HantourGo

Graduation Project, Part-II (CS482)

Computer Science Department

Faculty of Computer and Information

Luxor University

Project Advisors:

Dr. Hussein Elshafie

Submitted by:

Alamir Hussein Allam

Mahmoud Mohamed Faraaj **Mostafa Mahmoud Abdelmalek**

Sara Ahmed Abdullah **Mohand Mohamed Abdelshafiy**

ABSTRACT

We also know that horse-drawn carriage rides are one of the most popular activities among tourists, especially in Luxor and Egypt in general.
So we find the stagecoach abundant in those places, which may cause many problems that we do not need, as follows:

1- Obstruction of traffic
2- Reducing pollution from horse waste

And with the continuous development of information technology, this development pushes us to find solutions to the above-mentioned problems, which pose a direct threat to every tourist country, through mobile applications acting as intermediaries between car owners and tourists. The method for the tourist and working for the benefit of both parties is to inform the owner and tourist of the nearest place, to let them know the required fare within an hour of taking the stagecoach, and to enter his price. Filling in the fare in the app's profile, the traveler accepts or rejects the fare and determines the appropriate payment method for both parties.

What does this program offer drivers?

- After reaching a certain number of rides, a 10% return will be provided as a reward
- the driver with the highest score will appear in front of tourists first.
- whoever maintains a high rating within a month gets a code that can be used in the Agricultural Association to obtain a discount on the price of fodder to feed the horse by exchanging the horse dung and using the discount code for the fodder.

Contents List : Part I

Chapter 1

Introduction

1.1 Related tools

Table 1. Abbreviation

Keyword	Meaning
mobile application or **app**	Is a computer program or software application designed to run on a mobile device such as a phone, tablet, or watch. Mobile applications often stand in contrast to desktop applications which are designed to run on desktop computers, and web applications which run in mobile web browsers rather than directly on the mobile device.
Hybrid app	The concept of the hybrid app is a mix of native and web-based apps. Apps developed using Apache Cordova, Flutter, Firebase, React Native, Sencha Touch, and other frameworks fall into this category. These are made to support web and native technologies across multiple platforms. Moreover, these apps are easier and faster to develop. It involves use of single codebase which works in multiple mobile operating systems.
APIs or application programming	Are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols. For example, the weather bureau's software system

interface	contains daily weather data. The weather app on your phone "talks" to this system via APIs and shows you daily weather updates on your phone.
GPS **coordinates**	Are formed by two components that are a **latitude,** giving the north-south position, and a **longitude**, giving the east-west position. Use this map to convert any address in its GPS coordinates. You can also find the location of any GPS coordinates, and geocode its address if available.
Web Scrapping	Our application will then extract data related to the surroundings of a specific geographical location through Google Maps This is to guide drivers and provide them with sufficient information about a place It will also tell the riders about the appropriate and usual price to continue from one particular place to another and the time that the trip will take based on prior statistics from Google Maps. The incredible amount of data on the Internet is a rich resource for any field of research or personal interest. To effectively harvest that data, you'll need to become skilled at web scraping. The Python libraries requests and Beautiful Soup are

	powerful tools for the job. If you like to learn with hands-on examples and have a basic understanding of Python and HTML, then this tutorial is for you.

1.2 Introduction

The importance of tourism :

the construction and further development of the economy. Therefore, tourism in Egypt is one of the main sources of national income and one of the most important factors for the development and prosperity of the country as it earns hard foreign exchange and provides a lot of employment opportunities for individuals

Relationship between tourism and modern technologies:

Tourism activity is greatly affected by information technology, and information technology today is the backbone and the main pillar of the tourism industry, as the availability of information depends on making the decision to travel. Indeed, we find that information technology has entered many tourism sectors, including:

- Tourism companies
- Hotels
- Fly

What is the program offers:

In cities where our app operates, use the app to request a ride.

When a nearby driver accepts your request, the app displays an estimated time of arrival for the driver heading to your pickup location. The app notifies you when the driver is about to arrive.

The app also provides info about the driver with whom you will ride, including first name, price ride, and license plate number.

Use the app to enter your preferred destination any time before or during the ride. If you have a preferred route, it's helpful to talk

through the directions together.

When you arrive at your destination and exit from the car, your trip ends.

Your fare is automatically calculated and charged to the payment method you've linked to your account.

Somewhere, Our App allows you to pay your fare in cash. This option must be selected before you request a ride of after as you like.

Immediately after a trip ends, the app will ask you to rate your driver from 1 to 5 Stars. Drivers are also asked to rate riders. our feedback system is designed to foster a community of respect and accountability for everyone.

1.3 Abbreviation

Title	Authors	Year	Abstract
application programming interface (**API**)	British computer scientists Maurice Wilkes and David Wheeler	1940s	A way for two or more computer programs to communicate with each other. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build or use such a connection or interface is called an API specification. A computer system that meets this standard is said to implement or expose an API. The term API may refer either to the specification or to the implementation.
Software as a service (SAAS)			Is a way of delivering applications over the Internet—as a Service. Instead of installing and maintaining software, you simply access it via the Internet, freeing yourself from complex software and hardware management.
Dijkstra's algorithm	computer scientist Edger W. Dijkstra	1956	s an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks.
Web Scraping	British scientist Tim Berners-Lee	1989	Web scraping is a way of gathering data from web pages with a scraping bot, hence the whole process is done in an automated way. The technique allows people to obtain web data at a large scale fast. In the meantime, instruments like Regex (Regular Expression) enable data cleaning during the scraping process, which means people can get well-structured clean data one stop.
Users			Both Driver and Customer,

1.4 Background

Coach drivers are the most used category for this application; Where the application is considered as an intermediary between the customer and the cabriolet drivers, it helps the client to find the nearest cabriolet from the place where he is located and know the fare prices of the cabriolet driver before riding with him and also agree on the duration of the trip.

This application aims to benefit all users, as it guarantees the customer access to his destination as quickly as possible and in a safe manner. He can also evaluate the driver who rode with him, whether positively or negatively, in order to improve the performance of the service provided to the customer.

We will also provide the application with information about the nearest places, hotels and hospitals to the customer's destination by extracting data from Google Map API.

We will also motivate drivers to perform the service better by having the highest rating for a certain period of time be given a discount code on the price of fodder for the horse, then he will give the code to the nearest agricultural association and replace the horse manure with the remainder of the fodder price.

1.5 Similar Work

- ### Uber Application

Uber is a technology company. It has a smart phone application that employs the GPS system to connect drivers and passengers who are in the nearest place to them. The passenger can evaluate the driver's performance on each trip. Also, the driver evaluates the passengers on each trip. Which helps passengers to choose the driver with a good rating to ride with him. The driver also plays his role in evaluating the passengers so that the data is recorded in the future in case he is exposed to any legal issue. And the service began in March 2009 in San Francisco, USA. Uber is growing rapidly, reaching more than 250 cities around the world. Some of them are in the Middle East in Cairo, Dubai, Abu Dhabi, Riyadh and Doha. The Uber application is available on all modern phone systems such as iPhone (IOS), Android system and Windows system. Uber operates legally in accordance with the laws of each country.

- ### Indriver Application

The app is a special transportation program available on Android and iOS smartphones, in more than 300 cities around the world.

The application allows you to get a ride at a lower price than a traditional taxi, with many other features.

Chapter 2

Domain Analysis and Techniques

2.1 Domain Analysis

In order to discuss the exploitation of stagecoaches/car in a civilized manner in activating tourism, we present this application to facilitate communication between tourists or all people and riders.

The application provides the income for the carriage drivers by connecting it to customers easily, and it also provides it with the advantage of replacing horse manure with horse food, and this is to maintain the cleanliness of the city and eliminate the bad smells spread in it because of that dung.

One of the important advantages for tourists is not exploiting them and raising the cost of the fare without control, as the application determines the appropriate cost for each trip based on the distance traveled during the trip, and the application allows the tourist to give an evaluation to the driver after the end of the trip and also provides him with choosing the method that suits him when paying.

The application sends a notification to the tourist when passing near the archaeological sites to give him a simple glimpse of the place and its history.

One of the widespread problems is the presence of chariots without a license. The application addressed this problem by making it a condition for the driver to register on it, that he obtain a license and submit the necessary documents for approval.

2.2 Techniques

We will implement this as a back-end and host it on serve and provide web service to can use it on android or desktop or web or other organization.

So… in android:

Application components:

1. Login page: This page has the user name and password, and it also has a button for the registration page if the user is not registered.

2. Registration page: A page that enables the user to register, to be able to use the application as a driver or normal user. Here, the login page differs for the driver or the user, as the required documents differ between each of them .First, for the driver, he must enter his personal data (name, phone number, email, address) and then upload the documents required for the hantour, such as the license, a copy of the hantour, and etc. Secondly, for the user, he must register his personal data. Finally, each of them chooses a password.

3. The main pages for driver:

   A main page: map appears in front of him in which his location is determined, and a notification arrives for him.
   If a customer who is near to him wants to ride with him by clicking on this notification, the customer's location, destination, and contact information with the customer, whether chat or phone number, will be shown to him, and then he can choose either to press the approval button or Rejection.

- Upon arrival at the customer's place, a "start trip" button appears, which he can press to start the application in calculating the distance, and then the button turns to "end trip", which he can press upon completion to show him the cost of the trip. Here, payment options appear that the customer can specify.

- A submenu: through it he can know all the data of his personal page and the balance of the debt that he must pay for the application and the date of its payment.

4. The Normal user will see a page containing:
   - A main page: His location and destination are determined, then all the drivers near him are shown to him, and each of them is evaluated. He can choose any of them, after which he is shown the method of communication with the driver.
   - Option: At the end of the trip, the application asks him to evaluate the driver.

5. Controlling management:

The management control panel The characteristics of the management control panel are characterized by the availability of the following icons, which enable them to follow each of the drivers or passengers using the application, namely: - The main one and it contains

1- The total number of drivers
2- Total number of passengers
3- Total Coaches(Hantour) participating in the application
4- Total trips
5- Last 10 trips
6- Today's earnings

7- total profits
8- Ratings through which passenger ratings for drivers can be viewed.
9- Invoices: by viewing all invoices, exporting them, and searching for a specific invoice for review.
10- Who is online now: It shows a map revealing who is in it and the percentage of activity for each driver, as well as those who are on the map.
11- Passengers: Through it, the administration can view all passengers, modify or delete passengers, view their transaction history, search for passengers, add a user, and send notifications to passengers.
12- Notifications and alerts: through which notifications can be sent to passengers, individually or collectively, and notifications can be sent to drivers individually or collectively as well.
13- View on Map: You can see a map with all available and unavailable users and drivers.
14- Adding a new account: You can add an unlimited number of administrators and control them.
15- Payment settings: You can enter and activate or cancel payment methods easily, enter the tax value, increase the price during peak times, choose the application currency, enter the application commission, and enter the percentage on the order.
16- Discount coupons: You can view the available coupons, with the ability to modify, delete, or add new ones, and you can search for the coupon, and add a discount coupon by a fixed amount or percentage, and enter the coupon expiry date.
17- Coaches(Hantours): Adding cars by writing the name, the cost of the start of the trip and the photo, the cost of each seat, deleting the Coaches easily, the cost of the trip, the cost of each kilo, the cost of each hour or minute, and choosing the calculation that you want the car to operate with.
18- Settings: change the site logo, change the administration mail, change the site name, change the mobile number for assistance, change the administration password.

19- Invoices: View all invoices, export them, and search for a specific invoice.
20- Scheduled Trips: View all scheduled trips.

Chapter 3

Risk and Functional and Non-Functional Requirements

3.1 Risk/Constraints

3.1.1 Risk

Risk	Strategy	Priority	Probability
User may not know how to use the application	Make the UI sample and support a lot of languages to guide the user	Moderate	Moderate
Traffic crowded	Find or search for the best roads to avoid this crowd	Moderate	Moderate
Existence of competition from other companies	Provide advantages and solutions to problems in competing companies	High	High
Malfunction with the server or the API hosting the database	Hosting the database and the several servers	High	Low

3.1.2 Constraints

a. Users should install the application
b. Users should have phone number to register in application
c. Users must have internet on his device and always run when using application
d. Customer should order or book his travel to connect with nearest driver
e. Drivers should accept the order to connect with customer
f. Users should know how to deal with the application.
g. Driver should send his information and paper for work
h. Users and Driver should open permission of location

3.1.3 Privacy policy

In our Application "Hantour" we are committed to protecting your privacy. This policy describes our privacy practices, the personal information we collect, how we use and share it, as well as your choices and rights regarding this information.

This Privacy Policy applies to our Services wherever we provide them, and we will comply with local laws in relation to all practices described in this policy. If there is an inconsistency between this policy and the local law, we will comply with the local law to the extent of the inconsistency

This Privacy Policy applies to our Services wherever we provide them, and we will comply with local laws in relation to all practices described in this policy. If there is an inconsistency between this policy and the local law, we will comply with the local law to the extent of the inconsistency.

"Hantour" application is provided and controlled by the entity indicated in the Terms of Use applicable to your country of residence ("we" or "us").

This policy applies to all “Hantour” users, including all users, any user of the Hantour mobile applications (“Apps”), and any services we provide through the Website or Apps, as well as when you contact us via telephone, email, written correspondence, social media, in-person, or any other means of contact (collectively, “Services”). This policy does not cover how we handle information that we collect about our employees or business associates, including our corporate partners, vendors, and subcontractors, but does apply to our Drivers.

In this policy individuals who utilize our Services are referred to as “Users.” Users that request or receive transportation are referred to as “Riders” and individuals who provide transportation to Riders are referred to as “Drivers.” Capitalized terms that are not defined in this policy have the meaning given to them in the Terms of Use. If you accept the Terms of Use you agree to us dealing with your information in the way described in this policy.

What information we collect

We collect three categories of information about Users: information that user provide, information we collect automatically, and information we obtain from other sources.

We do not collect or process information about user racial or ethnic origin, political opinions or membership of any political association, religious or philosophical beliefs, trade-union membership, genetic data, biometric data.

