An-Najah National University

Faculty of Engineering & Information Technology

Department of Computer Engineering

Software Graduation Project

Garagy

Authors:

Mustafa Sokar , Amr abo Amr

Supervisor:

Dr.Emad Natsheh

Date: May, 2023

Disclaimer

This report was written by students (Mustafa Sokar and Amr abo Amr) at the Computer Engineering Department, Faculty of Engineering, An- Najah National University. Therefore, has not been corrected or modified as a consequence of assessment and it may contain language as well as content errors. The ideas and opinions stated, as well as any views or suggestions, are purely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Contents

Abstract

The purpose of this project is to help the users to maintain their cars and also to help car technicians finding jobs, linking user to a wide network of car maintenance centers with services that reach them without tiredness or effort, whether they need to transport the car to the workshop, or it's a small issue and requires a mobile workshop that reaches your location. The project consists of 3 main sides: user, service provider (technicians) and the admin. We will implement a mobile application and website for this project. Users can choose what kind of service they want e.g. air conditioning repair , electrical inspection , mechanical inspection , paint work ,balance or even asking for fuel. Also there is a map to help users choose the closest workshop or service provider. Users can rate the workshops ,communicate with them using the chat . In addition to that, there will be notifications to notify the user with all updates about his car (service) status. Also the store of our project will sell the car parts and accessories to help users find the best products according to his car model. Users can pay for service or products using this app. The app will notify the service provider with a form contains some information about the car that needs a maintenance (or will describe the problem), technicians could accept or deny, also could sell some parts on the app's store, the service provider could display information like location, price of the service, what his major, what car models did he prefer also he could display some of his work on his account. The admin will manage the app) the services it provides, the sales, workshops, store(, updating data and will solve problems faces the users

Chapter 1: Introduction

1.1 Problem

Car owners often struggle to find reliable and convenient car maintenance services. They face difficulties in identifying suitable workshops or service providers for specific maintenance needs such as air conditioning repair, electrical inspection, mechanical inspection, paintwork, balancing, fuel assistance, and more. Additionally, the process of transporting the car to a workshop or waiting for a mobile workshop to arrive at their location can be tiresome and time-consuming. Furthermore, car owners lack a platform to communicate and engage with service providers effectively.

On the other hand,service providers, including car technicians, often face challenges in securing jobs and connecting with potential customers. They struggle to showcase their skills and expertise to a wider audience. It becomes challenging for them to expand their customer base and build a reputable presence in the car maintenance industry.

1.2 Objective

User experience : the main objective of the project is to provide users with a convenient and user friendly platform to maintain their cars.

Efficient service provider matching : we aim to create a wide network of car maintenance within the app . By implementing an easy way to reach service providers that fit the client's needs .
On the other hand , we seek to provide workers with an easy way to get jobs and offers.

Easy communication : we seek to enhance communication between users and workers or service providers through the chat functionality .

1.3 Scope of the Work

1- **Car owners** :

Car owners are the primary target for the car maintenance app. These people own cars and are looking for convenient and reliable car maintenance services. We seek a digital solution that allows them to easily locate workshops or service providers based on their specific needs. The app provides an efficient process for requesting and monitoring car maintenance services. Additionally, car owners can shop through a wide range of car parts and accessories through the app's store.

2- **Service Providers:**

Including car technicians and workshops, are a importante audience for the app. They aim to improve their customer base and increase their visibility in the car maintenance field . The app serves as a platform for workers to show their skills and experience. They can efficiently manage their services, availability, and pricing information through the app's. Additionally, service providers have the opportunity to sell car parts and accessories, thereby maximizing their business opportunities.

1.4 Importance

This application is made to help users to maintain their cars and also to help car technicians finding jobs, linking users to a wide network of car maintenance centers with services that reach them without tiredness or effort.

1.5 Report Organization

This report is organized as the following, Chapter 1: abstract of our project, then in Chapter 2: an introduction and overview of the importance and scope of this project. In Chapter 3: the problems and constraints that we faced in this project, Chapter 4: literature review, Chapter 5: the methodology used to complete the project. In Chapter 6:results and discussion. Chapter 7: conclusion, Chapter 8:references

Chapter 2: Constraints and Earlier Coursework

2.1 Constraints

2.1.1 Programming language skill:

Some of the programming languages we used required knowledge so we had to learn it like Dart and mobile frameworks like flutter because this was the first time we had built a whole mobile app.We were able to solve this issue because the syntax looks similar to C, but the primary problem was the new background. The Node JS language that we also had to learn and deal with is based on JavaScript as well as Firebase database,which is a cloud-hosted database, to be used in some sub-properties such as chat and notification , as well as the MongoDB database, which is used as a database.

2.1.2 Network bottleneck and Internet connection:

Our application requires a fast and stable internet and connection to the Node JS server to run the API.

2.2 Earlier Coursework

There were many courses that helped us build this project, for example Web Programming, Object Oriented Programming, Database Course and software engineering course.In addition, we have taken online courses in Flutter, Dart and Node-js.

Chapter 3: Literature Review

Ezhalha App : Ezhalha… We fix your Car, wherever you Are! , Ezhalha is a mobile app for car maintenance , mobile workshops ,Towing Trucks ,Computer Inspection , Car Wash , Oil Change …etc .

Drivvo: is another app that lets you track fuel to help save money on gas, as well service and registration costs in one place. It's convenient for personal and professional drivers.

The Auto Care 1: app registers distance in miles or kilometers and fuel in gallons or liters. Users can set maintenance reminders based on mileage or time of the month.

CARFAX Car Care : you can track service on up to eight vehicles, making it a great choice for extended families or anyone managing a small motor pool. It provides service reminders based on your specific car model and will quickly alert you to any safety recalls. One user reported they loved the app because it "automatically uploads when you have service done.

While there is several car apps in the market , our app stand out with the following features:

- **Comprehensive solution** : our app includes all needs for car maintenance needs in one place
- **Geolocation and map**: our app utilizes geolocation technology to identify the user's location and provide them with nearby service providers.

- **Real time communication** : Through an integrated chat functionality, our software encourages easy communication between users and service providers.

Chapter 4: Methodology

4.1 Tools, Programming Languages and APIs Technologies:

4.1.1 Front-end:

- **User Interface:** We designed a simple, user-friendly application, so that the user can understand the application and switch between the interfaces simply. We also selected a color for the application that is relaxing to the eye.

- **Framework:** We chose the flutter framework to build our application.Flutter is Google's UI framework for crafting high-quality native interfaces on iOS, Android, web, and desktop. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source.

- **Programming language:** Flutter uses Dart which is a programming language developed by Google.

4.1.2 Back-end:

- **Framework:** For the server side, we chose the open-source Node.JS frame-work for mobile app and website.Node.js is a single-threaded, open-source, cross-platform runtime environment for building fast and scalable server-side and networking applications.

- **Programming Language:** Node.js uses JavaScript which is a lightweight, interpreted programming language.

4.1.3 Tools:

- Visual Studio Code.
- Android Studio.
- Android Emulator.
- Version control system using Git and GitHub.

4.1.4 API Endpoints Testing:

Because our application has a RESTful API in the back-end server, we need to test each endpoint before integrating it with the client. To do that we used Postman.
An excellent tool for examining RESTful APIs is Postman. We can test POST, GET, PATCH, DELETE, and other request types with Postman.

4.2 Database :

The database is essential for the majority of applications since it provides reading, updating, and deleting data and allows us to back up data.Because the information shown by our application depends directly on the database, we chose a NoSQL database because there were few relationships between the various collections and because it was simple to use.

NoSQL databases come in a variety of type.we used MongoDB which is an open source document database that uses a flexible schema to store data and is built on a horizontal scale-out architecture. It stores data in a type of JSON format called BSON.

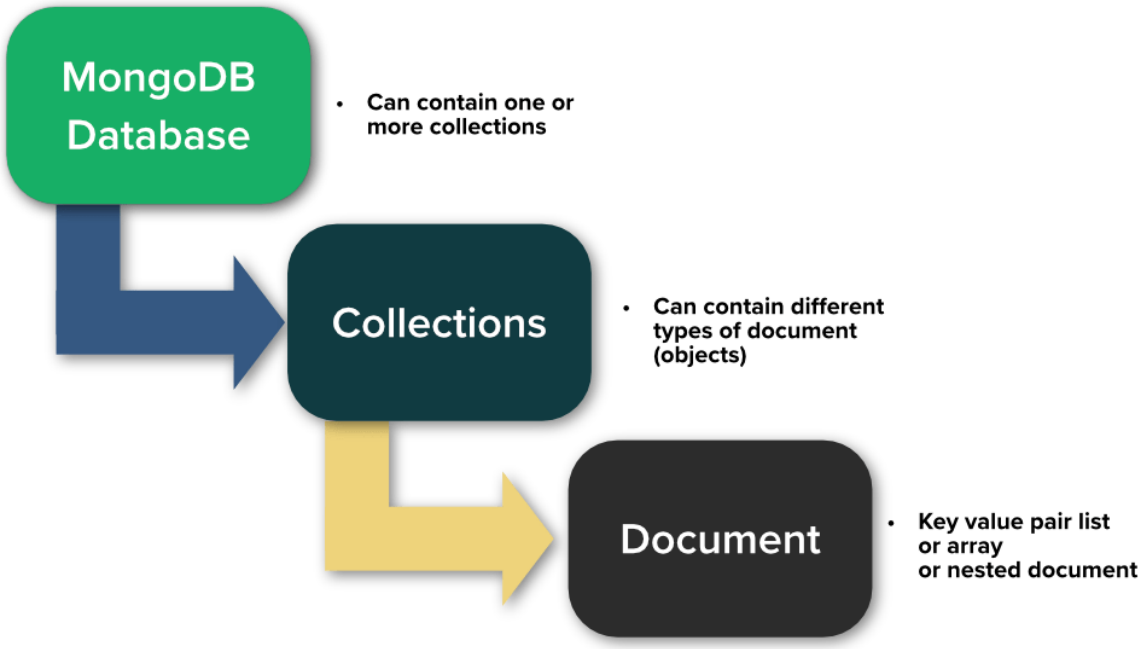


Figure 1.MongoDB

Example:some documents for product collection as shown in figure 2 and figure 3.


```
_id: ObjectId('646a029133f0a768eff283e5')
name: "Alternator"
brand: "Ford"
type: "Electrical"
price: 160
quantity: 55
image: BinData(0, 'iVBORw0KGgoAAAANSUhEUgAAAPoAAAD6CAIAAAAHjs1qAAAACXBIWXMAAA7DAAAOwwHHb6hkA…
__v: 0
```


Figure 2


```
_id: ObjectId('646a02ba33f0a768eff283e8')
name: "Battery"
brand: "Ford"
type: "Electrical"
price: 90
quantity: 30
image: BinData(0, 'iVBORw0KGgoAAAANSUhEUgAAAPoAAAD6CAIAAAAHjs1qAAAACXBIWXMAAAsTAAALEwEAmpwYA…
__v: 0
```


Figure 3

4.3 Features and Components of the Application:

- **Geolocation and map**: our app utilizes geolocation technology to identify the user's location and provide them with nearby service providers.

- **User management** :User profiles, settings, are handled by the user administration component. It enables users to establish and manage accounts, as well as update personal information.

- **Real time communication** : Through an integrated chat functionality, our software encourages easy communication between users and service providers.

- **Service Provider Management:** The purpose of this component is to manage service providers within the application. It provides features like managing their profiles, tracking their availability,managing their orders(Calendar), and allowing users and service providers to communicate .

- **Service Booking** : This feature allows customers to explore and pick from a variety of automobile maintenance services offered through the app. It has elements such as service descriptions and pricing information.

- **Order Tracking:** Once the order is accepted, the app will provide features for order tracking, allowing the client to monitor the progress of the service and stay updated on the status of their order. This can include real-time updates, notifications, remaining time and delivery date.

- **Payment Integration:** The payment integration component allows users to make secure and convenient payments within the app. It comprises connectivity with payment gateways or third-party payment providers to execute transactions securely and quickly. Users can select payment options, enter payment information, and receive confirmation of successful payments.

- **Online Shopping Feature:**The online shopping feature of our car maintenance app has provided users with a convenient way to purchase car parts and accessories. With the integration of the online store, users can browse through a wide range of products and make purchases directly through the app.

- **Order Management:** From placement to success, the order management component manages the lifecycle of user orders. It offers features such as order monitoring, status updates, notifications, and order history to allow users to track the progress of their service requests.

4.4 Implementation

4.4.1 Authentication Interfaces:

- **Login:** the login page of the application wo the primary entry point for application users , to allow them access the app's features.
  login to the application is done by entering the email and password.Also the page provided with effective error handling , showing informative error messages in case of incorrect login credentials or other issues .*Figure 4*

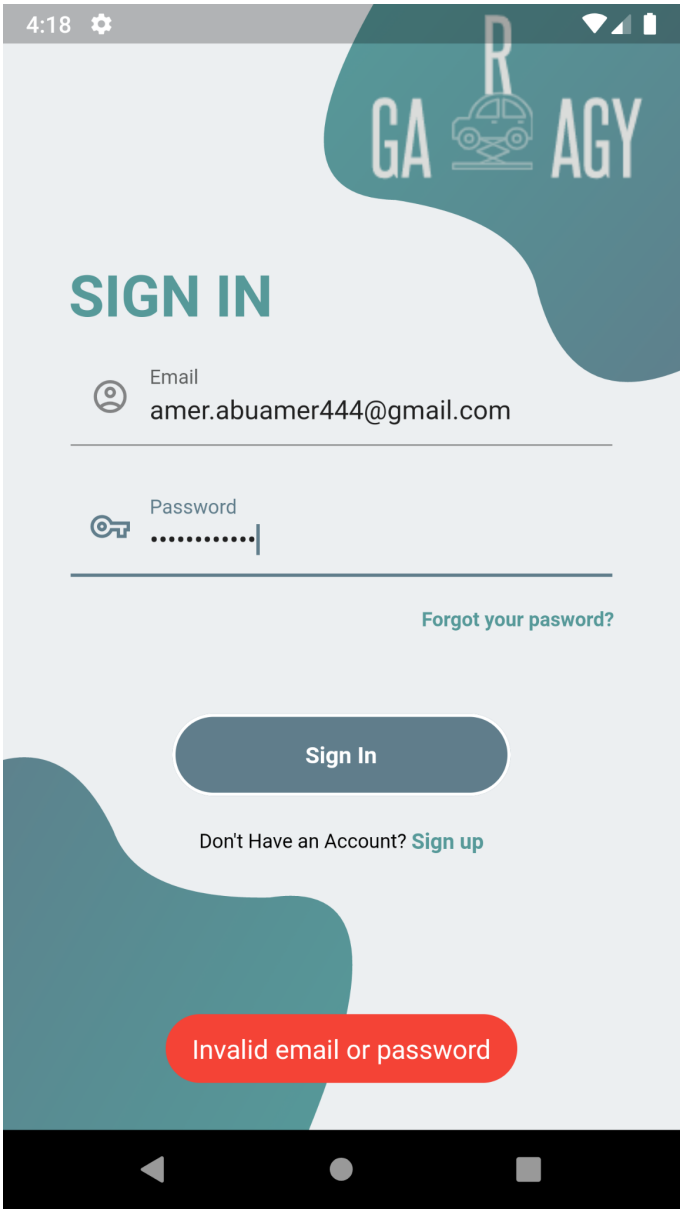


Figure 4.Login Page

- **Sign up:** the sign up page design to handle the registration process for new users , users could create their accounts and gain access to the app's features and services . Users have to enter their personal information , including first name , last name , email address ,password ,phone number ,car model which is a drop down list fetched from the server to ensure the user gets all new car models added to the system. Note : all fields are required , password must be at leasts 8 characters.

Figure 5

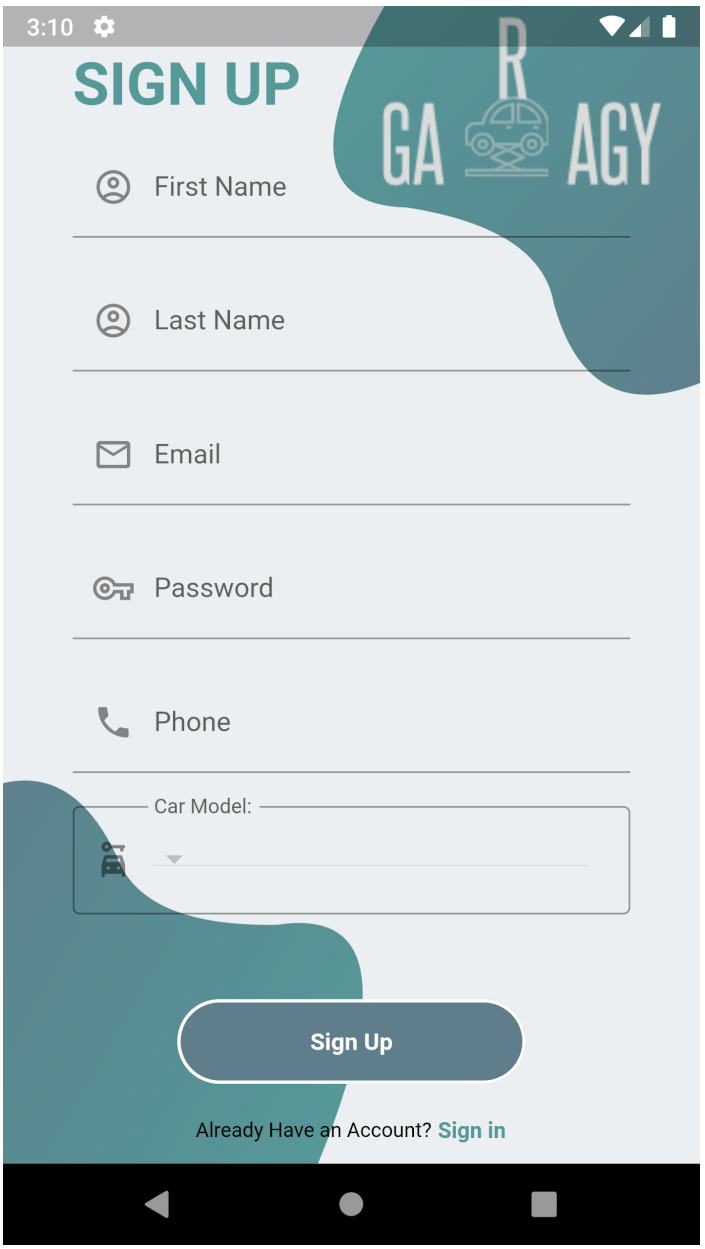


Figure 5.Sign up page

- **Forget password:** If you forget the password, you can reset it by checking the email by sending an email with a link to reset your password as shown in *Figure 6.*

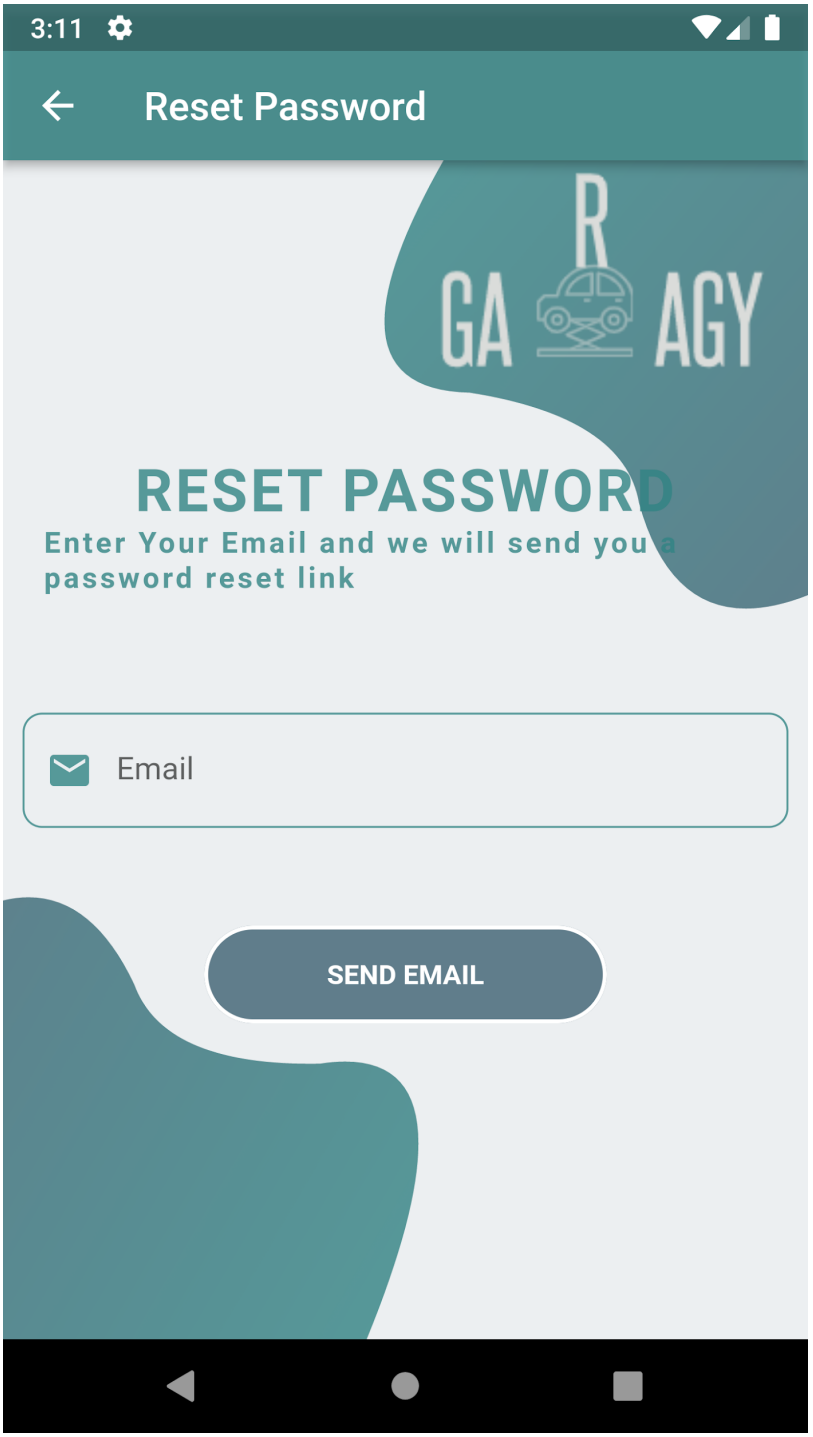


Figure 6.Forget password

- **About page:** this page provides information about the application and how to contact us, as shown in Figure 7.

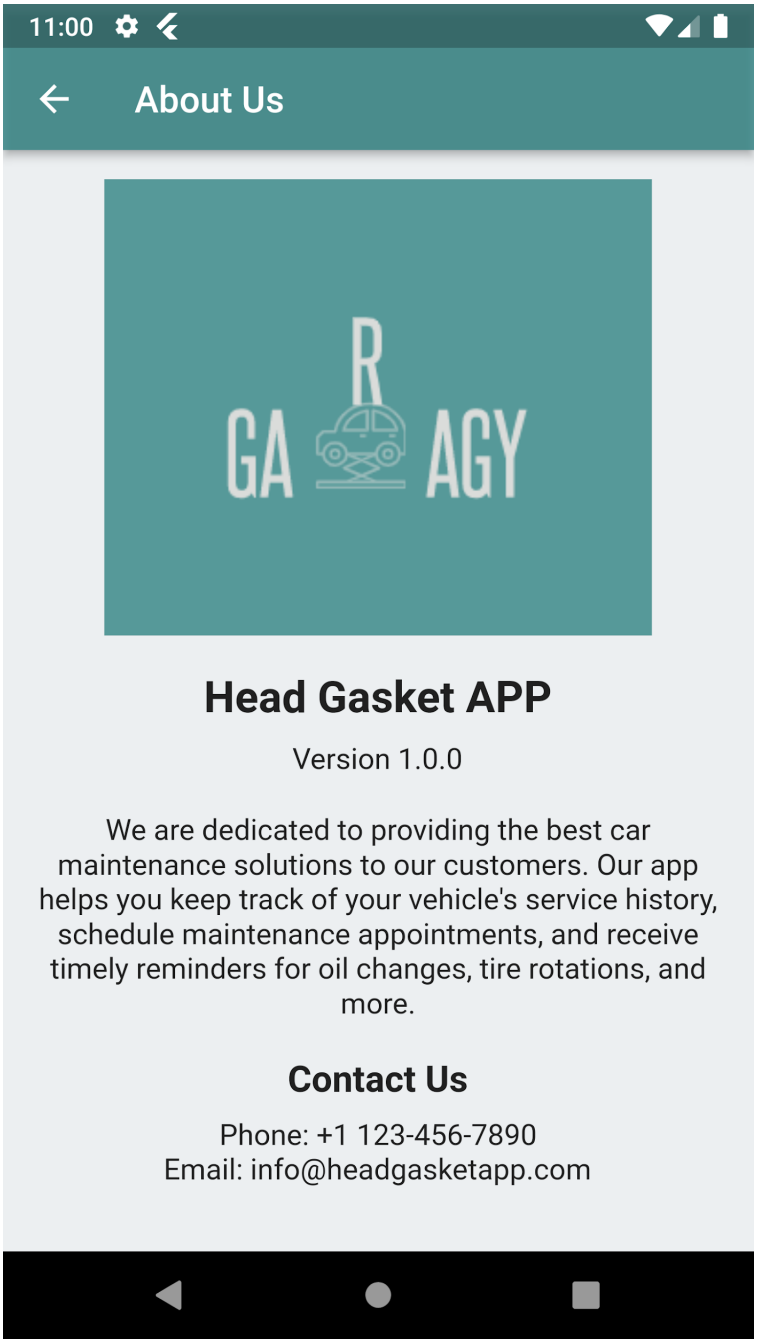


Figure 7.About page

4.4.2 Client Interfaces:

- **Home Page:**

  The client home page is designed to provide a user-friendly interface , offering a range of useful features and easy navigation . It is the central hub where clients can access various functionalities and explore sections of our app.
  The home page shows the profile picture ,name and car model ,also a list of random services from the system and the top workers in our family according to clients rating .
  In addition , a bottom bar is integrated into the home page , providing quick access to main sections of the app , including car services , app's store and my orders.*Figure 8*

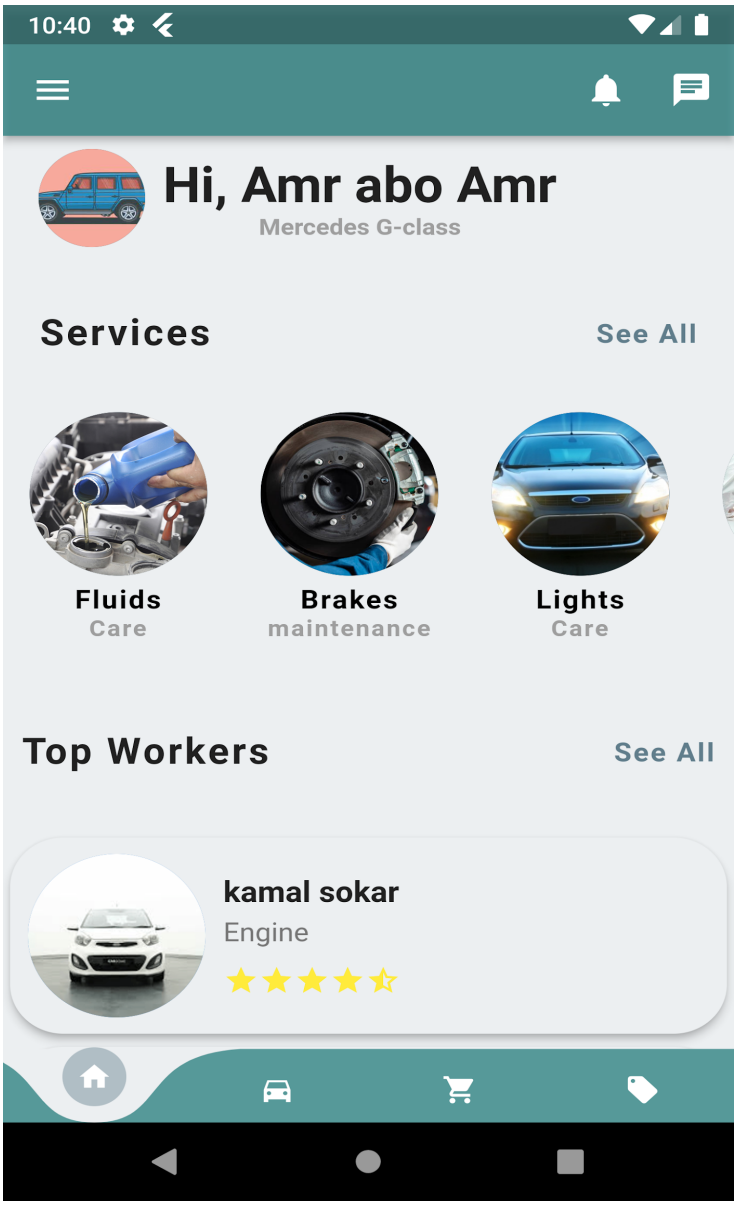


Figure 8.Home Page

A sidebar menu is provided on the home page , this menu includes options such as the client's profile , change passwords , join as a service provider , access information about the app , and logout .*Figure 9*

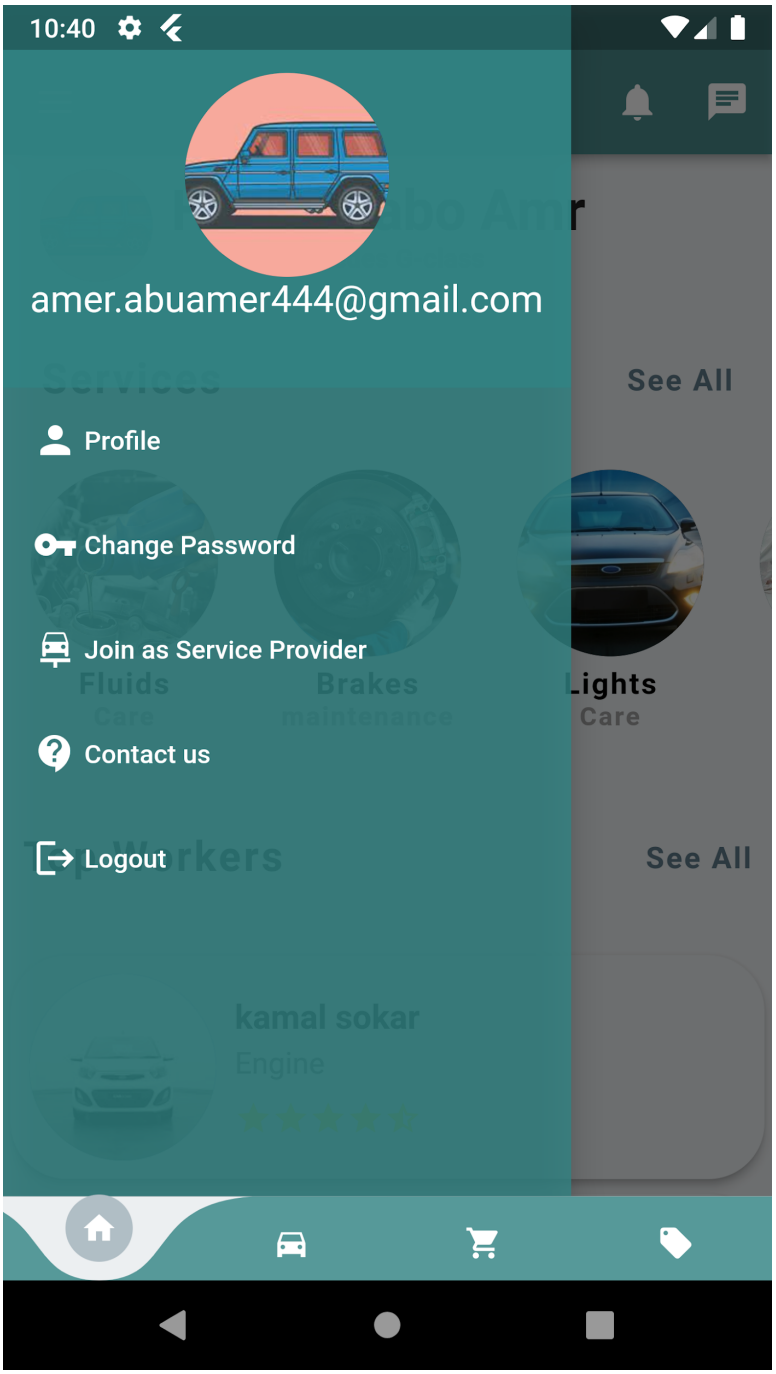


Figure 9.sidebar

- **Profile page:**
  In this page we display information about the user such as profile picture , car model , name , phone , location and a list to show the service history which is a list of maintenance or services completed on the car.*Figure 10*

