An-Najah National University

Faculty of Engineering & Information Technology Department of

Computer Engineering

Graduation Project II

E-commerce application.
Food & Delivery & Management

Done by:
Lama thaer kharaz

Supervised by:
Dr.Asmaa Afeefi

Presented in partial fulfillment of the requirements for
Bachelor Degree in Computer Engineering

Table of content

Acknowledgment

First of all, I would like to thank everyone who contributed to the success of this project and was the main factor in achieving it.

Special mention to my family, my main support from the beginning of my university journey to this moment and forever, my friends who were my strong wall when I needed to lean on, I thank the computer engineering department at my university for their care and support.

Special thanks to Dr.Asmaa Afeefi who was always on time when I needed any help and enquiry.

1. Abstract

Life technology has fundamentally changed the way many companies operate. For some restaurant companies, technology has had a huge impact and is one of the top marketing tools that companies rely on.

The market share of restaurant apps has grown exponentially in recent years, which has increased the demand for mobile programming apps for restaurants.
Designing and programming a restaurant app helps to boost its presence in the market and at the same time promote its brands online.

This project allows online restaurants to sell food on more professional platforms and Also allows the customer to view the many foods available within the store.

This project provides easy access to customers through dynamic chat and the use of tracking features through google maps.
In addition to their ability to benefit from the opinions of other customers through comment(feed back).

And also the presence of the notifications feature enables them to see all news.
And the presence of the recommendation feature that shows the customer the request that he desires or requests frequently.

We will use flutter - front end to launch the app on platforms that support Android and IOS. - PHP for background and Firebase for notifications.

2. Introduction

After the period of the spread of the Corona virus and the home quarantine, people started to prefer to request what they needed from different applications and websites while they were in their homes, and most people continued this habit even after returning to normal life.

In Palestine the problem of unemployment increased due to covid. Many shop owners faced a huge problem increasing their sales and maintain a reliable source of income. Because of all these recent problems Online shopping became a life saver for these people.

Anyone nowadays can create an online store using social media platforms. But using such method is exhausting for both shop owners and customers. In the other hand customers find it annoying to not being able to see all products in the store with their descriptive details and hate wasting time waiting the shop owner to reply to request of any kind.

Therefore, a restaurant project was created to give customers the advantage of browsing meals through an application.

Initially, the application helps customers to view meals from the restaurant, which gives them the advantage of choosing meals in an easier way, instead of going to the restaurant. In addition, customers can easily compare and view meals, and select the most suitable meal for them. Also, it can keep track of all restaurant releases without you having to keep checking. This gives app customers the best shopping experience available in the market.

It also provides an additional application called Restaurant Management. Restaurant owner can easily add/delete/modify meals and all changes will be automatically applied to the application database.

In addition to the process of adding, deleting or modifying the worker's personal data. and, the store owner will have a detailed list of the orders received from the customers of the app once they have placed their order. So that when receiving orders from customers, he will choose the appropriate delivery service that will deliver the order.

In addition, there is a process of adding, deleting, or modifying delivery information, so that each delivery person, upon adding it, has a personal account that shows him all orders issued under his name, so that each order shows him the person's information in terms of his name, phone number, and complete order details, and the website shows him Customer using Google Map (location sharing) without the need for a fixed address.

There is also a notification system that pushes notifications, for example, if the administrator adds a new rating.

There is also the idea of tracking the customer's location using GPS, so that the person delivering can track customer location without needing a fixed address to drop off the order. There is a customer communication feature, by using Dynamic Chat to answer customers to all their requests.

The general idea of this project is the already existing delivery service, but I have dedicated this application to a specific company or restaurant so that it combines the ordering application, the delivery application, and the main application of a restaurant.

In the following chapters I will talk about the limitations that we encountered during the development of this project, the training courses that helped us reach our goal in this work, discuss similar available applications and how this project differs from others, as well as the methodology, techniques and procedures that I followed to complete it, I will discuss the emerging results, and finally I will talk On conclusion, limitations, and future work.

3. Constraints, standards and earlier work

Poor internet connection was a problem because an application requires internet connection all the time due to online MYSQL database, but I increased internet speed to counter this limitation, reduced storage space of my laptop and its slow performance and lack of knowledge about flutter because this language is not mentioned in university courses.

The horrible problem I encountered is that the address of the computer is different every time depending on the network and has to be replaced every time during the project to work properly.

I used android studio and then found that it requires a lot of memory space, and the emulator needs to be replaced almost every day to run properly, which consumes memory.

I used the Flutter framework to manage the front end of the app, fetching and inserting data from the back end. So I learned how to use it through an online course, as well as the docs. Php backend, used MYSQL for databases and JSON format to pass data for some features, I learned by watching videos and searching online.

4. Literature review

Today e-commerce has become an important part of daily life. Access to e-commerce platforms is not a privilege but a necessity for most people.