Information we collect automatically

Location information. We collect Users' location data to enable rides, for user support, for safety and fraud detection purposes, and to satisfy legal requirements. We collect location information (including GPS coordinates and Wi-Fi data) based on your App settings, device permissions, and whether you are using the App as a Rider or a Driver (where applicable):

- Riders: We collect your device's precise location when the App is running in the foreground (App open and on-screen) and when the App is running in the background (App open but not on-screen) from the time you request a ride until it ends. Riders may use the App without enabling it to collect precise location data from their mobile devices by directly inputting your pick-up and drop-off address or coordinates into the App. This may affect the availability of App features.

- Drivers: We collect your device's precise location when the App is running in the foreground (App open and on-screen) and when the App is running in the background (App open but not on-screen) in the Driver mode. We may also collect precise location for a limited time after you exit Driver mode in order to detect and investigate ride incidents.

3.2 Project Plan

phase	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Gathering Information									
Define Requirements									
analysis									
design									
implementation									
Testing and Final Discussion									

3.3 Quality Assurance Plan

- Black box:
  - In this stage, we try the application, test its functionality, and see its performance.

- White box: -
  - Unity testing

    In this stage of testing, we will take every component of our Coach driving system(Hantour) such as web service, API connection, user interfaces, and android application to test them separately.

  - Integration testing:

    In this stage of testing, we will take every component of our Coach driving (El-hantour system) such as web service, API connection, user interfaces, and android application to test them separately.

  - Validation testing:

    Validation testing is the process of ensuring if the tested and developed application satisfies its functionality requirements. The business requirement logic or scenarios have to be tested in detail. All the critical functionalities of an application must be tested here.

  - Alpha:

    In this part, a group of testers in our team test the product in a laboratory environment to ensure efficiency of product and fix errors.

  - Beta:

    In this stage, we make a delivery request, wait for the request to be accepted by one of the drivers, and wait for his arrival.

3.4 Requirements

3.4.1 Functional Requirements

1- A fully functional mobile app that controls work-flow with servers, APIs, Drivers, Customers in organized, accurate, and quick methods.

2- The System must store all required data about drivers (ids, info, location), customer (ids, info, location, trip history, History of previous trips)

3- A user interface that enables drivers and customers to create accounts, book a trip, block a trip, show offers of trips.

4- Use the Web Service as mobile application.

5- API to ensure communication between the server and the application.

6- A server that stores data, requests of customer and locations.

7- A system to select appropriate way to trip.

8- The Customer and the Drivers should be able to begin the booking process.

9- Drivers and Customer should be able to receive booking or canceling report and confirm it through API.

10- The customer's location must be available for easy access.

11- Must do the process of login or registering in the application.

12- A system to display prices for trips or the number of kilometers

3.4.2 Non-Functional Requirements

1- Fast APIs and data retrieval from servers within seconds.

2- A clear, attractive, responsive, less-leggy, and efficient User-interface that guarantees a good user experience and handles a variety of tasks.

3- High security to prevent unauthorized access and guarantee the safety and integrity of data.

4- A well-organized and manageable system architecture that eases the process of maintenance, recoverability and scalability.

5- The speed of reaching to the customer.

3.5 System Request

- Mobile applications help solve many problems on the ground and save time and effort by offering services and facilitating communication between people. The Hantour application helps to increase the income of drivers and provides a way to increase the income of drivers, provide a means of transportation that does not harm the environment, and provide a comfortable and enjoyable way for customers and is available at all times. By booking a full trip to enjoy the historical atmosphere of the city or using it as regular and reliable transportation at reasonable prices with customers and good for drivers.

- Functionality.
  1- The ability to book a ride or take an enjoyable trip by trolleybus or any means of transportation added to the system.

  2- The ability to show the offers of trips

  3- The ability to generate full reports about the trip.

  4- The ability to store trips histories for the customer as a reference for them.

- Expected Value: -
  1- Less-cost for trip and book More-fast

  2- More organized in book trips.

  3- Easy dealings between tourists and drivers generally.

  4- More fast and reliable means of transportation

  5- Ease-to-use application and friendly User-interfaces.

6- Increase productivity.

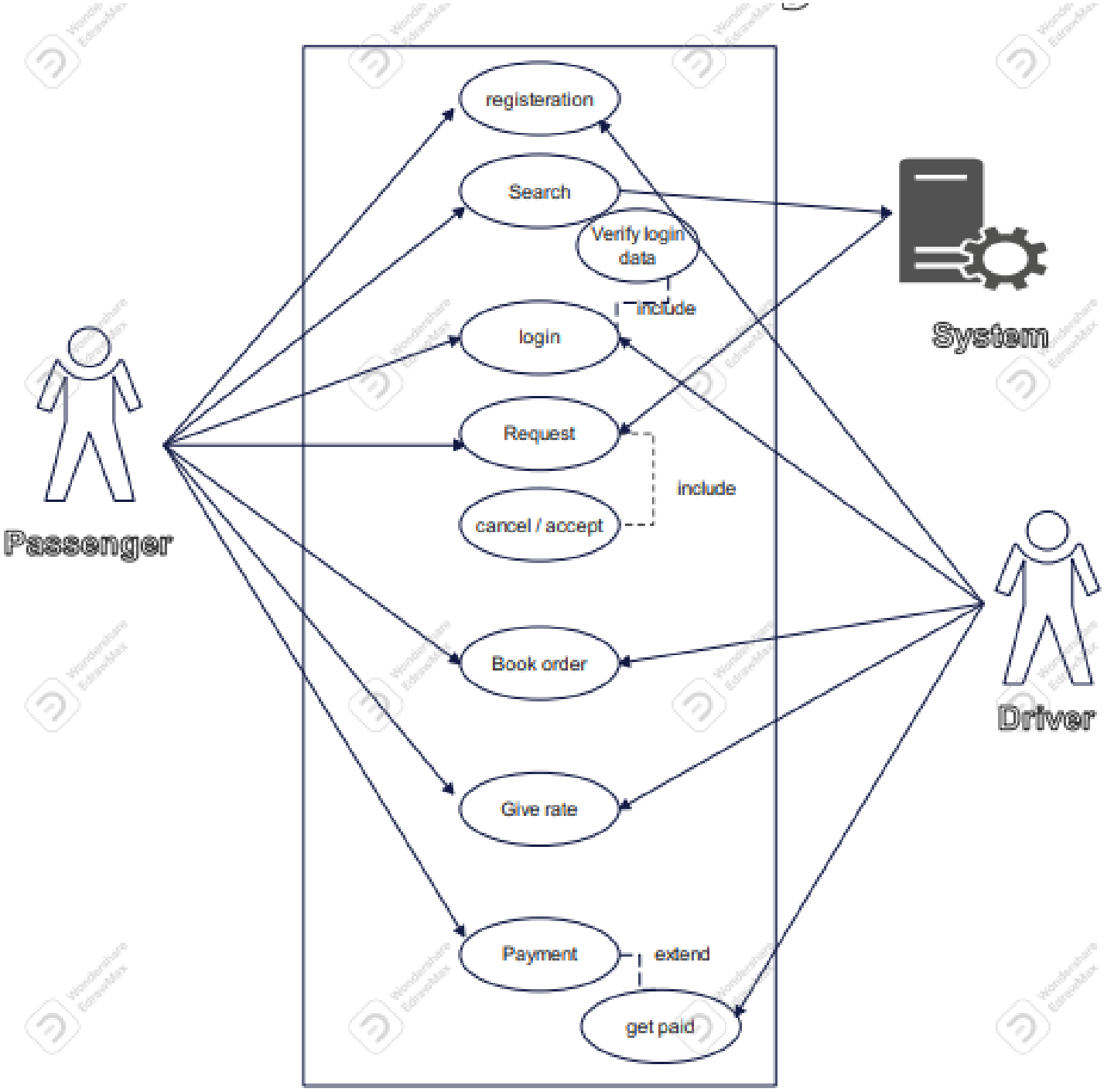
7- Customer-satisfaction.

- Special Constraints: -
  - The fast of replay the request of customer
  - Distance between Customers and Drivers.
  - The short development life-time.

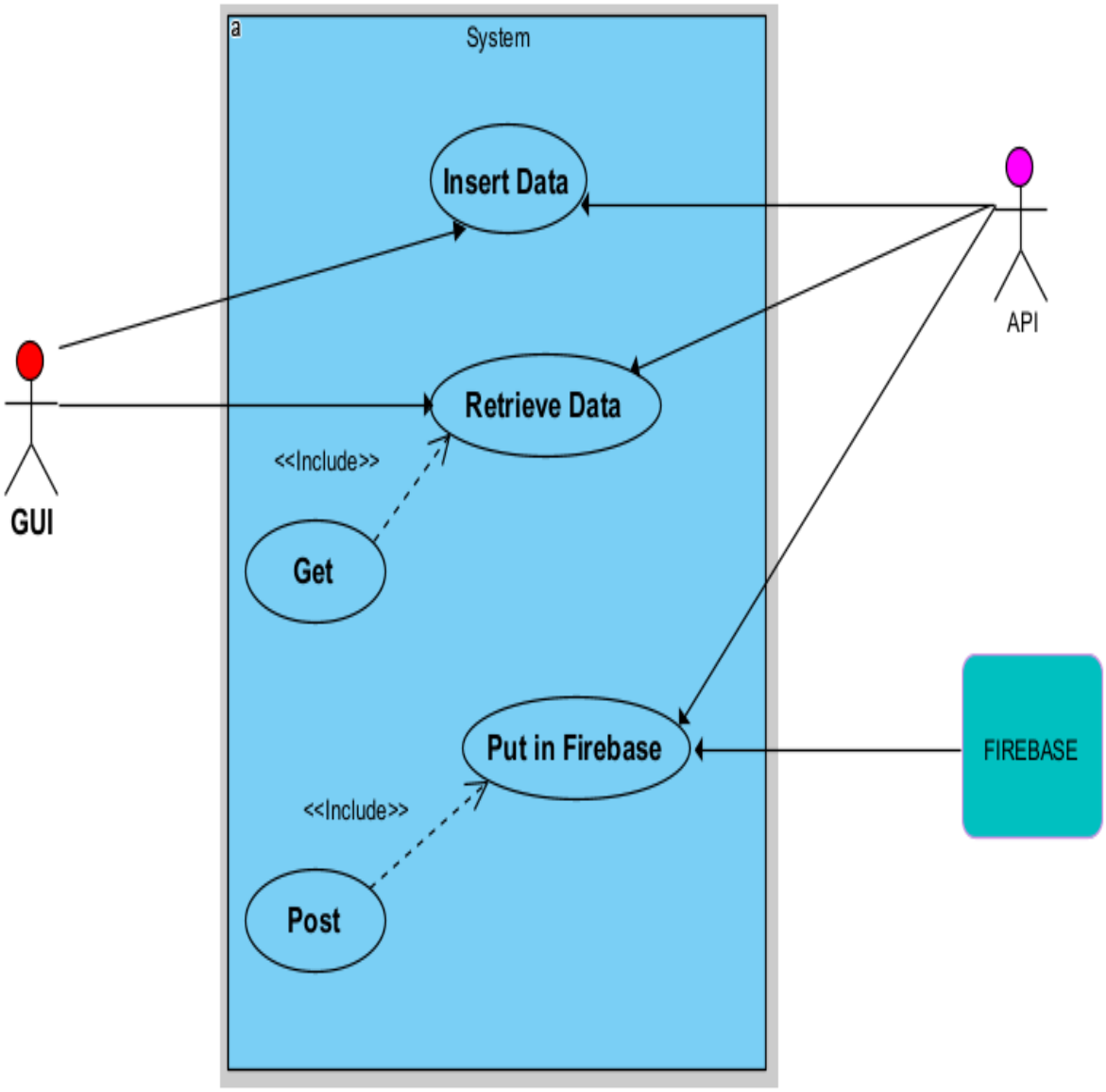
Chapter 4

Proposed System and Methodology

4.1 System Use Case

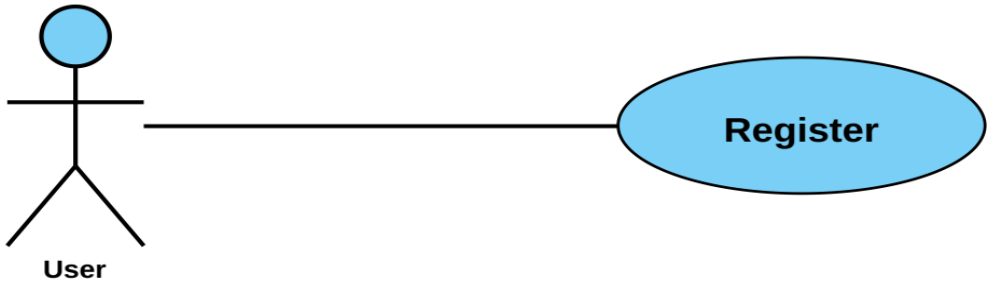


Use Case (GUI, API, and FIREBASE)



4.2 Use Case Description

- Register: user uses app to make account and enter required information

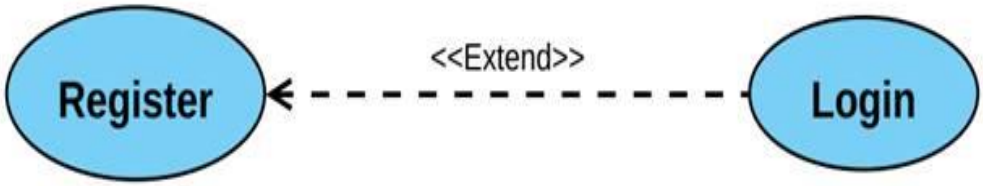


Use case name	Register
Unique ID	Hantour-User-001
Area	Hantour Application
Actor(s)	User (Driver, or Passenger)
Description	User creates account
Triggering Event	User click "Register" button in the application
Preconditions	- The user needs to download application then open it - The user needs to have internet access
Postconditions	- User has successfully create account
Assumptions	- User have Hantour application - A valid data

Steps Performed	**Information for Steps**
1- Open application 2- Choose if he is a Driver or passenger 3- User enters his data 4- Click on "Create Account" button 5- Validation of entered data by application	Step 3: Name, Username, Password, E-mail, SSN, EMSN (Driving License if he is a Driver)

Extensions (Alternative Flows)	- If the downloading interrupted for any reason, use should try again and download it - If user entered a non-valid data, a warning message should appear to him

- Log in: to log in to your account, users should enter username and passwords.

