" style="width: 100%; height: 100%; object-fit: cover;"/>
35         <mat-card-content>
36           <div>Vehicle Details</div>
37           <div>Vehicle ID: {{ vehicle.id }}</div>
38           <div>Vehicle Type: {{ vehicle.type }}</div>
39           <div>Vehicle Model: {{ vehicle.model }}</div>
40           <div>Vehicle Color: {{ vehicle.color }}</div>
41           <div>Vehicle Price: {{ vehicle.price }}</div>
42         </mat-card-content>
43       </mat-card>
44     </div>
45   </div>
46 </div>
```

Figure 201: code of UI vehicle available

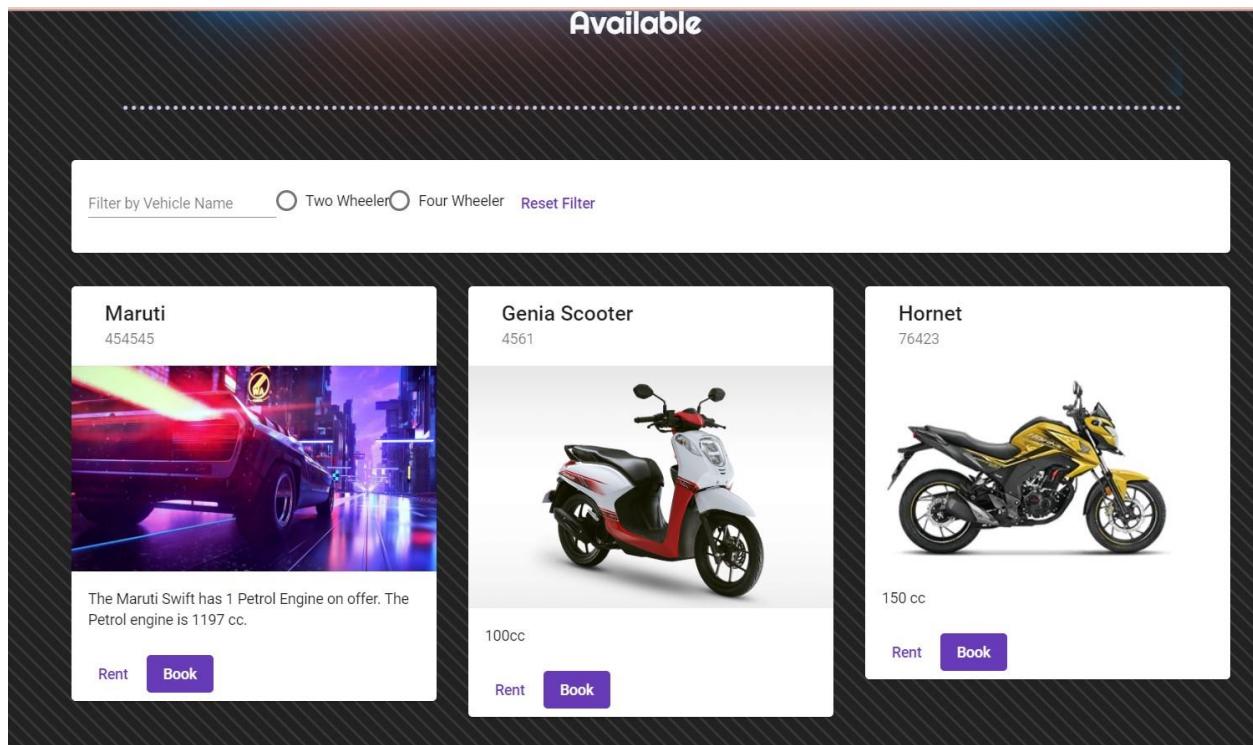
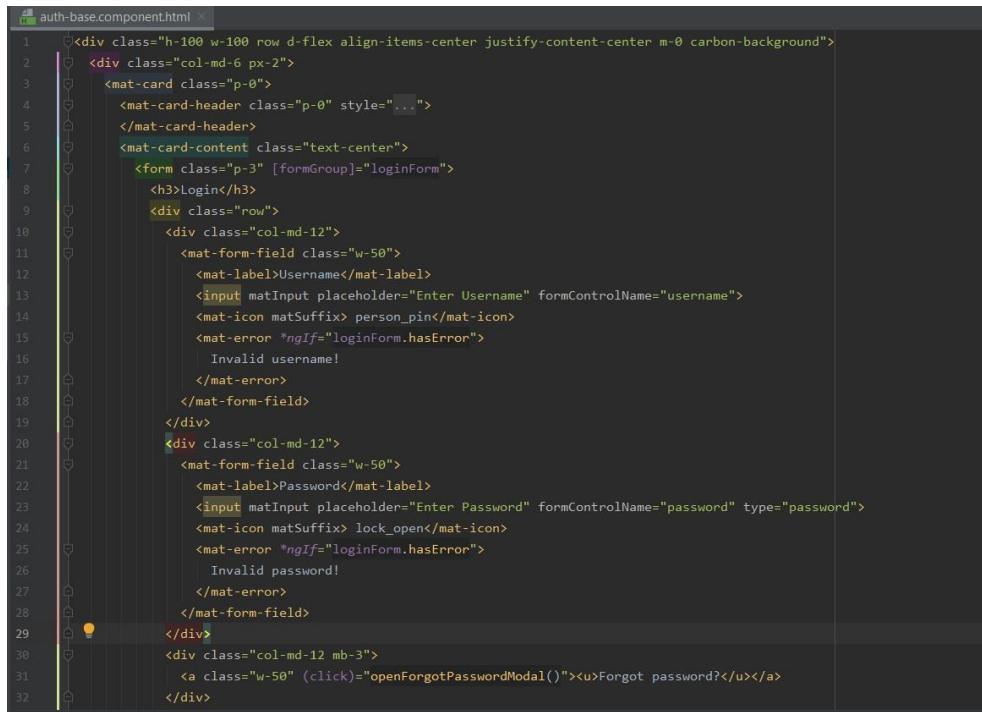


Figure 202: Vehicle Available layout



```
auth-base.component.html
1 <div class="h-100 w-100 row d-flex align-items-center justify-content-center m-0 carbon-background">
2   <div class="col-md-6 px-2">
3     <mat-card class="p-0">
4       <mat-card-header class="p-0" style="... ">
5       </mat-card-header>
6       <mat-card-content class="text-center">
7         <form class="p-3" [formGroup]="loginForm">
8           <h3>Login</h3>
9           <div class="row">
10             <div class="col-md-12">
11               <mat-form-field class="w-50">
12                 <mat-label>Username</mat-label>
13                 <input matInput placeholder="Enter Username" formControlName="username">
14                 <mat-icon matSuffix> person_pin</mat-icon>
15                 <mat-error *ngIf="loginForm.hasError">
16                   Invalid username!
17                 </mat-error>
18               </mat-form-field>
19             </div>
20             <div class="col-md-12">
21               <mat-form-field class="w-50">
22                 <mat-label>Password</mat-label>
23                 <input matInput placeholder="Enter Password" formControlName="password" type="password">
24                 <mat-icon matSuffix> lock_open</mat-icon>
25                 <mat-error *ngIf="loginForm.hasError">
26                   Invalid password!
27                 </mat-error>
28               </mat-form-field>
29             </div>
30             <div class="col-md-12 mb-3">
31               <a class="w-50" (click)="openForgotPasswordModal()"><u>Forgot password?</u></a>
32             </div>

```

Figure 203: code of UI login

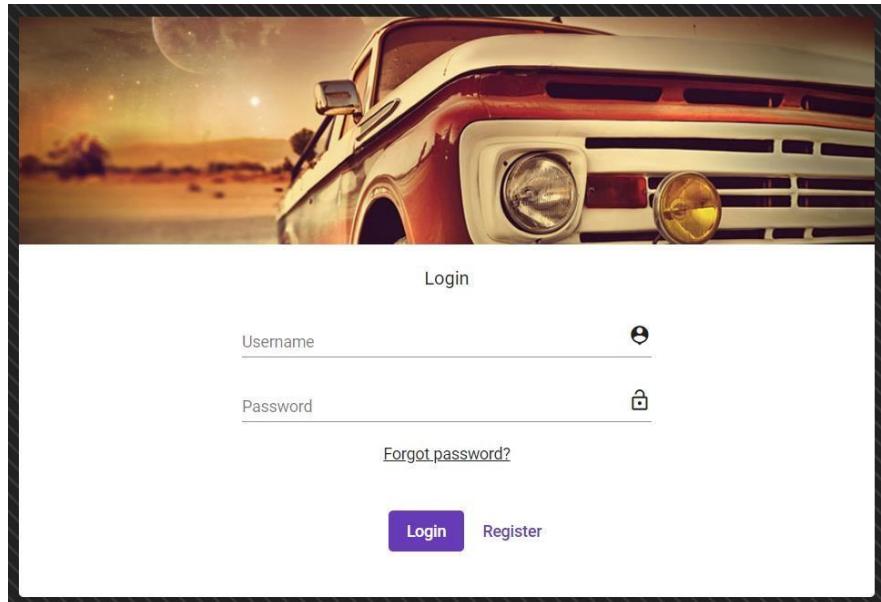
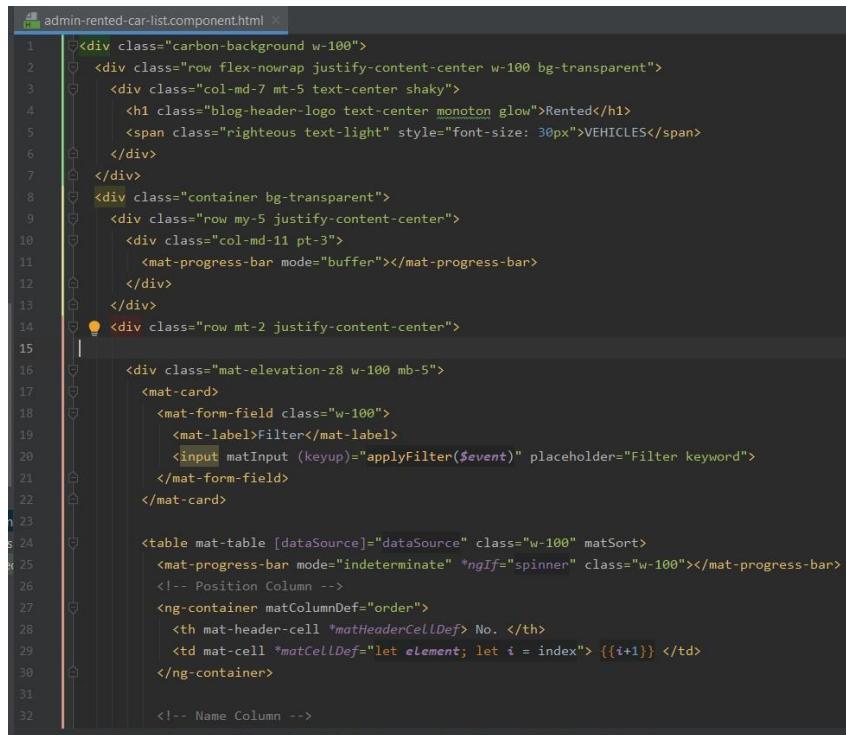


Figure 204: Login Form layout

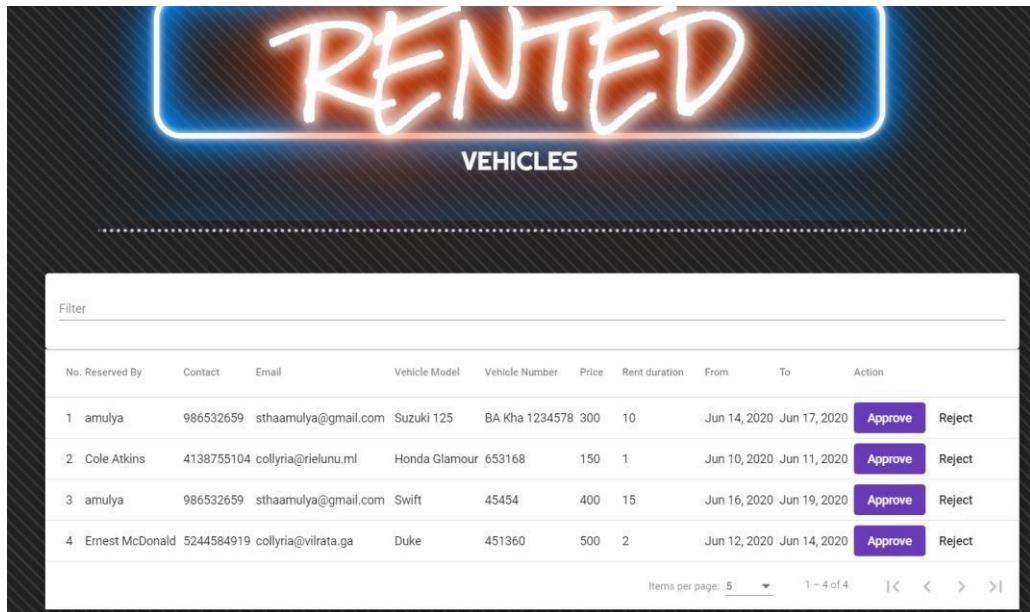


```

1 <div class="carbon-background w-100">
2   <div class="row flexnowrap justify-content-center w-100 bg-transparent">
3     <div class="col-md-7 mt-5 text-center shaky">
4       <h1 class="blog-header-logo text-center monoton glow">Rented</h1>
5       <span class="righteous text-light" style="font-size: 30px">VEHICLES</span>
6     </div>
7   </div>
8   <div class="container bg-transparent">
9     <div class="row my-5 justify-content-center">
10      <div class="col-md-11 pt-3">
11        <mat-progress-bar mode="buffer"></mat-progress-bar>
12      </div>
13    </div>
14    <div class="row mt-2 justify-content-center">
15      <div class="mat-elevation-z8 w-100 mb-5">
16        <mat-card>
17          <mat-form-field class="w-100">
18            <mat-label>Filter</mat-label>
19            <input matInput (keyup)="applyFilter($event)" placeholder="Filter keyword">
20          </mat-form-field>
21        </mat-card>
22
23      <table mat-table [dataSource]="dataSource" class="w-100" matSort>
24        <mat-progress-bar mode="indeterminate" *ngIf="spinner" class="w-100"></mat-progress-bar>
25        <!-- Position Column -->
26        <ng-container matColumnDef="order">
27          <th mat-header-cell *matHeaderCellDef> No. </th>
28          <td mat-cell *matCellDef="let element; let i = index"> {{i+1}} </td>
29        </ng-container>
30
31        <!-- Name Column -->
32      </table>

```

Figure 205: code of UI vehicles reserved by client



No.	Reserved By	Contact	Email	Vehicle Model	Vehicle Number	Price	Rent duration	From	To	Action
1	amulya	986532659	sthaamulya@gmail.com	Suzuki 125	BA Kha 1234578	300	10	Jun 14, 2020	Jun 17, 2020	<button>Approve</button> <button>Reject</button>
2	Cole Atkins	4138755104	collyria@rielunu.ml	Honda Glamour	653168	150	1	Jun 10, 2020	Jun 11, 2020	<button>Approve</button> <button>Reject</button>
3	amulya	986532659	sthaamulya@gmail.com	Swift	45454	400	15	Jun 16, 2020	Jun 19, 2020	<button>Approve</button> <button>Reject</button>
4	Ernest McDonald	5244584919	collyria@virlata.ga	Duke	451360	500	2	Jun 12, 2020	Jun 14, 2020	<button>Approve</button> <button>Reject</button>

Items per page: 5 | < < > >| 1 - 4 of 4

Figure 206: Vehicle Reserved by client layout

```

19      <mat-label>Vehicle number</mat-label>
20      <input formControlName="vehicleNumber" matInput placeholder="Enter Vehicle number">
21      <mat-icon matSuffix> directions_car</mat-icon>
22    </mat-form-field>
23  </div>
24  <div class="col-md-6">
25    <mat-form-field class="w-100">
26      <mat-label>Price</mat-label>
27      <input formControlName="price" matInput placeholder="Enter Price">
28      <mat-icon matSuffix> directions_car</mat-icon>
29    </mat-form-field>
30  </div>
31  <div class="col-md-6">
32    <mat-form-field class="w-100">
33      <mat-label>Type</mat-label>
34      <input formControlName="type" matInput placeholder="Enter Type">
35      <mat-icon matSuffix> directions_car</mat-icon>
36    </mat-form-field>
37  </div>
38  <div class="col-md-12">
39    <mat-form-field class="w-100">
40      <mat-label>Description</mat-label>
41      <input formControlName="description" matInput placeholder="Enter Description">
42      <mat-icon matSuffix> directions_car</mat-icon>
43    </mat-form-field>
44  </div>
45 </div>
46 <div class="col-md-12 text-right mt-3">
47   <button *ngIf="!forEdit" color="primary" mat-flat-button type="submit">Add</button>
48   <button *ngIf="forEdit" color="primary" mat-flat-button type="submit">Update</button>
49 </div>
50 </div>

```

Figure 207: code of UI vehicle action

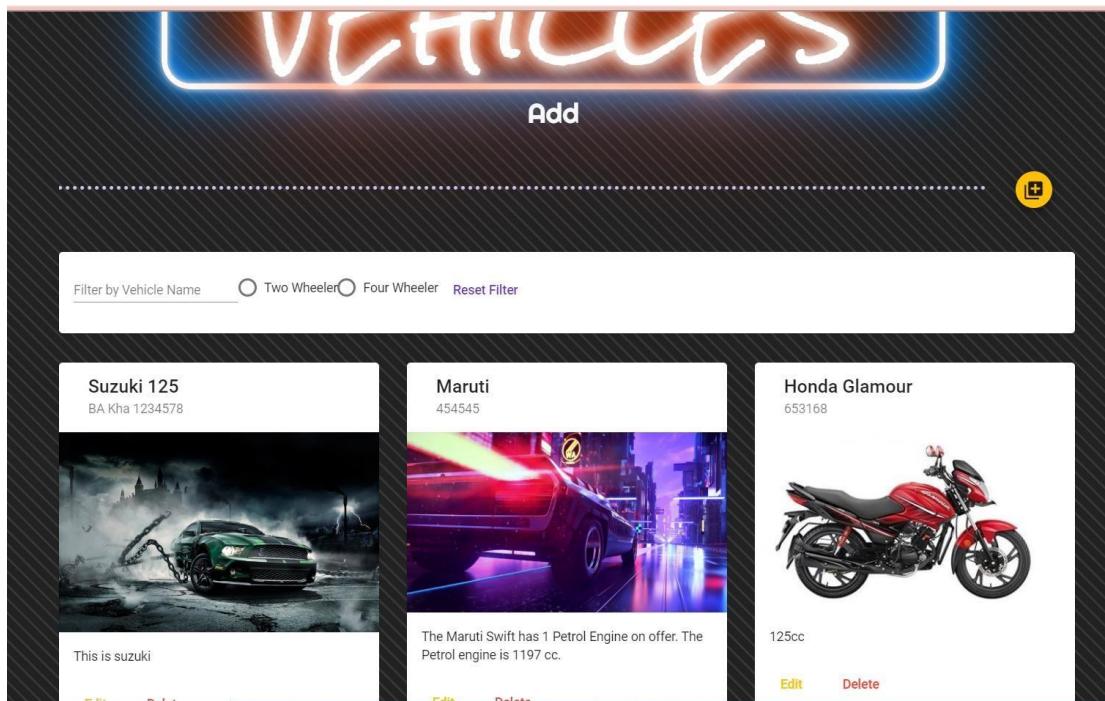
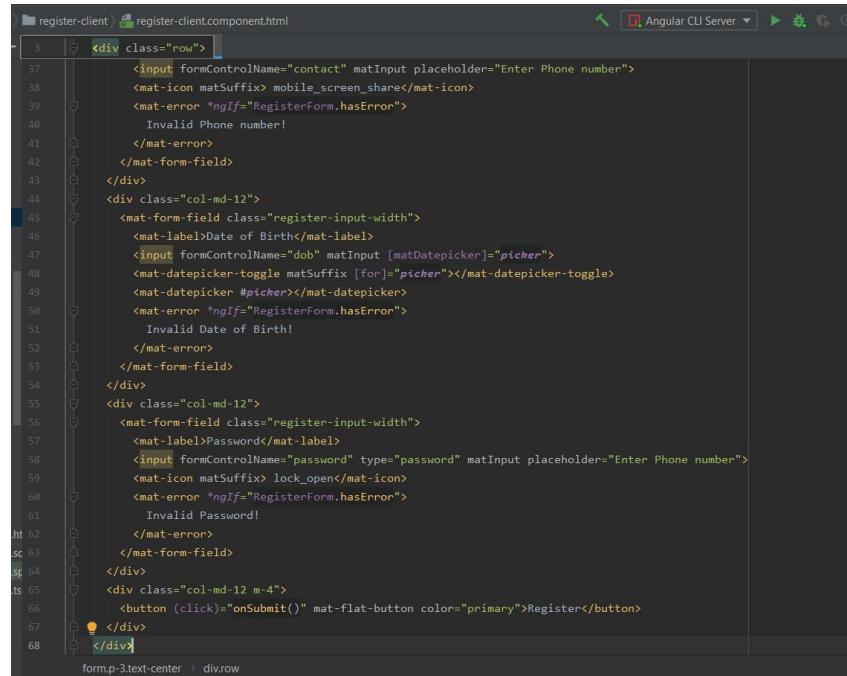


Figure 208: Vehicle action layout

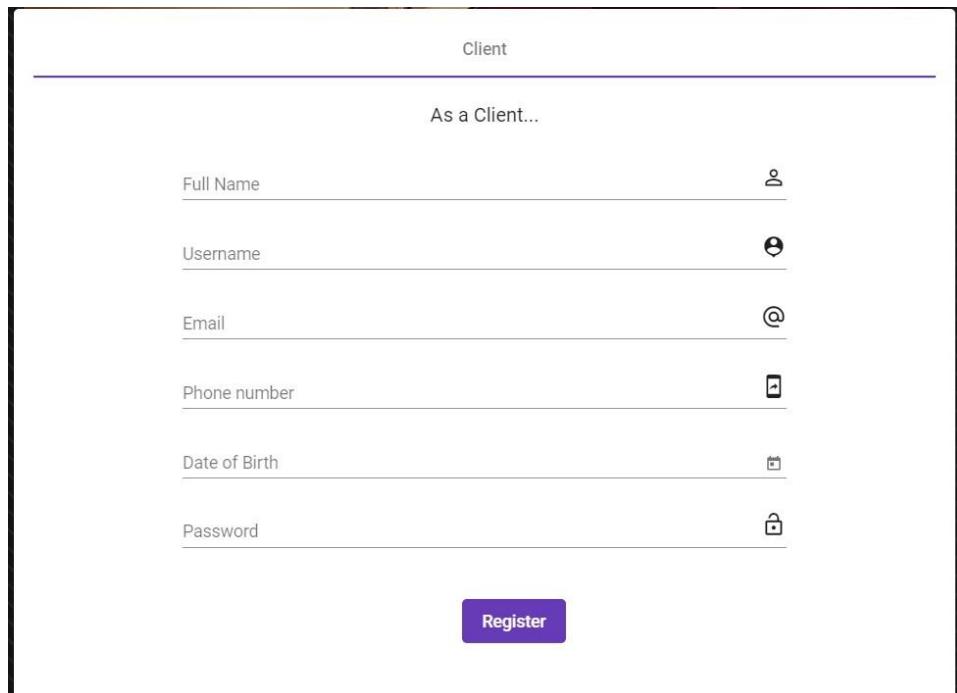


```

<div class="row">
  <div class="col-md-12">
    <mat-form-field class="register-input-width">
      <mat-label>Enter Phone number</mat-label>
      <input formControlName="contact" matInput placeholder="Enter Phone number">
      <mat-icon matSuffix> mobile_screen_share</mat-icon>
      <mat-error *ngIf="RegisterForm.hasError">
        Invalid Phone number!