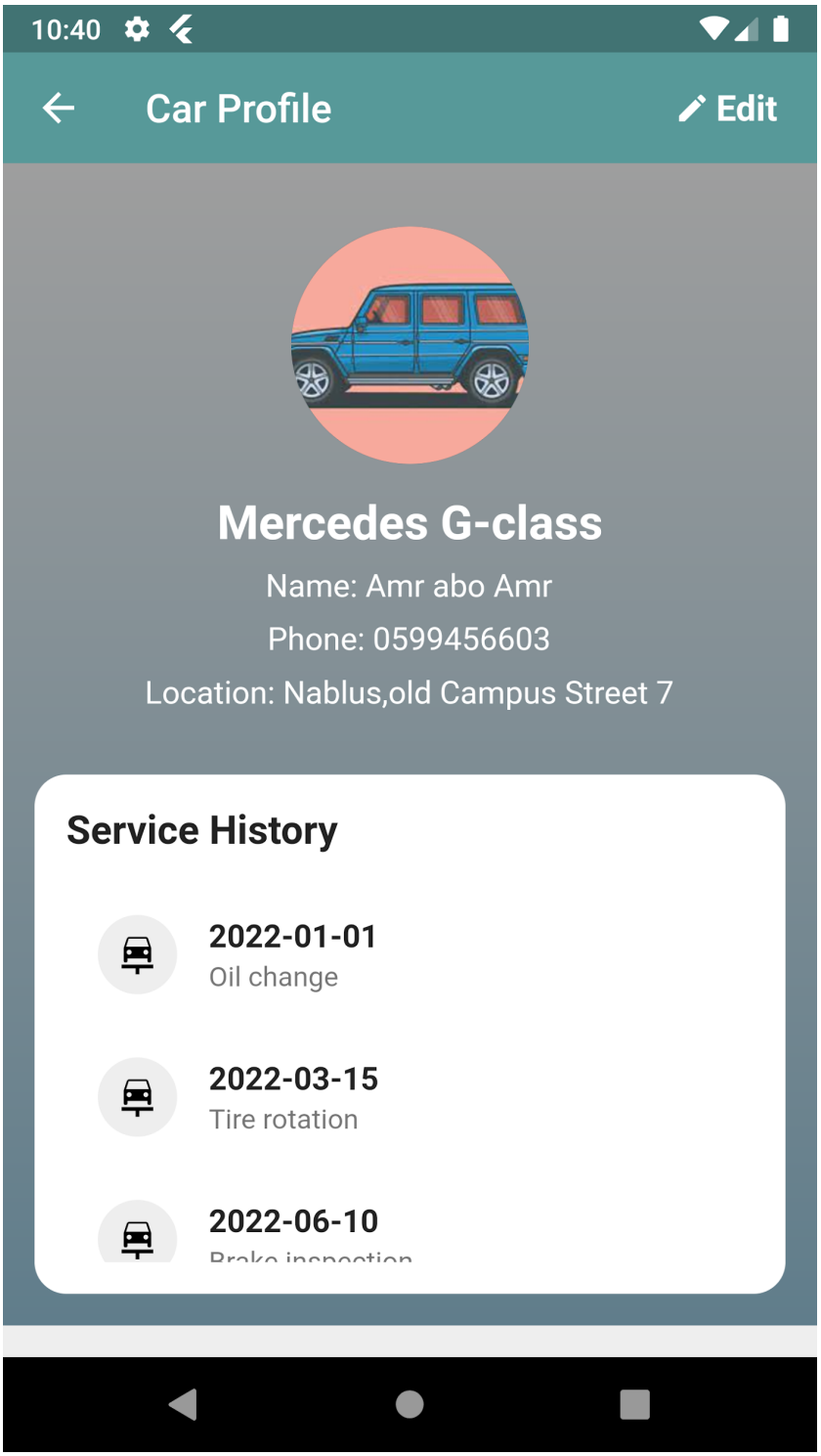


Figure 10.Profile page

- **Edit Profile page:**
  On the edit profile page we allow users to update and modify some information such as first name , last name , car model , phone number .And in this page a user could upload or update his profile picture .
  The field will have an initial value with the user data stored in the server .*Figure 11*

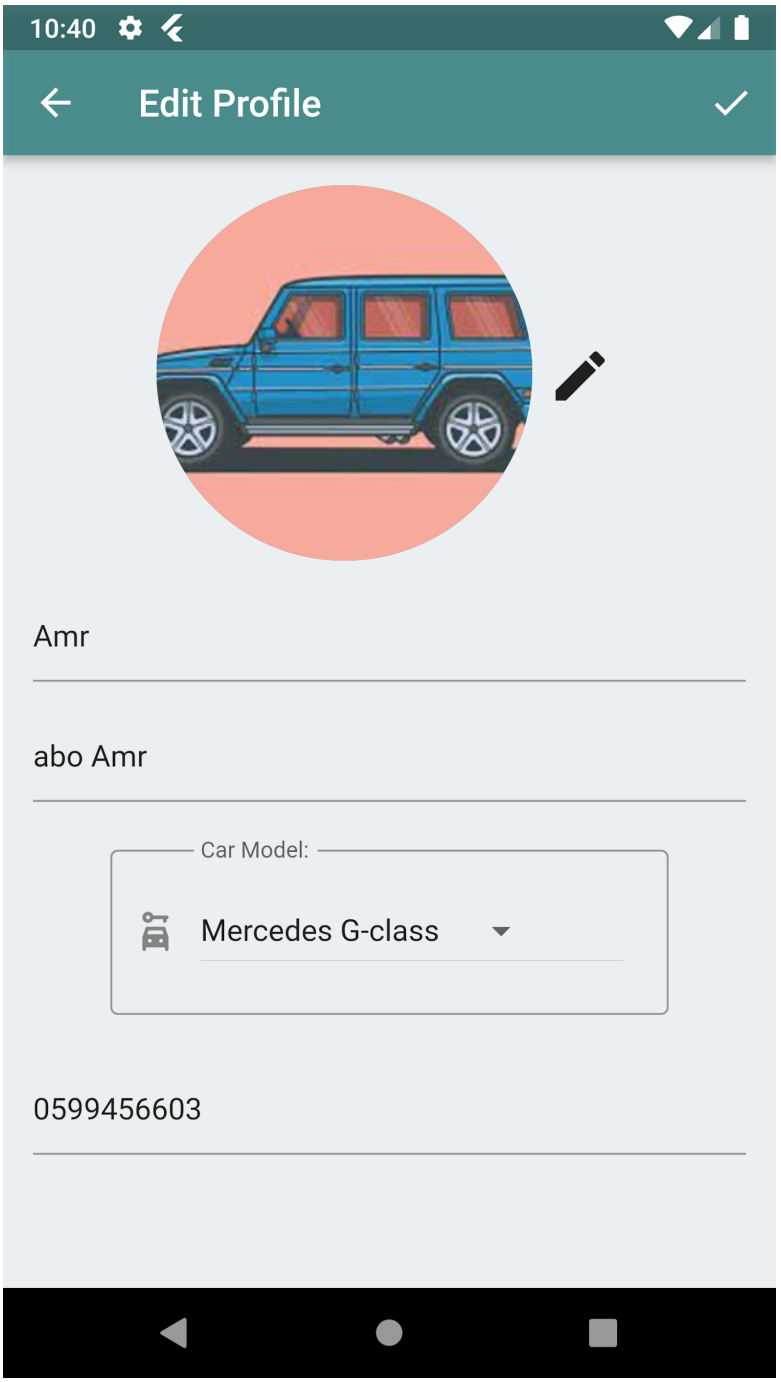


Figure 11.Edit Profile page

- **Change password page:**
  if user want to change his password due to any security reasons he could use this page to enter current password , new password , and confirmed new password
  Note : all fields are required , new and confirm passwords must be the same (we check that before sending the request).
  If any error or issue happens a toast message will appear for the user.*Figure 12*

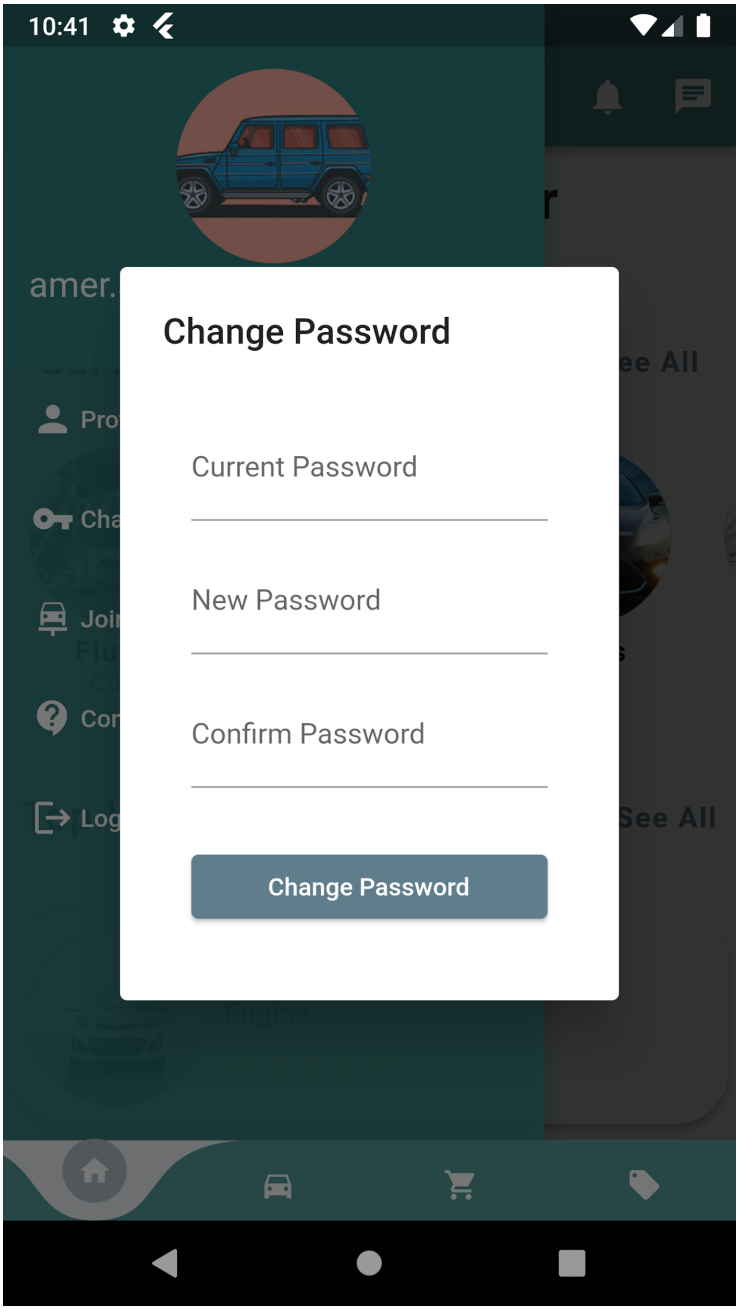


Figure 12.Change password page

- **Join as a service provider Page:**
  This page allows individuals who could provide a car service or who is a car technician to become part of our workers. It provides a list of available services in our system, fetched from the server, to let the service provider choose their field. Additionally, they can specify the car brand they prefer to work with. Moreover, there is a text field where they can provide information about their experience, skills, and previous work. Upon joining our team, the user becomes a worker and gains the ability to receive jobs and orders from clients. Furthermore, workers are required to upload their relevant certificates and credentials to ensure the highest quality of service provision and to build trust with clients

Figure 13

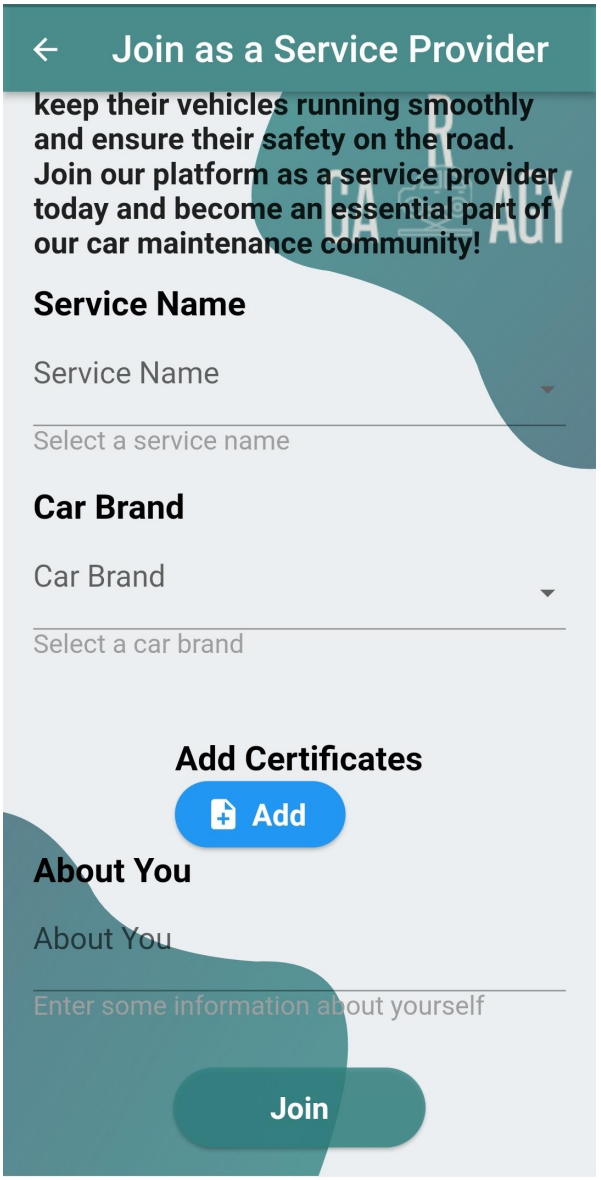


Figure 13.Join as a service provider Page

- **Car services Page:**

  In this page all available service for cars will be fetched from the server . in our system there is three types of services :
  1 - Emergency services including Battery recharge , flare tire repairing , and fuel delivery
  2- Maintenance : Brakes ,Engine Steering , chassis and exhaust …etc
  3- care : car washing , painting and polishing .
  user will choose the suitable service for his car and we will reach him as fast as we can.*Figure 14*

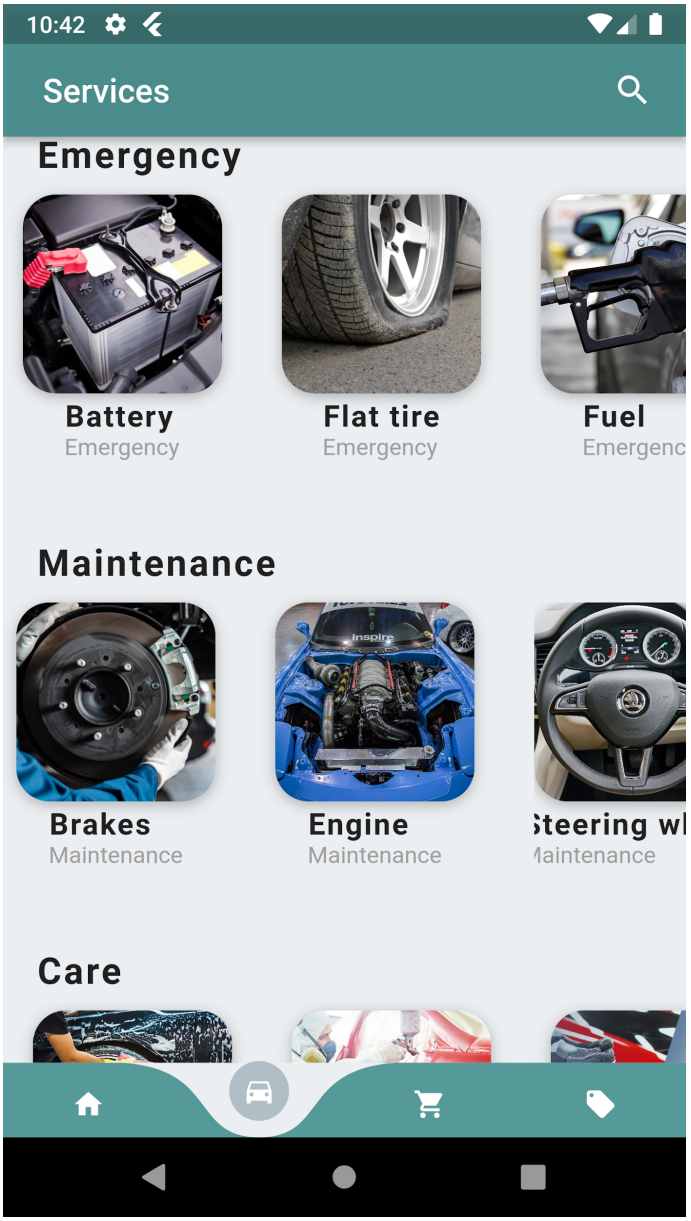


Figure 14.Car services Page

User also could search for service he want using the text field above *Figure 15*

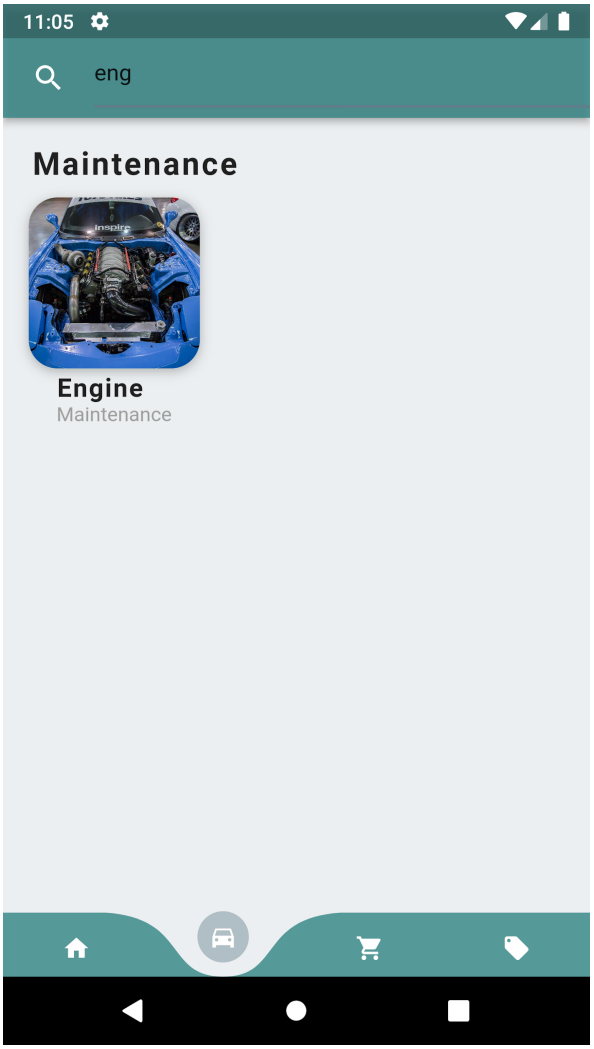


Figure 15

- **Service Page:**

  This page will appear after choosing the service user wants , this page is designed to provide users with an easy way to find service providers based on their location. It displays a list of workers or service providers sorted from nearest to the farthest based on the user's location.
  When users access this page , the app will take the user location stored in the database to calculate the distances between user and workers then sort them .

  The way used to calculate the distance:

  Convert the latitude and longitude coordinates from degrees to radians:
  lat1 = math.pi / 180.0 * worker.latitude;
  lat2 = math.pi / 180.0 * userLat;
  lng1 = math.pi / 180.0 * worker.longitude;
  lng2 = math.pi / 180.0 * userLng;

  Apply the Haversine formula on the differences to calculate the central angle between the two points:
  a = math.pow(math.sin(dlat / 2), 2) + math.cos(lat1) * math.cos(lat2) * math.pow(math.sin(dlng / 2), 2);

  then calculate the angular distance in radians:
  k = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a));

  Calculate the distance using the Earth's radius:
  distance =earth radius * k;

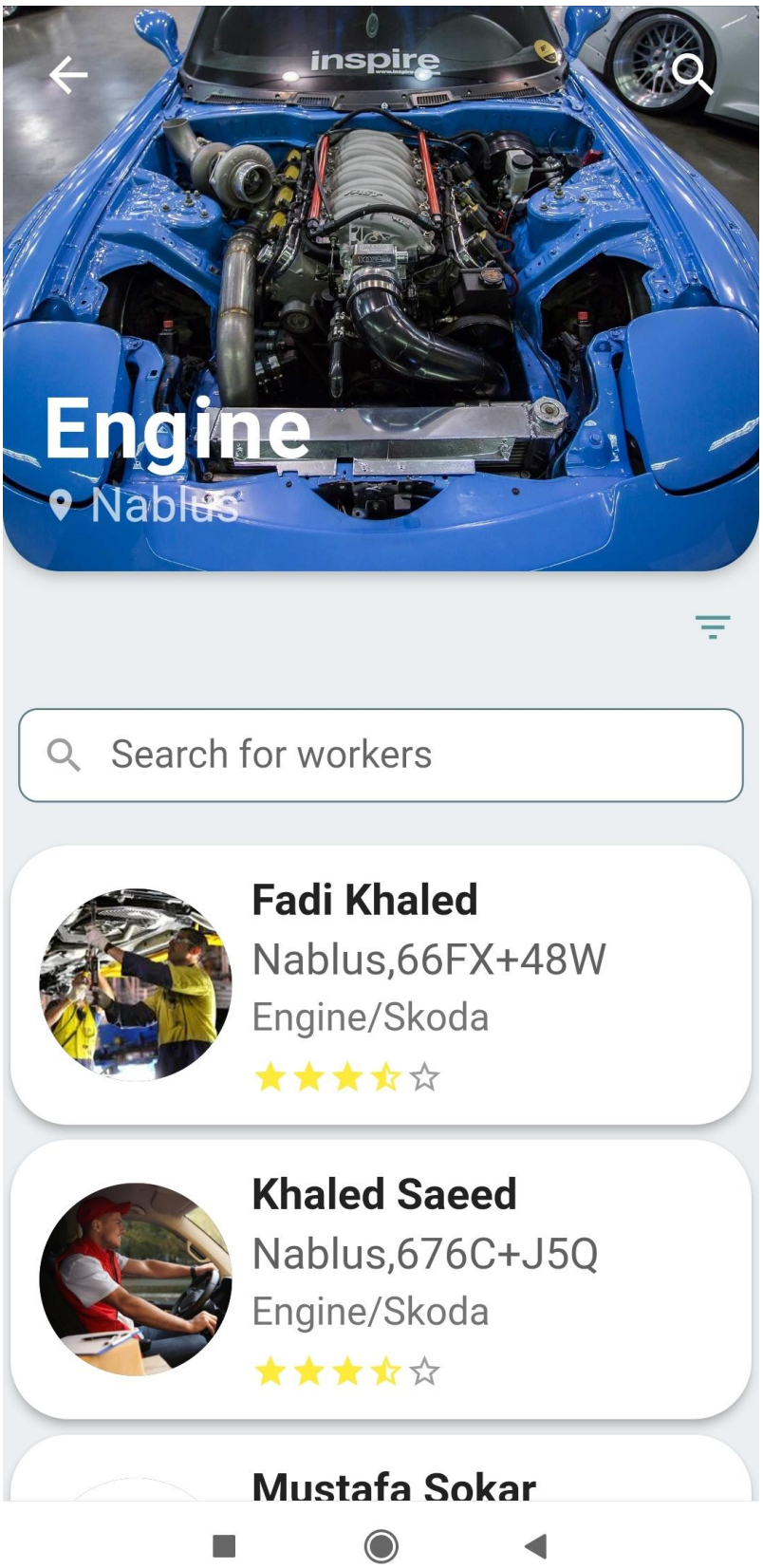


Figure 16.Service Page

- **Map Page:**

The map page provides users with the ability of choosing their current location or any desired location , also the map will display the profiles of nearby workers or service providers for easy identification .

Once the user opens the map the app will utilize the device's location to determine his current location .

Also there is a raised button to get the current location if the user chooses any other location by mistake.*Figure 17*

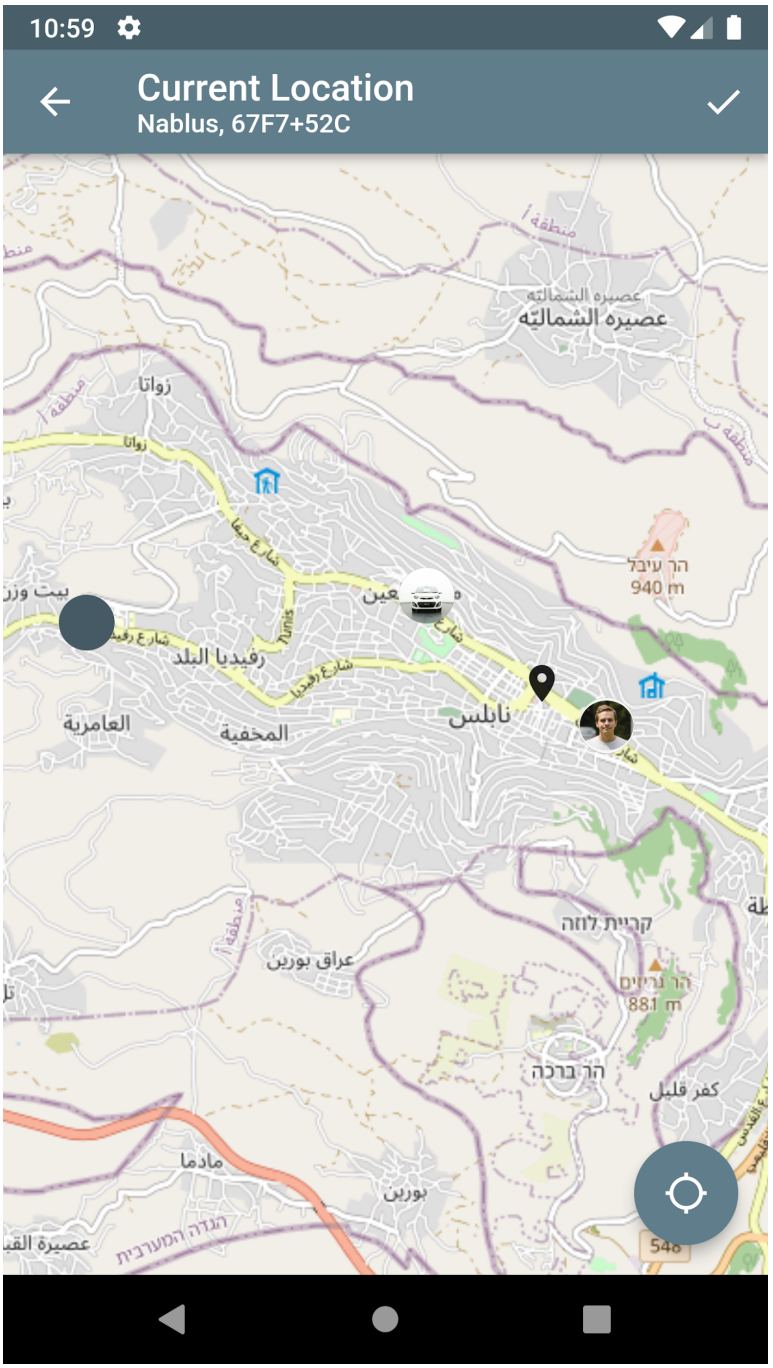


Figure 17.Map Page

- **Worker profile (from user side look):**

In this page users could discover the worker information such as name ,location , rate , services they provide ,experience , and contact including phone and email (note user could open a chat with worker we will describe this later).
Additionally , user could send a service request using the button called hire +'worker name', to send a request for this worker.

