**Ride Management Application for Bajajs in**

**Gondar City**

**by**



**University of Gondar**

**Faculty of Informatics**

**Department of Information System**

**Group Members**

Name Id No

1. Assefa Demses 00737/12
2. Dereje Aragaw 02236/12
3. Fikir Getu 01627/12
4. Firaol Teklu 01156/12
5. Henok Gashew 01181/12

A Group Project

Submitted to the Department of Information Systems, Faculty of Informatics, University of Gondar, in meeting the preliminary project requirement for partial fulfillment of the award of Bachelor of Science Degree in Information Systems.

Gondar, Ethiopia

Date/Year: March 2023

**Approval Sheet**

This Group Project entitled “Ride Management Application for Bajajs in Gondar City” has been read and approved as meeting the preliminary project requirements of the Department of Information Systems in partial fulfillment for the award of a Bachelor of Science degree in Information Systems, University of Gondar, Gondar, Ethiopia.

**Approved by:**

1. Name of Advisor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature:\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_
2. Name of Project Coordinator: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_

# **Acknowledgments**

First and foremost, We would like to thank God. Then We would like to express our deepest gratitude to Mrs. Tigist Ayeragn of the Transport Bureau of Gondar City and ZayRide’s Gondar Branch for providing the necessary information for the project. We would also like to give special thanks to our advisor Teshome Aychluhm for his tremendous support and guidance throughout the project.

Finally, we would like to thank everyone whose names were not mentioned but who helped us throughout the project.

Table of Contents

[**Acknowledgments** i](#_Toc132111667)

[**Executive project summary** v](#_Toc132111668)

[**CHAPTER ONE** 1](#_Toc132111669)

[**Introduction** 1](#_Toc132111670)

[**1.1.** **Background of the project** 1](#_Toc132111671)

[**1.2.** **Statement of Problem** 1](#_Toc132111672)

[**1.3.** **Project objectives** 1](#_Toc132111673)

[**1.3.1.** **General Objective** 1](#_Toc132111674)

[**1.3.2.** **Specific Objective** 1](#_Toc132111675)

[**1.4.** **Scope of the project** 2](#_Toc132111676)

[**1.5.** **Limitations of project** 2](#_Toc132111677)

[**1.6.** **System Development Methodology** 2](#_Toc132111678)

[**1.6.1.** **System development Approach** 2](#_Toc132111679)

[**1.6.2.** **System Development Tools** 3](#_Toc132111680)

[**Hardware tools** 3](#_Toc132111681)

[**Software tools** 3](#_Toc132111682)

[**Implementation languages** 3](#_Toc132111683)

[**1.7.** **Significance of Project** 4](#_Toc132111684)

[**1.8.** **Beneficiaries of the project** 4](#_Toc132111685)

[**1.9.** **Feasibility Study** 4](#_Toc132111686)

[**1.10.** **Project schedule** 5](#_Toc132111687)

[**1.11.** **Project Budget** 6](#_Toc132111688)

[**CHAPTER TWO** 7](#_Toc132111689)

[**2.** **Requirement Analysis** 7](#_Toc132111690)

[**2.1.** 7](#_Toc132111691)

[**2.1.** **Current System Description** 7](#_Toc132111692)

[**2.1.1.** **The major function of the current system** 7](#_Toc132111693)

[**2.1.2.** **The problem with the Existing System** 7](#_Toc132111694)

[**2.2.** **Requirement Gathering** 7](#_Toc132111695)

[**2.2.1.** **Requirement Gathering Methods** 7](#_Toc132111696)

[**2.2.2.** **Business Rules** 8](#_Toc132111697)

[**2.3.** **Proposed System Description** 8](#_Toc132111698)

[**2.3.1.** **Overview** 8](#_Toc132111699)

[**2.3.2.** **Functional Requirements** 9](#_Toc132111700)

[**2.3.3.** **Nonfunctional requirements** 10](#_Toc132111701)

[**2.3.3.1.** **Performance** 10](#_Toc132111702)

[**2.3.3.2.** **Scalability** 10](#_Toc132111703)

[**2.3.3.3.** **Availability** 10](#_Toc132111704)

[**2.3.3.4.** **Reliability** 10](#_Toc132111705)

[**2.3.3.5.** **Maintainability** 10](#_Toc132111706)

[**2.3.3.6.** **Security** 10](#_Toc132111707)

[**2.3.3.7.** **Environmental** 11](#_Toc132111708)

[**2.3.3.8.** **Usability** 11](#_Toc132111709)

[**Chapter Three** 12](#_Toc132111710)

[**3.** **System Model** 12](#_Toc132111711)

[**3.1.** **Scenario** 12](#_Toc132111712)

[**3.1.1.** **Use Case Model** 12](#_Toc132111713)

[**3.1.2.** **Use Case Diagram** 2](#_Toc132111714)

[**3.1.3.** **Description of Use Case Model** 2](#_Toc132111715)

[**3.1.4.** **Activity Diagram** 14](#_Toc132111716)

[**3.1.5.** **Object Model** 15](#_Toc132111717)

[**3.1.6.** **Data Dictionary** 15](#_Toc132111718)

[**3.1.7.** **Class Model** 15](#_Toc132111719)

[**3.1.8.** **Dynamic Modeling** 15](#_Toc132111720)

