Private Car Sharing Application (CAR2SHARE)

By

Yara Wael Taha

Sara Ahmed AlAnsary

Sara Fahad AlGhannam

Nawal Khalid Almujel

Supervised By

Dr. Khulood Almani

A Graduation Project Report Submitted to
College of Computer Sciences and Information at PNU
in Partial Fulfillment of the Requirements for the
Degree of
Bachelor of Science
in
Information Systems

CCIS, PNU

Riyadh, KSA

1441 - 1442H

Table of Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

PNU	Princess Nora bint Abdul Rahman University
GCC	Gulf Cooperation Council
KSA	Kingdom of Saudi Arabia
RAD	Rapid Application Development
UK	United Kingdom
US	United States
UAT	User Acceptance Testing

Acknowledgment

First of all, we would like to thank Allah for giving us the power, patience, and knowledge we needed to complete this portion of the project.

Secondly, we would like to express our gratitude towards the faculty of Computer and Information Sciences at Princess Nourah University without whom we wouldn't have learned the valuable knowledge and skills needed to build this project. We would also like to extend our deepest heartfelt thanks to our amazing supervisor Dr. Khuloud AlMaani who has supported us every step of the way.

Lastly, we would like to thank our wonderful families for their continuous support and motivation.

Abstract

This project is about developing an application platform that provides a peer-to-peer car rental service that allows for a more efficient and seamless car renting experience. Car2share provides its users with the opportunity to make an additional income by putting their car up for rental while being fully insured. Additionally, it enables people in need of a car to be able to obtain any kind of vehicle quickly and efficiently.

Keywords	Meaning
Rentee	The person renting out their car
Renter	The person renting a car
Peer-to-peer	Allows "**peers**" (individual computer systems) to connect over the internet to share files.
Testing	Conducted through scripts with the motive of finding errors in software. It deals with tests for the entire application.

Chapter 1: Introduction

Problem Statement

Car rental is a service that enables individuals to get around even though they do not personally own a car. With that being said, car rental companies only have a select number of locations to which you must go to rent a car, making them not easily accessible. With the rapid evolution of technology and the need for quick user satisfaction, the old-fashioned approach for car rental used in Saudi Arabia is not suitable for the modern approach to technology the Kingdom is striving for.

As of 2018, Saudi Arabia implemented a 5% (five percent) tax regulation [1]. This took a financial toll nationwide, especially towards the social class which was living comfortably pre-tax. That social class usually entails owning multiple cars which were considered common. The aftermath of the tax regulation included gas prices spiking, salaries decreasing, and an entire nation becoming financially cautious.

Careem [2] is a car booking service established in the middle east which became available in Saudi Arabia as of 2015, shortly followed by Uber [3]. Both companies provide one-way rides from any location to your desired destination. Their pricing is established based on availability, distance, timing, and vehicle type. When both companies entered the Kingdom they took advantage of a gap in the market, which was a lack of availability for car booking services, so they were able to set prices in a flexible manner that suits their preference. However, both applications simply provide car booking services, which are distinctly different from car rental services.

At the beginning of 2020, everything in regards to the economy changed, not only in Saudi Arabia but globally. It started when COVID-19 reached the Kingdom [4] causing individuals' paychecks to decrease, the country's resources being used to the furthest extent, oil prices decreasing worldwide which led to a large gap in the Kingdom's income. All of these factors combined caused an increase in taxes to be implemented from 5% (five percent) to 15% (fifteen percent) leading to a more financial struggle for individuals. With this increase, multiple car owners resorted to putting their other cars aside and only using one car. Rendering those extra cars useless. This is where Car2Share comes.

This project aims to develop an application service platform for peer-to-peer private car-sharing [5] that allows car owners to convert their personal vehicles into shared cars that can be rented to other drivers on a short-term basis. It would also provide a car rental service in areas and small towns that have no access to normal commercial car rentals. Car2share application service is the personalized alternative to traditional car rental, car finance, and inflexible business car leasing, which fits with the ‘on-demand mentality of the era. The proposed application service does not have the overheads of traditional car rental companies, which in turn provides substantial savings on the renter accompanied by low costs for vehicle owners which means that market expansion is possible. The platform manages all car rentals, insurance, and the end to end payments and billing through an online system.

Proposed Solution

Car2Share proposes making car rental services more efficient, allowing them to be beneficial for both the car owner and the renter, as well as providing a seamless experience of renting a car and receiving daily income from renting out your personal car.

With Car2share being a peer-to-peer platform, the process of picking up the car from the select locations available from traditional car rental companies is eliminated. Both the renter and the rentee are able to communicate and mutually agree on a location that is compatible with both parties.

A portion of Careem and Uber audience will be swayed towards Car2Share and the service it will provide, this is mainly because their prices have skyrocketed the past year making it less attractive and not as affordable as it was before, additionally, it will no longer be the only option for individuals who own a license but no personal car.

Women were granted the ability to obtain their driving licenses in 2018 [6], a large portion of those women were able to get their license instantly, however, they didn't have the financial ability to purchase a car yet. Car2share will be a great alternative solution to enable women to drive without owning a personal car.

With the initial implementation of taxes and the later increase [7], the individuals with multiple cars have the ability to have additional income from renting out their cars that are not

being used as much, helping them deal with the financial strain that has been developed due to the increase in expense of living in the Kingdom.

Project Description

Car2Share is an application that will allow car owners who do not use their cars all the time to rent them out either by the hour or by day. This is enabled through a peer-to-peer platform that provides for individuals who require a car for a longer period than just a one-time car ride. The application will include an interface for both the renter and the rentee, which will accommodate each of their needs. one of the main aspects of the platform is that insurance will be provided to protect all parties.

Project Methodology

An Agile Methodology [8] was chosen for Car2Share, specifically, Rapid Application Development. RAD [9] is a type of software development that does not need a lot of time or resources for planning but instead uses a prototyping method to introduce a product. RAD follows through with five steps that consist of Business Modeling, Data Modeling, Process Modeling, Application Generation, and Testing and Modeling. The previously mentioned steps are the best option when developing an application such as Car2Share as it is more flexible and dynamic.

RAD Model Diagram

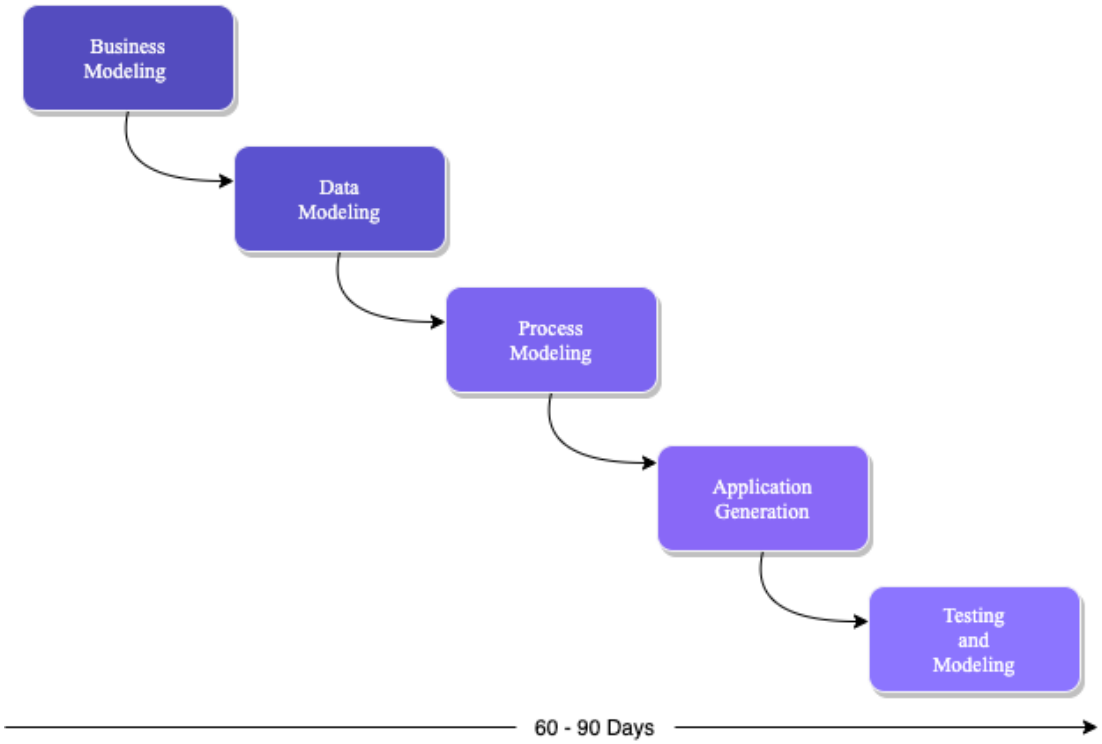


Figure 1.1

Stage 1: Business Modeling

The Business modeling step in RAD modeling entails retrieving information from multiple sources related to the platform through multiple business-related sources. The information is combined into a description that is useful how the information obtained can be used when processed, and what specific information is most valuable to help the interface succeed.

Stage 2: Data Modeling

In this Stage, the information gathered in the Business Modeling phase is reviewed and analyzed. The information is grouped into different categories that can be critical to our project. The quality of each data group is carefully observed and given an accurate description. This phase also helps to determine the RAD system development architecture

from the user feedback, allowing the creation of prototypes. Also, it is repeated many times as the project evolves.

Stage 3: Process Modeling

During the Process Modeling phase, all the information that has been gathered during the Data Modeling steps is converted into the required usable information needed to achieve specific business objectives as per the business model, furthermore, changes and optimization can be done, and the sets of data can be further defined. Any description for adding, removing, or changing the data objects is also conducted during this phase.

Stage 4: Application Generation

The Application Generation step is where the actual system is built, when all the needs, features, and information gathered for the application are put into codes, and where various tools are used to convert processes and data models into actual prototypes that can be used in the next step.

Stage 5: Testing and Turnover

The Testing and Turnover Stage involves testing the prototypes independently after each iteration so that the overall testing time can be reduced.

During this stage it is reassured, there should not be any major and preferably minor problems with our prototype since most of the elements have already been tested earlier.

Project Domain

Car2share will offer its services in Saudi Arabia and the Gulf Cooperation Council (GCC) countries. It will be a business with multiple locations operating in a specific area, in order to expand the project's overall economic activity and to lead to high-level competition among businesses.

Project Limitations

When it comes to limitations there are a few mentionable ones.

- Insurance, legal policies, and registration will differ from region to region, this may cause some limitations when expanding the service to multiple regions

- Cost requirements may be high at first to accomplish all the required tasks prototyping, testing, and implementation

System Aims and Objectives

This system aims to provide peer-to-peer car rental service, car owners will be able to list their cars making them available for renters to rent.

- Make car rental services more accessible, convenient, and flexible to use.
- Allow individuals with multiple cars to have another source of income.
- To ease consumer tasks whenever they need to rent a car.
- Reduce manual work of rental service.
- Make renting out your car more secure.
- Design a user-friendly interface for the application.

Chapter 2: Background Information

2.1 Background information

With the recent tax increase in Saudi Arabia, which led to some financial strain, leading people to seek extra forms of income, Car2Share provides the opportunity to those who are willing to rent out any cars they own. Through the platform, individuals will be able to register themselves and their vehicles into the database. Once registered and approved, they will be covered by the Car2Share insurance plan. At that point they will be able to begin renting out their car, so, their vehicle and information will be displayed on the interface for local users. As for the renters, they will be asked to register ith their personal information, especially their driver's license, and then be able to view the options available based on their needs and preferences.

Finding the right car for you:

When trying to find the right car to rent, there is a list of needs and preferences users have. In order to serve those needs, the following questions may be asked:

1. What size of car do you prefer?
2. What car brands do you prefer?
3. What features would you like to be available for you?
4. What is your car renting budget?
5. Would you prefer to rent by the hour or by the day?
6. What car model would you prefer?

Tools used for the project:

Mainly, as a source of communication between renters and rentees, there is a chat feature on the application which enables the exchange of information between the two parties. This feature enables the monitoring of communication to maintain the safety and security of everyone involved.

2.2 Related Work Survey

Through research, we were able to discover multiple similar platforms that are available across the world. Of which include, Turo, Drive Drive Car, and Nobobil.no

1. Turo

Figure 2.1 Related Work (Turo)

Turo [10] is an app available on both the Apple store and Google Play store. Turo is a car-sharing marketplace where guests can book any car they want, wherever they want it, from a vibrant community of local hosts across the US, Canada, and the UK. Guests choose from a unique selection of nearby cars, while hosts earn extra money to offset the costs of car ownership. They offer a wide range of vehicles like trucks, luxury cars, vans, etc. Turo is only available in cities in the UK, the US, and Canada. It seems to be a 35% more affordable option compared to traditional car renting options.

2. Drive Drive Car

Figure 3- Related Work (Drive Drive Car)

Drive Drive Car [11] is an online platform that enables both individuals and companies through its peer-to-peer car-sharing services. The platform allows individuals and companies to list their cars for rent whenever they are not using them for their personal use. Through this platform, they intend to make the process of renting or listing a car easier for clients, increase the availability of cars, a differentiation of cars, and other additional services.

3. Nobobil.no

Figure 4 - Related Work (Nabobil.no)

Nabobil.no [12] is a Norwegian peer-to-peer marketplace for underutilized personal vehicles. It combines people that need a car with people that own a car. You are also allowed to choose whether you prefer to rent the car by the hour, day, week, or month.
One of Nabobilo's advantages is it is completely free to join, as you get access to your own virtual garage with cars of all genres.

Nabobil is the perfect option of how a convenient service can make shared mobility not just in big cities, it is everywhere. With their peer-to-peer model, Nabobil makes mobility widely available and permits registered users to put existing cars to better use. With good quality and affordable car to rent for a weekend break or vacation.

2.3 Proposed & Similar System Comparison

Table 2.1 - Proposed & Similar System Comparison

Features	Car2share	Turo	Drive Drive car	Nabobil.no
Geographical Location (MENA)	✓	x	x	x
Insurance Providence	✓	✓	✓	✓
In-App Communication	✓	✓	x	x
Filtering Options	✓	✓	x	x
Rate and Review Renters/Vehicles	✓	✓	✓	✓
Nearest Car and Distance	✓	✓	✓	✓
Security Alerts for Regional Limits	✓	x	x	x
Quick Payments (Apple Pay, STC Pay)	✓	✓	✓	✓

Chapter 3: System Analysis

3.1 System Analysis

System analysis is a method where facts are gathered and interpreted, problems are defined, and a system is decomposed into its components. This is done in order to study the system and its elements to be able to identify the system components. The system analysis will enable us to ensure that the system components are all working efficiently and can accomplish their purpose. The following is what was established for Car2Share.

The Car2Share system aspires to allow peer-to-peer car rental services which somewhat eliminates the traditional car rental systems.

There will be two interfaces made available. One which will be created for the renters through which they will register their information, create an account, and then be able to use the application to rent a car. As for the rentee, the interface available to them will enable them to register their information and their vehicle information into the system, and once their information is validated and they are considered eligible they will be able to put their vehicles up for rent.

3.2 Requirements Determination and Collection

Collecting requirements can be done in several ways. In this project, information is gathered using s survey. This method is chosen as it is a more accurate, fast way to understand our potential customer's needs and expectations.

3.2.1 Surveys

A survey is a research method in which people are asked a set of questions to gather information, opinions, and insights on a particular topic[KA1]. Surveys can be carried out in many ways depending on the methods being used and the targets to be accomplished.