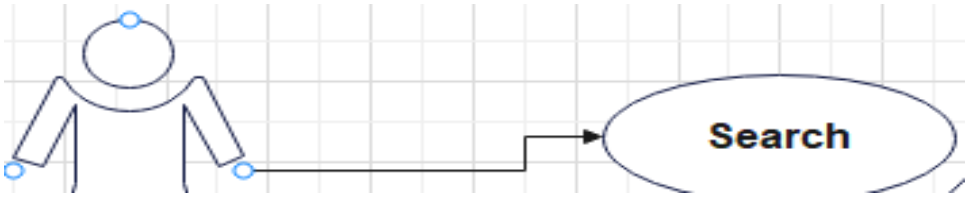


Use case name	Login
Unique ID	Hantour-User-002
Area	Hantour Application
Actor(s)	User(Driver , Passenger)
Description	User login to his account
Triggering Event	User click “Login” button in the application
Preconditions	- The user needs to download application then open it - The user needs to have internet access - The user needs to have account
Postconditions	- User has successfully logged in to his account
Assumptions	- User have GoFast application - A valid data

Steps Performed	**Information for Steps**
1- Open application	Step 2: E-mail, Password

2- User enters his data 3- Click on “Login” button 4- Validation of entered data by application	
Extensions (Alternative Flows)	- If user entered a non-valid data, a warning message should appear to him

- Search: user search about available Drivers to go to his destination.



Use case name	(Search)	**Unique ID**	Hantour-003
Area	Dashboard (Hantour App)		
Actor(s)	User(Passenger)		
Description	Passenger search about available Drivers and suitable price tour		
Triggering Event	Allow for Search for Drivers		
Preconditions	- The user login		
Postconditions	- Account and logging in is already done		
Assumptions	- User must make account		
Steps Performed		**Information for Steps**	

1- User logged in successfully 2- Open dashboard and choose Search button	Valid login
Extensions (Alternative Flows)	- If user doesn't make login, a warning message should appear to user that program couldn't login to his /him account.

 When user search about Nearest Driver and suitable cost too, and the true cost is tightly greater than user want the app ask user to raise the cash to be greater than or equal to true cost then the Driver could accept the order.

- **Book order**: user asked driver to drive him to his/her destination and Driver accepted the order.

Use case name	Book Trip with Driver
Unique ID	Hantour-user-001
Area	Dashboard, Application
Actor(s)	User (Passenger) with Driver when accept the order
Triggering Event	Driver Click Accept the Trip, send to Passenger time and distance Between him and passenger
Preconditions	- Driver and passenger login to them account - Passenger should share his destination with Driver - Driver accept the trip and go on
Postconditions	- Driver and Passenger should have accounts and logged in
Assumptions	- Driver earn money - Passenger has been to his distention quickly

Steps Performed	**Information for Steps**
1- Open application	Step 2: E-mail, Password

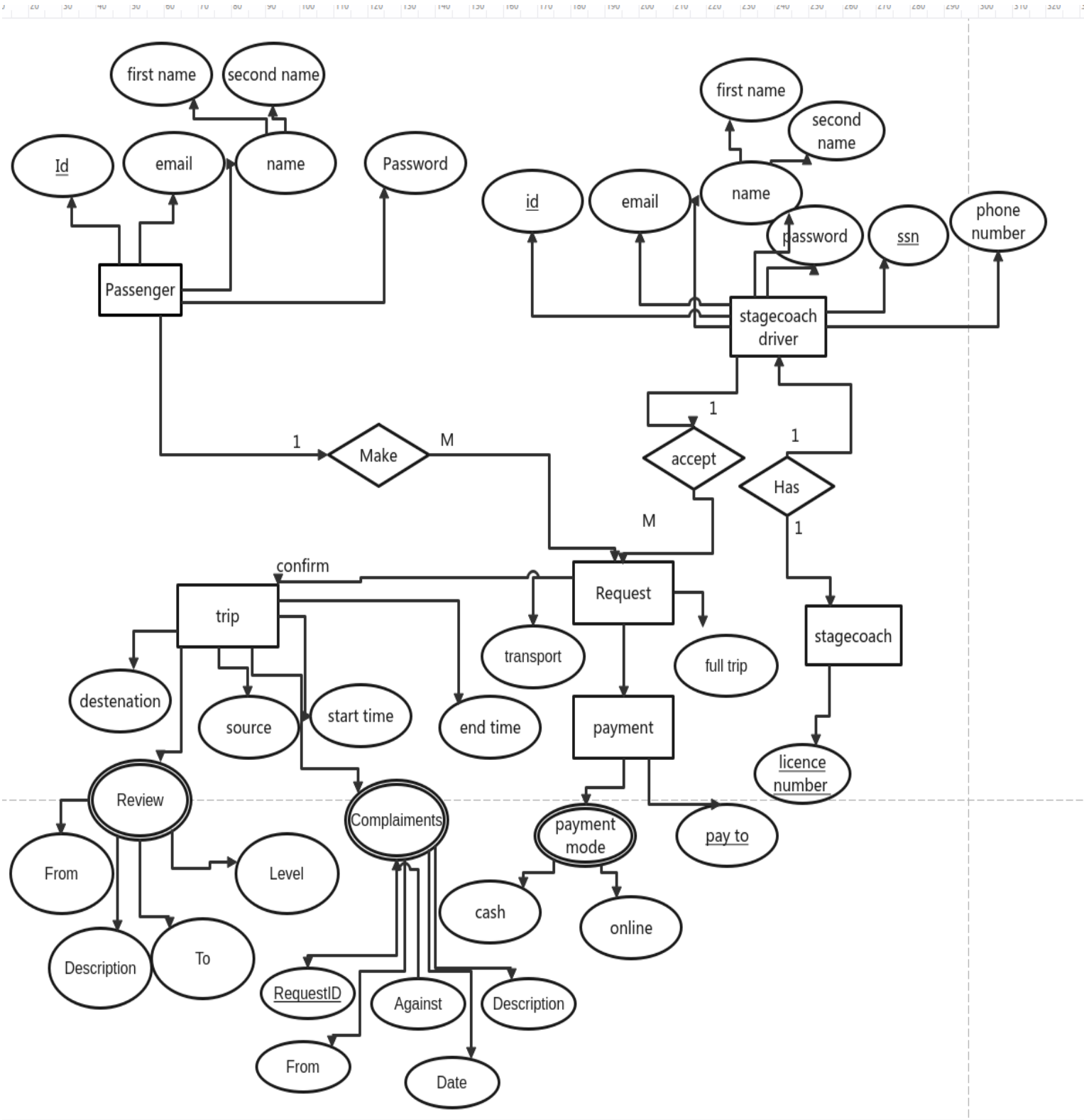
2- Driver and Passenger should log in 3- Passenger Search for his destination 4- Driver look at the Trip requests 5- Driver Accept or reject trip	Step 3:location
Extensions	- If invalid data entered Warning message appea

- Rate the Driver: User(passenger)

  Driver should be rated by his traveler or passengers to get feedback about him, the more his rate is good the more work and money comes.

Use case name	Hantour-user-003	
Area	Dashboard, Application	
Actor(s)	User (Passenger) give rate to Driver using the application rate button	
Preconditions	- Passenger should log in to Application	
Assumptions	- Passenger Give rate about his Driver	
Steps Performed		**Information for Steps**
6- Open application 7- Driver and passenger should log in to app 8- Diver get rated by passenger		
Extensions (Alternative Flows)	- If Passenger entered a non-valid data in login, a warning message should appear to him - If Driver entered a non-valid data in login, a warning message should appear to him -	

4.3 ER Diagram



Entity relationship diagrams

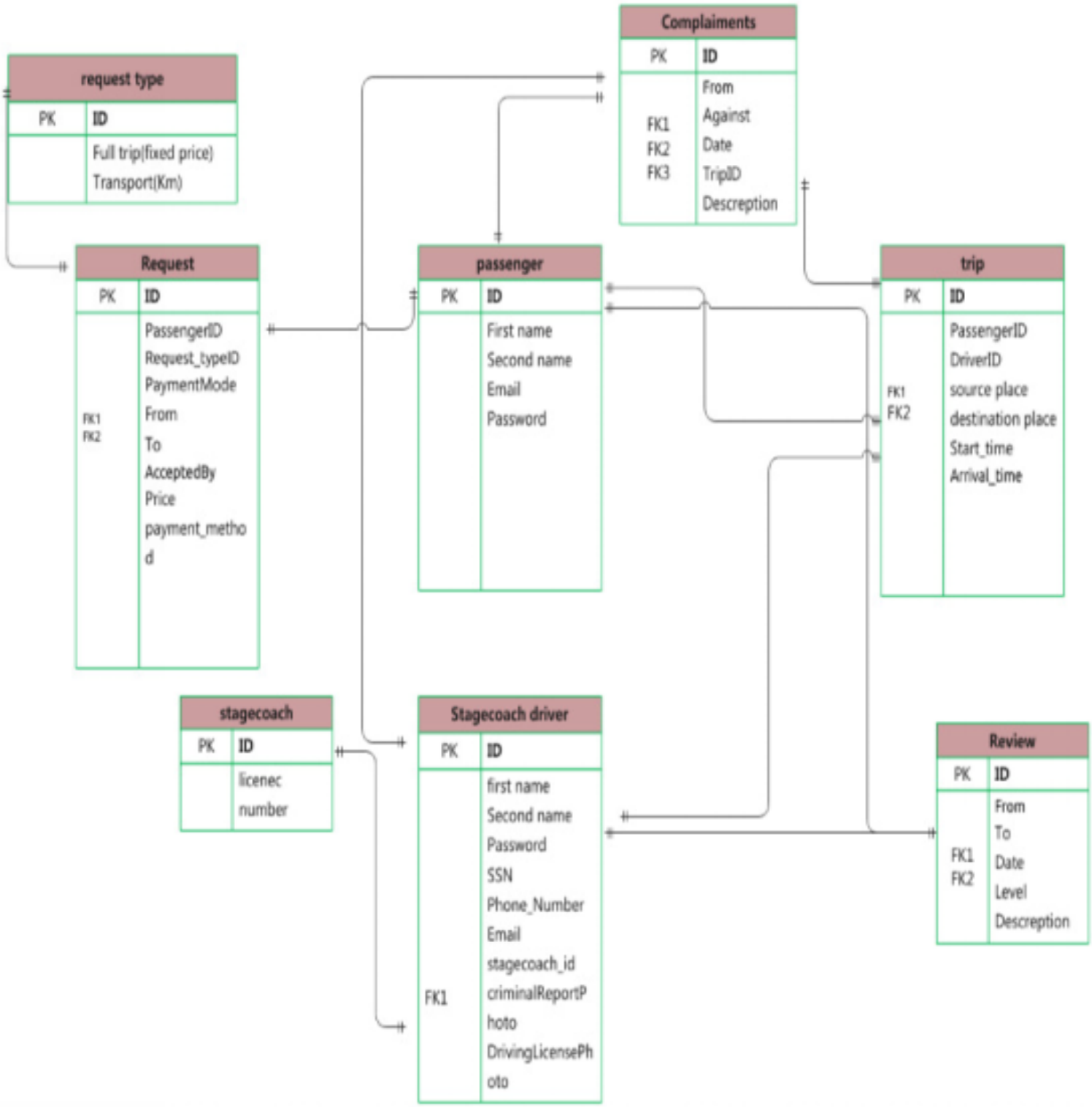
An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. An ERD uses data modeling techniques that can help define business processes and serve as the foundation for a relational database.

Importance of Entity relational diagrams

Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a reference point, should any debugging or business process re-engineering be needed later.

However, while an ERD can be useful for organizing data that can be represented by a relational structure, it can't sufficiently represent semi-structured or unstructured data. It's also unlikely to be helpful on its own in integrating data into a pre-existing information system.

4.4 Database Schema



Defining Database schema:

When constructing the backend of an application, you need to take into account how the frontend will talk to the backend. More important, however, is the construction and design of your database. The relationships your data forms will lead to the construction of your database schema.

A **database schema** is an abstract design that represents the storage of your data in a database. It describes both the organization of data and the relationships between tables in a given database. Developers plan a database schema in advance so they know what components are necessary and how they will connect to each other.

In this guide, we will learn what a database schema is and why they are used. We will go through a few common examples so you can learn how to configure a database schema on your own.

4.5 Sequence Diagrams

- Capture the interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- Capture high-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)

Purpose of Sequence Diagram

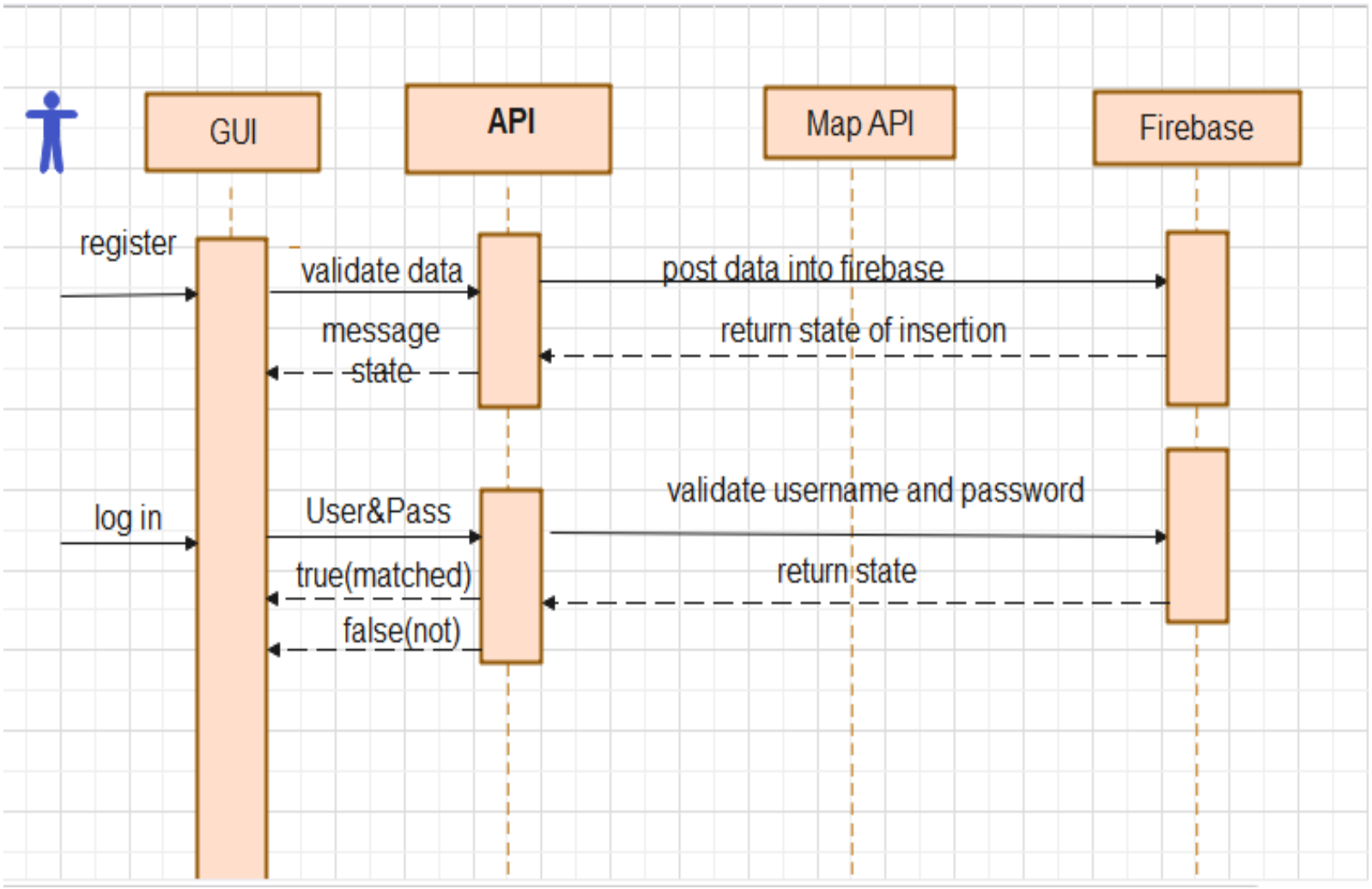
Model high-level interaction between active objects in a system

Model the interaction between object instances within a collaboration that realizes a use case

Model the interaction between objects within a collaboration that realizes an operation

Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

1. **User Sequence diagram(Passenger):**



As we show in sequence diagram User(passenger) should have account to be able to take advantage of our application.

