PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
UNIVERSITY OF LAGHOUAT

FACULTY OF SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Field : Mathematics and Computer Science
Option : Computer Science
Specialization : Systems and Computer Science Networks

Dissertation Submitted in Partial fulfillment of the requirements for the Master degree in Computer Science

Submitted by : Zebda Ahmed Yacine & Remmache Mohammed Idris

THEME

Bladi App Transport Service

Examination Committee :

Mr	Tahar Allaoui	(University of Laghouat)	President
Mr	Nouredine Chaib	(University of Laghouat)	Examiner
Mr	Mohamed Lahcen Bensaad	(University of Laghouat)	Supervisor

Class of 2019/2020

Acknowledgments

First of all, All praise and gratitude is due to allah, for giving us the will and strength to complete this work. We would like to express our sincere gratitude to our supervisor Mr.Bensaad Lahcen, for his support and the amount of time and effort he put into guiding us through the duration of this project.
We would also like to thank our teachers that have given us the knowledge required to complete this work.
A special thanks to our families: both our amazing fathers and mothers, our sisters and brothers who have supported us through thick and thin, and have been an inspiration to us through our life.We could never have asked for better families.
Finally we thank our friends Rafik , Saad, Taha, Seddik, Harran, Islam and Bakar for their continuous help and support as well as all our friends in Masters 1 class.
This work is dedicated to all of you.

Contents

List of Figures

List of Tables

Introduction

The world has seen an unprecedented economical development due to the rapid evolution of technology. It has facilitated a lot of complex and otherwise time consuming tasks.Also it made it easier to earn more money in less time and with minimal effort.
The field of commerce has known some big strides over time.Starting with simple trade of basic goods between people.It has since evolved to intercontinental trade, which involves massive chains of production, delivery and large scale transactions.

Commerce helps to fuel other fields as well like:

- Hotels
- Finance
- Industry
- Marketing

Commerce is what drives the world economy forward, as people historically have been drawn to places with good trade business searching for opportunities. The merchandise involved in trade was limited due to distance, size and type of merchandise traded and the absence of proper mechanisms to overcome those limitations.
People used to carry everything in horse-drawn carriages and took sometimes weeks to deliver the goods to their respective destinations, but that's no longer the case as things continue to evolve.
the evolution of commerce to what we know today is all thanks to the revolutions made in the transportation field.This can be seen clearly with automobiles that have really shrunk distances between remote points and can be used for multiple purposes, saving a lot of time and money in the process.
the transportation field generated an estimated 5.1 trillion dollars worldwide in 2018. Travel became more affordable, the size and weight limits of the

cargo transported have increased.The time needed to transport something is dramatically lower now than before the invention of automobiles (what took weeks then takes a few hours now) and the cost of transportation is at an all time low. this helped other fields develop and expand to create new jobs and job opportunities for an ever growing population.

commercial vehicle sales continue to rise every year, in 2018 they reached 26.37 million units[1], but the demand for transportation is also increasing rapidly. Nowadays, commerce is almost completely reliant on transportation.
the development seen in the transportation field resulted in big gains and progress in the trade business.This progress has made more and different types of goods accessible to more and more markets.For example this made items like fish accessible to remote locations like desert environments.Today everything is accessible from anywhere as delivery and transport are widespread and come at a reasonable cost.
from all the above we can surmise that the field of transportation is the beating heart of the world economy, as if it stops then the economy pretty much comes to a standstill.

Problem With new innovation come new challenges that were not present before.As the demand for transportation is at an all time high and is continuing to grow, and the number of transport vehicles (trucks, SUVs, vans) is also growing fast.The problem remains in the inability to find work for the drivers and transportation for the clients on a local and national level. As the supply and demand are there and evenly matched but the connection between them is the hardest part.
for instance the client could have a driver willing to transport his cargo, but neither of them is aware of the other, so neither of them benefits.The client then has to resort to traditional methods to find transportation (Phone number or asking around)

Solution Our goal is to design a platform where drivers list their vehicles, and the clients can browse them and choose what suits their needs.The clients can also post job offers then the drivers can choose which one they do.The two sides can then contact each other through the platform to negotiate the price and other related things.

Thesis Outline Our thesis is organized as follows:

- Chapter 1: Background and related works
  in this chapter we give a general definition of transport and why it is important, we will also provide statistics to back it up as well as introducing some related works and comparing them from various aspects.

- Chapter 2: Design and Analysis
  in which we provide details about the functionalities of our platform as well as the tools used in the design and development of the project as well as diagrams to further explain how we conceived this work.

- Chapter 3: Implementation
  in this chapter we describe the tools and platforms used for implementing our system and why we chose them, we also explain the functionality of our system and we include screenshots to further elaborate and clarify the system's functionalities.

- General Conclusion And Future Work
  finally we will finish by giving a general conclusion summarizing the different steps in completing the project and the possible routes that we could take in furthering its development.

Chapter 1

Background And Related Works

In this chapter we are going to talk about the Transportation field and its importance, as well as provide some statistics to backup our point,Then we are going to see some related works:

1.1 Background

1.1.1 Transport

Transport is one of the most important systems in the world.It consists of moving people,merchandise and animals from point A to point B[2].This system has evolved through the ages from using animals in centuries past to using automobiles, trains, planes and boats[3].In this section, we will focus on land transport, specifically road transport of cargo.

1.1.2 The importance of road transport

Road transport is the dominant form of transport especially at the local and national levels.It is widely available as cars, vans, trucks and bicycles are quite common.If you need fast food delivery for example then the most suitable form of transport is by bicycle.Seeing how the size and weight of the cargo in that case is small, it would be impractical to transport it via a larger vehicle.If the distance however was higher and the cargo's size and weight were larger, then a car, van or truck would be ideal.All this tells us that there is a niche for every type of vehicle to fill when it comes to transport.because the demand is here, especially with the Corona virus pandemic and the worldwide state

of quarantine.The demand for land transport has never been higher.Land transport can keep the economy flowing locally and nationally.This is more true today than any time before, as the other modes of transport become more crippled by the Corona virus pandemic.

1.1.3 The influence of technology on road transport

Before the internet delivery and transport services were based on physical paper contracts.Large scale transport companies had contracts with one or multiple production companies, or government agencies.For example the American company Anderson Trucking Service which is in the trucking business for 65 years.This company has diversified it's services to not only transport raw charcoal but also agriculture, mining, construction and many more.After the smartphone revolution, road transport and delivery became more mainstream and easy to get.One could say that it became more convenient than traditional transport and delivery services.Uber for example heavily disrupted the taxi cab service industry.In New York city for example, the number of taxis was not enough to satisfy the high demand for transport.There were strict regulations on taxis and their numbers were only at 13000[4].Uber managed to undercut taxis by offering on demand rides for cheaper.It also managed to avoid strict regulation by being just a platform,and not a transport company.Uber also labeled it's drivers as "independent contractors"[5].It also didn't limit the number of active drivers like the taxi service industry.

Some Statistics Here are some statistics to further backup what we have seen in the last paragraph:

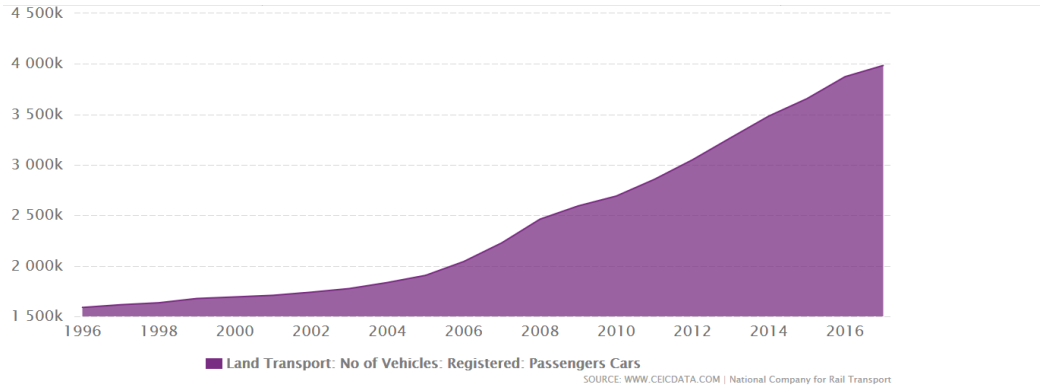

Figure 1.1: Number of passenger vehicles in Algeria over time

As we can see in figure 1.1, the number of vehicles in Algeria nearly tripled

over the course of 20 years (1996-2016) [6].

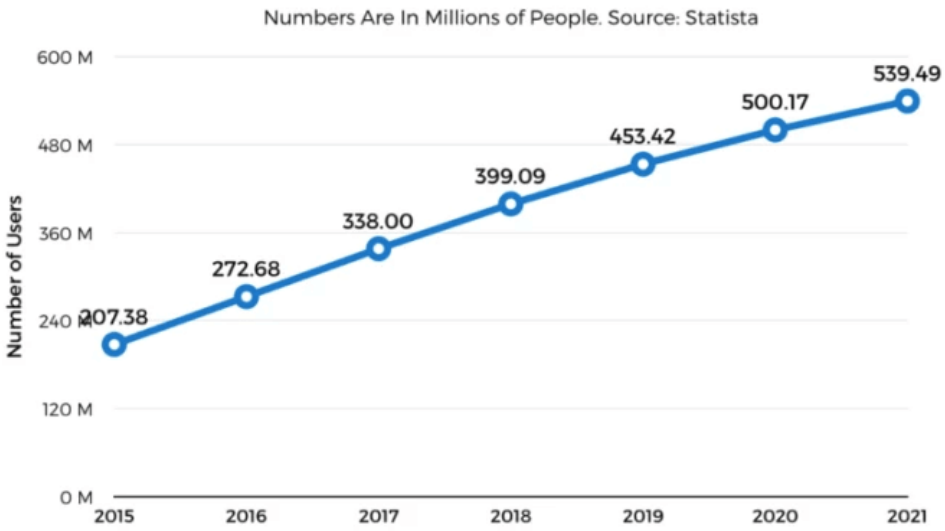

Figure 1.2: Current and projected number of ride sharing users globally

We can see the evident rise in popularity of ridesharing apps in the second half of the last decade.The number of ridesharing users in the last 5 years has more than doubled reaching more than 500 million users this year.It is expected that this number may grow to reach 540 million users by 2021 [7].

1.2 Related Works

in this section we will be taking a look at some related works, and in the end we will make a comparison between them and our project.

1.2.1 Yassir

Yassir is an android application focused on the road transport business.It offers transport and delivery services inside select areas of several countries.It functions as follows:

- Register by entering your first and last name,email and phone number.You then receive a confirmation SMS with a code.When you type that code, you are then registered successfully.

- You then have the choice to choose your departing point either manually or by using your current location.Then choose the type of vehicle you want to pick you up.
- The app then calculates the fare based on the distance and type of vehicle you chose and the current traffic.Then you select the order button.
- A notification is then sent to the closest YASSIR driver to your rendezvous point who's vehicle has the requirements you selected prior.
- After the drop off you can rate the driver based on your experience.
- YASSIR has a separate app for the drivers that lets them recieve notifications for rides.
- The drivers can also communicate through the app with the custommers and vise versa.

1.2.2 Uber

Uber is an android application focused on urban road transport.Uber is active in several countries.it offers several services to it's passengers and drivers.As it has separate apps that provide different interfaces for both sides:

- After registering, passengers can choose a meeting point and request a ride to their selected destination.They can also see the nearest drivers to their location.
- The application then gives the rider the estimated time of arrival for the driver.
- The application provides a communication interface between the driver and the rider.
- Payment is handled in-app, meaning no cash required.
- After ending the trip, the rider is prompted to rate the driver and vise versa, this two way rating system helps to give a fair assessment of both sides' experience.

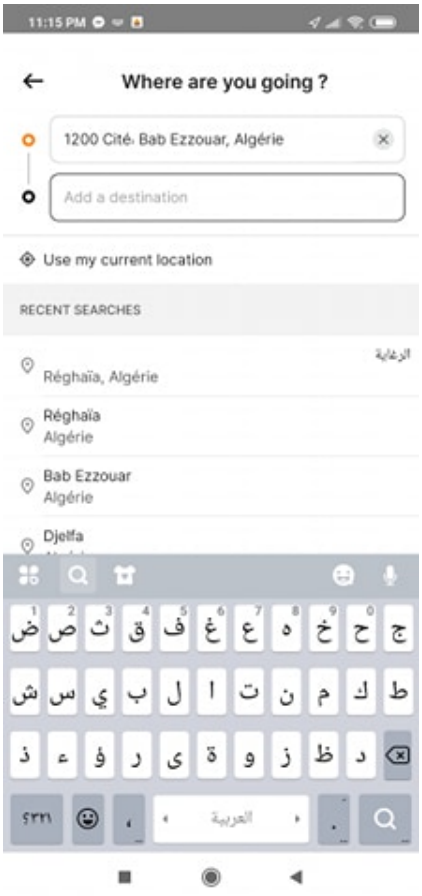

Figure 1.3: Location and destination selection interface

1.2.3 Uber eats

Uber Eats is a part of uber's app suite and is specialized for food delivery.like Uber, it has separate applications for drivers and customers.It functions like this:

- After logging in, customers are greeted with a pre-selected list of restaurants.
- Customers can choose a restaurant from that list or search for the restaurant of their choice.
- After clicking on the restaurant ,the customers are then invited to choose from the restaurant's menu.Then they place their order.
- The driver nearest to the passenger then gets a notification for the order and can choose whether to take it or not.

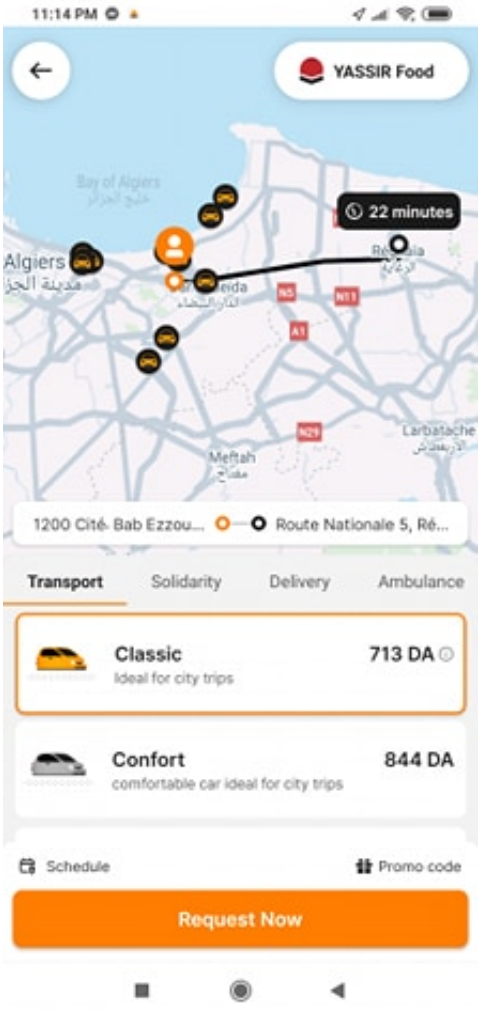

Figure 1.4: service and vehicle selection interface

- In case the driver accepts, they heads for the chosen restaurant and picks up the order.After that they head to the delivery point.
- After the delivery is complete, the client and driver are invited to give a rating to each other based on their experience.