In this study, the survey was created on Google forms, with six total questions for potential car renters and rentees. The survey aims to explore peoples' willingness to rent their own cars to strangers if their vehicles were fully insured by Car2share while also measuring peoples'

needs and acceptance for peer-to-peer car rentals. This survey was circulated using social media outlets such as Whatsapp, Twitter, and Snapchat with a total of 566 responses.

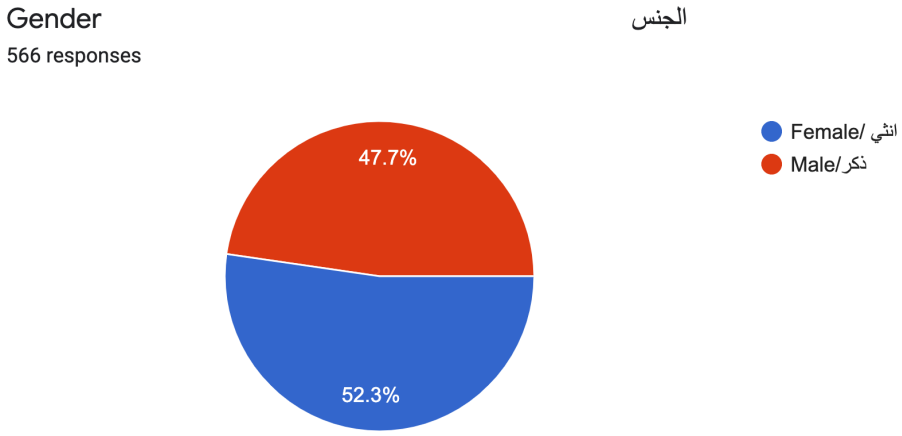


Figure 5 - Survey Question 1 Statistics

This chart (Figure 5), suggests that most of our participants and potential users are females, which further supports our initial statement that since the lift of the women driving ban, they have created a larger need for car services.

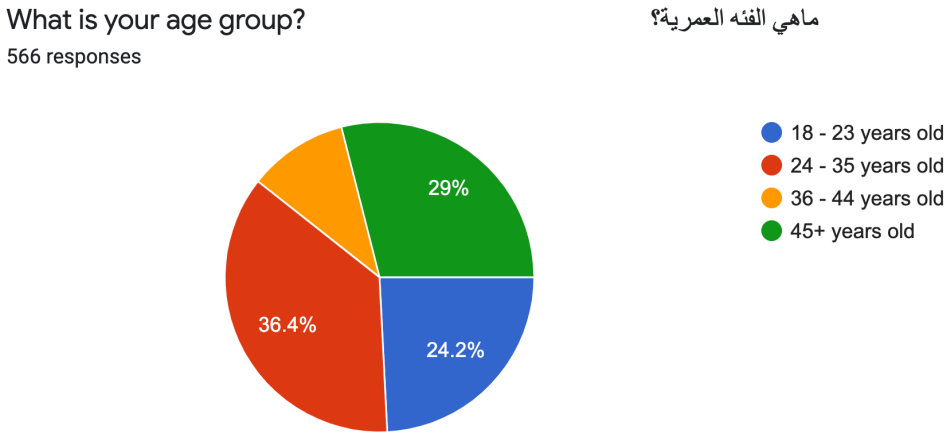


Figure 6 - Survey Question 2 Statistics

In figure 6, the question asked was the age group of the participants, which resulted in the majority being the age group between 24-35-year-olds which amounts to 36.4% of all users. This suggests that most of our potential users are within that age group which provides important insights.

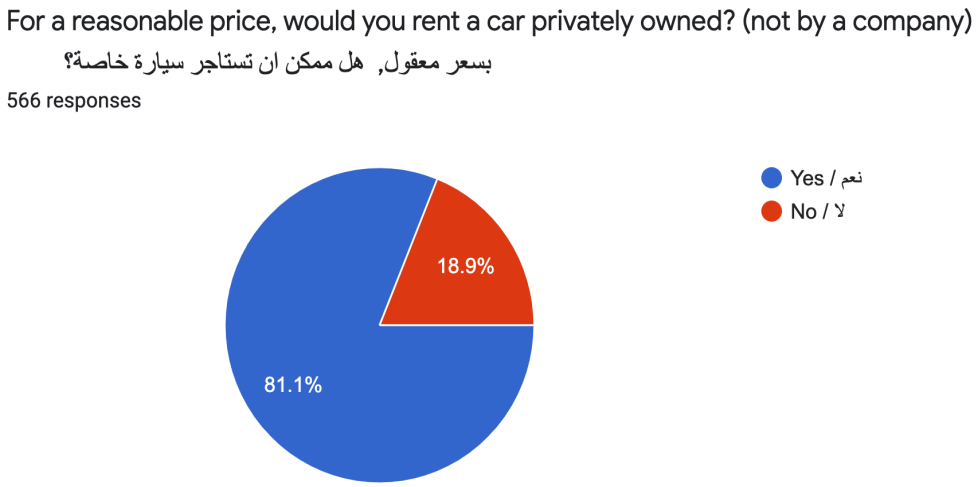


Figure 7 - Survey Question 3 Statistics

Given the response shown in this chart (Figure 7), it indicates that 81.1% of people are willing to rent out their personal cars for additional income. These statistics demonstrate that given the current circumstances, with the rise of taxes and financial strains, people are more inclined to rent out their cars.

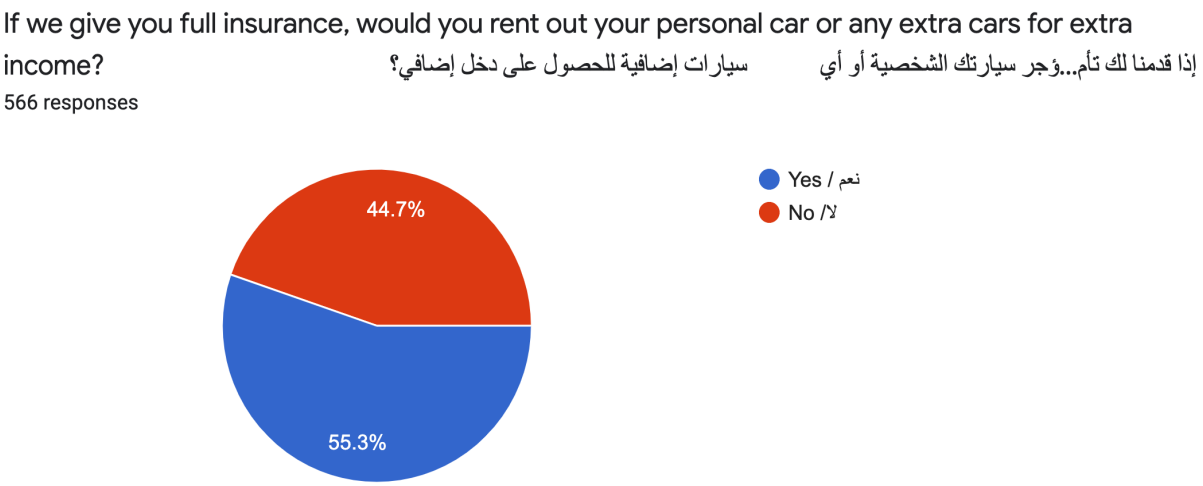


Figure 8 - Survey Question 4 Statistics

We asked the target audience in the chart above (Figure 8) if they would be willing to rent out their personal car, this almost came down to; a little over half of the population saying yes, and the other rest saying no.

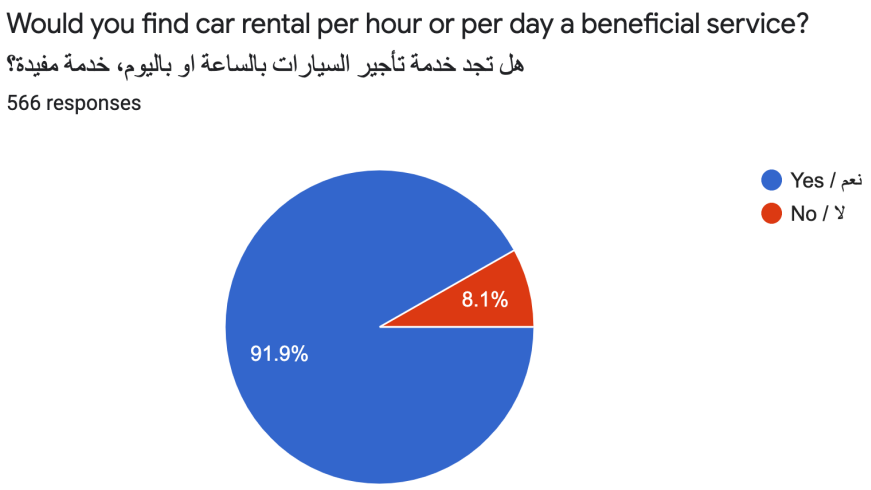
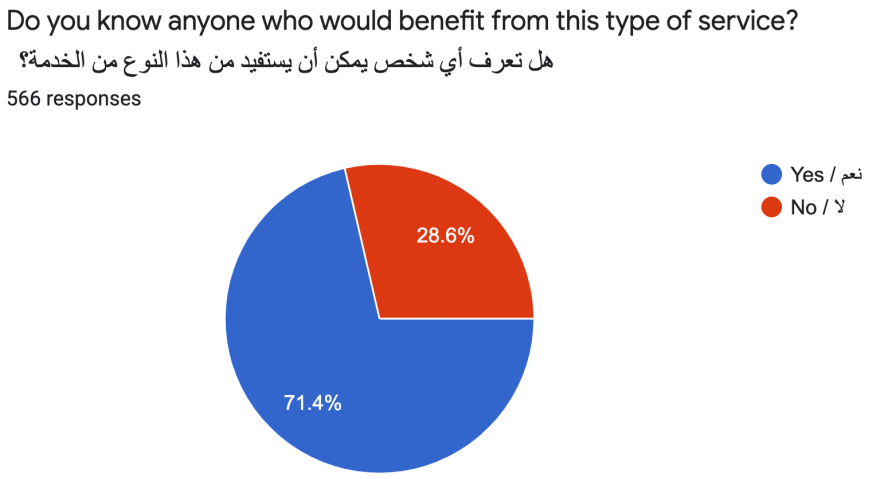


Figure 9- Survey Question 5 Statistics

One of the questions asked, shown in figure 9, was if the survey participants found car rental by the hour/by day a beneficial service. 91.9% (520 votes) chose yes, while 8.1% (46 votes) chose no. These numbers display a positive vision for how the audience views our service.

Figure 10 - Survey Question 6 Statistics



According to this chart (Figure 10), We asked our participants if they think anyone would benefit from this type of service, And we got 260 Responses saying yes and 119 saying no.

3.2.2 Functional Requirements

Functional requirements [13] define what the system is supposed to accomplish, what it should present, as well as the behavior of the system.

Table 3.1 - Functional Requirements Table

Rentee Registration / Sign Up	The system will allow the rentee to register on the system, requiring all basic information including; - full name - ID number - email address - phone number - password - car type - car model - car registration
Renter Registration / Sign Up	The system will allow car renters to register in the system before being able to rent a car, they'll need to input; - full name - ID number - driver license - email address - phone number - password
Login	If Rentee or Renter is already registered they can directly login to their accounts
Browse and Select Car	The system will allow renters to browse

	cars they'd be interested in and will have the option to filter selection.
Automatic Update	Automatic updates to the displayed cars are updated in the database once the reservation has been made by a renter
Rentee Notified	After a customer/renter has reserved a car, the owner of the car (rentee) is notified of the rental reservation
Feedback	Renter: Renters will be given the option to rate the car they rented as well as any additional comments Rentee: Rentee will be given the option to rate the renter's use of the car as well as any additional comments

3.2.3 Non-Functional Requirements

Non-Functional Requirements [14] defines the system's attributes that should be considered for implementing our project.

Table 3.2 - Non-Functional Requirements Table

1- Usability:	- Users should be able to understand the flow of the App easily without any guidelines. - The Graphical user interface(GUI) will be user friendly

2-Security:	All of the Car2share users(both the renter and the rentee)have a privilege and can access the system using the email/username/number as well as the driver's license will be required for becoming a Car2share user.
3- Responsiveness:	The application has to be responsive to user input, if any external interrupt occurs it has to return to the same state. For example: if the user was using the app and got interrupted by a Phone call/Facetime etc, The app should return to the same page and save the state.
4-Performance:	The application's performance is measured by its responsive time. For example: when an Application is made to start up it shouldn't take more than 3 seconds to load the initial screen.

3.3 Software/Hardware Requirements:

Hardware Requirements:

The hardware requirements are the physical computing resources that we need to consider while implementing our project.

Table 3.3 - Hardware requirements Table

Item	Purpose
Personal Laptop	Used for researching purposes, programming, creating an interface
External hard disk	For more storage space to keep the data
Smart Phone	It gives us the ability to test and use the application, and communicate between team members.

Software Requirements:

Software Requirements represents the programs we will be using for Car2Share that perform different tasks on a computer system.

Table 3.4 - Software Requirements Table

Item	Example
Internet Browser	Safari and google chrome to search about any related work.
Microsoft Office Programs	Word and PowerPoint for documenting and presenting purposes.
Operating Systems	Since The operating system plays a pivotal role in all development. Our application supports Android and iOS because they are the most popular.
Diagram's Applications	Draw.io, DrawExpress, and for the interface: proto.io