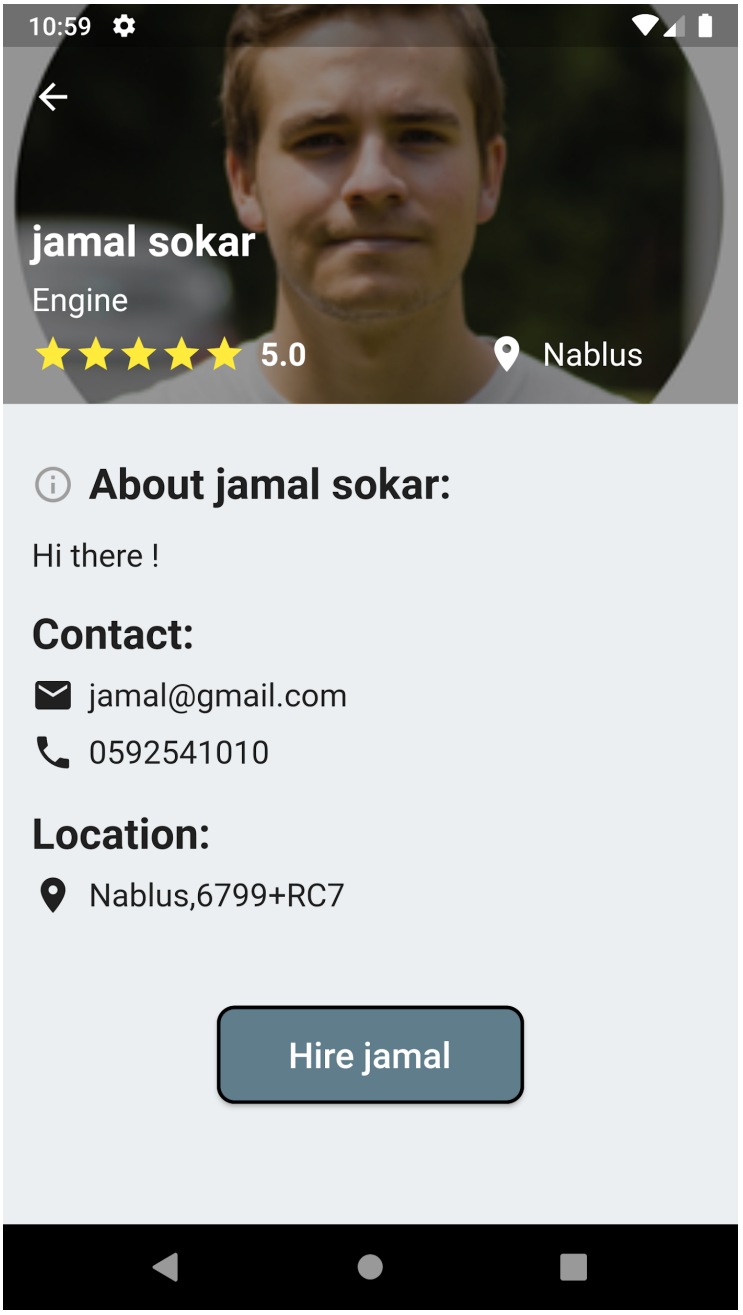
- 

Figure 18.Worker profile

When the user taps on the rate stars , a pop up message will appear to open rating functionality for the user using the 5 stars way .*Figure 19*

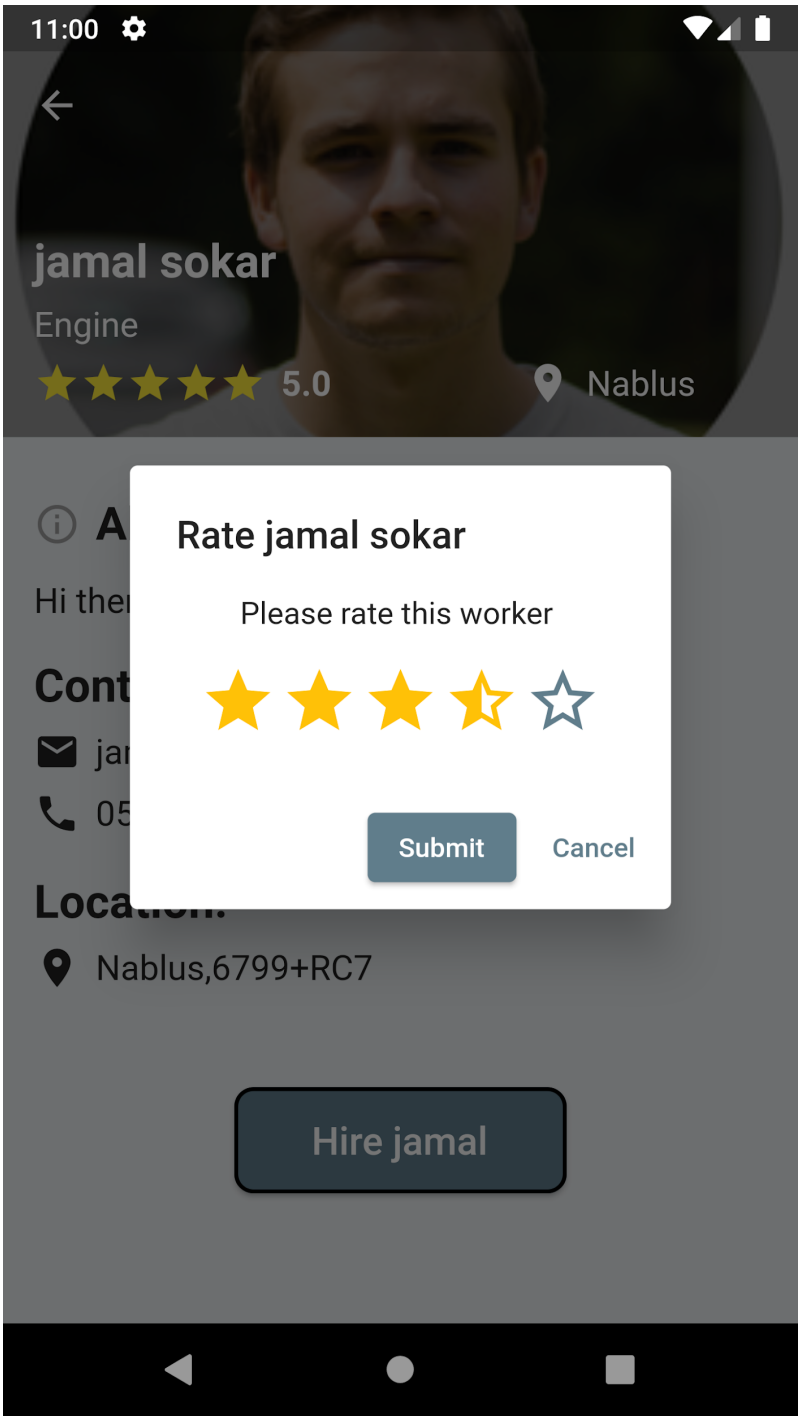


Figure 19. rate

- **Order Page:**
  This page appears when clicking on the hire button , this page will show all information that will be sent to the worker to let him know everything needs to help and reach the client , including first name , last name , email , phone , location , car model , service and note.
  In the note field we give the user a space to write anything that helps the worker .
  When a user clicks on submit a request order will be sent to the worker with all information about the service he wants (the worker then will accept or deny).*Figure 20*

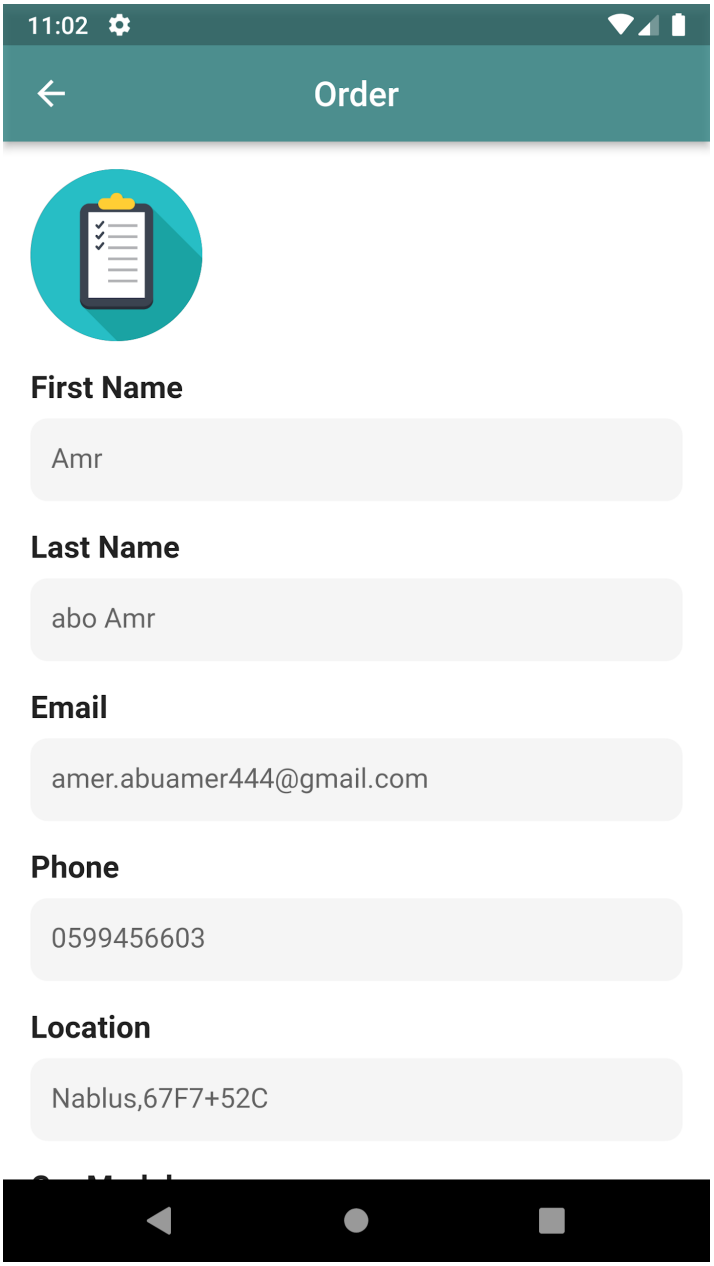


Figure 20.Order Page

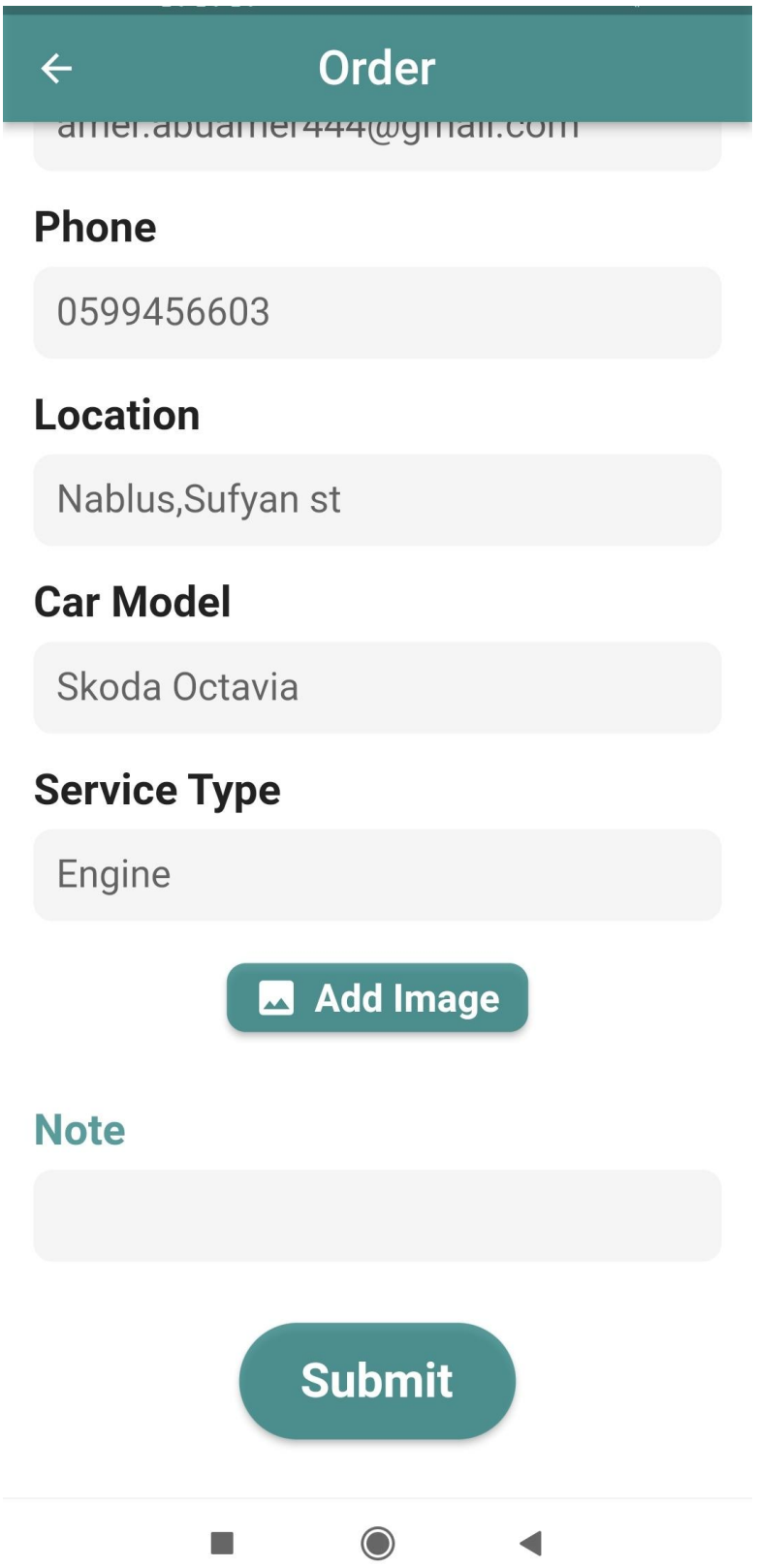

Order
Phone
0599456603
Location
Nablus,Sufyan st
Car Model
Skoda Octavia
Service Type
Engine
Add Image
Note
Submit

Figure 21

- **My orders Page:**
  In this page we provides users with a comprehensive view of their orders allows them to filter orders based on their status (Requested , waiting , processing , canceled , completed)
  In the requested section all request will appear for example when we send a request to the worker (Jamal) in the previous page the request will appear like this (waiting for jamal to accept or deny).*Figure 22,23*

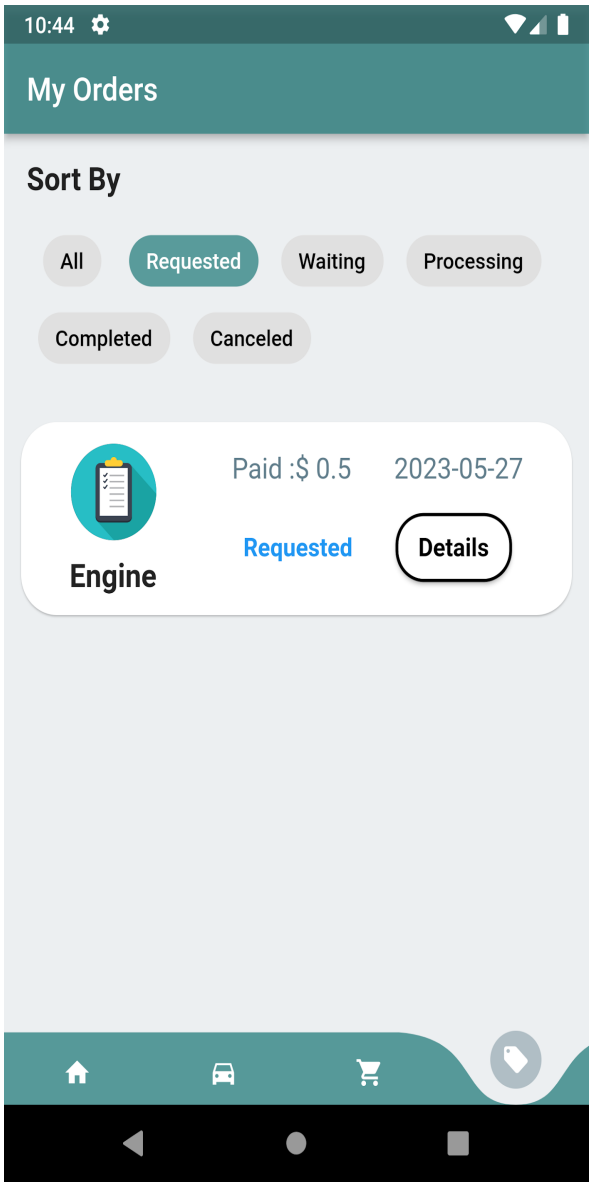


Figure 22

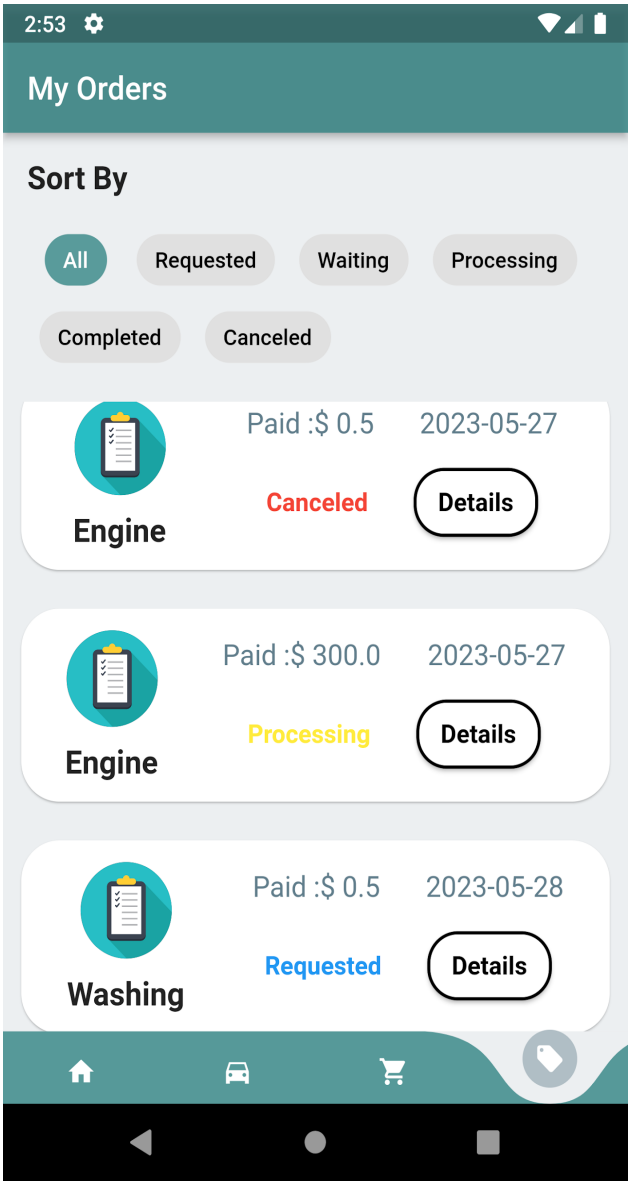


Figure 23

Importante note:
When worker accepts the order the app will **notify** the client to let him know that his order has been accepted as shown below:

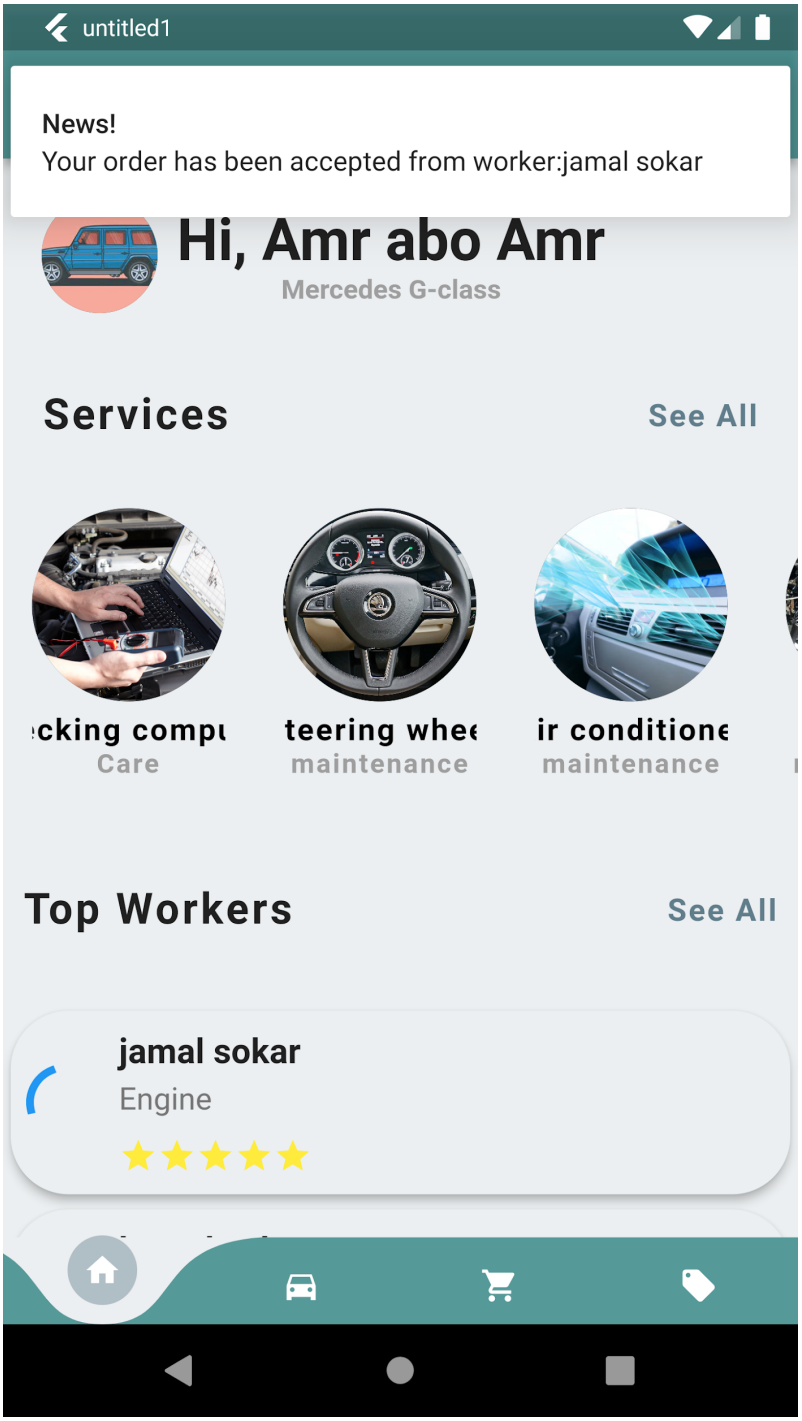


Figure 24

In this case the order status will change to ’Waiting’ (waiting for client to accept the price(if it exists) ,choose the delivery method and payment method (visa or cash on delivery)).

In the picture below shows the way order appears in waiting status
The details button will show a detailed version of the order as we will see later

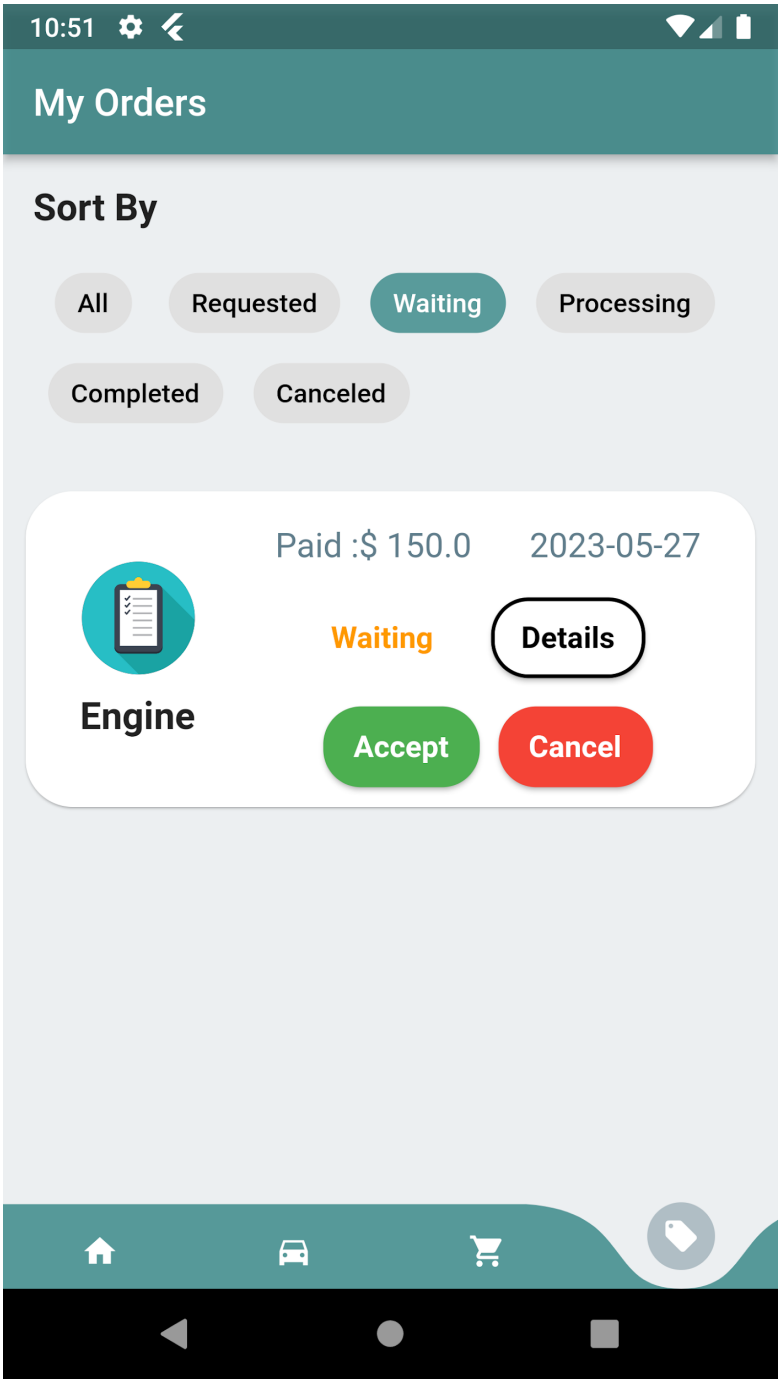


Figure 25

When the client accepts the order the Accept order page will be shown.

- **Accept order page:**
  In this page the client will review the order details and make selections for payment method , delivery method and accept the price provided by the worker. This page is a critical step in the order process.

  The delivery method : the user will choose to get a tower truck or will deliver his car by himself .

  The payment method : client will choose to pay online or on delivery.

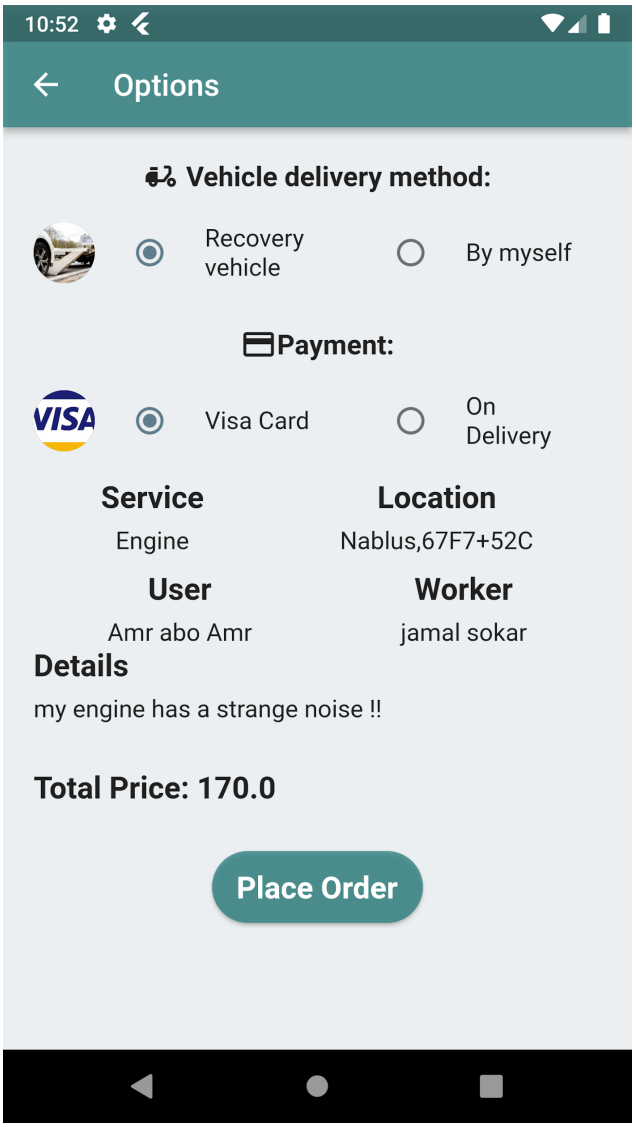


Figure 26

Once the client placed the order , a notification will be sent to the assigned worker to let him start working on the order . At this stage , the status of the order will be set to “processing” .
Once the worker completed the testing , he had the option to mark the order as “Completed ” and complete the report about the car maintenance requirments . This status signifies that the service has been successfully executed .
We will see this step in the worker pages.

- **The online payment page:**
  In this part we used Razorpay API to simulate the process of actual online payment using a visa card. This page will shown if user choose to pay online .

  Note : also we used this way for the Store .
  Here is an example of paying online for the order .

  - In this part user will choose the visa card way

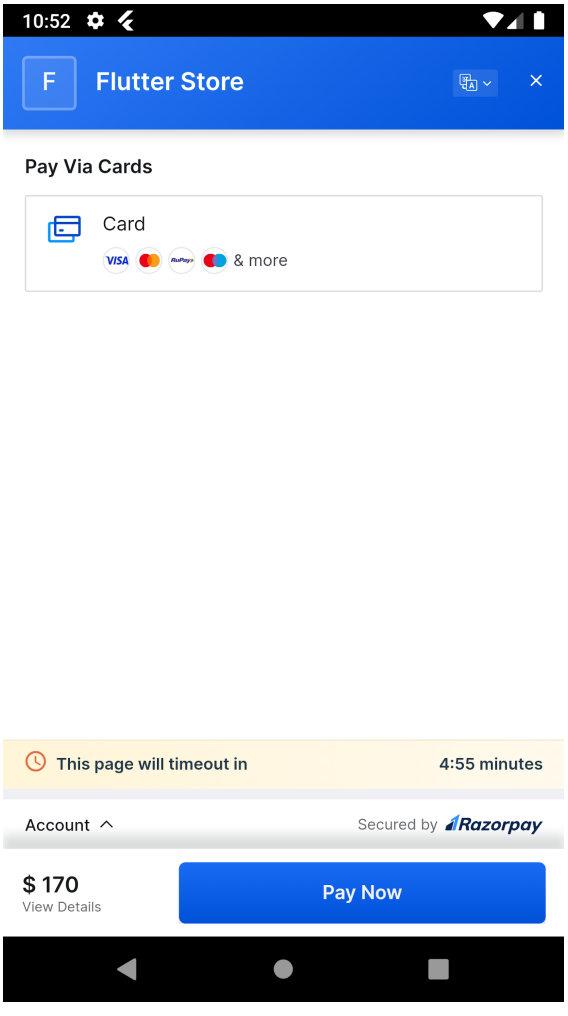


Figure 27

Then the user will enter the Card information. Including card number , expiry date , card holder name , and cvv code

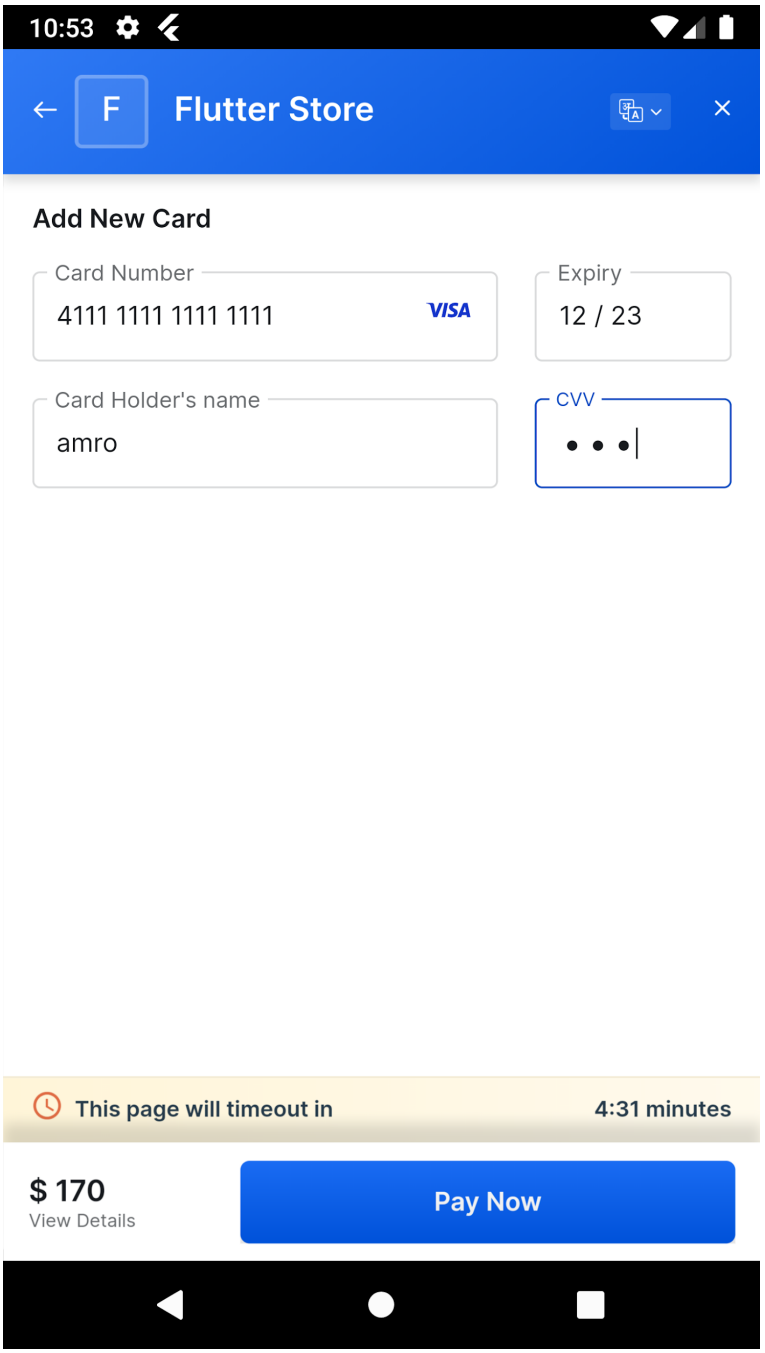


Figure 28

After clicking on the Pay now button a code will be sent to the user Phone to make sure of the user identity .
This page is the last step before determine the success or failed of the payment method .