      </mat-error>
    </mat-form-field>
  </div>
  <div class="col-md-12">
    <mat-form-field class="register-input-width">
      <mat-label>Date of Birth</mat-label>
      <input formControlName="dob" matInput [matDatepicker]="picker">
      <mat-datepicker-toggle matSuffix [for]="picker"></mat-datepicker-toggle>
      <mat-datepicker #picker></mat-datepicker>
      <mat-error *ngIf="RegisterForm.hasError">
        Invalid Date of Birth!
      </mat-error>
    </mat-form-field>
  </div>
  <div class="col-md-12">
    <mat-form-field class="register-input-width">
      <mat-label>Password</mat-label>
      <input formControlName="password" type="password" matInput placeholder="Enter Phone number">
      <mat-icon matSuffix> lock_open</mat-icon>
      <mat-error *ngIf="RegisterForm.hasError">
        Invalid Password!
      </mat-error>
    </mat-form-field>
  </div>
  <div class="col-md-12 m-4">
    <button (click)="onSubmit()" mat-flat-button color="primary">Register</button>
  </div>
</div>

```

Figure 209: code of UI register user



Client

As a Client...

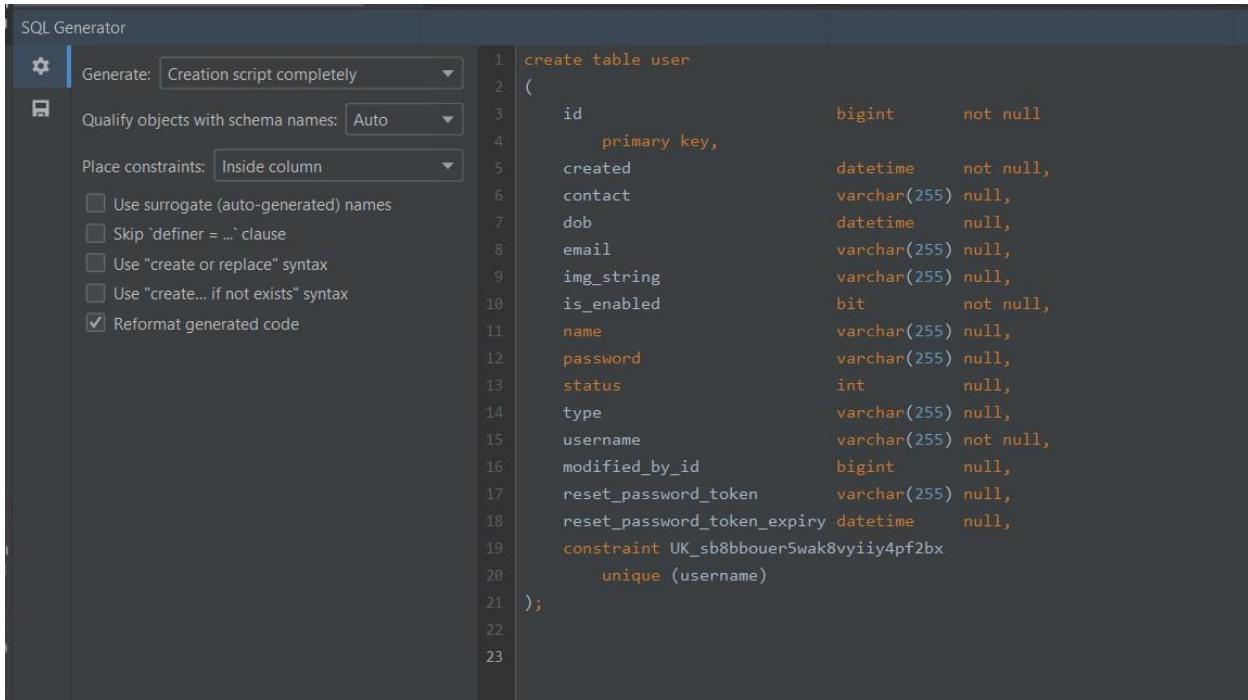
Full Name	<input type="text"/>	♂
Username	<input type="text"/>	✉
Email	<input type="text"/>	@
Phone number	<input type="text"/>	☎
Date of Birth	<input type="text"/>	📅
Password	<input type="password"/>	🔒

Register

Figure 210: Registration form layout

8.3.2 Sample code for the Automation Script

User table



The screenshot shows the SQL Generator interface with the following configuration:

- Generate: Creation script completely
- Qualify objects with schema names: Auto
- Place constraints: Inside column
- Checkboxes:
 - Use surrogate (auto-generated) names
 - Skip `definer = ...` clause
 - Use "create or replace" syntax
 - Use "create... if not exists" syntax
 - Reformat generated code

The generated SQL code for creating the `user` table is as follows:

```
1 create table user
2 (
3     id          bigint      not null
4         primary key,
5     created      datetime    not null,
6     contact      varchar(255) null,
7     dob          datetime    null,
8     email         varchar(255) null,
9     img_string   varchar(255) null,
10    is_enabled    bit        not null,
11    name          varchar(255) null,
12    password      varchar(255) null,
13    status         int        null,
14    type          varchar(255) null,
15    username      varchar(255) not null,
16    modified_by_id bigint      null,
17    reset_password_token  varchar(255) null,
18    reset_password_token_expiry datetime    null,
19    constraint UK_sb8bbouer5wak8vyiiy4pf2bx
20        unique (username)
21 );
22
23
```

Figure 211: Automation script of table user

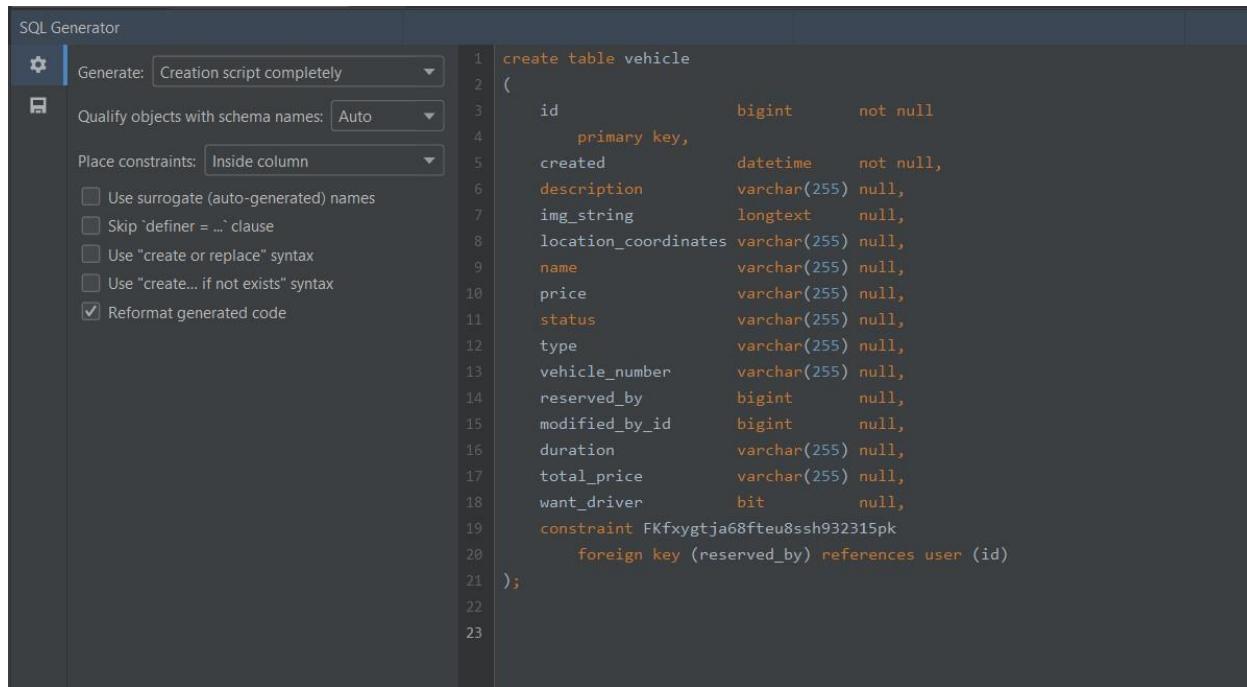
ws	25	Filter rows:	Search this table	Sort by key:	None				
id	created	contact	dob	email	img_string	is_enabled	name	password	status
1	2020-06-01 14:09:55	9546131	NULL	safishrestha12@gmail.com	NULL	0	The Administrator	\$2a\$10\$/97TLqet94FHpZBWWYsrS..A.2w5jFUyz2oLxRQ32al...	1
17	2020-05-28 10:10:57	9860027618	1999-11-19 18:15:00	safishrestha19@gmail.com	NULL	0	John Shrestha	\$2a\$10\$W2Q4/cj/GOyPntYdOh5EWeZzCx67BxGakI.PgE8tq7U...	1
32	2020-06-03 13:09:12	986532659	2020-06-10 18:15:00	sthaamulya@gmail.com	NULL	0	amulyadon	\$2a\$10\$WIEfUyWKs4YlcF.VCk62neAUNCKwgfTJEoDnfZ0u/IH...	1
53	2020-06-03 19:09:23	9815234678	1997-07-08 18:15:00	collyria@qqfullbet.club	NULL	0	Georgie Ruiz	\$2a\$10\$i57QU.uwRa/a37ejJatK7uK7qgGBOFLmkhYNt7AXGPW...	1
58	2020-06-03 19:55:05	6498458595	1996-02-12 18:15:00	collyria@k1069.com	NULL	0	Ella Weber	\$2a\$10\$8Vktl0QoHsXldzyp3cB2leCLjkzJAY7H.szfO1Abwc...	1
60	2020-06-04 01:44:20	4433346740	2020-04-13 18:15:00	collyria@rielunu.ml	NULL	0	Beatrice Pittman	\$2a\$10\$TePLXO4wDg4uHnYfmj8ccOUEPF2j63Sx4yQ/xAp8HaT...	1
61	2020-06-04 01:45:18	4138755104	1985-03-12 18:30:00	collyria@rielunu.ml	NULL	0	Roxie Matthews	\$2a\$10\$qsiBwjWF8DB4.IHVmVIF5OV9WG4Ue.Z2XPMSquMRiAQ...	1
62	2020-06-04 01:46:16	8187479834	1978-01-01 18:30:00	collyria@atstifin.tk	NULL	0	Eula Gutierrez	\$2a\$10\$WsGjaQ2lAtu/z1v7qq7umeO5QuM7.esoE3uyA1MNTRq...	1
63	2020-06-04 01:47:53	5244584919	1992-07-21 18:15:00	collyria@vlrata.ga	NULL	0	Micheal Dixon	\$2a\$10\$FZFCLYQs0aCHuZ/MscDC6OZlxVyLSSjYBhG.WqjLvf6...	1

img_string	is_enabled	name	password	status	type	username	modified_by_id
NULL	0	The Administrator	\$2a\$10\$/97TLqet94FHpZBWWYsrS..A.2w5jFUyz2oLxRQ32al...	1	admin	administrator	NULL
NULL	0	John Shrestha	\$2a\$10\$W2Q4/cj/GOyPntYdOh5EWeZzCx67BxGakI.PgE8tq7U...	1	client	John	NULL
NULL	0	amulyadon	\$2a\$10\$WIEfUyWKs4YlcF.VCk62neAUNCKwgfTJEoDnfZ0u/IH...	1	client	amulya	NULL
NULL	0	Georgie Ruiz	\$2a\$10\$i57QU.uwRa/a37ejJatK7uK7qgGBOFLmkhYNt7AXGPW...	1	client	George	NULL
NULL	0	Ella Weber	\$2a\$10\$8Vktl0QoHsXldzyp3cB2leCLjkzJAY7H.szfO1Abwc...	1	client	Ella	NULL
NULL	0	Beatrice Pittman	\$2a\$10\$TePLXO4wDg4uHnYfmj8ccOUEPF2j63Sx4yQ/xAp8HaT...	1	client	Bill Rice	NULL
NULL	0	Roxie Matthews	\$2a\$10\$qsiBwjWF8DB4.IHVmVIF5OV9WG4Ue.Z2XPMSquMRiAQ...	1	client	Cole Atkins	NULL
NULL	0	Eula Gutierrez	\$2a\$10\$WsGjaQ2lAtu/z1v7qq7umeO5QuM7.esoE3uyA1MNTRq...	1	client	Jay Roberson	NULL
NULL	0	Micheal Dixon	\$2a\$10\$FZFCLYQs0aCHuZ/MscDC6OZlxVyLSSjYBhG.WqjLvf6...	1	client	Ernest McDonald	NULL

Figure 212: Data populated into table user

Above figure is SQL-generated script to create table user. Data's populated in the table user are shown in figure as well.

Vehicle table



The screenshot shows the SQL Generator interface with the following configuration:

- Generate: Creation script completely
- Qualify objects with schema names: Auto
- Place constraints: Inside column
- Checkboxes (unchecked): Use surrogate (auto-generated) names, Skip 'definer = ...` clause, Use "create or replace" syntax, Use "create... if not exists" syntax.
- Checked checkbox: Reformat generated code

The generated SQL script for the vehicle table is as follows:

```
1 create table vehicle
2 (
3     id          bigint      not null
4         primary key,
5     created      datetime    not null,
6     description  varchar(255) null,
7     img_string   longtext    null,
8     location_coordinates  varchar(255) null,
9     name         varchar(255) null,
10    price        varchar(255) null,
11    status       varchar(255) null,
12    type         varchar(255) null,
13    vehicle_number  varchar(255) null,
14    reserved_by  bigint      null,
15    modified_by_id bigint      null,
16    duration      varchar(255) null,
17    total_price   varchar(255) null,
18    want_driver   bit        null,
19    constraint FKfxygtja68fteu8ssh932315pk
20        foreign key (reserved_by) references user (id)
21 );
22
23
```

Figure 213: Automation script of table vehicle

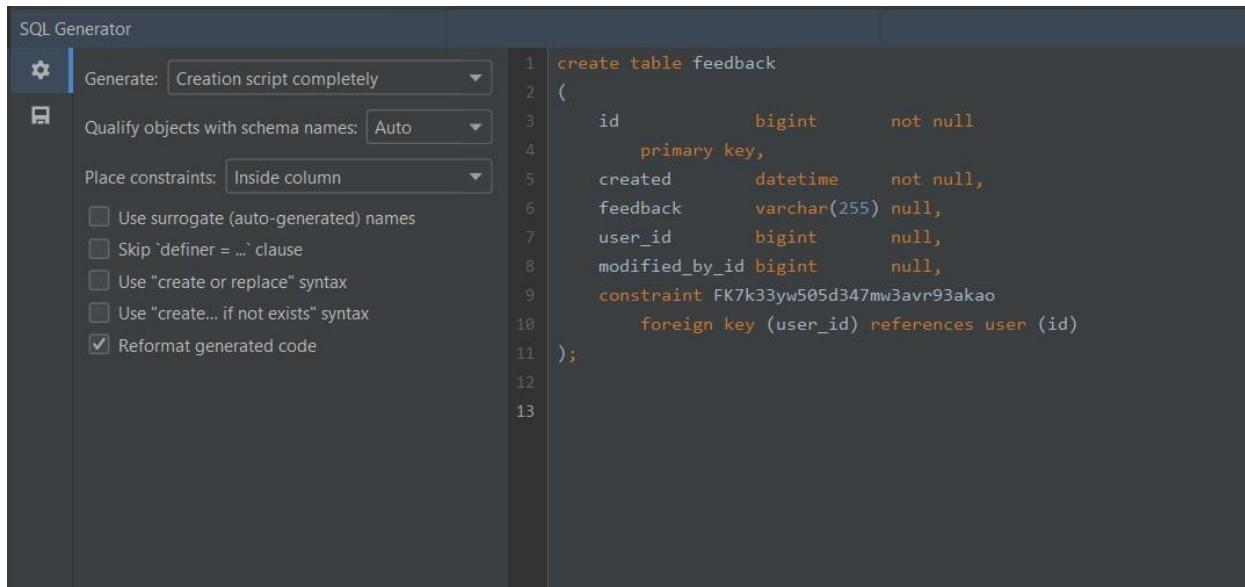
id	created	description	img_string	location_coordinates		name	price
e_21	2020-06-03 13:53:52	This is suzuki Swift has 1 on offer.	/9j/4AAQSkZJRgABAQAAAQABAAAD/2wCEAMCAgMCAgMDAwMEA... The Maruti Swift has 1	NULL		Suzuki 125	300
e_22	2020-06-03 20:31:42	Petrol Engine	/9j/4AAQSkZJRgABAQAAAQABAAAD/4QAqRXhpZgAASUkqAAgAAA... on offer. The...	27.73251152731478,85.36978638224308		Maruti	500
e_23	2020-06-04 01:50:29	125cc	/9j/4AAQSkZJRgABAQAAAQABAAAD/2wBDAAYEBQYFBAYGBQYHBW... Honda Glamour	27.57192430931789,85.58305363495548		Honda Glamour	150
e_24	2020-06-03 20:17:20	100cc	/9j/4AAQSkZJRgABAQEAYABgAAD//gA7Q1JFQVRPUjogZ2Qtan... Genia Scooter	NULL		Genia Scooter	150
e_25	2020-06-03 14:06:05	1000cc petrol engine	/9j/4AAQSkZJRgABAQAAAQABAAAD//gA8Q1JFQVRPUjogZ2Qtan... Swift	NULL		Swift	400
e_43	2020-06-03 20:25:18	2000cc	/9j/4AAQSkZJRgABAQIBLAEsAAD/2wBDAAYEBQYFBAYGBQYHBW... Creta	27.828253875900014,85.57405489031687		Creta	600
e_44	2020-06-03 18:19:37	150 cc	UklGRiwtAABXRUJQVIA4ICAtAADwkJAcDASpoAcoAPIUmj0Wjoi... Hornet	NULL		Hornet	100
e_45	2020-06-03 18:20:31	This is ferrari	UklGRgQZAABXRUJQVIA4IPgYAAAQaQCDASpoAfAAPi0Sh0lhoQ... Ferrari	NULL		Ferrari	1500
e_46	2020-06-03 18:21:14	150 cc	UklGRvCQAABXRUJQVIA4IOSQAAQLQGdASrcAQwBPgQBQAAJC... Honda Shine	NULL		Honda Shine	150
e_47	2020-06-04 01:49:04	450 cc	/9j/4AAQSkZJRgABAQEASABIAAD/4gxYSUNDX1BST0ZJTEUAAQ... Duke	27.622775562699378,85.26198687256563		Duke	500
e_48	2020-06-03 18:22:49	110 cc	UklGRkptAABXRUJQVIA4ID5tAAQBwGdASrcAQwBPgQBjgAACJ... Aviator	NULL		Aviator	200

status	type	vehicle_number	reserved_by	modified_by_id	duration	total_price	want_driver	rent_from	rent_to
booked	Four-wheeler	BA Kha 1234578	32	NULL	10	3000	1	2020-06-13 18:15:00	2020-06-16 18:15:00
none	Four-wheeler	454545	NULL	NULL	5	2500	0	2020-05-31 18:15:00	2020-06-04 18:15:00
booked	Two-wheeler	653168	61	NULL	1	150	0	2020-06-09 18:15:00	2020-06-10 18:15:00
none	Two-wheeler	4561	NULL	NULL	5	750	0	2020-06-06 18:15:00	2020-06-09 18:15:00
booked	four-wheeler	45454	32	NULL	15	6000	0	2020-06-15 18:15:00	2020-06-18 18:15:00
rented	Four-wheeler	56321	58	NULL	3	1800	1	2020-06-15 18:15:00	2020-06-18 18:15:00
none	Two-Wheeler	76423	NULL	NULL	NULL	NULL	NULL	NULL	NULL
none	Four-Wheeler	456315	NULL	NULL	NULL	NULL	NULL	NULL	NULL
none	Two-Wheeler	784026	NULL	NULL	NULL	NULL	NULL	NULL	NULL
booked	Two-Wheeler	451360	63	NULL	2	1000	0	2020-06-11 18:15:00	2020-06-13 18:15:00
none	Two-Wheeler	45610378	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 214: Data populated into table vehicle

Above figure is SQL-generated script to create table vehicle. Data's populated in the table vehicle are shown in figure.

Feedback table



The screenshot shows the SQL Generator interface with the following configuration:

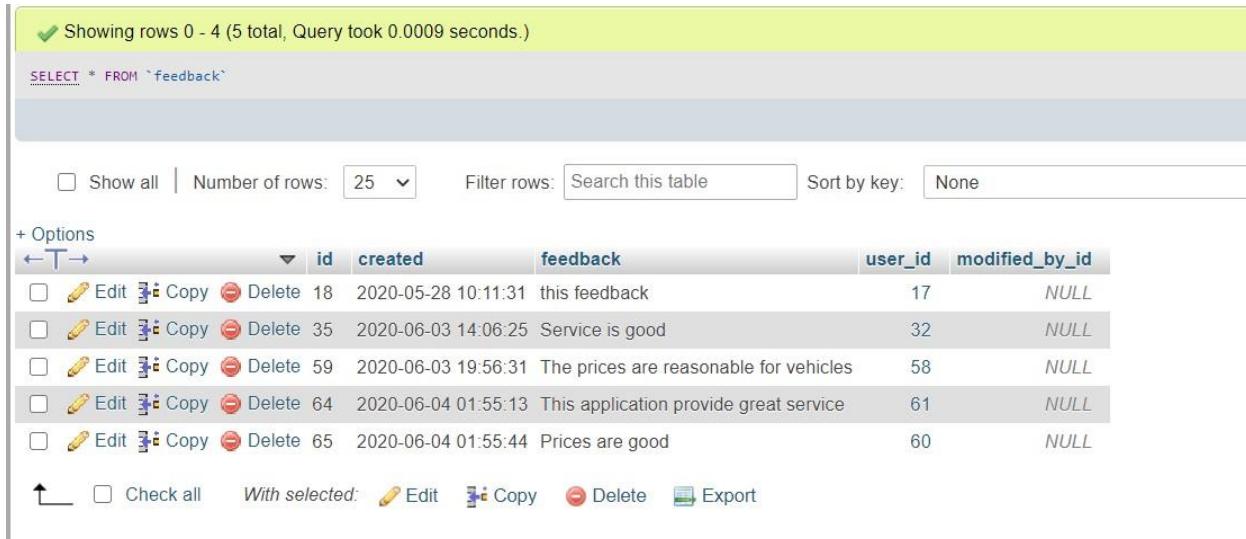
- Generate: Creation script completely
- Qualify objects with schema names: Auto
- Place constraints: Inside column
- Reformat generated code (checkbox checked)

```

1 create table feedback
2 (
3     id          bigint      not null
4         primary key,
5     created      datetime    not null,
6     feedback     varchar(255) null,
7     user_id      bigint      null,
8     modified_by_id bigint      null,
9     constraint FK7k33yw505d347mw3avr9akao
10        foreign key (user_id) references user (id)
11 );
12
13

```

Figure 215: Automation script of table feedback



The screenshot shows the MySQL Workbench interface with the following details:

- Query results: Showing rows 0 - 4 (5 total, Query took 0.0009 seconds.)