3.4 Requirements Analysis

3.3.1 Use Case Diagram

Figure 11 - Use Case Diagram

3.3.2 Class Diagram

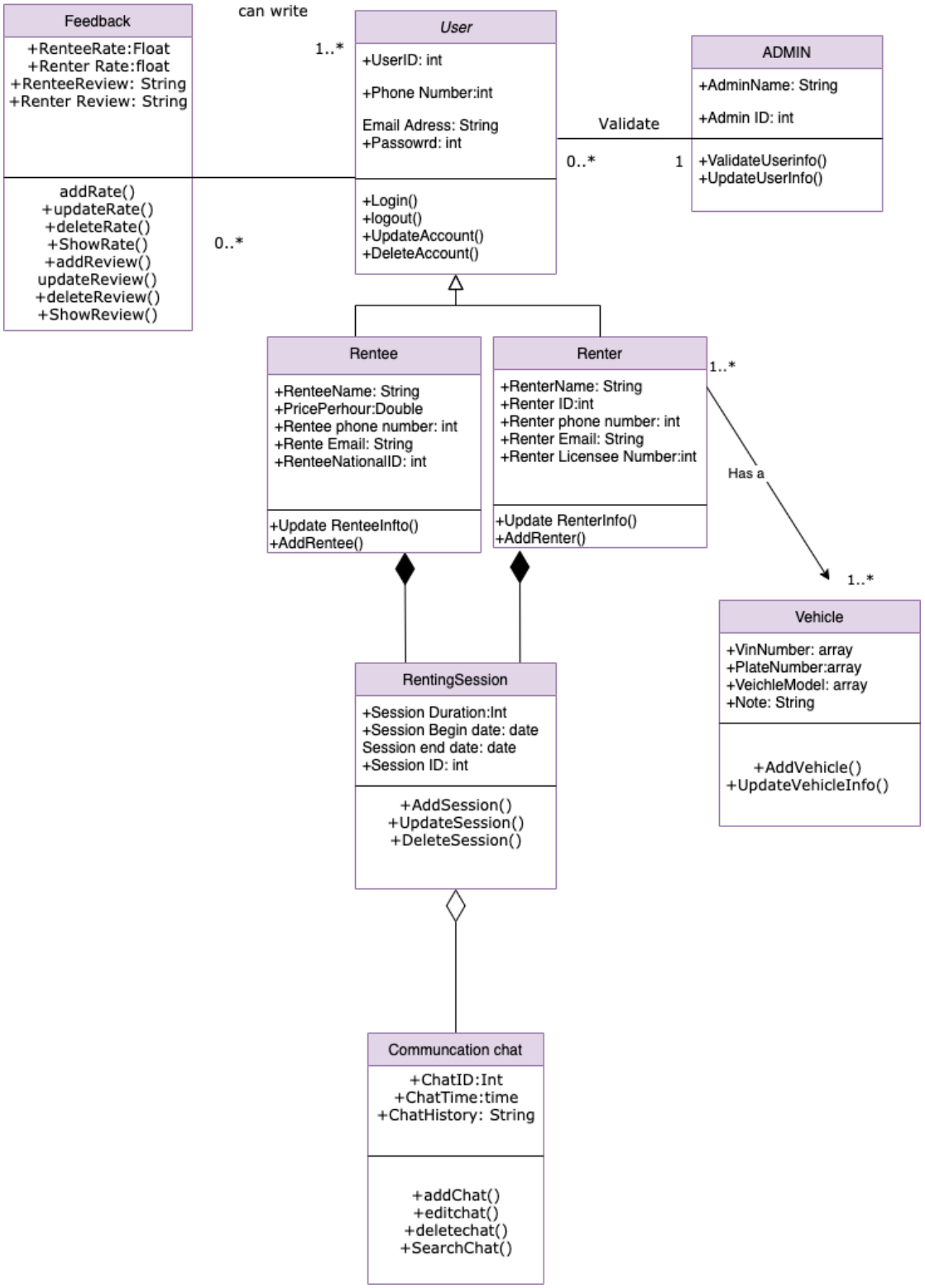


Figure 12 - Class Diagram

3.3.3 Sequence Diagram

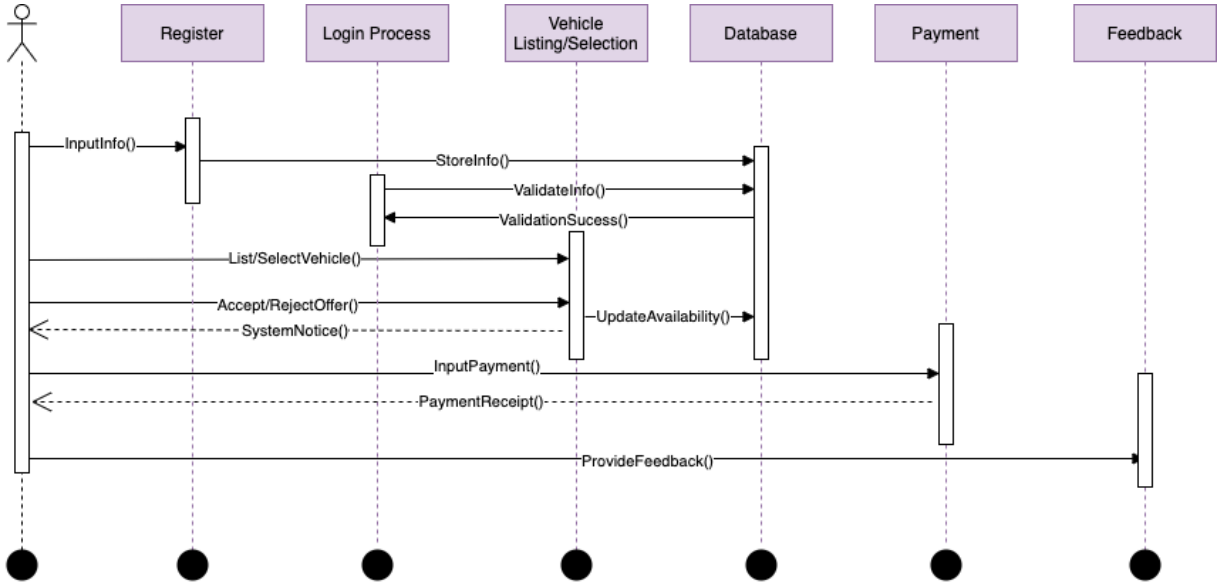


Figure 13 - Sequence Diagram

Chapter 4: System Design

4.1 System Architecture

In this section, the overall structure of the application will be demonstrated, as well as its elements, components, and how they operate to provide service for the users.

For our application, we chose the three-layered system [15] architecture which divides the system into the presentation layer, business layer, and data access layer. Each of the layers has a special role within the application. The presentation layer which is at the top of the architecture, specializes in handling the user interface, while the business layer handles performing specific request-related processes. The third and last layer in this architecture is the data access layer, which enables simplified access to data stored in permanent storage such as databases.

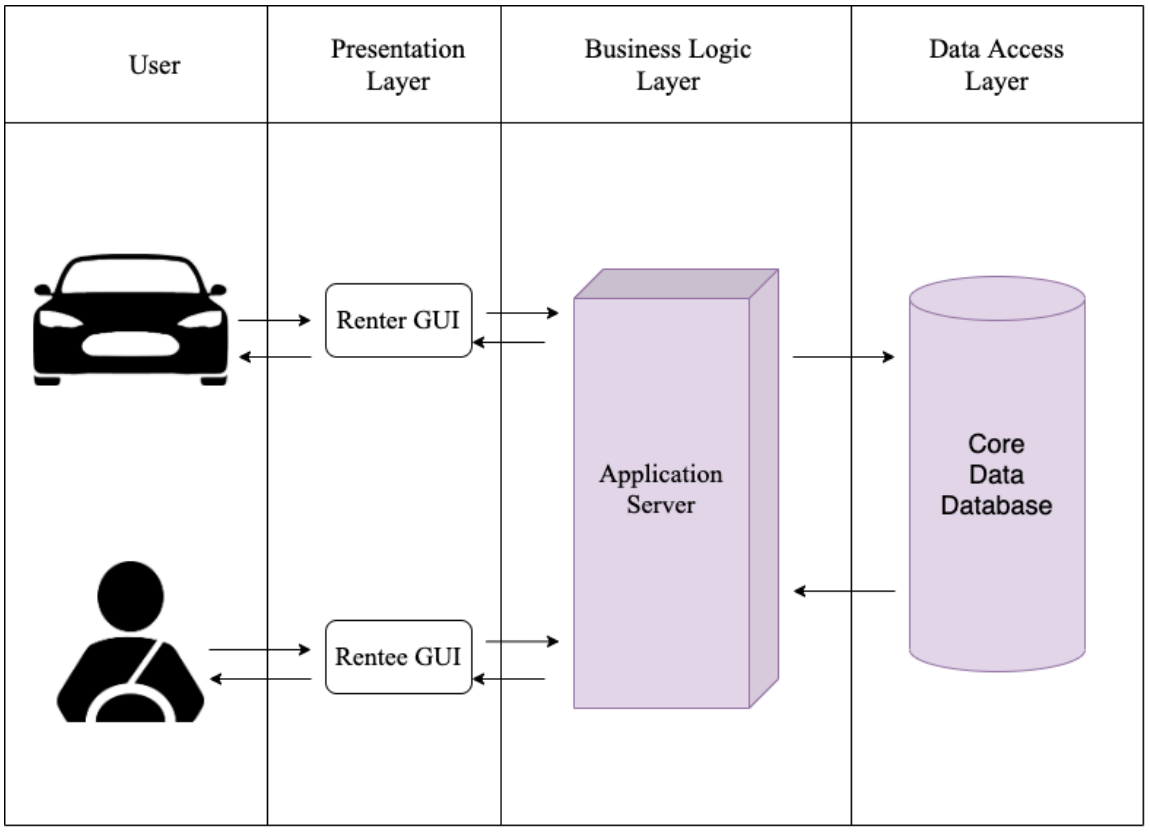


Figure 14 - System Architecture

.

4.1.1 Description of components

1. **Users**

The first and most important aspect of our system is the user, who could either be a renter or rentee. The interaction between these users and our application occurs in the presentation layer.

2. **Presentation Layer**

The primary function of this layer is the formatting and delivery of information to the business layer for further processing and display. The presentation layer provides services through using GUI to present content to the end-user, the GUI may be accessed through the client's mobile phones.

3. **Business Logic Layer**

The business logic layer is the middle stage. IT contains the functional business logic that drove the core capabilities of an application. This limits the actions of the app to meet the needs of the users. Java, Swift, C#, Python, C++, etc

User operations in our application are processed and include registration, login, and logout. Rentees are able to perform operations, such as, searching for available vehicles to rent, requesting communication with a renter, and reviewing the experience afterward, which are all also processed in this layer.

As for the rentee, they're able to view requests and either accept or deny them, speak with rentees, and review the experience afterward.

4. **Data Access Layer**

Contact with external structures such as databases and networks are embedded in the data access layer. This breaks down to data being collected from either the Web or being cached in Core Data or User Defaults. For our application, we will be using Core Data as the basis.

4.2 System Design Tools

Several tools can be used to help facilitate system design such as flowcharts, UML as well as storyboards. To represent car2share design tools most effectively a flowchart is the optimal decision. A flowchart is a visual representation of the separate steps of a process in sequential order, it can also be used to describe various processes such as administrative or the user's process.

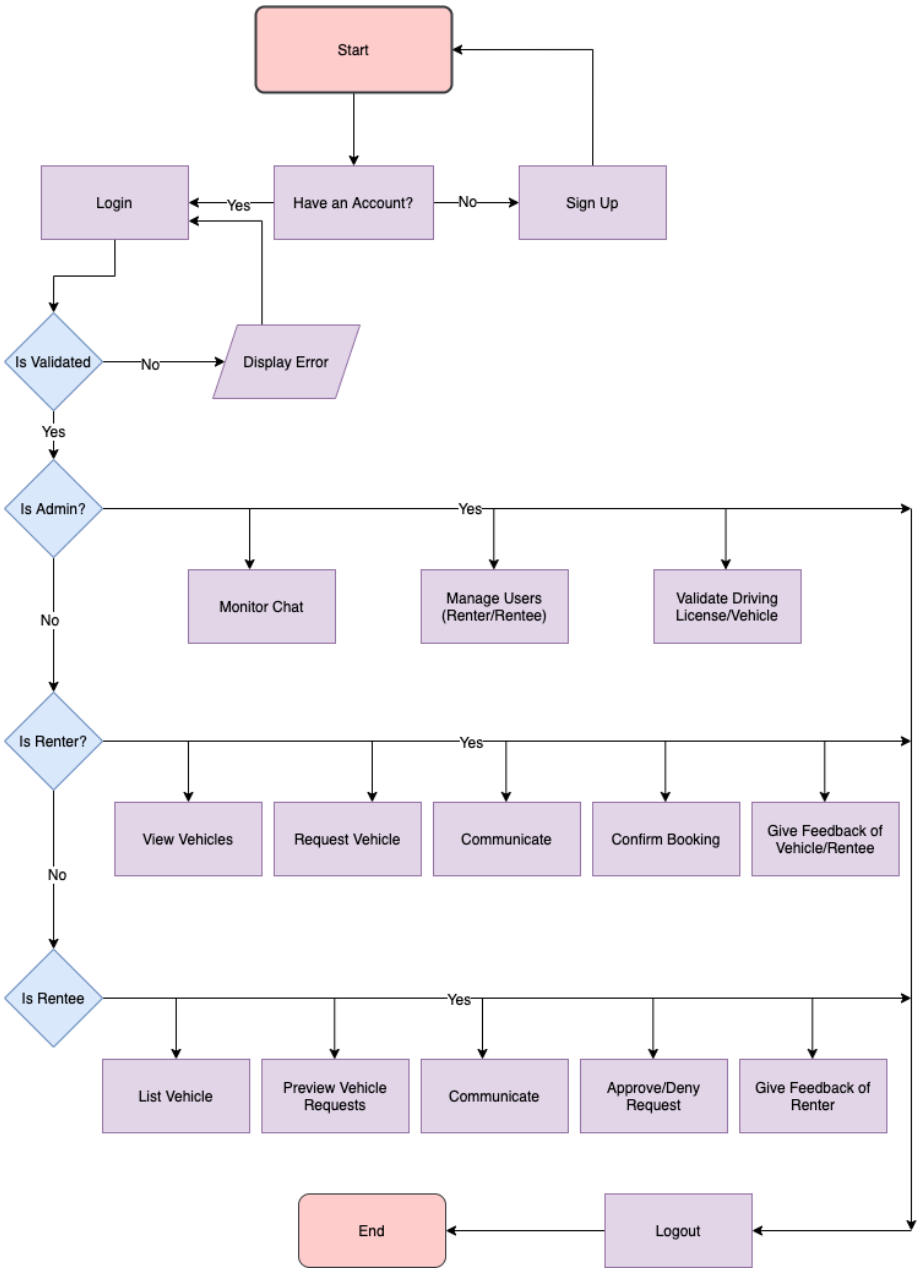


Figure 15 - System Design Tools

4.3 ERD

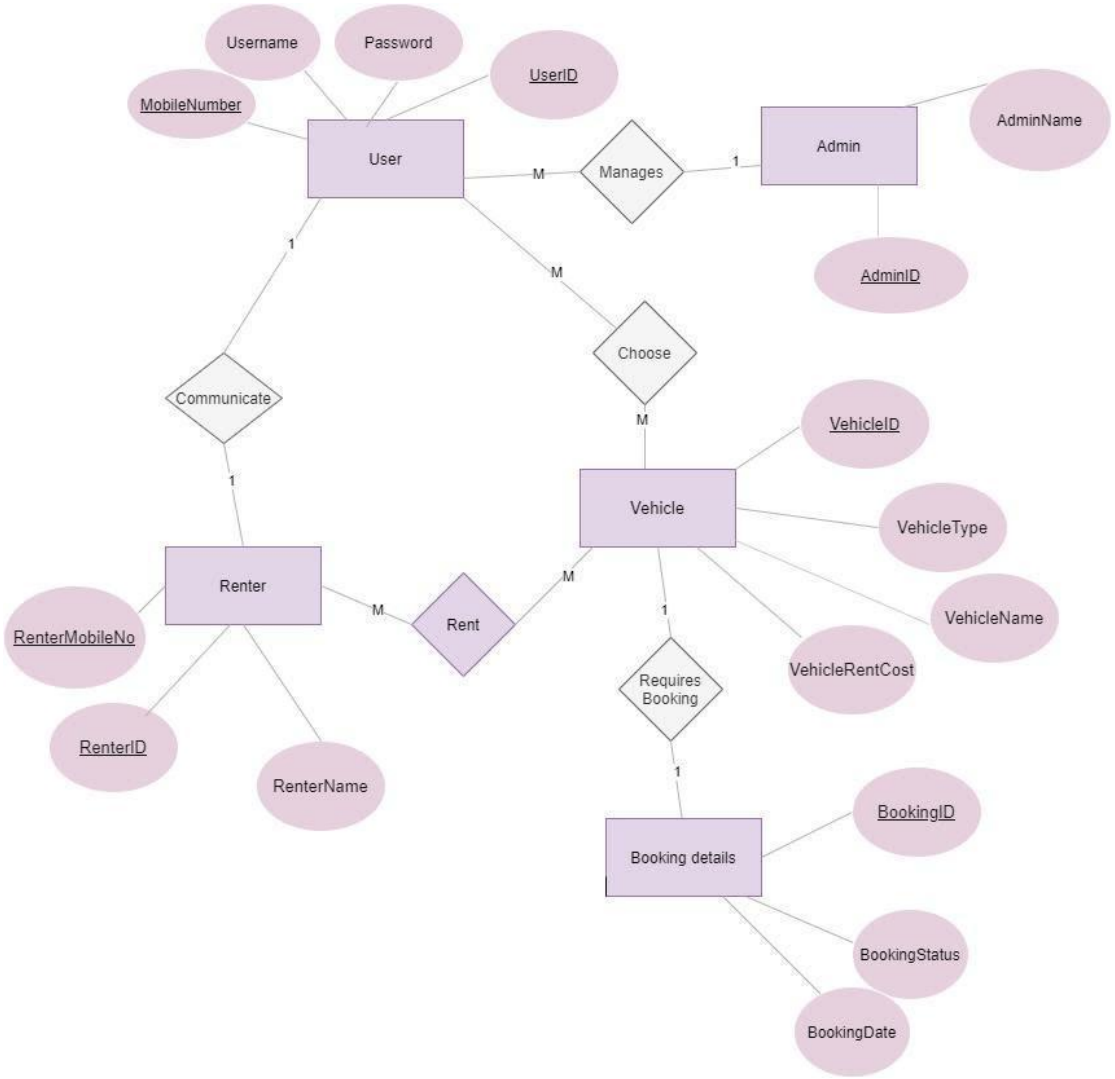
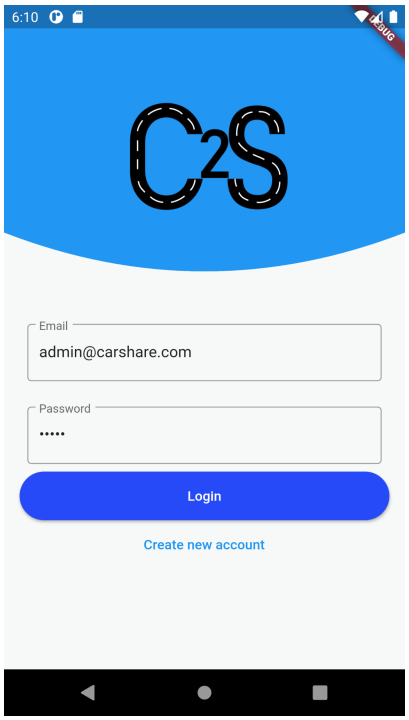