The number of Internet users in the world today is close to 3 billion. This penetration of the Internet along with the increased confidence of Internet users in purchasing online, has led to an exponential growth in the field of e-commerce, with an increase in the number of customers registered in e-commerce applications and purchasing through the use of mobile phones. So, I chose to create an online food shopping app for a specific restaurant.

There are not many applications that have the same idea and the same features in Palestine. I did research before implementing the project and discovered that there is only one application that has a similar idea called Yummy, and it is an application that contains online restaurants, but what makes me better is that I specialized in a specific restaurant and added many The advantages of the application so that the user will not find it difficult to navigate through the application and find what he wants faster.

An application is a special application available in Palestine. It collects the process of purchasing food from the restaurant, and at the same time helps the restaurant owner to display his meals to customers in a more professional and organized manner.

The main reason that encouraged me to develop and work on the idea of the application is the high demand of people to buy food online and without hesitation, as well as without the need to go to the restaurant.

5. Methodology

5.1. Tools, Programming Languages, and Technologies

5.1.1. Mobile application

There are many frameworks and languages that were used to build the mobile application.

- Frameworks

We chose to use Flutter which is an open-source software development kit which enables smooth and easy cross-platform mobile app development. We can build high quality natively compiled apps for IOS and Android quickly, without having to write the code for the two apps separately. Also we use flutter for website dashboard.

- Programming Languages

The programming language we use is php because we use flutter as the framework. php is easy to understand and has extreme documentation so we can easily fix and debug any errors that appear, it is the most lightweight and most popular programming language in the world.

5.1.1. Application Dashboard

- Framework
  We used flutter for the frontend for backend
- Languages
  We used dart for frontend and php for backend

5.1.2. **Database**

We use MYSQL, which is an open source relational database management system. It is more secure, scalable, capable of handling large amounts of data, and easy to use with PHP in operations that happen on tables.

Full Database:

	Table	Action	Rows	Type	Collation	Size	Overhead
☐	admin	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	bill	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	category	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
☐	comment	Browse Structure Search Insert Empty Drop	16	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	customer	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_bin	32.0 KiB	-
☐	delivery	Browse Structure Search Insert Empty Drop	6	InnoDB	armscii8_bin	16.0 KiB	-
☐	detail_bill	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_bin	16.0 KiB	-
☐	favorite	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	food	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8_bin	32.0 KiB	-
☐	user	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-

Figure 1: Database

Database For Application :

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**use_id** 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**use_name**	varchar(250)	utf8mb4_unicode_ci		No	*None*			Change Drop More
☐	3	**use_mobile**	int(11)			No	*None*			Change Drop More
☐	4	**use_pwd**	varchar(250)	utf8mb4_unicode_ci		No	*None*			Change Drop More
☐	5	**use_datetime**	datetime			No	*None*			Change Drop More
☐	6	**use_active**	tinyint(4)			No	*None*			Change Drop More
☐	7	**use_token**	varchar(500)	utf8mb4_bin		No	*None*			Change Drop More
☐	8	**use_lastdate**	datetime			No	*None*			Change Drop More
☐	9	**use_note**	varchar(500)	utf8mb4_bin		No	*None*			Change Drop More

Figure: Employee

☐	2	**cus_name**	varchar(250)	utf8mb4_unicode_ci	Yes	*NULL*	Change	Drop	More
☐	3	**cus_pwd**	varchar(250)	utf8mb4_unicode_ci	Yes	*NULL*	Change	Drop	More
☐	4	**cus_mobile**	varchar(250)	utf8_bin	Yes	*NULL*	Change	Drop	More
☐	5	**cus_email**	varchar(250)	utf8mb4_unicode_ci	Yes	*NULL*	Change	Drop	More
☐	6	**cus_regdate**	datetime		Yes	*NULL*	Change	Drop	More
☐	7	**cus_block**	int(11)		No	*None*	Change	Drop	More
☐	8	**cus_token**	varchar(500)	utf8_bin	Yes	*NULL*	Change	Drop	More
☐	9	**cus_address**	varchar(1000)	utf8mb4_unicode_ci	Yes	*NULL*	Change	Drop	More
☐	10	**cus_lan**	varchar(150)	utf8_bin	Yes	*NULL*	Change	Drop	More
☐	11	**cus_lat**	varchar(150)	utf8_bin	Yes	*NULL*	Change	Drop	More
☐	12	**cus_note**	varchar(1000)	utf8mb4_unicode_ci	Yes	*NULL*	Change	Drop	More
☐	13	**cus_thumbnail**	varchar(250)	utf8_bin	Yes	*NULL*	Change	Drop	More
☐	14	**cus_image**	varchar(250)	utf8_bin	Yes	*NULL*	Change	Drop	More
☐	15	**cus_notificatefood**	varchar(250)	utf8_bin	No	*None*	Change	Drop	More