- User should enter Data required to registration (first name, last name, phone number, email address and social security number SSN)
- Our System should validate the Data required to registration before posting it into data base.
- API post validated data to our data warehouse (post function) and return state of success or failure of insertion.
- After registration, user can benefit from functionalities of our application.

- **Registration(Passenger)**

Sequence name	Register
Unique ID	Hantour-User-001
Area	Hantour Application
Actor(s)	User (passenger)
Description	User (passenger) creates account
Triggering Event	User (passenger) click "Register" button in the application
Preconditions	- The (passenger)needs to download application then open it - The (passenger)needs to have internet access - (passenger)should register at the application
Postconditions	- the(passenger) has successfully create account
Assumptions	- the(passenger)have Hantour application account - A valid data

Steps Performed	**Information for Steps**
6- Open application 7- passenger enters his data 8- Click on "Create Account" button 9- Validation of entered data by application	First name last name, phone number, email Date of birth

Extensions (Alternative Flows)	- If the downloading interrupted for any reason, use should try again and download it - If user(passenger) entered a non-valid data, a warning message should appear to him

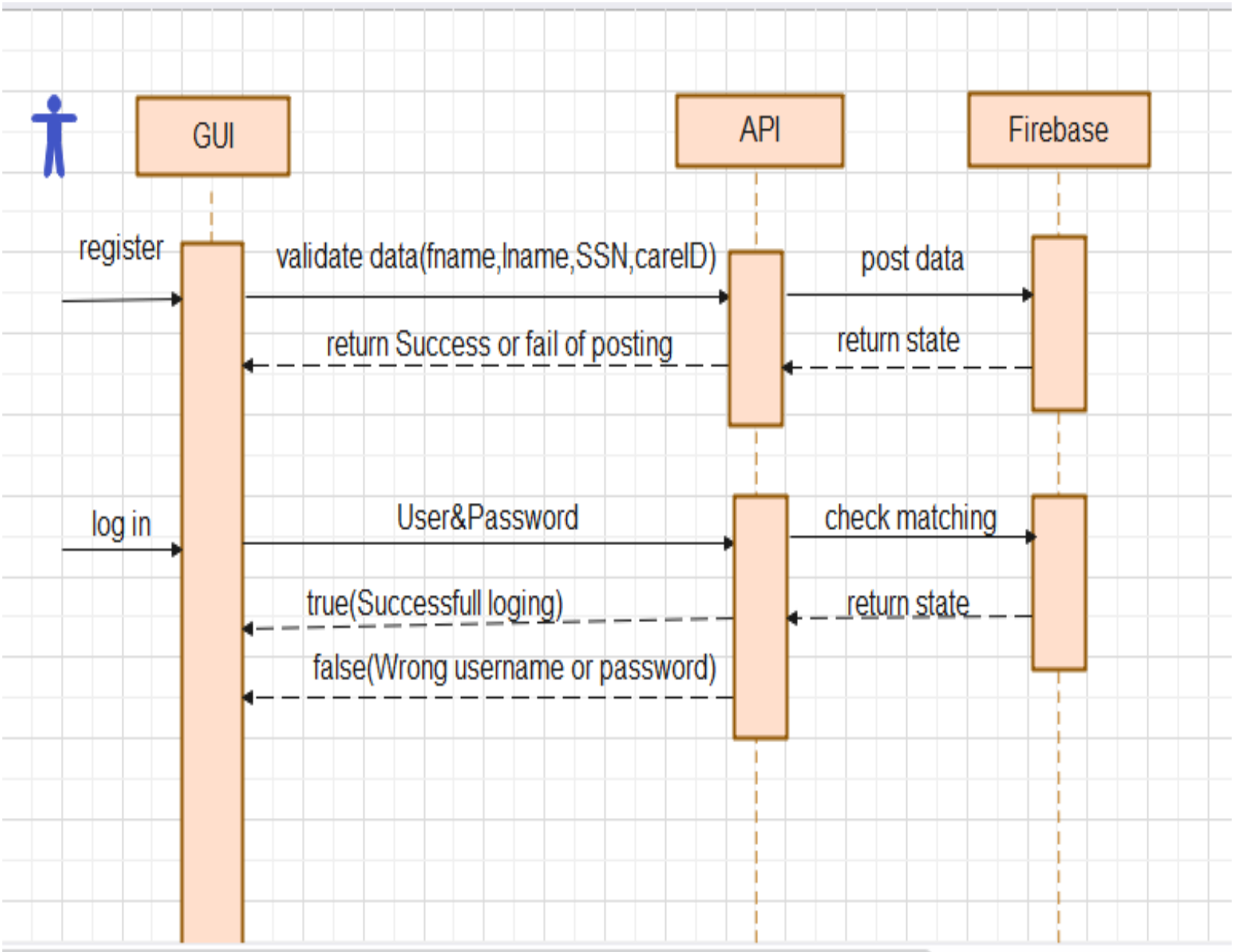
- **Log in process (Passenger)**

sequence name	Log in
Unique ID	Hantour-User-002
Area	Hantour Application
Actor(s)	User (passenger)
Description	User(passenger) creates account
Triggering Event	User (passenger) click "Login" button in the application
Preconditions	- The user (passenger) needs to download application then open it - The user(passenger) needs to have internet access
Postconditions	- User (passenger) has successfully logged in
Assumptions	- User (passenger) have Hantour application - A valid data

Steps Performed	**Information for Steps**
10-Open application 11-User (passenger) enters his Email and password 12-Click on "login" button 13-Validation of entered data by application	Username, Password, E-mail,

Extensions (Alternative Flows)	- If the downloading interrupted for any reason, use should try again and download it - If user (passenger)entered a non-valid data, a warning message should appear to him

2. Driver Sequence Diagram(Driver): -



- **Registration(Driver)**

Sequence name	Register
Unique ID	Hantour-Driver-001
Area	Hantour Application
Actor(s)	User (Driver)
Description	User (Driver) creates account
Triggering Event	User (Driver) click "Register" button in the application
Preconditions	- The (Driver)needs to download application then open it - The (Driver)needs to have internet access - (Driver)should register at the application
Postconditions	- the(Driver) has successfully create account
Assumptions	- the(Driver)have Hantour application account - A valid data

Steps Performed	**Information for Steps**
14-Open application 15-Driver enters his data 16-Click on "Create Account" button 17-Validation of entered data by application	Step 3: Name, Username, Password, E-mail, SSN, EMSN (Driving License if he is a Driver)

Extensions (Alternative Flows)	- If the downloading interrupted for any reason, use should try again and download it - If user(Driver) entered a non-valid data, a warning message should appear to him

- **Log in process (Driver)**

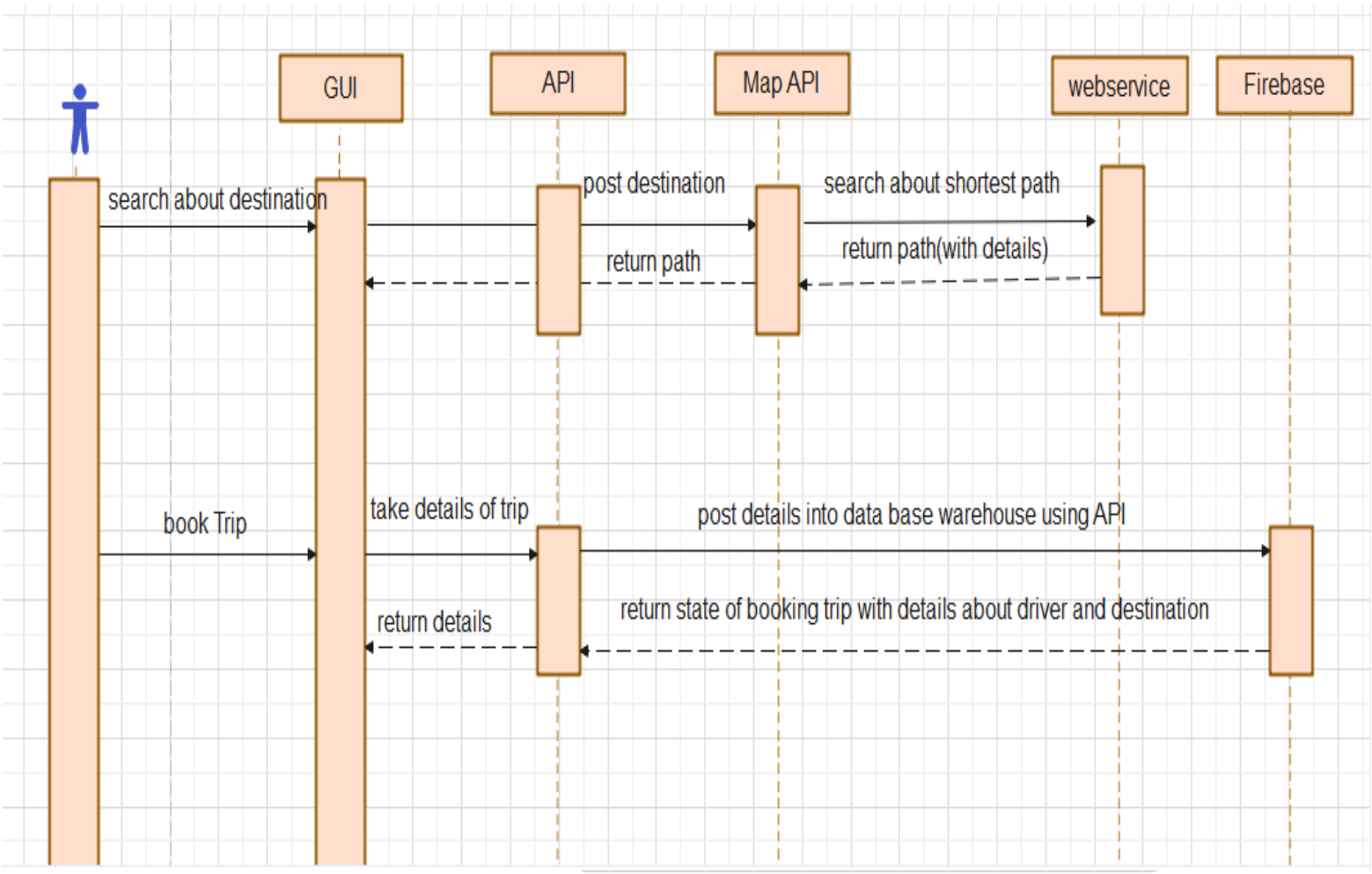
sequence name	Log in
Unique ID	Hantour-Driver-002

Area	Hantour Application
Actor(s)	User (Driver)
Description	User(Driver) creates account
Triggering Event	User (Driver) click "Login" button in the application
Preconditions	- The user (Driver) needs to download application then open it - The user(Driver) needs to have internet access
Postconditions	- User (Driver) has successfully logged in
Assumptions	- User (Driver) have Hantour application - A valid data

Steps Performed	**Information for Steps**
18-Open application 19-User enters his Email and password 20-Click on "login" button 21-Validation of entered data by application	Username, Password, E-mail,

Extensions (Alternative Flows)	- If the downloading interrupted for any reason, use should try again and download it - If user entered a non-valid data, a warning message should appear to him