Figure 16 - ERD

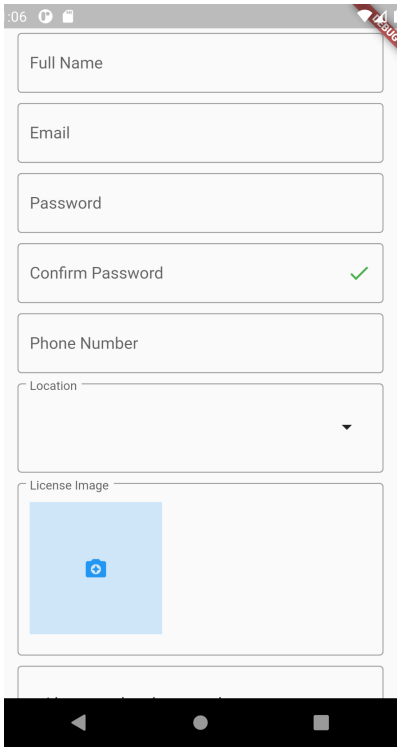
4.4 User Interface



For this interface, this is the first thing that's going to appear once the user launches the Car2Share app. The user can either choose to Log in to his/her existing account or to Sign up for a new account.

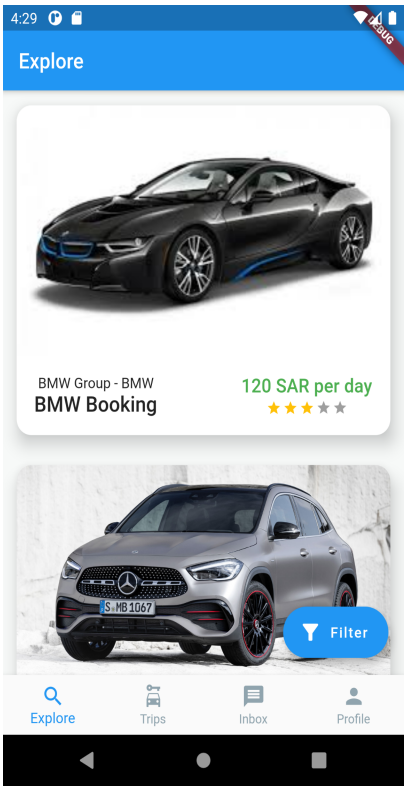
Once the user has chosen to Log in, he/she can either choose to enter the app by using their email as well as their password for verification purposes.

Figure 17 - Interface 1



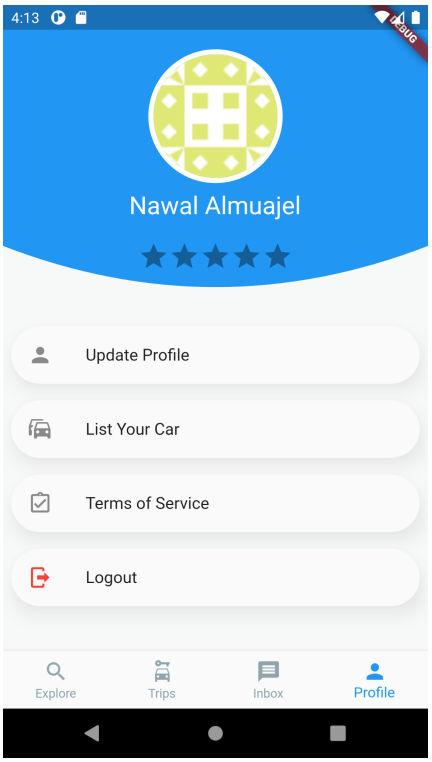
This interface has the sign-up page in which it prompts both the users, renter, and rentee to enter critical information such as the first and last name, email, password, phone number, and driving license ID to continue the sign-up process.

Figure 18- Interface 2



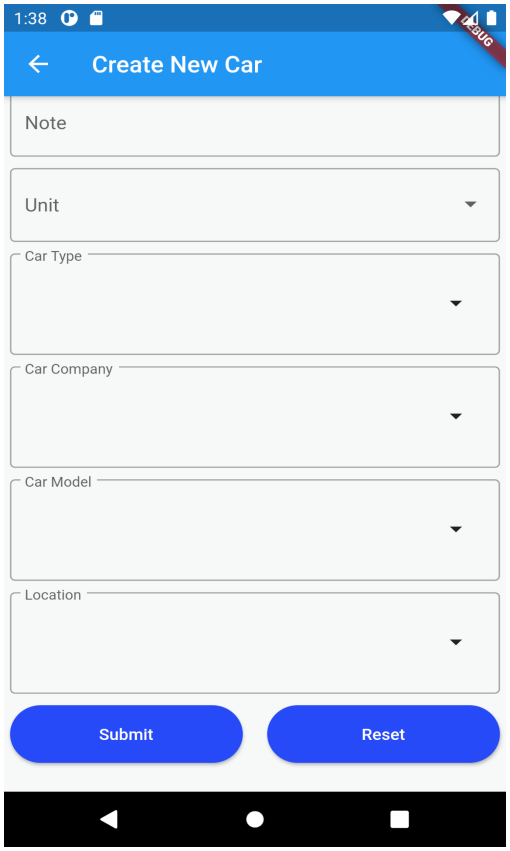
In this interface, it shows the explore page in which all the possible choices for the renter are displayed. Additionally, it also includes the filter feature which summarizes the user's options based on the user's preference and needs.

Figure 19 - Interface 3



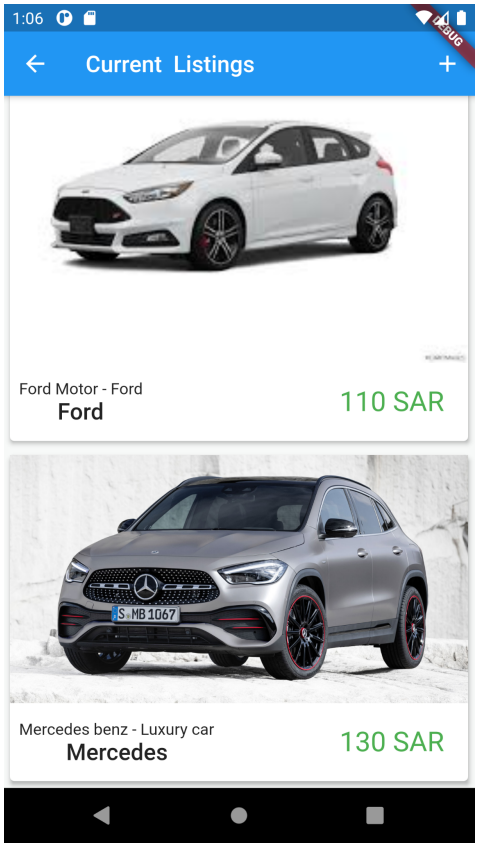
For the profile interface, the profile figure gives you the option to redirect to your personal information, Update profile, list your car, Current Listing, Term of Service. It also gives the user the option to log out.

Figure 20 - Interface 4



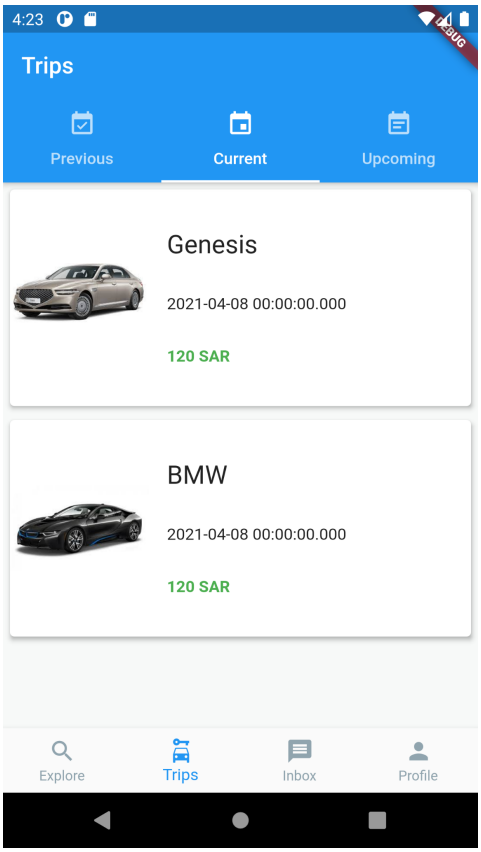
For the List your car interface, each user has the option to set up their car for listing and assign all the information needed.

Figure 21 - Interface 5



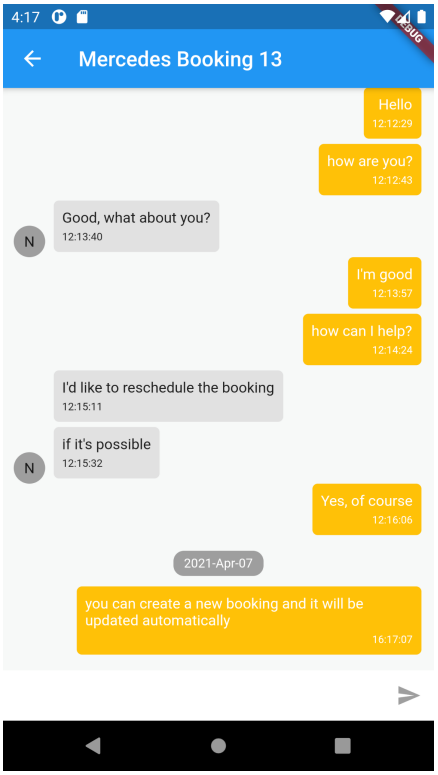
Once a vehicle has been registered, the user will be able to view their listings on the "Current Listings" page

Figure 22 - Interface 6



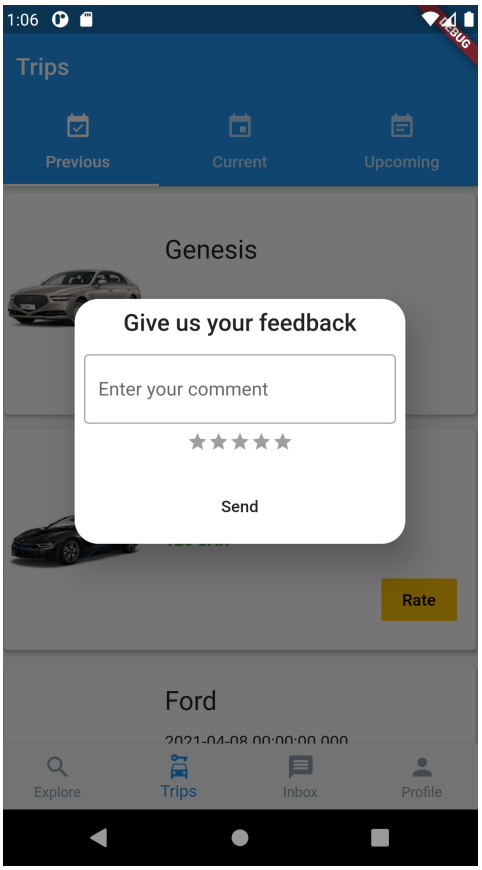
This interface allows the user to view previous trips. The page allows you to view the date, price, and more details about the trips the user has taken before and also the upcoming, current trips.

Figure 23 - Interface 7



This interface shows the exchange between a potential renter with the rentee where they specify the possible time and the availability of the requested car.

Figure 24 - Interface 8



The feedback interface enables the user to give feedback that car2share may benefit from. the feedback is provided through both filling out stars that resemble the level of satisfaction, and through enabling the user to express their opinion in a text box.

Figure 25 - Interface 9

Chapter 5: Implementation

Implementation

In this chapter, we will introduce the implementation techniques, hardware, and software that were used in this portion of the project. We will showcase snapshots of various interfaces of our system both in its last and revised version which may show some minor differences from our first perspective of the application in the previous chapters.

The system implementation is the fourth phase which is the Application Generation phase of the system life cycle of the RAD model. In this phase the project comes to life and begins to take shape, it also involves putting the final requirements and features together. The project becomes visible to the users by the end of this phase.

5.1 Implementation requirements

5.1.1 Hardware Requirements

In this project, the hardware requirements are:

1 - Personal Computer

- Handheld pointing device like a mouse
- Text entry device like a keyboard
- OS: Windows or Mac
- Possessor: 1.5 GHz o
- RAM: 2 GB o
- Wireless connectivity

2- Smartphone

- It gives us the ability to test and use the application.
- communicate between team members.

5.1.2 Software Requirements

In software requirements, the requirements are:

- Online Diagram Software to draw the UML diagrams
- Chrome to search about the needed information
- Microsoft Office programs such as Word processor, PowerPoint to document and present the research.
- MySQL database and web browser
- Web framework which is Laravel
- Intellij idea to analyze the code and to provides in-depth coding assistance, quick navigation, clever error analysis
- AndroidStudio to create an Android Virtual Device (AVD) that the emulator can use to install the Car2Share application

5.1.2.1 Admin interface

We will use two programming languages in admin interface implementation, Dart for the frontend and PHP for the backend. We will also use two programs to work and write the code, the first program, MAMP for database and host server, and the second program is PhpStorm it is a commercial, cross-platform IDE (integrated development environment) for PHP and works with our environment, which is Laravel valet.

5.1.2.1.1 Programming languages

1- Dart

Dart is a client-optimized programming language for apps on multiple platforms. It is used to build mobile, desktop, server, and web applications

2- PHP

It stands for "Hypertext Preprocessor." A prevalent scripting language that is used to create dynamic Web pages. Php combining syntax from the C, Java, and Perl languages.