[**3.1.9.** **User Interface** 15](#_Toc132111721)

[**Chapter Four** 16](#_Toc132111722)

[**4.** **System Design** 16](#_Toc132111723)

[**4.1.** **Introduction** 16](#_Toc132111724)

[**4.2.** **Current software architecture (if any)** 16](#_Toc132111725)

[**4.3.** **Proposed software architecture** 16](#_Toc132111726)

[**4.3.1.** **System Decomposition** 16](#_Toc132111727)

[**4.3.2.** **Hardware/ software mapping** 16](#_Toc132111728)

[**4.3.3.** **Persistent data modeling** 16](#_Toc132111729)

[**4.3.4.** **Access control and security** 16](#_Toc132111730)

[**4.3.5.** **Detailed class diagram** 16](#_Toc132111731)

[**4.3.6.** **Package Diagram** 16](#_Toc132111732)

[***4.3.7.*** ***Deployment*** 16](#_Toc132111733)

# **Executive project summary**

# **CHAPTER ONE**

# **Introduction**

## **Background of the project**

Currently in the city of Gondar transportation is one of the hot issues. More than the availability of transportation, the Cost and security of transport are major sources of concern. Meter taxi applications like Taxiye, ZayRide, and others present a solution to the security problem but they are very costly. Further such systems are inapplicable in the city of Gondar as a result of their costliness.

## **Statement of Problem**

The main reason for the inapplicableness of already existing systems like ZayRide or Zoble Ride is first they are developed in wide cities with low transportation availability signified by long lines of taxi queues in mind and are mostly developed for cars. And the even bigger problem is their initial count of around 90 birrs. In a city like Gondar, the longest transport journey costs a maximum of 30 birrs with a taxi and 35 birrs with a bajaj.

Since it's impractical to pay 90 birrs or more for a trip that costs 30 birrs or so and since only cars register meter taxi services in a city with a larger number of bajaj than cars that is one big problem area.

The second problem area is the availability and security of bajaj transport, cost of a private contract driver bajaj costs a minimum of 100 birrs and even that is not readily available. They are signified by a large variance in cost between drivers and a low sense of security in passengers as a result of the current rise in the number of criminal activities.

## **Project objectives**

## **General Objective**

The general objective of the project is to develop a system that communicates with bajaj drivers and riders providing availability at the exact time and place of need and at a cost that is acceptable to both parties.

## **Specific Objective**

The specific objectives the project aims to achieve and meet are the following

* Collect data, analyze, design system requirements and identify problems in the existing systems in detail
* Design a new system that solves the identified problems
* Design an easy-to-use and responsive user interface that allows users to track details of their orders and travel information
* Create a backend store that stores the users’(driver and rider) information
* Evaluate the system

## **Scope of the project**

The various activities performed by the system are; it allows for bajaj drivers to register their services, users to create an account and register to use the services, an interface to select start and destination, and a payment page that shows the travel details and cost of the travel according to system fare agreed to by both parties.

The other function provided by the system is for frequent users to keep a list of favorite locations and frequently traveled places for selection.

## **Limitations of project**

The system does not provide an electronic payment system and follows cash on delivery of service model. This limitation is currently not addressed because the crediting system is a large system on its own or it requires the implementation of banking contracts. This can be addressed in future improvements and updates.

## **System Development Methodology**

## **System development Approach**

The chosen development methodology for its adaptiveness and with the limitation of time in mind is Rapid Application Development(RAD).

* Allows to move back and forward between design phases, for updating and changing
* Easy to make improvements to the design and implementation
* Important issues addressed before final system delivery
* Allows for faster code generation

The drawback of this methodology is it needs huge collaborations and joint efforts from more than the development team. The other main problem is flexibility, since it's easy to change and update t requirements coming from users the changes made may end up dragging the project beyond the scheduled scope.

## **System Development Tools**

The tools that were used for the execution of the project both hardware and software are as follows

### **Hardware tools**

* **laptop Computer**: - To store files, prepare documentation, install and use required software for development
* **Internet Connector (broadband cable)**: - To get relevant information from the internet.
* **Printer:** - helps to print documentation
* **Mobile phone:** - to check the design implementation of the system
* **USB type-c cable: -** to connect the development PC with the mobile phone
* **Mouse**

### **Software tools**

* **Window 10:** - This Operating system is to be used to install and run the development software needed for this project.
* **Browser:** - like chrome, used for retrieving information for designing and developing and also used for running designing tools like.
* **Microsoft Word:** -For writing documentation.
* **Microsoft PowerPoint:** -will be used for preparing the presentation of this project documentation.
* **Draw.io:** - used to design the UML diagram for the project.
* **Visual Studio code:** - code editor used for implementation.
* **Figma:** - for UI/UX designing
* **GitHub:** - version controlling and team collaboration tool, also used to contact the advisor.

### **Implementation languages**

## **Significance of Project**

After the completion of the project, the system developed will provide the listed important services to the following parties:

For Bajaj drivers:

* Solve the availability problem of passengers
* Provide places where their services are required
* Ensure their security
* Provide means to save fuel consumption
* Provide means for work as a side job

For passengers:

* Make bajaj readily available when the need arises
* Ensure their security
* Removes the need to stand by the side of the road and wait for bajaj
* Pick their start and finish location and know how much it cost beforehand

## **Beneficiaries of the project**

We can see our beneficiaries as direct beneficiaries and indirect beneficiaries.