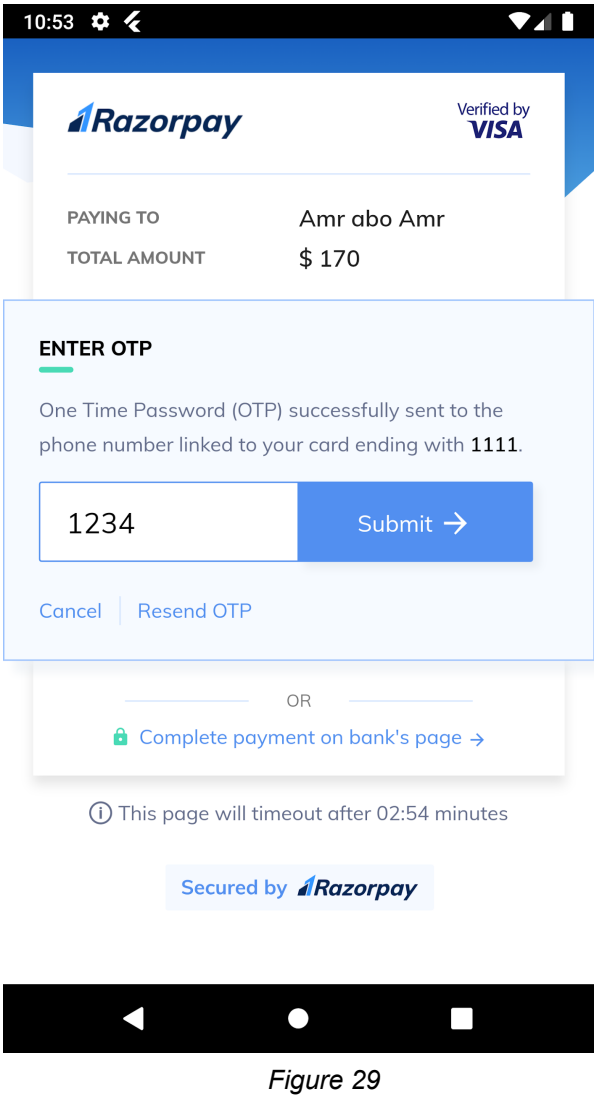


Figure 29

If everything is alright the payment will succeed .

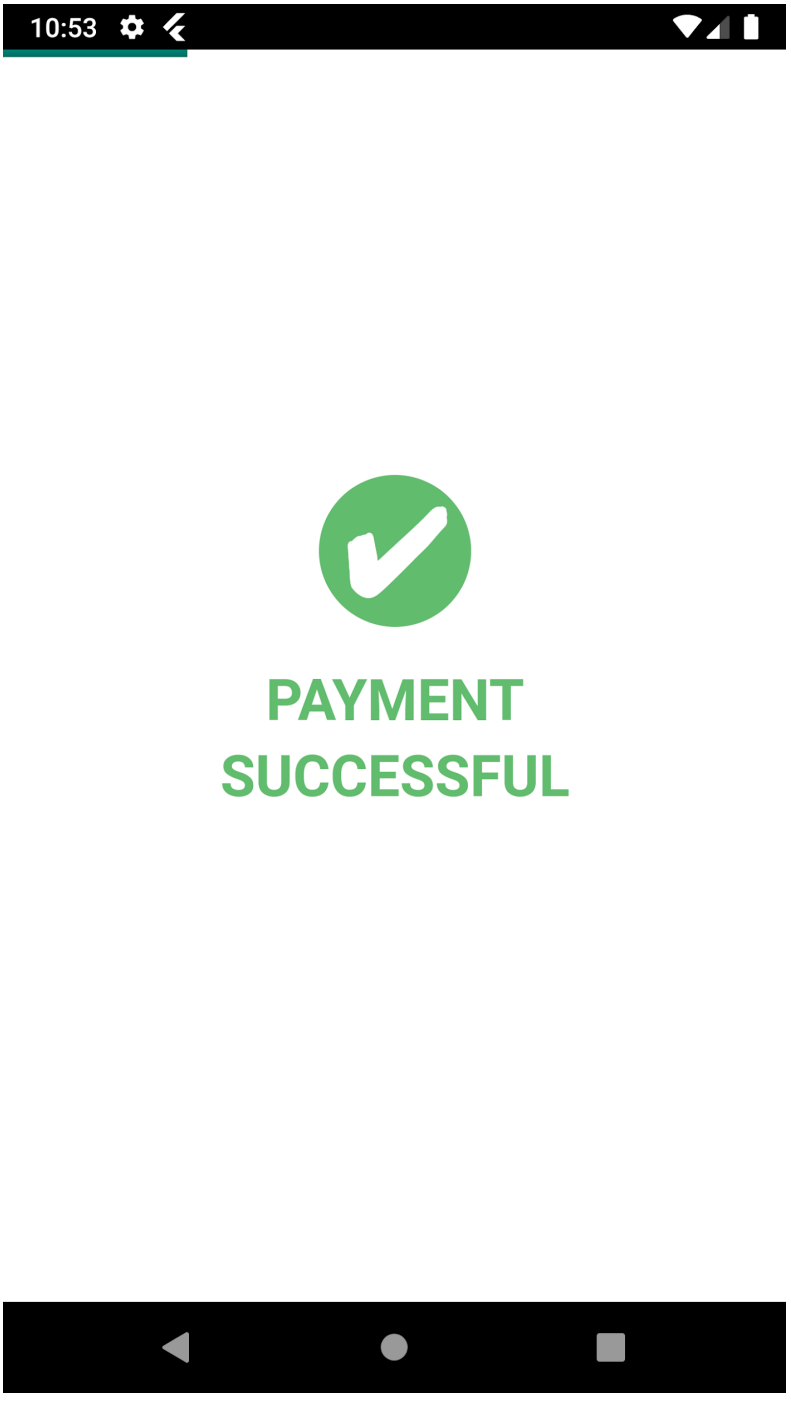


Figure 30

After payment succeed the order will placed successfully

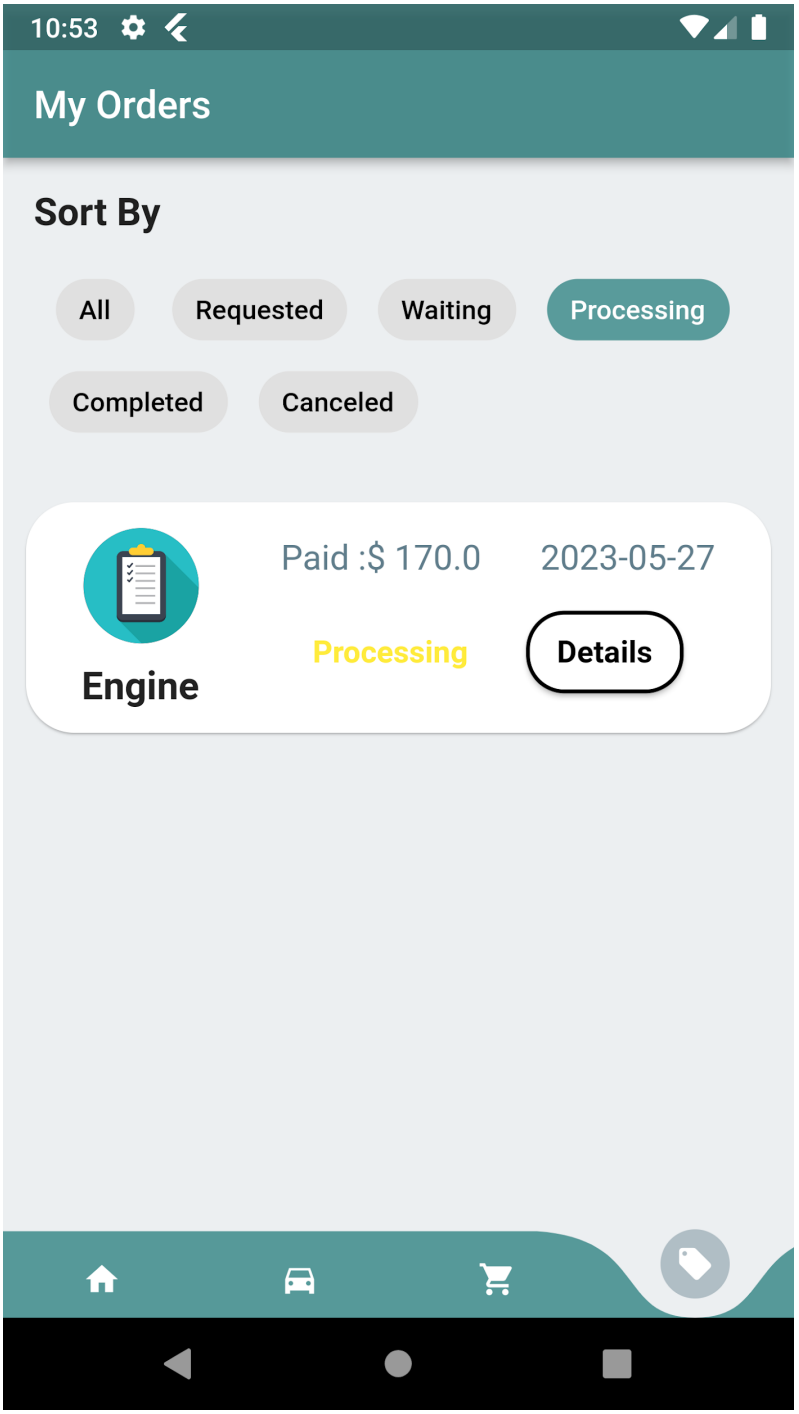


Figure 31

- **Order Details page:**
  This page provides clients with a comprehensive overview of their current order . this page contains all information about the order Such as service name , date of the order , price , client name , service provider name ,delivery method , payment method , location , status ,and any other details client provides.
  This page helps the client to take a summary of the order and monitor any changes happen on it.

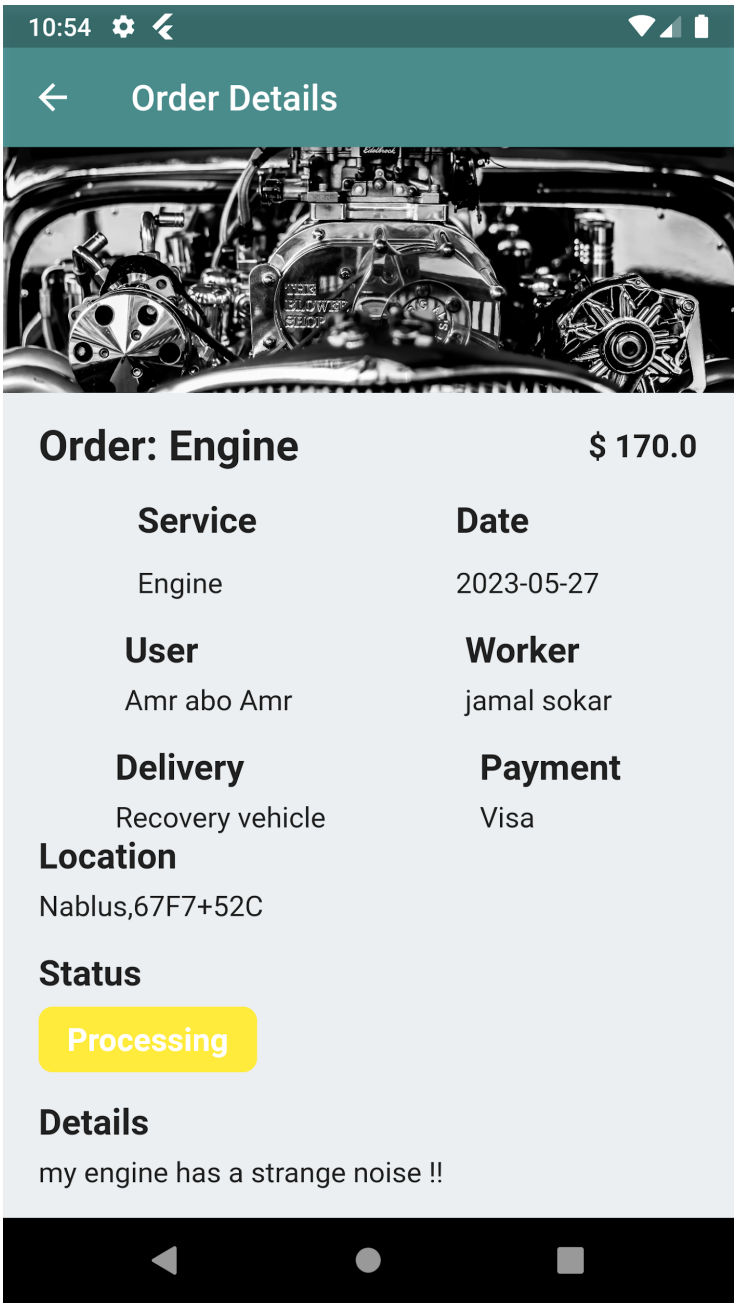


Figure 32.Order Details page

Note : when order completed client will notified as shown

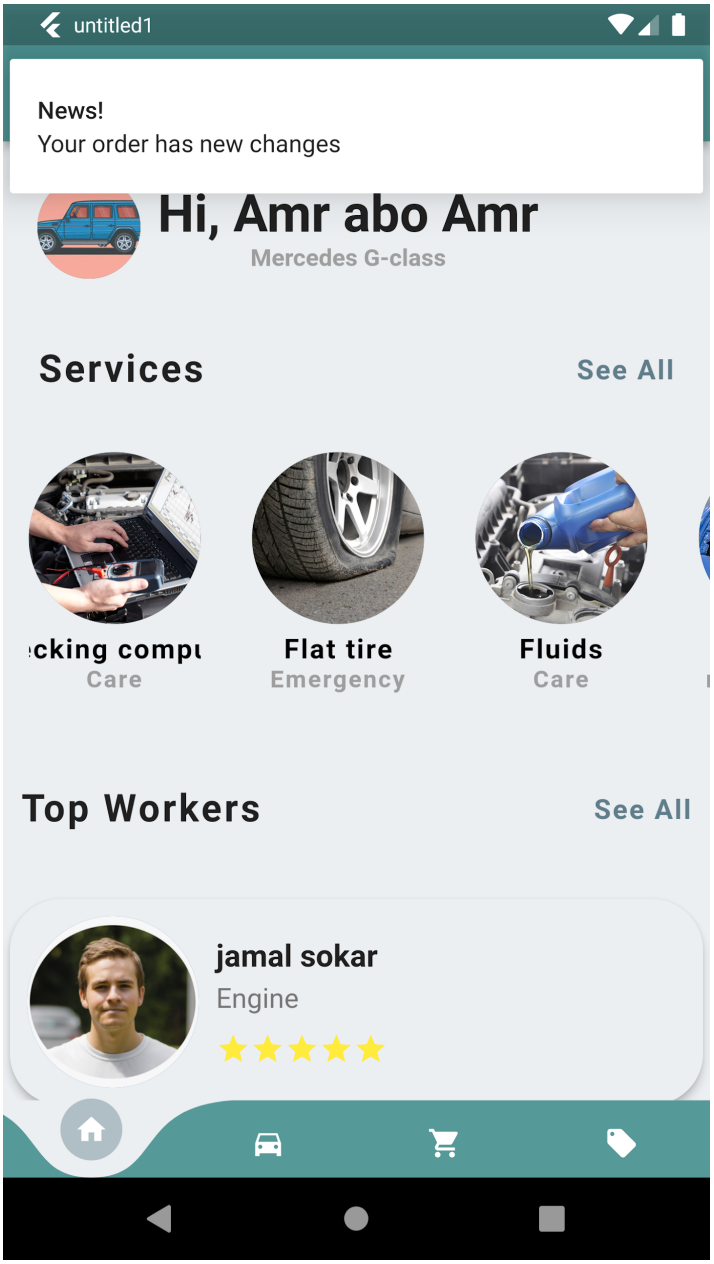
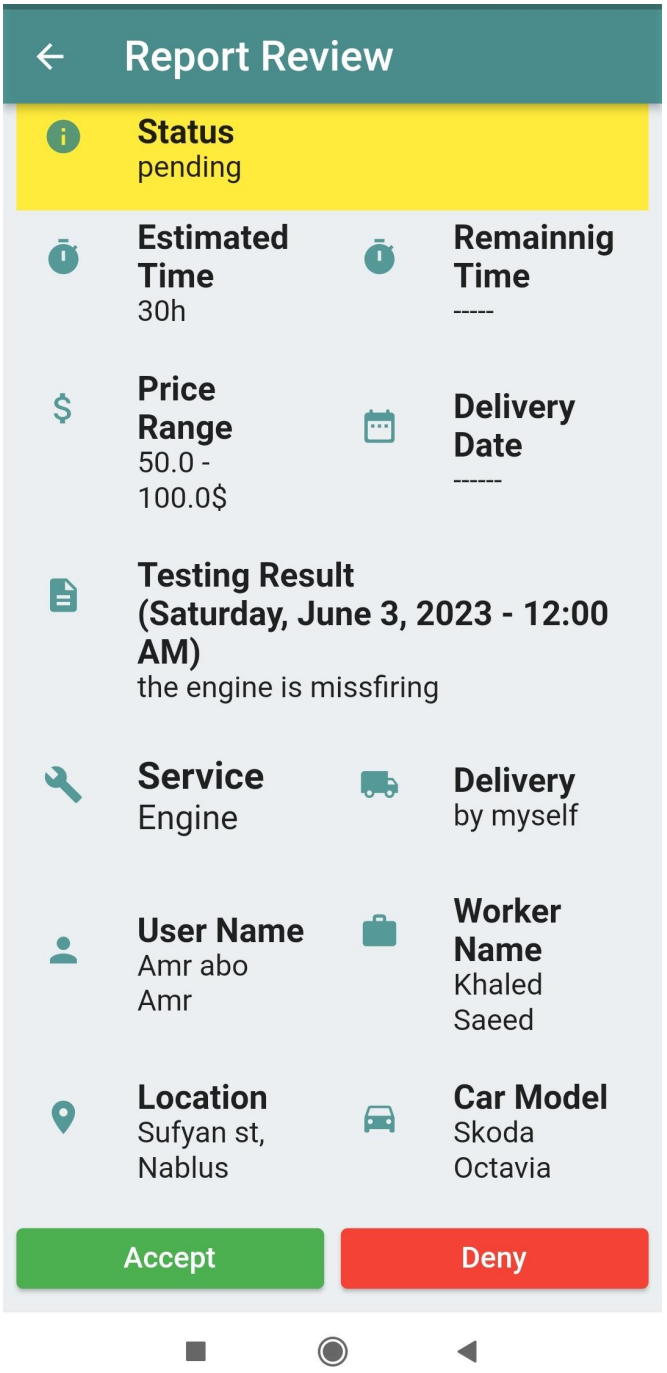


Figure 33

- **Car Report page:**
  This page will appear after finishing the car test showing the test result , estimated time , and price range



- **StorePage:**

  The store page in our app offers clients a convenient platform to browse and purchase a wide range of car accessories , parts and anything needed for clients or workers .
  This page provides the filtration functionality to let users choose the categories he want and also choose the brand of the product

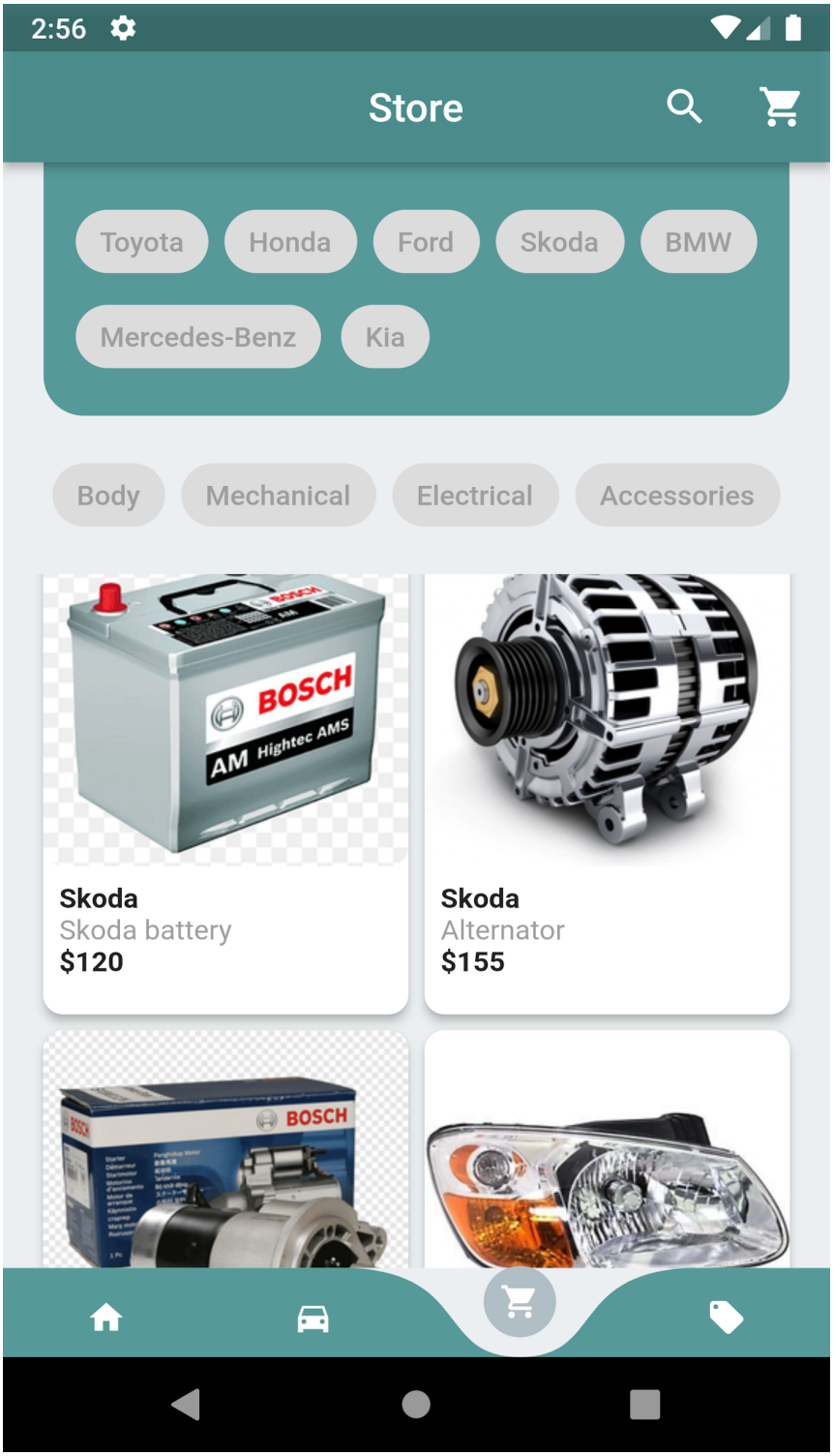


Figure 34.StorePage

Users could also search (by name , brand , type) and filter the products

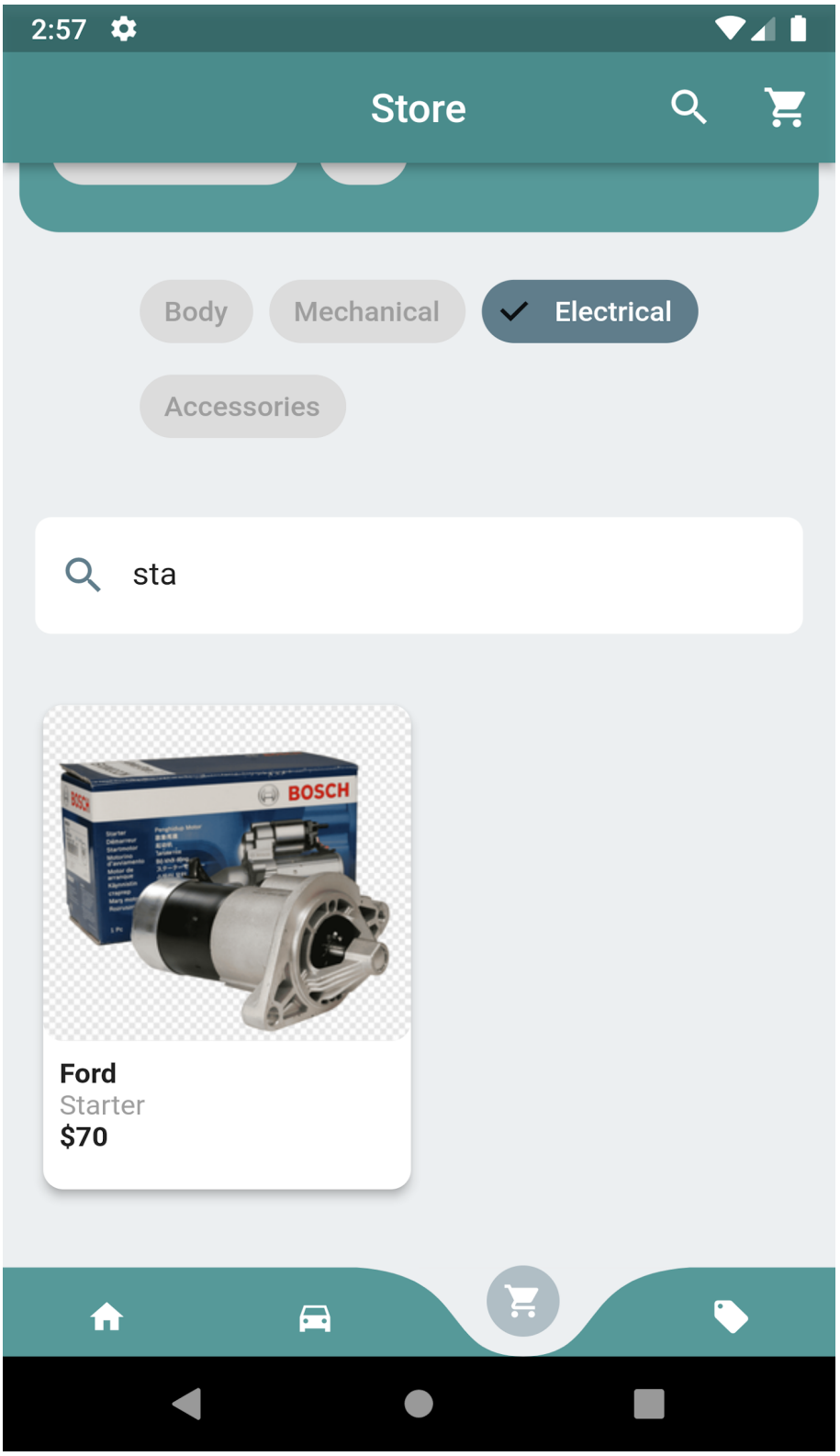


Figure 35

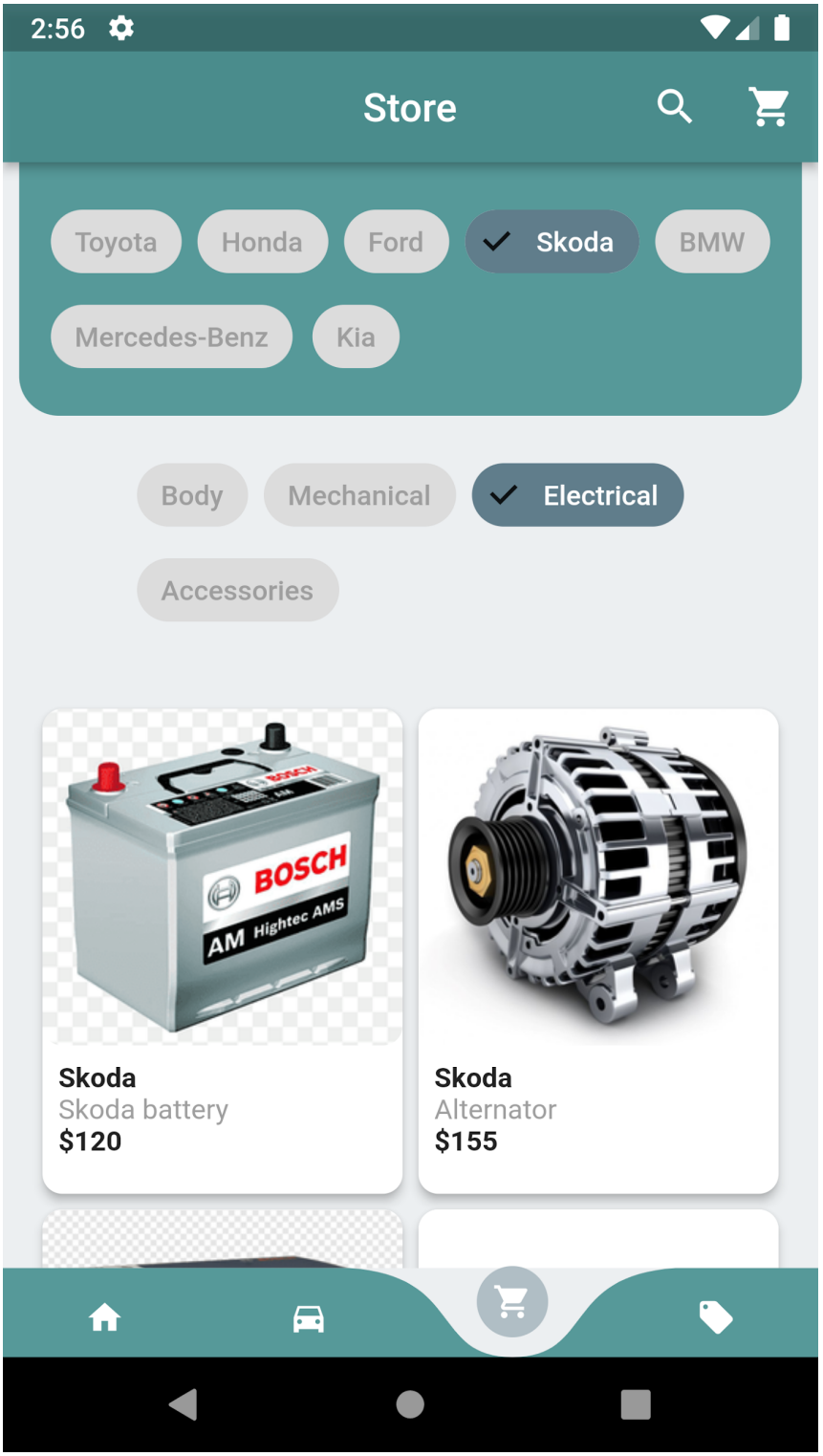

2:56
Store
Toyota
Honda
Ford
Skoda
BMW
Mercedes-Benz
Kia
Body
Mechanical
Electrical
Accessories
BOSCH
AM Hightec AMS
Skoda
Skoda battery
$120
Skoda
Alternator
$155

Figure 36

- **Product details page:**
  In this page we use will see all info about the product including Name , type ,brand , picture , price , in stock number ,description. User can specify the quantity he wants to buy from this product (user couldn't choose quantity less than 1 or more than in stock) , the price will change according to the quantity .

  The add to cart button will add this product to the user cart with the quantity user already set .