- SQL query: `SELECT * FROM `feedback``
- Table data:

	<input type="checkbox"/> Show all	Number of rows: 25	Filter rows: Search this table	Sort by key: None		
+ Options	<input type="checkbox"/>					
<input type="checkbox"/> <input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	<input type="button" value="T"/>	id	created	feedback	user_id	modified_by_id
18	2020-05-28 10:11:31	this feedback	17	NULL		
35	2020-06-03 14:06:25	Service is good	32	NULL		
59	2020-06-03 19:56:31	The prices are reasonable for vehicles	58	NULL		
64	2020-06-04 01:55:13	This application provide great service	61	NULL		
65	2020-06-04 01:55:44	Prices are good	60	NULL		

Buttons at the bottom: Check all, With selected:

Figure 216: Data populated into table feedback

Above figure is SQL-generated script to create table feedback. Data's populated in the table feedback are shown in figure.

8.4 Appendix D: Designs

8.4.1 Gantt Chart

Initial Gantt chart

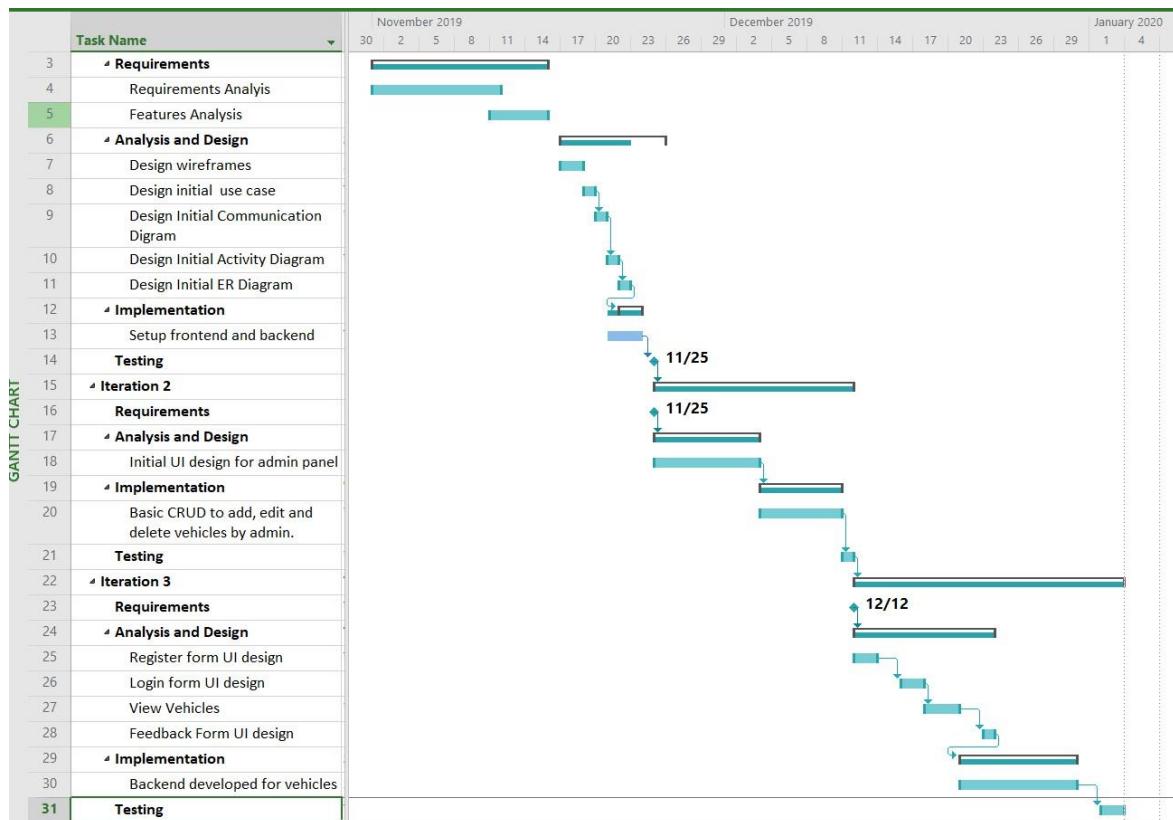


Figure 217: Initial Gantt chart

Final Gantt chart

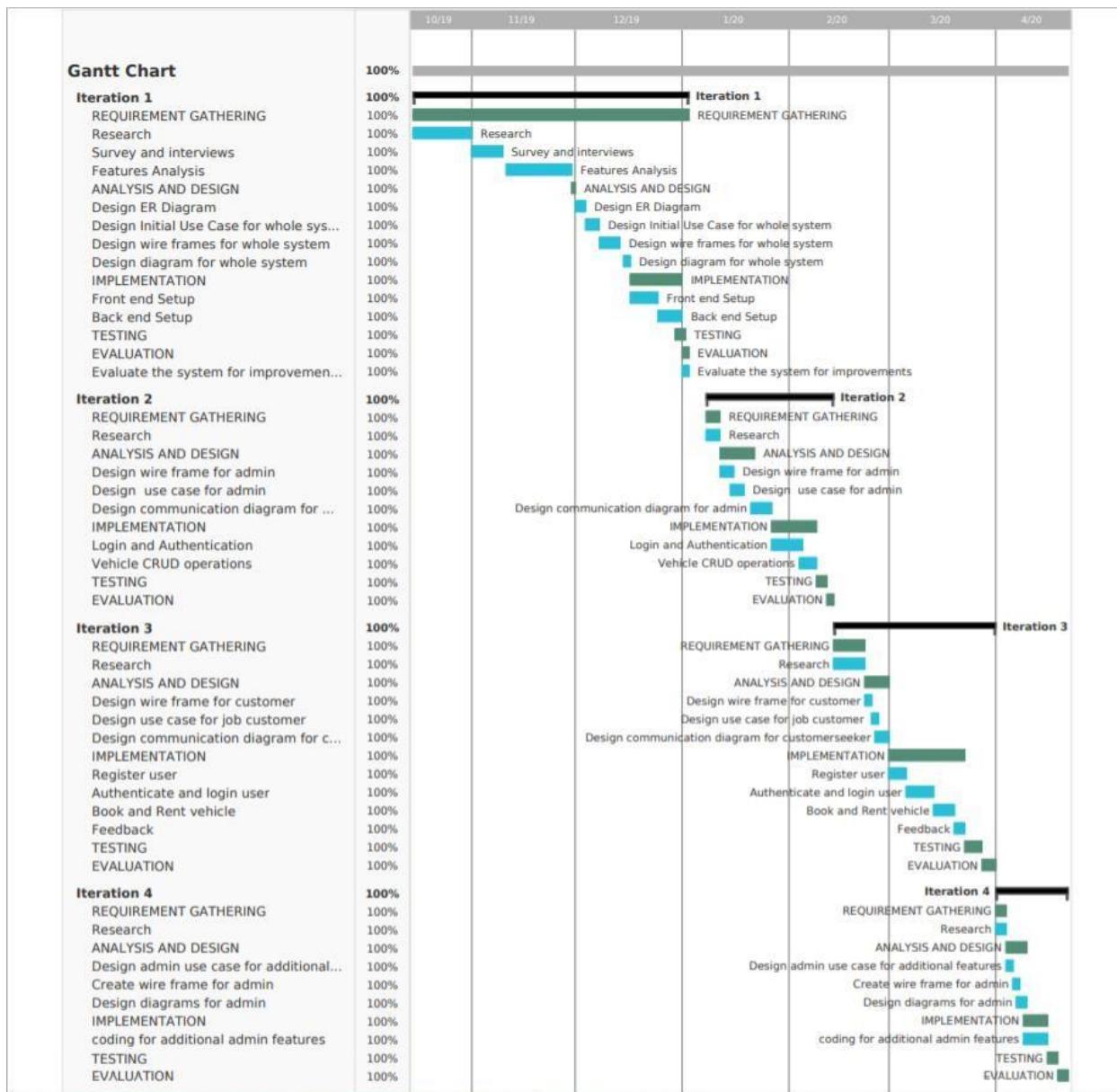


Figure 218: Final Gantt chart

8.4.2 Work breakdown structure

Initial Work breakdown structure

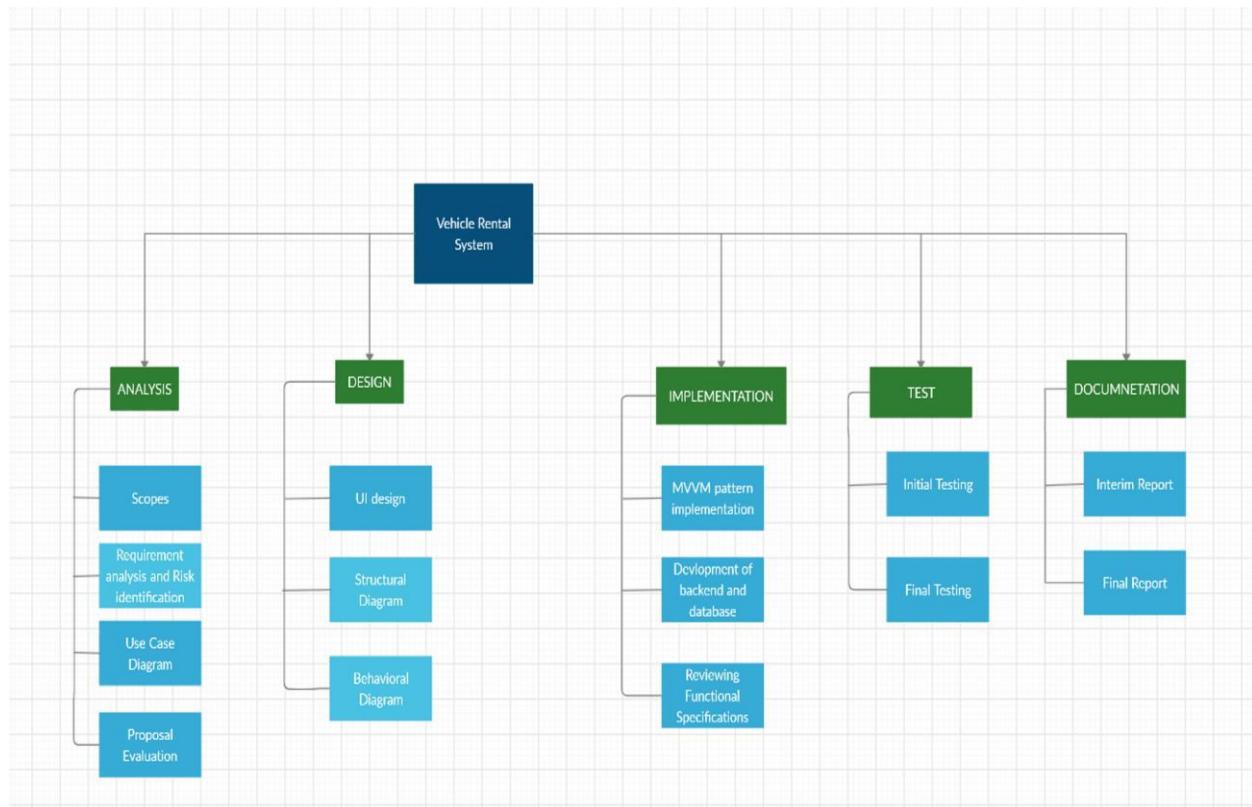


Figure 219: Initial Work breakdown structure

Final Work breakdown structure

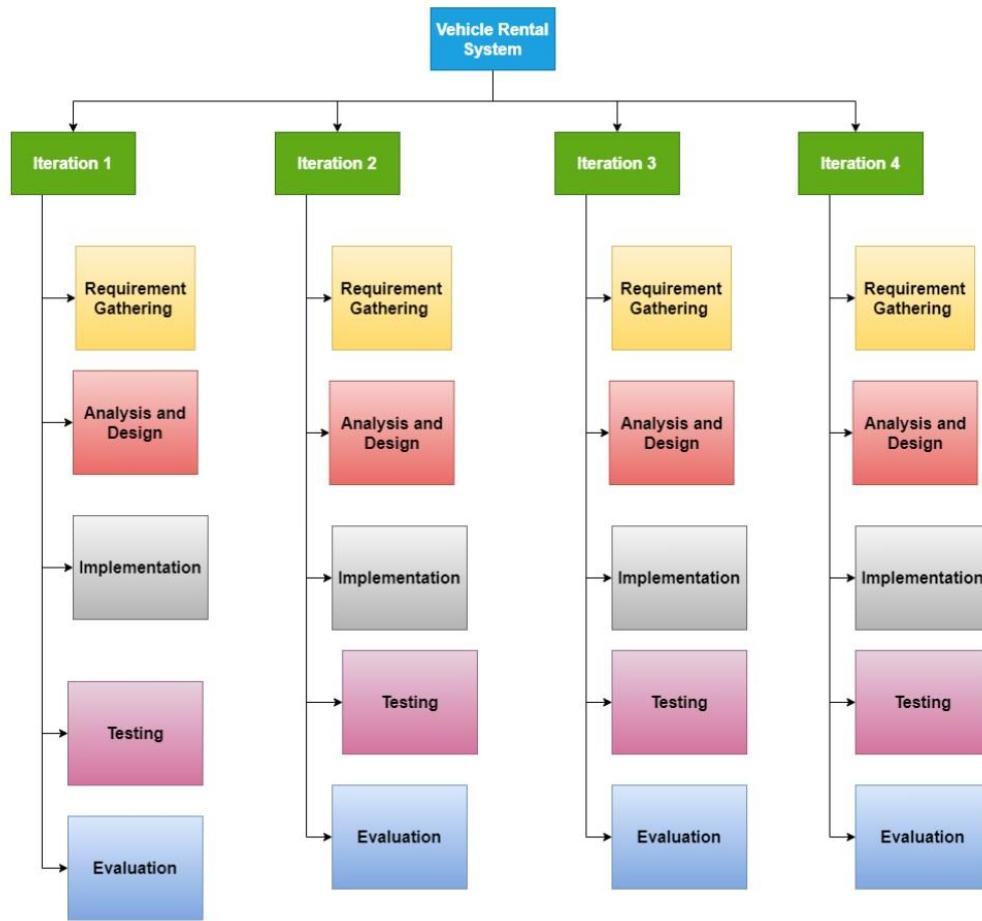


Figure 220: Final Work breakdown structure

WBS for Iteration 1

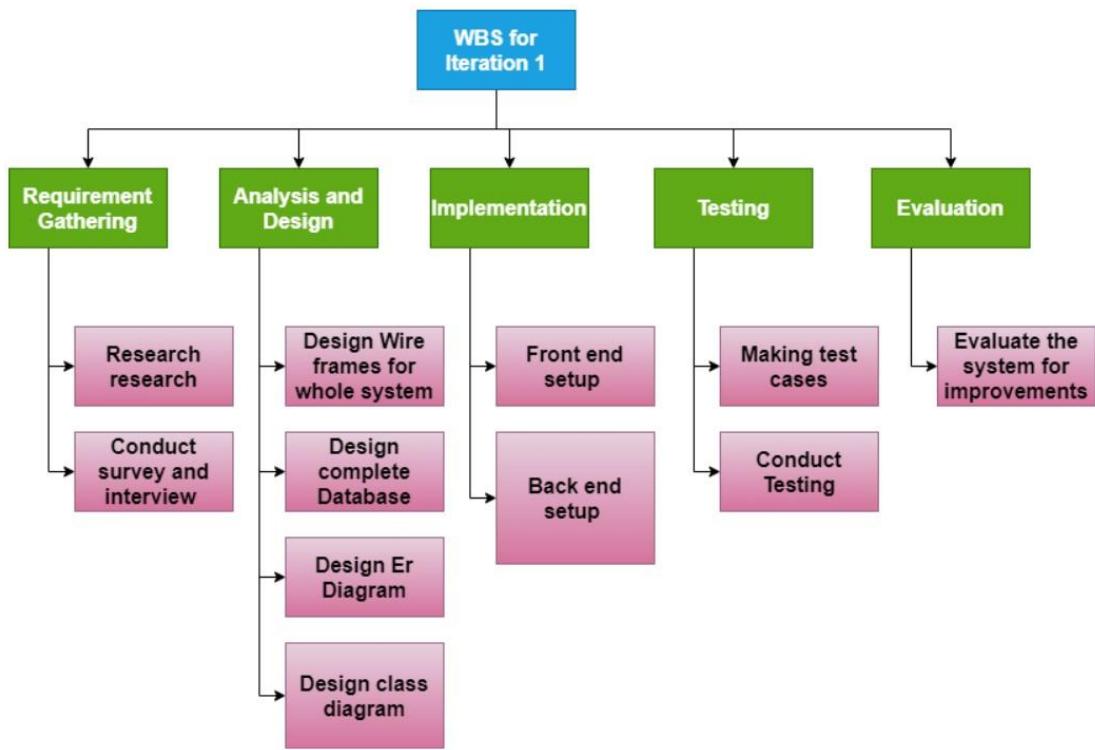


Figure 221: Final WBS for Iteration 1

WBS for I teration 2

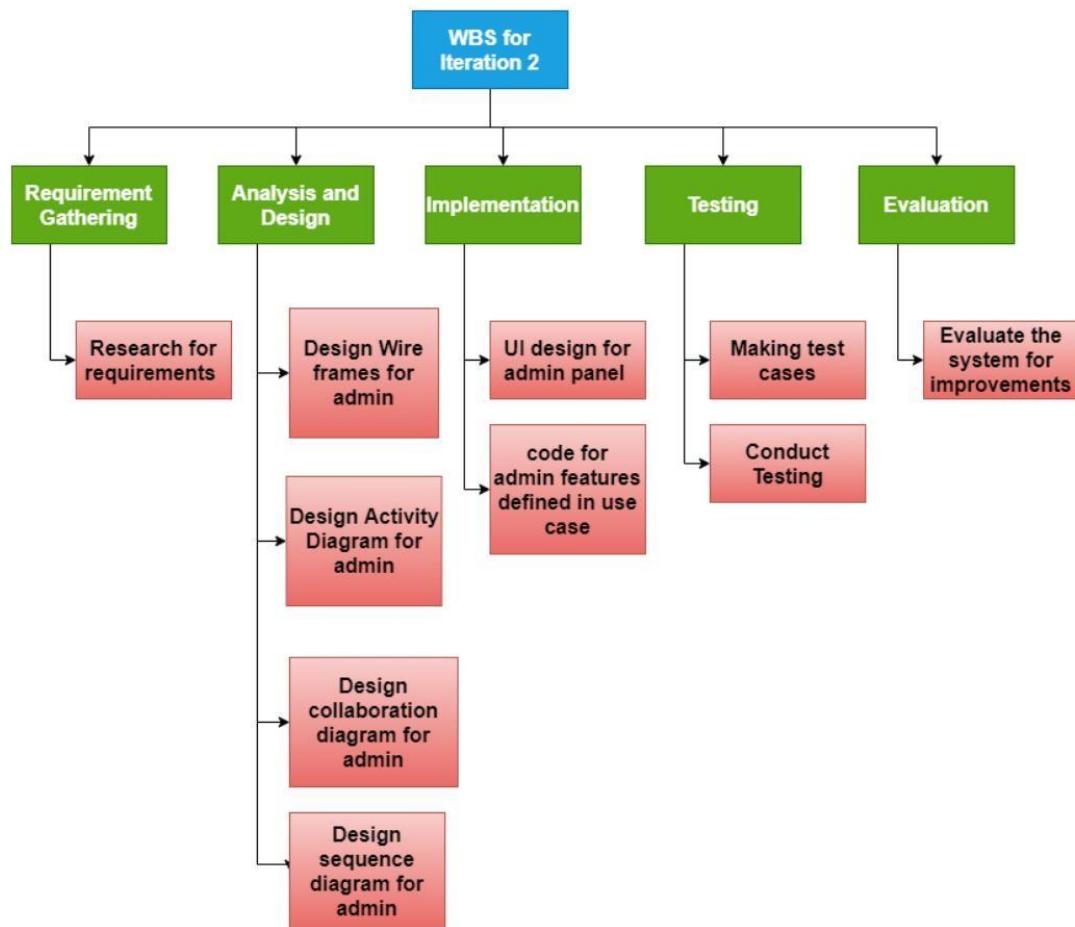


Figure 222: Final WBS for Iteration 2

WBS for I

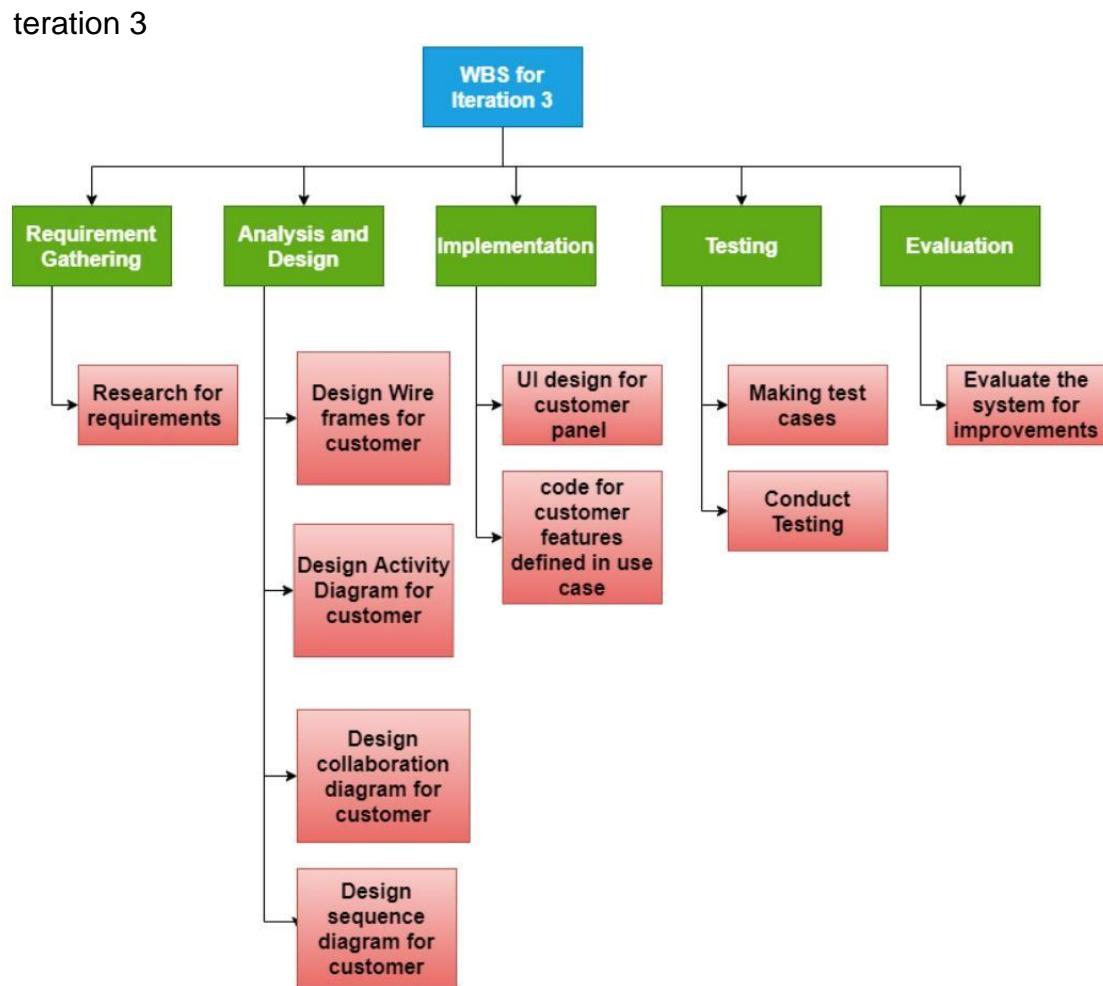


Figure 223: Final WBS for Iteration 3

WBS for I

teration 4

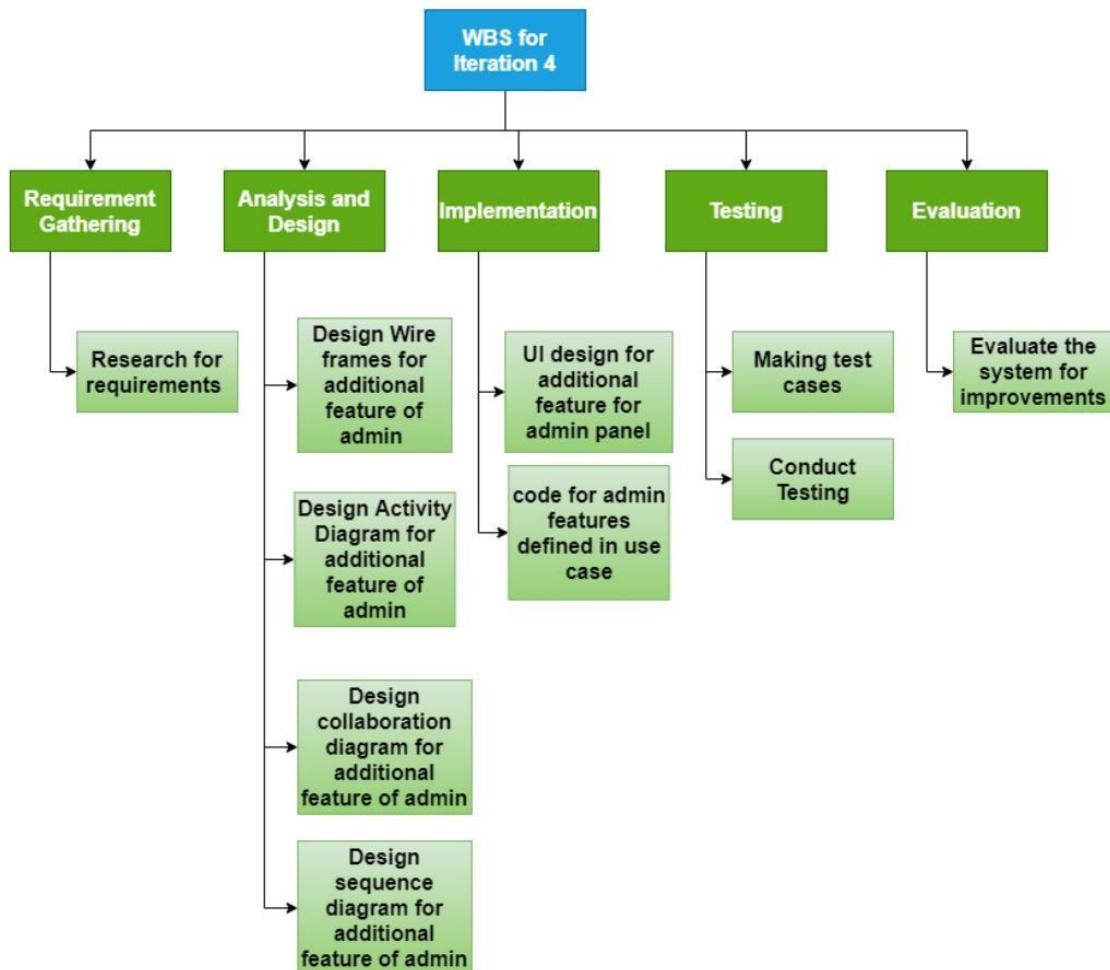


Figure 224: Final WBS for Iteration 4

8.4.3 Algorithms and flowcharts

Algorithms:

Step 1: Login with username and password

Step 2: If registered user go to Step 3 else go to step 4.