5.1.2.1.2 Programs

1- MAMP

It stands for "Macintosh, Apache, Mysql, and PHP." It is a local server environment used to create hosts, is a stack composed of open-source, and software used to develop dynamic websites. It also provides a Phpmyadmin, an administration tool for MySQL; it is a portable web application written by PHP.

2- PhpStorm

PHP is a server-side scripting language. that is used to develop Static websites or Dynamic websites or Web applications.

3- IntelliJ IDEA

IntelliJ IDEA is an Integrated Development Environment for application development. IntelliJ IDEA is one of the top powerful code editors and tools. Also, it supports many programming languages such as C++, Java, and Dart.

4- Emulators

The emulators are programs created to Create and manage virtual devices to test the application with real devices' capabilities; we will use Android Emulator to test our application on Android OS and use XCode to test our application on iOS OS.

5.1.2.1.2 Framework Environment

Laravel Valet

After in-depth research on framework environments, we found the Laravel valet is the best framework for our project's requirements. Valet is a framework environment for Php; it is a fully integrated environment that provides fast and easy development tools and packages with minimal resources. We chose Laravel valet in our project; because we do not need a fully virtualized development environment for this project, and Laravel valet only requires PHP and MySQL.

5.1.2.2 User Interface

We will use Flutter framework and dart programming language in user interface implementation for both frontend and backend. We will also, for coding, use IntelliJ IDEA to write the code. For the code testing, we will use two emulators, Android Emulator, to test the application on the Android operating system and XCode to test the application on the iOS operating system

5.2 Implementation details

We are working in car2share to improve car rental by making it an easy and enjoyable experience. After research and drawing the use case, Entity-Relationship Diagram (ERD) and class diagram, sequence diagram. we will start programming by using PHP and Dart for admin and users. We will also use DB and coding programs, as explained in detail in the software requirements.

5.3 Code Screenshots

1-Database

For this PHP code, we have created a database table that contains rows and columns for ID, name, email, password, driver's license.

```php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{

    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->string('driver_license')->nullable();
            $table->string('phone_number')->nullable();
            $table->boolean('active')->default(0);;
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        }); }
    public function down() {
        Schema::dropIfExists('users');
```

Figure 26 - Code 1

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	driver_license	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
5	phone_number	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
6	active	tinyint(1)			No	0			Change Drop More
7	email_verified_at	timestamp			Yes	NULL			Change Drop More
8	password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
9	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
10	created_at	timestamp			Yes	NULL			Change Drop More
11	updated_at	timestamp			Yes	NULL			Change Drop More
12	fcm_token	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
13	location_id	bigint(20)		UNSIGNED	Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Print Propose table structure Move columns Normalize

Figure 27 - Code 1 Output

And for this PHP code, we have created another database table for creating a car

```php
class CreateCarsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('cars', function (Blueprint $table) {
            $table->id();
            $table->string('title', 100);
            $table->decimal('price');
            $table->string('unit', 250);
            $table->string('fule_type', 100);
            $table->longText('note');
            $table->string('image', 400);
            $table->timestamp('published_at')->nullable();
            $table->unsignedBigInteger('car_model_id');
            $table->unsignedBigInteger('user_id');
            $table->timestamps();
        });
    }

    /**
```

Figure 28 - Code 2

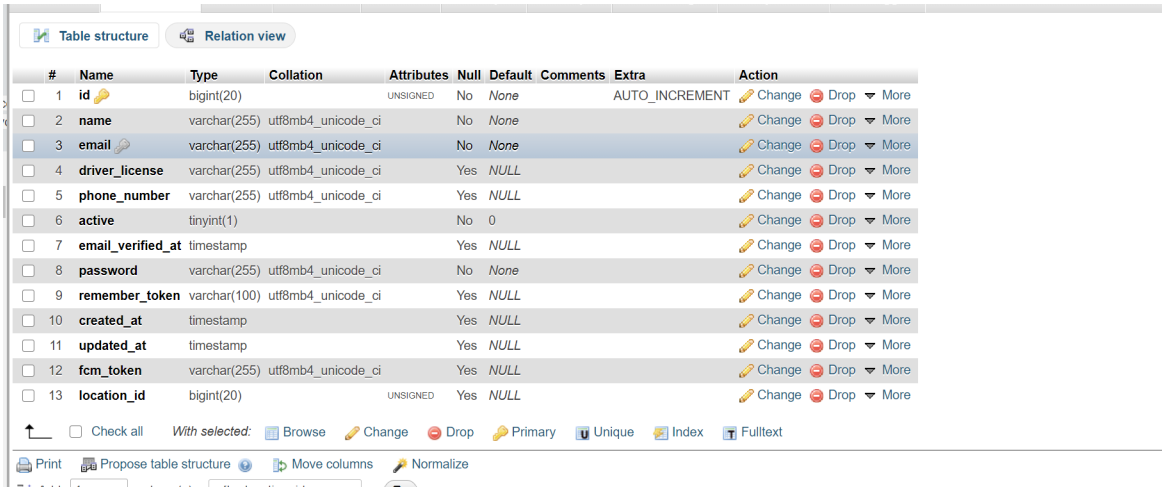


Figure 29 - Code 2 Output

2- Users registration

In the user registration code, we have imported some dart packages to access built-in and user-defined packages into our dart source file so that each class can refer to a class that is in another package by directly using its name. Also, we have provided a registration form with email and password fields. When a user provides a username and password, the inputs need to be validated, as we have added some validation functions to the field

Figure 30 - Code 3

Figure 31 - Code 4

```
child: FormBuilder(
  key: _formKey,
  autovalidateMode: AutovalidateMode.onUserInteraction,
  child: Padding(
    padding: const EdgeInsets.all(14.0),
    child: Column(
      children: [
        const SizedBox(height: 10),
        FormBuilderTextField(
          name: 'name',
          decoration: InputDecoration(
            labelText: 'Full Name',
          ),  // InputDecoration
          validator: FormBuilderValidators.compose([
            FormBuilderValidators.required(context),
          ]),
        ),  // FormBuilderTextField
        const SizedBox(height: 10),
        FormBuilderTextField(
          name: 'email',
          decoration: InputDecoration(...),  // InputDecoration
          validator: FormBuilderValidators.compose(...),
        ),  // FormBuilderTextField
        const SizedBox(height: 10),
        FormBuilderTextField(
          name: 'password',
          decoration: InputDecoration(...),
```

Figure 32 - Code 5

2-Login page:

For the login code page, we have two text fields, email, a, and password, to get login/sign-in credentials from the user. Also, we have used a function that will validate the passed value for the email and password.

```
children: [
const SizedBox(height: 30),
Padding(
padding: const EdgeInsets.all(8.0),
child: FormBuilderTextField(
name: 'email',
initialValue: 'admin@carshare.com',
decoration: InputDecoration(
labelText: 'Email',
errorText: _emailError,

),  // InputDecoration
]),  // FormBuilderTextField, Padding, ListView
),  // FormBuilder
),Padding(  // Container
padding: const EdgeInsets.all(8.0),

child: FormBuilderTextField(
name: 'password',
initialValue: '12345678',
decoration: InputDecoration(labelText: 'Password'),
obscureText: true,
validator: FormBuilderValidators.compose([
FormBuilderValidators.required(context),
FormBuilderValidators.minLength(context, 8),
```

Figure 33 - Code 6

3- Car details page

The renter set up their car for listing and assigned all the information needed, such as the booking date, car company, car type, car model, price.

```
const CarDetails({Key key, this.car}) : super(key: key);

@override
_CarDetailsState createState() => _CarDetailsState();
}

class _CarDetailsState extends State<CarDetails> {
// DateRangePickerController _dateRangePickerController;
DateTime start;
DateTime end;
List<DateTime> disable = [];
DatePeriod _selectedPeriod;
Future<List<Booking>> booked;
TimeRangeResult timeRange;

@override
void initState() {
  booked = Network.booked(widget.car.id);
  booked.then((value) {
    value.forEach((e) {
      print(e);
      setState(() {
        disable.addAll(calculateDaysInterval(
            DateTime.parse(e.startsAt), DateTime.parse(e.endsAt)));
      });
```

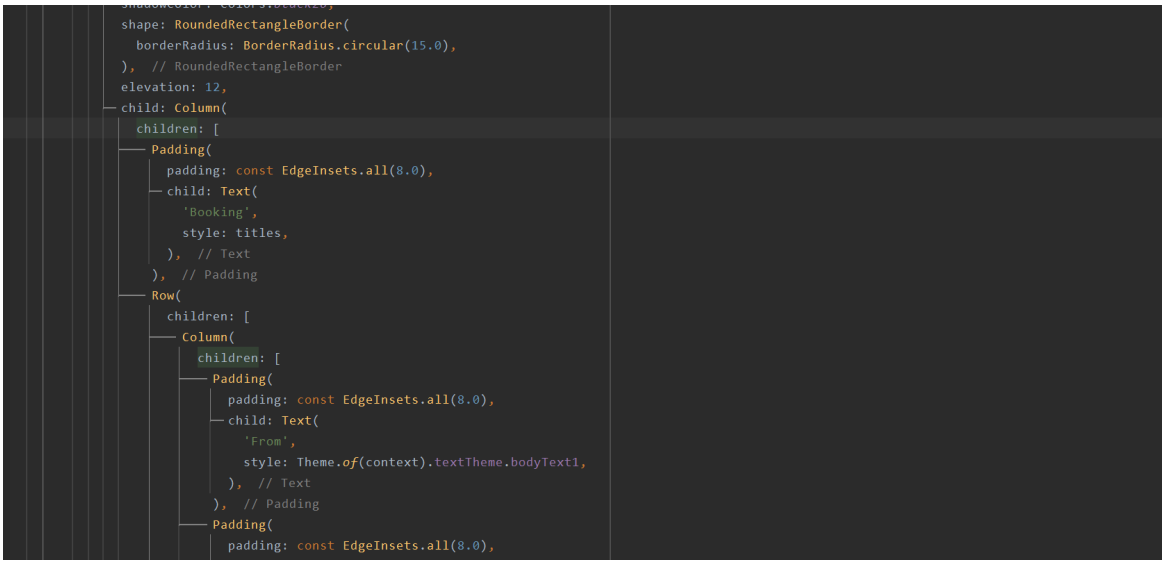
Figure 34 - Code 7

```
autovalidateMode: AutovalidateMode.always,
child: Column(
  children: <Widget>[
    FormBuilderImagePicker(
      name: 'image',
      decoration: const InputDecoration(labelText: 'Car Photo'),
      maxImages: 1,
    ),  // FormBuilderImagePicker
    FormBuilderTextField(
      name: 'title',
      decoration: InputDecoration(labelText: 'Title'),
      validator: FormBuilderValidators.compose([
        FormBuilderValidators.required(context,
            errorText: 'Title is required.'),
        (val) {
          return null;
        }
      ]),
    ),  // FormBuilderTextField
    const SizedBox(height: 10),
    FormBuilderTextField(
      name: 'price',
      decoration: InputDecoration(labelText: 'Price'),
      validator: FormBuilderValidators.compose([
        FormBuilderValidators.numeric(context,
            errorText: 'Price is required.'),
```

Figure 35 - Code 8

3- Trip details page

This code page allows users to view the date, price, and more details about the trips the user has taken before and also the upcoming, current trips.



```
shape: RoundedRectangleBorder(
  borderRadius: BorderRadius.circular(15.0),
), // RoundedRectangleBorder
elevation: 12,
child: Column(
  children: [
    Padding(
      padding: const EdgeInsets.all(8.0),
      child: Text(
        'Booking',
        style: titles,
      ), // Text
    ), // Padding
    Row(
      children: [
        Column(
          children: [
            Padding(
              padding: const EdgeInsets.all(8.0),
              child: Text(
                'From',
                style: Theme.of(context).textTheme.bodyText1,
              ), // Text
            ), // Padding
            Padding(
              padding: const EdgeInsets.all(8.0),
```

Figure 36 - Code 9

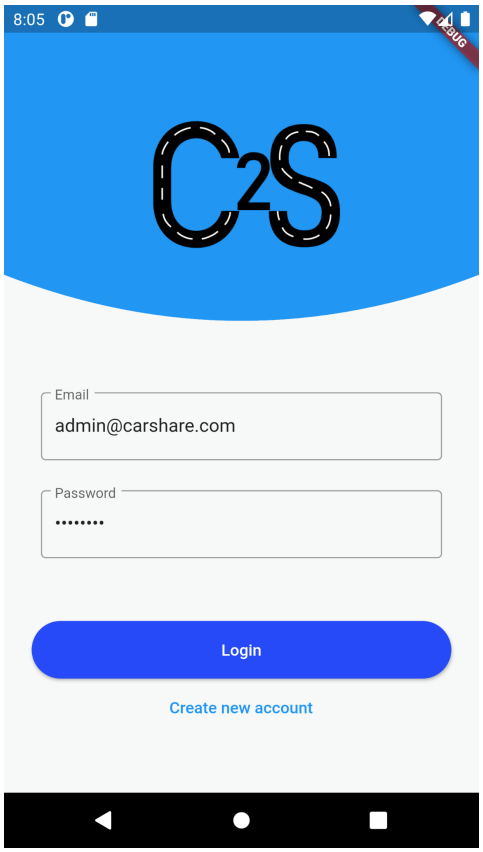
4- Chat page

For this page, we have used the firebase cloud messaging package to enable a renter to send and receive messages in real-time through the communication chat

```
class ChatPage extends StatefulWidget {
  final Thread thread;

  const ChatPage({Key key, this.thread}) : super(key: key);

  @override
  _ChatPageState createState() => _ChatPageState();
}

class _ChatPageState extends State<ChatPage> {
  Thread _thread;
  Future<Thread> _future;

  @override
  void initState() {
    _future = Network().chat(widget.thread.id);
    //if (_future != null) {
    print('chat message loaded');
    FirebaseMessaging.onMessage.listen((RemoteMessage message) {
      print('Got a message whilst in the foreground!');
      print('Message data: ${message.data}');
      if (message.notification != null) {
        if (message.data['thread_id'] == widget.thread.id.toString())
          setState(() {
            _thread.messages.add(Messages(
                id: int.parse(message.data['msg_id']),
                body: message.notification.body,
```

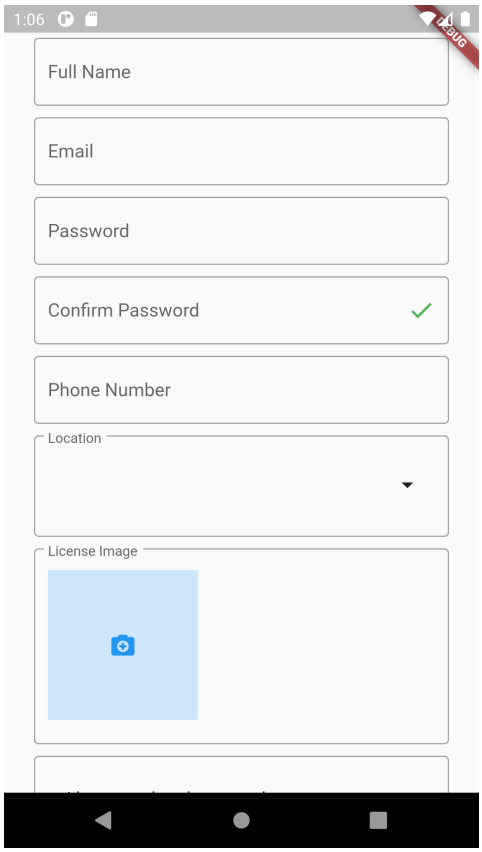
Figure 37 - Code 10

5.3.1 I\O Screens



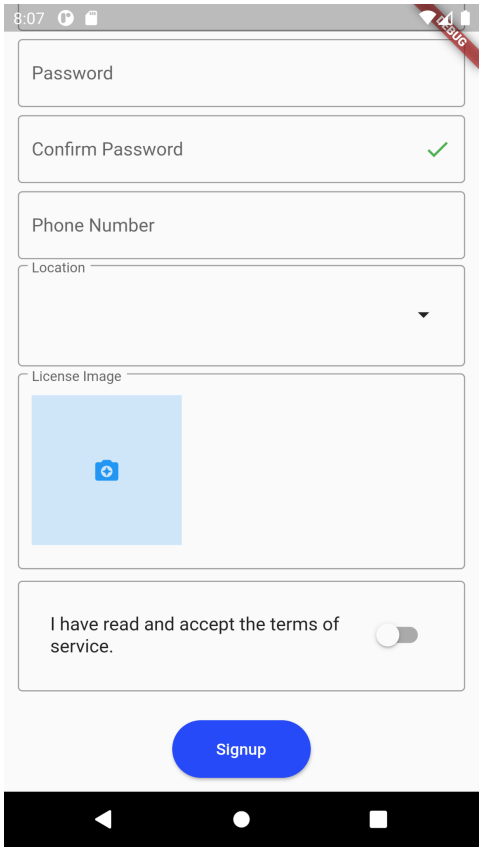
When the user opens the Car2Share application, this is the first thing that appears. The user has the option of signing in to an existing account or creating a new one.