Our direct beneficiaries are:

* Bajaj drivers
* Passengers
* The owning company that gets profit from the users

Our indirect beneficiaries are:

* The government further divided to
  + The financial sector (Tax and revenue minister)
  + Regional police force
  + The community of Gondar

## **Feasibility Study**

**Economic Feasibility**

The proposed system is economically feasible because the opportunity cost of the proposed system on the market is very compensating in the long run. Since the market is open for the service, the production and deployment cost is high but the financial gain to be extracted from it is higher.

**Technical Feasibility**

In terms of technical feasibility, the project uses already existing technologies and concepts we have already learned. Thus, we have both the technology and the required personnel for it.

**Operational Feasibility**

The proposed project is operationally feasible because the earlier identified problems are addressed to the best extent. Further, the users will be presented with an easy-to-use UI hence promoting use.

## **Project schedule**

The project schedule is more extended than initial estimations. none the less the schedule Gantt chart is developed with the initial given start date. The chart was developed using an online tool called `teamgantt.com`.

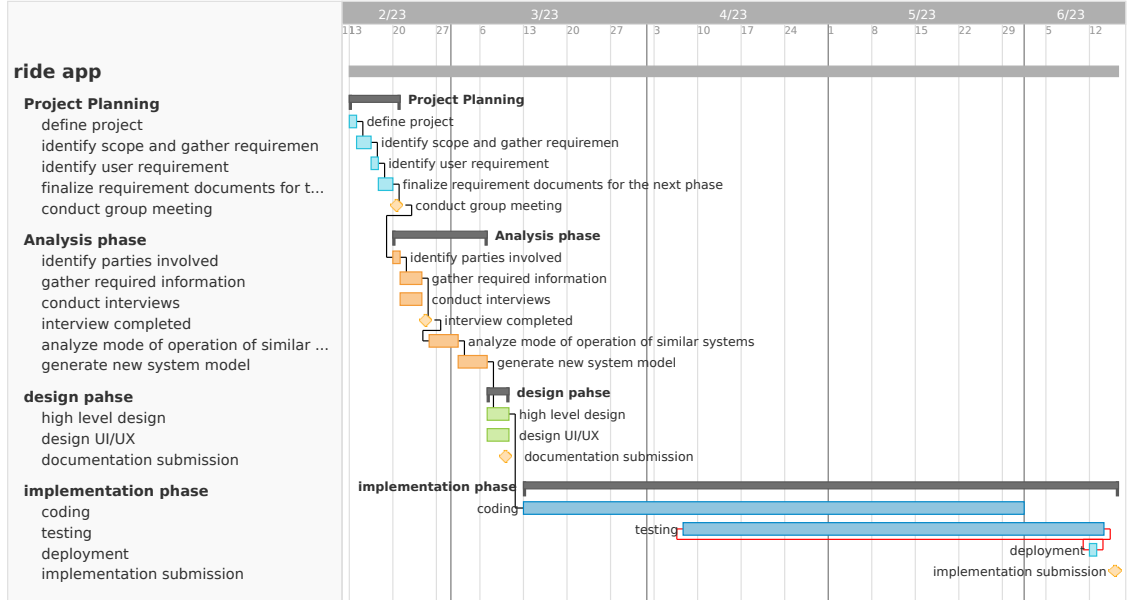


Figure 1.1: Schedule Gantt Chart

## **Project Budget**

**The budget for the project for tangible expenses is collected from a team member, but most technology expenses written are used on a trial basis thus their costs are not incurred expenses but rather estimated expenses.**

|  |  |  |  |
| --- | --- | --- | --- |
| *Item* | Quantity | Unit Price | Total cost in birr |
| *Paper* |  | ETB 2 | ETB 50 |
| *Google API* |  |  |  |
| *Transport cost* | 5 | ETB 20 | ETB 100 |
| *Total* |  |  | 480 |

# **CHAPTER TWO**

1. **Requirement Analysis**

## 

## **Current System Description**

The existing transport system is a manual one. From the passenger’s point of view whoever needs to use a transport goes directly to the road and waits for a transport to come whether a taxi or bajaj. After that, if they have any luggage, they go back to their houses for loading transportation or go out with their luggage to the road, sometimes if they have the phone numbers of bajaj drivers they call from home. As for the driver’s point of view, they drive around looking for passengers or stay in place waiting for a client to call or come talk to them.

## **The major function of the current system**

* Negotiable pricing
* Found everywhere
* Doesn’t require literacy

## **The problem with the Existing System**

* Time wasting
* No record of travels
* Prone to criminal activities like theft, kidnapping, rape, and murder
* Costs more than the set tariff
* not accessible to everyone
* the feeling of insecurity by both driver and riders

## **Requirement Gathering**

## **Requirement Gathering Methods**

The methodologies that were used to gather information for the development of the project are interview and document analysis.

**Interview**

As an interview, we went to the transport bureau of Gondar city and talked with \*\*\*\*\*\*\*.we asked for the required information and we got referred to Mrs. Tigist Ayeragn. And from the information we got, we found out about the existence of ZayRide in Gondar and went to their office too.