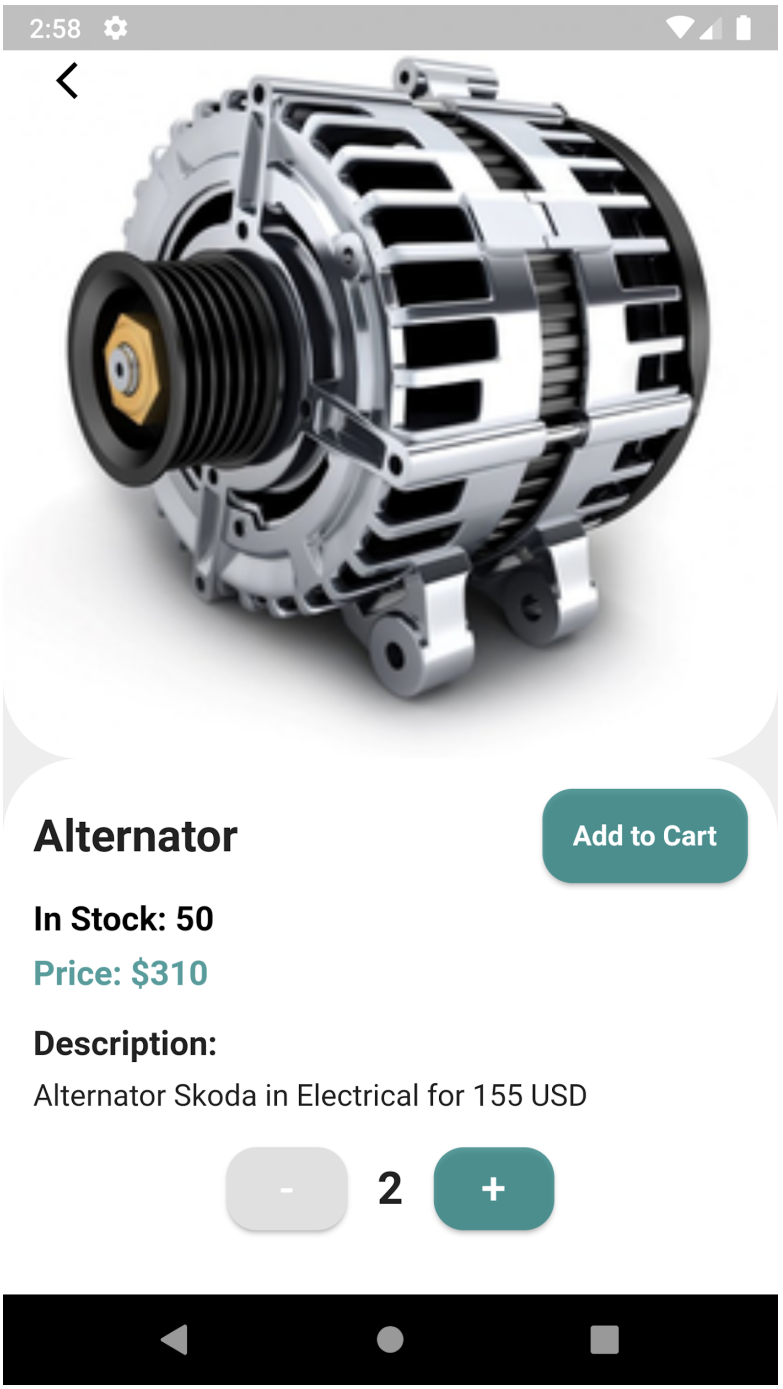


Figure 37.Product details page

- **Cart page:**
  In the cart page all products user added to the cart will appear there, to let the user review his cart before checkout .
  Users could modify the quantity of a product and clear the whole cart .
  This page is a central hub where clients can review and manage the items they have added to the shopping cart.

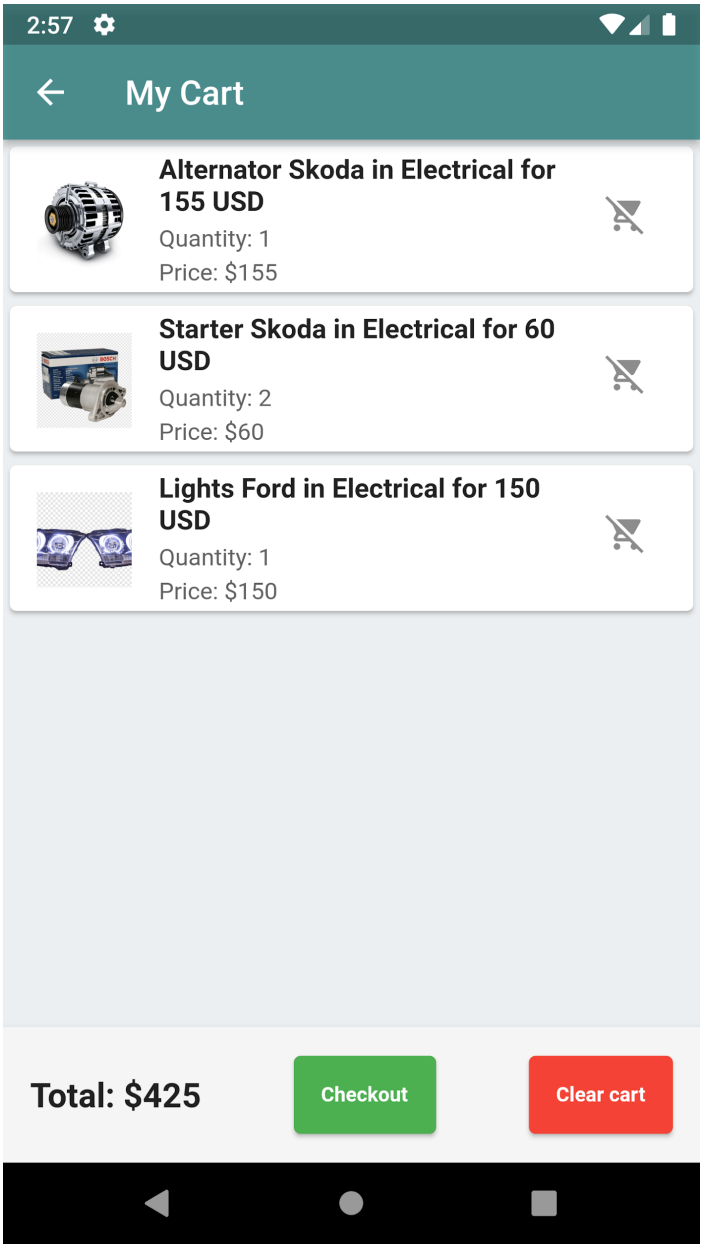


Figure 38

When checkout the online payment method will apply the same as the previous one in the orders .

4.4.3 Worker Interfaces:

- **Home Page:**The profile image, name, and car model are displayed on the home page with a list of random services and the processing orders according to the status of order.
  The home page also includes a bottom bar that gives rapid access to the app's primary features, such as car services, the app store, and orders.*Figure 39*

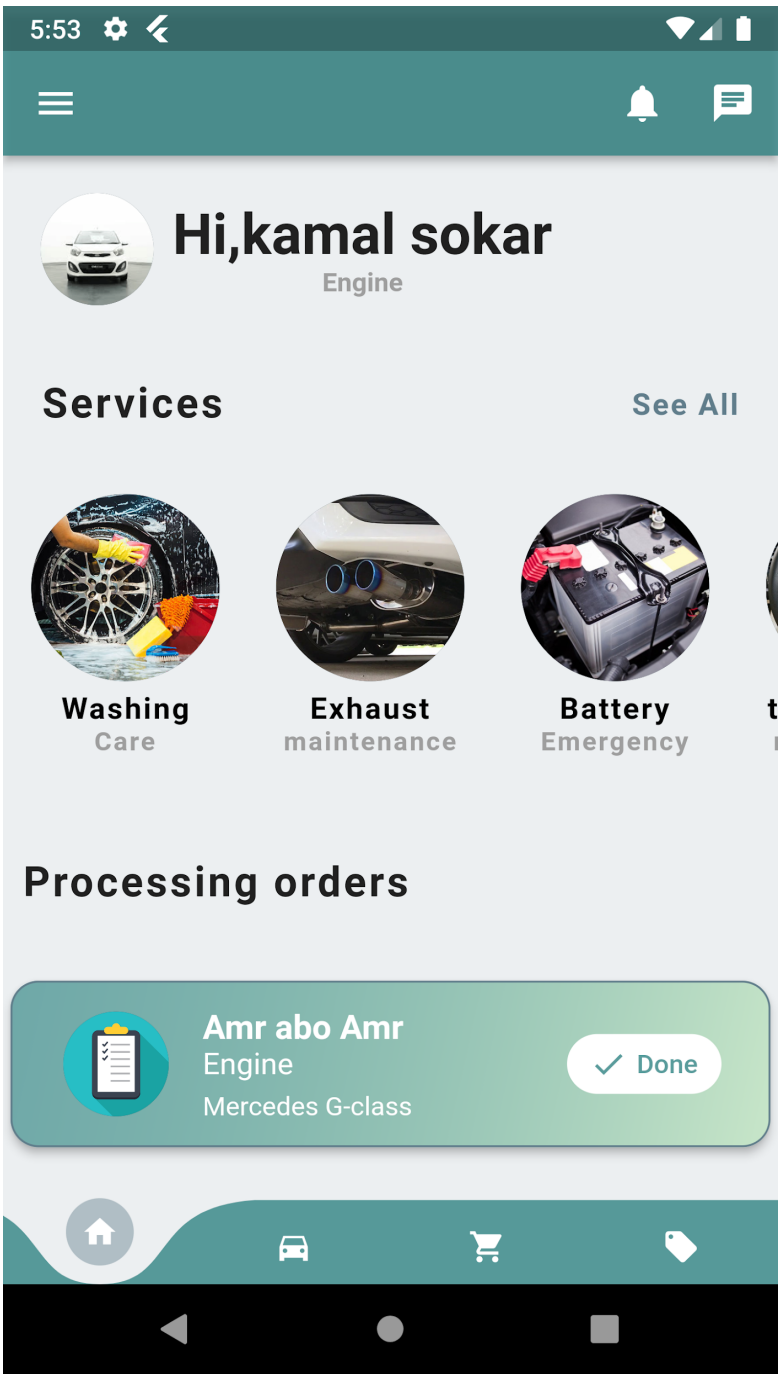


Figure 39.Home Page

- **Sidebar Page:**On the home page, there is a sidebar menu with choices to view the worker's profile, orders, reset password, learn more about the app, and logout.*Figure 40*

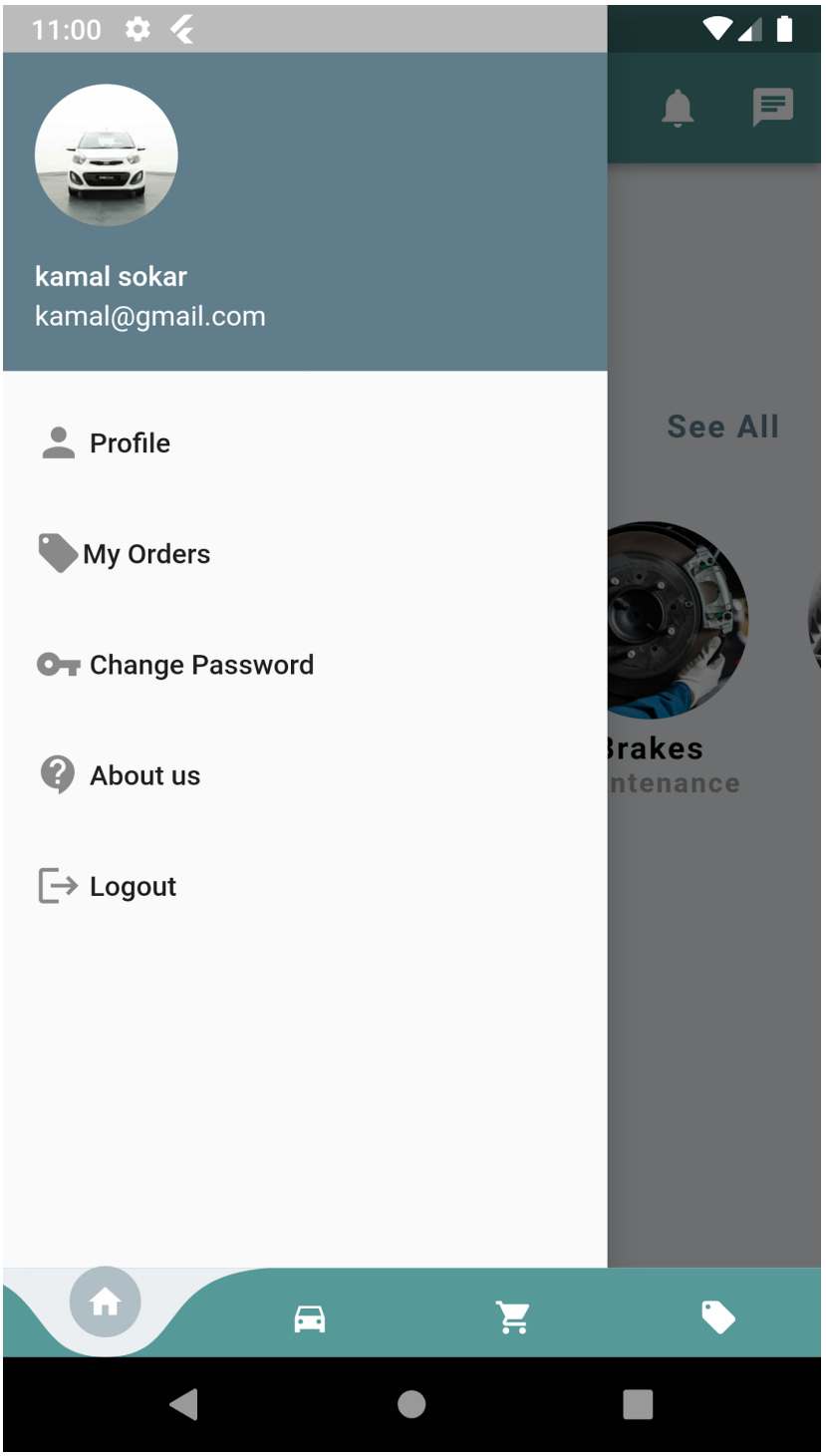


Figure 40.Sidebar Page

- **Profile page:** we show the worker's profile image, car model, name, about ,major, rating and location.*Figure 41*

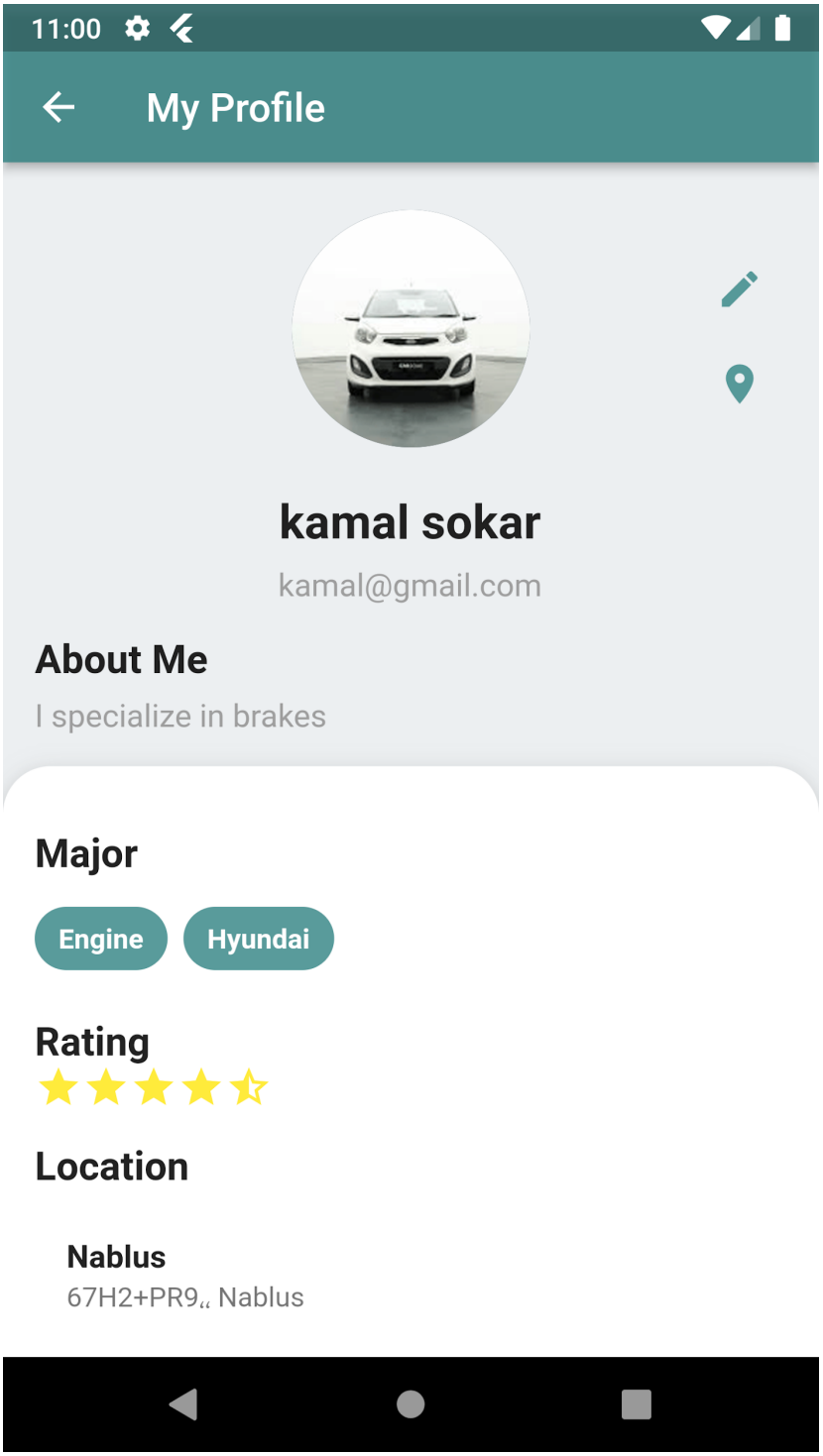


Figure 41.Profile page

- **Edit Profile page:** worker can change and update some information, like his first and last name, car model, major and phone number, on this page.Additionally, a worker's profile photo could be updated on this page.*Figure 42*

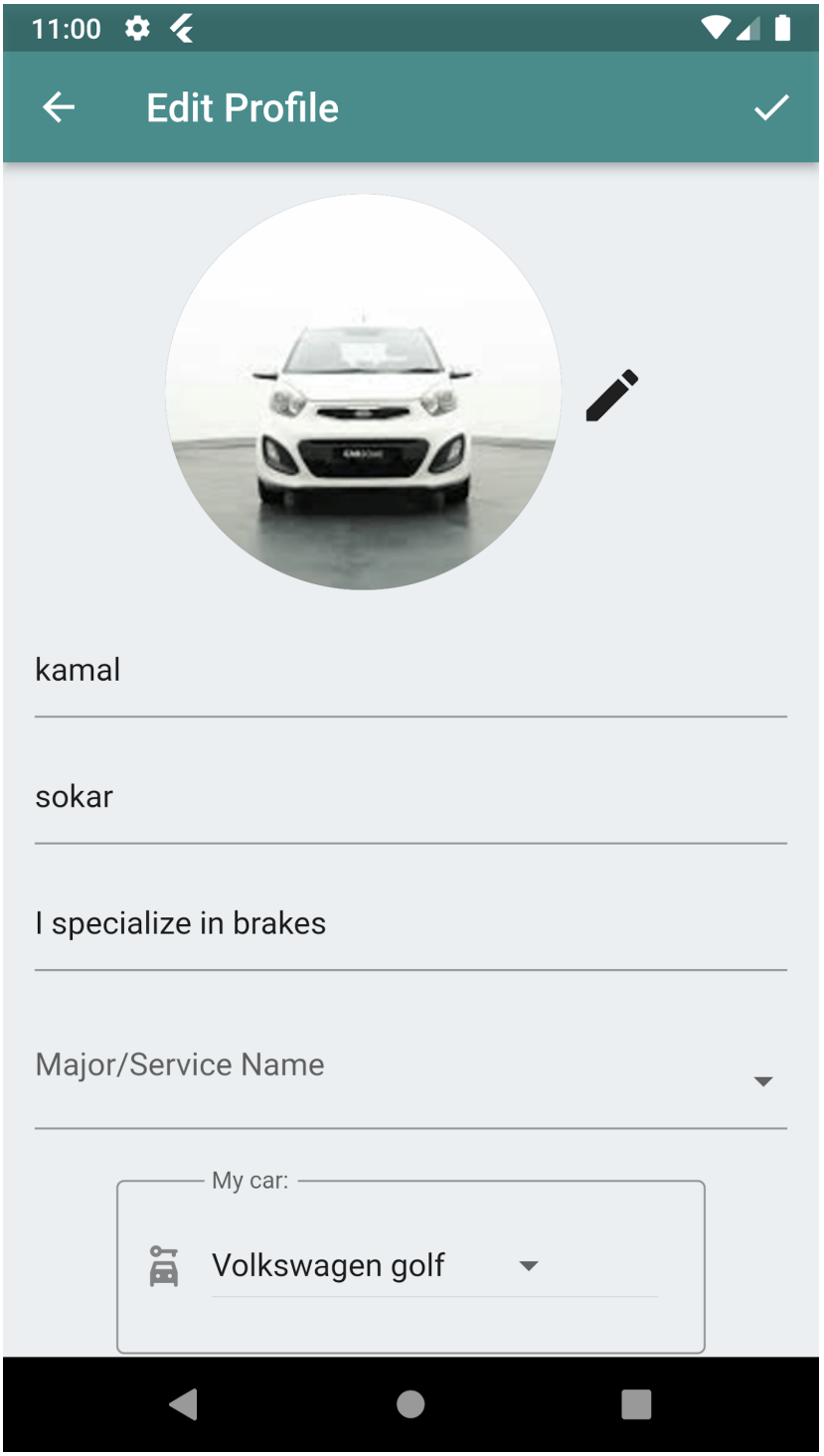


Figure 42.Edit Profile page

- **Note:** after we changed the first name for the worker to jamal and the worker's major to engine ,the home page became as below:

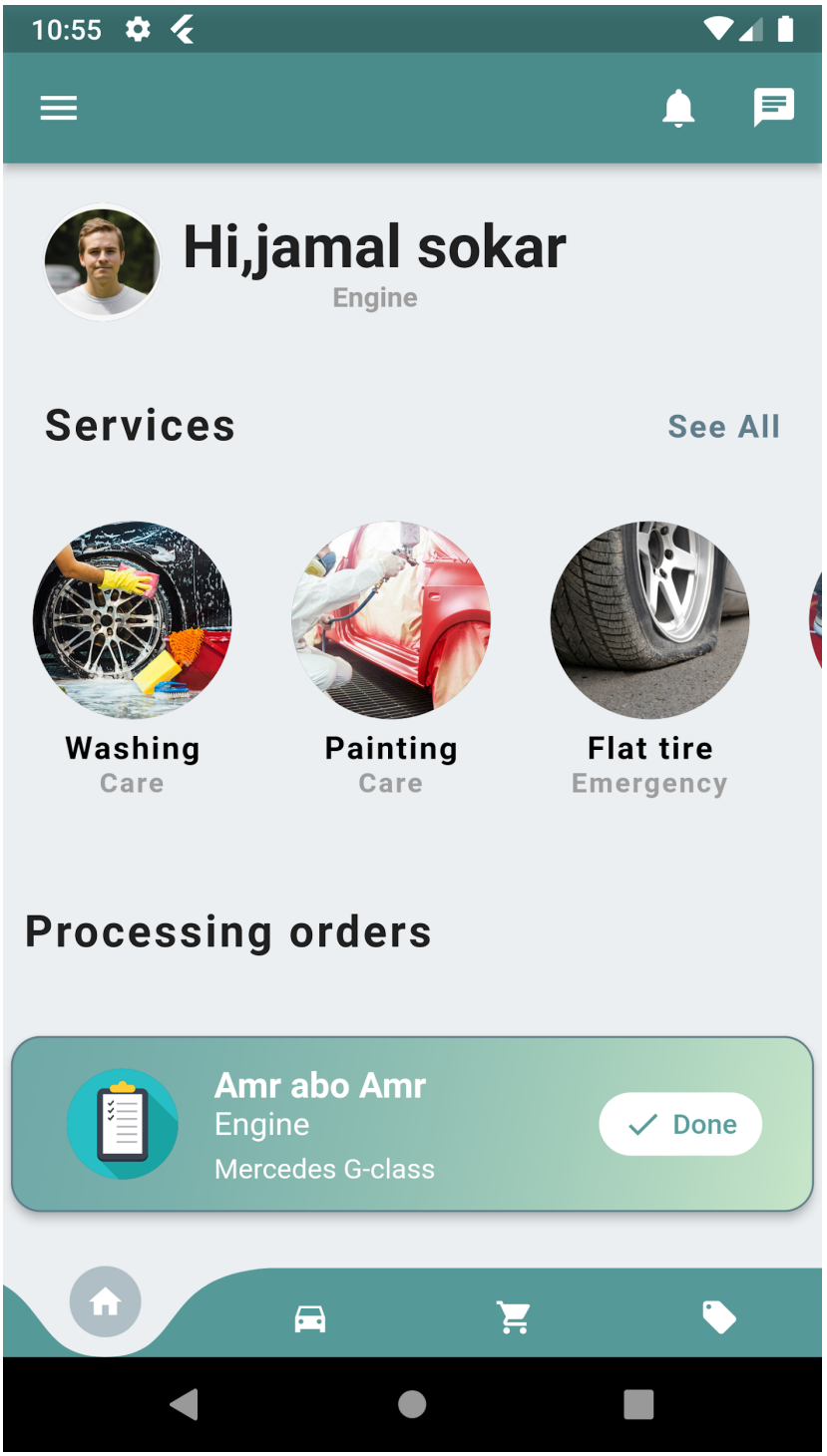


Figure 43

- **Orders Page:** in this page we show all orders for this worker and he can filter the orders on this page based on their status (requested, waiting, processing, canceled, and completed). As shown in *Figure 44*

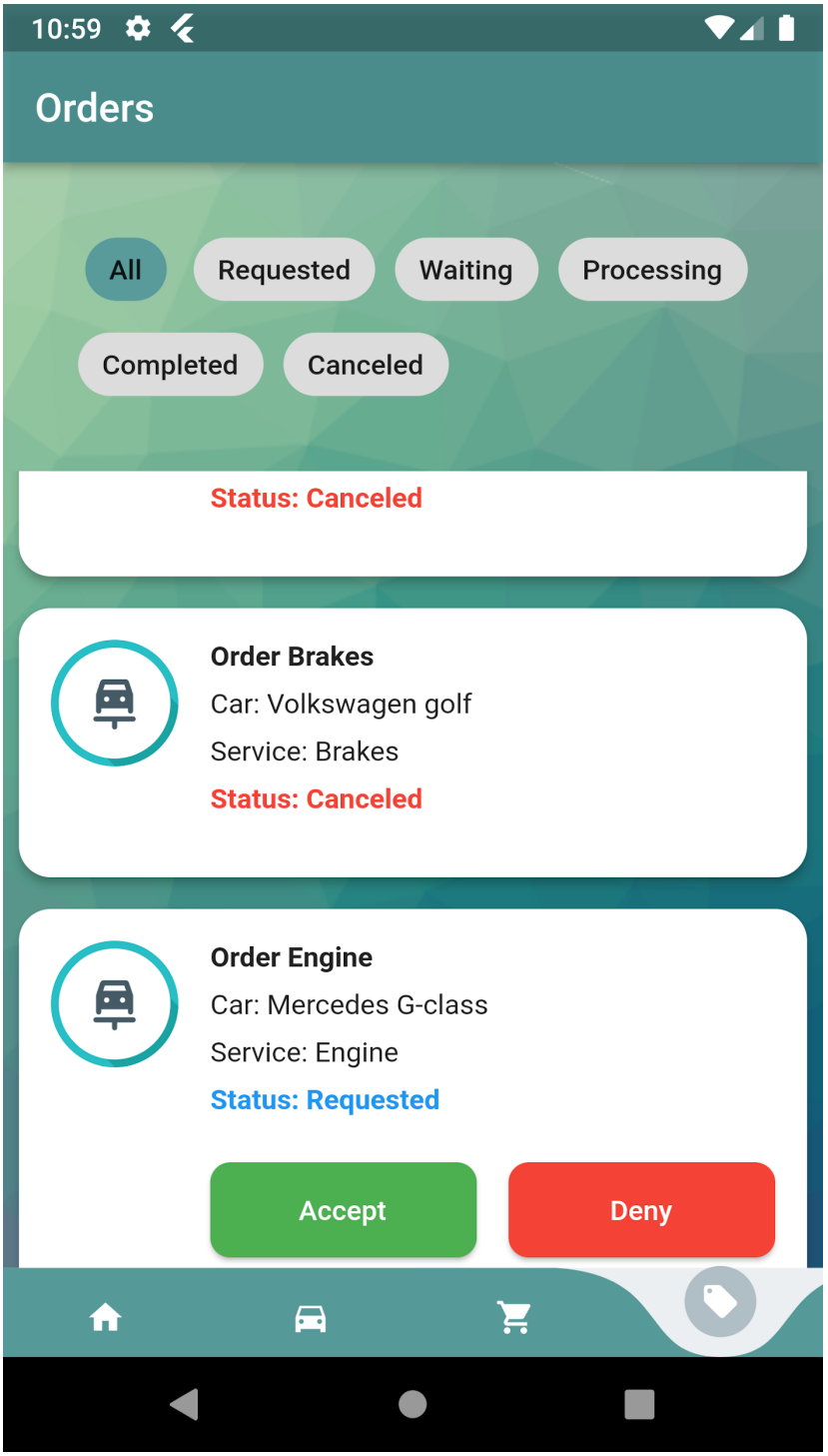


Figure 44

- **Orders Page:**In the requested section all requested orders for this worker will appear and he can accept or deny the requested order.*Figure 45*