Figure: Customer

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	cat_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	cat_name	varchar(250)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
☐	3	cat_name_en	varchar(250)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
☐	4	cat_regdate	date			Yes	NULL			Change Drop More
☐	5	cat_thumbnail	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	6	cat_image	varchar(1500)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More

Figure: Category

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	foo_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	foo_name	varchar(250)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	3	foo_name_en	varchar(250)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	4	foo_regdate	datetime			No	None			Change Drop More
☐	5	foo_image	varchar(250)	utf8_bin		No	None			Change Drop More
☐	6	foo_thumbnail	varchar(250)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	7	foo_price	int(11)			No	None			Change Drop More
☐	8	foo_offer	int(11)			No	None			Change Drop More
☐	9	foo_info	varchar(1000)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	10	foo_info_en	varchar(1000)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	11	cat_id	int(11)			No	None			Change Drop More

Figure: Food

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	bil_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	cus_id	int(11)			No	None			Change Drop More
☐	3	del_id	int(11)			No	None			Change Drop More
☐	4	bil_regdate	datetime			No	None			Change Drop More
☐	5	bil_address	varchar(1000)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	6	bil_before_note	varchar(1000)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	7	bill_rate	varchar(1000)	utf8mb4_bin		No	None			Change Drop More
☐	8	bil_after_note	varchar(1000)	utf8mb4_unicode_ci		No	None			Change Drop More

Figure: Bill

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**det_id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**foo_id**	int(11)			No	*None*			Change Drop More
☐	3	**det_price**	int(11)			No	*None*			Change Drop More
☐	4	**det_qty**	int(11)			No	*None*			Change Drop More
☐	5	**bil_id**	int(11)			No	*None*			Change Drop More

Figure: details Bill

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**foo_id**	int(11)			No	*None*			Change Drop More
☐	2	**cus_id**	int(11)			No	*None*			Change Drop More
☐	3	**fav_regdate**	datetime			Yes	*NULL*			Change Drop More
☐	4	**fav_id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More

Figure: Favorite

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**com**	varchar(250)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	2	**foo_id**	int(11)			No	*None*			Change Drop More

Figure: Comment

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**del_id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**del_name**	varchar(250)	armscii8_bin		Yes	*NULL*			Change Drop More
☐	3	**del_mobile**	varchar(250)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop More
☐	4	**del_pwd**	varchar(250)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop More
☐	5	**del_regdate**	datetime			Yes	*NULL*			Change Drop More
☐	6	**del_lastdate**	datetime			Yes	*NULL*			Change Drop More
☐	7	**del_thumbnail**	varchar(250)	armscii8_bin		Yes	*NULL*			Change Drop More
☐	8	**del_image**	varchar(250)	armscii8_bin		Yes	*NULL*			Change Drop More
☐	9	**del_note**	varchar(1000)	armscii8_bin		Yes	*NULL*			Change Drop More

Figure: Delivery

5.1.3. IDEs and Code Editors

As a free source code editor, we used Android Studio for developing the application front end, it supports dart language, Android Studio offers build automation, dependency management, and customizable build configurations.

We used Visual Stdio, which is a cross-platform IDE for PHP, to build the administration web portal, and XAMPP for the server, which is the most popular PHP development environment.

5.2. Architecture

Our project contains 2 parts that work in syn to reach the optimal results, these parts are:

- Mobile Application
- Data and API server

5.2.1. Mobile Application

Users make different requests when they use the mobile applications, these requests vary depending on the complexity and the goal wanted from , the most used request is GET request, it is used to get specific data from the database and display it next to the client in the place provided for it. And POST request which is used to submit changes to the database.

5.3. **Implementation**

5.3.1. Mobile Application

❖ **Splash screen**

This screen consists of Loading and phrase Welcome in Resturant App.