**Document analysis**

Since there are similar ride applications, we used their documents to learn and understand more about the project requirements. we found documents on the mode of operations of systems from abroad.

We reviewed various diagrams and documentation(developers’ and users’) from Uber.

## **Business Rules**

A business rule is a statement that defines or constrains some aspect of the business. [1] we have listed the business rule statements below.

* Drivers must follow traffic laws and drive safely at all times.
* Riders must treat drivers and other riders with respect and refrain from any behavior that makes others feel unsafe or uncomfortable.
* Drivers and riders are prohibited from using drugs or alcohol while using the platform.
* Both drivers and riders must adhere to community standards and policies, including those related to discrimination, hate speech, and harassment.
* Drivers and riders are encouraged to report any safety concerns or issues
* A driver currently a client must identify himself as occupied
* A driver can not serve more than one order at a time without the rider’s permission
* A rider can not refuse payment after a service
* A rider refusing to pay is enforceable by law
* A rider cannot cancel an order after the order has started

## **Proposed System Description**

## **Overview**

We are in a technology era. This is signified by large access to smartphones, the internet, and other technologies. In this technology era, by exploiting this access a lot of countries and cities have changed the way they operate a lot of things from manual to computer-based. The transportation system is one of them. Similarly, many cities in Ethiopia have adopted systems like rides as a possible alternative for transport.

That being said such systems have not been implemented in Gondar city administration. The current high demand for transportation and high number of thefts related to bajaj transports, as more kidnappings related to bajaj make such systems appear in demand.

The system we propose is similar to that of the others currently operational in Addis Ababa and others. The extra capability we propose is for it to support bajaj drivers as service givers. The proposed system is signified by live GPS tracking of the trip, easy UI/UX both for both the rider and driver, storing the history of travel details, and they can know how much distance they are traveling and how much is left.

The proposed system will have big impact on the tourism of Gondar by bot international and local tourists because it will provide convenient transportation, especially for those who have never visited the historical city. Thus, our system will play great role on the secure and sustainable development of Gondar.

The other support our system may provide will be in the work of Gondar city administration’s police. In Gondar crimes involving bajaj are high. Although it is declining in the central part of the city it is the same as ever in the outer skirts of the city. The help our system will provide here is it will put location tags and details on both registered drivers and riders. As a result if any crime or accident occurred involving the users the system will help the police identify each individual.

## **Functional Requirements**

The functional requirements of the proposed system are classes of functions that have further subfunctions and are listed as follows.

* **GPS tracking:** the system must keep track of location details with the identity of users at given locations. The data it is to keep track of is as follows:
  + **Riders**: location details of riders currently ordering a ride, their pickup and drop off location
  + **Drivers:** location details of drivers available for orders and on duty
  + **Trip:**  start location and finishing location of an ordered trip
* **User details registering:**  the users identified are both the riders and the drivers, the system keeps track of both users' detailed information for controlled sharing on later orders.
* **Calculating trip cost:** calculate fare based on factors more than travel distance

## **Nonfunctional requirements**

## **Performance**

In terms of performance, the system proposed is aimed to address various aspects to the maximum possible. The first is number of user requests. Others are fast response time and efficiency, which are designed to work by demanding low CPU, memory, and the already existing operating system on the device.

## **Scalability**

As for the scalability aspect, the system proposed will have an elastically scalable character, this will be good for current small user starts and still give the intended service regardless of increasing user traffic on the system.

## **Availability**

The availability of the system is a server-side concern, as long as the server that’s leased is up the system is designed to be up 24/7 and the availability of night shift bajaj drivers becomes the key controller too.

## **Reliability**

The system proposed is not designed to have failures raising from prolonged use because of intended periodic updates. But worst case, if it was to happen it can be reported via user feedback and resolved promptly.

## **Security**

Since the account creation process is designed to work with OTP and not create individual accounts with usernames and passwords the security issue is not a big concern. All that is because accounts are created with phone numbers and OTP(one-time password) confirmation numbers sent directly to the registering users' phone number.

## **Environmental**

According to environmental nonfunctional requirements, the system we propose doesn’t even need any actions. It uses already existing technologies like smartphones. Thus, it doesn’t affect the environment significantly making the requirement of addressing green computing issues and reducing carbon footprint motions useless.

## **Usability**

The design of the UI is aimed to be easy for all users whether proficient with smartphones or not. It's designed to be easily understandable with clear instructions and steps of application.