Figure 38 - I/O 1



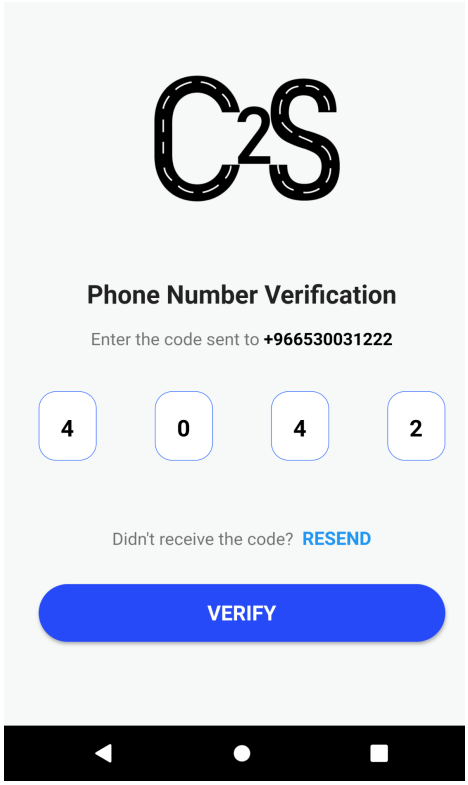
The sign-up page of this interface asks users to enter essential details such as first and last names, email, password, phone number, and driving license image, and location in order to begin the sign-up process.

Figure 39 - I/O 2



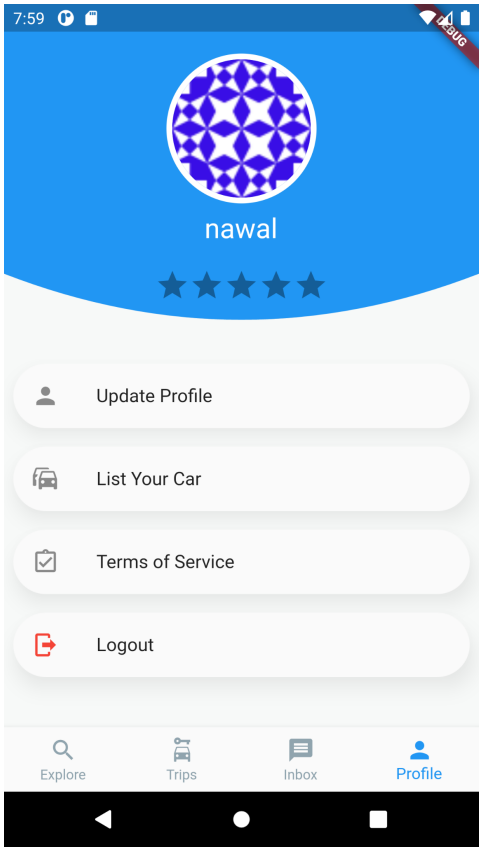
This interface allows the users to list their vehicle by inputting the required information.

Figure 40 - I/O 3



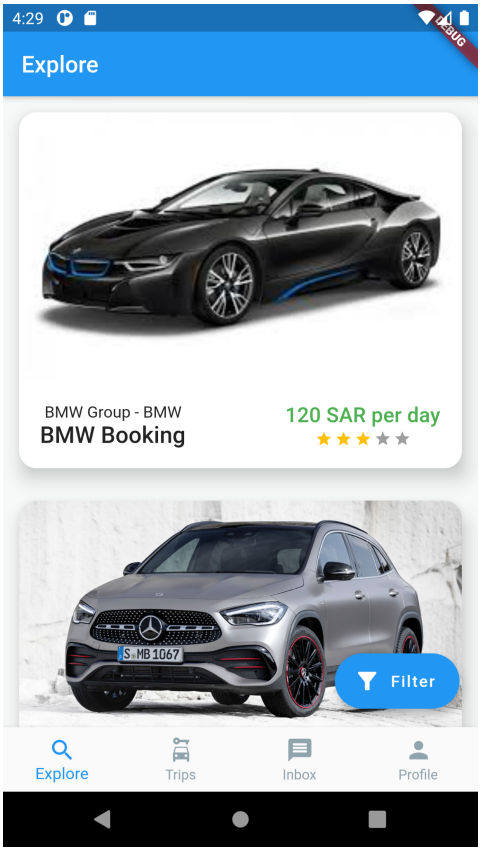
After the user registers to car2share, a verification code will be sent to their mobile phone number to log in to their account successfully.

Figure 41 - I/O 4



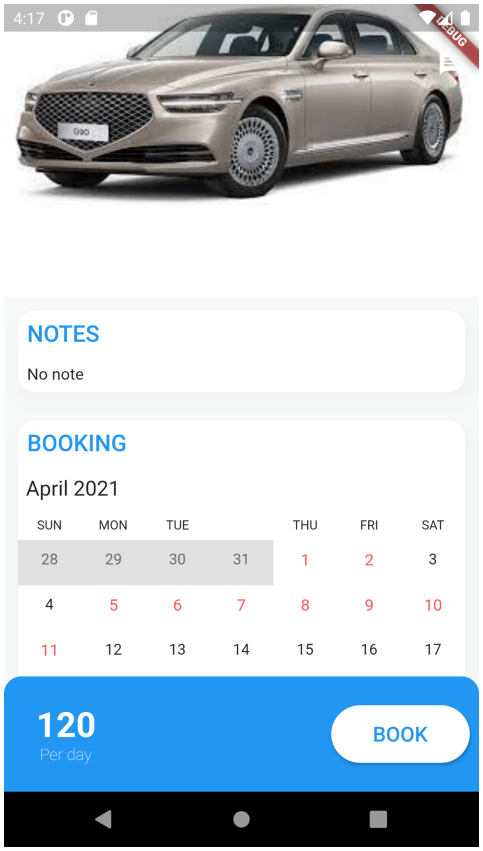
The profile page allows you to go to your personal details, update your profile, list your vehicle, view the Terms of Service. The user can also log out of the system.

Figure 42 - I/O 5



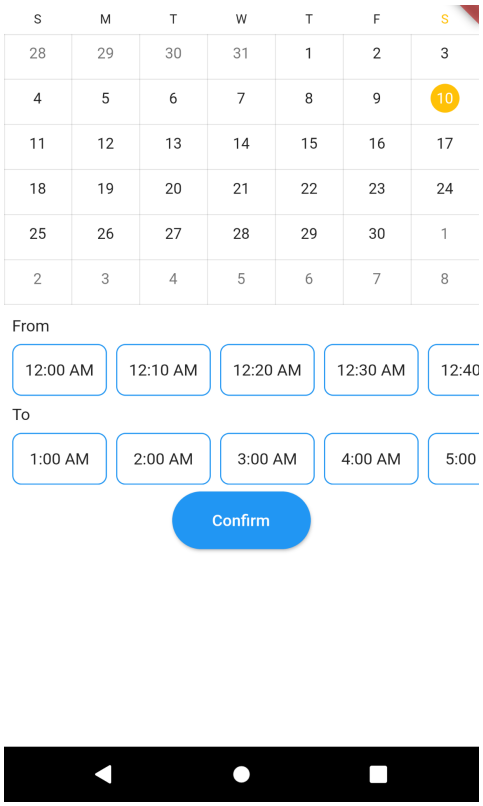
This interface displays the explore tab, which lists all of the options available to the renter. It also has a filter function that summarizes the user's choices based on their preferences and requirements.

Figure 43 - I/O 6



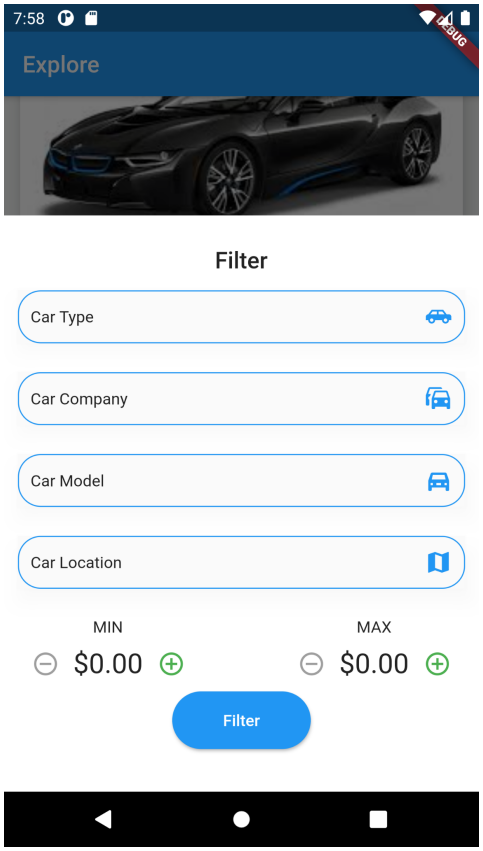
For this interface, it shows that this car can be booked by the date

Figure 44 - I/O 7



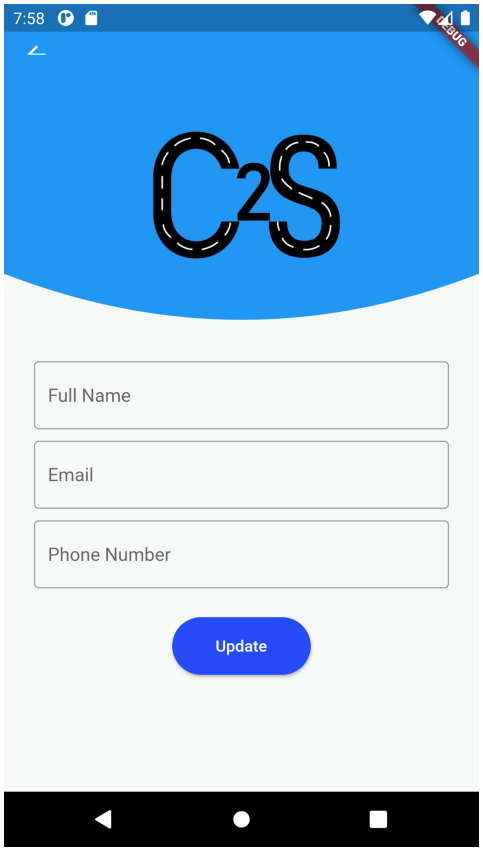
As for this interface, the user can book the car by the hour

Figure 45 - I/O 8



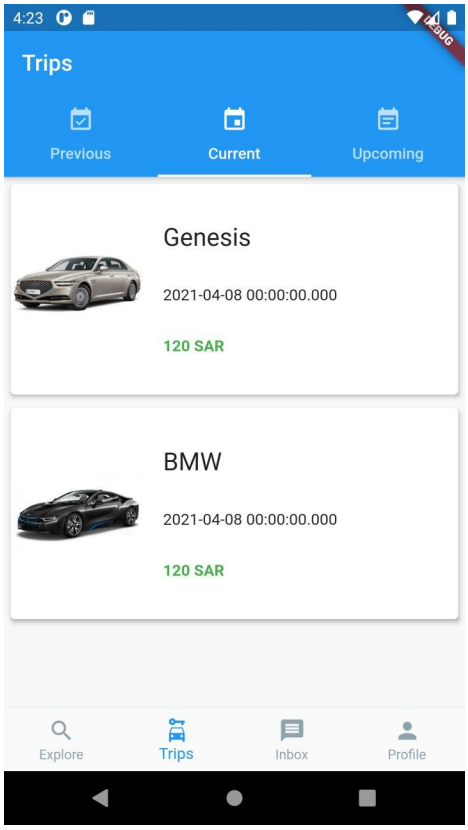
For this interface, the user can filter the cars on the explore page according to their preference of the car type, car company, car model, car location as well as car price

Figure 46 - I/O 9



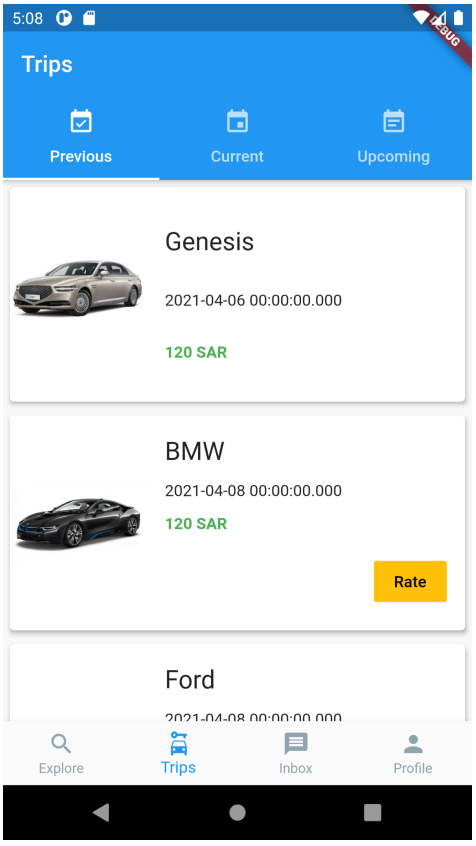
On the profile page, once the user clicks on the update profile, the users will be allowed to update and modify their name, email, phone number at any time if needed.

Figure 47 - I/O 10



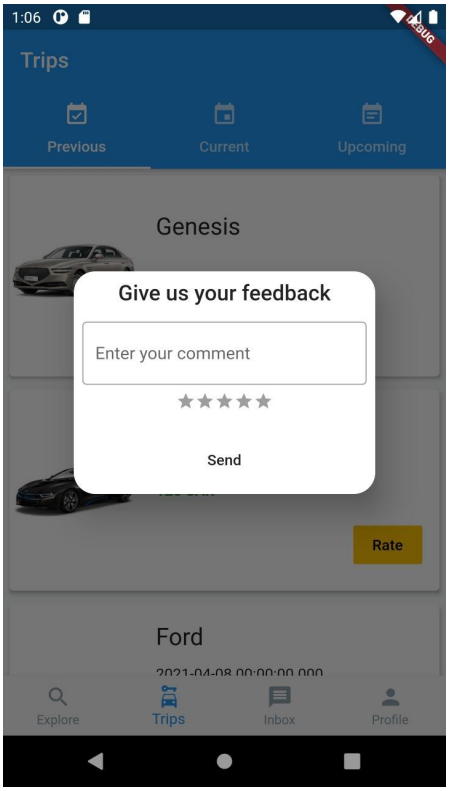
On the trips page, once the user clicks on the update profile, the users will be allowed to view previous, current and upcoming trips as well as the required information.