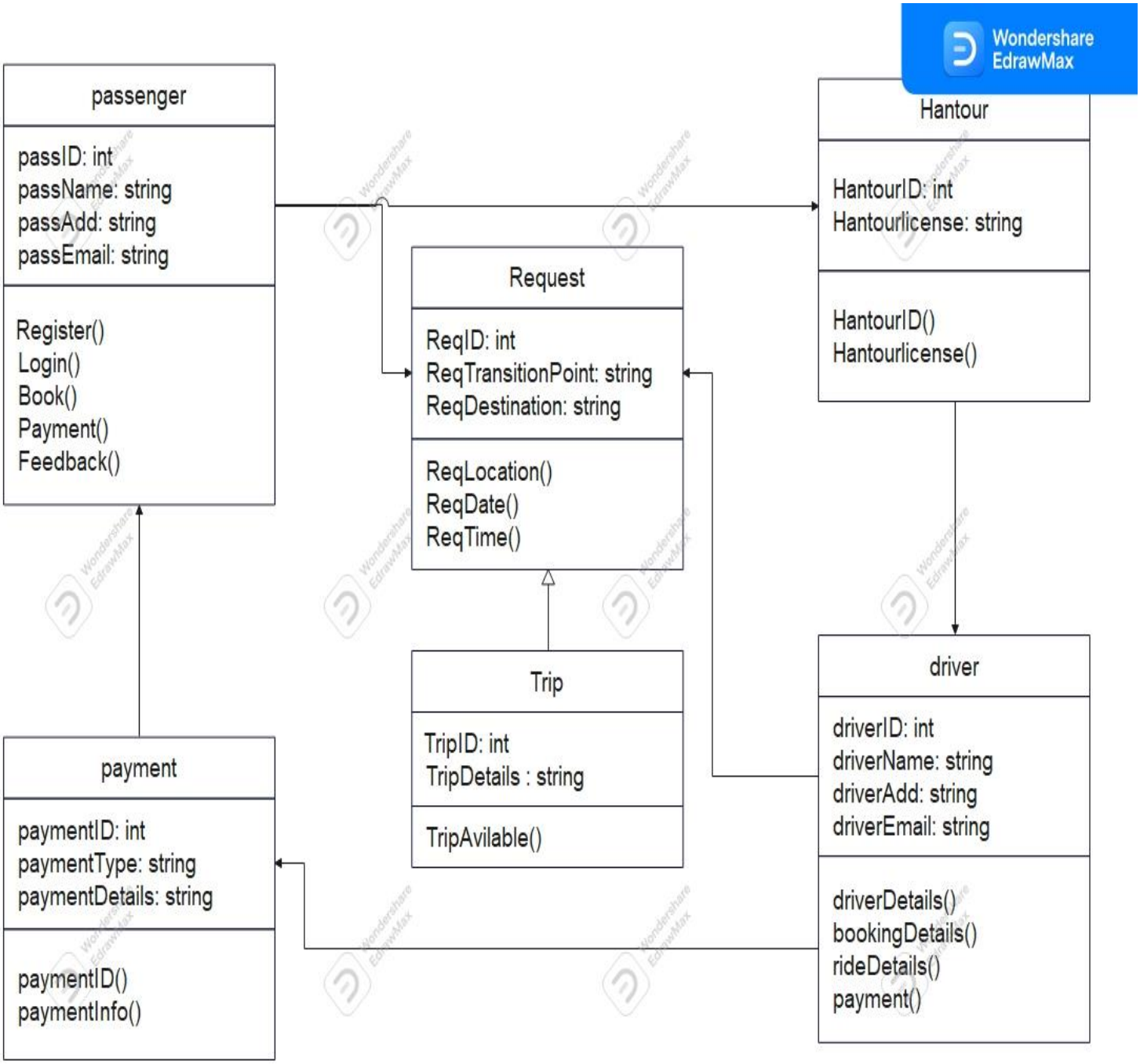
- **Sequence diagram cont.(Passenger): -**



sequence name	Search about destination
Unique ID	Hantour-User -003
Area	Hantour Application
Actor(s)	User (passenger)
Description	User(passenger) search about destination

Triggering Event	User (passenger) choose or search about his /her destination Using maps or by typing the destination name	
Preconditions	- The user (passenger) needs to download application then open it and register. - The user(passenger) needs to have internet access	
Postconditions	- User (passenger) has found his /her destination and seen the price cost.	
Steps Performed		**Information for Steps**
-Open application -User enters his Email and password -Search about destination -Validation of entered data by application		Username, Password, E-mail,
Extensions (Alternative Flows)	- If the downloading interrupted for any reason, use should try again and download it - If user entered a non-valid data, a warning message should appear to him	

4.6 Class Diagram



A **class diagram** is a static structure diagram used to show class relationships in object-oriented programming. It is also a good way to show the class structure of a system. For businesses, these can be used to illustrate class relationships in a business application.

Benefits of using a class diagram

Using a class diagram is a good way for businesses to communicate a class structure to project stakeholders and teams. A class diagram is especially useful for communicating class hierarchies and collaborations between classes.

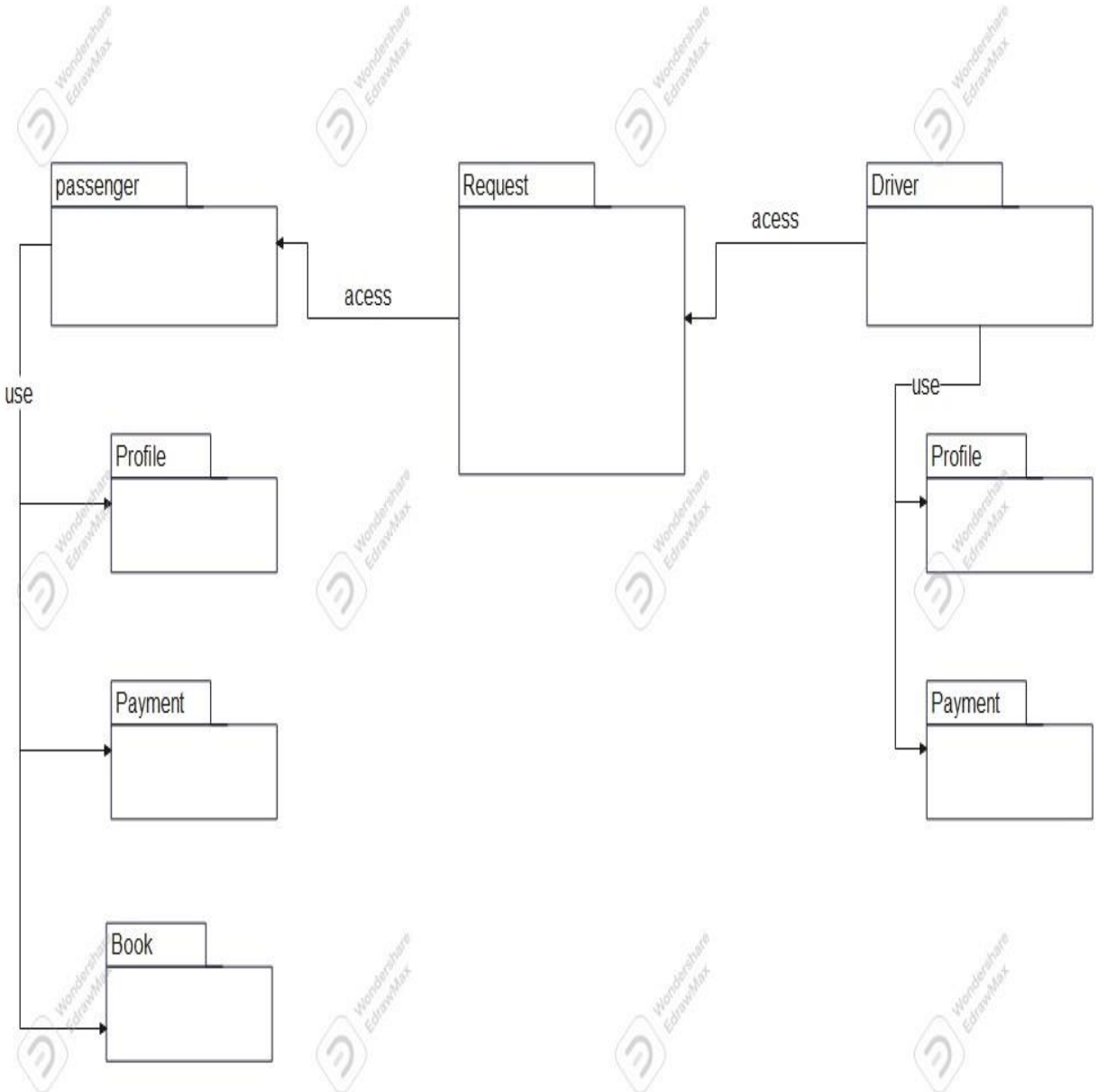
Class diagram notations

Classes – Classes are represented by rectangles with three compartments for class name, attributes, and operations. A rectangle with double borders represents an abstract class that cannot be instantiated. Each subclass is depicted by a rectangle inside each superclass' compartment, with the same notation as the superclass'.

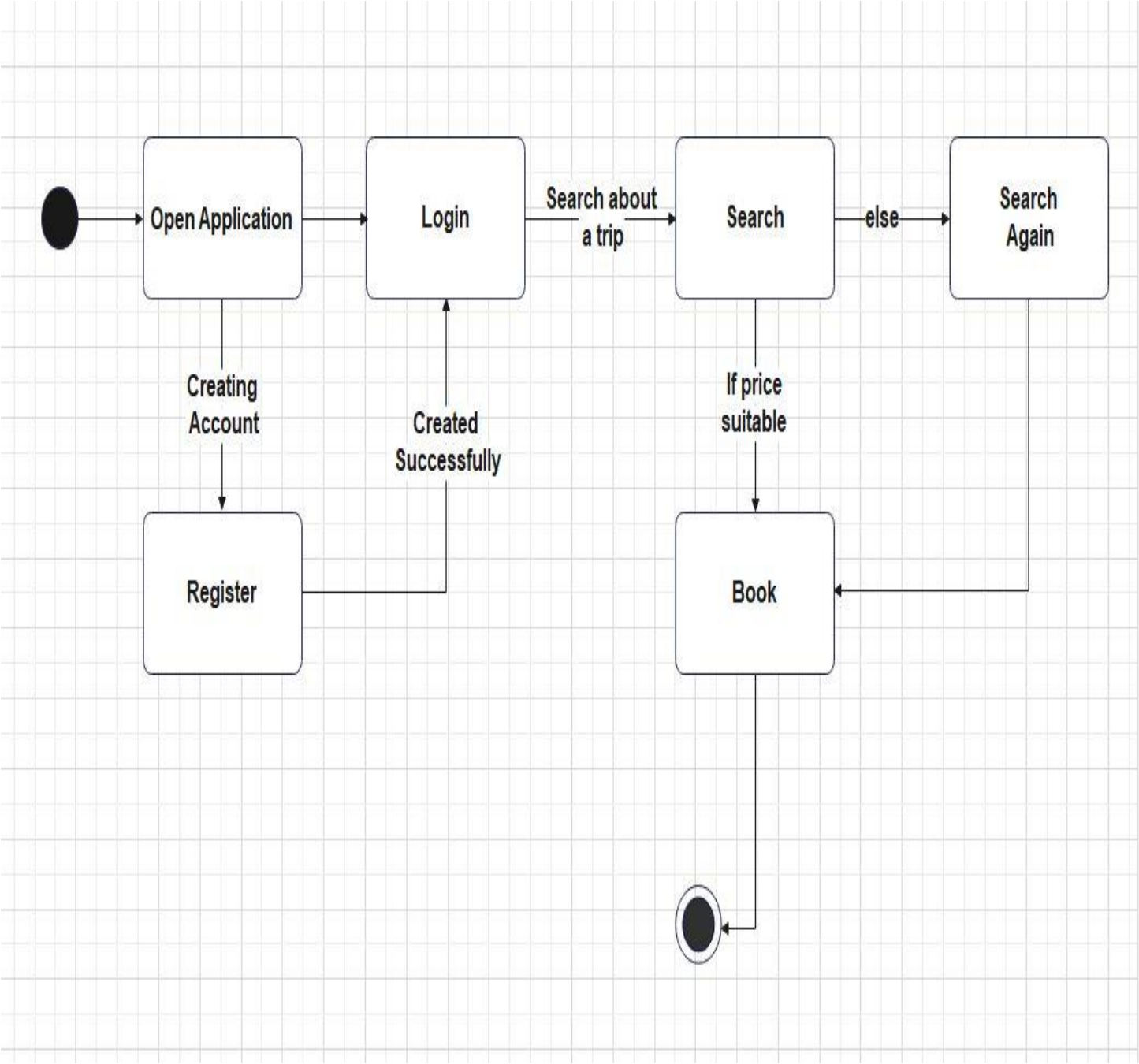
Objects – Objects are represented as ovals that contain class names inside class name compartments. A single object is depicted by an oval that contains only one class name inside the object's class compartment. If there are multiple classes, then the ovals are nested inside each other to represent which objects contain which classes.

Attributes – Attributes are listed inside the attribute compartment of a class rectangle. Attributes can be either attributes or operations of a class depending on whether they are used in an association relationship with another class