# **Chapter Three**

1. **System Model**

## **Scenario**

## **Use Case Model**

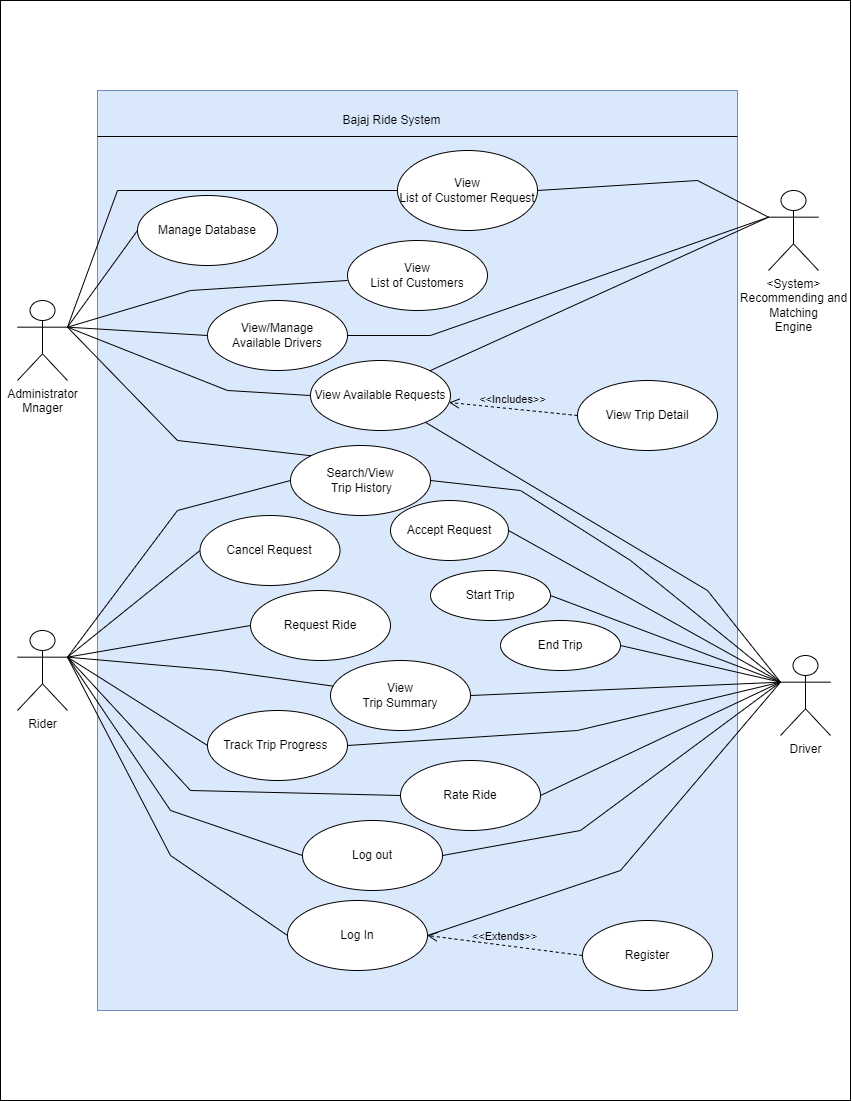
Actors**:** In the use cases an actor interacts with the system to perform a piece of meaningful work that helps them to achieve a goal and has access to define their overall role in the system and the scope of their action. Depending on the above explanation, the actors in this system are the following:

* Driver: These are individuals who use the Bajaj ride app to accept ride requests, pick up riders, and provide transportation services.
* Riders: These are individuals who use the Bajaj ride app to request rides and make payments for the service.
* Admins: These are individuals who are responsible for managing the Bajaj ride platform, including overseeing driver and rider accounts, monitoring safety and security, and maintaining the technical infrastructure.

Use cases: Use Case represents the interaction between a user (human or machine) and the system. which can include another Use Case's functionality or extend another Use Case with its behavior. The most important and basic use cases of this system are the following: -

1. View list of customer request
2. Manage database
3. View list of customers
4. View and manage available drivers
5. View available requests
6. View trip detail
7. Search and view trip history
8. View trip summary
9. Track trip progress
10. Rate ride
11. Log out
12. Log in
13. Register
14. Request ride
15. Cancel ride
16. Accept ride
17. Start ride
18. End ride

## **Use Case Diagram**



## **Description of Use Case Model**

Table 3.1: Use case description for Login

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_001 | |
| **Use Case Name** | Login | |
| **Actor** | Rider, Driver | |
| **Description:** | Users can log in to their account with their registered phone number and OTP password to access the Bajaj ride app. | |
| **precondition** | User must have a registered account with Bajaj ride. | |
| **Post condition** | User is logged in and has access to the app's features. | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪User opens the Bajaj ride app. 2. 🡪 User enters their registered pone number 3. 🡨 user enters given otp as password | 1. 🡨 System sends OTP to the entered phone number. 2. 🡨 System verifies password and logs the user into their account and displays the app's home screen. |
| **Alternative Flows** | 1. If the user enters an incorrect password, the system will display an error message and prompt the user to try again. 2. If the user enters an unregistered phone number, the system will take the user to account setup interface. | |

Table 3.2: Use case description for Request ride

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_002 | |
| **Use Case Name** | Request ride | |
| **Actor** | Rider | |
| **Description:** | Riders can send a request for a ride. | |
| **precondition** | Rider must be logged in to their Bajaj ride account. | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 Rider opens the Bajaj ride app. 2. 🡪 Rider selects the "Where to?" option. 3. 🡪Rider enters their destination. 4. 🡪Rider confirms the ride request 5. 🡪Driver accepts the ride request. | 1. 🡨System displays the estimated cost and wait time for the ride. 2. 🡨System searches for a nearby available driver and assigns the ride to them. 3. 🡨System provides the rider with the driver's name, photo, and vehicle information. 4. 🡨System notifies the rider that the driver is on their way. 5. 🡨 engine updates list of active requests |
| **Post condition** | Rider's ride request has been sent and assigned to a driver. | |
| **Alternative Flows** | None | |