1.2.4 Lyft

Lyft is an application dedicated to road transport of people.It is a platform that connects clients to drivers like Yassir and Uber.Lyft also does background checks on it's drivers as well as vehicle inspections.It also offers a

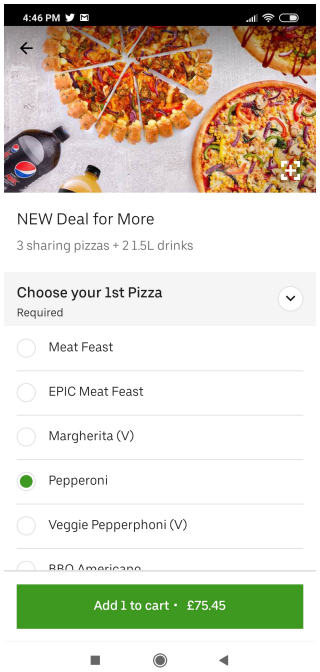

Figure 1.5: Uber eats selection menu

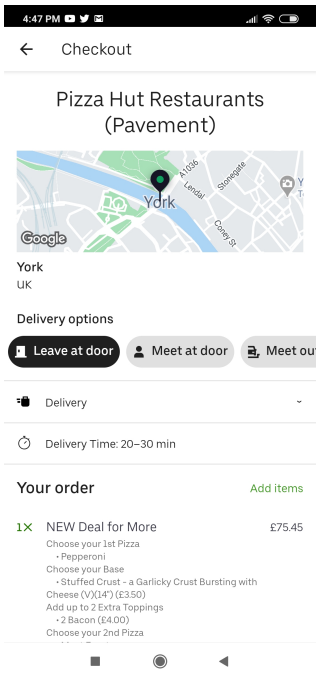

Figure 1.6: Uber eats map

rating system like the other two platforms, plus the rider can give feedback about the driver from the app.

1.3 Comparison

In this section we are comparing the related works sited previously from various aspects:

	Comparison Criteria			
Platforms	Availability	Car Options	Rating System	Driver General Opinion
Uber	Widely Available in the west (not so much in developing countries)	Low, Middle and High end options	Two Way Rating System	Average Driver Opinion and Satisfaction
Lyft	Widely Available in the west (not so much in developing countries)	Low and Mid-Range Options	Two Way Rating System	High Driver Satisfaction
Yassir	Widely Available in the west and in developing countries	Low, Middle and High end Options	Two Way Rating System	No Data Available

Table 1.1: Comparison between the 3 platforms

1.4 Conclusion

In this chapter we have talked about transportation and it's importance with an emphasis on road transportation and the influence of technology on it.we have also seen statistics that show us the development occurring in this field.Lastly we have seen some related works to our project and compared them from specific aspects.In the next chapter we are going to define and analyze the functionalities of our platform.

Chapter 2

Design And Analysis

In this chapter we look at the blueprint of our work in the form of conceptual diagrams.We used UML for our project, because it provides a set of diagrams that are the easiest to implement and translate to actual code in our case:

2.1 UML

UML or Unified Modeling Language is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.It was created by the Object Management Group (OMG) in 1997 and is now one of their standards.UML is used to model the structure as well as the behavior of systems, software or not.UML has a multitude of diagrams to represent various aspects of a systems constitution and behavior:

2.1.1 Structural Diagrams

As the name suggests they describe a system's structure,there are 7 structural diagrams in total:

- Class Diagram:
  Commonly used in OOP (Object Oriented Programming).A class consists of a name, attributes that describe it's structure and operations which give a general idea on it's behavior.The relations between classes are represented by the connecting lines between them.the combination of classes and their relations form the class diagram.

- Object Diagram:
  An object diagram is essentially an instance of a class diagram.It is used to remove the abstraction of the class diagram and test it's structure in practice.

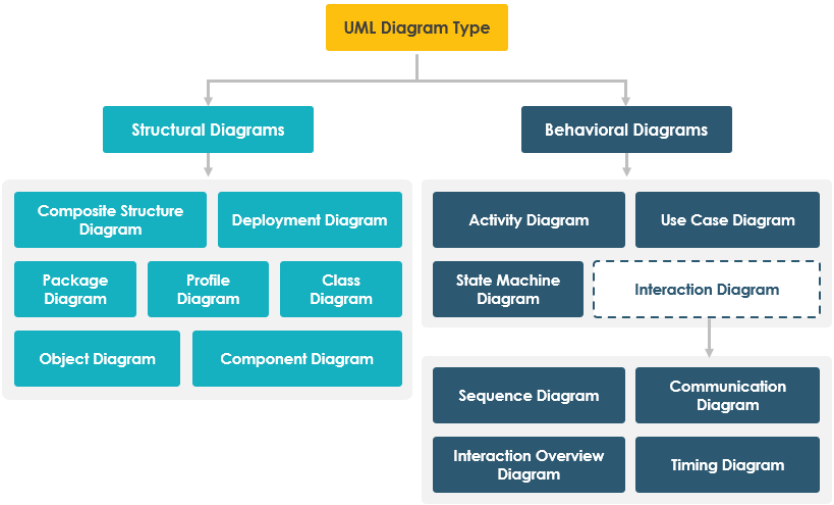

Figure 2.1: UML Diagram Types

- Component Diagram:
  Component diagram is used in the documentation of complex systems.It simplifies the documentation by breaking the system down into smaller components which helps to get a top level view of the system.It also helps to isolate system functionalities for better understanding.

- Composite Structure Diagram:
  It represents the internal structure of a class and the interactions between different class components.

- Deployment Diagram:
  This Diagram is used to visualize the deployment of software components on the hardware.

- Package Diagram:
  The package diagram is used to represent the relation between different macro-components of the system.It essentially does the same as the composite structure diagram, but at a larger and opposite scale.

- Profile Diagram:
  This type of diagram is used as an extension to UML itself so you can customize it and add new notions to it.

2.1.2 Behavioral Diagrams

these diagrams describe the behavior of the system, and like the structural diagrams, there are 7 in total:

- Use Case Diagram:
  This diagram is used to represent the general operations and functionalities that are required of a system.These functionalities are called use cases.There are 3 main components of a use case diagram:
  - Actors: represent the parties involved in interacting with the system, can be a human or a group or an organization.
  - Use Cases: encircled verbs that represent the functional requirements of the system.
  - Relationships: between use cases such as extension or inclusion.
- Activity Diagram:
  This diagram is used to describe the flow of activities and operations of the system within the scope of the objects involved.
- Interaction Overview Diagram:
  It is like a specialized activity diagram that is an activity diagram of other interaction diagrams.
- Timing Diagram:
  This diagram is used to represent the interaction between different objects through time.
- State Machine Diagram:
  Used to represent the states of an object and how it is affected by internal or external factors(events).
- Communication Diagram:
  it used to be called collaboration diagram.It focuses on the communication between the different objects of the system.
- Sequence Diagram:
  This diagram depicts the behavior of the system in detail as well as the state of the objects and actors involved.From this diagram we can see the use cases in motion[8].

We used 3 diagrams in our thesis, one Structural which is the class diagram, and 2 behavioral, which are the use case and sequence diagrams.

2.2 Class Diagram

this here is our class diagram:

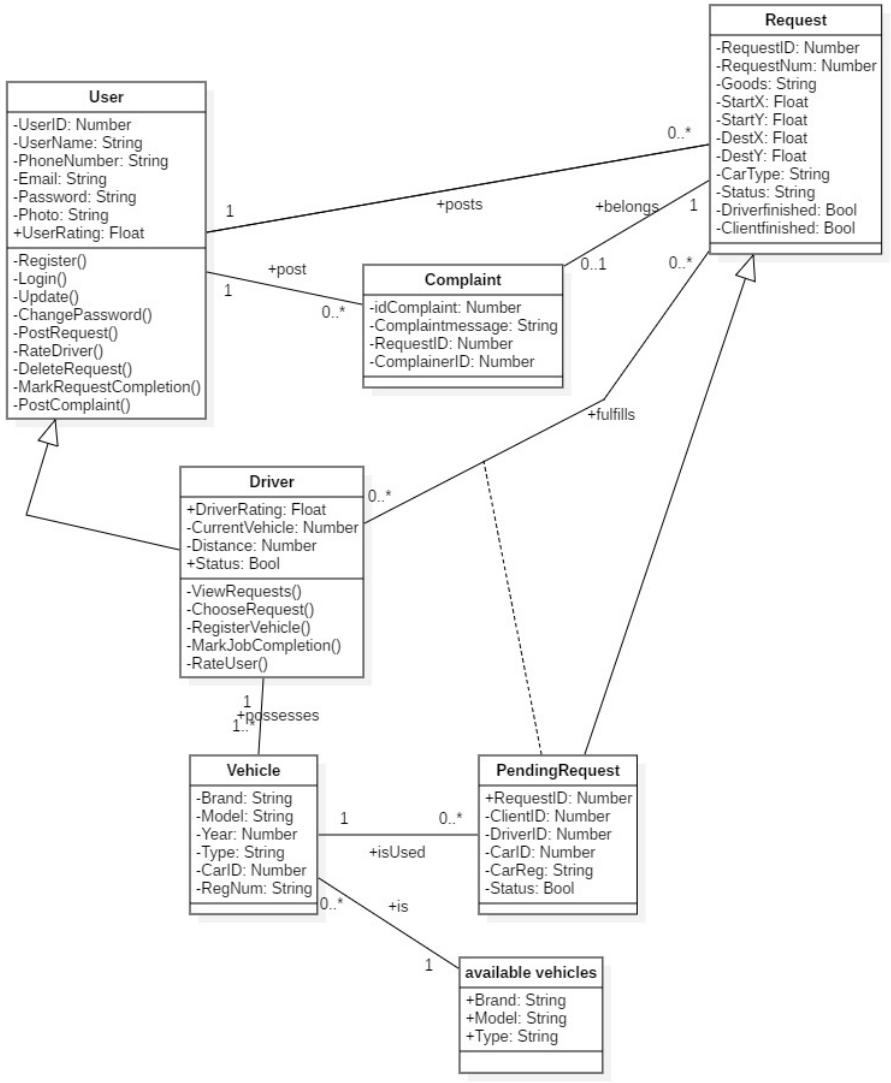

Figure 2.2: Class Diagram

2.3 Use Case Diagram

This here is our use case diagram:

Figure 2.3: Use Case Diagram

2.4 Sequence Diagrams

This section will be divided into 3 subsections:

1. User Sequence Diagrams that represent the functionalities available to all users of the application.
2. Client Sequence Diagrams that detail the functionalities available to the users of the transport service only.
3. Driver Sequence Diagrams showing in detail what the Drivers can do when using the transport service of our application
4. Shared Sequence Diagrams that represent the functionalities that are shared between the clients and drivers in the transport service.

2.4.1 User Sequence Diagrams

Register User This sequence diagram shows the process of registration of a new user:

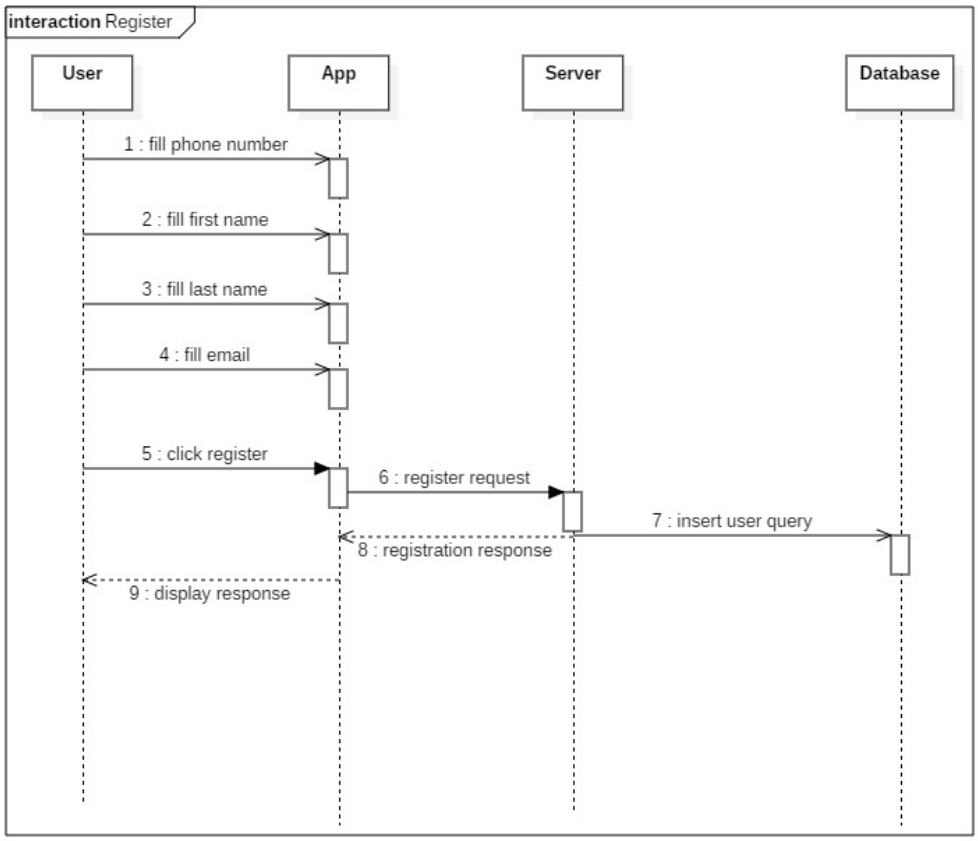

Figure 2.4: User Registration Sequence Diagram

login This sequence diagram describes the process of signing in to the platform with an existing account:

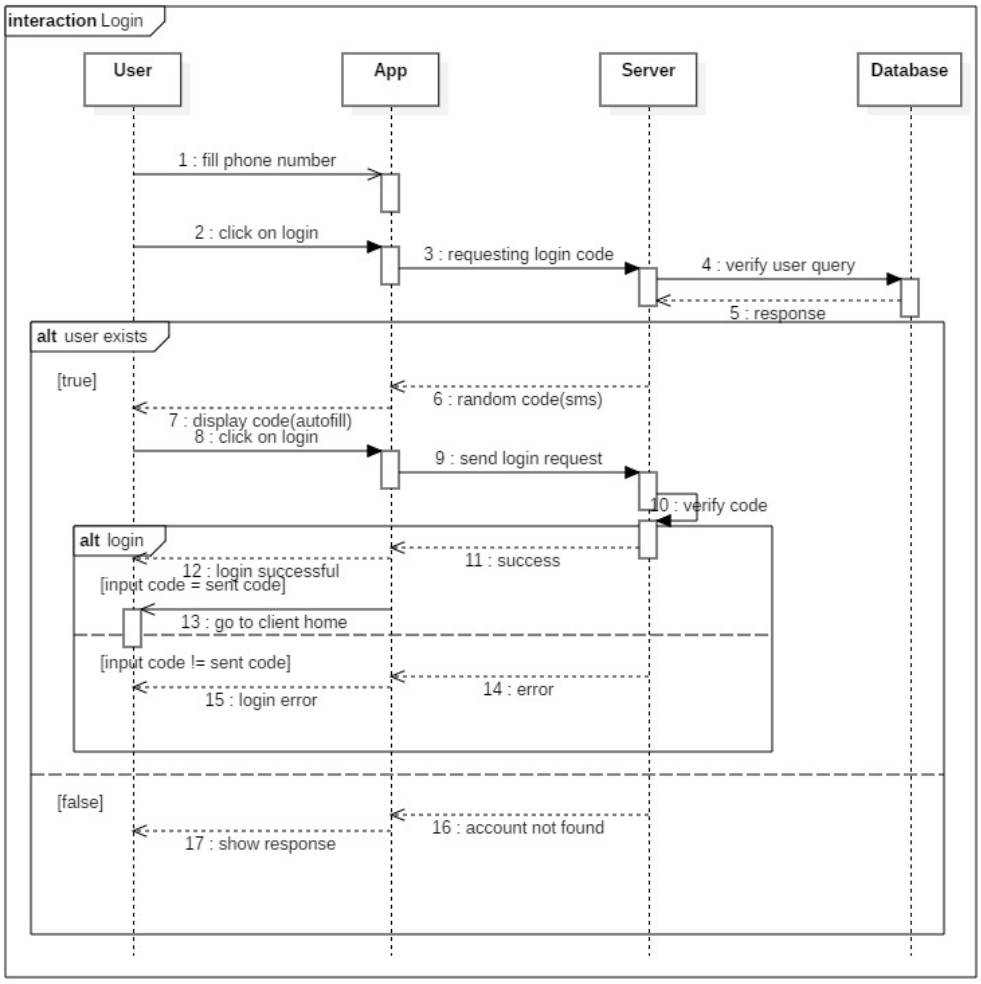

Figure 2.5: User Login Sequence Diagram

Edit profile This sequence diagram describes the process of editing user information (profile):