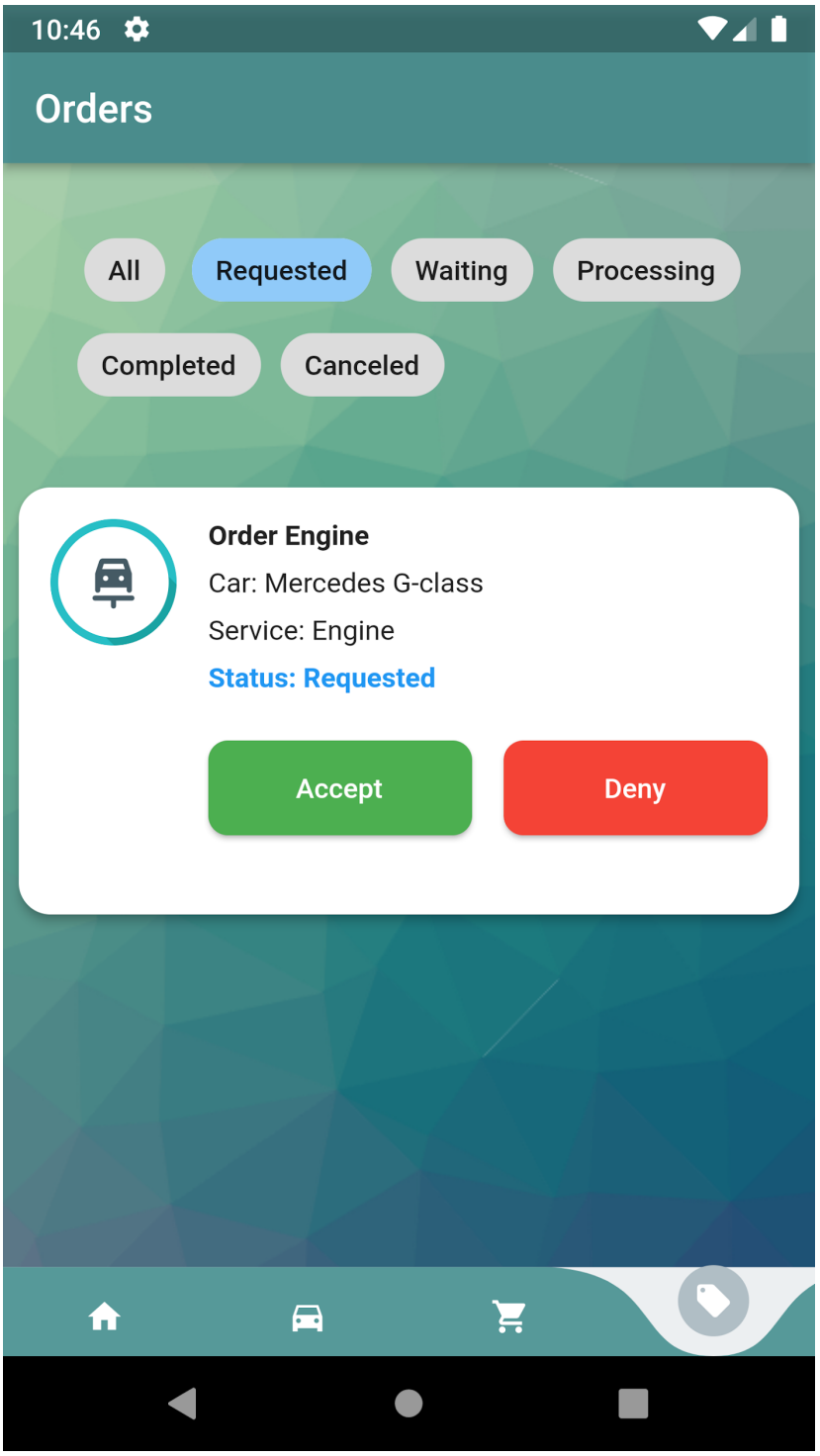


Figure 45.Orders Page

- **Cancel Order:**if the worker denies the order this window will appear to confirm the denial .*Figure 46*

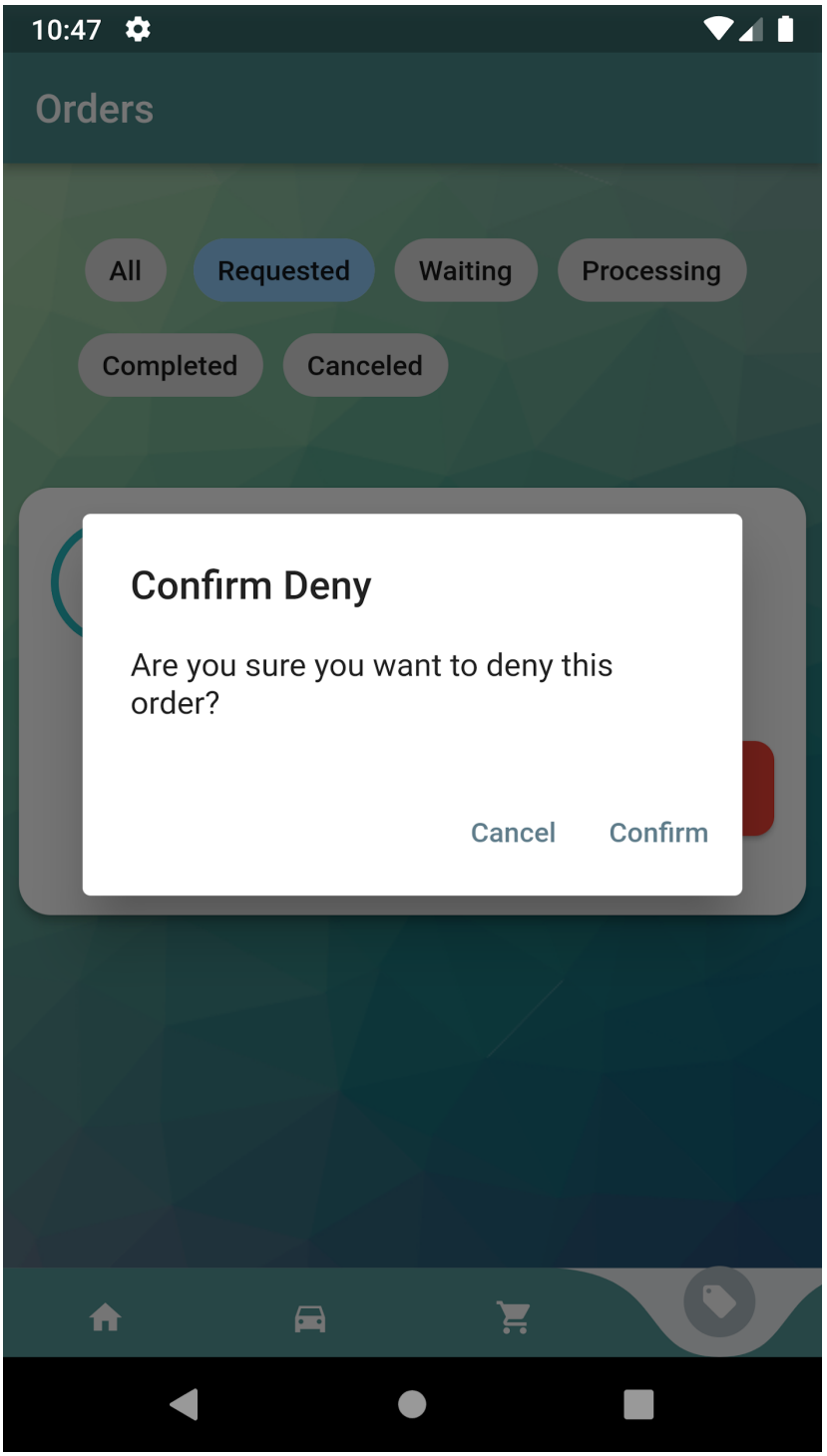


Figure 46.Cancel Order

- **Accept Order:**if the worker accepts the order this window will appear to add the price of the testing then the app will **notify** the client to let him know that his order has been accepted .
  As shown in *Figure 47*

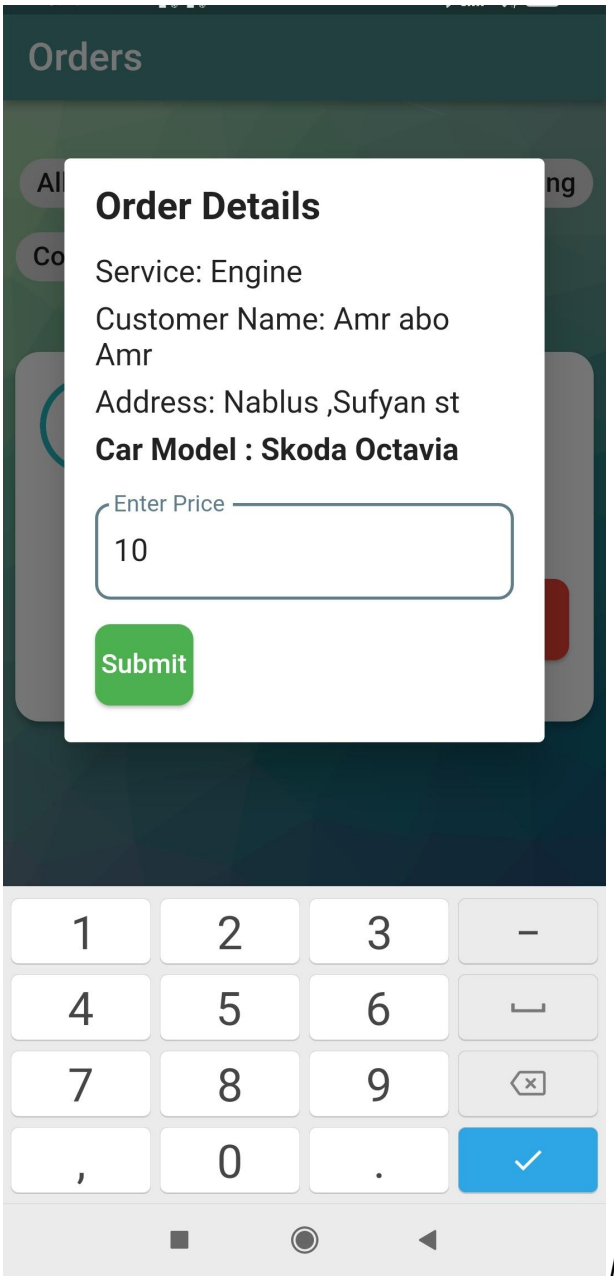


Figure 47.Accept Order

- **After the worker accepts the order:** in this case the order status will change to 'Waiting' (waiting for client to accept the price).
  As shown in *Figure 48*

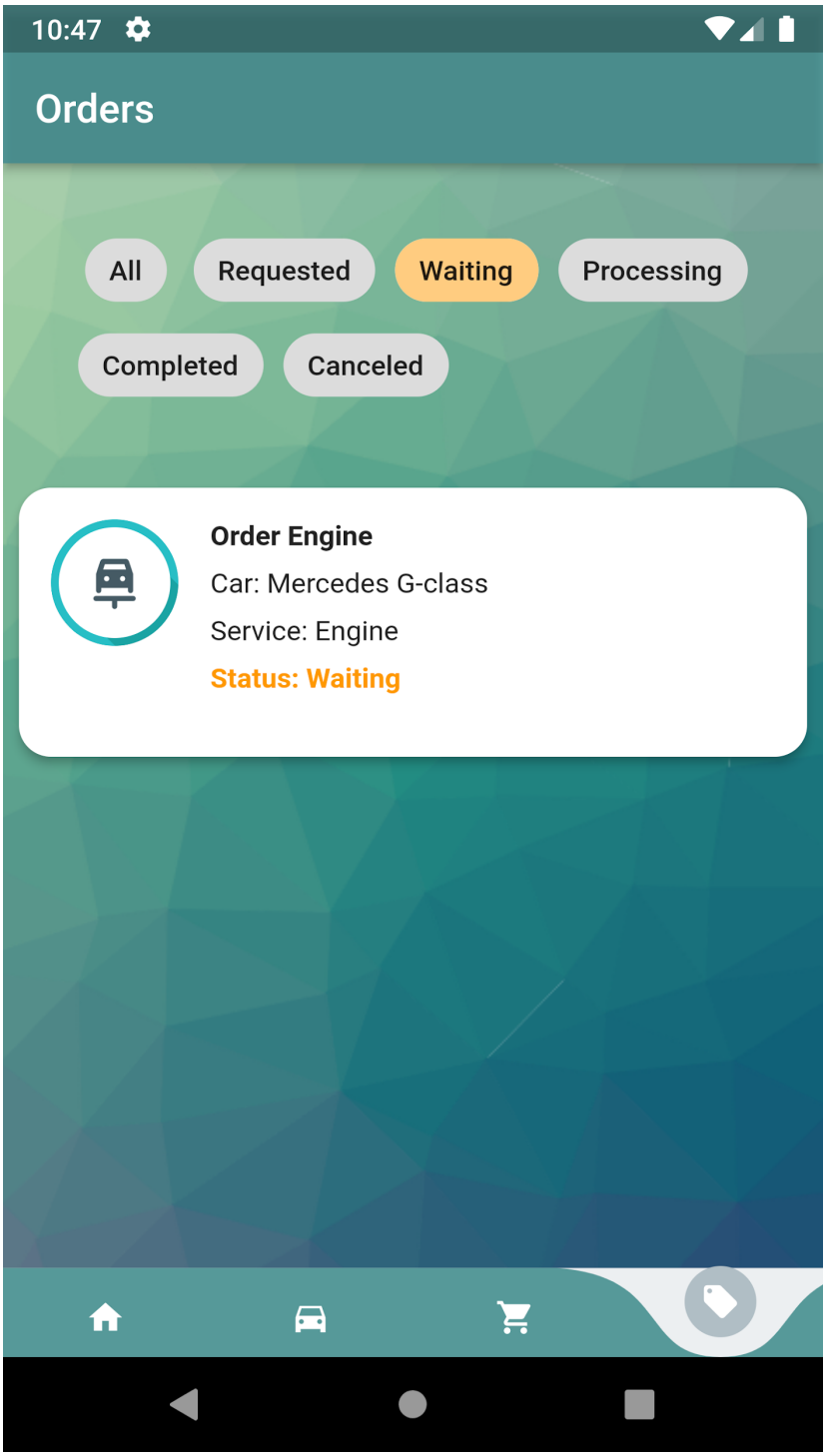


Figure 48

- **Note:** When the client accepts the order the app will **notify** the worker to let him know that his order has been accepted as

shown below.In this case the order status will change to ’processing’ .*Figure 49*

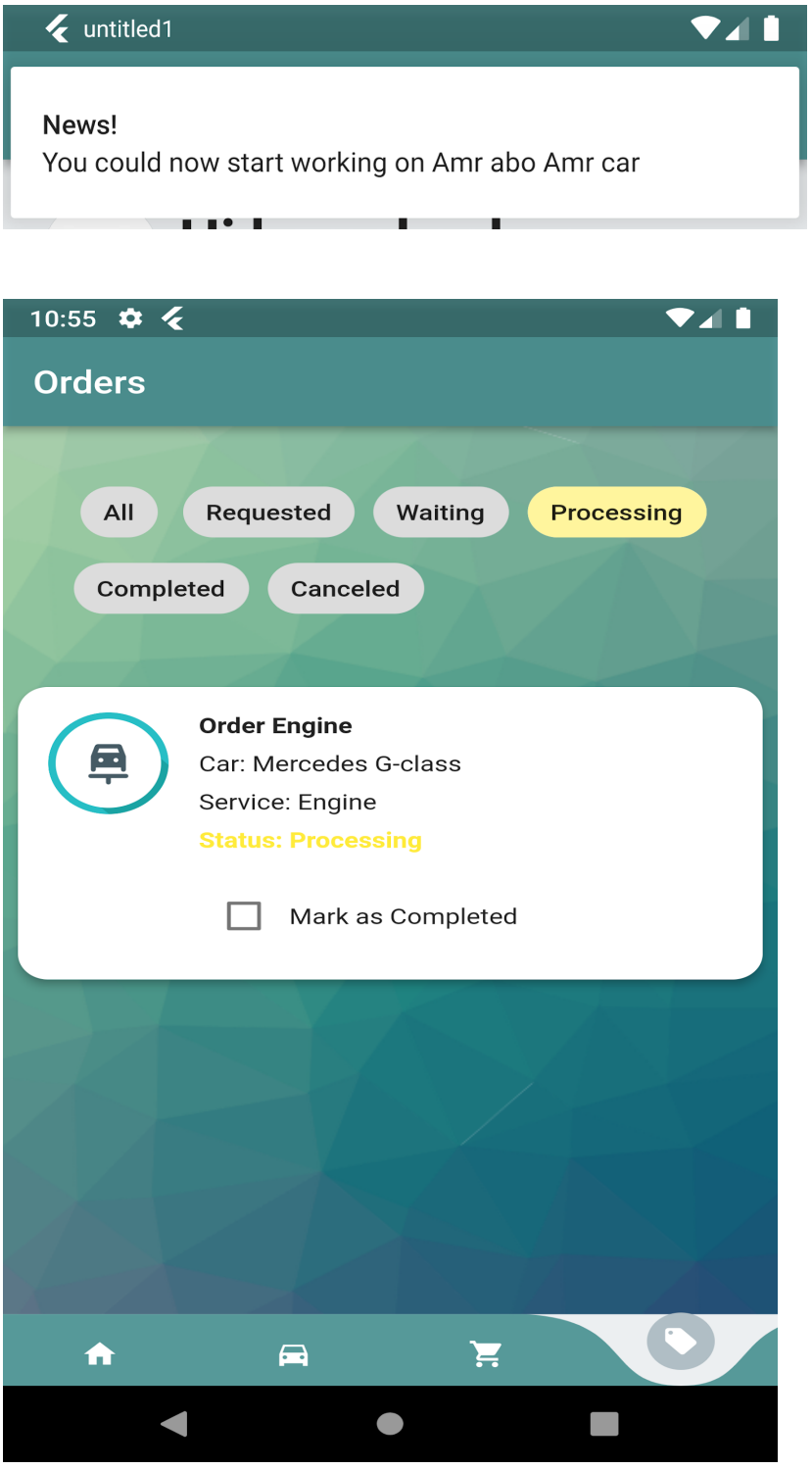


Figure 49

- **Once the worker completed the service,** he should mark the order as “Completed ” . Then he has to complete the testing report to add the maintenance price range , estimated time to finish the service and the problem description which is the testing

result This status signifies that the testing has been successfully executed .The app will **notify** the client.
As shown in *Figure 50*

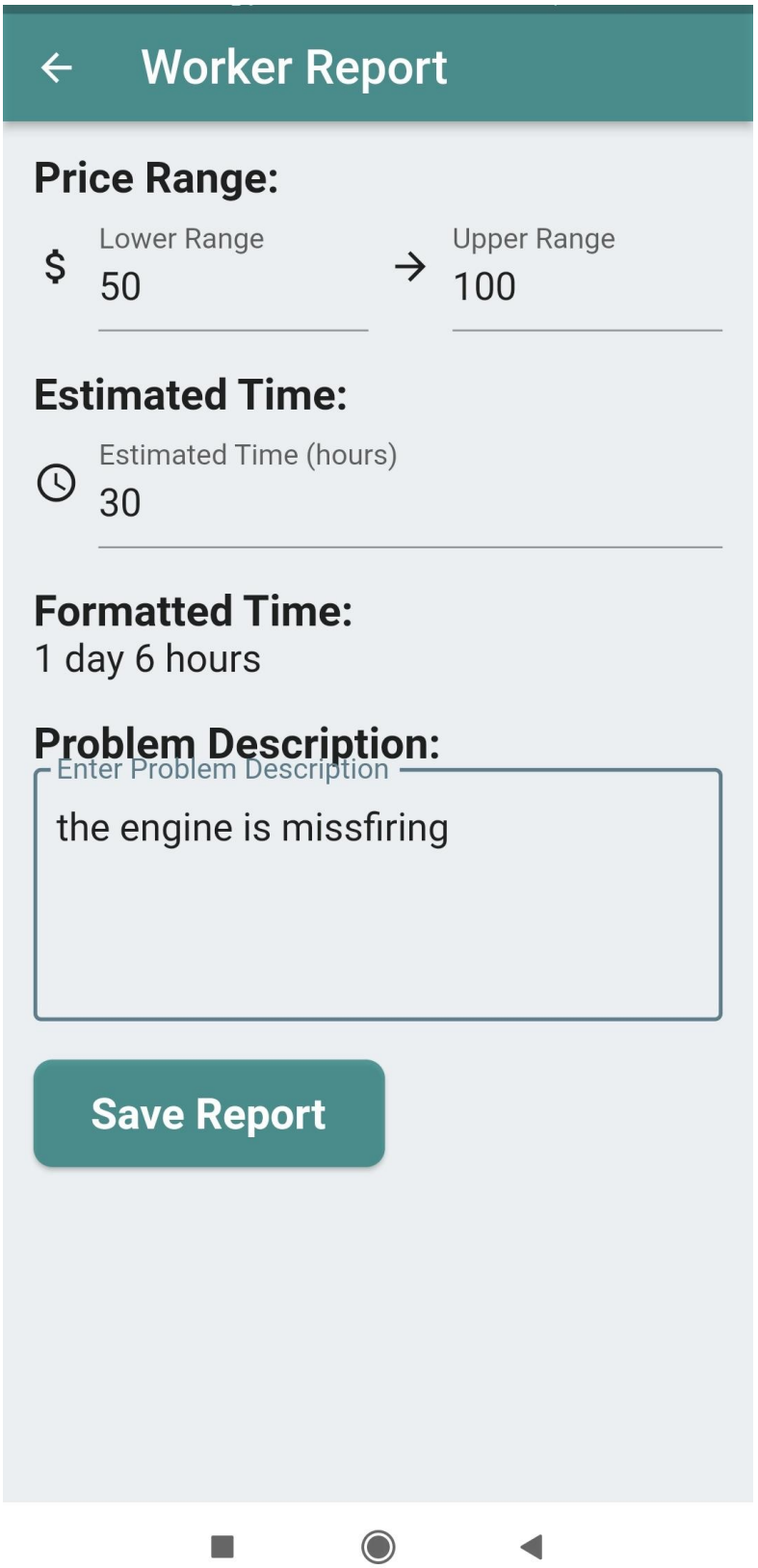


Figure 50

- **Note:** Once the worker marks the service as completed,the app will **notify** the client.*Figure 51*

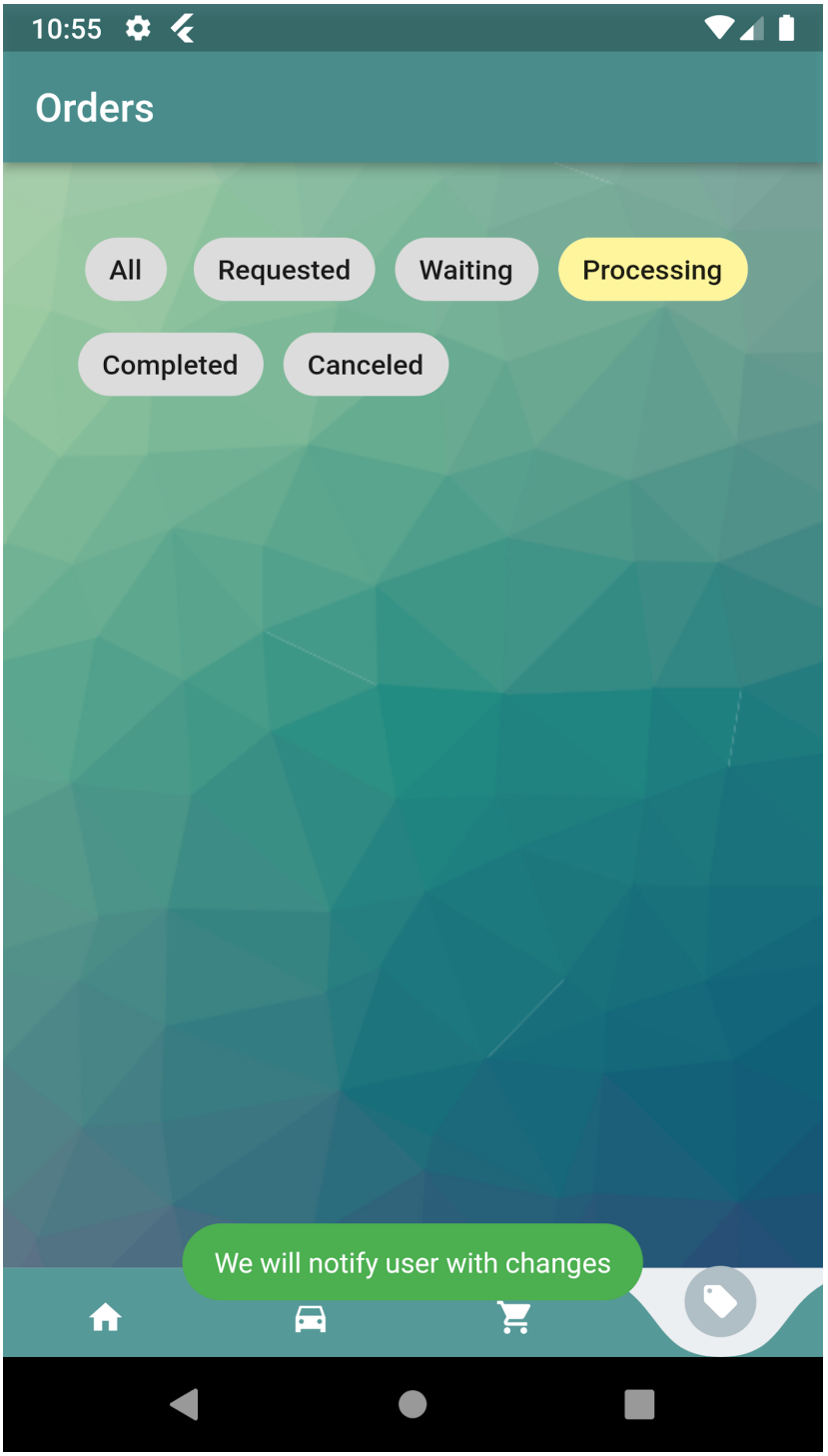


Figure 51

- **Note:** Once the worker marks the service as completed the order status will change to ’completed’ .*Figure 52*

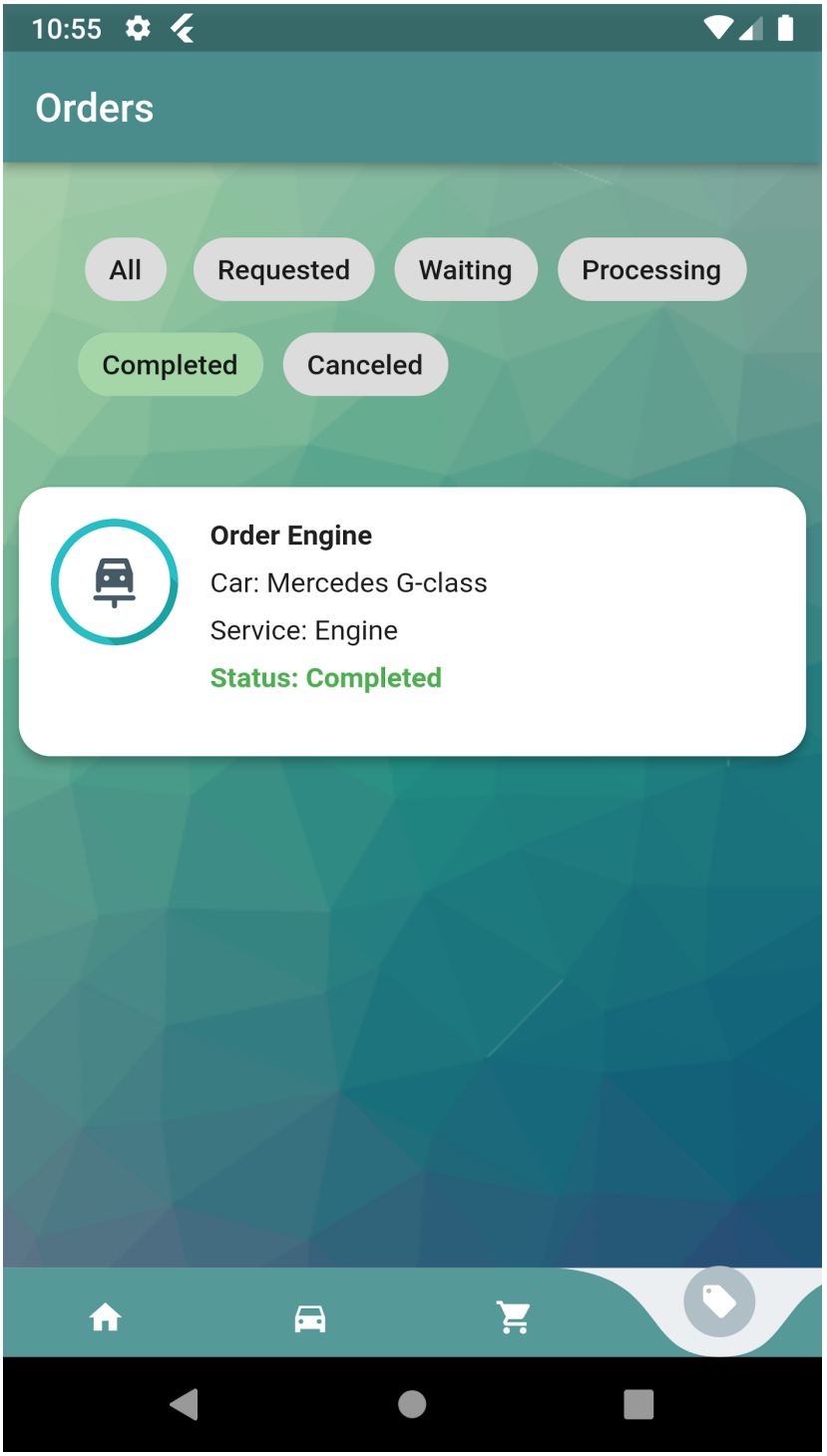
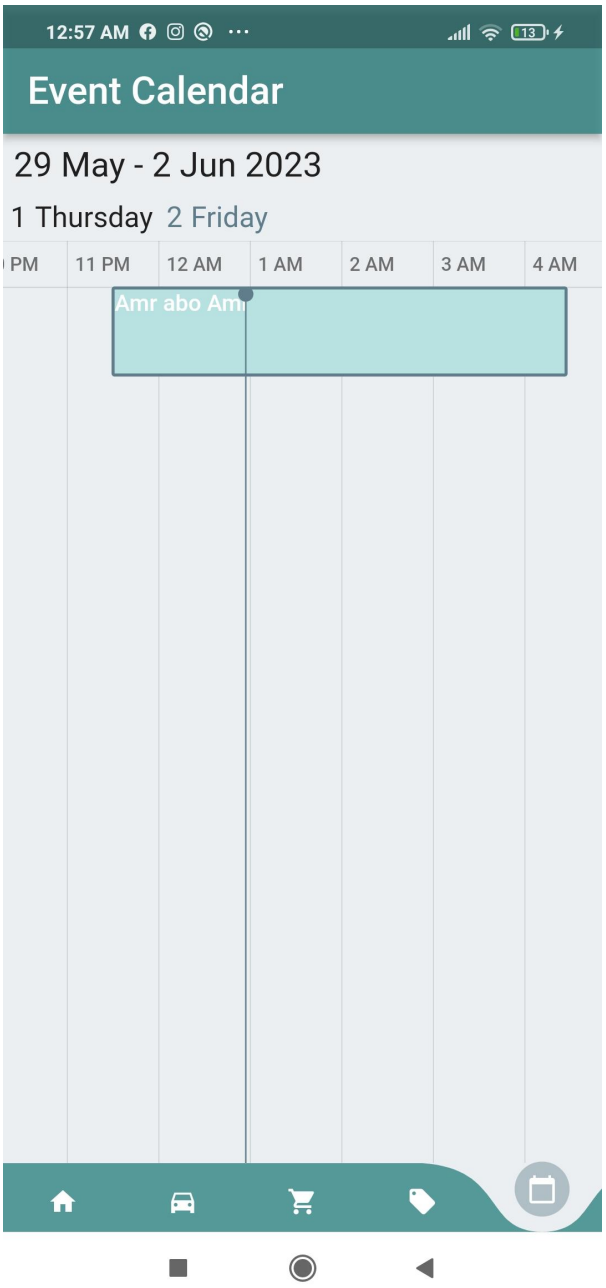
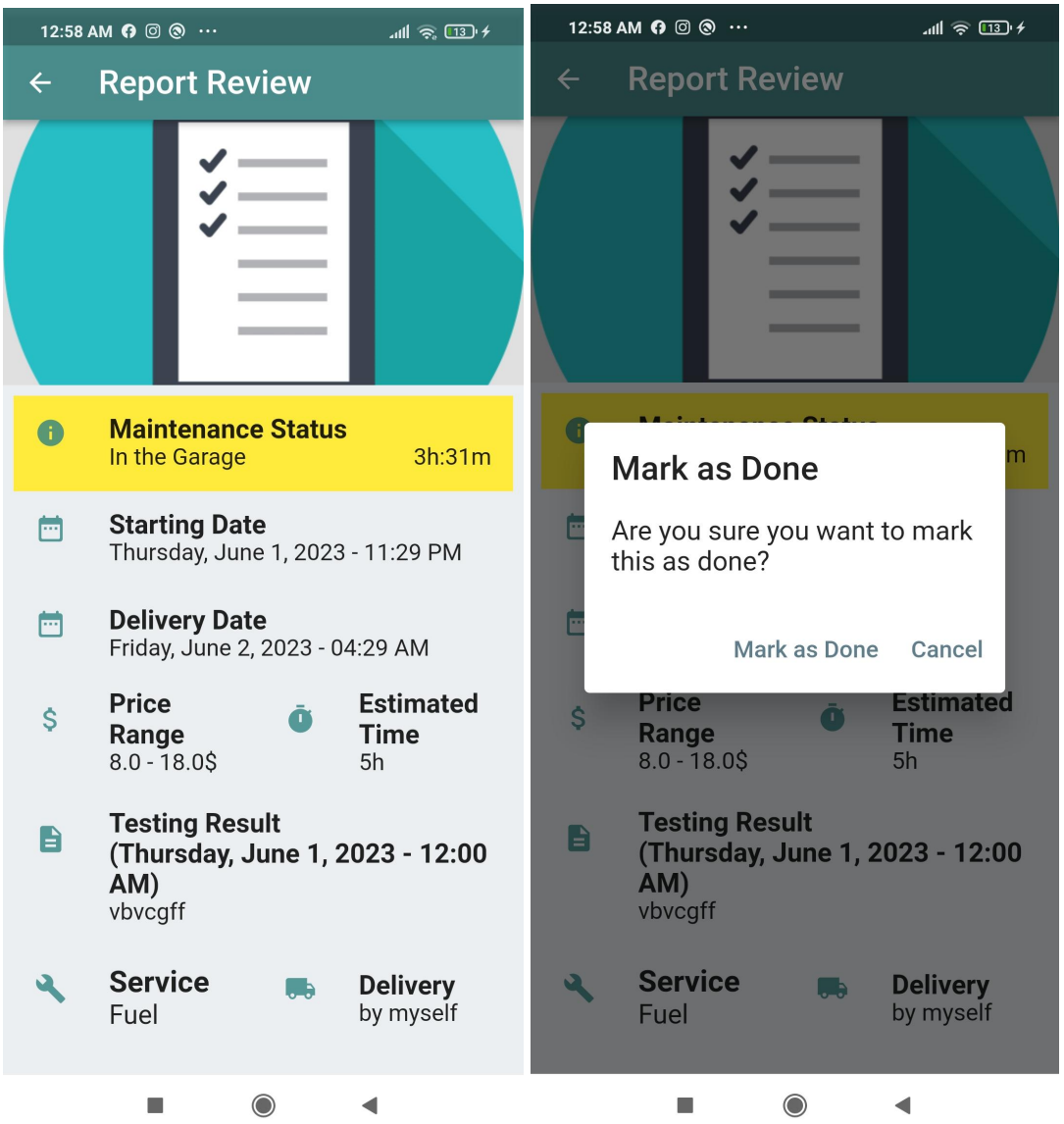


Figure 52

- **Calendar page:**
  The calendar feature in our app provides a comprehensive view of the worker's tasks and schedule. It displays the assigned jobs and appointments for the worker, allowing them to effectively manage their time and resources. The calendar provides a clear overview of upcoming tasks, including the date, time, and details of each assignment. Workers can easily navigate through different dates and view their tasks for a specific day, week, or month. This feature enables efficient scheduling, helping workers to plan their



- **Task page:**
  When clicking on the task on the calendar the task's details page will show the worker an overview of his client car report ,testing result , remaining time ,maintenance requirements .