Table 3.3: Use case description for Track trip progress

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_003 | |
| **Use Case Name** | Track trip progress | |
| **Actor** | Rider, Driver | |
| **Description:** | Riders and drivers can track the ride's progress and estimated time of arrival to the destination through the Bajaj ride app. | |
| **precondition** | Actors must have an active ride request. | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 Rider opens the Bajaj ride app. 2. 🡪 Rider selects the "Track Ride" option. | 1. 🡨 System displays the ride's progress and ETA to the user. |
| **Post condition** | Rider is able to track the ride's progress and ETA. | |
| **Alternative Flows** | None | |

Table 3.4: Use case description for Rate ride

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_004 | |
| **Use Case Name** | Rate ride | |
| **Actor** | Rider, Driver | |
| **Description:** | Riders can rate their driver and ride experience after their ride has ended. | |
| Driver rates the ride experience exchanged with riders. | |
| **precondition** | Riderand driver must complet ride the ride. | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 user opens the Bajaj ride app. 2. 🡪user selects the "Rate Ride" option. 3. 🡪 user selects the rating for the driver. 4. 🡪user may provide optional comments to provide feedback | 1. 🡨 System displays the ride details and prompts the user to rate their trip and the other user(rider or driver) on a scale of 1-5. 2. 🡨 System records the rating and feedback and updates the user’s profile with the rating. |
| **Post condition** | Rider's rating and feedback has been recorded and the driver's profile has been updated. | |
| **Alternative Flows** | None | |

Table 3.5: Use case description for View and manage available drivers

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_005 | |
| **Use Case Name** | View and manage available drivers | |
| **Actor** | Administrator Manager, Engine | |
| **Description:** | System automatically updates available drivers list. | |
| administrator can modify or delete drivers on the available drivers list. | |
| **precondition** | None | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 administrator views results and aproves | 1. 🡪 system check drivers list and active trips list. 2. 🡪 system updates available drivers list from the above lists |
| **Post condition** | Available drivers list is updated for request assignment. | |
| **Alternative Flows** | 1. 🡪 administrator modifies the resulting available drivers list | |

Table 3.6: Use case description for Manage database

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_006 | |
| **Use Case Name** | Manage database | |
| **Actor** | Administrator | |
| **Description:** | The admininistration manager manages database of entire system | |
| **precondition** | none | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 manager access, modify and maintatin system database |  |
| **Post condition** | Database maintained | |
| **Alternative Flows** | none | |

Table 3.7: Use case description for View trip summary

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_007 | |
| **Use Case Name** | View trip summary | |
| **Actor** | Rider,driver | |
| **Description:** | Rider and driver view entire summary of their trip. | |
| **precondition** | The trip must be finised. | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 user opens the Bajaj rider app. 2. 🡪rider and driver view finised trip information | 1. 🡪System displayes summary page |
| **Post condition** | Rider's continue to rating and payment | |
| **Alternative Flows** | None | |

Table 3.8: Use case description for Search and view trip history

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_008 | |
| **Use Case Name** | Search and view trip history | |
| **Actor** | Administrator, rider, driver | |
| **Description:** | Users view their past trip history and detailed information. | |
| Administrator views trip history of all customers. | |
| **precondition** | Trips must have been conducted in the past | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 user openes bajaj ride application. 2. 🡪 user goes to history page | 1. 🡪 System shows individual trips list |
| **Post condition** | None | |
| **Alternative Flows** | 1. 🡪 administrator opens all conducted trips from database | |

Table 3.9: Use case description for Register as Driver

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_009 | |
| **Use Case Name** | Register as driver | |
| **Actor** | Driver | |
| **Description:** | Users can apply to become a Bajaj ride driver. | |
| **precondition** |  | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 User opens the Majaj ride app. 2. 🡪 User selects the "Become a Driver" option. 3. 🡪 User provides their personal and driver information. 4. 🡪 System Administration verifies the user's information and approves their driver application. |  |
| **Post condition** | User's driver application is approved. | |
| **Alternative Flows** | 3a. If the user's driver information is incomplete or inaccurate, the system prompts the user to update their information.  4a. If the user's driver application is rejected, the system notifies the user and provides a reason for the rejection. | |

Table 3.10: Use case description for View Request

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC\_010 | | |
| **Use Case Name** | View available requests | | |
| **Actor** | Driver, Engine, administrator | | |
| **Description:** | Driver can view available or assigned ride requests. | | |
| Administrator and the system view list of all active ride requests that haven’t been complited. | | |
| Engine updates the available request list when passengers make requests. | | |
| **precondition** | Driver must be logged into their Bajaj ride account. | | |
| User must make ride requests. | | |
| **Main Flow** | **User action** | | **System response** |
|  | 1. 🡪 Driver opens the Bajaj ride Driver app. 2. 🡪 Driver selects the "View Requests" option. 3. 🡪 Driver can view ride requests and their details, such as pick-up location, drop-off location, and fare amount. | |  |
| **Post condition** | Driver has viewed their ride requests. | | |
| **Alternative Flows** | 1. 🡪Rider makes request 2. 🡪 |  | |