Figure 48 - I/O 11



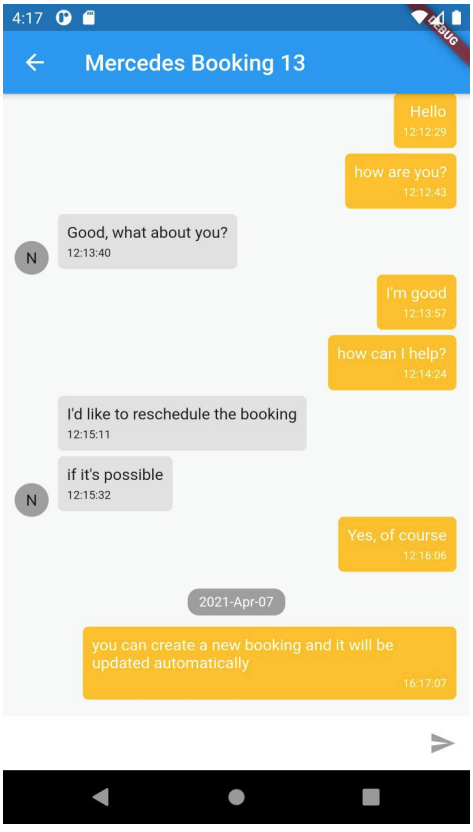
On the trips page, here we have the previous trips tab, the user will be permitted to rate their experience from the yellow Rate button.

Figure 49 - I/O 12



After clicking the Rate button, the interface will prompt the user to rate from 1-5 stars as well inputting any additional comment.

Figure 50 - I/O 13



On the inbox page, the rentee and renter will be able to communicate back and forth for any required communication, inquiry or extra information about the vehicle.

Figure 51 - I/O 14

Dashboard

When the admin accesses Car2Share, the first thing displayed will be the Dashboard in which the admin can manage, view, and monitor the information. These Dashboards organize, store and display the vital information that the admin must see and be aware of into one easily accessible place. In our dashboard, the admin can see the number of new users, newly registered car types, active users, booking count, and new cars in statistical charts. The admin has the choice of displaying the information for one day, one week, 30 days, or one year. Additionally, the admin has a panel displayed on the left of their screen that shows the options which showcase their various privileges in the application such as Car Companies, Car Models, Car Types, Cars (that are registered).

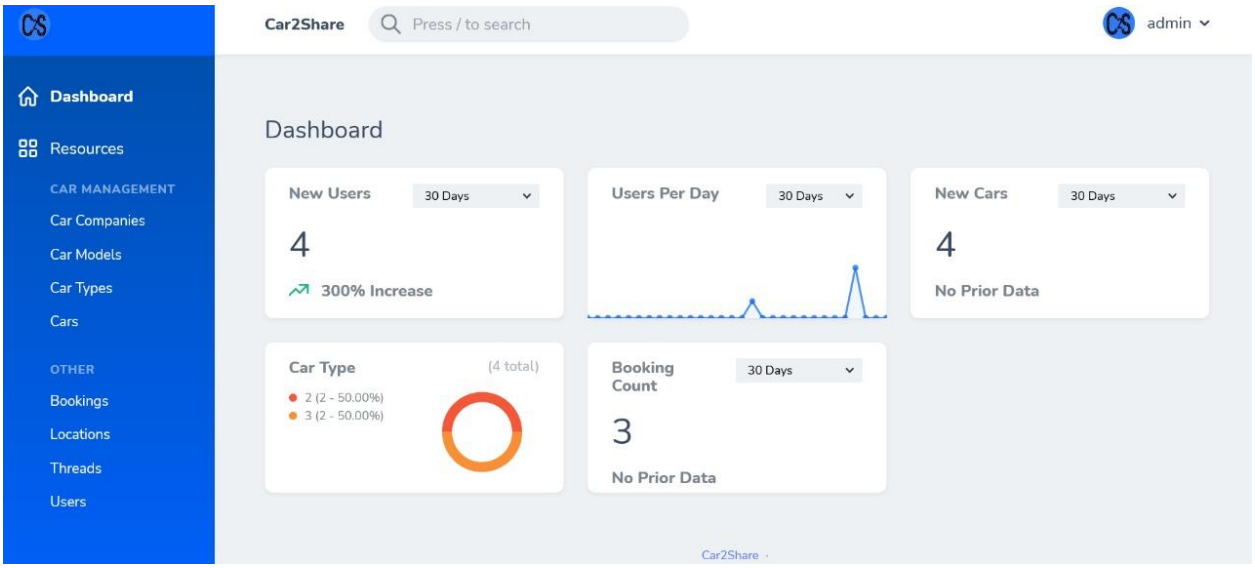


Figure 52 - Admin Interface 1

Car Companies page

When the admin opens the car companies page, it will display the list of the car companies that users can book. Also, The admin has the privilege to view, edit, delete any of the listed car companies. as well as creating new car companies.

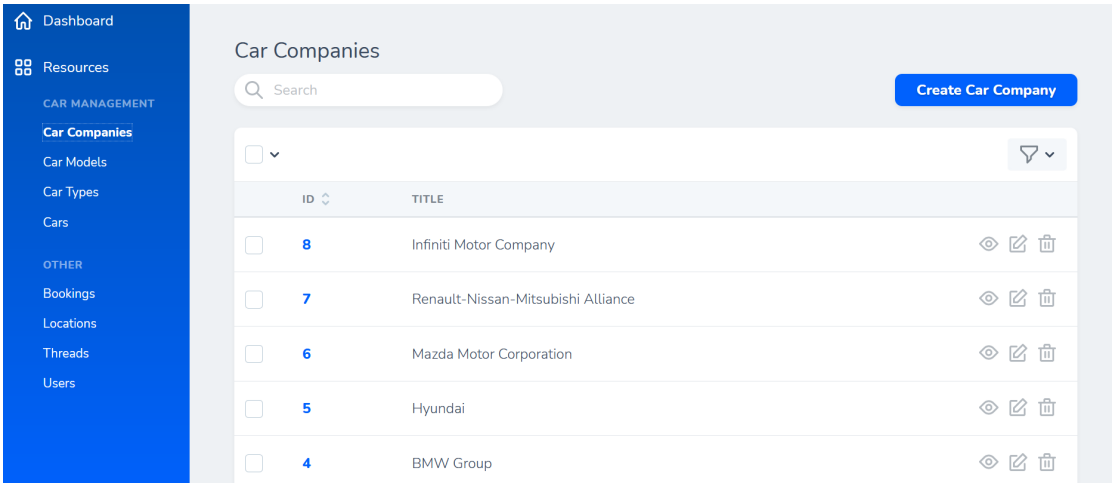


Figure 53 - Admin Interface 2

Car models page

On the car models page, it will display table details about each car, including; the title, car type, and the car company.

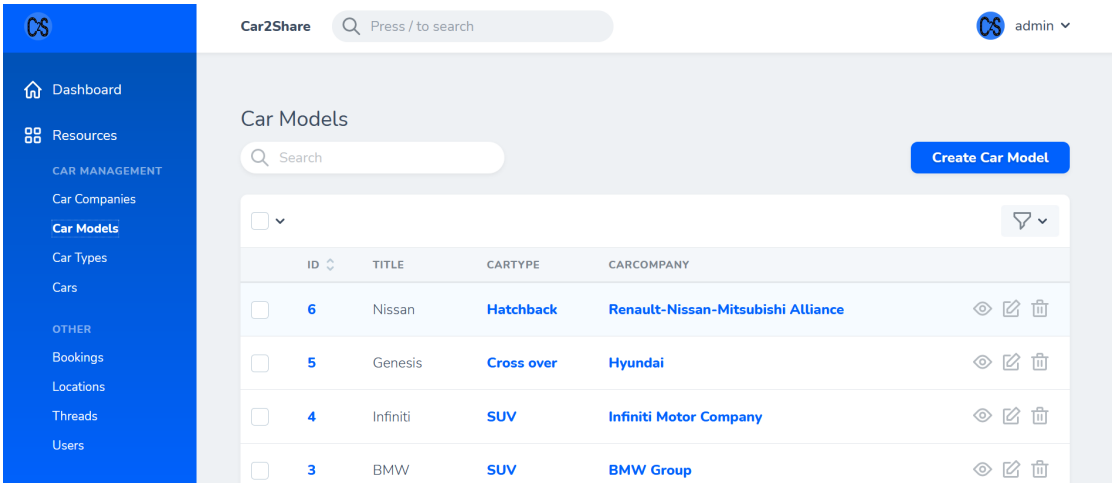


Figure 54 - Admin Interface 3

Bookings page

From the admin's control panel, the admin has access to the bookings page from which they can see all bookings that took place, the dates of those bookings, the prices, users, and types of cars booked.

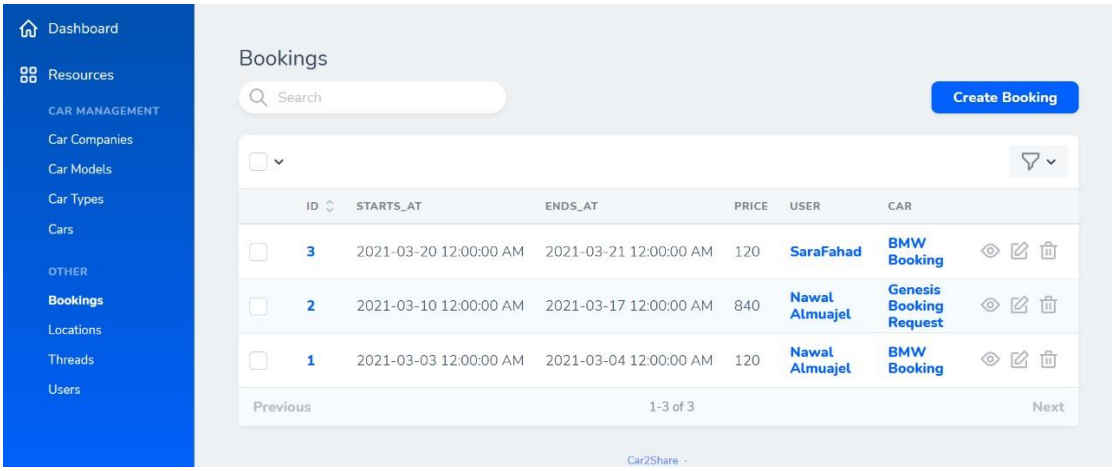


Figure 55 - Admin Interface 4

Threads page

On this page, the admin has full access to all communication that takes place between the renter and the rentee.

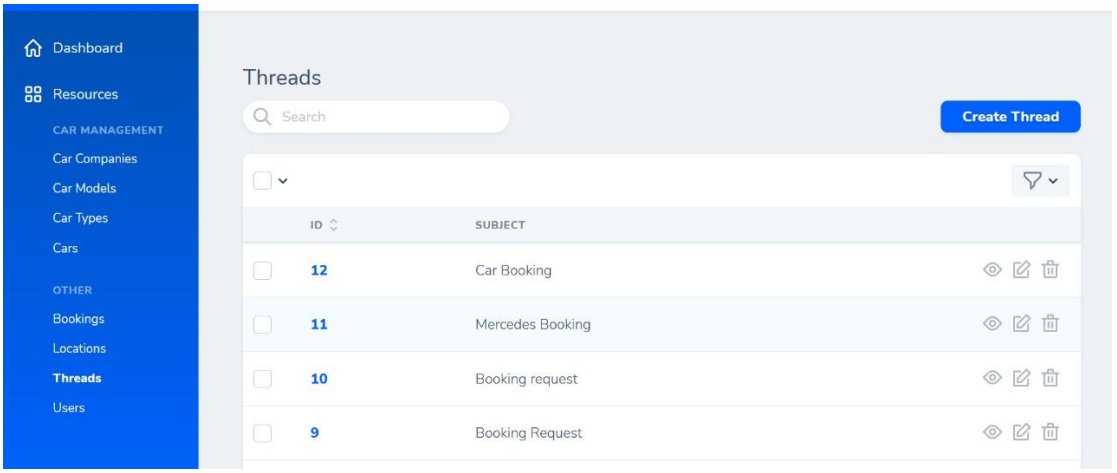


Figure 56 - Admin Interface 5

Users page

Users are managed from the Admin page through the Users tab. In the Users table, the admin can edit and delete users in our system. the table provides the following information about Car2share users:

- **ID**: The user's ID in car2share.
- **Name** The name of the user.
- **Email**: The user's email address.
- **Driver's license:** This shows an image of the Driver's license
- **Phone Number:** This shows the phone number of the user
- **Rating:** Users give rating points and the system takes an average rating and displays it in the overall review summary.
- **Active**: It shows if the users are active/inactive
- **Location**: It shows the registered location of each user.

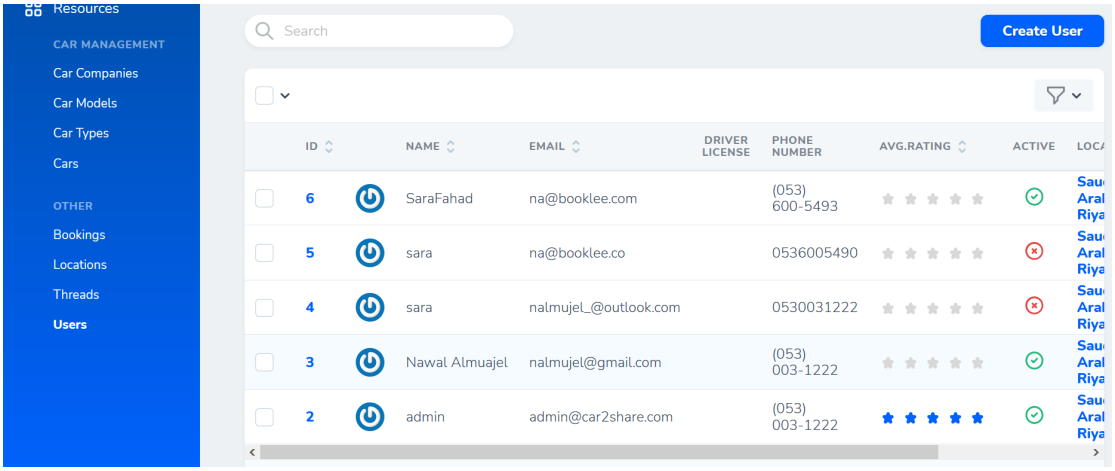


Figure 57 - Admin Interface 6

Chapter 6: Testing

Testing:

Chapter 6 will cover the testing phase of the implementation process. The testing phase is very important as it allows us to try out the application before officially “going live”, working out any possible issues, and ensuring the application is in its best form. In this chapter, the methods used to test will be presented. Specifically, the Android Emulator.

6.1 Testing Plan

The testing plan is the portion of this report in which the scope of the testing and activities are detailed. The Android Emulator, provided by Android studio, was used to test all aspects and functions of the system. The details will be covered in this report. The system was tested repeatedly during the implementation phase. It included testing all the functionality, performance, features, and database. The testing strategy begins with the unit components, moving forward to the system’s entire functionality, and finally with the user acceptance testing (UAT).