Note : after marking the task as done the client will be notified that his car is ready .

- **Chat:**
  The chat feature in our app serves as a powerful communication tool, a real-time interaction between users and service providers. Through the chat functionality, users can easily communicate with service providers to discuss service details, ask questions, or provide additional instructions. The chat feature enhances transparency, as users can receive updates and progress on their car maintenance tasks, The chat feature ensures a smooth and satisfactory service experience for our app users.
  As shown in *Figure 53,54*

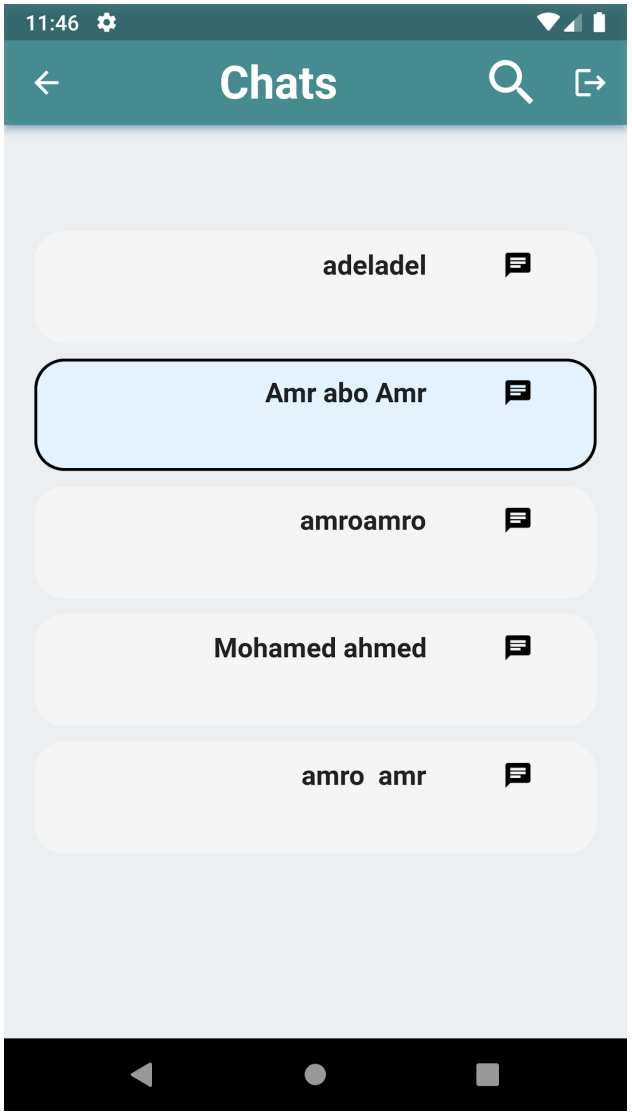


Figure 53

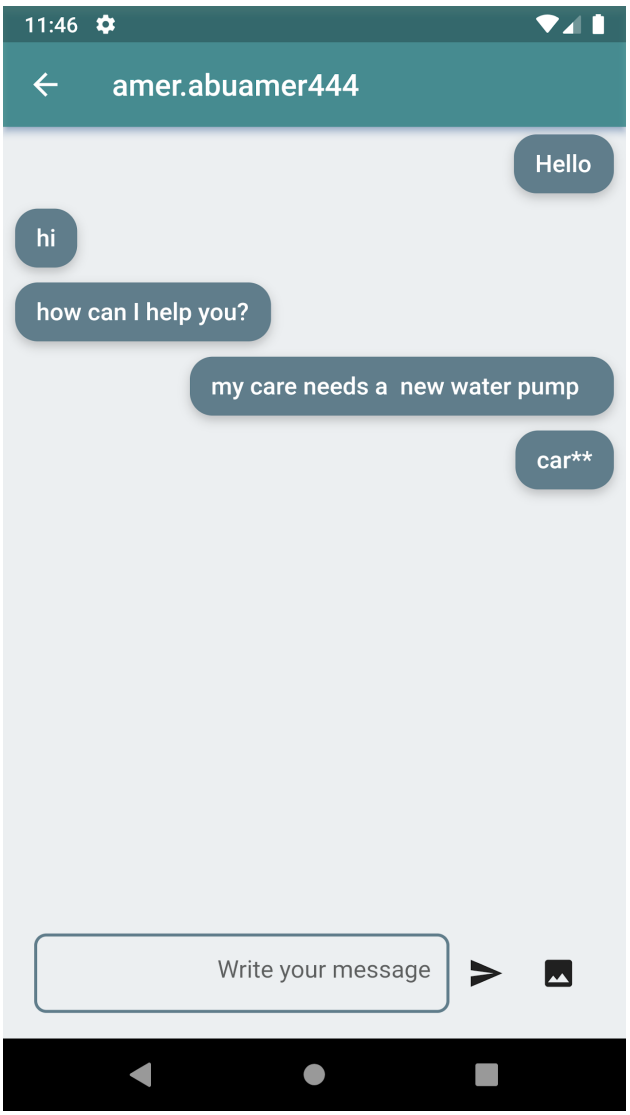


Figure 54

4.4.4 Admin Interfaces:

Note:The pictures we took for the Admin's pages are varied from the web and mobile.

- **DashBoard Page:**this page displays the total number of users,workers,orders and services in the application. In addition to displaying the recent users and recent orders.*Figure 55*

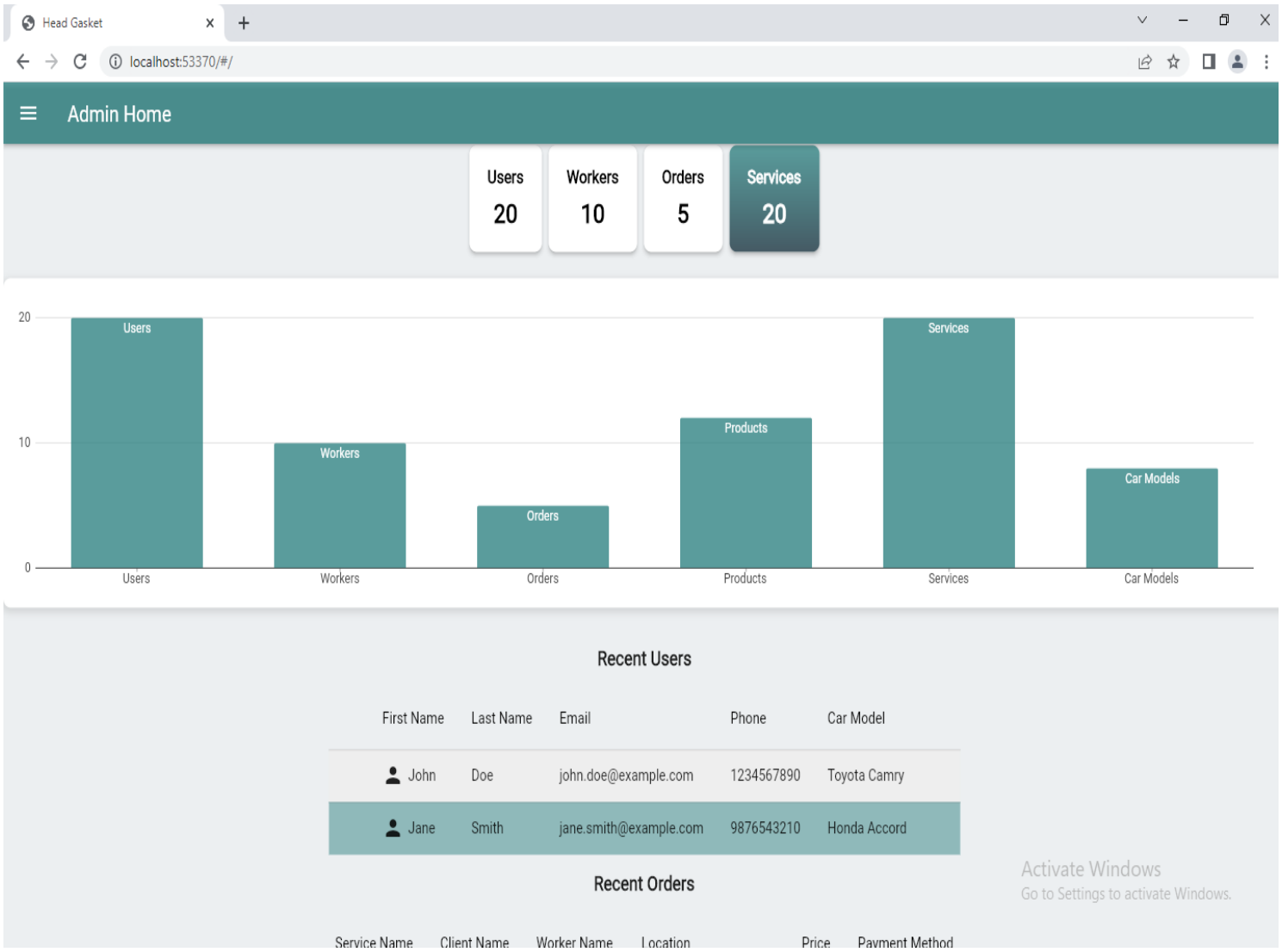


Figure 55.DashBoard Page

- **Sidebar Page:**On the DashBoard page, there is a sidebar menu with choices to view the users, workers, orders , products and cars. *Figure 56*

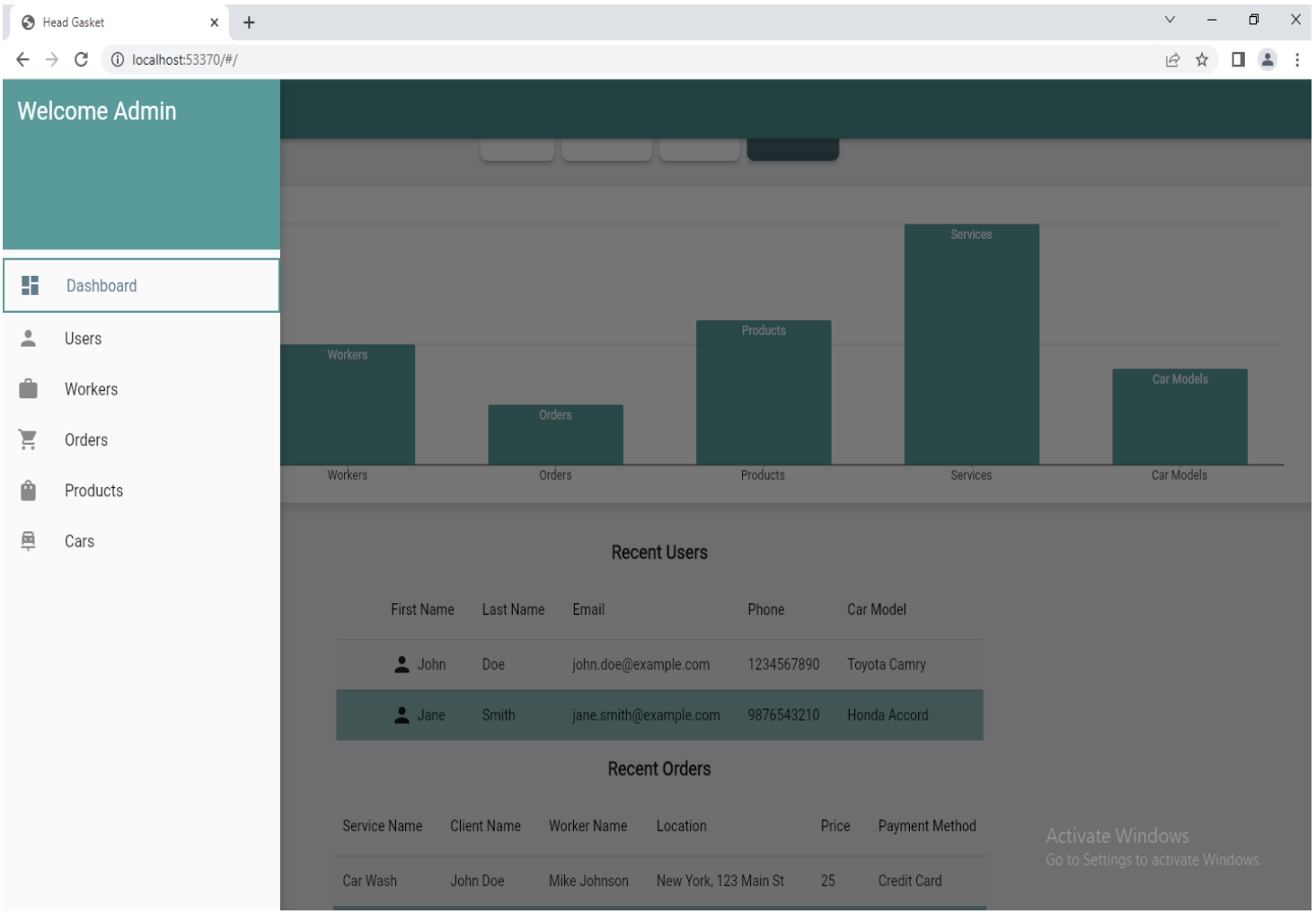


Figure 56.Sidebar Page

- **Users Page:**this page displays all the users.Admin can search for a specific user and he can also delete users.*Figure 57*

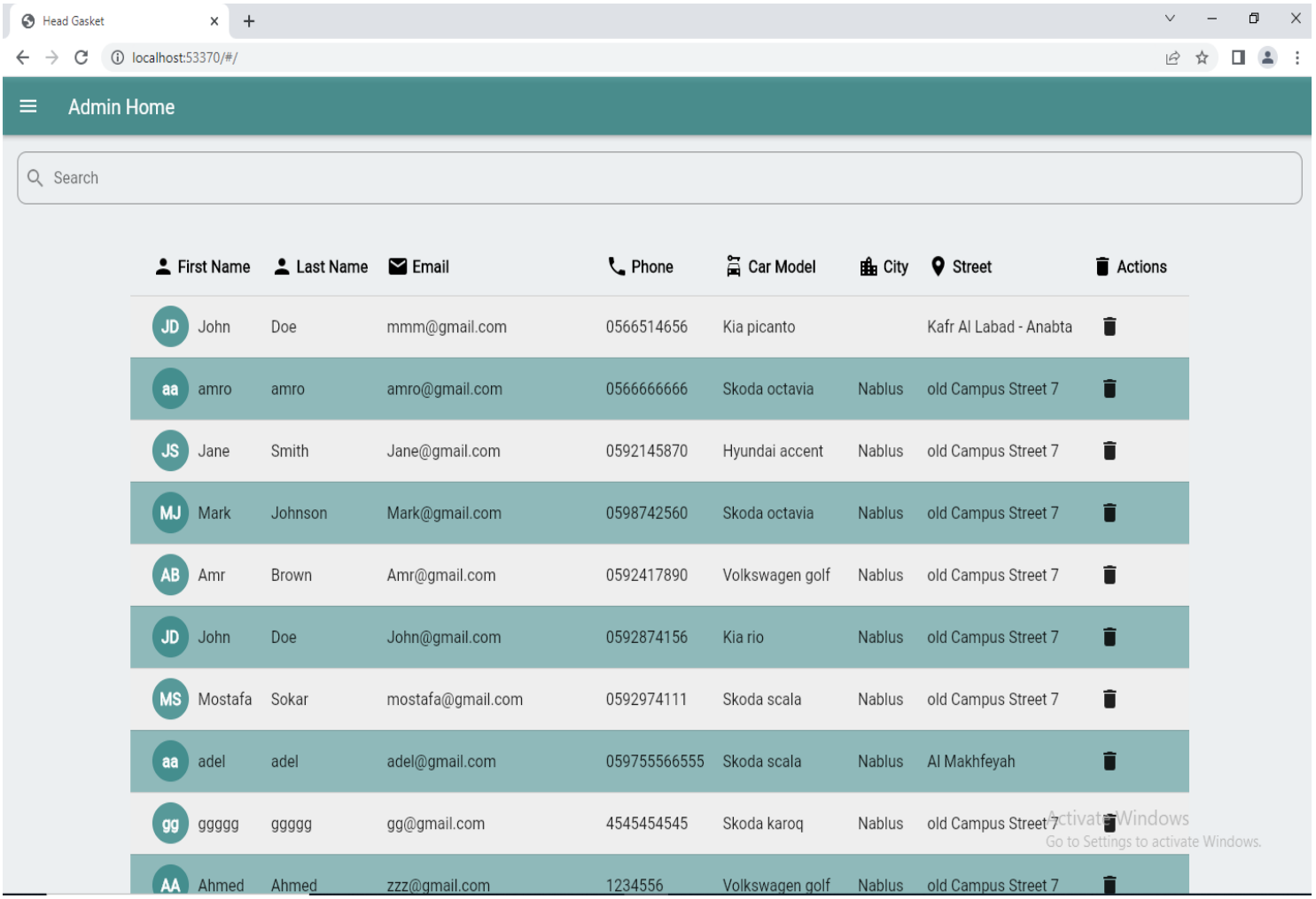


Figure 57.Users Page

- **Workers Page:**this page displays all the workers.Admin can search for a specific worker and he can also delete workers. As shown in *Figure 58*

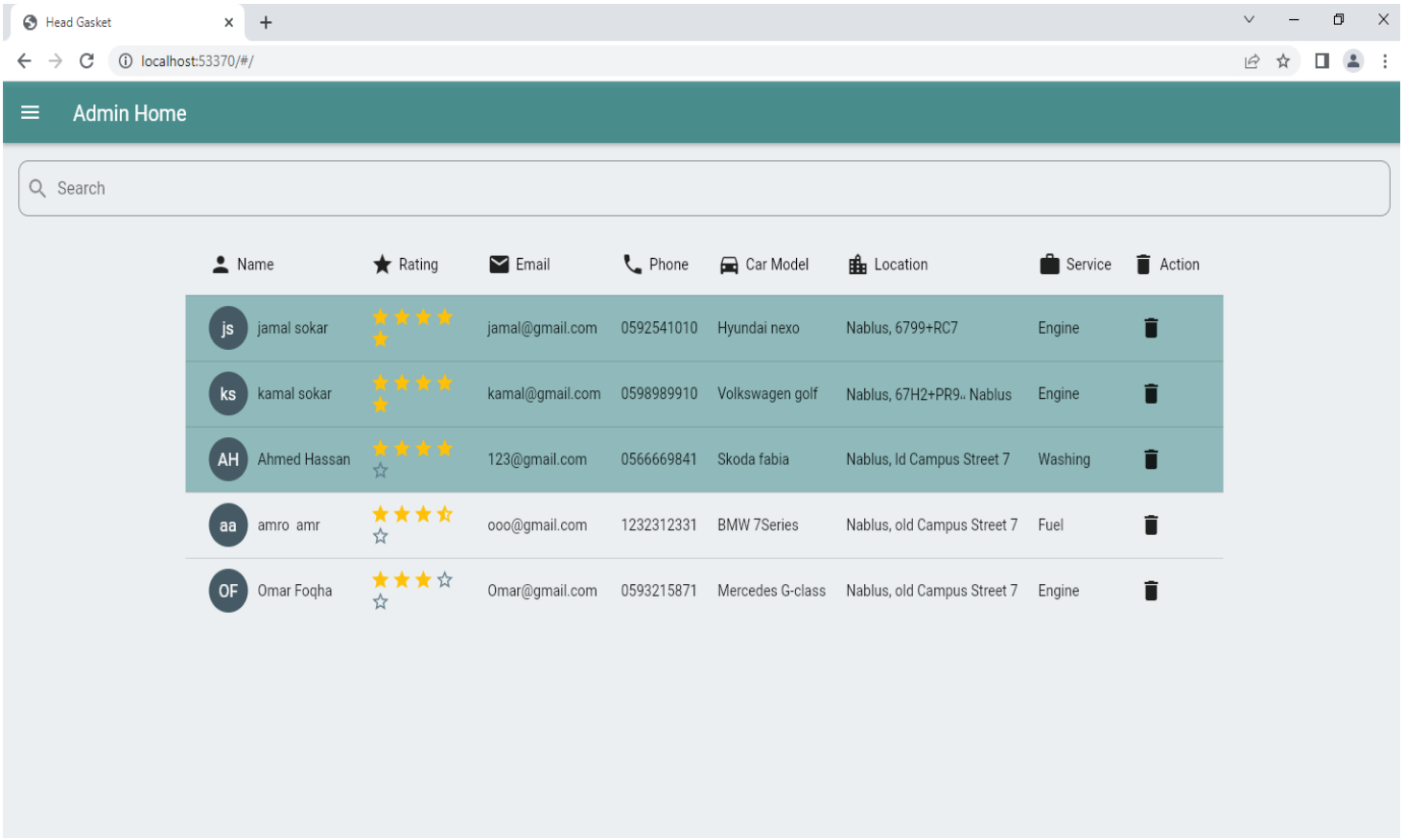


Figure 58.Workers Page

- **Orders page:** this page displays all the orders.Admin can filter the orders based on their status (Requested , waiting , processing , canceled , completed) and he can cancel the order.*Figure 59*

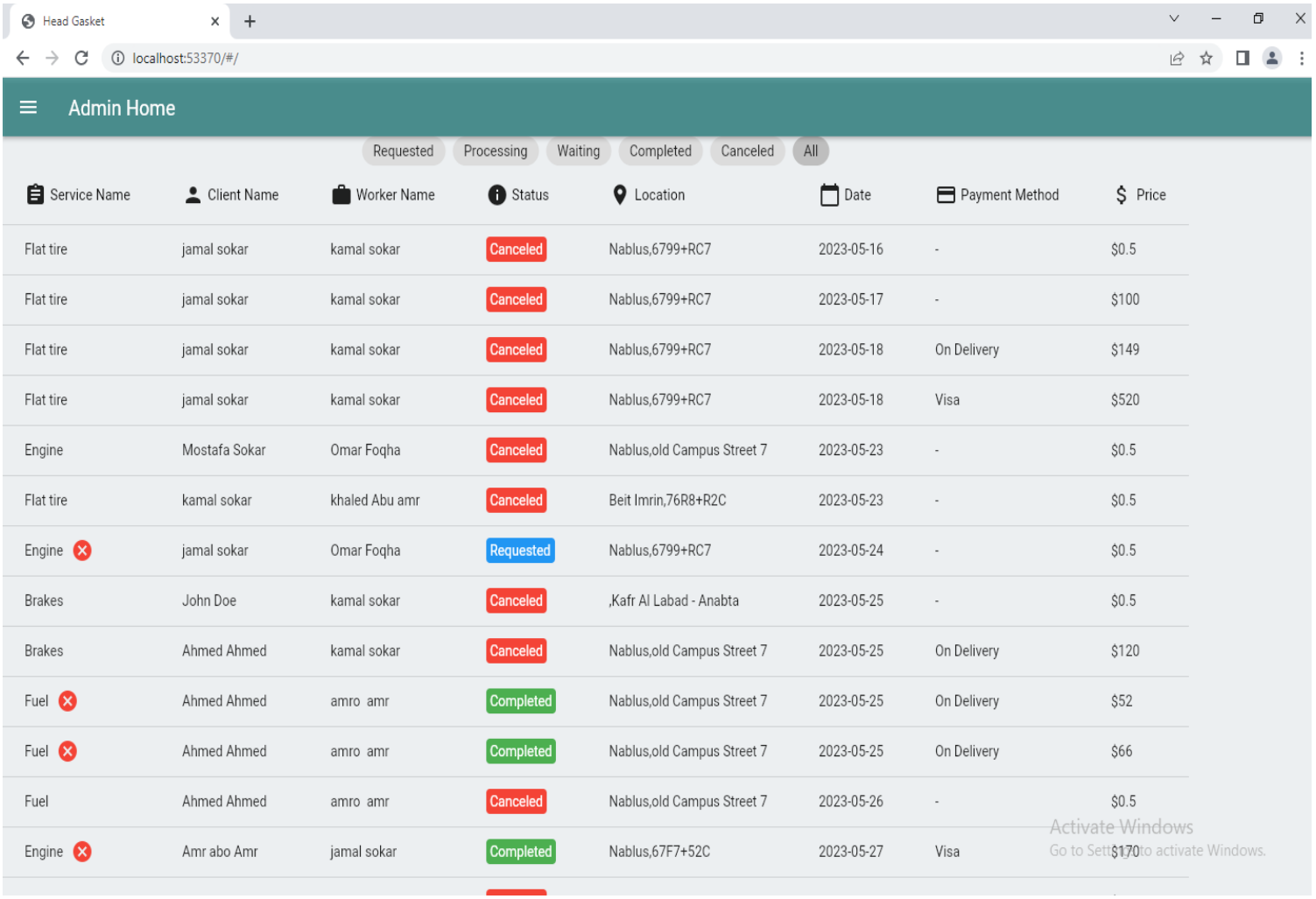


Figure 59.Orders page

- **Example:**the admin filtered the orders based on completed orders. As shown in *Figure 60*

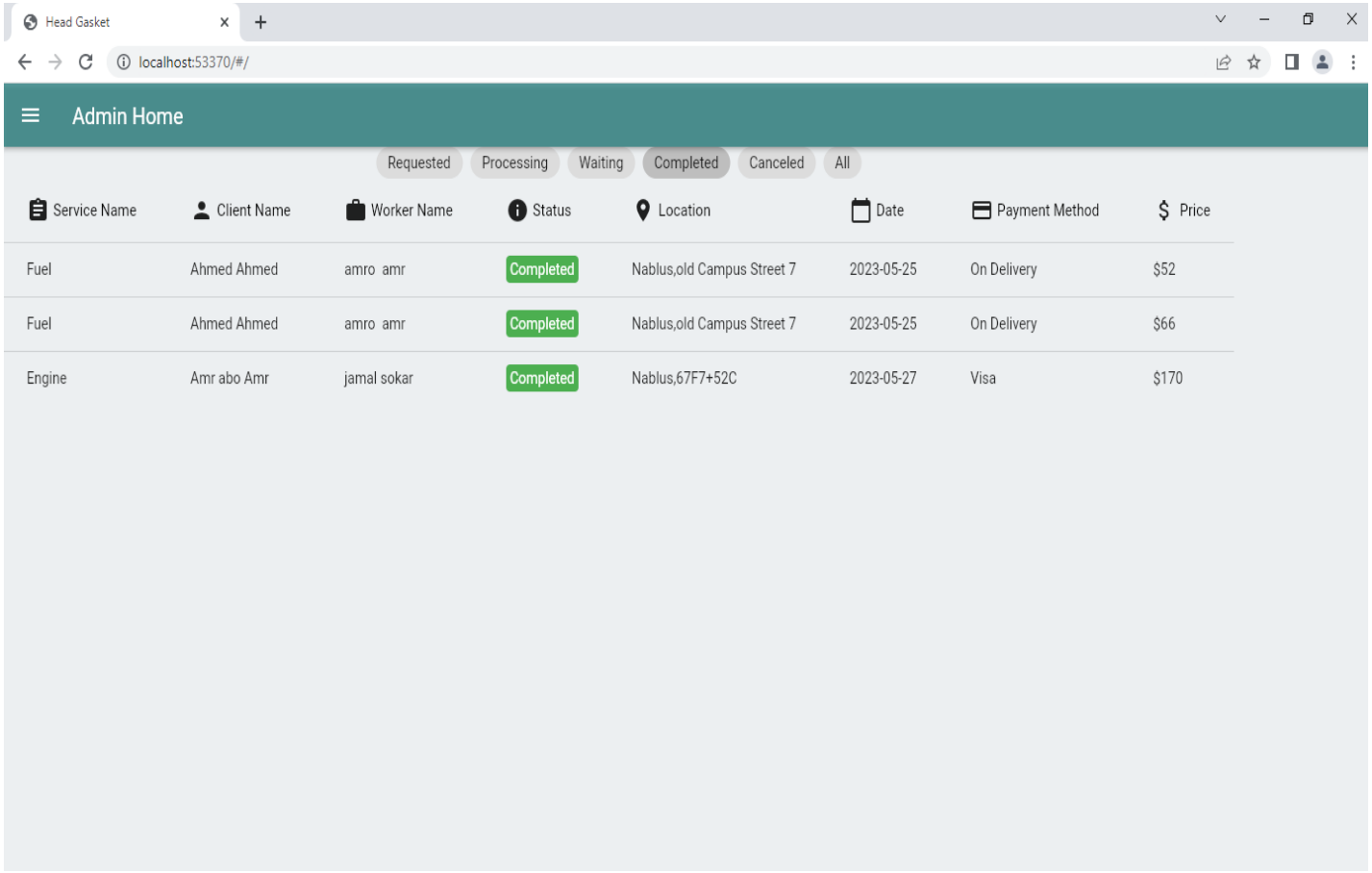


Figure 60

- **Products page:** this page displays all the products in the store .Admin can filter the products based on type (Body , Mechanical , Electrical , …) or based on car brand (Toyota,Ford,Skoda,...) or both .He can also add , edit and delete products.*Figure 61*