Table 3.11: Use case description for Accept Request

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_011 | |
| **Use Case Name** | Accept request | |
| **Actor** | Driver | |
| **Description:** | Driver must be logged into their Bajaj ride account. | |
| **precondition** | Driver must have viewed a ride requests. | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 Driver selects the assigned request. 2. 🡪 Driver confirms that they want to accept the ride request. 3. 🡪 Driver is directed to the pick-up location of the rider. |  |
| **Post condition** | Driver has accepted the ride request. | |
| **Alternative Flows** | 2a. 🡪driver declines the asigned request .  2b🡪 driver select request manually | |

Table 3.12: Use case description for Start Ride

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_012 | |
| **Use Case Name** | Start ride | |
| **Actor** | Driver | |
| **Description:** | Driver start the ride on meeting the rider to. | |
| **precondition** | Driver must have accepted a ride request. | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 Driver arrives at the pick-up location. 2. 🡪 Driver confirms that the rider is present. 3. 🡪 Driver selects the "Start Ride" option. 4. The ride officially begins and the system starts tracking the ride. |  |
| **Post condition** | Driver has started the ride and both users can track progress. | |
| **Alternative Flows** | None | |

Table 3.13: Use case description for End Ride

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_013 | |
| **Use Case Name** | End ride | |
| **Actor** | Driver | |
| **Description:** | Driver can end the ride once rider reaches the requested final destination. | |
| **precondition** | Driver must have started a ride. | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 Driver arrives at the drop-off location. 2. 🡪 Driver confirms that the rider has been dropped off. 3. 🡪 Driver selects the "End Ride" option. 4. 🡪 The rider pays the fare with cash. | 1. 🡨 The system calculates the fare amount and displays it to the driver and rider. |
| **Post condition** | Driver has ended the ride and rider has paid the fare. | |
| **Alternative Flows** | 2a. If the rider requests to change the drop-off location, the driver can select the "Change Destination" option and follow the updated directions.  4a. If there is a dispute over the fare amount, the driver and rider can contact Bajaj ride customer service for assistance. | |

Table 3.14: Use case description for View Earnings

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_014 | |
| **Use Case Name** | Cancel ride | |
| **Actor** | Rider | |
| **Description:** | The rider cancels the ride request before the driver reaches or before the trip starts. | |
| **precondition** | Trip request must be made. | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 The rider opens bajaj ride application. 2. 🡪 rider choose “cancel” option | 1. 🡪 . show rider current order. 2. 🡪 The system update available request list |
| **Post condition** | The rider has canceled request.  The system notifies the driver of cancelation. | |
| **Alternative Flows** | none | |

Table 3.15: Use case description for Register

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_015 | |
| **Use Case Name** | Register | |
| **Actor** | Rider | |
| **Description:** | New users that setup their account after verifying their phone numbers with OTP. | |
| **precondition** | User login with new phone numbers. | |
| **Post condition** | User is logged in andis taken to account setup page. | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪User enteres phone number and OTP . 2. 🡪 User enters name, email, image(optional) | 1. 🡨 application takes user to account creation page. 2. 🡨 name and email are saved in database |
| **Alternative Flows** | 1. If the user enters an invalid email, the system will display an error message and prompt the user to try again. 2. If the user enters an image of >2MB, the system will ask user to try another picture. | |

Table 3.16 Use case description for View trip detail

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_016 | |
| **Use Case Name** | View trip detail | |
| **Actor** | Driver, administrator, engine | |
| **Description:** | Driver sees the deatail of a request before accepting ride. | |
| **precondition** | Request must be selected | |
| **Post condition** | Driver accept or decline ride request | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪driver selects the request. . 2. 🡪 driver sees details of the request | 1. 🡨 application shows details page |
| **Alternative Flows** | None | |

Table 3.17 Use case description for View list of customers

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_017 | |
| **Use Case Name** | View list of customers | |
| **Actor** | Administrator | |
| **Description:** | Administrator views list of all users registered on the system. | |
| **precondition** | None | |
| **Post condition** |  | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪administrator enters credentials 2. 🡪 accesses database and sees detailed information of all customers. | 1. 🡪 system verifies credentials and allows access |
| **Alternative Flows** | None | |

Table 3.18 Use case description for View list of customer request

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_018 | |
| **Use Case Name** | View list of customer request | |
| **Actor** | Administrator, engine | |
| **Description:** | View list of all customer requests(finished, ongoing, active, canceled)  The engine views this list to sort or update other related lists | |
| **precondition** |  | |
| **Post condition** |  | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 administrator requests accesses all list of requests 2. 🡪 administrator/ engine reads table | 1. 🡪 system grants access to administrator |
| **Alternative Flows** |  | |

Table 3.19 Use case description for Log out

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_019 | |
| **Use Case Name** | Log out | |
| **Actor** | Rider, Driver | |
| **Description:** | The users log out of the system to stop the service. | |
| **precondition** | User must be logged in | |
| **Post condition** |  | |
| **Main Flow** | **User action** | **System response** |
|  | 1. 🡪 user chooses the log out option 2. 🡪 user confirms to log out | 1. 🡪 system asks for confirmation. 2. 🡪 user services are terminated and logged out 3. 🡪 log in page viewed |
| **Alternative Flows** | 3a. 🡪 user denies log out  4a. 🡪 system cancels log out processes and goes back | |