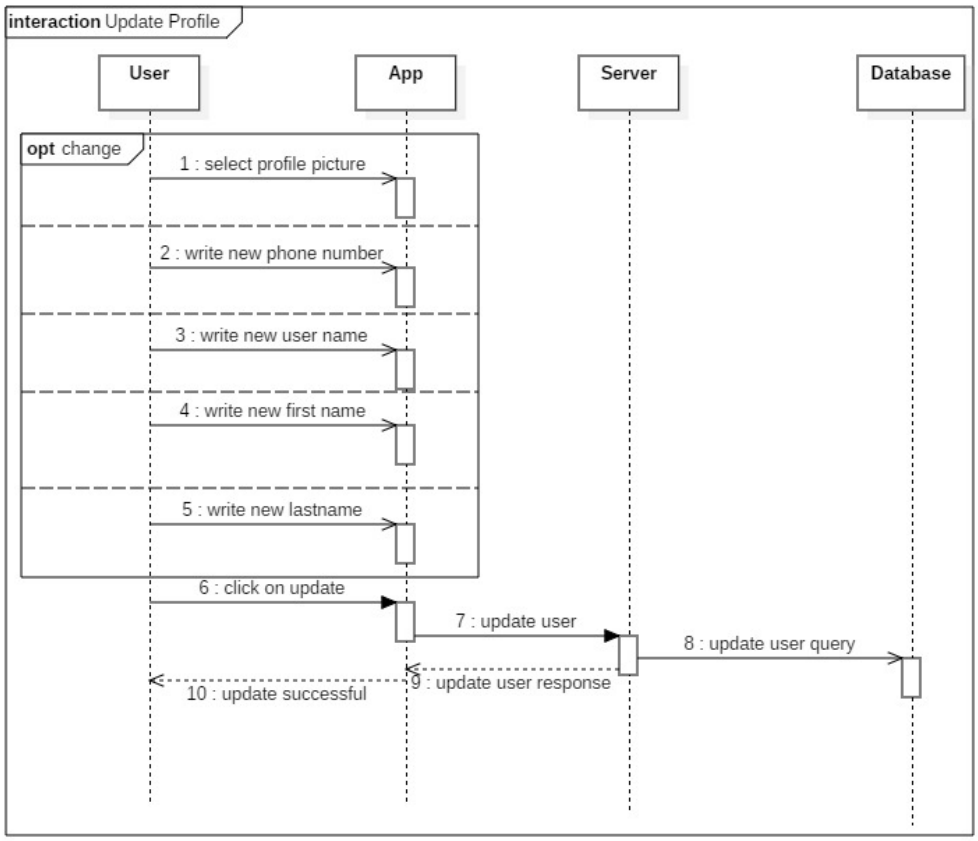

Figure 2.6: Edit Profile Sequence Diagram

2.4.2 Client Sequence Diagrams

This subsection describes the various functionalities available to the users of the transport service of our platform:

Post Request This sequence Diagram describes the process of a client submitting a transportation request:

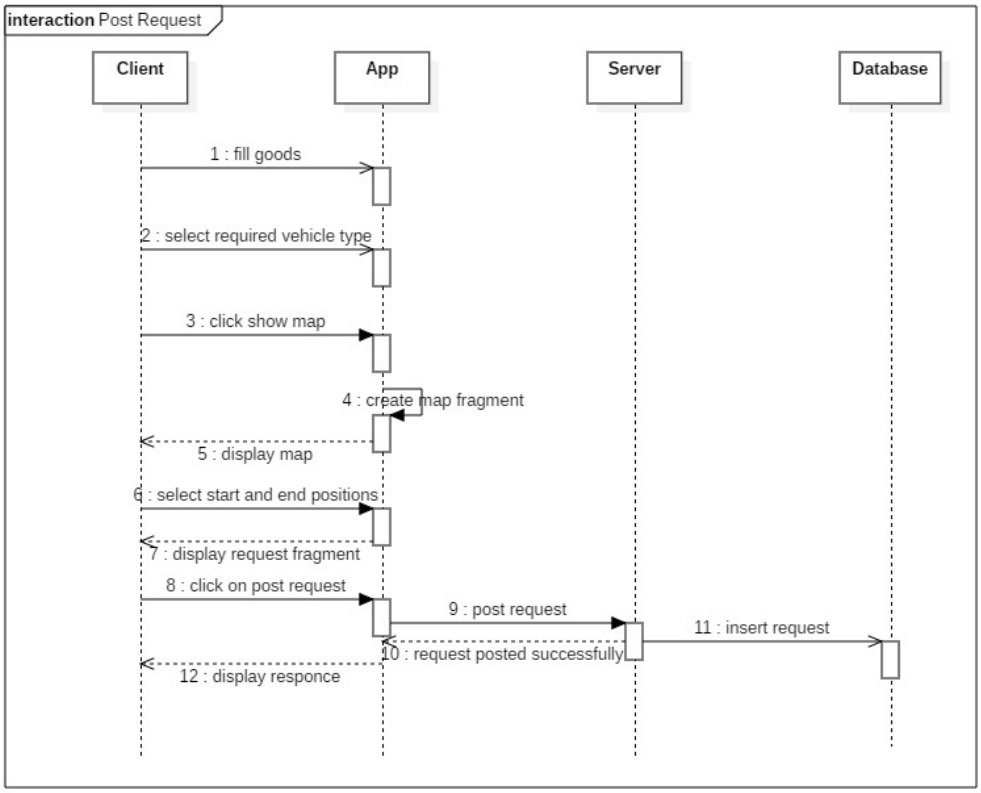

Figure 2.7: Post Request Sequence Diagram

Cancel Request This sequence diagram represents the process of canceling a request: request:

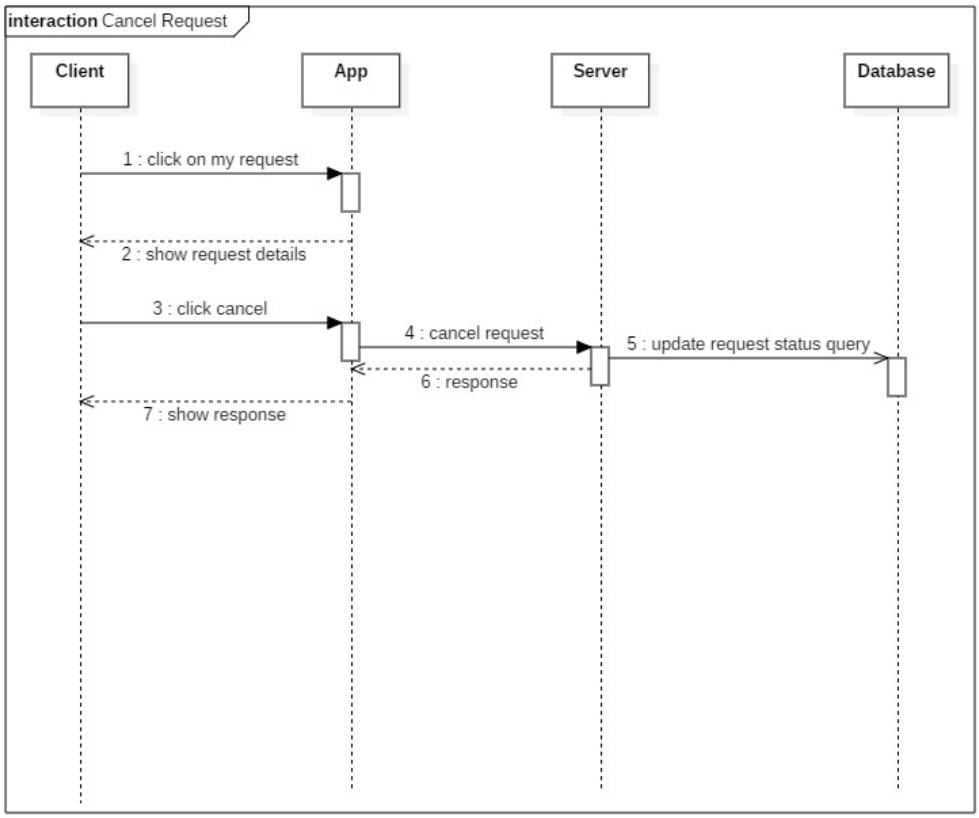

Figure 2.8: Cancel Request Sequence Diagram

Update Request this sequence diagram represents the process of a client editing the information of a certain request:

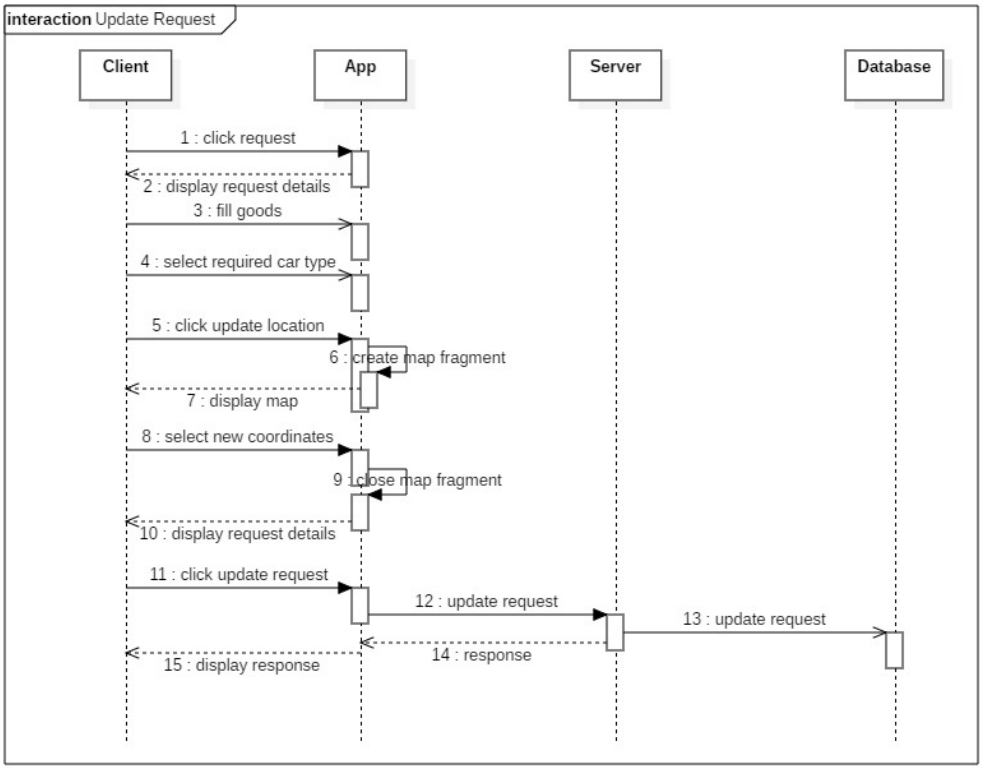

Figure 2.9: Update Request Sequence Diagram

List Bids This sequence diagram describes the process of displaying the bids of a certain request:

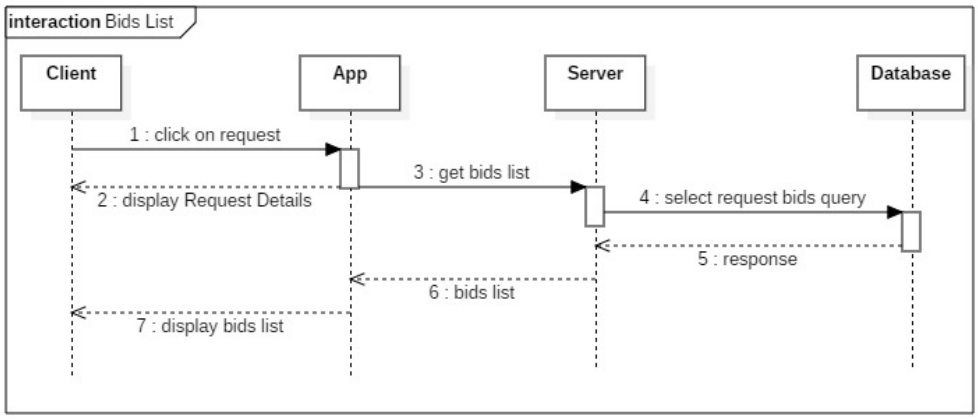

Figure 2.10: List Bids Sequence Diagram

List Bids in map This sequence diagram describes the process of viewing certain bids details in a map fragment:

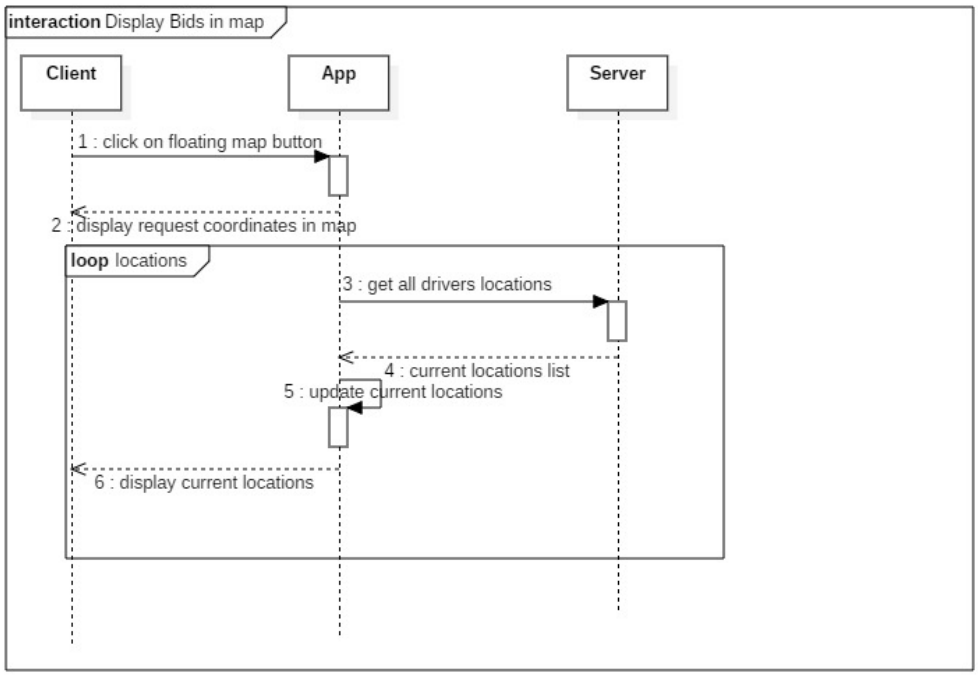

Figure 2.11: List Bids in map Sequence Diagram

Get Drivers Locations This sequence diagram describes the process of a client receiving a list of driver locations which are used either in displaying a single bid's information in map or multiple bids information:

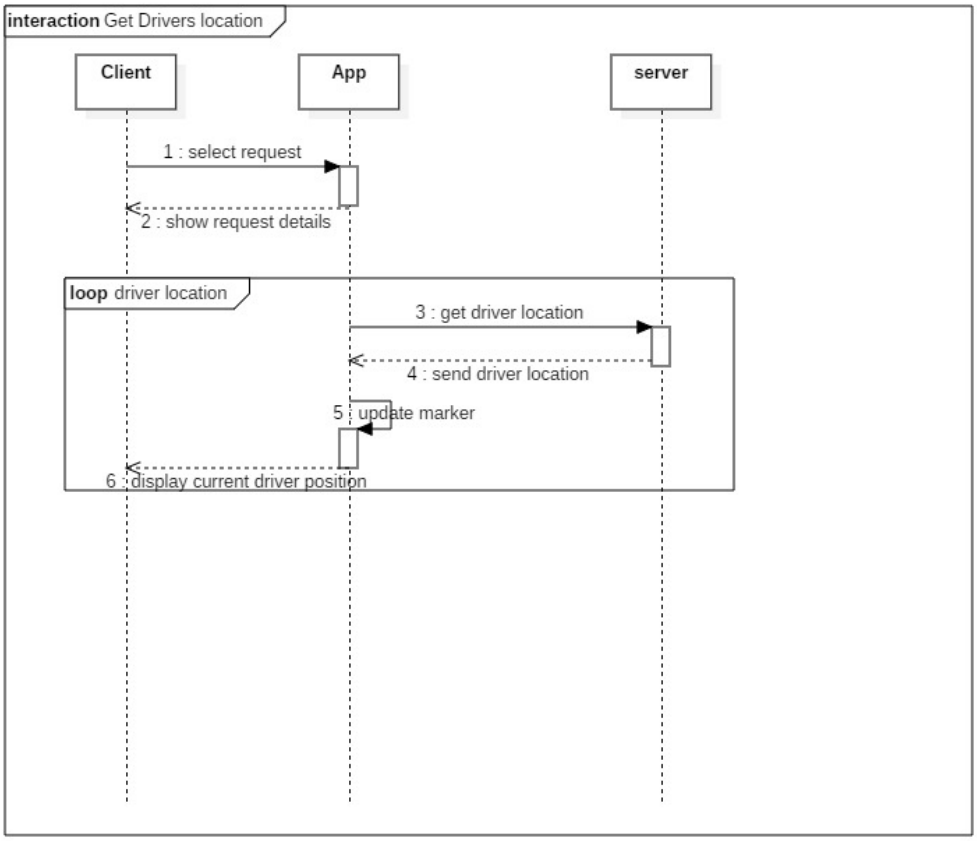

Figure 2.12: List Bids in map Sequence Diagram

Refuse Offer This diagram describes the process of a client refusing an offer (Bid) from a driver:

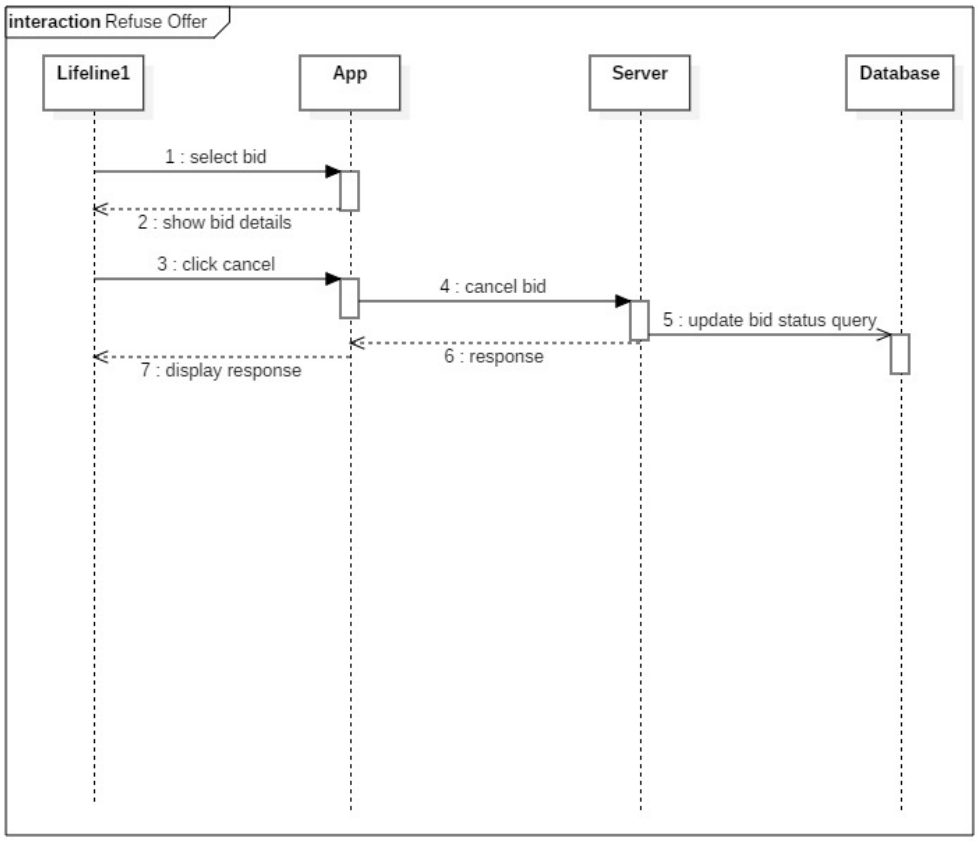

Figure 2.13: Refuse Offer Sequence Diagram

Confirm Bid This sequence diagram represents the process of a client accepting an offer from a driver:

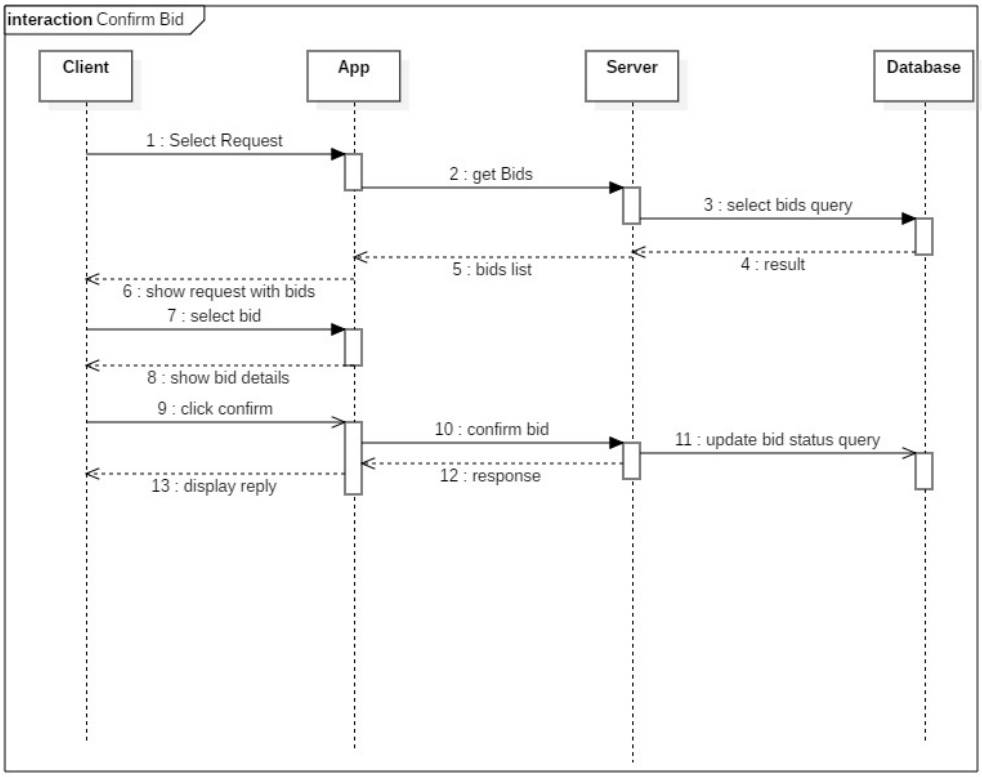

Figure 2.14: Accept Offer Sequence Diagram

2.4.3 Driver Sequence Diagrams

These sequence diagrams describe the operations available to the driver in the transport service of our platform.

Register Driver This diagram describes the process of a user applying for registration as a driver in the transport service:

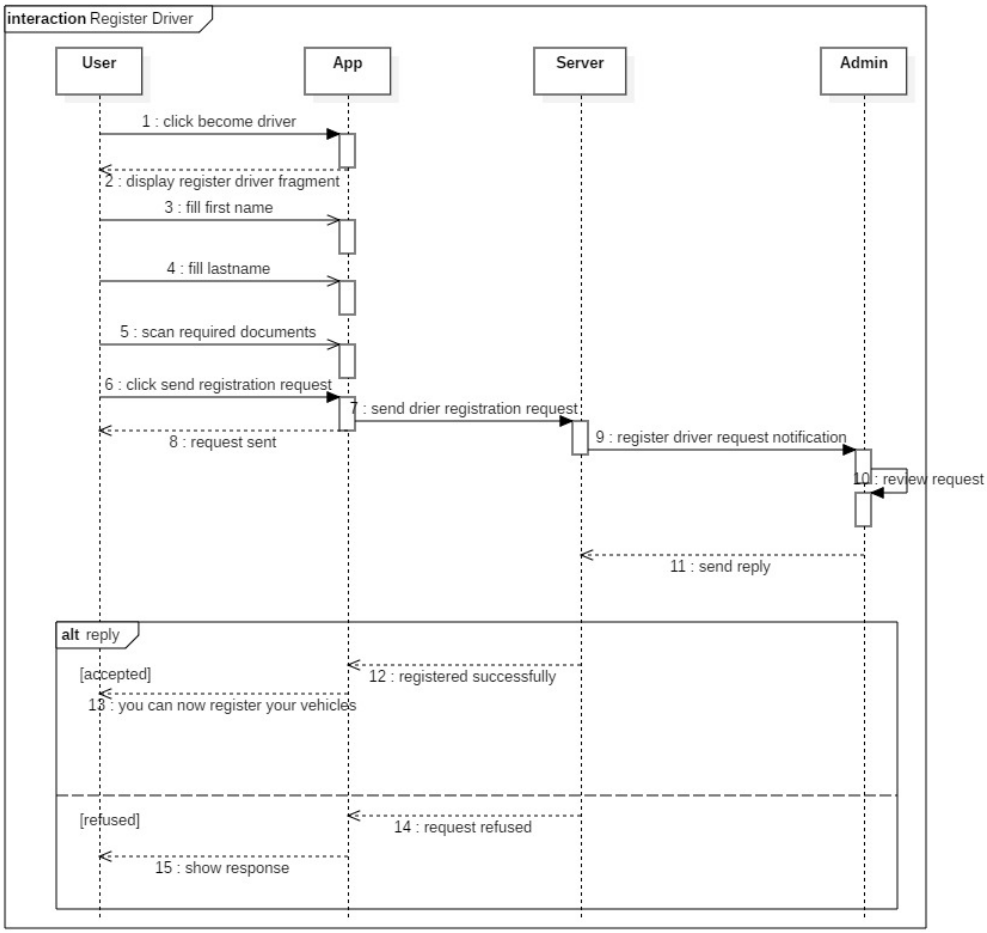

Figure 2.15: Driver Registration Sequence Diagram

Add Vehicle This sequence diagram represents the process of a driver registering a new vehicle with which to run errands for the clients:

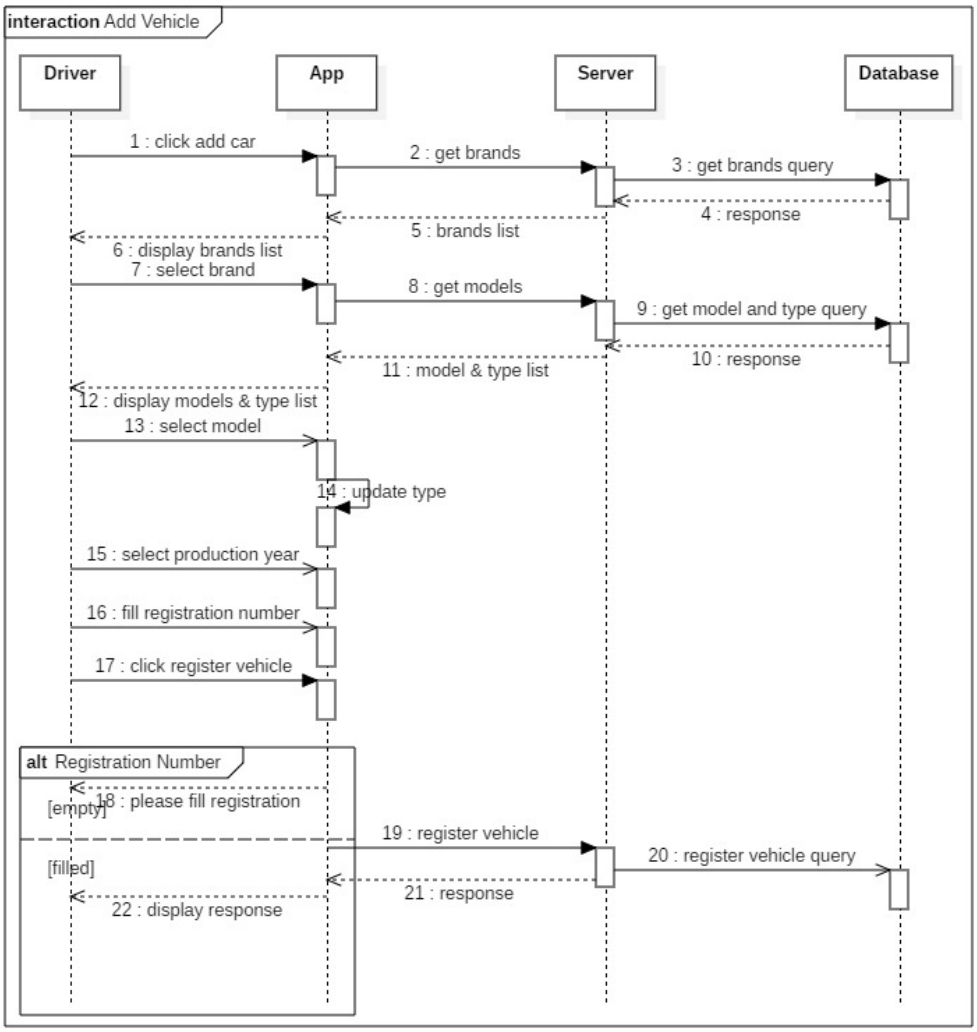

Figure 2.16: Vehicle Registration Sequence Diagram

Edit Vehicle This sequence diagram represents the process of editing a vehicle's information:

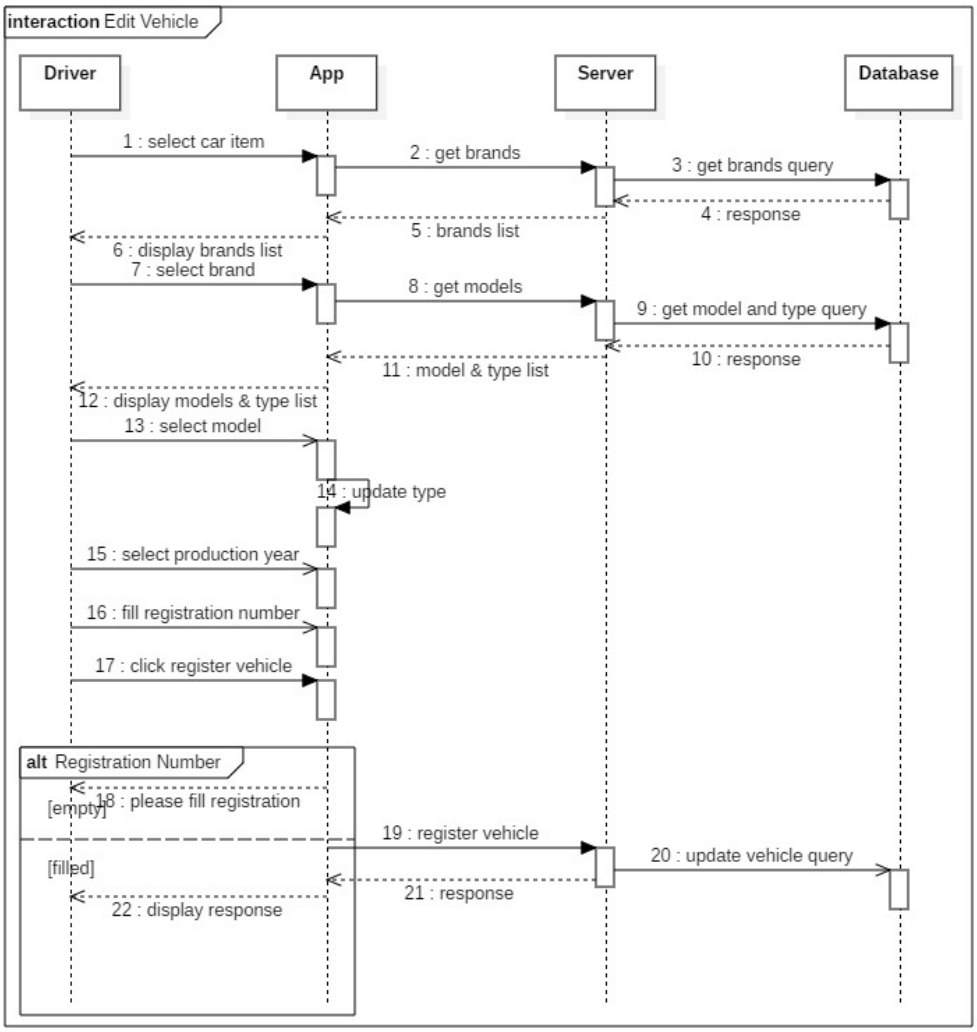

Figure 2.17: Edit Vehicle Sequence Diagram

See Requests This diagram shows the procees of a driver viewing available requests to bid on:

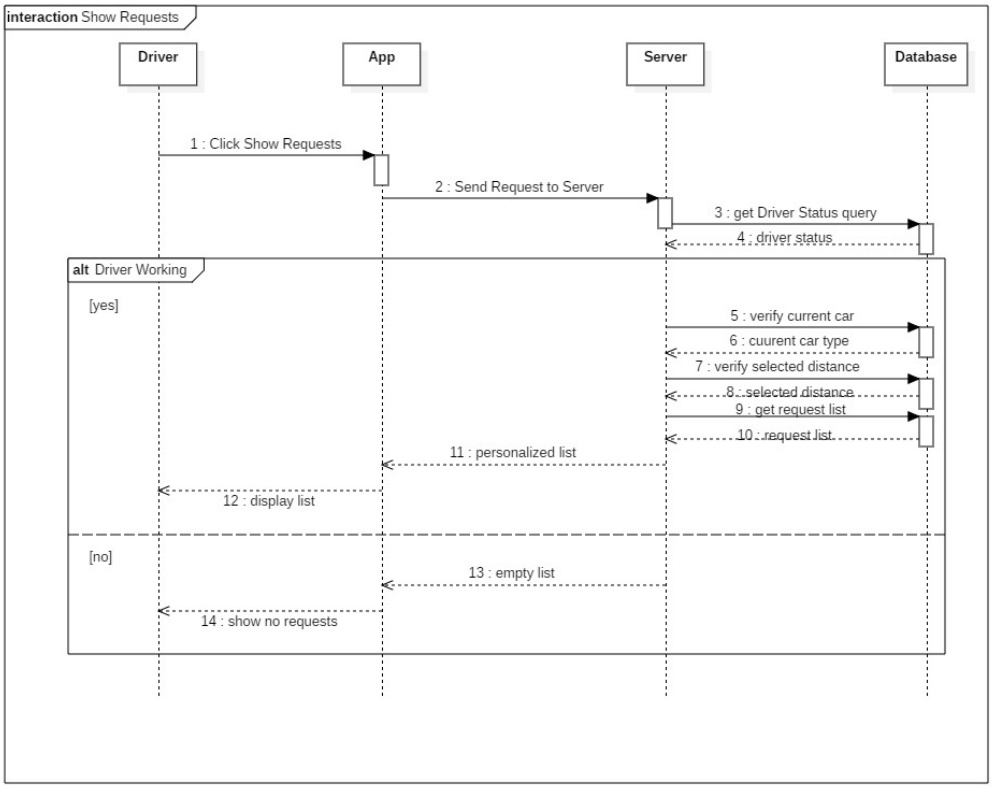

Figure 2.18: See Requests Sequence Diagram

Bid This sequence diagram represents the process of a driver giving an offer (Bid) to the client on a specific request posted by the client:

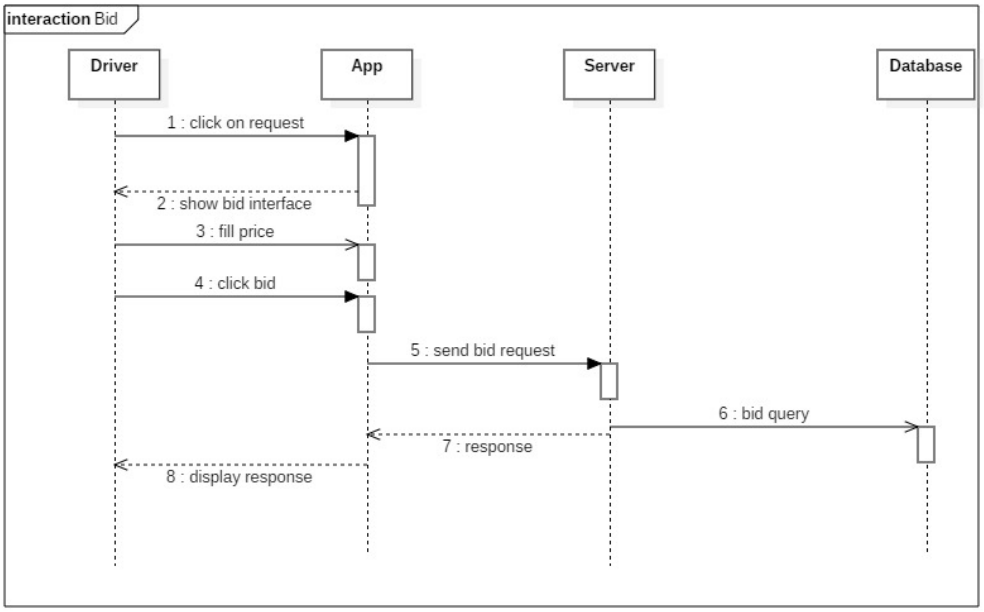

Figure 2.19: Bid Sequence Diagram

Show Bids This sequence diagram describes the process of listing the not yet accepted offers the driver has made on several requests to clients:

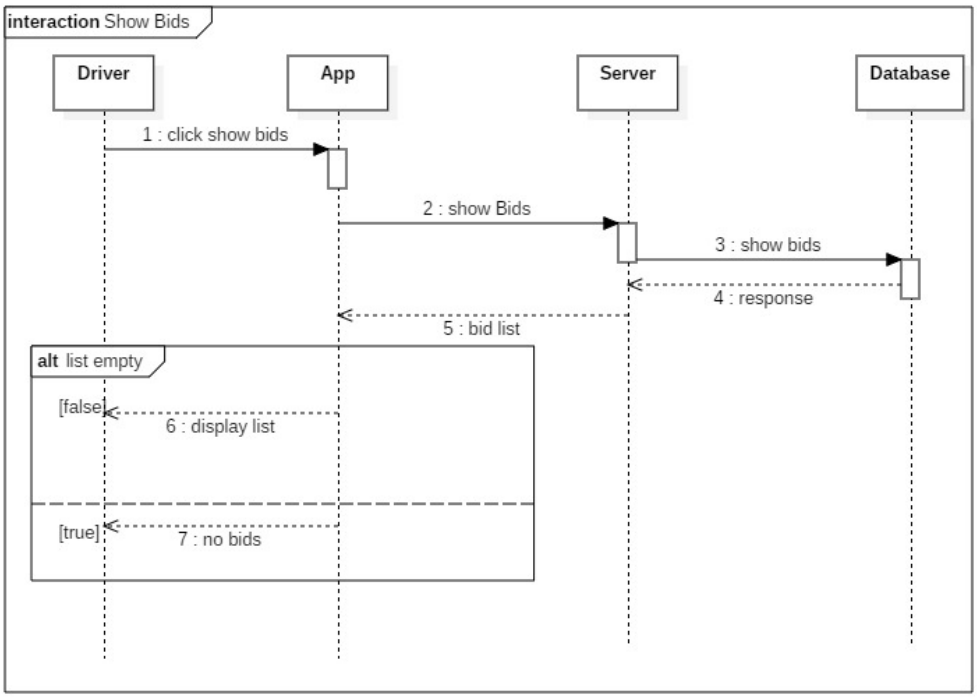

Figure 2.20: Show Bids Sequence Diagram

Cancel Bid This diagram concerns the process of canceling an offer previously made by the driver on a certain request to the client:

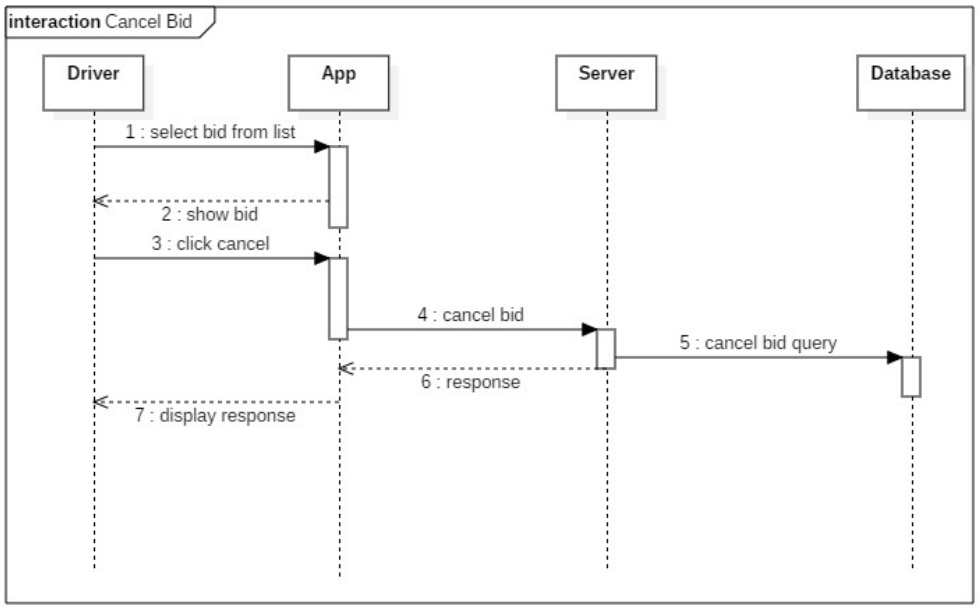

Figure 2.21: Cancel Bid Sequence Diagram

Send Location This diagram describes the process of driver tracking which occurs automatically when the driver is working:

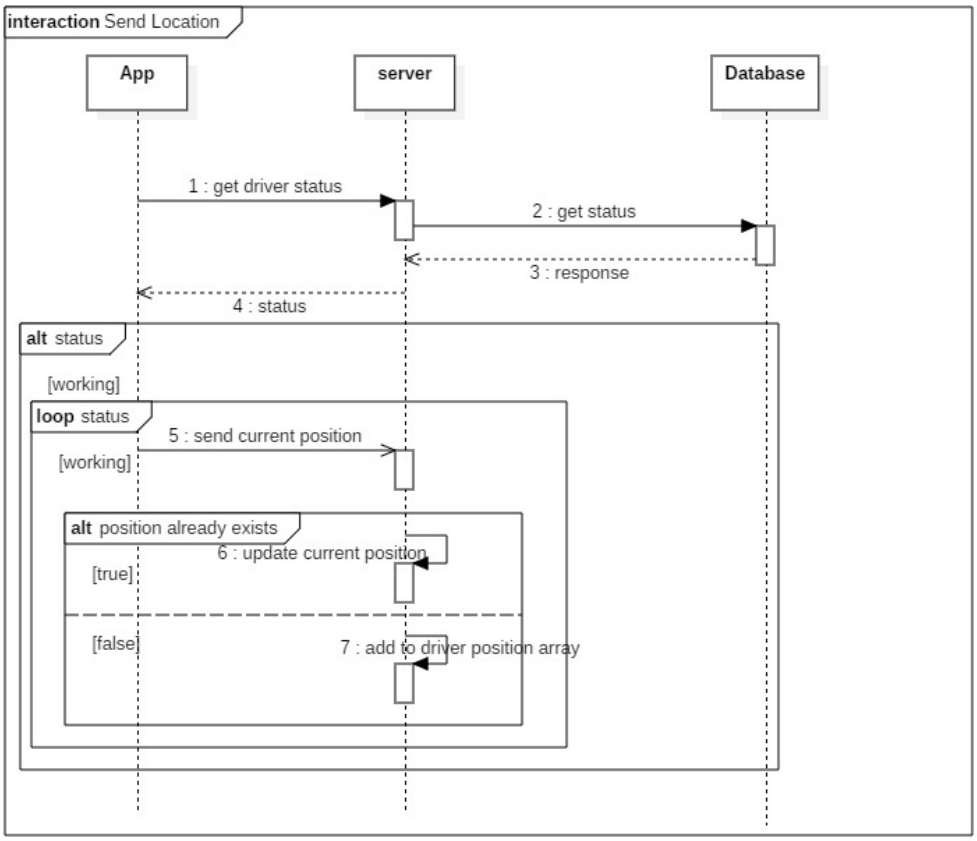

Figure 2.22: Send Location Sequence Diagram

Update Config This diagram represents the process of choosing the suitable settings to better personalize the experience for the driver:

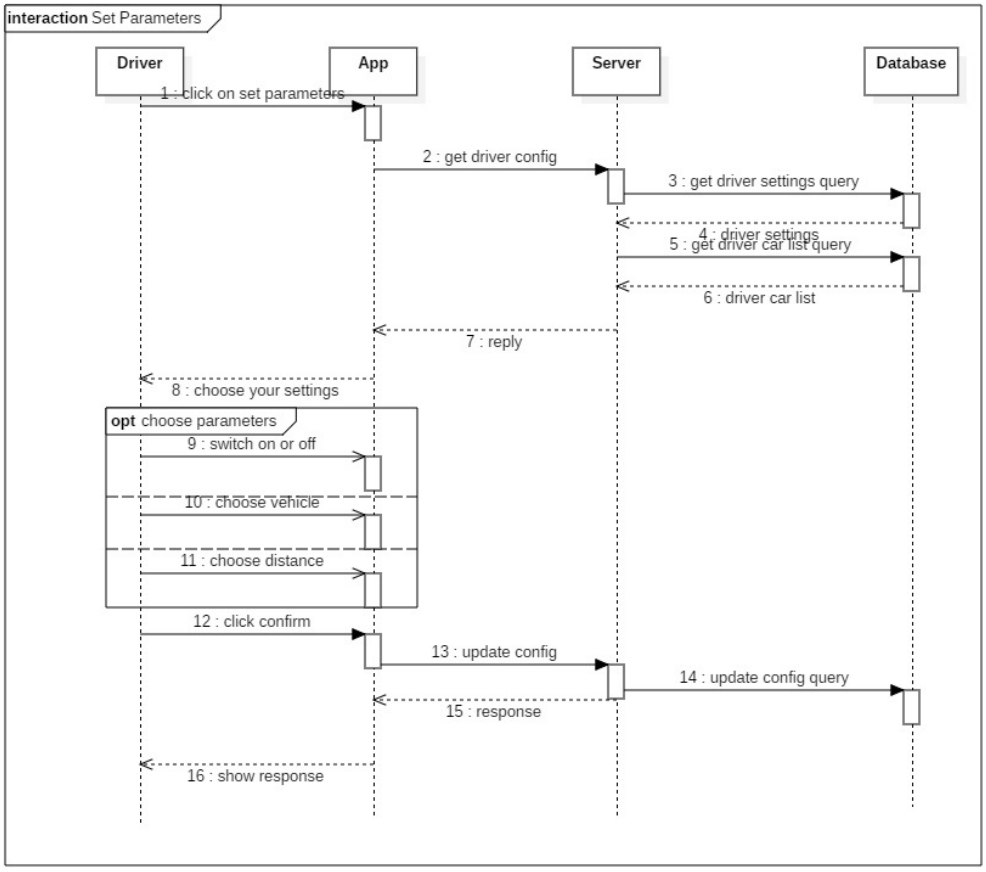

Figure 2.23: Set Driver Parameters Sequence Diagram

2.4.4 Shared sequence diagrams

These sequence diagrams represent the shared functionalities that are available to both types of users of the transport service of our platform (client and driver):