4.7 Package Diagram

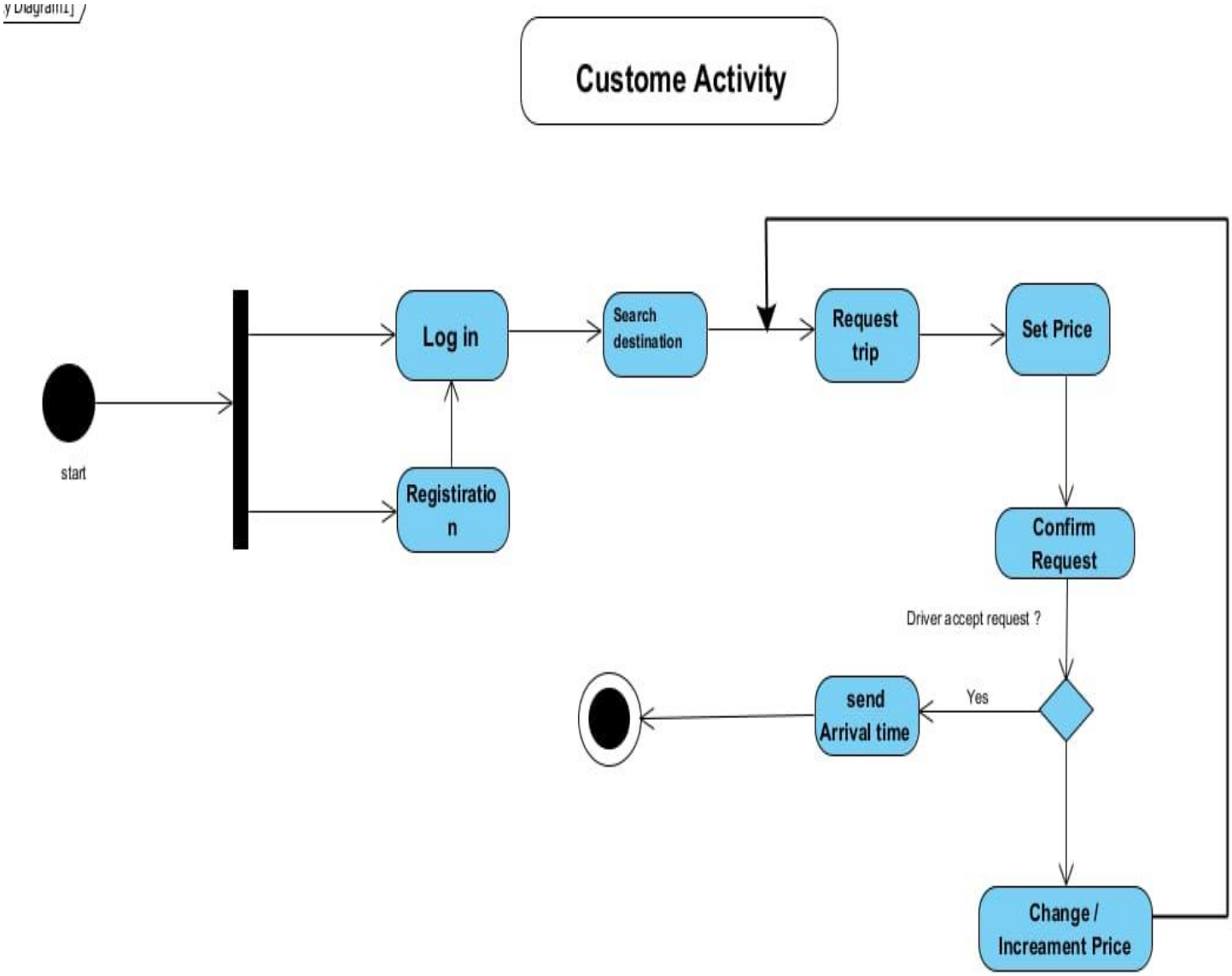


4.8 State Diagram



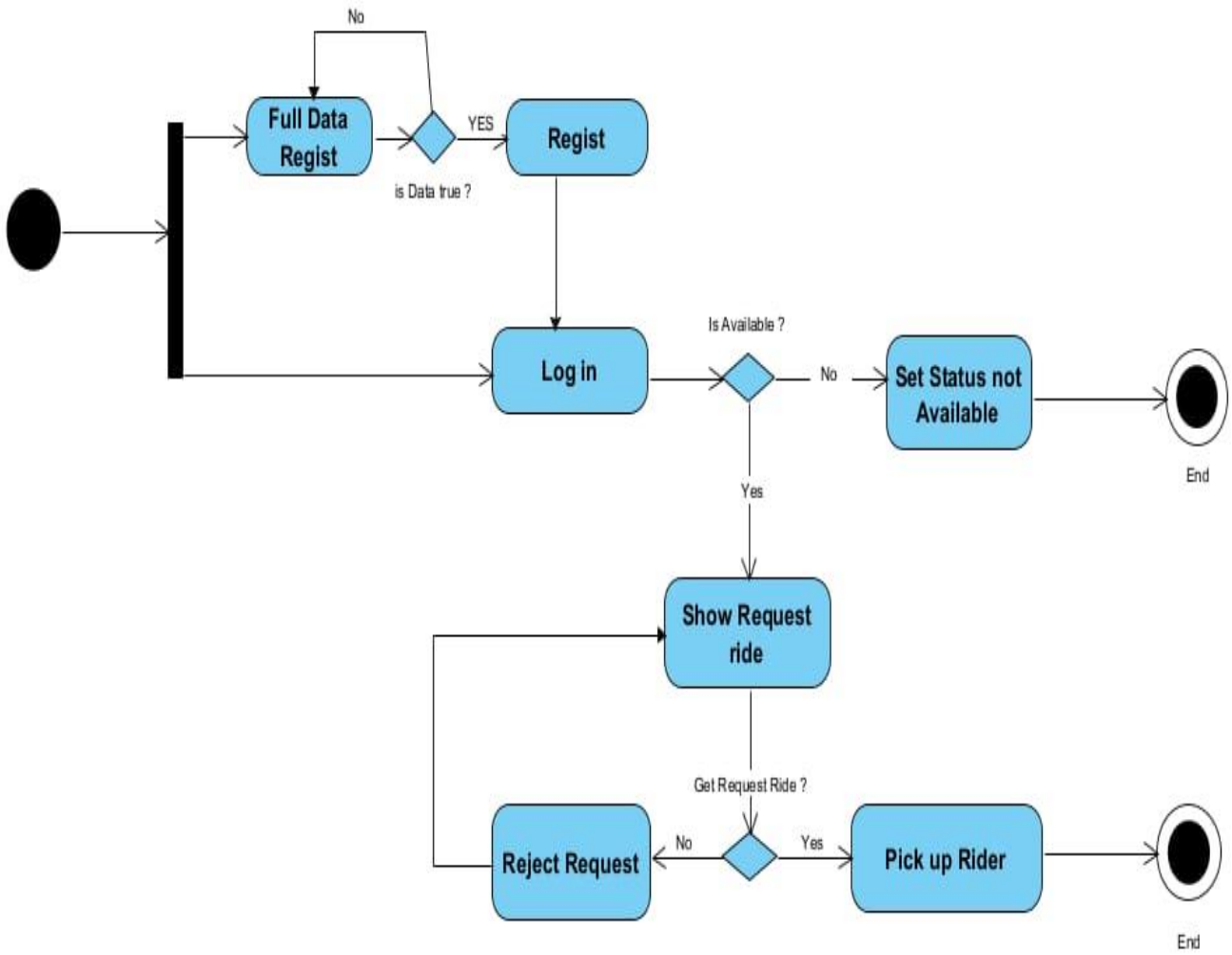
4.9 Activity Diagram

1. Activity diagram(Passenger):



2. Activity diagram(Driver):

act [Driver]

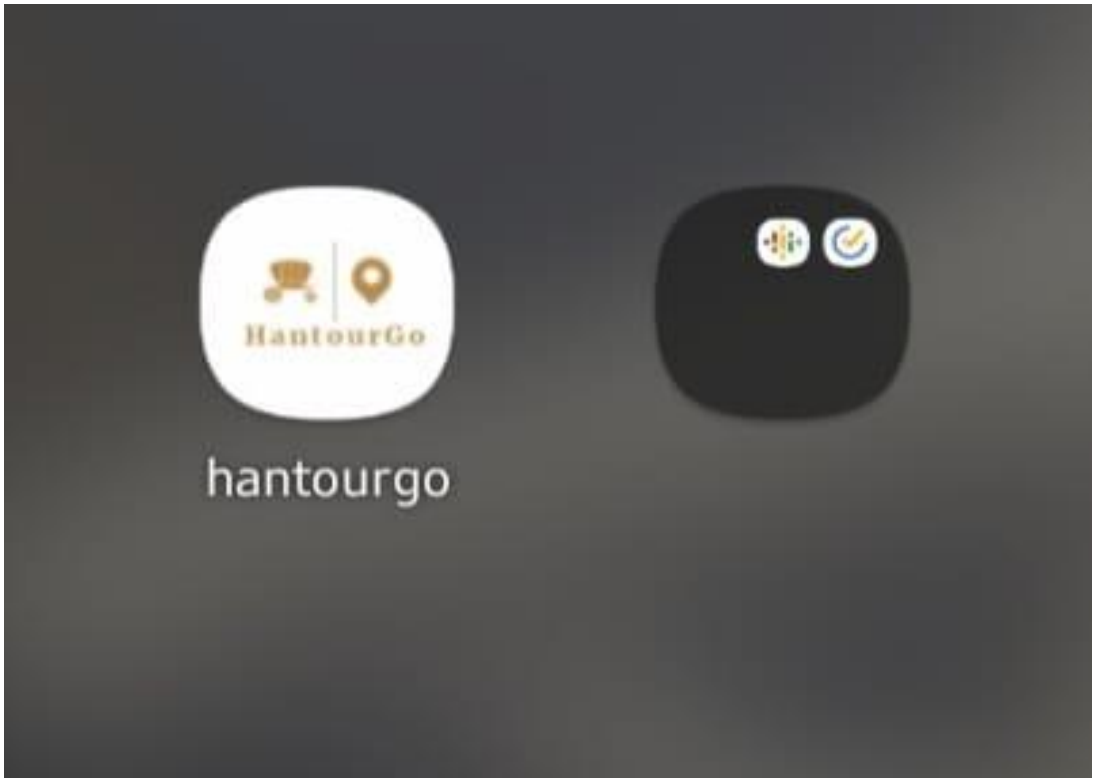


Chapter 5

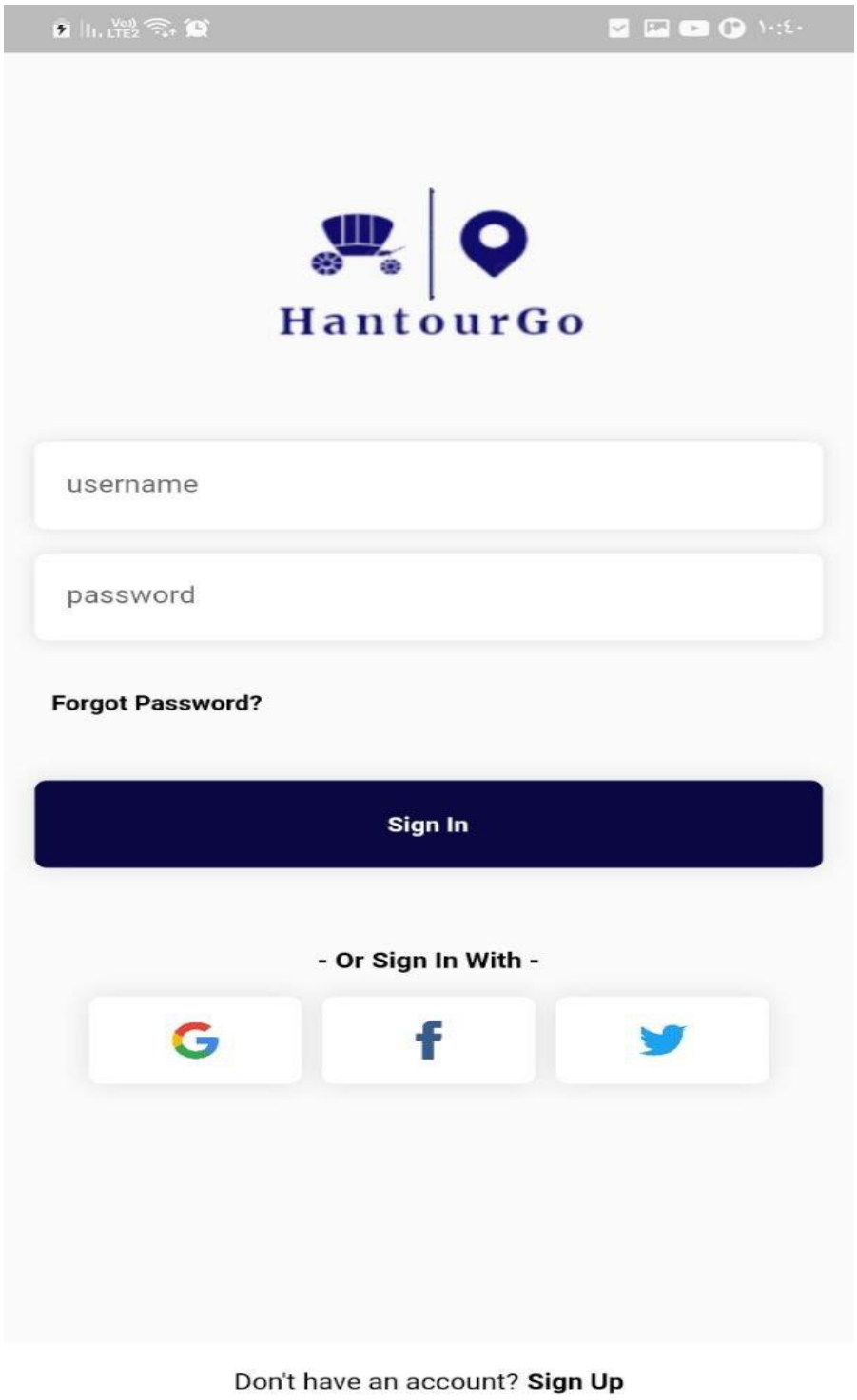
System design and Implementation

5.1. **User Interface**

HantourGo’s user guide on how to install, run, and use the application. The user manual is designed to be as easy and simple and possible to enable all types of users to easily access the application. After downloading and installing the application, the application will show in your app drawer as shown in Figure