Step 3: If valid credentials go to step 4. If invalid credentials go to Step 1

Step 4: Register user

Step 4: If user type admin go to Step 5 and Step 6 else go to Step 7 and 8

Step 5: Redirect to admin panel

Step 6: Perform admin actions like add, update, delete vehicles, view vehicle report

Step 7: Redirect to customer panel

Step 8: Perform actions like book and rent vehicles.

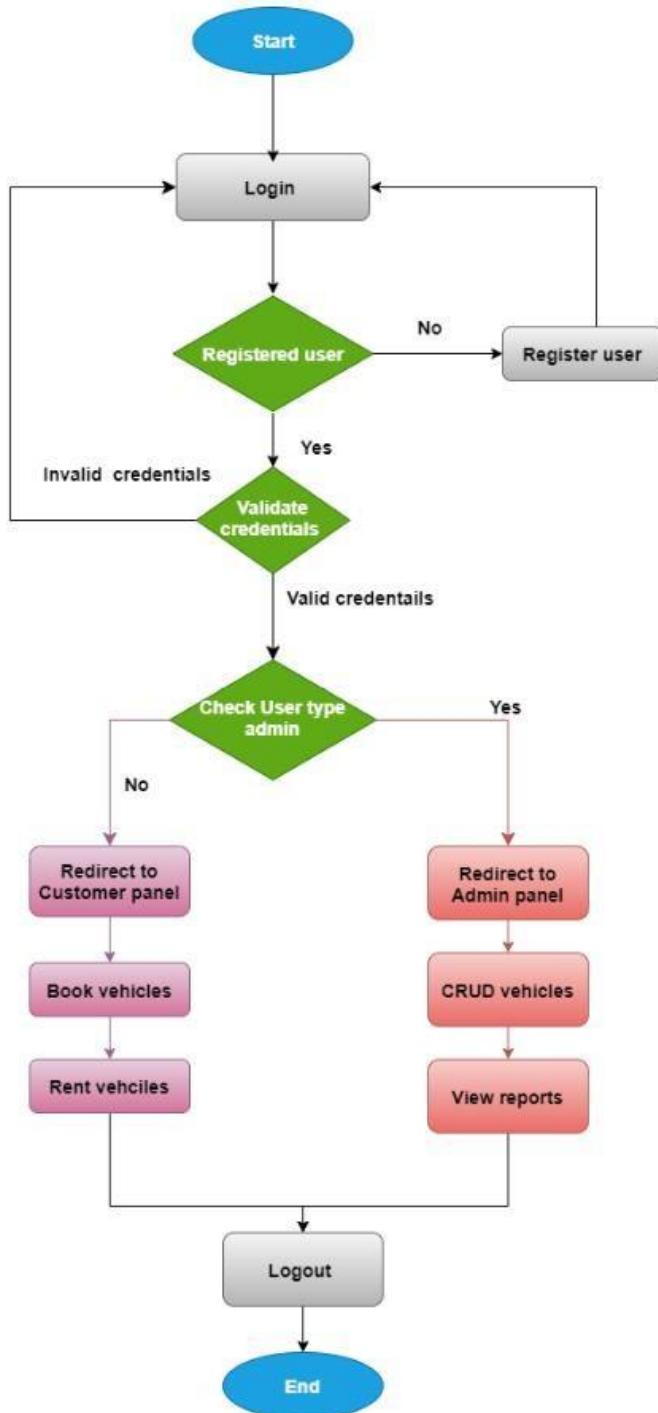
Flowchart:

Figure 225: Flowchart of whole system

8.4.4 3D Modelling

The screenshot shows a dark-themed Admin panel for vehicle management. On the left, a sidebar lists navigation options: Vehicles, Users, Vehicle type, Locations, REPORT, and Feed-backs. A prominent red "Add" button is located in the top right corner of the main content area. The central part of the screen displays a pink modal window titled "Add Vehicle". Inside the modal, there are two input fields: "Vehicle Name" with placeholder text "Enter Vehicle name" and "Rent price" with placeholder text "Price in Rupees". Below the modal, a table lists existing vehicles with columns: #, VEHICLE NAME, RENT PRICE, BOOKED STATUS, RENT STATUS, and ACTIONS. Two entries are shown:

#	VEHICLE NAME	RENT PRICE	BOOKED STATUS	RENT STATUS	ACTIONS
1	asd	12121			
2	zxc	12121			

Figure 226: 3D modelling of Admin panel



Figure 227: 3D modeling of registration form

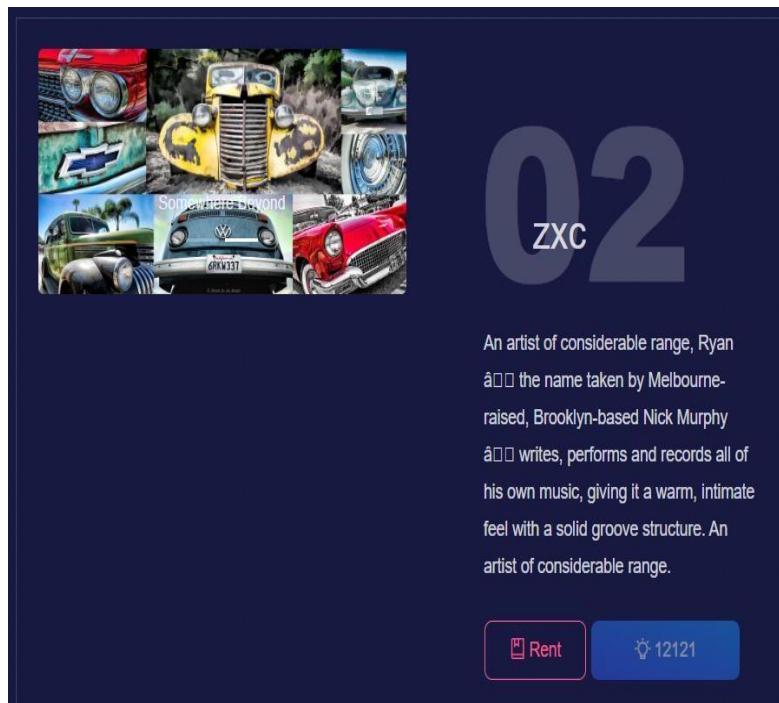


Figure 228: 3D modelling of rent vehicle

8.4.5 Use case

Iteration 2 High level Use case

1.

Use case Name: **Admin login**

Actor: Admin

Description: Admin enters his or her email and password on the login form provided to the user by the system to login.

2.

Use case Name: **Forgot Password**

Actor: Admin

Description: Admin enters their email address on a form the system provides after pressing the forget password button which sends OTP to the user's email. By entering the code admin they can enter their new password.

3.

Use case Name: **Change Password**

Actor: Admin

Description: On the profile page an admin can change their password by entering new password and clicking update button.

4.

Use case Name: **Manage vehicles**

Actor: Admin

Description: Admin selects vehicle action then they can add vehicles, edit and delete vehicles.

Expanded Use Case

1.

Use case Name: Admin login

Actor: Admin

Description: Admin enters his or her email and password on the login form provided to the user by the system to login.

Typical Course of Events:

Actor Action	System Response
	1. GUI asking users login detail
2. User enters login detail	3. The system verifies the information given
	4. Login Successful message

Alternative Courses:

Line 3: The entered login information is incorrect and displays error message. Use case ends.

2.

Use case Name: **Forgot Password**

Actor: Admin

Description: Admin enters their email address on a form the system provides after pressing the forget password link which sends OTP to the user's email. By entering the code admin they can enter their new password.

Typical Course of Events:

Actor Action	System Response
	1. GUI having forget password link is displayed
2. User clicks forget password link	3. GUI asking for users' email is shown
4. User enters their email	5. System validates the email
	6. System sends OTP to the users' email
	7. GUI showing reset password form is displayed
8. User enters verification code	
9. User enters new password	10. Password successfully changed

Alternative Courses:

Line 5: Email validation fails. Use case ends.

3.

Use case Name: **Change Password**

Actor: Admin

221

9

0

0

Description: On the profile page an admin can change their password by entering new password and clicking update button.

Typical Course of Events:

Actor Action	System Response
1. User clicks on profile	2. Profile GUI is displayed
3. User enters new password	4. System update the information given
	5. Redirect to login page

Use case Name:

4.

Manage vehicles

Actor: Admin

Description: Admin selects vehicle action then they can add vehicles, edit and delete vehicles.

Typical Course of Events:

Actor Action	System Response
1. User clicks on vehicle action	2. Vehicle action GUI is displayed
3. User add/edit/delete vehicles	4. System add/edit/delete the vehicles
	5. Success message displayed.

Alternative Courses:

Line 4: add/edit/delete vehicle fails. Use case ends.

Iteration 3 High level Use case

1.

Use case Name: **Register Customer**

Actor: Customer

Description: Customer enters his or her details on the registration form provided to the user by the system to register.

2.

Use case Name: **Customer login**

Actor: Customer

Use case Name:

Description: Customer enters his or her email and password on the login form provided to the user by the system to login.

3.

Use case Name: **Forgot Password**

Actor: Customer

Description: Customer enters their email address on a form the system provides after pressing the forget password button which sends OTP to the user's email. By entering the code customer they can enter their new password.

4.

Use case Name: **Change Password**

Actor: Customer

Description: On the profile page a customer can change their password by entering new password and clicking update button.

5

Book vehicles

Actor: Customer

Description: Customer clicks book button then enter booking details and click book button.

6.

Use case Name: **Rent vehicles**

Actor: Customer

Description: Customer clicks rent button then enter rent details and click rent button.

7.

Use case Name: **Search vehicles**

Actor: Customer

Description: Customer filters vehicles by search name or vehicle type.

8.

Use case Name: **View available vehicles**

Actor: Customer

Description: Customer selects available vehicles then all the available vehicles are displayed.

8.

Use case Name: **Send feedback**

Actor: Customer

Description: Customer selects feedback button then write comment and send feedback.

Expanded Use Case

1.

Use case Name: **Register Customer**

Actor: Customer

Description: Customer enters his or her details on the registration form provided to the user by the system to register.

Typical Course of Events:

Actor Action	System Response
	1. GUI asking users detail

Use case Name:

2. User enters users details	3. The system verifies the email
	4. Register User Successful message

Alternative Courses:

Line 3: Email verification failed. Use case ends.

2

Customer login

Actor: Customer

Description: Customer enters his or her email and password on the login form provided to the user by the system to login.

Typical Course of Events:

Actor Action	System Response
	1. GUI asking users login detail

2. User enters login detail	3. The system verifies the information given
	4. Login Successful message

Alternative Courses:

Line 3: The entered login information is incorrect and displays error message. Use case ends.

3.

Use case Name: **Forgot Password**

Actor: Customer

Description: Customer enters their email address on a form the system provides after pressing the forget password link which sends OTP to the user's email. By entering the code customer they can enter their new password.

Typical Course of Events:

Use case Name:

Actor Action	System Response
	1. GUI having forget password link is displayed
2. User clicks forget password link	3. GUI asking for users' email is shown
4. User enters their email	5. System validates the email
	6. System sends OTP to the users' email
	7. GUI showing reset password form is displayed
8. User enters verification code	
9. User enters new password	10. Password successfully changed

Alternative Courses:

Line 5: Email validation fails. Use case ends.

4

Use case Name: **Change Password**

Actor: Customer

Description: On the profile page an customer can change their password by entering new password and clicking update button.

Typical Course of Events:

Actor Action	System Response
1. User clicks on profile	2. Profile GUI is displayed
3. User enters new password	4. System update the information given
	5. Redirect to login page

5

Book vehicles

Description: Customer clicks book button then enter booking details and click book button.

Typical Course of Events:

Actor Action	System Response
1. User clicks on vehicle available	2. Vehicle available GUI is displayed
3. User clicks book button	4. Booking details GUI is displayed
5. User enters booking details	
6. User clicks book button	7. System stores details in database
	8. Success message

9

0

0

Use case Name:

Actor: Customer

Alternative Courses:

Line 7: Error storing in database. Use case ends.

6

Rent vehicles

Description: Customer clicks rent button then enter rent details and click rent button.

Typical Course of Events:

Actor Action	System Response
1. User clicks on vehicle available	2. Vehicle available GUI is displayed
3. User clicks rent button	4. Rent details GUI is displayed
5. User enters rent details	
6. User clicks rent button	7. System stores details in database
	8. Success message

Alternative Courses:

Line 7: Error storing in database. Use case ends.

7

Search vehicles

Description: Customer filters vehicles by search name or vehicle type.

Use case Name:

Actor: Customer

Typical Course of Events:

Actor Action	System Response
1. User enters vehicle name in search field.	2. System filters vehicles by vehicle name searched.
3. User search by vehicle type selecting radio button	4. System filters vehicles by vehicle type searched.

Alternative Courses:

Line 2: Error filtering vehicle list. Use case ends.

Line 4: Error filtering vehicle list. Use case ends.

8

View available vehicles

Description: Customer selects available vehicles then all the available vehicles are displayed.

Typical Course of Events:

Actor Action	System Response
1. User select available vehicles	2. System displays all the available vehicles. 3. Redirect to available vehicle.

Alternative Courses:

Line 2: Error displaying vehicle list. Use case ends.

9

0

0

Use case Name:

Actor: Customer

8.

Use case Name: **Send feedback**

Actor: Customer

Description: Customer selects feedback button then write comment and send feedback.

Typical Course of Events:

Actor Action	System Response
1. User click feedback button	2. Feedback form is displayed
3. User submit feedback	4. System stores feedback in the database.

Alternative Courses:

Line 4: Error storing feedback. Use case ends.

Iteration 4 High Level Use case

1.

Use case Name: View vehicle report

Actor: Admin

Description: Admin selects vehicle rented by client then vehicle on rent report is displayed with customer and vehicle details.

2.

Use case Name: Send confirmation

Actor: Admin

Description: Admin approve or reject requests by client and send email respectively.

3.

Use case Name: View feedback

Actor: Admin

Description: Admin selects feedback then all feedbacks are displayed.

Expanded Use case

1.

Use case Name: View vehicle report

Actor: Admin

Description: Admin selects vehicle rented by client then vehicle on rent report is displayed with customer and vehicle details.

Typical Course of Events:

Actor Action	System Response
1. User selects vehicle approved	2. System displays vehicle approved for rent or vehicle that are on rent report

Alternative Courses:

Line 2: Failed to fetch data. Use case ends.

2.

Use case Name: **Send confirmation**

Actor: Admin

Description: Admin approve or reject requests by client and send email respectively.

Typical Course of Events:

Actor Action	System Response
1. User selects vehicle reserved by client	2. System displays all rent and booked vehicles.
3. User click approve button	4. System sends approval email to customer.
5. User click reject button	5. System sends rejection email to customer.

Alternative Courses:

Line 2: Email send fails. Use case ends.

Line 4: Email send fails. Use case ends.

3.

Use case Name: **View feedback**

Actor: Admin

Description: Admin selects feedback then all feedbacks are displayed.

Typical Course of Events:

Actor Action	System Response
1. User selects feedback	2. Feedback report is displayed by system

Alternative Courses:

Line 2: Error displaying feedbacks. Use case ends.