See Jobs This sequence diagram describes the process of viewing pending jobs or errands:

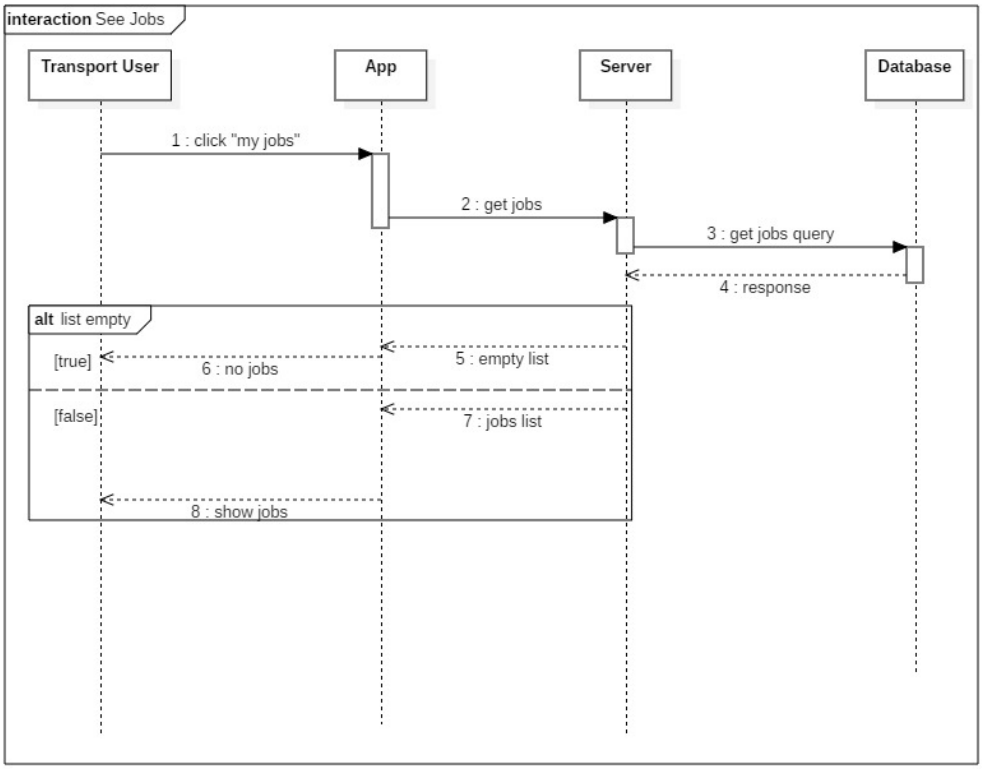

Figure 2.24: See Jobs Sequence Diagram

Commit Job This diagram shows us the process of confirming the completion of an errand submitted by the client:

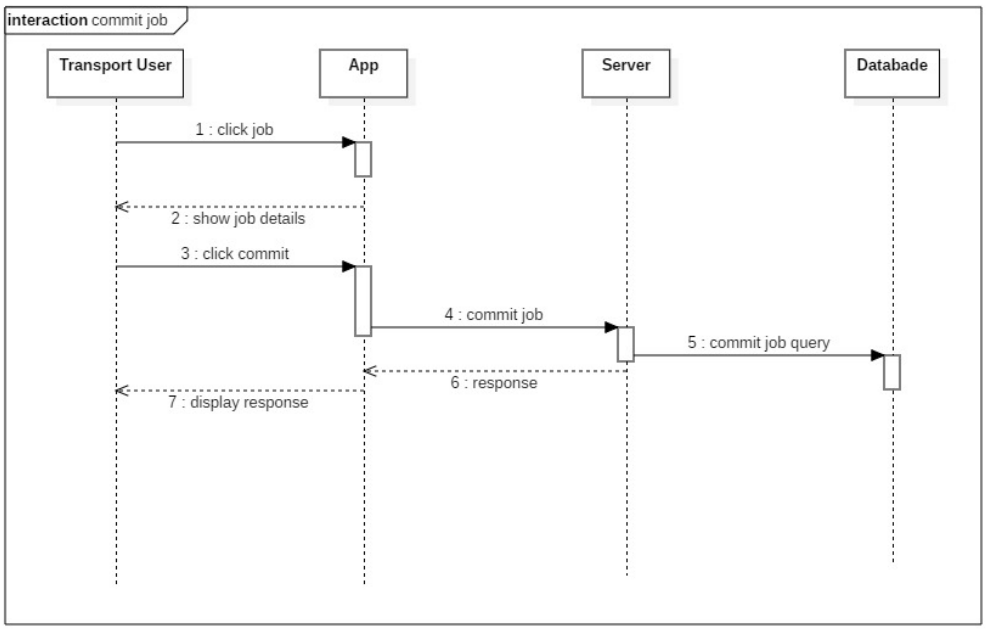

Figure 2.25: Commit Job Sequence Diagram

Rate This qequence diagram describes the process of rating the other participant in the request.Either the client rates the driver or vice versa.

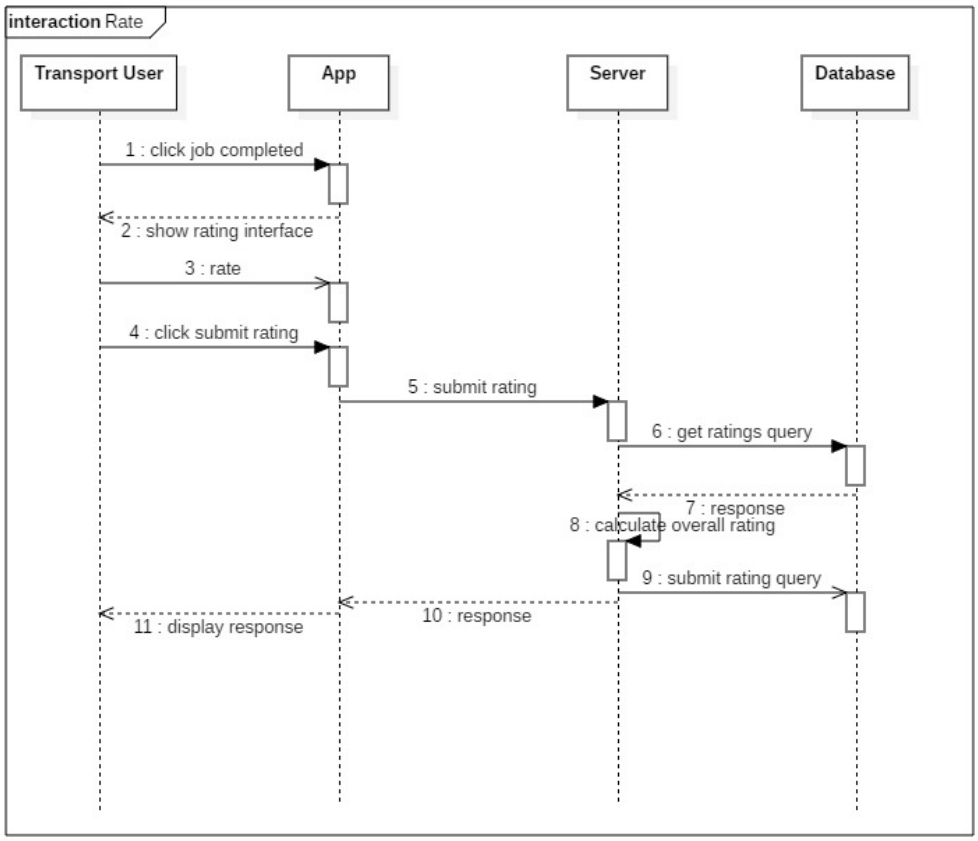

Figure 2.26: Rate Sequence Diagram

File Complaint This diagram describes the process of a Transport user filing a complaint against the other participant in the request in case of a miss conduct on the part of the latter:

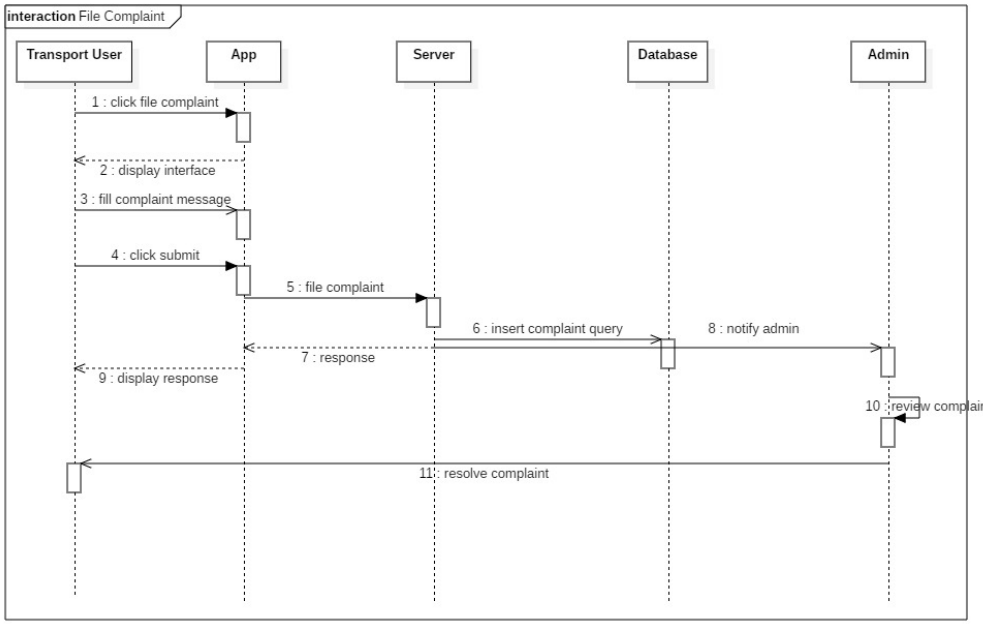

Figure 2.27: File Complaint Sequence Diagram

2.5 Conclusion

In this chapter, we have seen the abstraction of the functionalities of our platform in the form of UML diagrams, that will form the basis of the next phase: implementation, which we are going to see in the next chapter.

Chapter 3

Implementation

In this chapter we will talk about everything related to the implementation of our work:

1. The Hardware used for the development and testing of the project
2. The software used for the development of this project.
3. The Architecture we chose to implement the project
4. Other android architectural patterns
5. The reason we chose MVVM architecture
6. System Functionality

3.1 Hardware Configuration

3.1.1 Development

- Asus Vivobook X541UJ
  - CPU: Intel Core i7-7500U @2.7-3.5Ghz (2 cores, 4 threads, 4MB L3 cache)
  - RAM: 8GB DDR4 @2133 Mhz
  - Disk: 1TB HDD 5400 RPM
- Toshiba Satellite
  - Intel Core i7-6500U @2.6-3.1Ghz (2 cores, 4 threads, 4MB L3 cache)

  - RAM: 8GB DDR4 @2133 Mhz
  - Disk: 1TB HDD 5400 RPM

3.1.2 Testing

- Xiaomi Redmi Note 8 pro
  - Chipset: Mediatek Helio G90T @2.05 Ghz(8 cores)
  - RAM: 6GB LPDDR4X RAM
  - Storage: 128 GB UFS 2.1
  - OS: Android 10 API level 29
- LG Xpower 3
  - Chipset: Qualcomm Snapdragon 425 @1.4 Ghz (4 cores)
  - RAM: 2GB LPDDR4 RAM
  - Storage: 16GB eMMC 5.1
  - OS: Android 8.1 Oreo API level 27

3.2 Software

3.2.1 Android Studio

Android Studio is an IDE (Integrated Development Environment) made by google for the development of android applications.It is based on Jetbrain's Intellij IDE.It is available to download on the three major desktop operating systems: Windows, Linux and MacOS[9].

3.2.2 Kotlin

Kotlin is a cross-platform statically typed, general purpose programming language.It is deigned to interoperate fully with Java and its JVM depends on the java class libraries.On 7 May 2019, Google announced that the Kotlin programming language is now its preferred language for Android app developers[10].

3.2.3 SQL

SQL is a domain specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables[11].

3.2.4 MySQL

MySQL is an open-source relational database management system (RDBMS).It is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses.[12]

3.2.5 gRPC

gRPC (gRPC Remote Procedure Calls) is an open source remote procedure call (RPC) system initially developed at Google in 2015. It uses HTTP/2 for transport, Protocol Buffers as the interface description language, and provides features such as authentication, bidirectional streaming and flow control, blocking or nonblocking bindings, and cancellation and timeouts. It generates cross-platform client and server bindings for many languages. Most common usage scenarios include connecting services in microservices style architecture and connect mobile devices, browser clients to backend services.[13]

3.2.6 Protocol Buffers

Protocol Buffers (Protobuf) is a method of serializing structured data. It is useful in developing programs to communicate with each other over a wire or for storing data. The method involves an interface description language that describes the structure of some data and a program that generates source code from that description for generating or parsing a stream of bytes that represents the structured data.[14]

3.3 Architecture

3.3.1 MVVM Architecture

MVVM or Model View ViewModel is an architectural pattern for implementing user interfaces and it consists of 3 parts:

1. Model:This holds the data of the application. It cannot directly talk to the View. Generally, it's recommended to expose the data to the ViewModel through Observables.

2. View: It represents the UI of the application devoid of any Application Logic. It observes the ViewModel.

3. ViewModel: It acts as a link between the Model and the View. It's responsible for transforming the data from the Model. It provides data streams to the View. It also uses hooks or callbacks to update the View. It'll ask for the data from the Model[15].

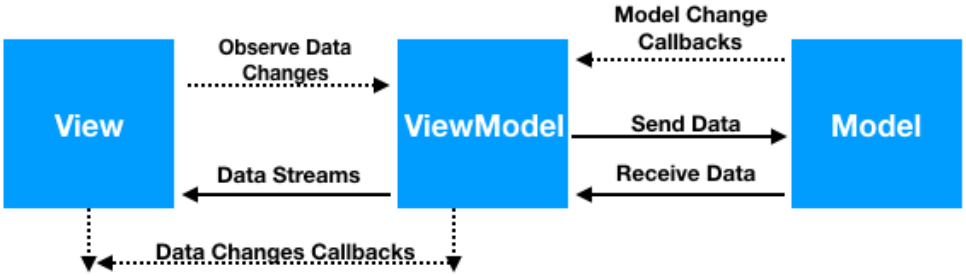

Figure 3.1: MVVM architectural pattern simplified

3.4 Other Android architectural patterns

There are three main architectural patterns in android: MVC, MVP, MVVM.