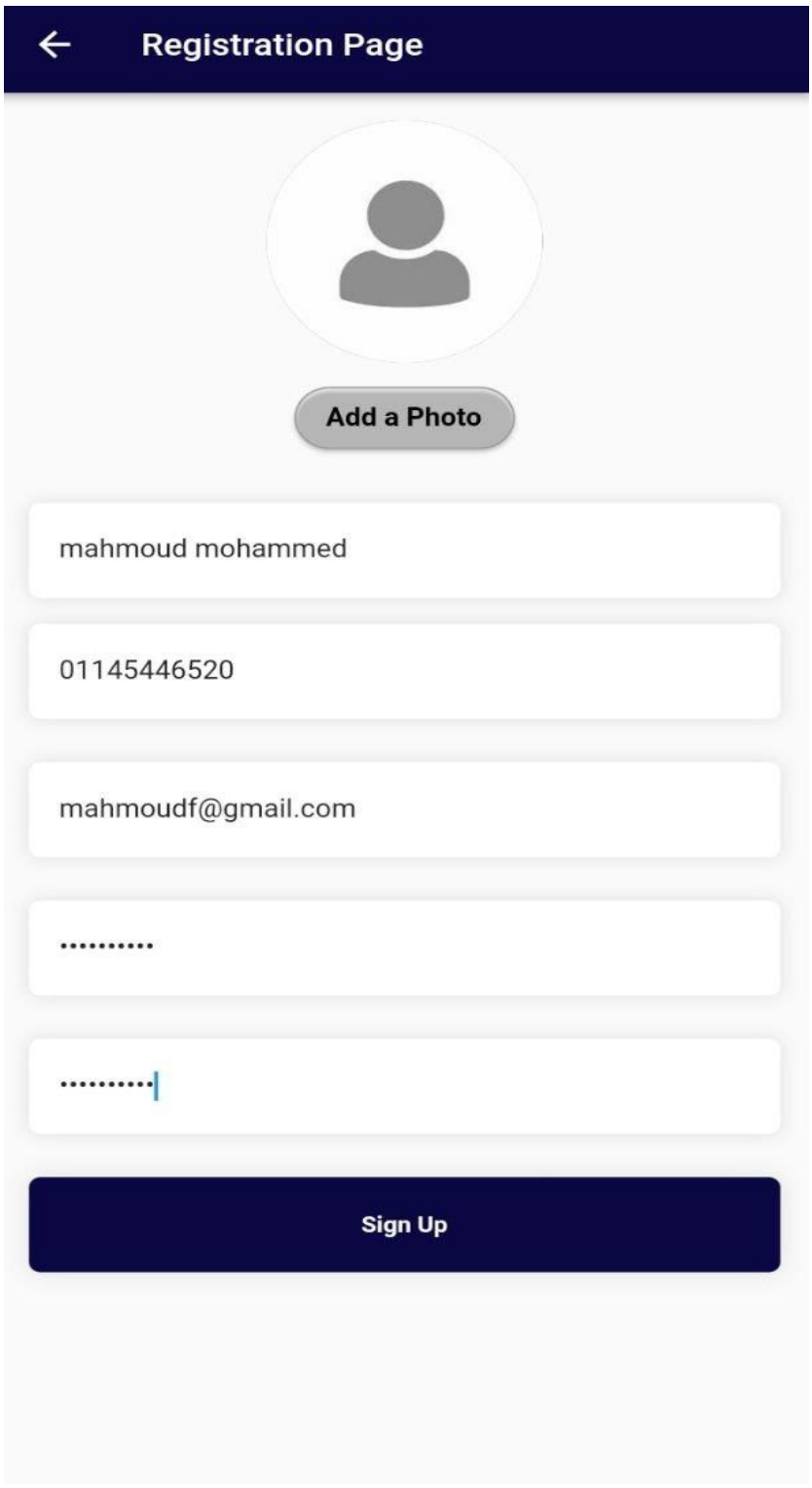


Opening the app takes you to the sign in page from it you can enter to the application and sign in to use the services, if you don’t have an account you can sign up with Google account or Facebook or twitter

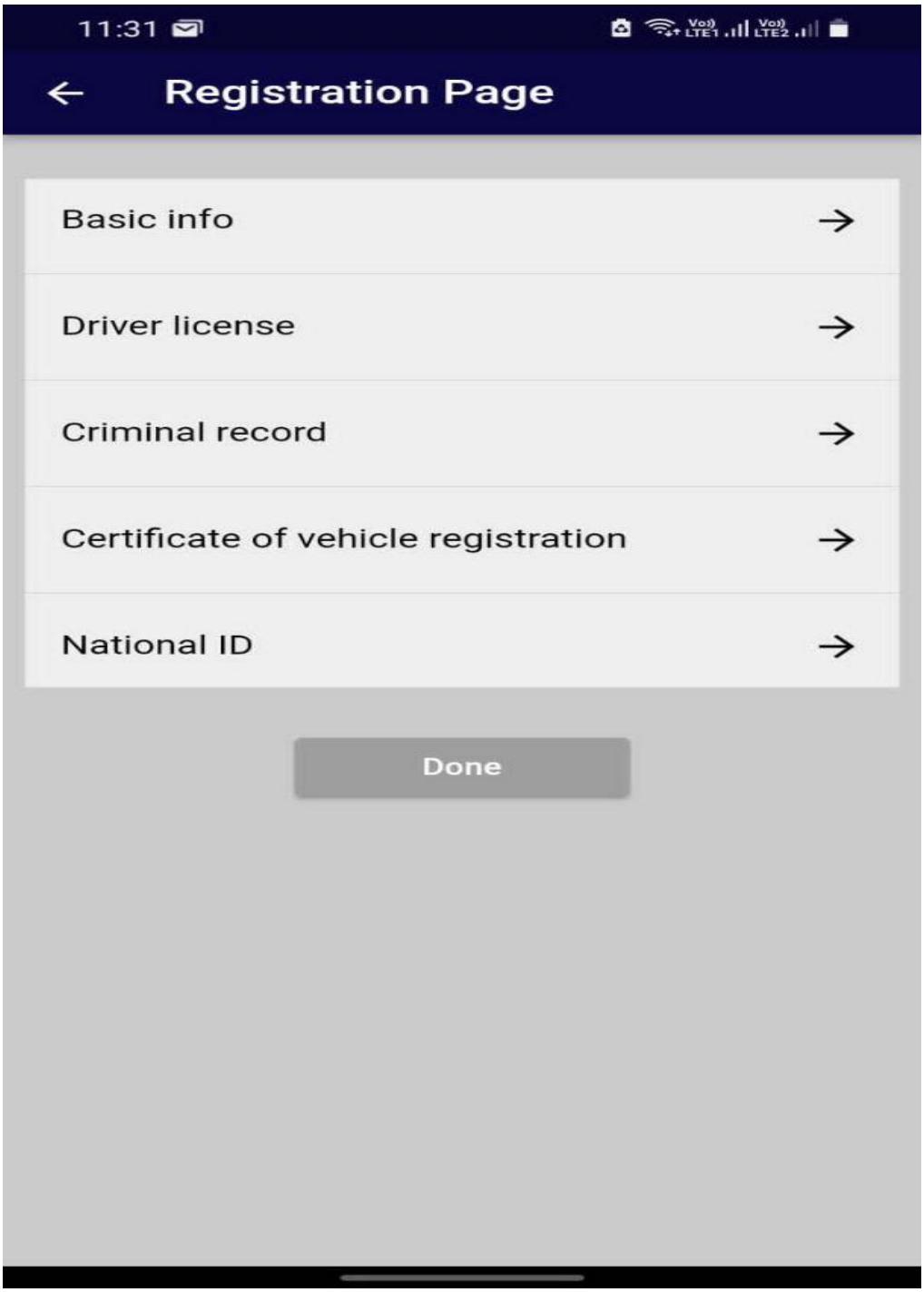


If you don’t have an account you can sign up with Google account or Facebook or twitter, you should choice the type of user Driver or Passenger

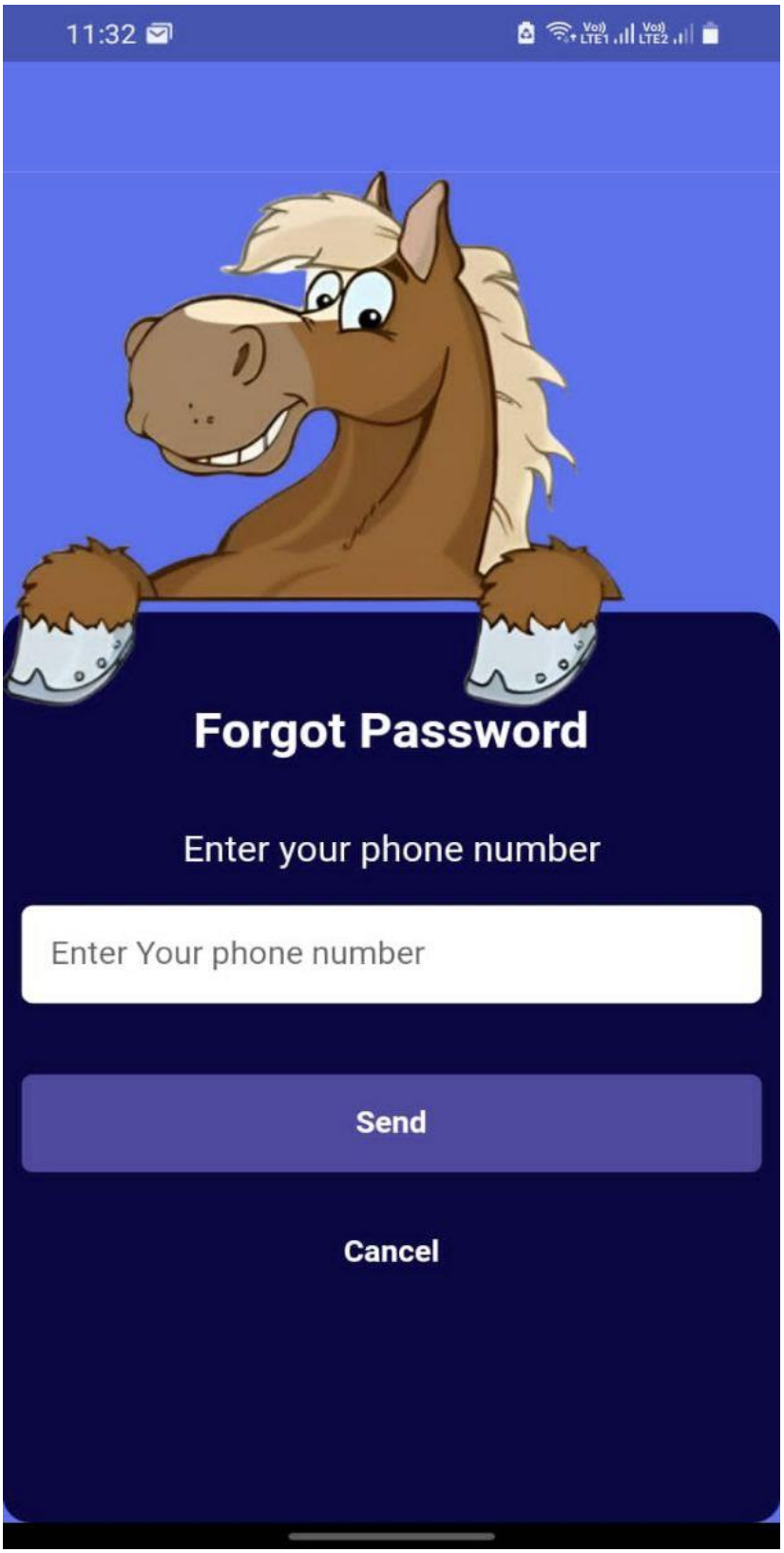
When choice a passenger user this screen will appear to input your data for passenger, after inserting the data you will click to sign up button to save data and go to sign in page to log in to the application as a passenger



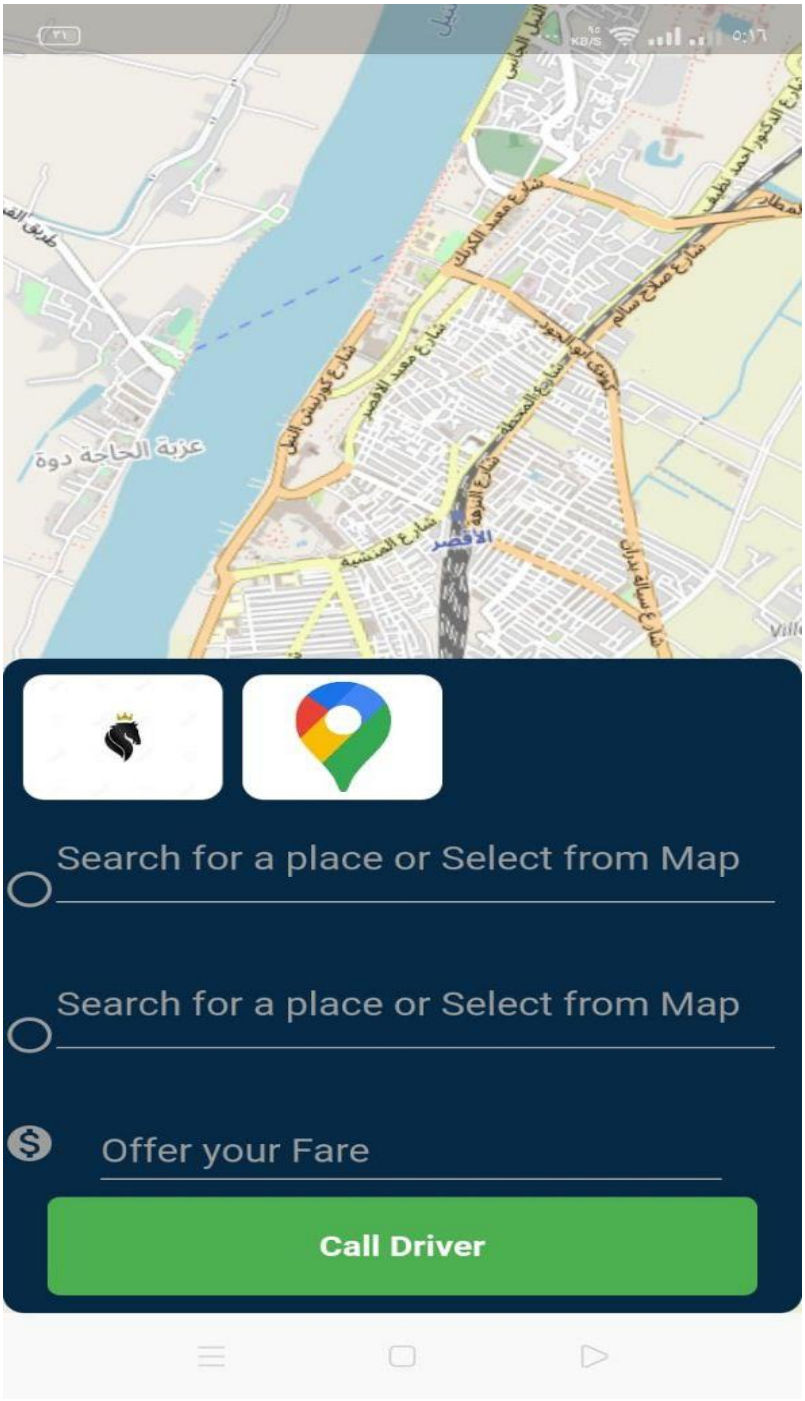
When choice a Driver user this screen will appear to input your data for Driver, after inserting the data you will click to sign up button to save data and go to sign in page to log in to the application as a Driver