8.4.6 Wireframe

Initial Wireframe

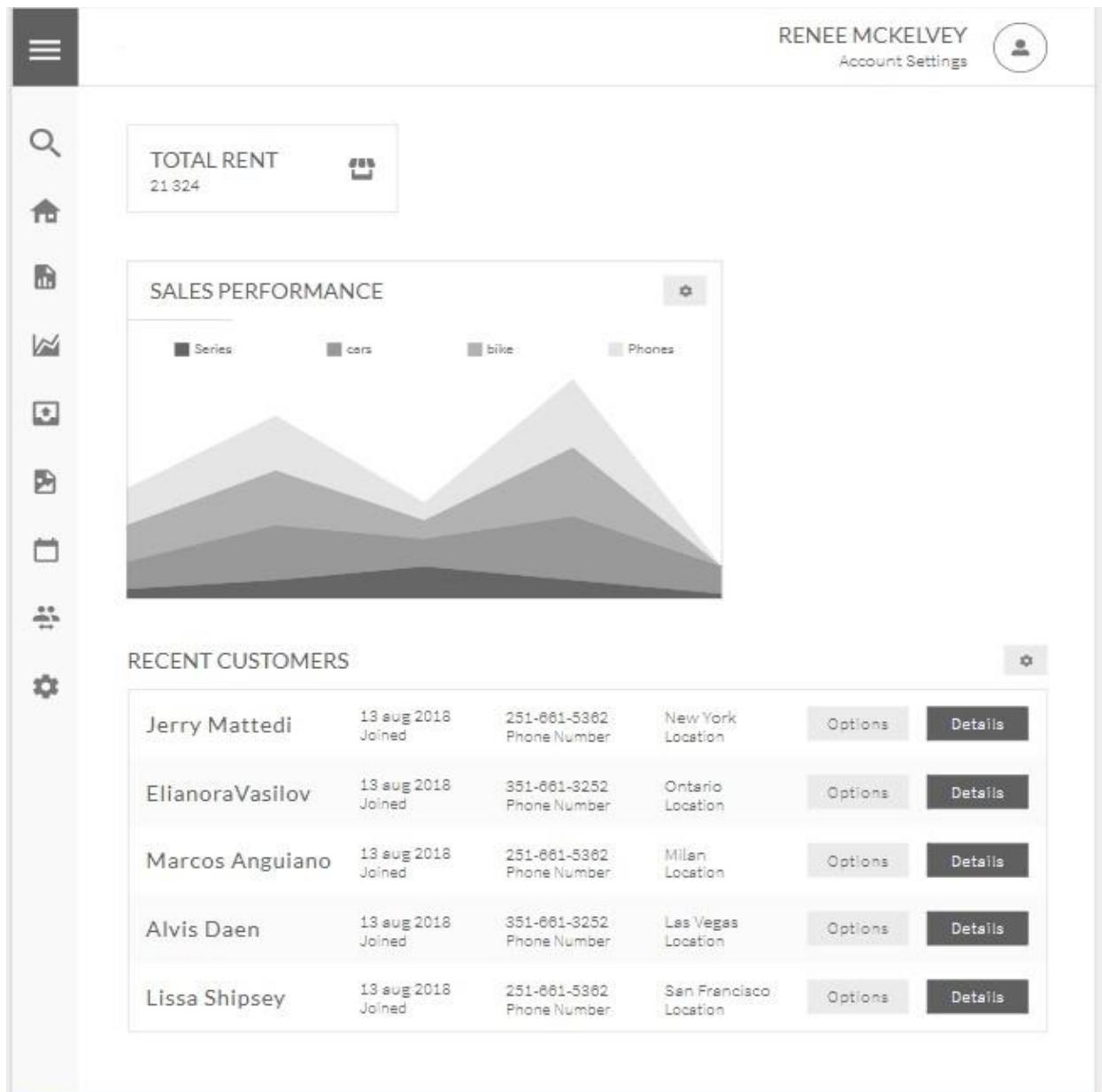


Figure 229: Initial Wireframe of Admin panel

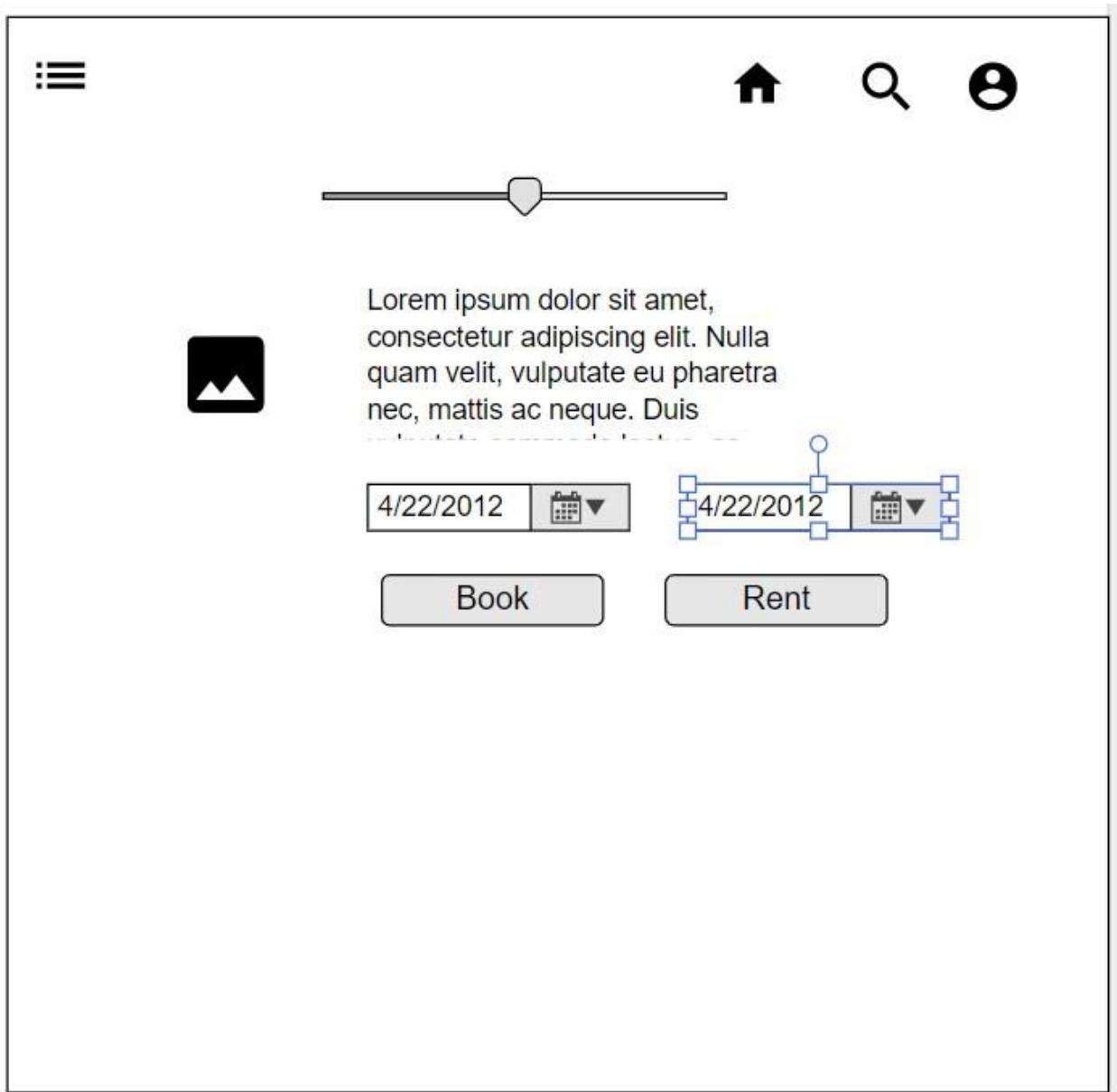
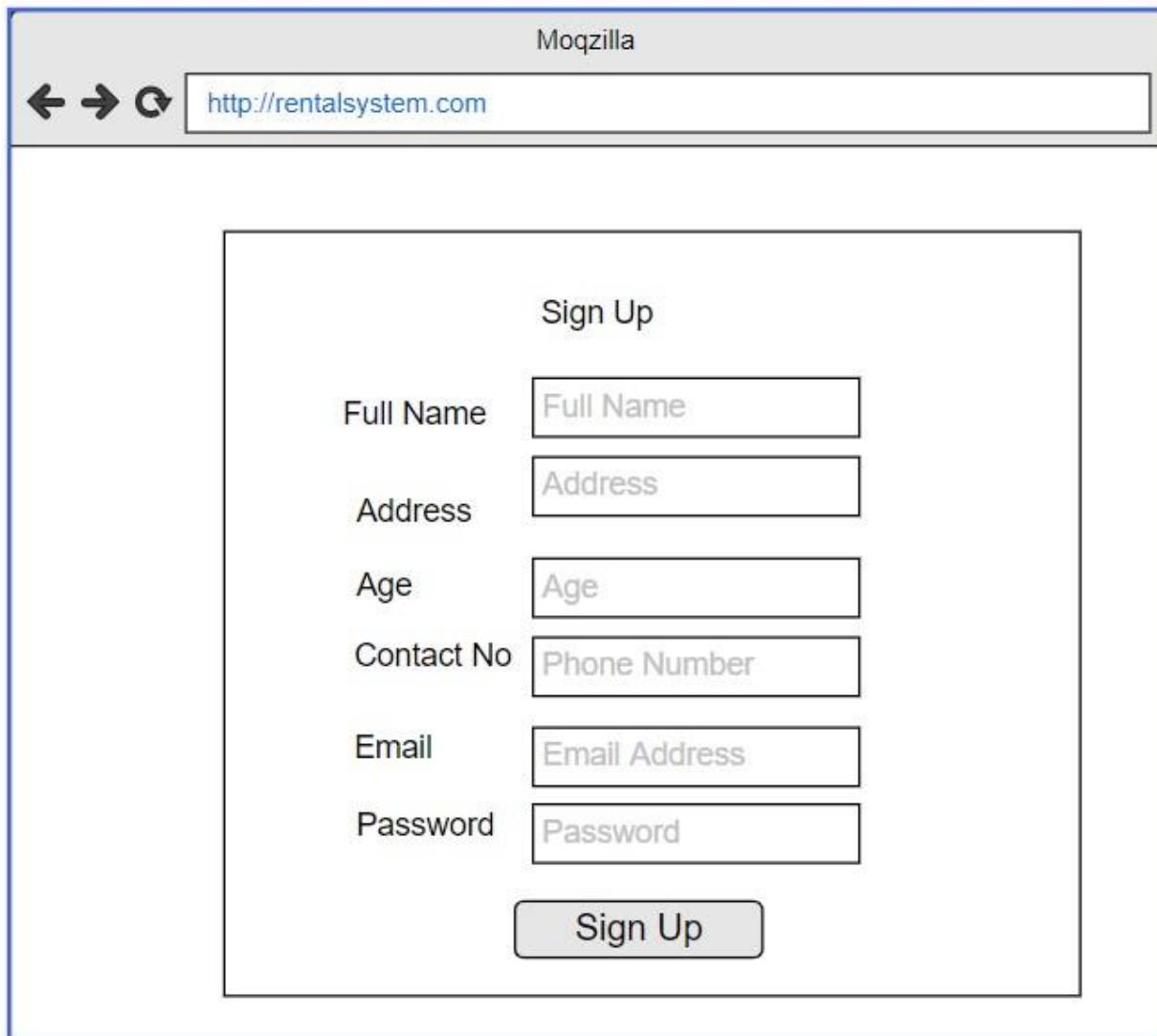


Figure 230: Initial Wireframe of Customer panel

The wireframe depicts a web browser window with the title 'Moqzilla'. The address bar shows the URL 'http://rentalsystem.com'. The main content area is a login form titled 'Sign In'. It features two input fields: 'UserName' and 'Password', each followed by a placeholder 'Email Address'. Below the password field is a large 'Sign In' button. To its right is the text 'OR', and further down is another button labeled 'Create Account'.

Figure 231: Initial Wireframe of Login form



The wireframe shows a registration form titled "Sign Up". It includes fields for Full Name, Address, Age, Contact No, Email, and Password, each with a corresponding input box. A "Sign Up" button is at the bottom.

Label	Input Type
Full Name	Text
Address	Text
Age	Text
Contact No	Text
Email	Text
Password	Text

Sign Up

Figure 232: Initial Wireframe of Registration form

8.4.7 Designs

Initial Sequence Diagram

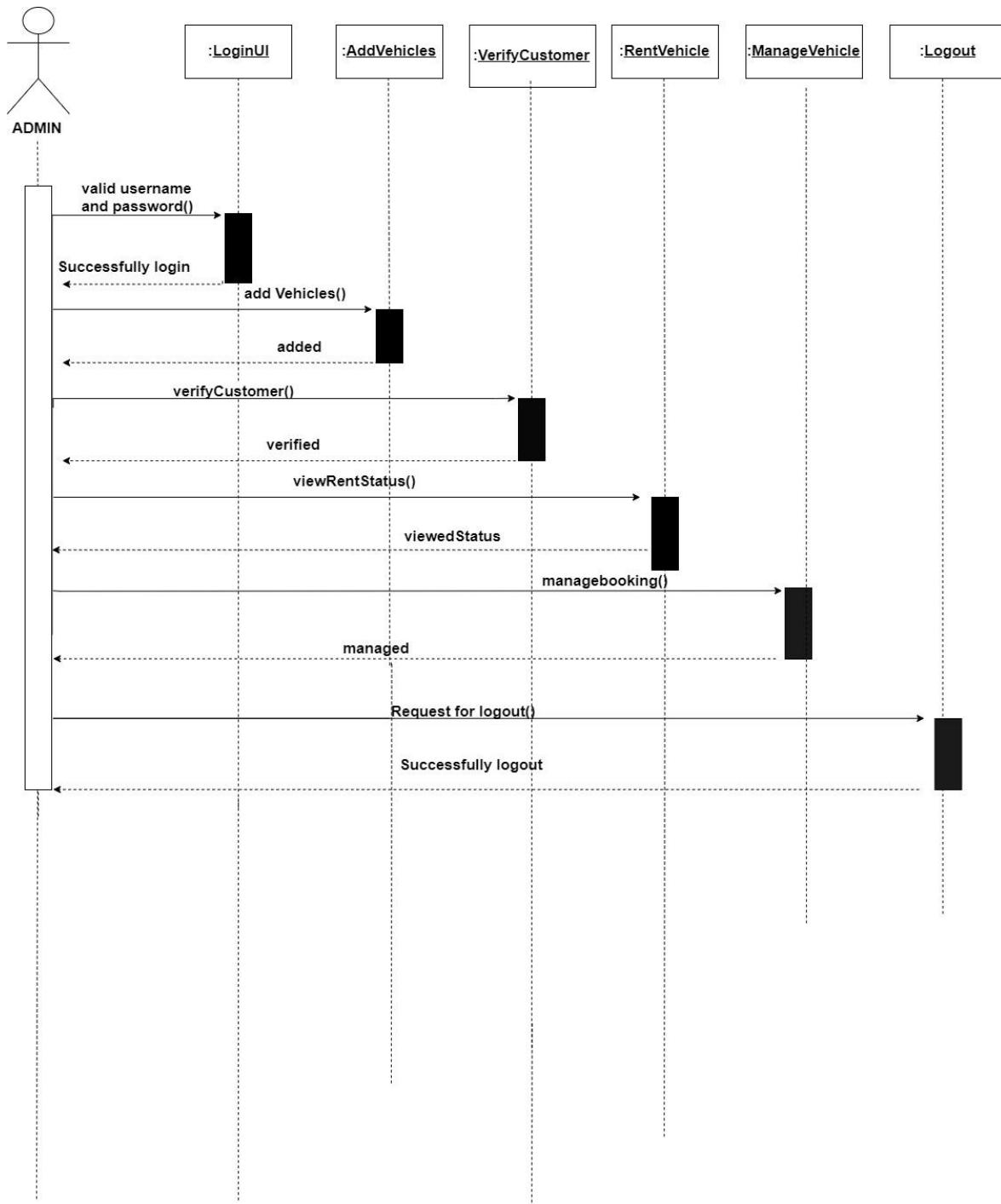


Figure 233: Initial Sequence Diagram for Admin

Customer

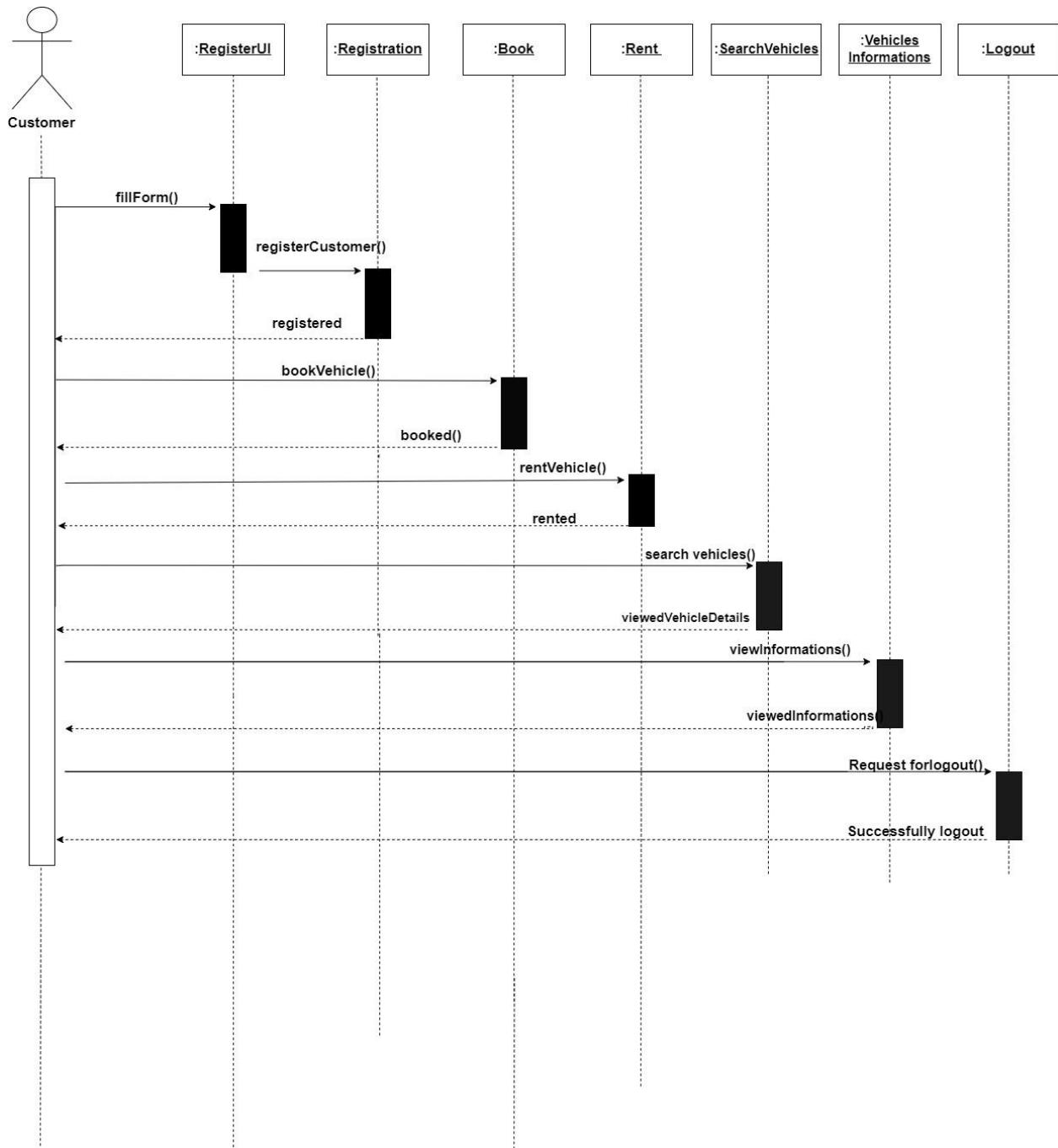