3.4.1 MVC(Model View Controller)

In this architectural pattern, the view has direct access to the model(data) through a controller.The model is completely exposed to the view which may cause some security concerns.

3.4.2 MVP(Model View Presenter)

In this architecture the view accesses the model (data) through a presenter class.The view demands the data from the presenter which in turn demands

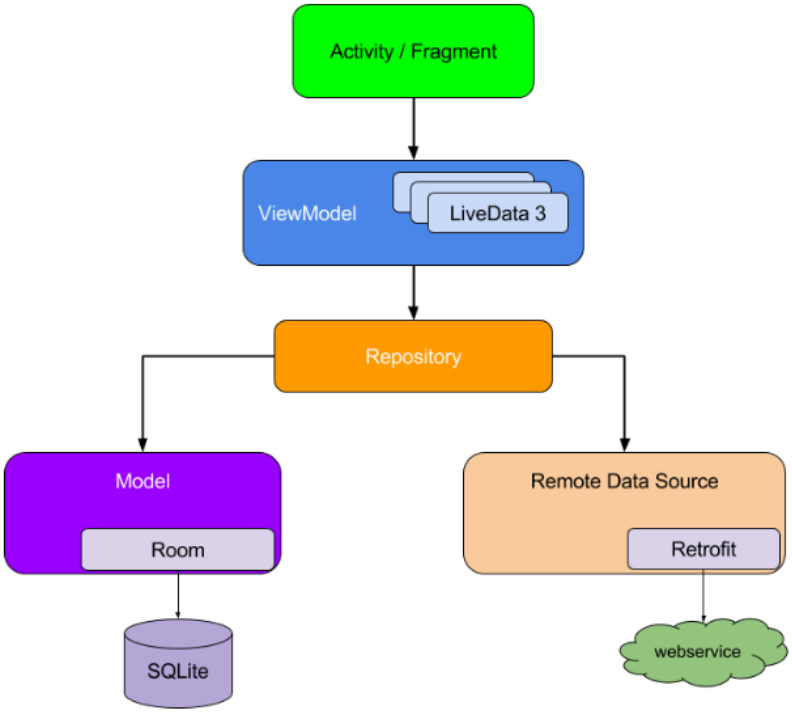

Figure 3.2: MVVM architectural pattern

it from the model.The model then sends the data to the presenter that sends it to the view.The presenter in this case is aware of the view[16].

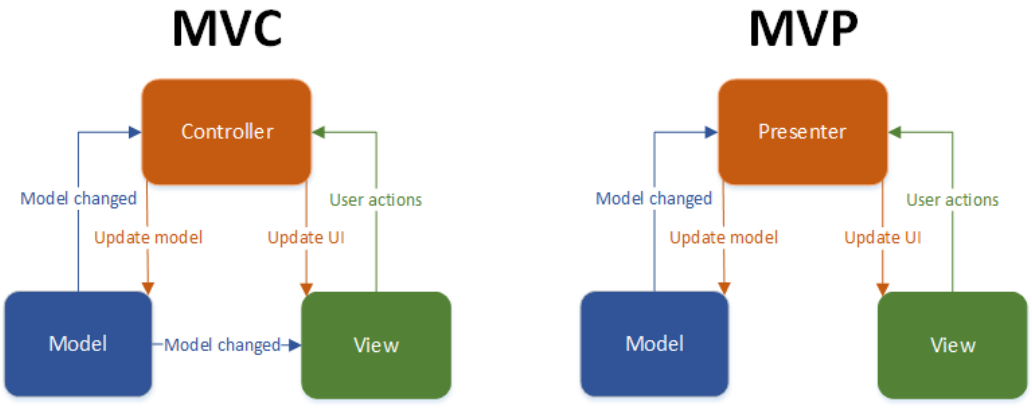

Figure 3.3: MVC and MVP architectural patterns simplified

3.5 The reason we chose MVVM

MVVM provides an advantage over the other architectures.As the complexity of the code grows, its maintenance also becomes harder and more time consuming.This is remedied by The decoupling of the architectural components of MVVM (Building in a distributed manner).As our code became more and more complex, we avoided wasting time in maintenance and error correction thanks to MVVM.It is also the main architecture that's supported and promoted by google for application developments.Add to that, this architecture is the most suitable for multi-service applications like ours.It allows for greater flexibility in adding new functionalities to the application.MVVM is also recommended to use for development teams.

3.6 System Functionality

In this section we present the final product of our application in the form of screenshots of the interfaces of our application:

3.6.1 User

The functionalities shown here are available to all the users of our application:

Register This screen is for user registration, where you enter your phone number, first name and last name, then you click the register button and you are automatically logged in and taken to the Home screen.

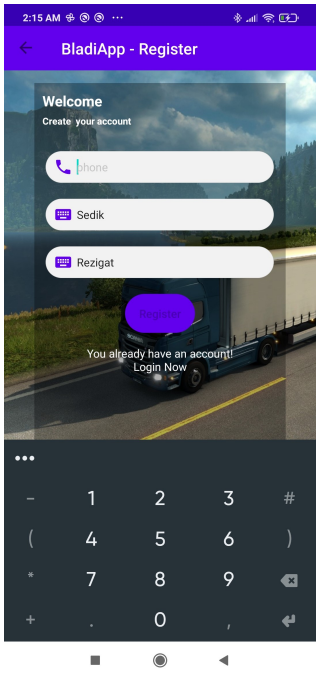

Figure 3.4: Register Screen

Login This is the first screen that greets the user after launching the app,if they are logged out or don't have an account:

- If the user already has an account, they only have to enter their phone number and press Log in, a random code is then generated from the server and sent to the user as an sms message.after the code is filled in the user presses login again and is taken after that to the Home screen.
- If the user doesn't have an account, they just have to click on register a new account to be taken to the Registration fragment.

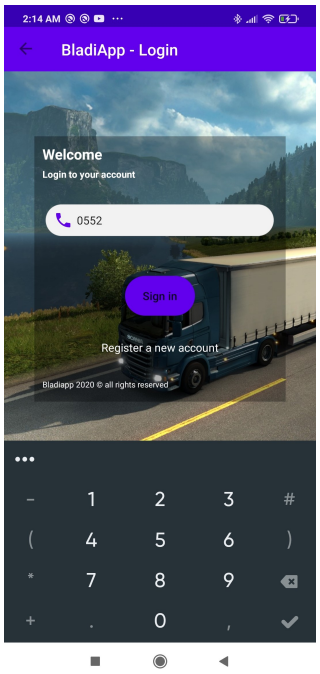

Figure 3.5: Sign in Screen

Home This is the home screen of our application in which the user can access different services, if the user is logged in then this is the first screen that's displayed to them.

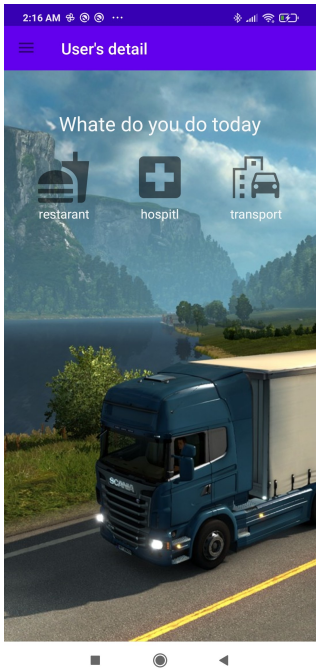

Figure 3.6: Home Screen

Sidebar The Sidebar hosts shortcuts to convenient functionalities for easy access like:

- Home: for going back to the Home fragment to switch services.
- Profile: to display the user's profile and personal information.
- About: Display information about the application.
- Driver Mode: to have access to the driver functionalities in the transport service.
- Logout: to sign out of the application.
- Exit: to close the application.

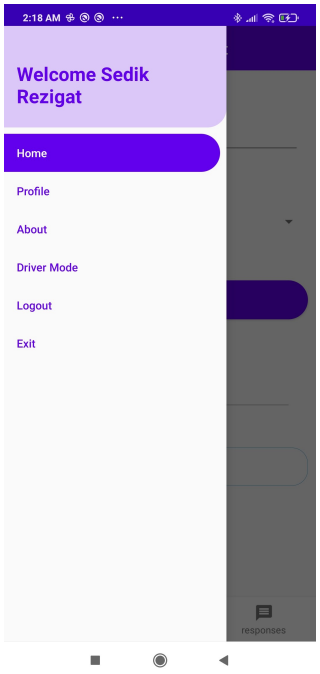

Figure 3.7: Side Bar

Profile The user Accesses this Fragment through the sidebar.The user can see their personal info and edit it if they wish.

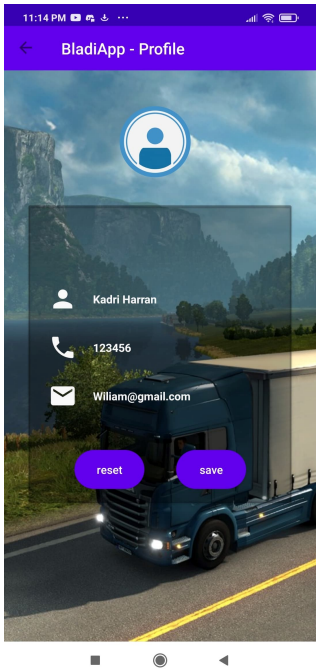

Figure 3.8: User Profile

3.6.2 Client

The functionalities shown here are available to the users of the transport service.

Post Request The first screen that's displayed when entering the transport service.The client can fill information about the cargo as well as the type of vehicle required for its transportation, then he pinpoints the departure and arrival points on the map and then submits the request.

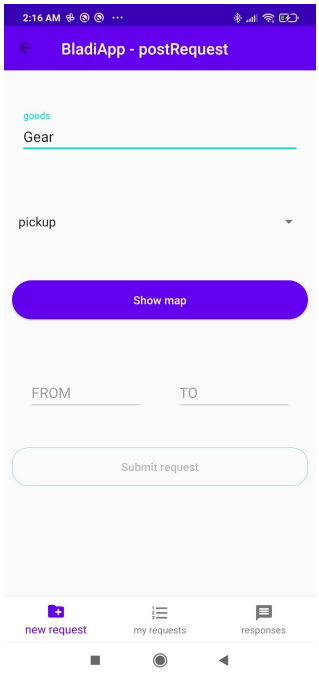

Figure 3.9: Post Request Screen

Figure 3.10: Select Coordinates screen

My Requests in This screen the client can see all their requests which they have yet to chose an offer on.

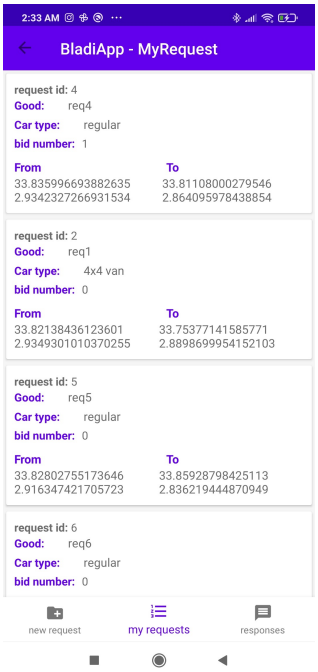

Figure 3.11: My Requests Screen

Edit Request This screen allows the client to edit one of their requests or cancel it.The client can access it by clicking on the request in the request and bids fragment.

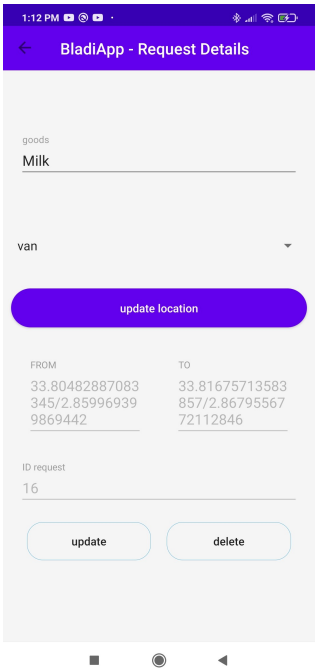

Figure 3.12: Edit Request Screen

Request and Bids this fragment displays the request with all the offers (bids) on that request. the client can also chose a distance from the spinner that will automatically display the bids in map fragment.The latter displays the positions of all the bidding drivers that are closer to the start position of the departure point of the request than the selected distance.

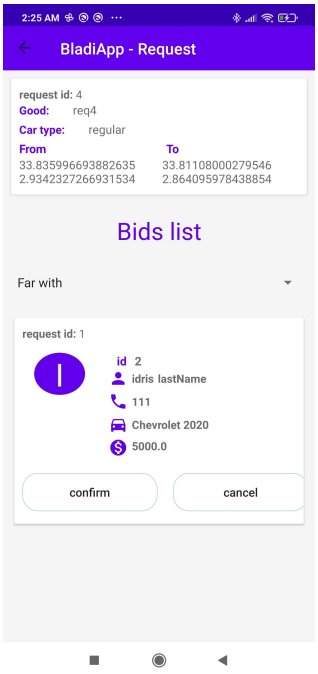

Figure 3.13: Request with bids

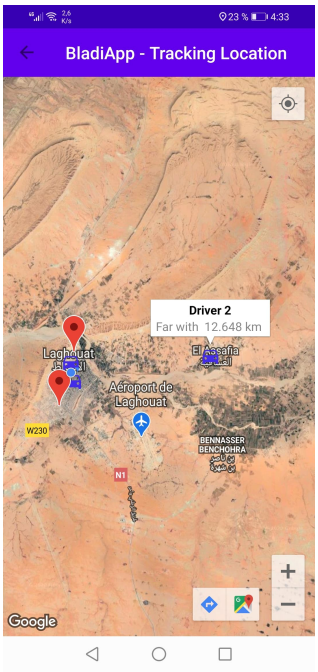

Figure 3.14: Bids in map

Bid Details when clicking on a bid from the list of bids on a request, the app displays the bid details fragment where the client can see all the information related to the offer as well as the current location of the driver who made the offer.

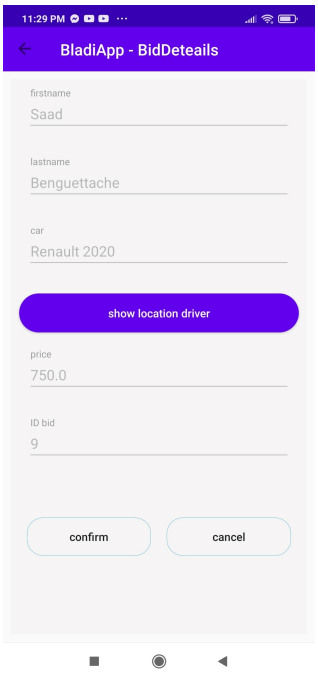

Figure 3.15: Bid Details screen

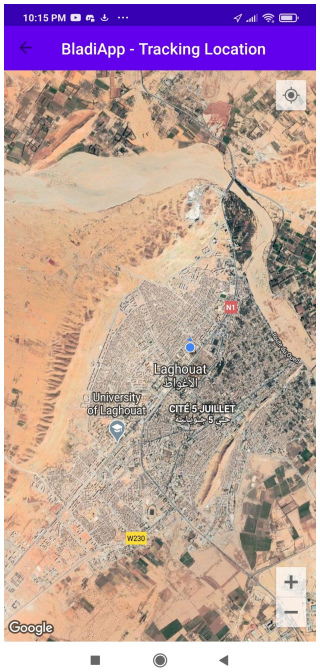

Figure 3.16: Bidding Driver Location

Errands This fragment shows the pending errands of the client that are not marked as complete yet.once the errand is marked complete by both the client and the driver it will disappear from the errands list.