## **Activity Diagram**

## **Object Model**

## **Data Dictionary**

## **Class Model**

## **Dynamic Modeling**

## **Sequence diagram**

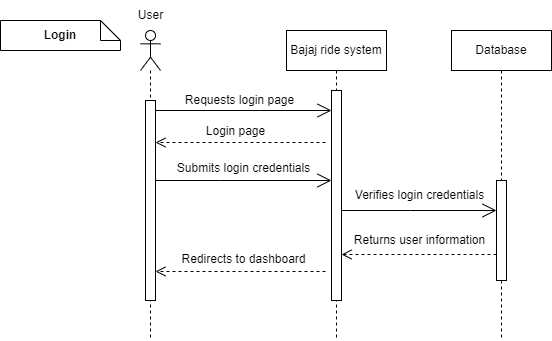


Figure 3 1: sequence diagram for log in

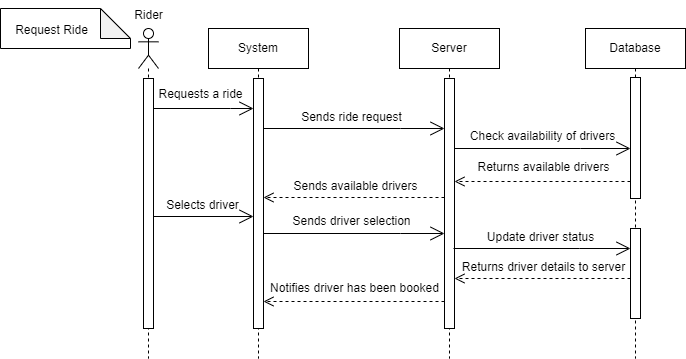


Figure 3 2 sequence diagram for ride request

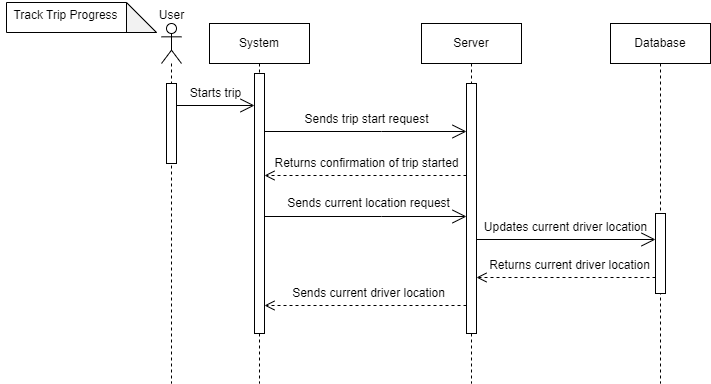


Figure 3 3 sequence diagram for tracking trip progress

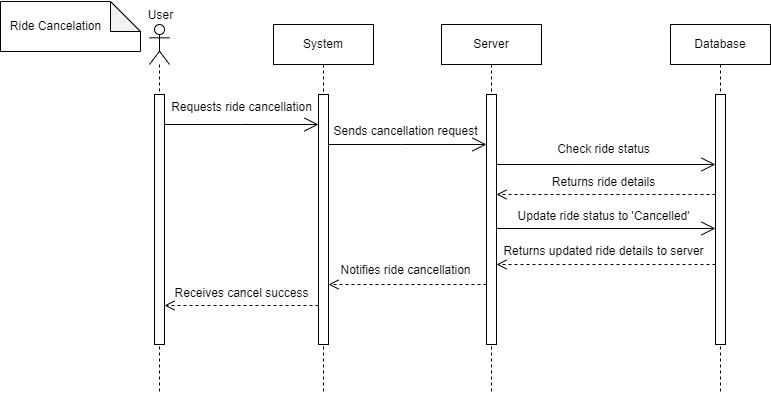


Figure 3 4 sequence diagram for canceling ride

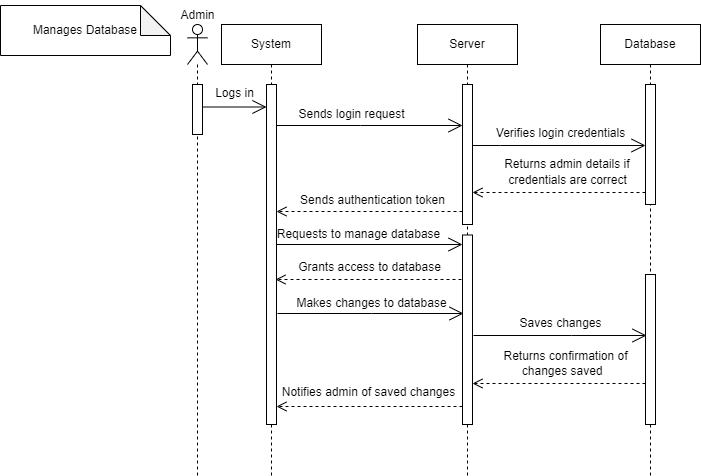


Figure 3 5 sequence diagram for managinging the database

## **User Interface**

# **Chapter Four**

1. **System Design**

## **Introduction**

## **Current software architecture (if any)**

## **Proposed software architecture**

## **System Decomposition**

## **Hardware/ software mapping**

## **Persistent data modeling**

## **Access control and security**

## **Detailed class diagram**

## **Package Diagram**

## ***Deployment***