Figure 234: Initial Sequence Diagram for Customer

Initial Collaboration Diagram

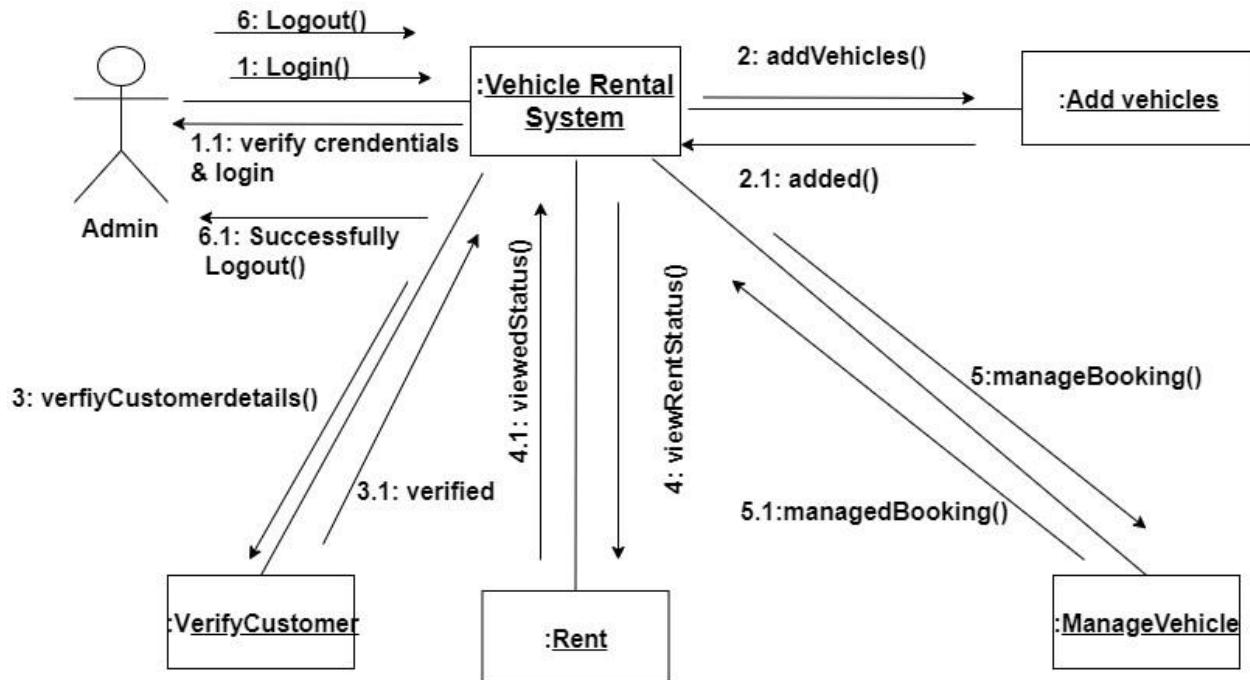


Figure 235: Initial Collaboration Diagram for Admin

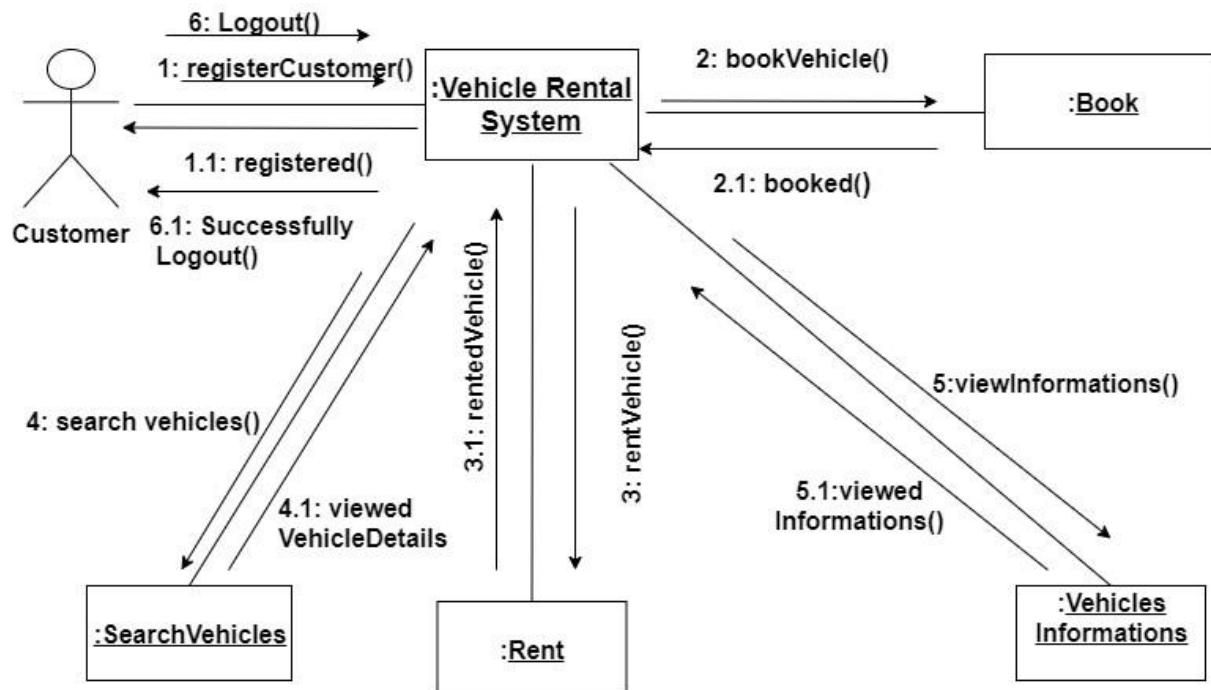


Figure 236: Initial Collaboration Diagram for Customer

ER Diagram

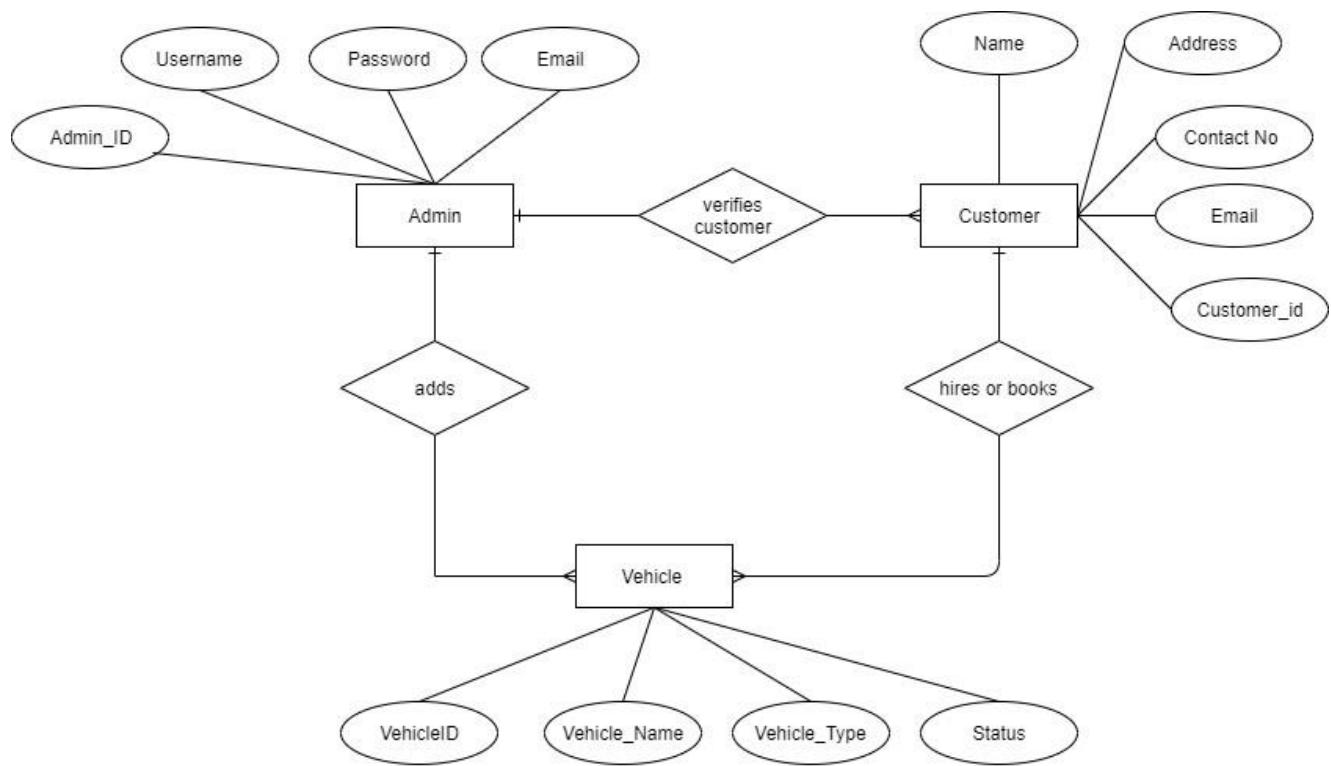


Figure 237: Initial ER Diagram

Initial Activity Diagram

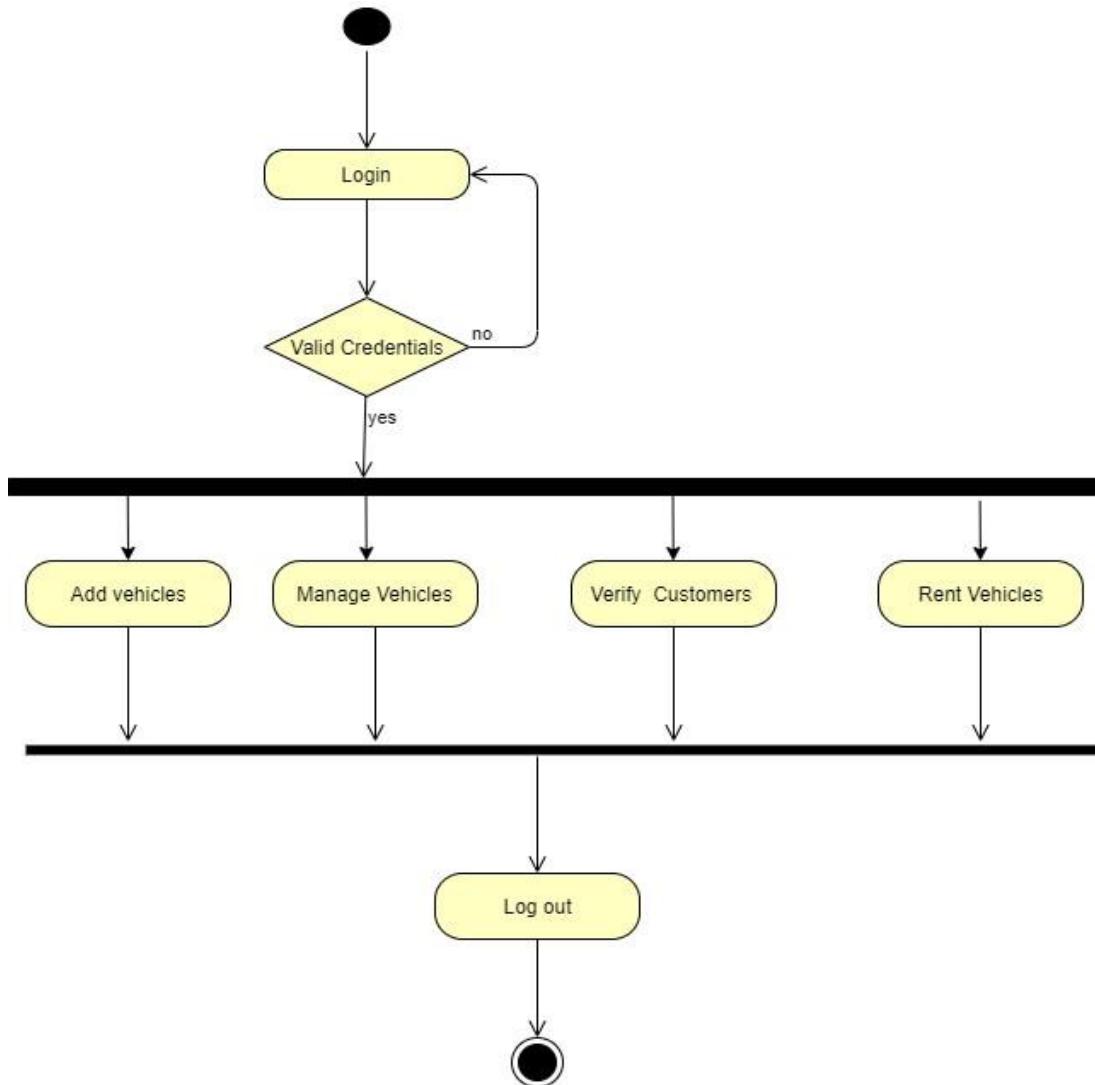


Figure 238: Initial Activity Diagram for Admin

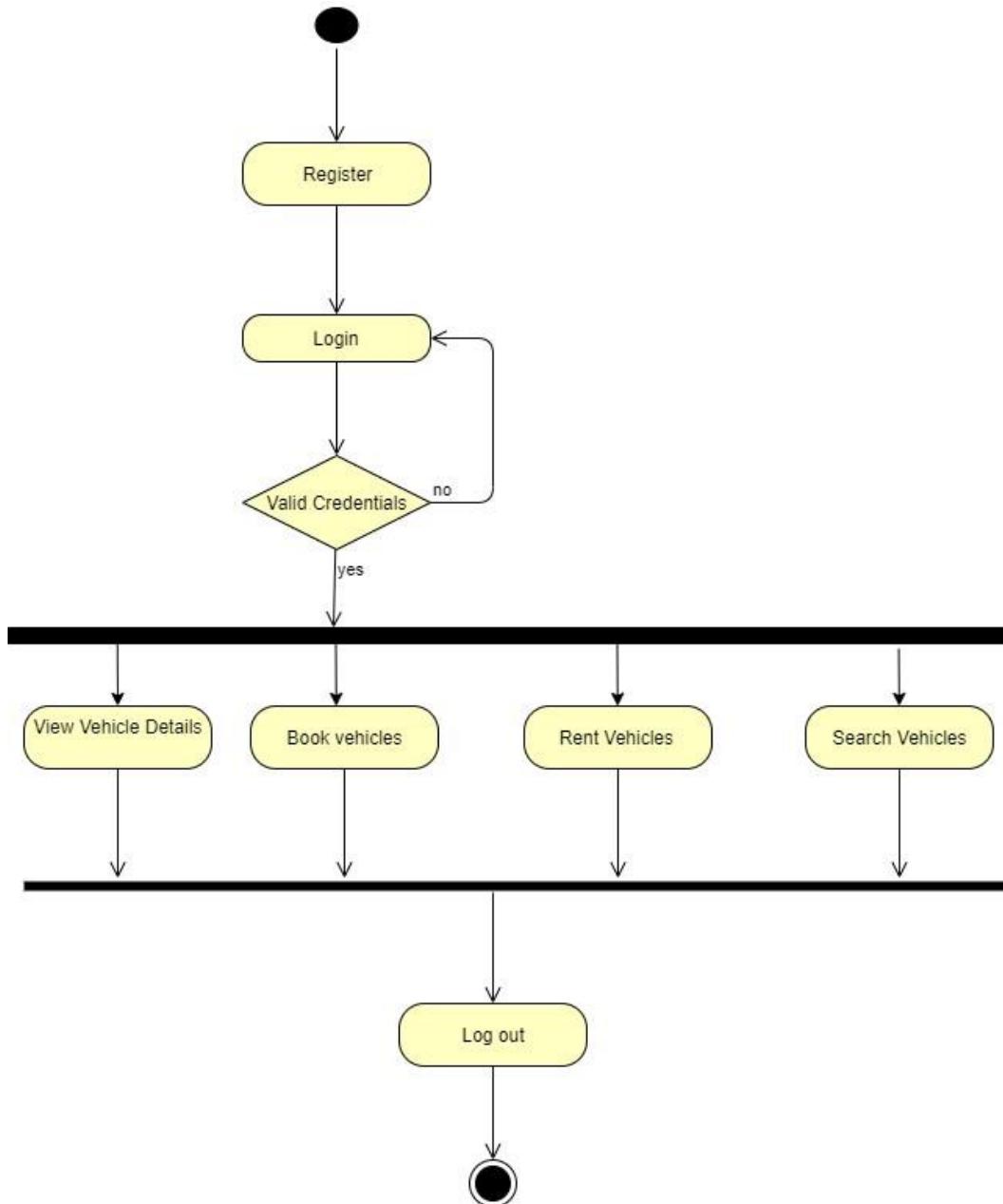


Figure 239: Initial Activity Diagram for Custom

8.5 Appendix E: Screenshots of the system

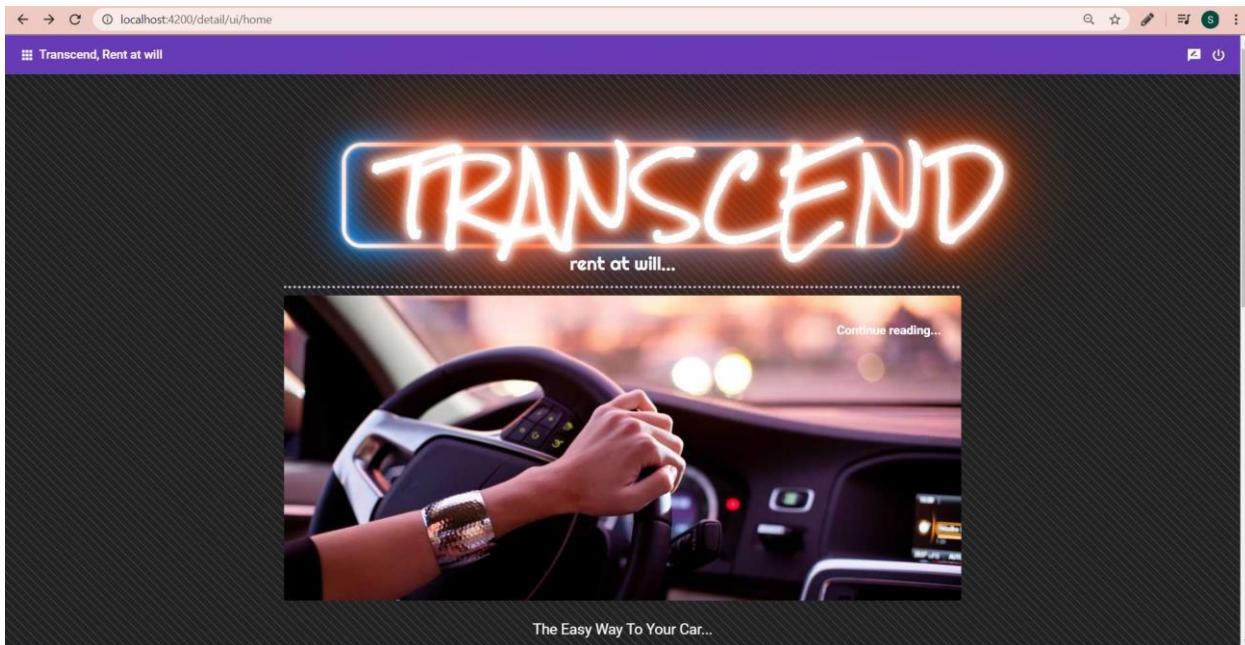


Figure 240: Home page

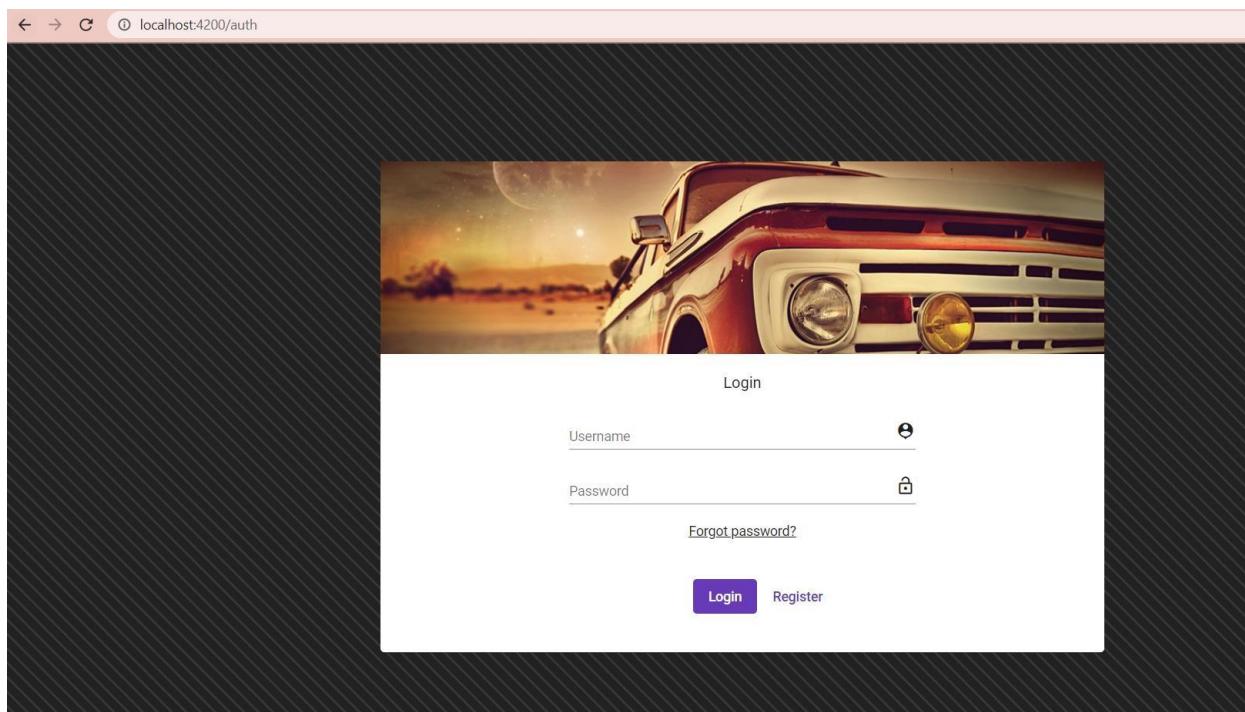
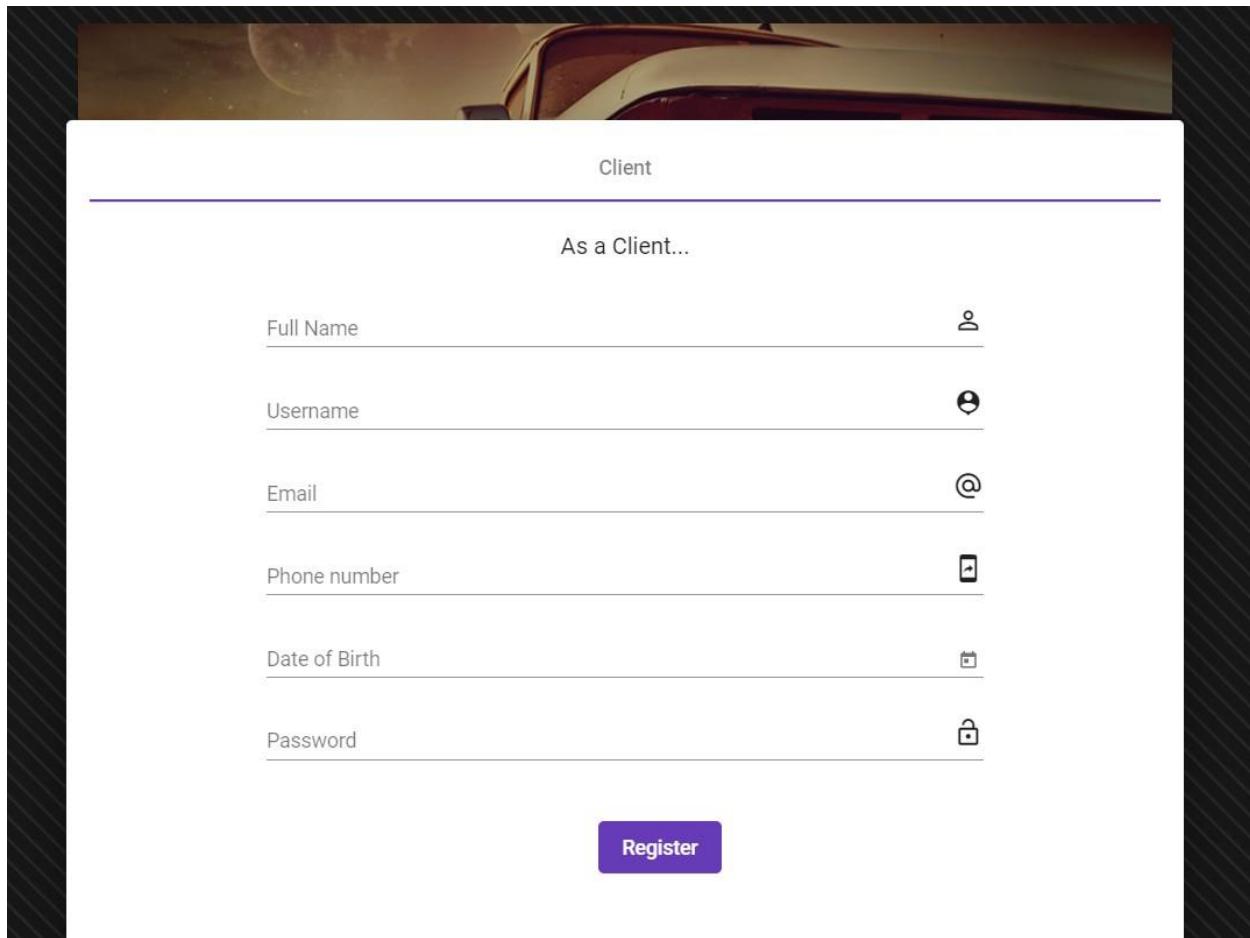
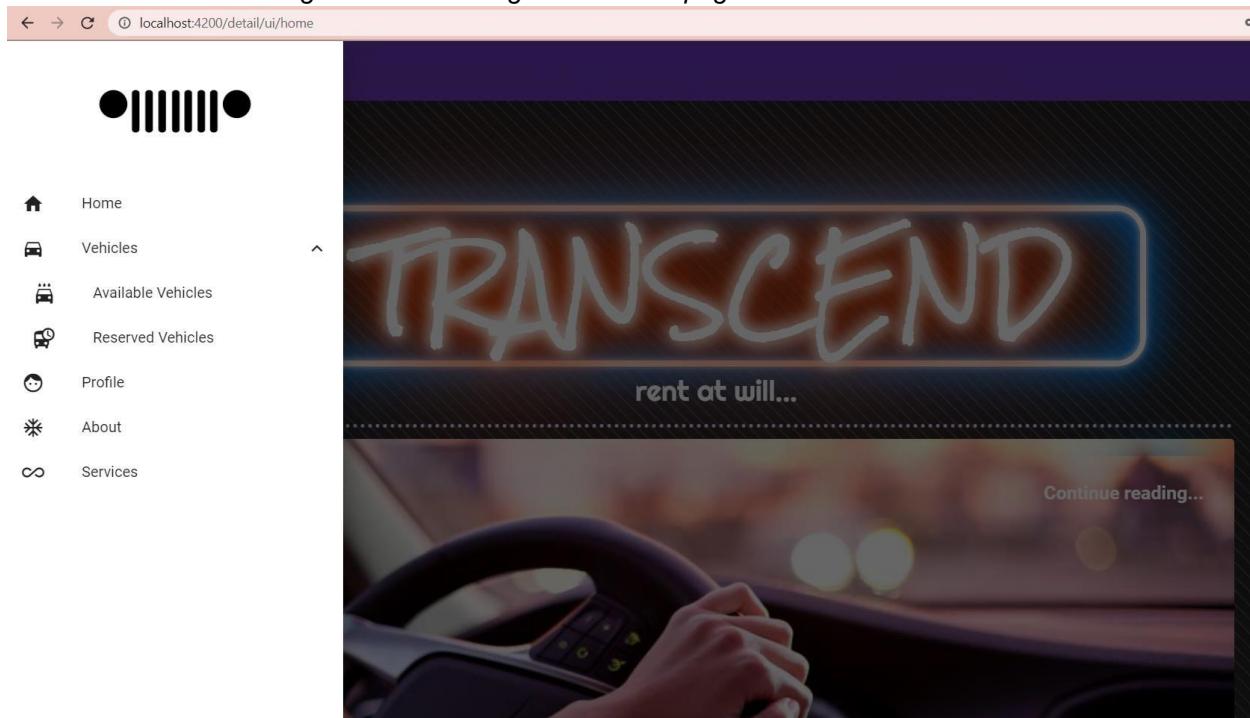


Figure 241: Login Page



The image shows a user registration form titled "Client". The form is divided into sections for "Full Name", "Username", "Email", "Phone number", "Date of Birth", and "Password". Each input field has a corresponding icon: a person icon for name, a user icon for username, an '@' symbol for email, a phone icon for phone number, a calendar icon for date of birth, and a lock icon for password. A "Register" button is located at the bottom of the form.