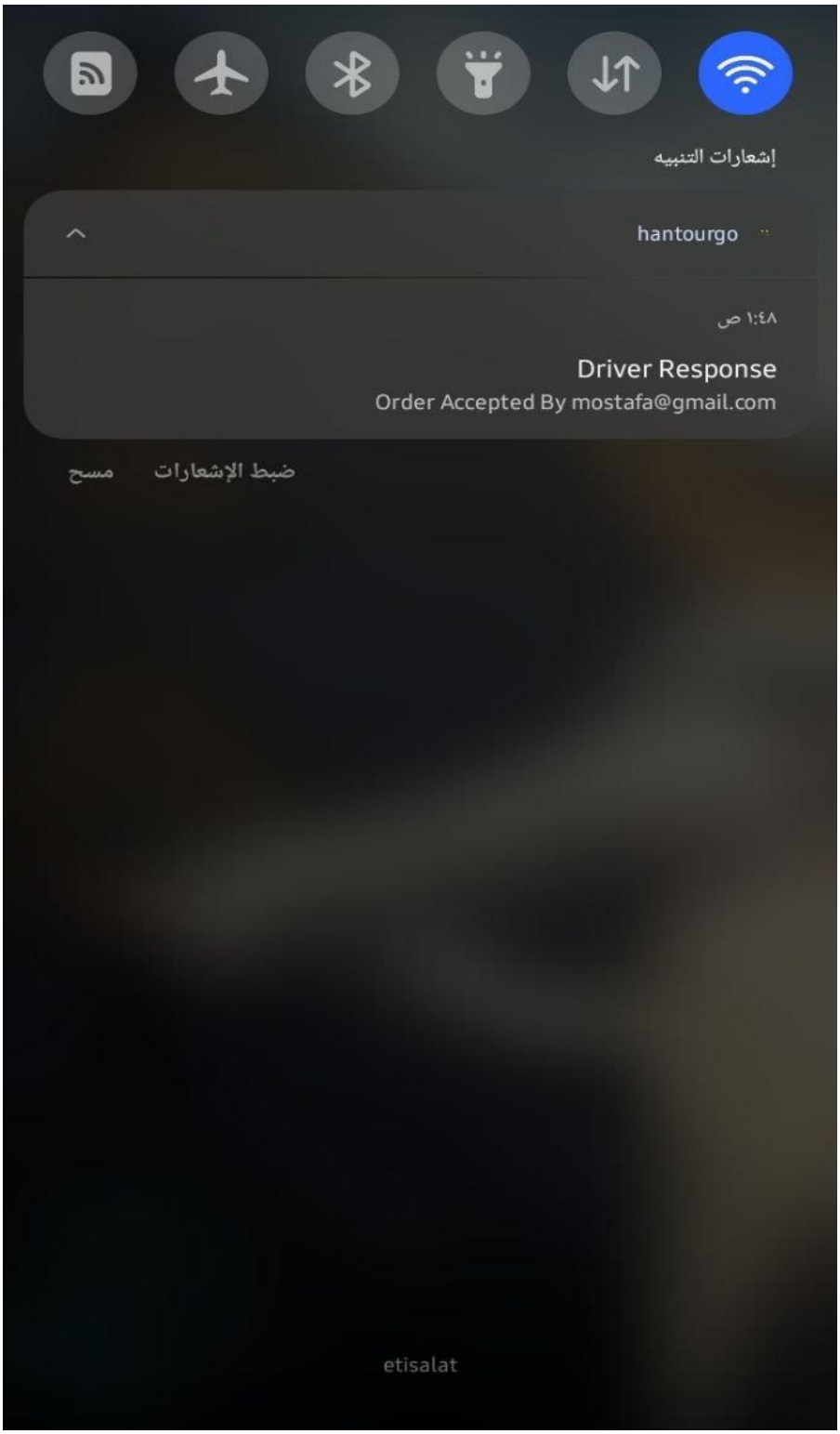
If you have forget the password of your account you can press the word “forgot password” to take you to this screen then you will insert the phone number or email to reset the password of your account



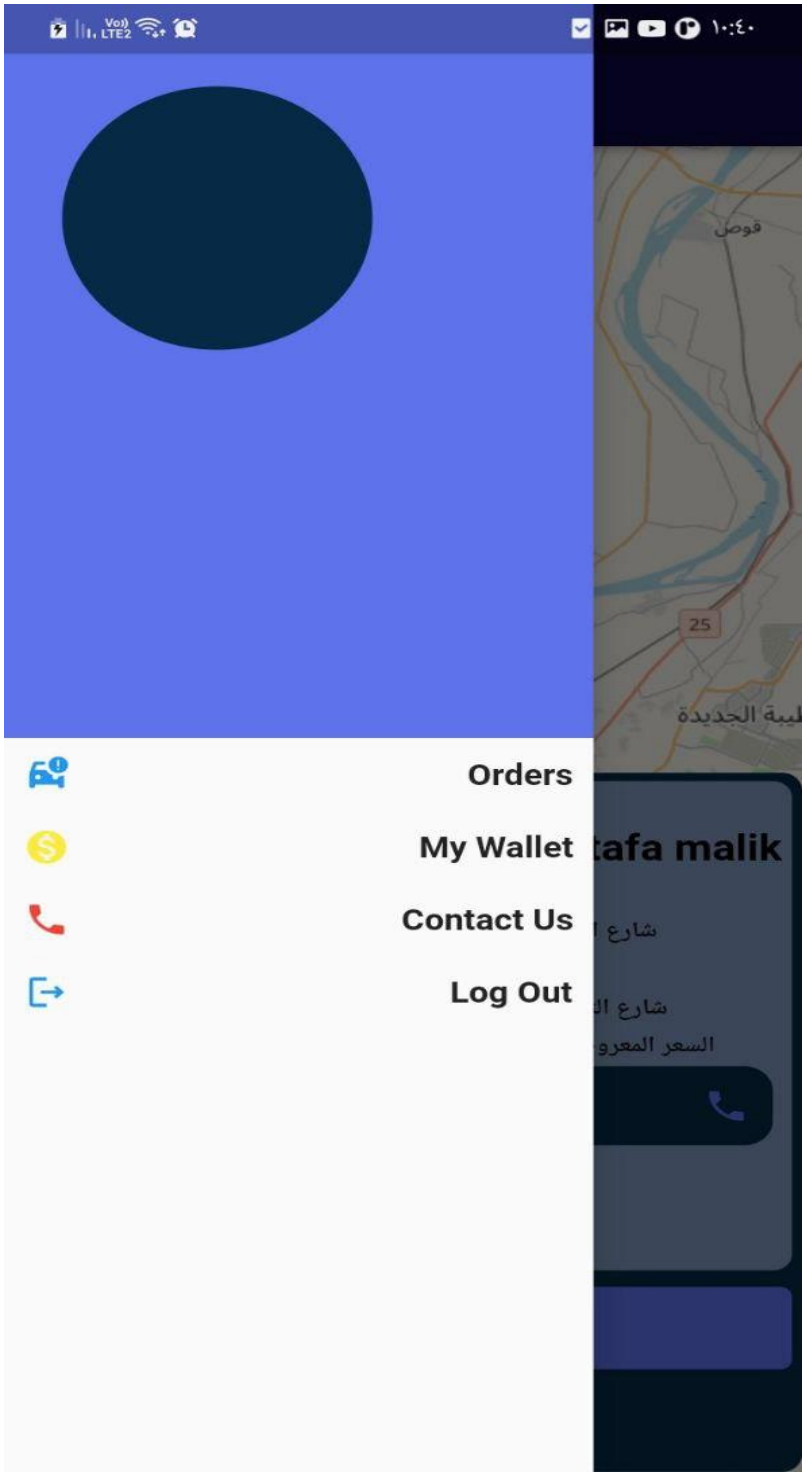
After success log in as a passenger, screen to choice your location and location that you want to go to, after choice the distention of the trip we can choice the price of the trip and click the button of “Call Driver” then it will show in the dashboard of the Drivers



After the passenger call driver, passenger will get a notification of the driver that accept the order of passenger, it will notification to the driver mobile



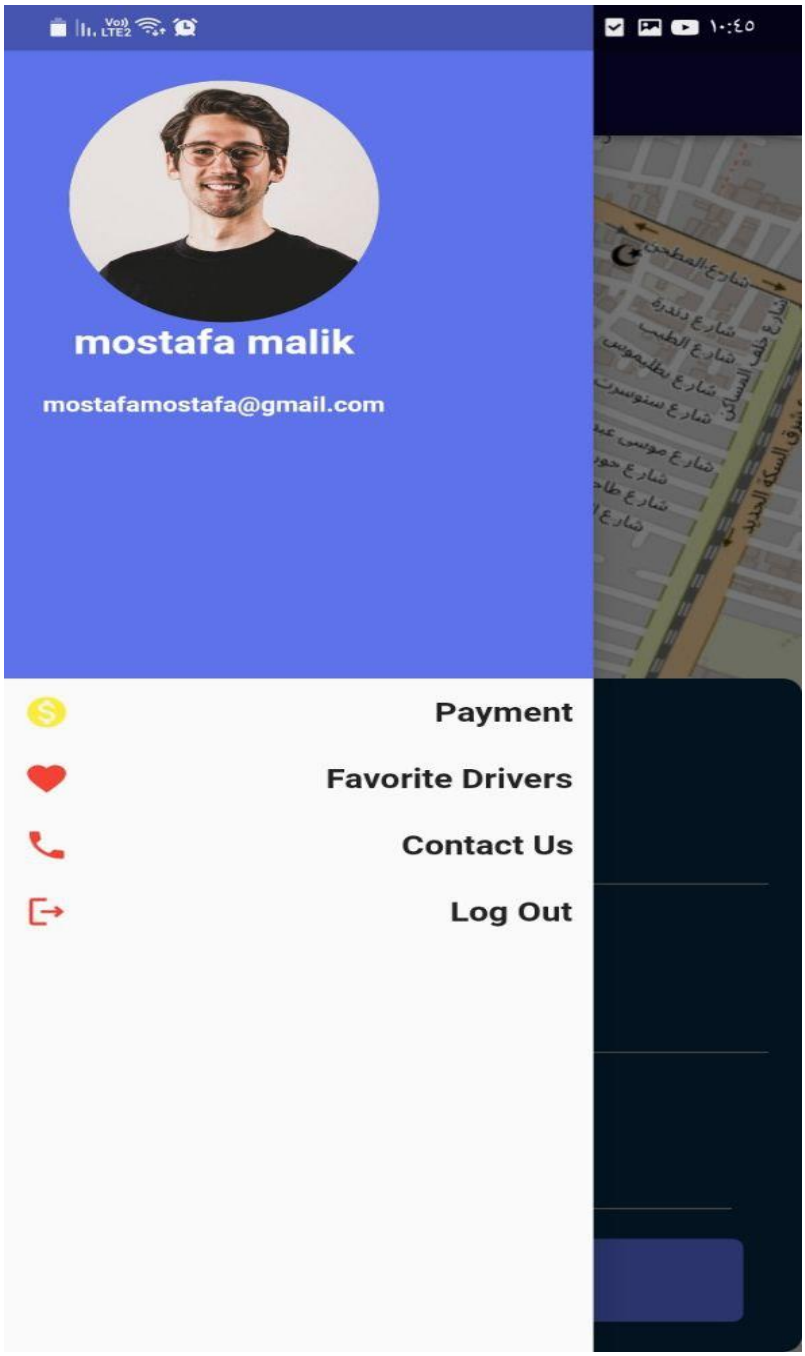
This is the dashboard of driver shows the screen orders, my wallet, contact us and log out



When driver log in this screen will show for first time, this will show the map and destination

When driver click the “orders” button in the dashboard, this will show the orders of passengers that request a trip, after the driver press on the request to accept the order
this will send a notification to passenger to told him that his request has been accepted by a driver and can communicate with him

In button log out in both dashboard in passenger or driver will log out and show the log in screen



5.2. Application and Language Used

1- Android studio
2- Visual Studio
3- Google cloud
4- Figma
5- Flutter
6- Dart
7- Firebase

5.3. What we are striving

- ✓ Making movement between tourist places easier and in an enjoyable way by using Hantour as a means of transportation
- ✓ Reviving the popular and cultural heritage by encouraging horse-carriage
- ✓ Assisting in improving the income of the owners of the vehicles and improving the living
- ✓ Provide an exceptional experience for the tourist by providing a distinguished and comfortable service
- ✓ Help improve the arena and encourage national projects

Chapter 6

Conclusion and Future work

6.1. Conclusion

During the past period, we noticed the development of commercial and profitable applications in the field of transportation, but no contribution or thought was made to preserve the cultural heritage, especially since horse carriages are among the important heritage of interest to tourists, which helps to attract tourists and provide a good idea about this heritage and also help in increasing income For owners of horse carriages, which positively affects the national income of the state.

There is no doubt that this idea is not a new idea or does not exist, but it has not been applied to the means of transportation that help increase the tourist attraction.

Therefore, our application aims to solve the problem of overcrowding or the lack of a good and enjoyable means of transportation suitable for the tourist experience, to be an entertaining, enjoyable and practical means, and also it will help increase the income for the owners of horse carts, which helps in providing hard currency and a good income for the owners of horse carts

6.2. Feature work

We strive to improve the tourist experience through continuous improvement of the application and through commenting on the application and what should be added and modified.
We also strive to improve the quality and performance of the application by searching for the best solutions and the best algorithms to improve speed and response through the application.
Make an evaluation of drivers to make the competition more intense and provide excellent service with excellent quality and reasonable prices
Improvement in the design quality of the application with what suits the users

Reference

1- https://stackoverflow.com/questions/48292305/uml-activity-diagram-with-variable-number-of-concurrent-flows

2- https://bottega.devcamp.com/trails/23/campsites/186/guides/project-solution-uber-activity-diagram

3- -to-introduction-engineering-https://www.geeksforgeeks.org/software ref=gcse?/engineering-software

4- -functional-non-vs-https://www.geeksforgeeks.org/functional /requirements

5- /uber-like-app-an-com/build.https://marutitech

6- -uber-like-app-an-make-to-https://www.thedroidsonroids.com/blog/how 2021-in

7- -the-create-to-how-study-case-https://alconost.medium.com/indriver d85da7bc7549-different-with-countries-31-for-app-sharing-ride-perfect

8- https://www.youtube.com/watch?v=qKr8uQuYbic

9- https://www.javatpoint.com/uml-diagrams

10- /https://www.freecodecamp.org

11- https://en.wikipedia.org/wiki/Software_documentation

12- -project-eeringengin-software-some-are-https://www.quora.com/What ideas

13- -software-understanding-reecodecamp.org/news/whyhttps://www.f /engineer-software-a-as-you-to-matter-requirements

14- https://flutter.dev/

15- https://www.figma.com/

16- https://pub.dev/

17- https://firebase.google.com/?gad=1

18- https://cloud.google.com/cloud-console