The project was separated into two parts, the admin interface, and the user interface. The admin interface was made as a website while the user interface is the application. Once those were established, the work was segmented into units and tested individually after implementation to ensure that any errors were identified and fixed.

6.2 Testing Strategy

6.2.1 Unit Testing

The primary goal of unit testing is to divide the system into the part and test them each individually. That is the process followed for Car2Share to help ensure that each unit behaves ideally before it is integrated into the system as a whole.

6.2.2 Functional Testing

The functional testing helps ensure the application is meeting the set specifications appropriately. Thus, the performance of the system is tested to examine the functionality, performance, features, database, communication chat, and security, ensuring that it simulates the system's real use accurately.

6.2.3 Acceptance Testing

Also known as UAT, acceptance testing is the most critical phase of testing. It is the factor that decides if the client accepts the interface or not. As well, the test aims to assess compliance with business requirements which involves usability, functionality, user interface, and performance.

6.3 Test Cases

6.3.1 Admin Test Case

Table 6.1 – Admin Test Case

Unit	Test Case	Expected Result	Actual Result	Success/Fail
	Login/Logout	Login: Admin should be able to login into the application. Logout: Admin should be able to log out of the application.	As expected	Success
	Add	Enable the admin to add new information whether it be about the users or the vehicles.	As expected	Success
	Delete	Enables the admin to delete information about either the users or the vehicles.	As expected	Success
	Edit	Enables the admin to edit information about either the users or the vehicles.	As expected	Success
	Search	Available on all pages, allowing the admin to search by ID, vehicle, or username.	As expected	Success
	Check Password	Checks password quality when registering, later checks if the password is correct when signing in.	As expected	Success
	View Information	Enables admins to access all user information, excluding passwords, as well as all service information.	As expected	Success
	Filter	Enables renter to filter the through table as well as the vehicles table.	As expected	Success

6.3.2 Renter Test Case

Table 6.2 – Renter Test Case

Unit	Test Case	Expected Result	Actual Result	Success/Fail
	Login/Logout	Login: renter should be able to login into the application. Logout: renter should be able to log out of the application.	As expected	Success
	Create Account	Enables renter to create a new account.	As expected	Success
	Check Password	Checks password quality when registering, later checks if the password is correct when signing in.	As expected	Success
	Authentication	Renter will receive an SMS when registering to verify the inputted phone number	As expected	Success
	List Vehicle	Enables the renter to list their vehicle on the application.	As expected	Success
	Delete	Enables the renter to delete information about either the vehicle/s listed or their account.	As expected	Success
	Edit	Enables the renter to edit information about either the vehicle/s listed or their account.	As expected	Success
	Change Password	Enables renter to change their account password if needed	As expected	Success
	Search	Enables the renter to search any page in the application.	As expected	Success
	View	Enables renter to view all pages that are in the application.	As expected	Success
	Communicate Through Chat	Enables renter to send and receive messages in real-time through the communication chat	As expected	Success
	Feedback	Enables renter to review the experience after it has ended	As expected	Success

6.3.3 Rentee Test Case

Table 6.3 – Admin Test Case

Unit	Test Case	Expected Result	Actual Result	Success/Fail
	Login/Logout	Login: rentee should be able to login into the application. Logout: rentee should be able to log out of the application.	As expected	Success
	Create Account	Enables rentee to create a new account.	As expected	Success
	Check Password	Checks password quality when registering, later checks if the password is correct when signing in.	As expected	Success
	Authentication	Rentee will receive an SMS when registering to verify the inputted phone number	As expected	Success
	Delete	Enables the rentee to delete information on their account.	As expected	Success
	Edit	Enables the rentee to edit information on their account.	As expected	Success
	Change Password	Enables rentee to change their account password if needed	As expected	Success
	Search	Enables the rentee to search any page in the application.	As expected	Success
	View	Enables renter to view all pages that are in the application.	As expected	Success
	Filter	Enables rentee to filter through the given options on the explore page	As expected	Success
	Communicate Through Chat	Enables rentee to send and receive messages in real-time through the communication chat	As expected	Success
	Feedback	Enables rentee to review the experience after it has ended	As expected	Success

6.4 Test Results

6.4.1 Admin Test Results

Table 6.4 – Admin Test Case Results

Unit	Test Case	Percentage of Completeness
	Login/Logout	100%
	Add	100%
	Delete	100%
	Edit	100%
	Search	100%
	Check Password	100%
	View Information	100%
	Filter	100%

6.4.2 Renter Test Results

Table 6.5 – Renter Test Case Results

Unit	Test Case	Percentage of Completeness
	Login/Logout	100%
	Create Account	100%
	Check Password	100%
	Authentication	100%
	List Vehicle	100%
	Delete	100%
	Edit	100%
	Change Password	100%

	Search	100%
	View	100%
	Communicate Through Chat	100%
	Feedback	100%

6.4.2 Rentee Test Results

Table 6.6 – Rentee Test Case Results

Unit	Test Case	Percentage of Completeness
	Login/Logout	100%
	Create Account	100%
	Check Password	100%
	Authentication	100%
	Delete	100%
	Edit	100%
	Change Password	100%
	Search	100%
	View	100%
	Filter	100%
	Communicate Through Chat	100%
	Feedback	100%

Chapter 7: Conclusion

7.1 Evaluation

By spending a lot of time studying and exploring the best ways to tackle our project, we were able to fulfill our targets at the end of the project. We aimed to modernize the conventional car rental process, making it more beneficial to both customers (the renter and the rentee). The test results show that Car2Share has been developed successfully and without errors. Our project aims to make the process of renting a car as easy, convenient, and effective as possible. A high degree of protection has also been achieved, as part of the registration process on Car2Share, the user is required to provide a verification number via phone number.

7.2 Future Work

In the future, we hope to have Car2Share available on the App Store and Playstore, we hope to accumulate numerous investors and sponsors that help elevate and improve the application to expand beyond the Middle East. In the future, we aim to include several more features like “best offer”, discounts and coupons. Additionally, we hope to maintain the app’s usability constantly through continuous updates as we strive for a user-friendly experience that enhances the efficiency of the system and encourages our customers to share and use the application regularly.

7.3 Conclusion

In conclusion, Car2Share is a peer-to-peer car rental service primarily based in the MENA area. Our application is unique in that it provides new sources of income for its users who want to put their car up for rental, it provides people that have a license and require a car with the fastest way to access the nearest car of their preference.

Car2Share is both an Android and IOS application that has been successfully developed and tested while also achieving its initial objectives. It also sufficiently provides its unique features to its customers in an effortless and timely manner. This project was done through the work of an integrated application such as the Dart language that adequately provides all the needs of Car2Share. We strongly hope that the targeted users of this project will benefit from this application.

References

[1] "Saudi Arabia to implement VAT in 56 days", *Gazt.gov.sa*, 2020. [Online]. Available: https://gazt.gov.sa/en/MediaCenter/News/Pages/News_099.aspx. [Accessed: 15- Nov- 2020].

[2] "Get a ride, drive, order food and pay with Careem", *Careem.com*, 2020. [Online]. Available: https://www.careem.com/. [Accessed: 15- Nov- 2020].

[3] "About Us | Uber", *Uber*, 2020. [Online]. Available: https://www.uber.com/sa/en/about/. [Accessed: 15- Nov- 2020].

[4] "Saudi economy and COVID-19: 'Good can come of evil'", *Arab News*, 2020. [Online]. Available: https://www.arabnews.com/node/1725751/saudi-arabia. [Accessed: 15- Nov- 2020].

[5] C. Valor, "Anticipated emotions and resistance to innovations: the case of p2p car sharing," *Environ. Innov. Soc. Transit.*, vol. 37, pp. 50–65, 2020.

[6] A. Quito, "In Saudi Arabia, women can finally drive", *Quartz*, 2020. [Online]. Available: https://qz.com/1313101/saudi-arabias-women-are-finally-allowed-to-drive-a-car-on-their-own/. [Accessed: 15- Nov- 2020].

[7] "KSA VAT rate to increase to 15% from 1 July 2020", *Deloitte*, 2020. [Online]. Available: https://www2.deloitte.com/sa/en/pages/tax/articles/ksa-vat-rate-increase-15percent-1-july-2020.html. [Accessed: 15- Nov- 2020].

[8] S. Shaikh and S. Abro, "COMPARISON OF TRADITIONAL AND AGILE SOFTWARE DEVELOPMENT METHODOLOGY: A SHORT SURVEY", *International Journal of Software Engineering and Computer Systems*, vol. 5, no. 2, pp. 1-14, 2019. Available: 10.15282/ijsecs.5.2.2019.1.0057.

[9] B. Systems, "Agile Methodologies", *Blueprintsys.com*, 2020. [Online]. Available: https://www.blueprintsys.com/agile-development-101/agile-methodologies. [Accessed: 15- Nov- 2020].

[10] "Turo | The world's largest car sharing marketplace", *Turo.com*, 2020. [Online]. Available: https://turo.com/. [Accessed: 15- Nov- 2020].

[11] "Rent a car or earn money sharing your car - Drive Drive Car", *Drivedrivecar.com*, 2020. [Online]. Available: https://www.drivedrivecar.com/. [Accessed: 15- Nov- 2020].

[12] "Nabobil.no - Private car rental in your neighbourhood", *Nabobil.no*, 2020. [Online]. Available: https://nabobil.no/en. [Accessed: 15- Nov- 2020].

[13] S. Alsaleh and H. Haron, "The Most Important Functional and Non-Functional Requirements of Knowledge Sharing System at Public Academic Institutions: A Case Study", *Lecture Notes on Software Engineering*, vol. 4, no. 2, pp. 157-161, 2016. Available: 10.7763/lnse.2016.v4.242.

[14] H. Kaur and D. Sharma, "Non-Functional Requirements Research: Survey", *International Journal of Science and Engineering Applications*, vol. 3, no. 6, pp. 172-182, 2014. Available: 10.7753/ijsea0306.1003.

[15] L. Hylving and U. Schultze, "Accomplishing the layered modular architecture in digital innovation: The case of the car's driver information module", *The Journal of Strategic Information Systems*, vol. 29, no. 3, p. 101621, 2020. Available: 10.1016/j.jsis.2020.101621.

[16] P. real and S. 6, "Proto.io - Prototypes that feel real", *Proto.io*, 2020. [Online]. Available: https://proto.io/. [Accessed: 15- Nov- 2020].

[17]"Corporate", *Tawuniya.com.sa*, 2020. [Online]. Available: https://www.tawuniya.com.sa/en/corporate. [Accessed: 15- Nov- 2020].

[18] "Home Page", *Gulfunion*, 2020. [Online]. Available: https://gulfunion.com.sa/. [Accessed: 15- Nov- 2020].

Appendix

```php

    public static $model = \App\Models\Booking::class;

    public static $title = 'id';

    public static $search = [
        'id',
    ];


    public function fields(Request $request)
    {
        return [
            ID::make(__('ID'), 'id')->sortable(),
            DateTime::make('starts_at'),
            DateTime::make('ends_at'),
            Number::make('price'),
            BelongsTo::make('user'),
            BelongsTo::make('car'),

        ];
    }
```

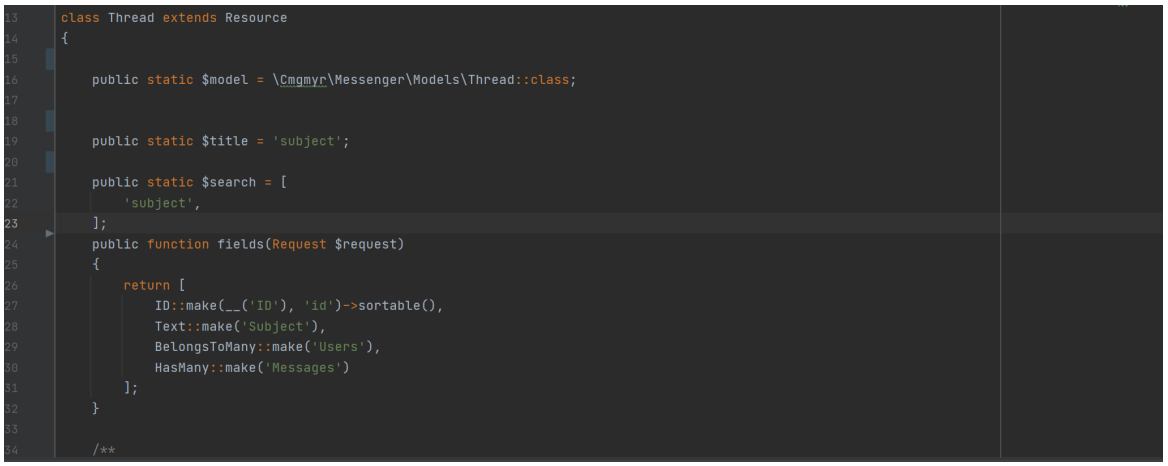
Figure 58 - Booking code

```php
use Laravel\Nova\Fields\ID;
use Laravel\Nova\Fields\Text;
use Laravel\Nova\Http\Requests\NovaRequest;

class Location extends Resource
{

    public static $model = \App\Models\Location::class;


    public static $title = 'name';


    public static $search = [
        'name',
    ];
    public function fields(Request $request)
    {
        return [
            ID::make(__('ID'), 'id')->sortable(),
            Text::make('Name'),
            HasMany::make('cars')
        ];
```

Figure 59 - Location code

```
class CarModel extends Resource
{
    public static $model = \App\Models\CarModel::class;

    public static $title = 'title';
    public static $search = [
        'id',
    ];
    public static $group = 'Car Management';

    public function fields(Request $request)
    {
        return [
            ID::make()->sortable(),

            Text::make('Title')
                ->rules('required', 'string', 'max:100'),


            BelongsTo::make('CarType', 'carType', CarType::class),
            BelongsTo::make('CarCompany', 'carCompany', CarCompany::class),
        ];
```

Figure 60 - Car database code



```
class Thread extends Resource
{

    public static $model = \Cmgmyr\Messenger\Models\Thread::class;


    public static $title = 'subject';

    public static $search = [
        'subject',
    ];
    public function fields(Request $request)
    {
        return [
            ID::make(__('ID'), 'id')->sortable(),
            Text::make('Subject'),
            BelongsToMany::make('Users'),
            HasMany::make('Messages')
        ];
    }

    /**
```

Figure 61 - Thread code

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	driver_license	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
5	phone_number	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
6	active	tinyint(1)			No	0			Change Drop More
7	email_verified_at	timestamp			Yes	NULL			Change Drop More
8	password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
9	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
10	created_at	timestamp			Yes	NULL			Change Drop More
11	updated_at	timestamp			Yes	NULL			Change Drop More
12	fcm_token	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
13	location_id	bigint(20)		UNSIGNED	Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Print Propose table structure Move columns Normalize

Figure 62 - Attributes database

```
children: [
const SizedBox(height: 30),
Padding(
padding: const EdgeInsets.all(8.0),
child: FormBuilderTextField(
name: 'email',
initialValue: 'admin@carshare.com',
decoration: InputDecoration(
labelText: 'Email',
errorText: _emailError,

), // InputDecoration
]), // FormBuilderTextField, Padding, ListView
), // FormBuilder
),Padding(  // Container
padding: const EdgeInsets.all(8.0),

child: FormBuilderTextField(
name: 'password',
initialValue: '12345678',
decoration: InputDecoration(labelText: 'Password'),
obscureText: true,
validator: FormBuilderValidators.compose([
FormBuilderValidators.required(context),
FormBuilderValidators.minLength(context, 8),
```

Figure 63 - Login Page Code