Figure 242: User registration form page



The image shows a dashboard interface for a vehicle rental service. On the left is a sidebar with icons and links: Home, Vehicles, Available Vehicles, Reserved Vehicles, Profile, About, and Services. The main area features a large banner with the word "TRANSCEND" in white, stylized letters, followed by the tagline "rent at will...". Below the banner is a photograph of a person's hands on a steering wheel, driving a car at night. A "Continue reading..." link is visible in the bottom right corner of the banner area.

247

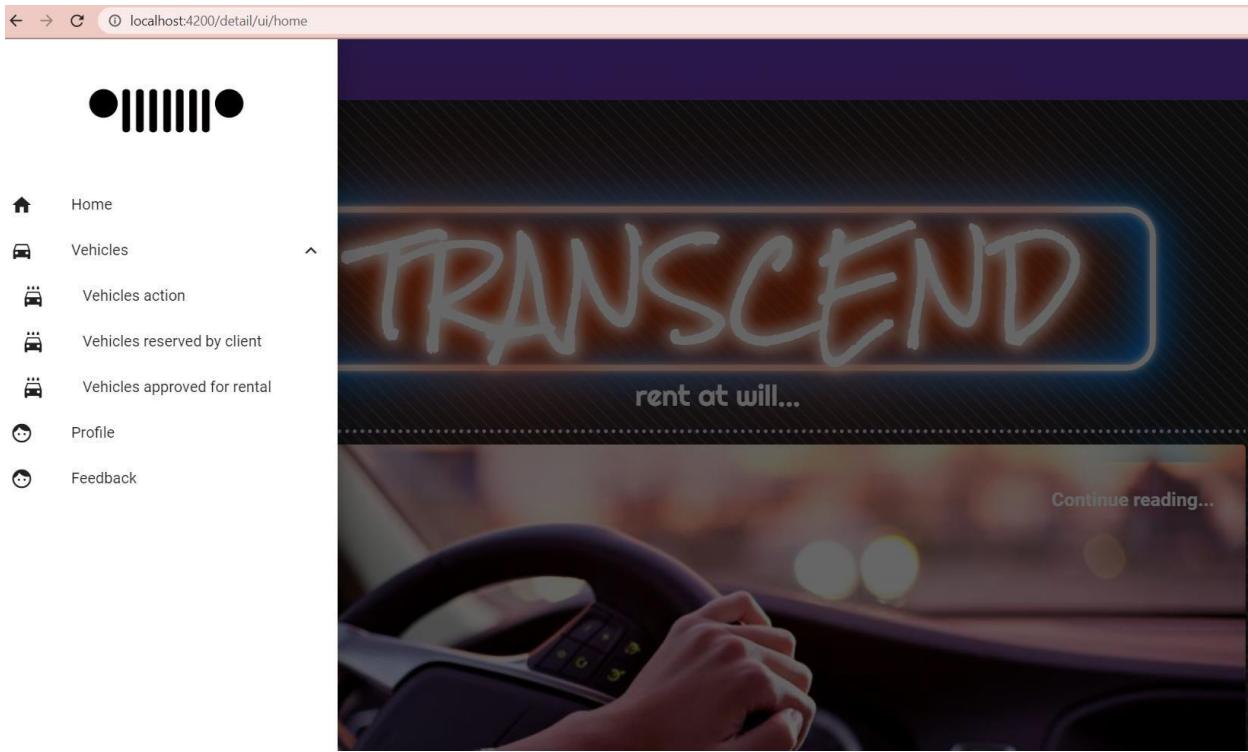
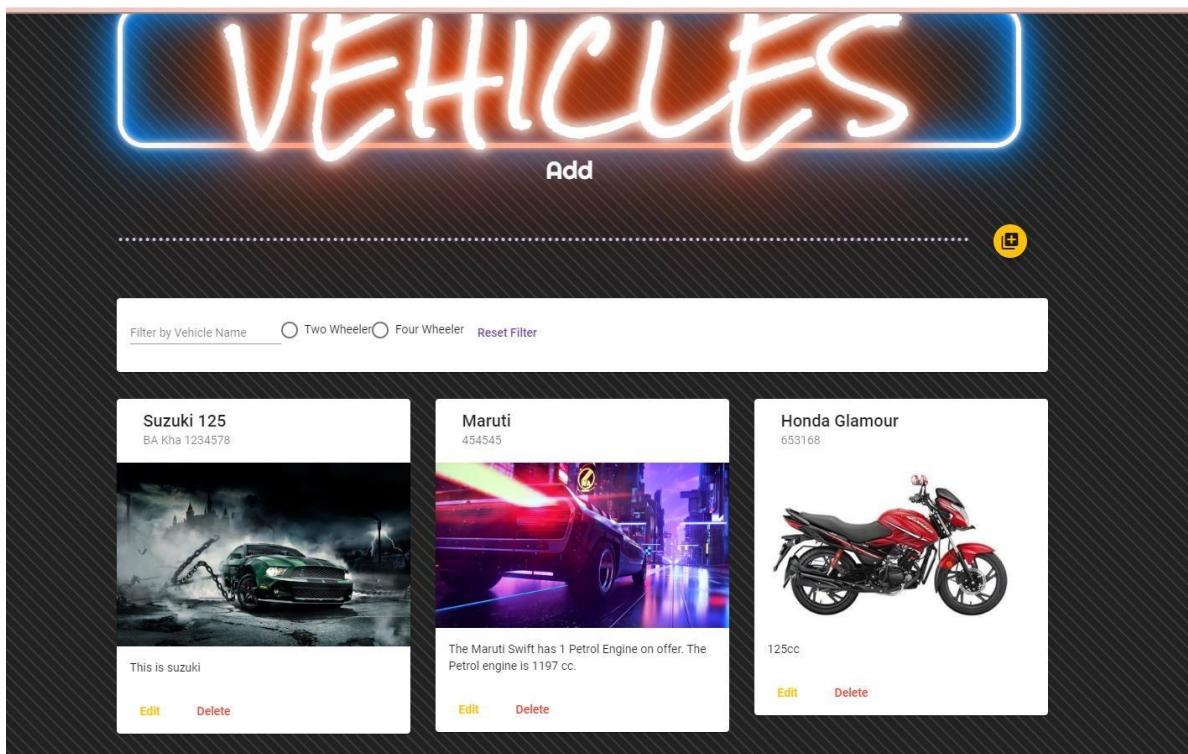
Figure 243: Customer dashboard*Figure 244: Admin dashboard*

Figure 245: Vehicle action page

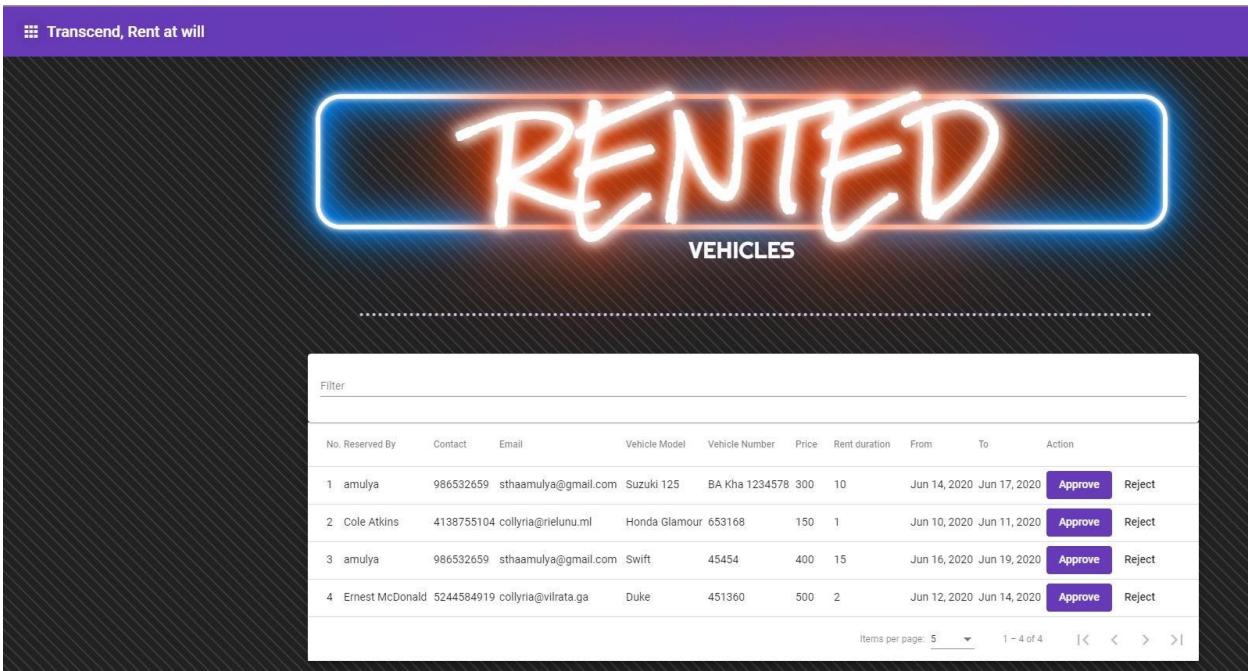


Figure 246: Vehicles reserved by client

The screenshot shows a web application interface titled "RENTED VEHICLES". At the top, there is a purple header bar with the text "Transcend, Rent at will". Below the header is a large, stylized "RENTED" sign with a blue and orange glow. A sub-header "VEHICLES" is centered below the sign. A search bar labeled "Filter" is present. The main content area displays a table of reserved vehicles:

No.	Reserved By	Contact	Email	Vehicle Model	Vehicle Number	Price	Rent duration
1	Ella	6498458595	collyria@k1069.com	Creta	56321	600	3

At the bottom right, there are buttons for "Items per page: 5" and navigation arrows.

Figure 247: Vehicle rented by client

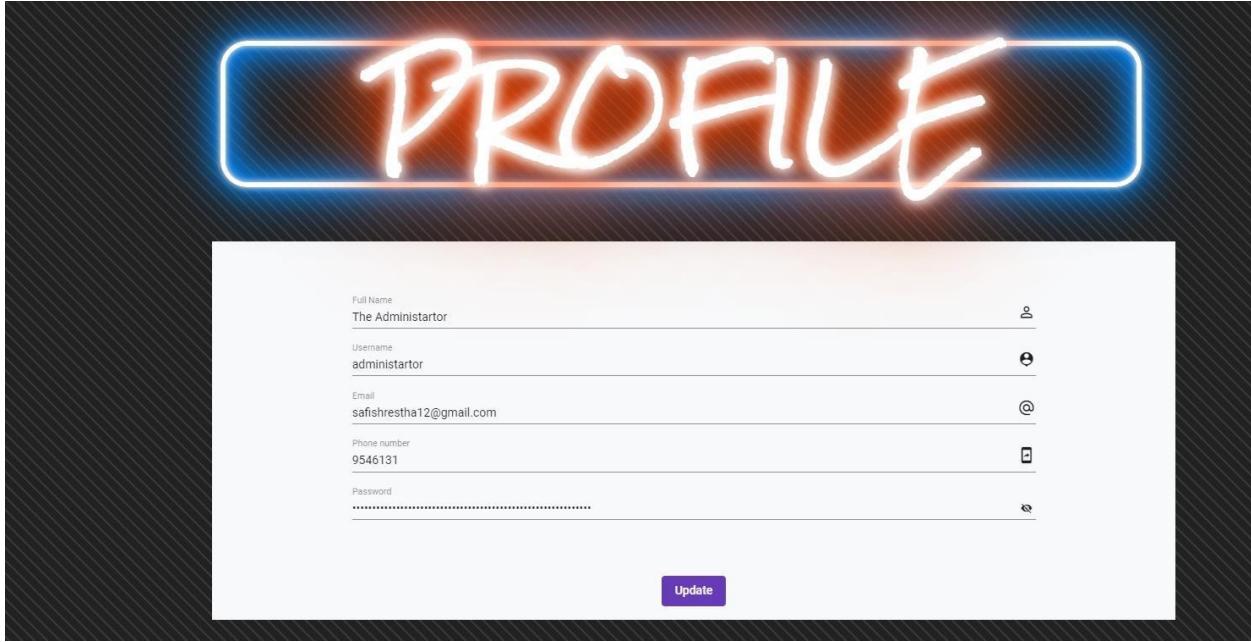


Figure 248: Admin Profile page

No.	User name	Feedback description
1	John Shrestha	this feedback
2	amulyadon	Service is good
3	Ella Weber	The prices are reasonable for vehicles
4	Roxie Matthews	This application provide great service
5	Beatrice Pittman	Prices are good

Figure 249: View Feedback page

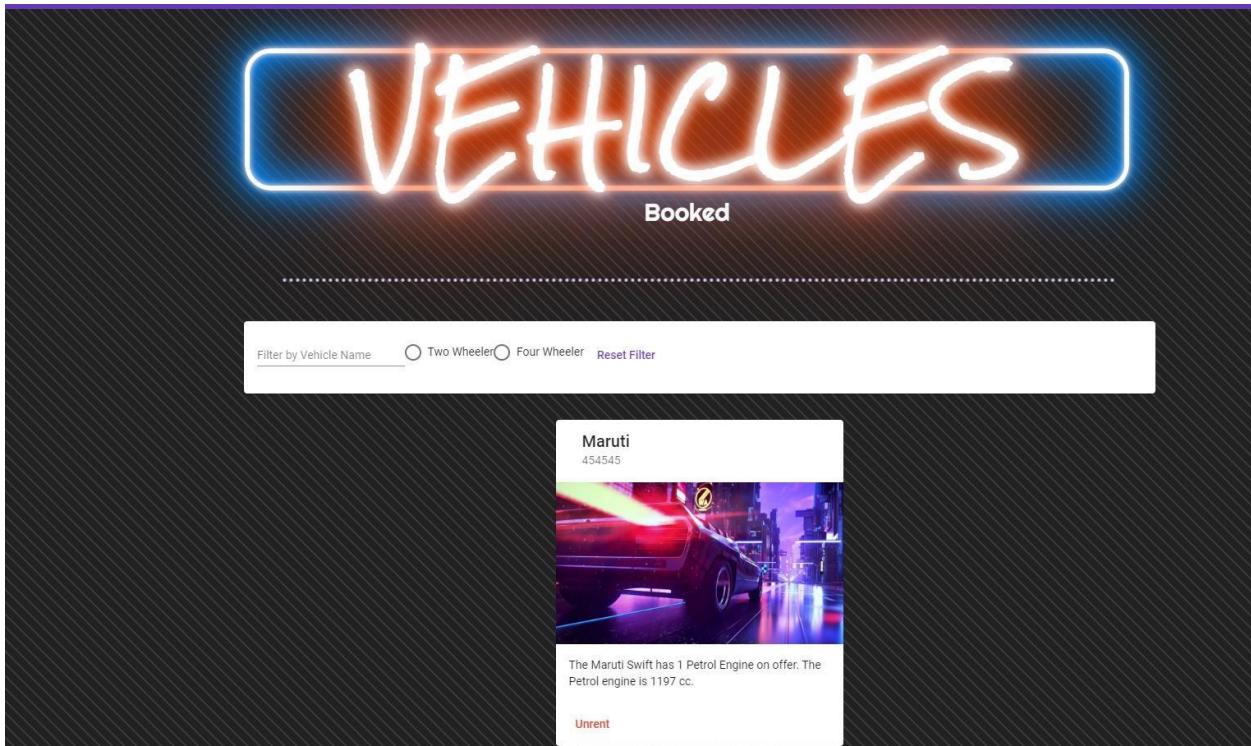


Figure 250: Reserved vehicles

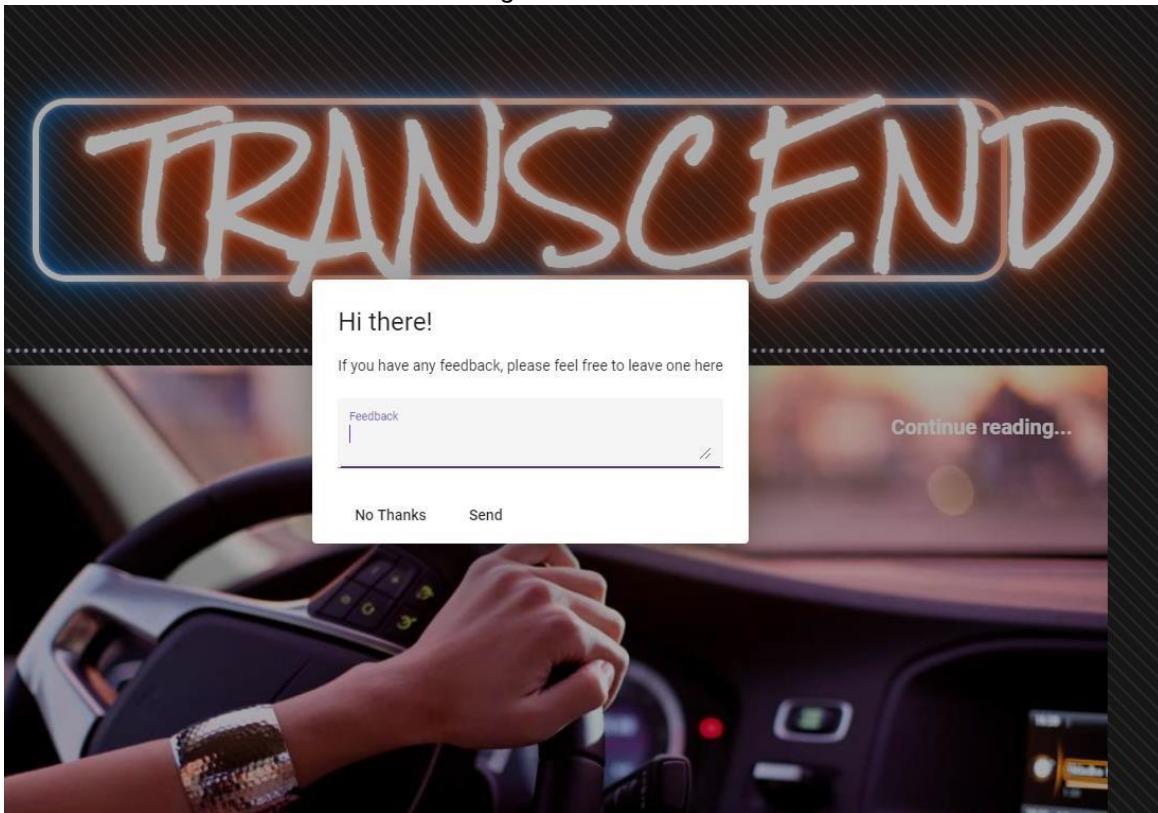


Figure 251: Feedback form

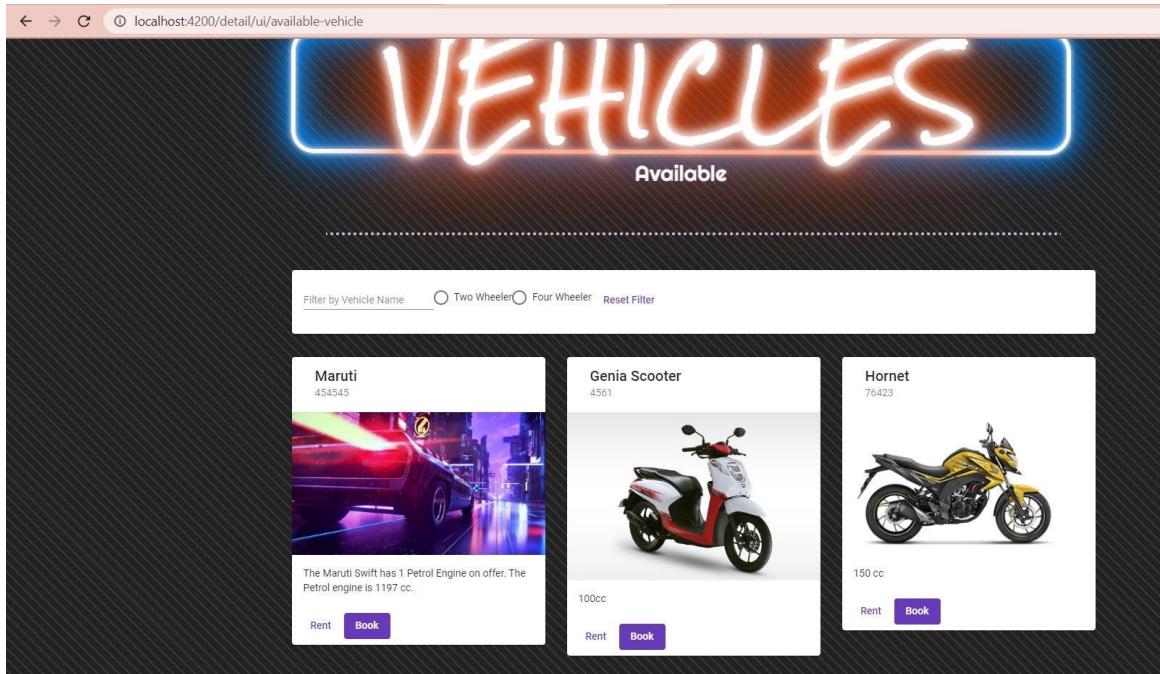


Figure 252: Available vehicle page

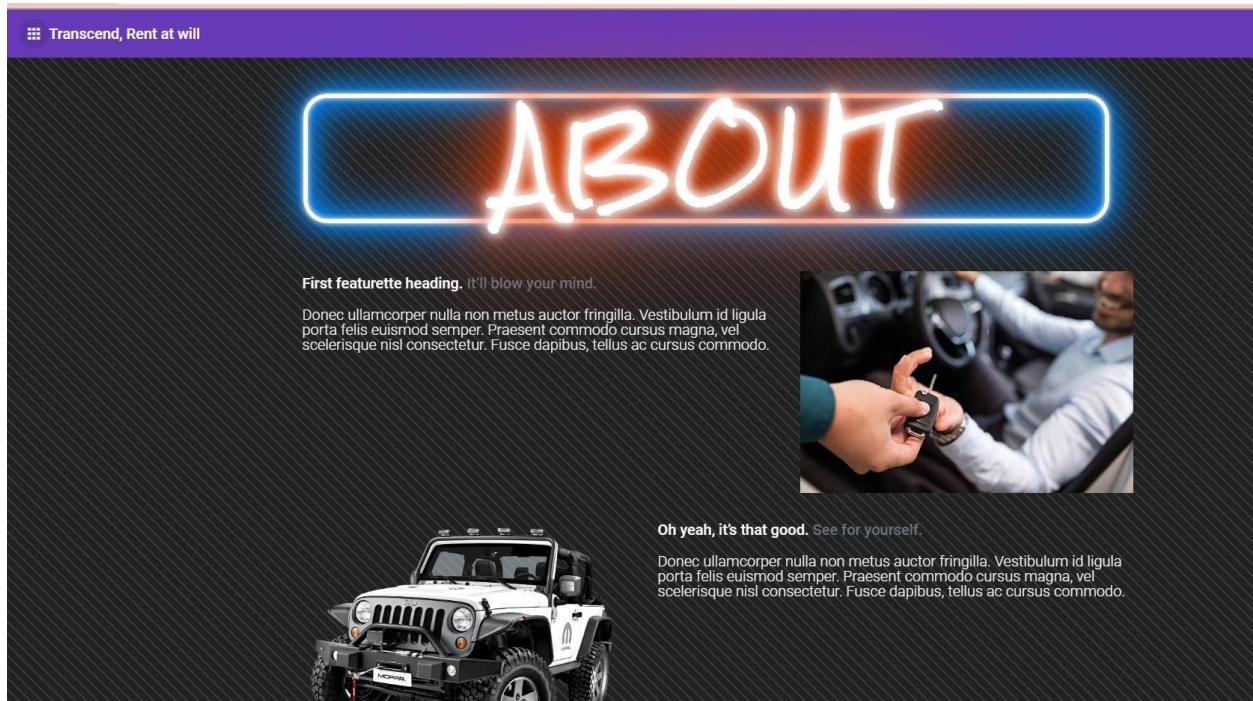


Figure 253: About page

8.6 Appendix F: User Feedback

8.6.1 Client Approval Letter

Madhu Mahat
Namaste Nepal Travels & Tours
Jadibuti, Kathmandu Nepal
15th Jan 2020

Safi Shrestha
Vehicle Rental System
Islington College
Kamalpokhari
Kathmandu

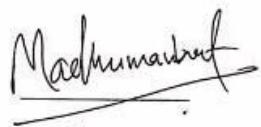
Subject: Permission Granted for the role of Client.