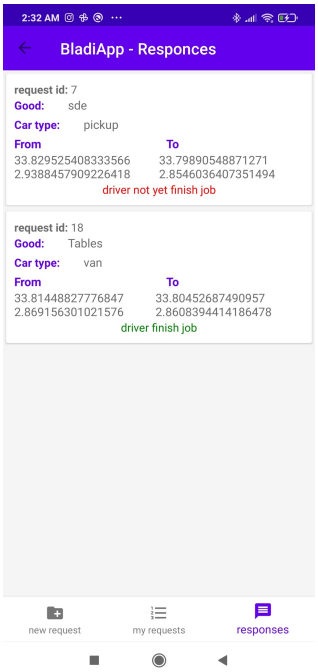

Figure 3.17: Errands Screen

Errand This screen displays the details of a selected errand from the errands list.It also displays the current position of the driver responsible for the errand.The client can mark the completion of the errand after it has been marked by the driver.

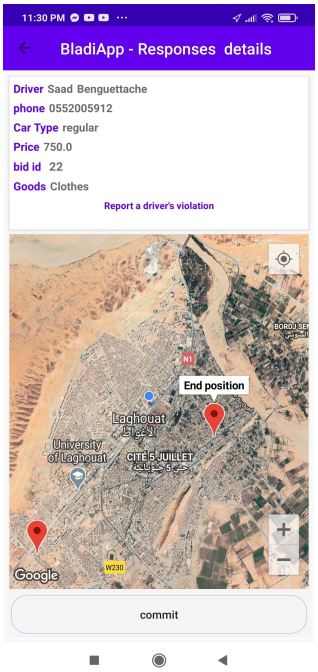

Figure 3.18: Errand Screen

Complaint The client can post a complaint on the driver responsible for the errand if the driver committed any type of behavior that is deemed unacceptable.

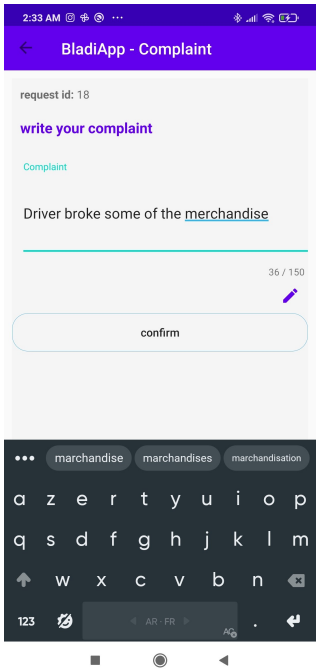

Figure 3.19: Client Complaint Screen

3.6.3 Driver

These functionalities are available to the drivers of the transport service.

Show Requests This Screen displays The requests available to the driver based on the car that he drives as well as the distance that he selected.It is the first screen that the driver sees after entering driver mode.The driver can also click the floating map button to display all request coordinates in a map fragments.

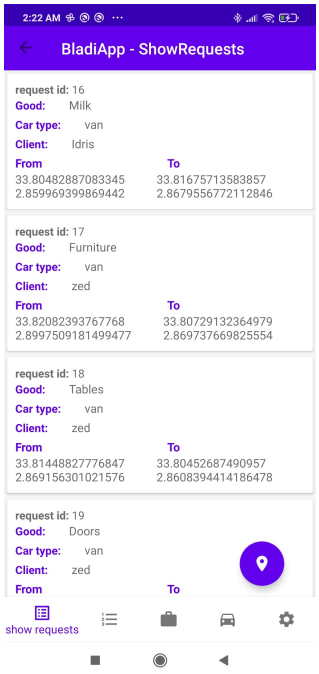

Figure 3.20: Show Requests Screen

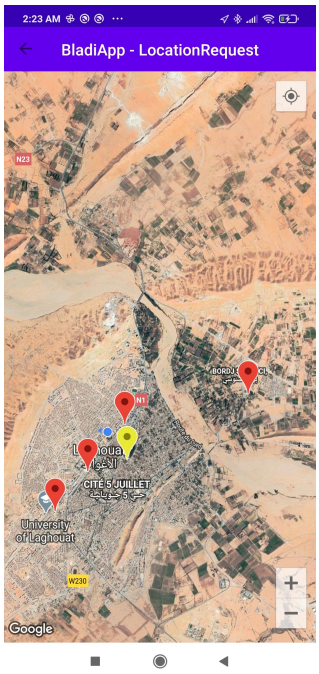

Figure 3.21: Show Requests in map

Bid In this fragment, the driver can view the details of a selected request and they can name their price and send the offer by clicking the bid button.

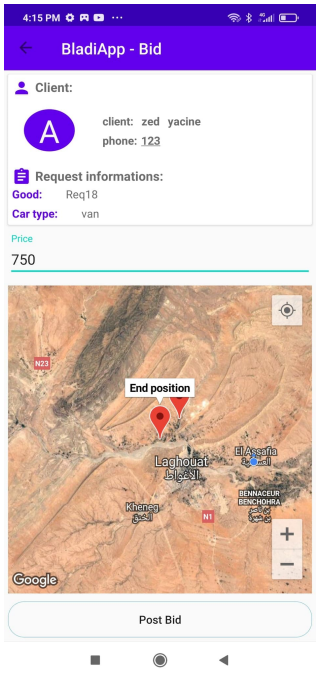

Figure 3.22: Bid Screen

My Bids In this screen, the driver can view the bids they made that are not yet confirmed or rejected by the client.They have the option of either calling the client for further information by clicking on the bid, or canceling the bid by clicking the cancel button.

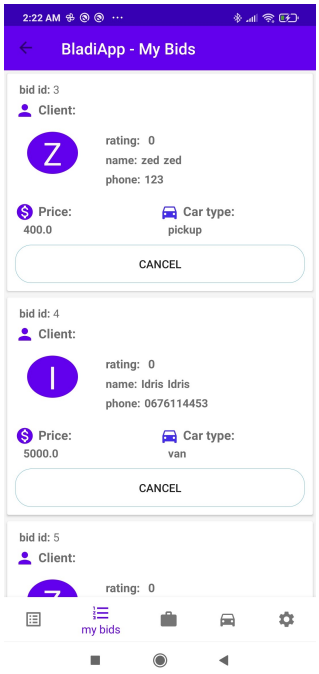

Figure 3.23: My Bids Screen

My Jobs This screen displays the pending (not yet completed) jobs that the driver has to do.The jobs are generated when the client accepts an offer from the driver.The driver's bid is then removed and a new job is created.

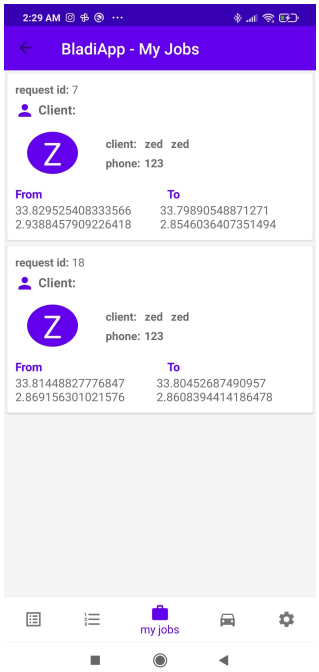

Figure 3.24: My Jobs Screen

Job This fragment displays job details.The driver has the option of reporting the client if the latter committed any type of behavior that is deemed unacceptable.There's also the option of calling the client by clicking on the displayed phone number.Once the job is complete, the driver confirms its completion by clicking commit.

Figure 3.25: Job Screen

File Complaint The driver can report the client for a misbehavior.

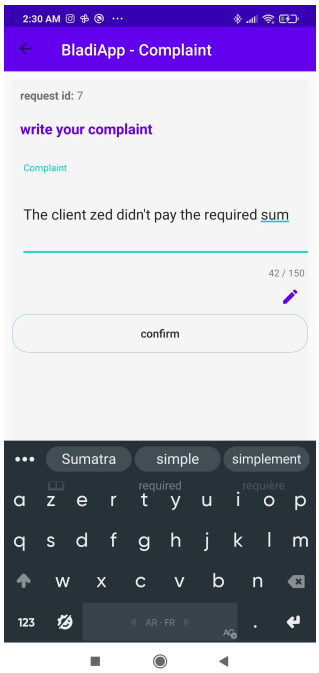

Figure 3.26: File Complaint Screen

My Cars In this screen The driver can view the vehicles he registered.The driver can add new vehicles by clicking the + floating button in the bottom of the fragment.Clicking on a vehicle item opens up the edit vehicle interface.

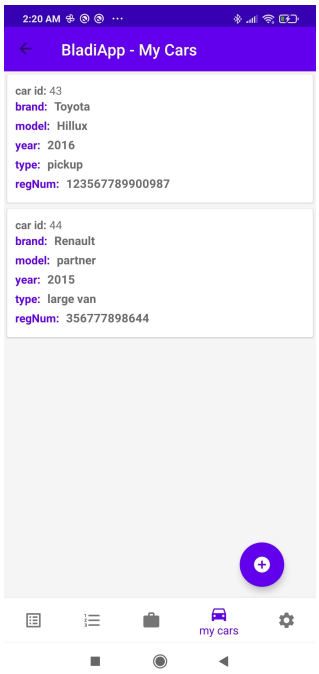

Figure 3.27: My Cars Screen

Register Vehicle In this fragment, the driver can add new vehicles.This is also the first fragment that's displayed to the regular user when they enter driver mode.

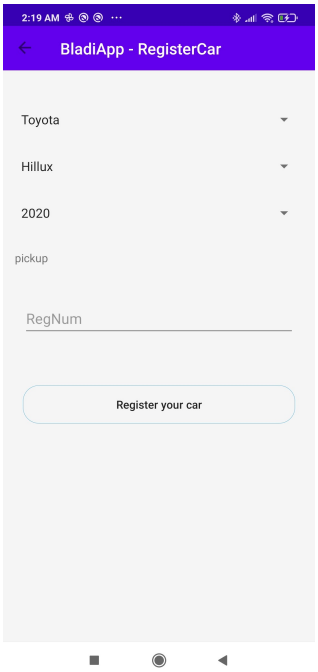

Figure 3.28: Register Car Screen

Edit Vehicle In this fragment, the driver can edit their vehicle in case a mistake was made by them while registering the vehicle.

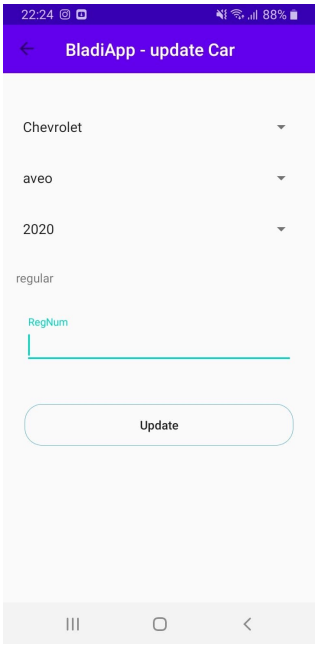

Figure 3.29: Edit Car Screen

Configure In this fragment, the driver can choose a configuration (settings) like whether or not he is available and working, the vehicle that they're currently driving.And the distance that specified the size of area they are willing to work in.

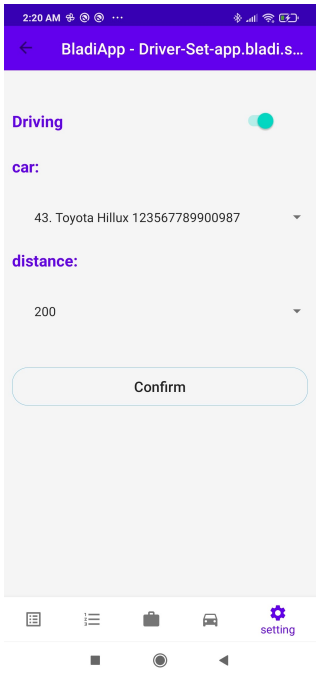

Figure 3.30: Set Config Screen

3.7 Conclusion

In this chapter we revealed the final implementation and design of our work via screenshots of the different interfaces that represent the functionalities of our platform.We have also talked about the hardware, software and architecture we used for its development.

General Conclusion And Future Work

General Conclusion

Our work was inspired by a lack of a real solution to the problem of local and national freight transportation.We aimed for our platform to be the link between possible clients and drivers to make transportation more widely available and convenient.In the first chapter we talked about transport in general and we gave a historical view on it.Then we emphasized on the importance of road transport and the effects and implications of the current technological development on the field.We then gave a view on some related works as well as compared them to each other from different aspects. In the second chapter we defined the structure of our platforms through UML diagrams that helped us make up our minds on how to implement the system's functionality. Finally in the third chapter we listed the hardware used for the development and testing of this work and we detailed our choices in software and architecture.Then we provided screenshots to further clarify the functionality of the system in more detail.

What we have learned Through the duration of our project we have learned many new things that helped change and shape our perspective on software development:

- We learned a new programming language (Kotlin).
- We learned a new development architecture(MVVM) that helped us better organize our code.
- We learned to use new programming tools (Android Jetpack, Grpc) that have facilitated the development of our platform.
- We learned how to conduct a proper research and organize our findings.

- Finally we learned how to write a proper thesis using latex.

Future work

We plan to further develop this system in the future inshallah.We thought we may start by adding the following functionalities:

- A rating system: where the client and driver rate each other based on their experience after the completion of a job.
- An admin console to better manage complaints and registration requests.

Bibliography

[1] *Commercial Vehicle sales worldwide*, https://www.statista.com/statistics/265902/worldwide-commercial-vehicle-sales/, Accessed september 18, 2020.

[2] *Transport Definition* , https://en.wikipedia.org/wiki/Transport, Accessed september 18, 2020.

[3] *Transport History*, https://en.wikipedia.org/wiki/Transport#History, Accessed september 18, 2020.

[4] *New York Taxis*, https://en.wikipedia.org/wiki/Taxicabs_of_New_York_City#Changes_to_cabs, Accessed september 18, 2020.

[5] *Status Of Uber Drivers*, https://www.insurancejournal.com/news/west/2019/09/12/539645.htm, Accessed september 18, 2020.

[6] *Land Transport in Algeria: Number of Vegicles Registered* , https://www.ceicdata.com/en/algeria/land-transport-number-of-vehicles/land-transport-no-of-vehicles-registered-passengers-cars, Accessed september 18, 2020.

[7] *Current and projected numbers of ride sharing users worldwide* , https://buildfire.com/uber-statistics/, Accessed september 18, 2020.

[8] *UML Diagrams*, https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/#activity-diagram, Accessed september 18, 2020.

[9] *Android Studio*, https://en.wikipedia.org/wiki/Android_Studio, Accessed september 18, 2020.

[10] *Kotlin Definition*, https://en.wikipedia.org/wiki/Kotlin_(programming_language), Accessed september 18, 2020.

[11] *SQL Definition*, https://en.wikipedia.org/wiki/SQL, Accessed september 18, 2020.

[12] *MySQL* , https://en.wikipedia.org/wiki/MySQL, Accessed september 18, 2020.

[13] *gRPC* , https://en.wikipedia.org/wiki/GRPC, Accessed september 18, 2020.

[14] *Protocol Buffers* , https://en.wikipedia.org/wiki/Protocol_Buffers, Accessed september 18, 2020.

[15] A. Chugh, *MVVM Architectural Pattern*, https://www.journaldev.com/20292/android-mvvm-design-pattern#android-mvvm, Accessed september 18, 2020.

[16] A. Osmani, *MVC vs MVP vs MVVM*, https://www.oreilly.com/library/view/learning-javascript-design/9781449334840/ch10s09.html, Accessed september 18, 2020.