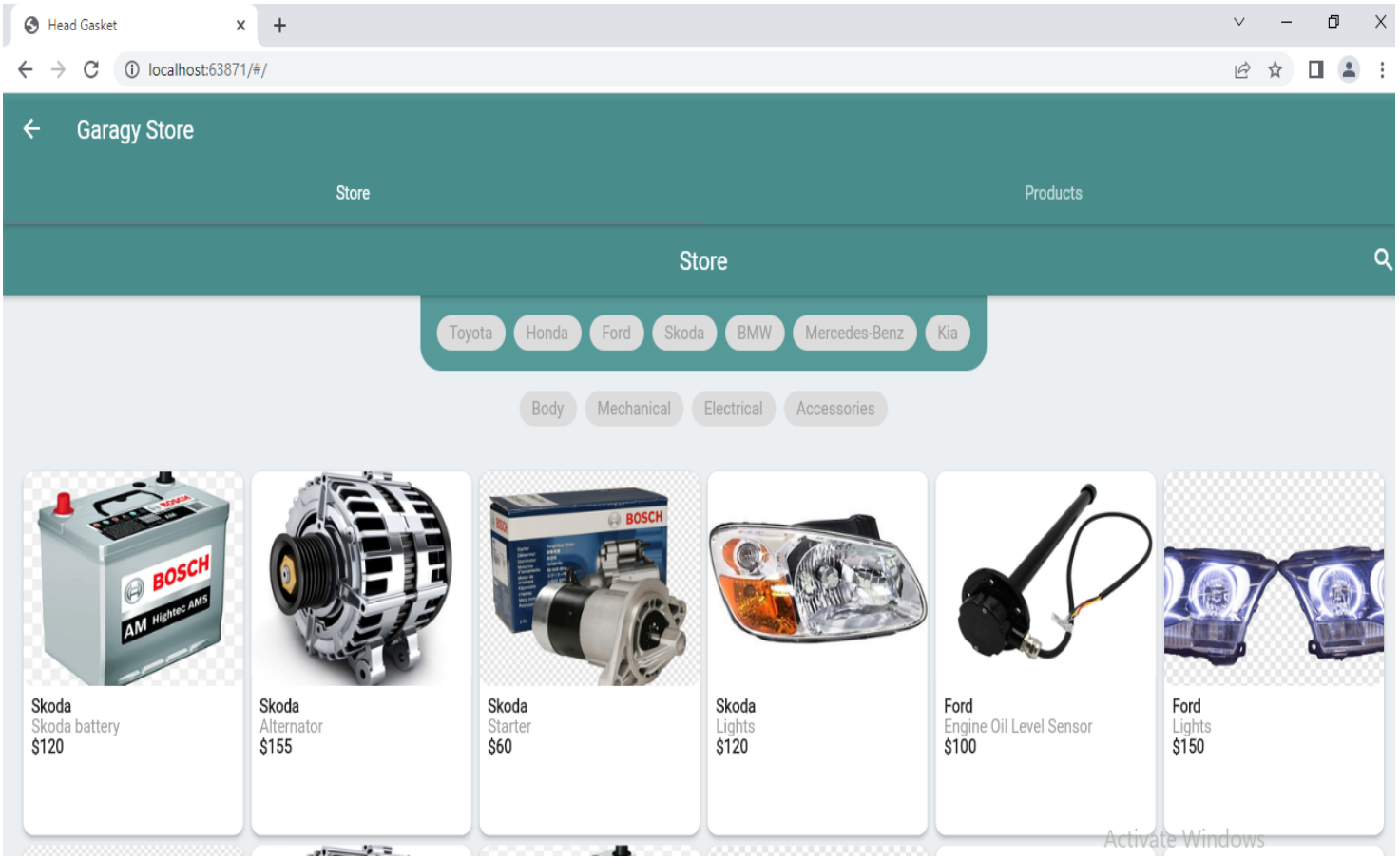


Figure 61.Products page

- **Note :**when the admin click on any one of the products he can edit or delete as shown below:*Figure 62*

Figure 62

- **Edit Product page:** in this page the admin updates and modifies the product such as name , price and type. .He can also upload or update the product's picture .The fields will have an initial value with the product data stored in the server .*Figure 63*

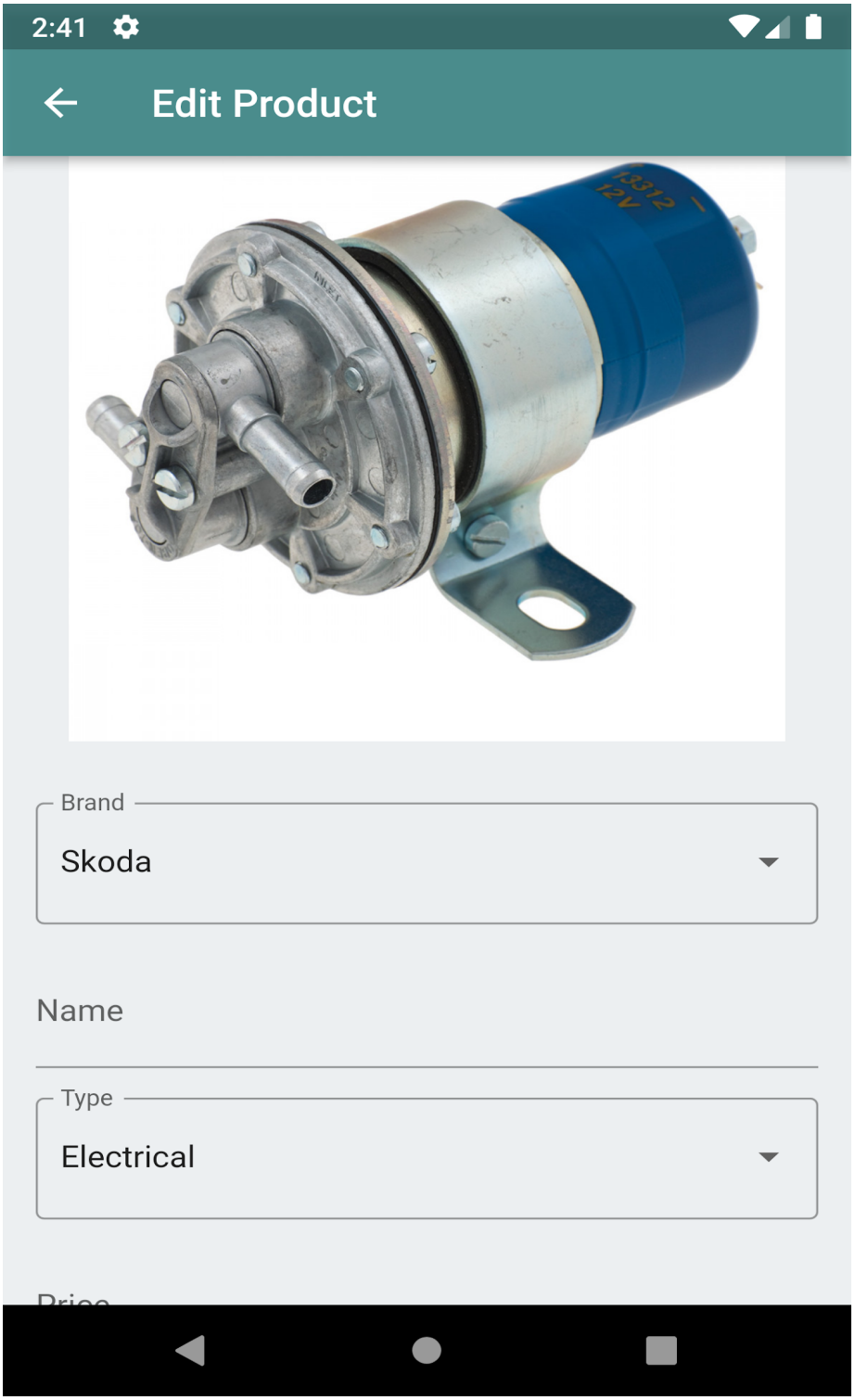


Figure 63.Edit Product page

- **Delete Product:** if the admin deletes the order this window will appear to confirm the delete.*Figure 64*

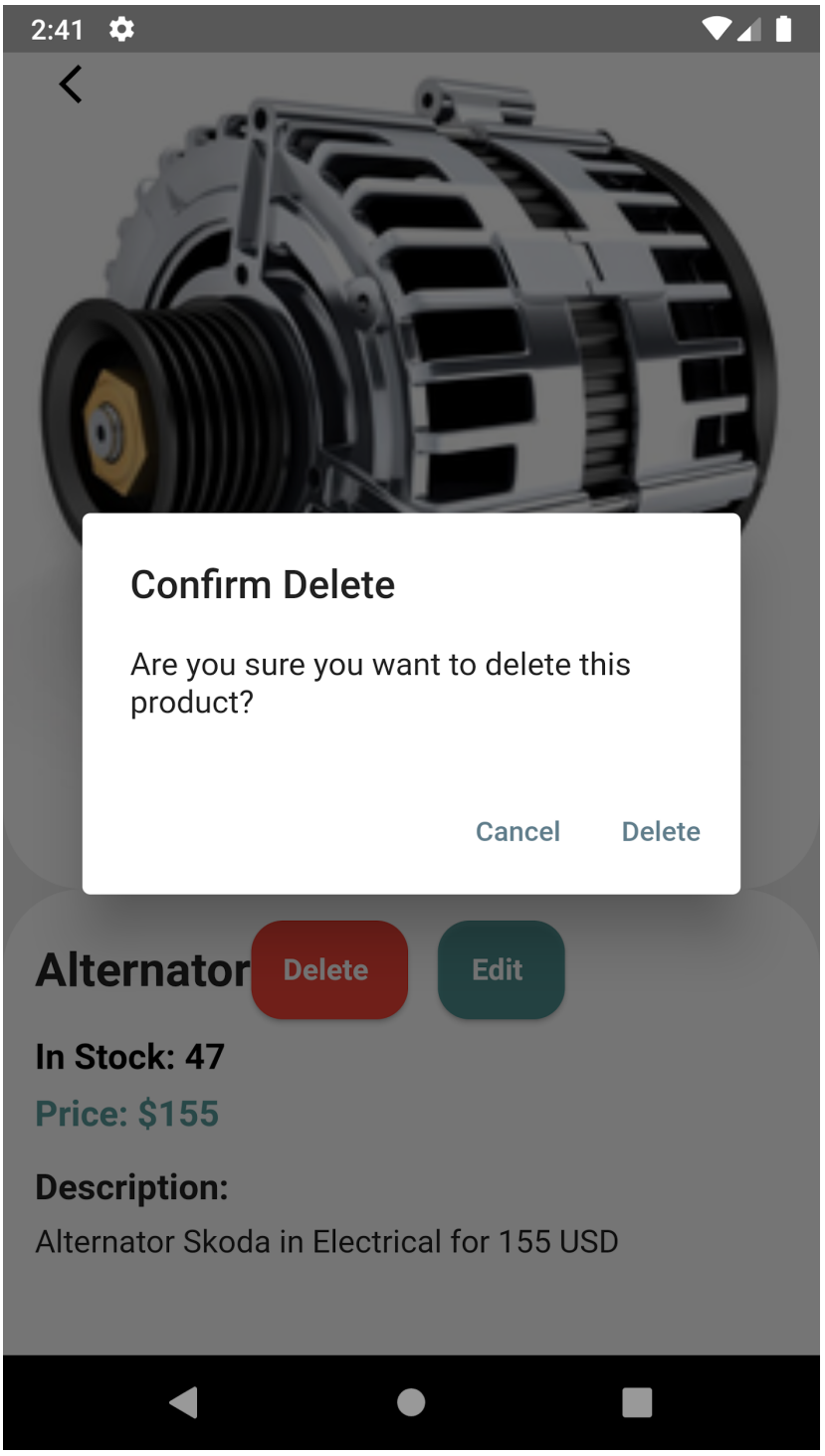


Figure 64.Delete Product

- **Add Product Page:** in this page the admin can add a new product. As shown in *Figure 65*

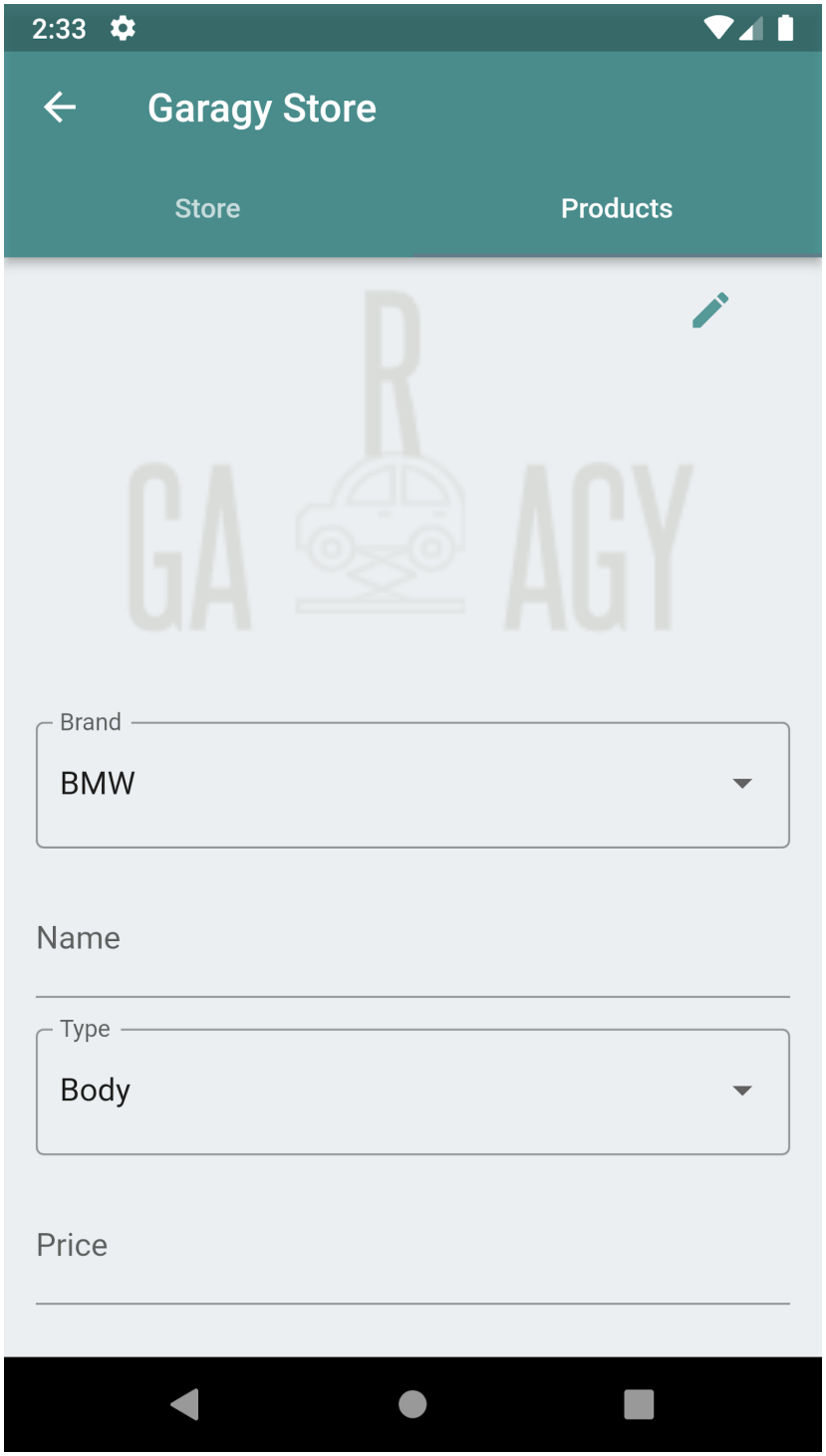


Figure 65.Add Product Page

- **Car Brands and Models Page:** this page displays all the car brands and models.Admin can add and delete.*Figure 66*

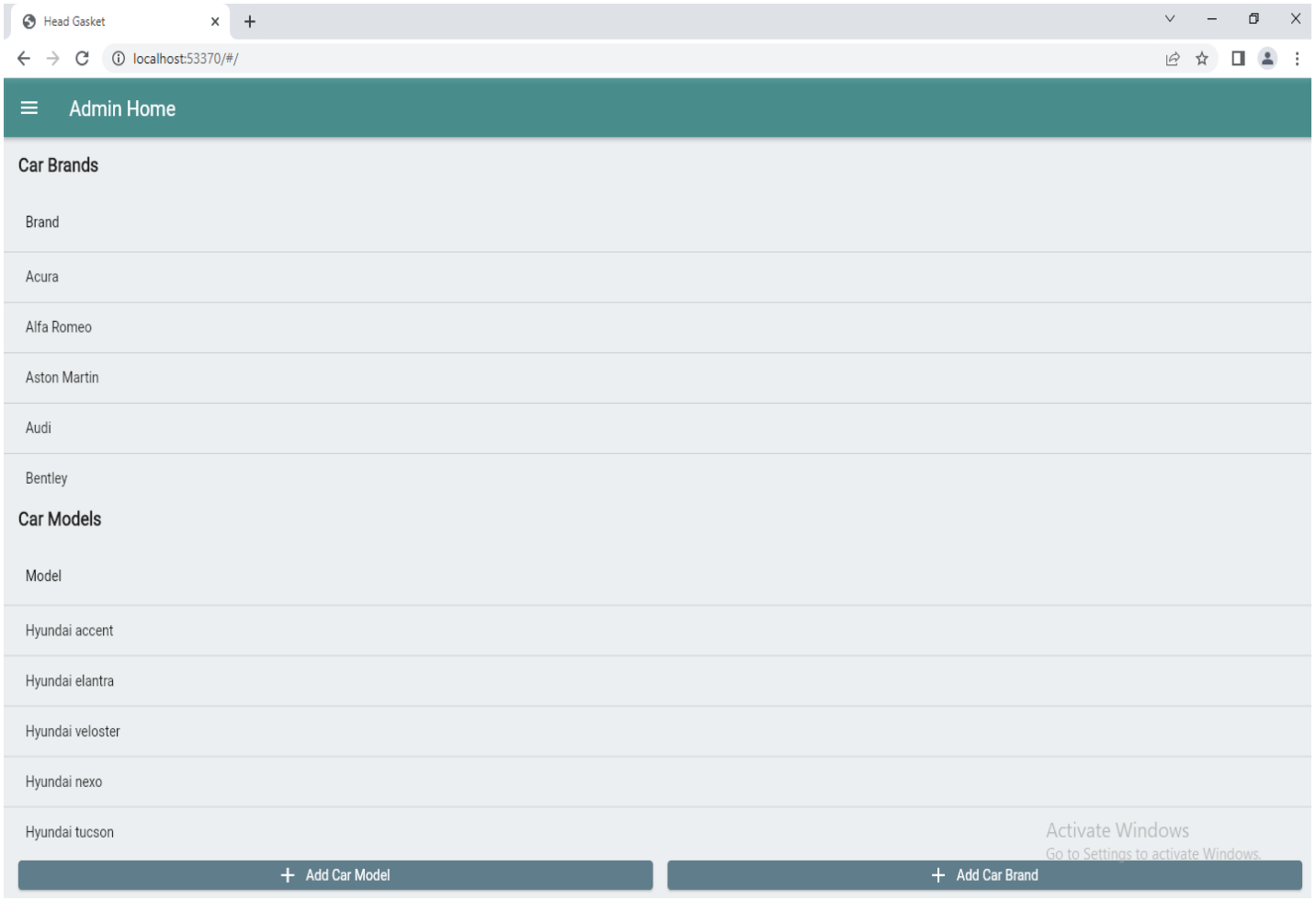


Figure 66.Car Brands and Models Page

- **Add Car Model :**

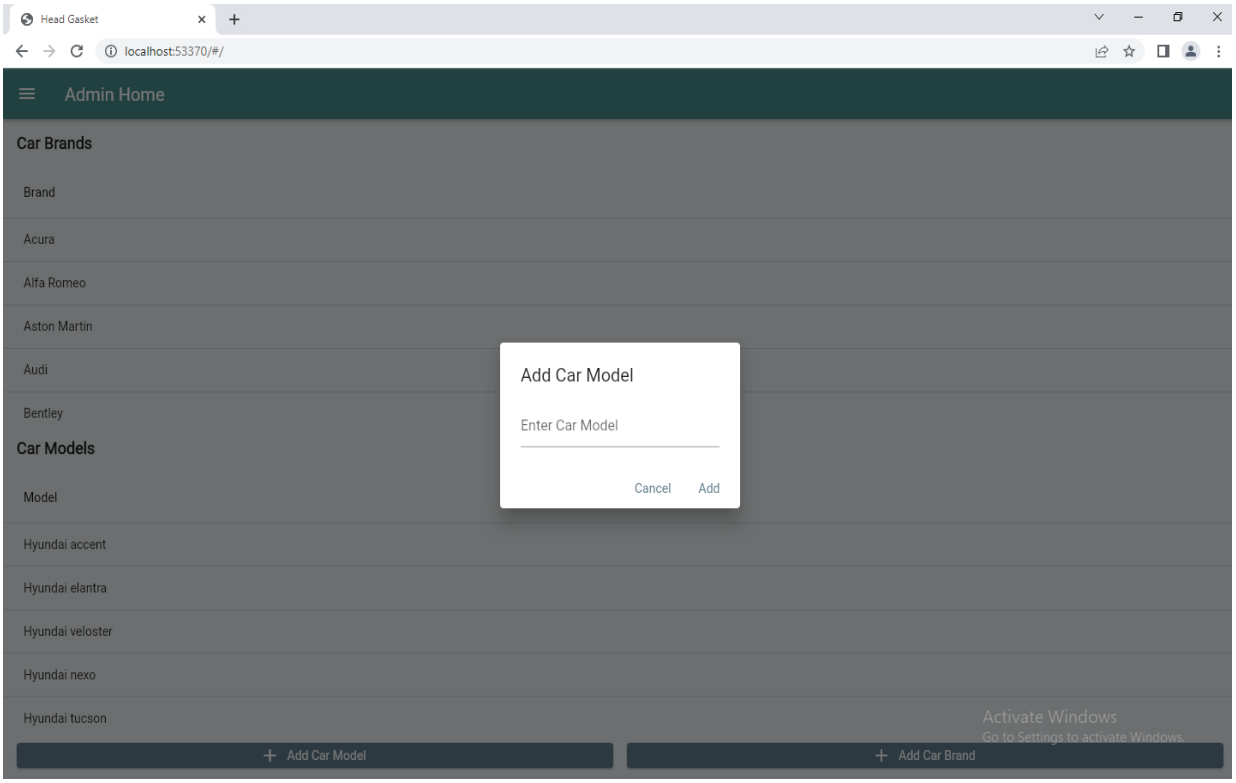


Figure 67.Add Car Model

- **Add Car Brand :**

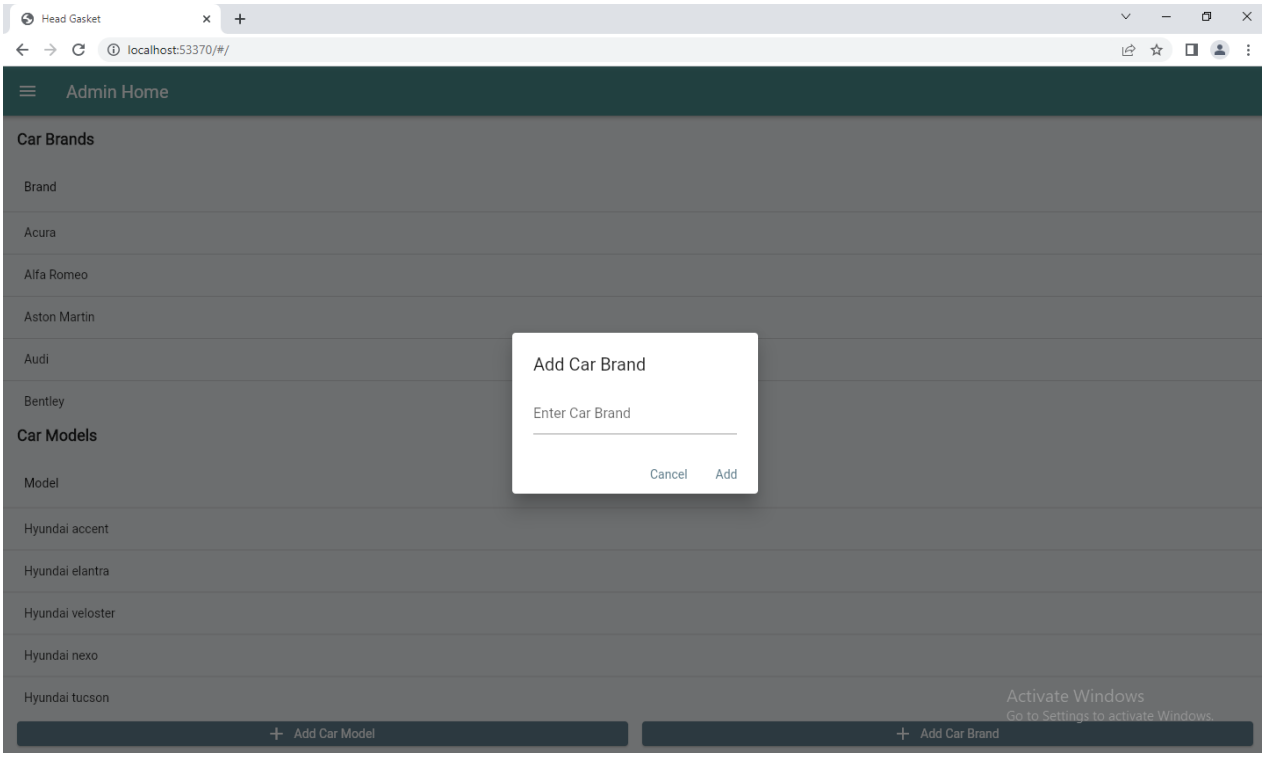
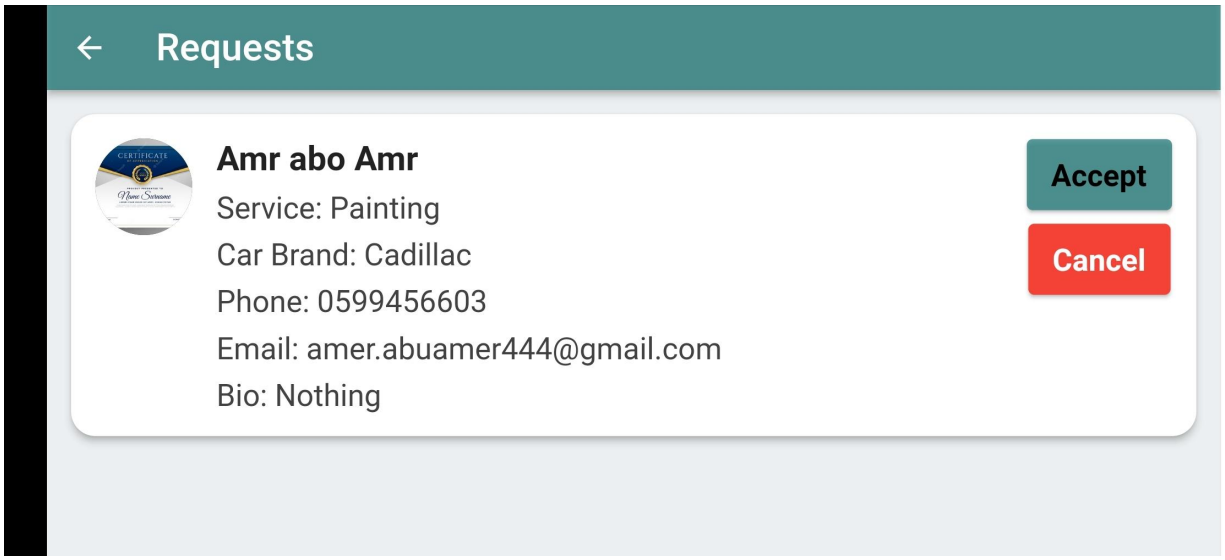


Figure 68.Add Car Brand

- **Requests page ：**
  In this page the admin will review the join as service provider page



Chapter 5: Results & Discussion

We used a variety of tools and languages, e.g. flutter and NodeJS, as well as two databases,The main one is MongoDB and Firebase for notification and chat .
Here,the impact of our app on the overall car maintenance services and process is discussed.

The app's effectiveness in improving user convenience , simplifying service provider selection , and enhancing the communication between clients and service providers , the app has simplified the process of finding,booking , and managing car maintenance services. Additionally , the impact on service provider efficiency and the overall customer experience.

Chapter 6: Conclusion

6.1 Summary:

Our main goal throughout the project was to create comprehensive car maintenance software that would facilitate user access to car maintenance services and connect them with trusted service providers. We aimed to improve users' ease, effectiveness, and overall experience with car maintenance.

We successfully developed a user-friendly and easy platform that allows customers to simply browse and select from a variety of car maintenance services through the development of the app. The addition of geolocation technology allows consumers to locate local service providers, increasing convenience. The addition of a store for parts and accessories expands the app's functionality, giving users **one place to go for all of their car maintenance needs.**

6.2 Learned Subjects:

- Flutter cross-platform.
- Rest API.
- MongoDB.
- Git and GitHub.
- Node js.
- Testing endpoints APIs using postman.

6.3 Future Work:

There are several avenues for the future improvements includes

- **Expansion of service coverage :** the app focuses on car maintenance and care services , however, there is potential to expand the scope to include Car modification and sale.

- **Integration with Smart Car Technologies :** With the rise of smart car technologies, integrating the app with car systems and sensors can provide real-time data about the car's condition and maintenance requirements.

- **Advanced Analytics and Reporting:**Implementing advanced analytics and reporting features within the app may give useful insights into user behavior, service provider performance, and overall app usage.
- **Automation of Joining Process based on Certificates:** We plan to enhance the join as a service provider process by implementing an automated acceptance system based on certificates. Currently, the approval of service provider requests is manually reviewed by our adminto ensure the authenticity and credibility of each applicant.

References

- Freitas, E. (2019). Flutter succinctly. *Morrisville: Syncfusion Inc*, *55*.
- Dagne, L. (2019). Flutter for cross-platform App and SDK development.
- *Mongoose v7.2.1: Schemas*. (n.d.). Mongoose v7.2.1: Schemas. https://mongoosejs.com/docs/guide.html
- *Welcome to the MongoDB Documentation*. (n.d.). MongoDB Documentation. https://www.mongodb.com/docs
- *Razorpay 3.0.2*. (n.d.). NuGet Gallery | Razorpay 3.0.2. https://nuget.org/packages/Razorpay/
- *flutter_map | Flutter Package*. (n.d.). Dart Packages. https://pub.dev/packages/flutter_map
- *Drivvo App*. (n.d.). Drivvo App. https://www.drivvo.com/en
- *Auto Care 1*. (n.d.). Auto Care 1. https://www.xper2solutions.com/autocare1.html
- *CARFAX Car Care: The Best Car Maintenance App for Service Alerts*. (n.d.). CARFAX Car Care: The Best Car Maintenance App for Service Alerts. https://www.carfax.com/Service/