Dear Safi Shrestha,

I have reviewed your request for the role of client, and I am pleased to inform you that I have agreed to be your client for the project of the fitness application. Taking reference from our recent verbal meeting I am willing to provide the requirements and the necessary information that I need in the {FYP Project} application.

If you have any questions, or if I can be of further service to you, please call me or arrange a meeting for which we can discuss the issues and present the progress of the application development till date.

Sincerely,
Madhu Mahat
Vehicle Rental System



Signature

Phone Number: 9851073592

Email: madhumahat@gmail.com

Figure 254: Client approval letter

8.6.2 User feedback form

Untitled form

*Required

<p>Email address *</p> <p>Your email address</p>		<p>Do you think all the requirements and features were implemented properly ? *</p> <p><input type="radio"/> Yes <input type="radio"/> No</p>																																
<p>Full Name *</p> <p>Your answer</p>		<p>Overall are you satisfied with the final outcome and developed application? *</p> <table border="1"> <thead> <tr> <th>Excellent</th> <th>Very good</th> <th>Good</th> <th>Satisfactory</th> <th>Poor</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> </tr> </tbody> </table> <p>Rate the application</p>					Excellent	Very good	Good	Satisfactory	Poor	<input type="radio"/>																						
Excellent	Very good	Good	Satisfactory	Poor																														
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>																														
<p>Position in Namaste Nepal Travels and Tours *</p> <p>Your answer</p>		<p>How much are you satisfied with the User Interface(UI) of the application? *</p> <table border="1"> <thead> <tr> <th>Very Good</th> <th>Good</th> <th>Satisfactory</th> <th>Bad</th> <th>Poor</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> </tr> </tbody> </table> <p>Rate the UI</p>					Very Good	Good	Satisfactory	Bad	Poor	<input type="radio"/>																						
Very Good	Good	Satisfactory	Bad	Poor																														
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>																														
<p>How was the overall experience of using Vehicle Rental application? *</p> <table border="1"> <thead> <tr> <th>Excellent</th> <th>Good</th> <th>Satisfactory</th> <th>Poor</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </tbody> </table> <p>Experience</p>		Excellent	Good	Satisfactory	Poor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<p>After the success completion of this application, how beneficial do you think this application will be for customer? *</p> <table border="1"> <thead> <tr> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> <th>8</th> <th>9</th> <th>10</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/></td> </tr> </tbody> </table> <p>Not so useful Very useful</p>					1	2	3	4	5	6	7	8	9	10	<input type="radio"/>									
Excellent	Good	Satisfactory	Poor																															
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																															
1	2	3	4	5	6	7	8	9	10																									
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>																									
		<p>What feature of this application is most useful ? *</p> <p>Your answer</p>																																

Figure 255: User feedback form

8.6.3 Sample of filled User feedback forms

Email address *			
madhumahat@gmail.com			
Full Name *			
Madhu Mahat			
Position in Namaste Nepal Travels and Tours *			
Owner			
How was the overall experience of using Vehicle Rental application? *			
	Excellent	Good	Satisfactory
Experience	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Poor		

Do you think all the requirements and features were implemented properly? *

Yes
 No

Overall are you satisfied with the final outcome and developed application? *

	Excellent	Very good	Good	Satisfactory	Poor
Rate the application	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

How much are you satisfied with the User Interface(UI) of the application? *

	Very Good	Good	Satisfactory	Bad	Poor
Rate the UI	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

After the success completion of this application, how beneficial do you think this application will be for customer? *

1	2	3	4	5	6	7	8	9	10	
Not so useful	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Very useful						

What feature of this application is most useful? *

Overall all the features of the application is good and useful.

Figure 256: Filled User feedback form

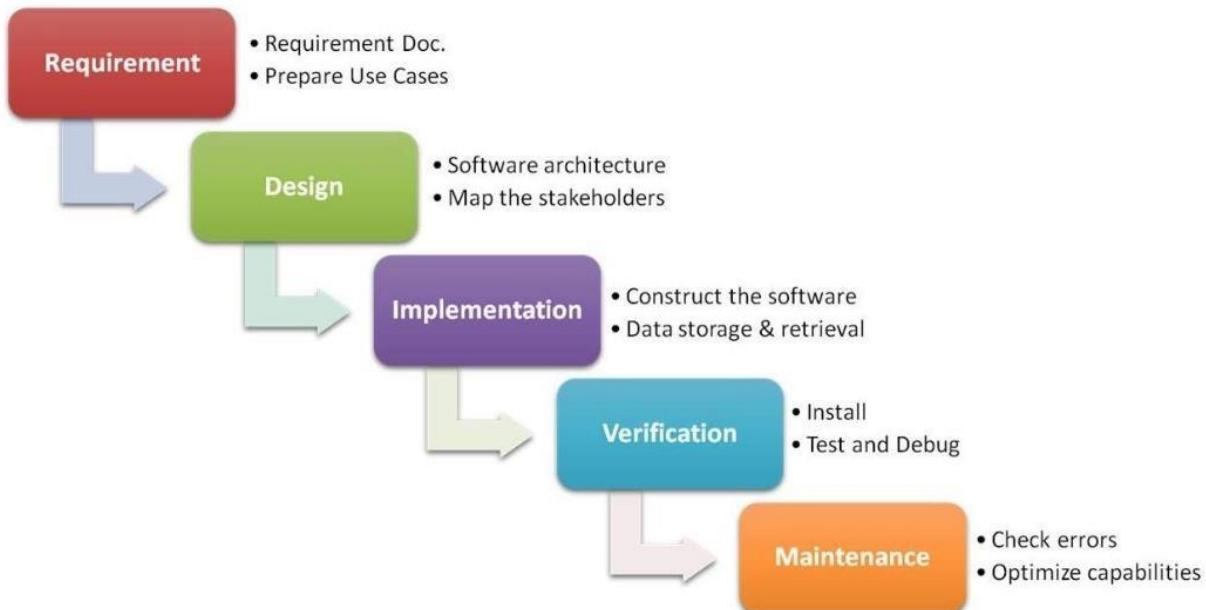
8.7 Appendix G: Development

8.7.1 Considered Methodology

Waterfall Model

The Waterfall methodology is considered a classic approach and is widely used during software development process. The entire approach complements the methodology name itself. As in waterfall, the development process is divided into multiple phases. Each phase occurs after the completion of the phase prior to it. The following figure depicts how Waterfall model functions. (AirBrake, 2020)

(Try QA, 2020)



(UKEssays, 2020)

Figure 257: Phases of waterfall methodology

- **Feasibility**

The potential requirements are identified and defined in this phase.

- **Analysis**

We produce proper models and business logic based on the system specifications identified in the previous phase.

- **Design**

A system design is generated where necessary software and hardware specifications are mentioned which is further utilized to define the system architecture.

- **Coding/Implementation**

The system design is further implemented to produce a viable source code which utilizes the models and logic generated in the previous phase.

- **Testing**

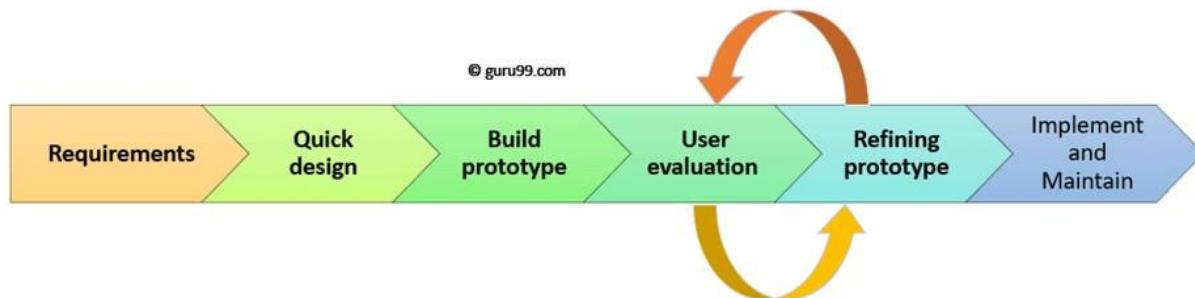
The source code is tested in units and merged into a functional system after successful testing of each unit.

- **Maintenance**

In case of unwanted issues in the client side, patches are used to overcome the encountered issues.

Prototype Model

Unlike other development methodologies mentioned above, prototype model kicks off with a prototype of the system being built before the requirements are specified in detail. A light concept of what the user expects is noted down to provide an experience to end user to some extent via a prototype as mentioned above. The user can then provide feedback in parts of the prototype itself for further modifications. This information is documented as specification for further development which is as per the standard of what the user is looking for and also minimizing any form miscommunications. (Java T Point, 2020) (searchcio, 2020)



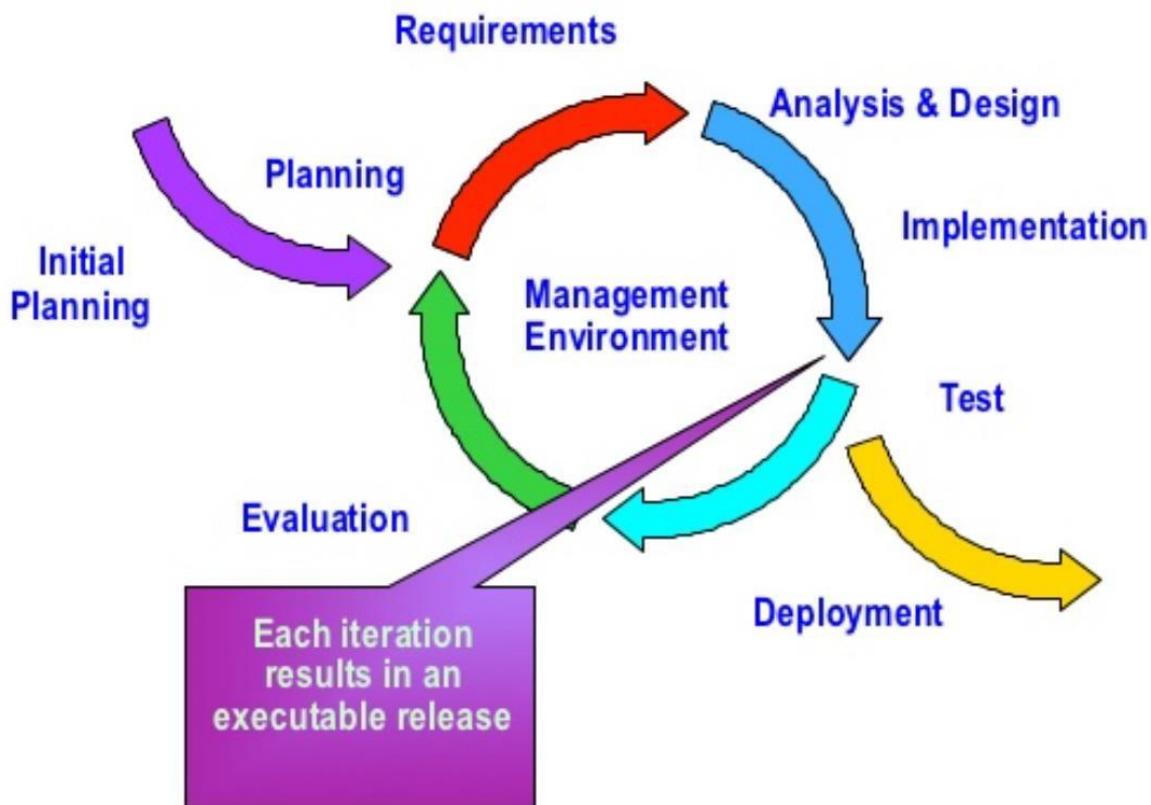
(guru99, 2020)

Figure 258: Phases of prototype methodology

8.7.2 Selected Methodology

Iterative and Incremental Model

Speaking of iterative and incremental model, we generally kick off with planning the development process which is followed by series of feedback session from the end user themselves. Corrections are made to previous development plans in an iterative as per the user feedback which can at times contain added requirements which is the incremental part. The diagram below depicts a simple iterative process carried out in software development. (Techopedia, 2020)



(LinkedIn Corporation, 2020)

Figure 259 : Phases of Iterative and incremental methodology

8.8 Appendix H: Initial Software Requirement Specification

Functional Requirements:

- New users can register themselves online and log in.
- Users can rent vehicles online and also make online reservations for vehicles
- Notification of approval or cancelation of booking or renting vehicles is provided to user.
- Users can give feedback after using the service.

Non-Functional Requirements:

- Users can view all pages of application.
- Users can view vehicles details.
- Users can Search vehicles.

Useable Requirements:

- Users can logout themselves from the application.
- Users can sign up.
- Users get notifications for confirmation of booking and renting vehicles.
- Users can give feedbacks.

8.9 Appendix I: Proposal Introduction

a. Problem Scenario

A country like Nepal where most of the population belongs to middle class family. Most of the people can't afford a car or bike. People are always on rush they have less time and more work so willing or unwilling they have to travel to different places for meeting,

work business etc. Public vehicle is not the best option as it is generally running late, crowded and one cannot reach to its destination with comfort.

b. Project as a solution

This application “Vehicle Rental System” comes up as a solution to the current scenario for people who can't afford vehicle. They can book a car trip, business meeting, and office to home, home to office or anywhere they want to travel. This application is an online system through which clients can view car and bikes, signup, login, view profile, book car. One who has license can drive themselves but in case they don't have license then driver will be assigned with car.

Aims and Objectives

Aims

To make rental system easy, comfortable and available for everyone in best deals fro travelling.

Objectives

- User can login and register themselves to rent a vehicle.
- Provides list of all vehicles available.
- Provides online vehicle reservation.
- To save customer time and provide them with best service in best prices.

Expected Outcomes and Deliverables

At the end of the project, a proper application is to be built that will make vehicle renting system easy, saves time of customer and make travelling comfortable for them. They can book vehicle online, select their destinations and in case customer don't know how to drive driver will be provided to the customer.

Project risks, threats and contingency plans

No matter how robust our system is, we somewhat manage to pack some risk when we build and refurbish a system. The unforeseen casualties somewhere in the future gets triggered when we ignore risks or leave it uncalculated. When we address potential risks in order to understand and analyze them, we refer to it as Risk Management. Keeping various aspects of the system in mind, we calculate the risk and manage them such that the system remains healthy and safe. The steps involved in risk management are as mentioned below.

- Risk identification
- Assess impact of risk
- Lessen critical risk
- Risk control

Likelihood	Value
Low	1
Medium	2
High	3

Consequences	Value
Very Low	1
Low	2
Medium	3
High	4
Very high	5

With strong reference to the tables mentioned above, we calculate the impact of a specific risk with respect to the formula: **Impact = Likelihood * Consequences**

In case of my project, the following risk management matrix is presented which follows certain Risk Management standards.

Risk Type Description		Likelihood	Consequence	Impact	Action
Non- Technical	Natural Disaster	1	5	5	Consulting cloud backup (Dropbox, One drive etc.)
	Deadline Overruns	2	4	8	Following a schedule development for every y t phases.
	Unforeseen growth of project scope	1	5	5	Keeping the project codes and implementation methodology simple as such for future viable changes.

	Gold plating – including features irrelevant to project scope	2	4	8	Properly going through user requirements and only including required features.
Technical	Encountering unwanted bugs and glitches	3	4	12	Investing more time and budget on testing for possible bugs and glitches.
	Hardware failure	2	5	10	Always having backup hardware components just in case.
	Complex algorithms resulting memory leakage and CPU overhead	2	5	10	Consulting conventional algorithms to achieve goals with minimum casualties.
	Hard-disk and backup failure	1	5	5	Backing up important files in cloud.

Methodology

Incremental and iterative methodology will be used for this project.

Resource Requirements

Following are the tools that are to be used while developing this application.

IDE – IntelliJ IDEA 2018.3.5

Programming language – Typescript/ JavaScript

Frameworks used – Angular 8 (Front-end)

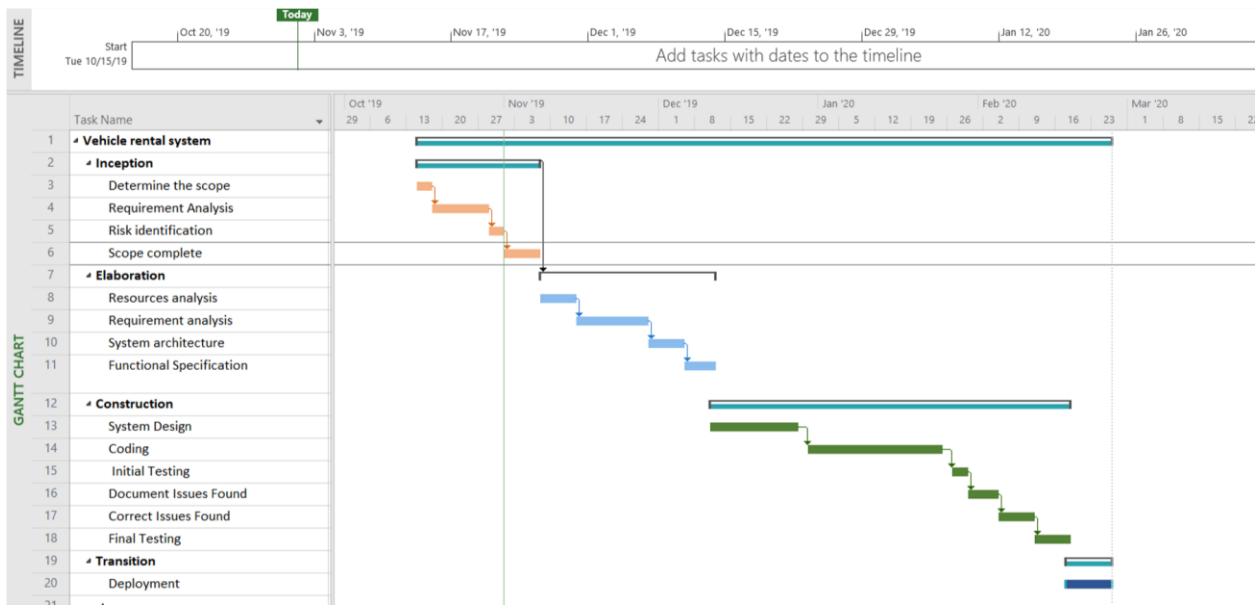
Modelling tool – draw.io,balsamiq

Work Breakdown Model

WBS Task Description		Estimated time in days
0	Interactive tutorial platform	123
1	Feasibility study and Planning	15
1.1	Scoping	5
1.2	Proposal Writing	10
2	Analysis	29
2.1	Achieve functional requirements	11
2.2	Achieve non-functional requirements	11
2.3	Use case and scenario description	7
3	Design	26

3.1	Behavioral Modelling	12
3.1.1	Sequence diagram	7
3.1.2	Activity diagram	5
3.2	Structural Modelling	9
3.2.1	Class diagram	6
3.2.2	ER diagram	3
3.3	UI design	5
4	Implementation	31
4.1	Reviewing functional specifications	5
4.2	Application development	14
4.3	Implementation of MVVM pattern	5
4.4	Development of backend and database	7
5	Testing	7
5.1	Perform unit testing	4
5.2	Perform Integration testing	3
6	Documentation	15
6.1	Report writing	15

Gantt chart



Conclusion

The primary objective of this project is to create an Online Vehicle Rental System that user can use to enhance their travelling experience, in this case, the application will provide details about the vehicles and customer can rent the vehicle available as per their choice in best prices. This project will also help me to learn about new tools and technologies and help me to further increase my skills.