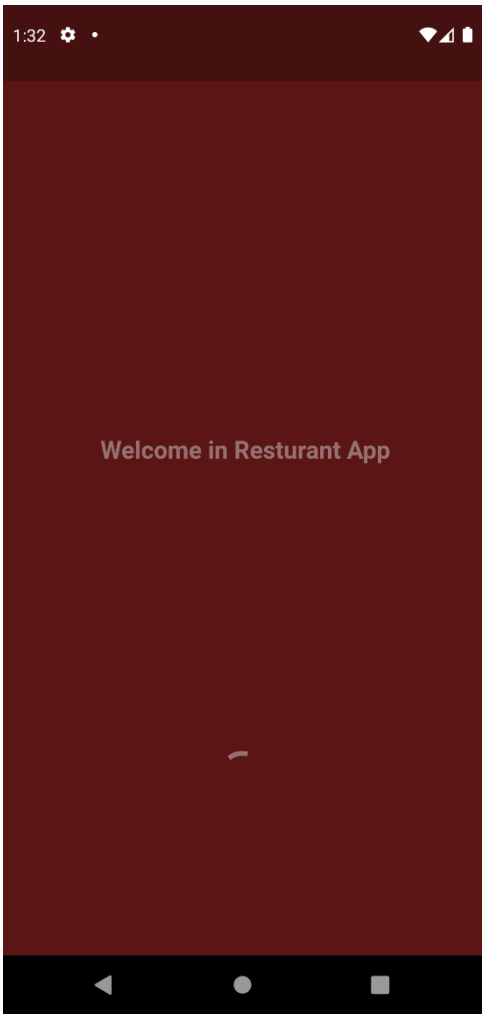


Figure : Loading Screen

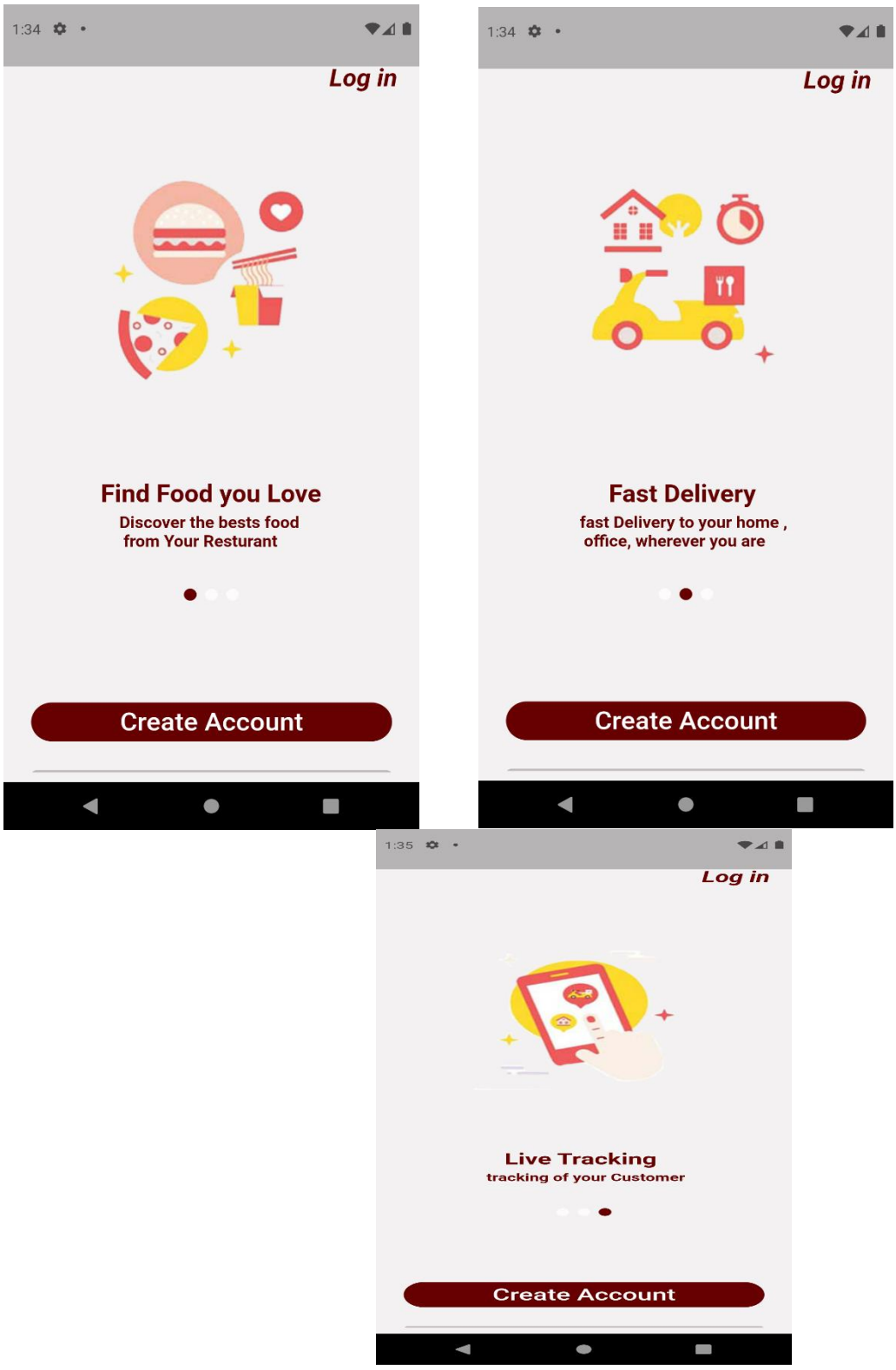


Figure : Welcome Screen

❖ Sign up Screen

In the sign-up page the customer enters the required information, which is the username, Email, password, confirm password, address and phone number.

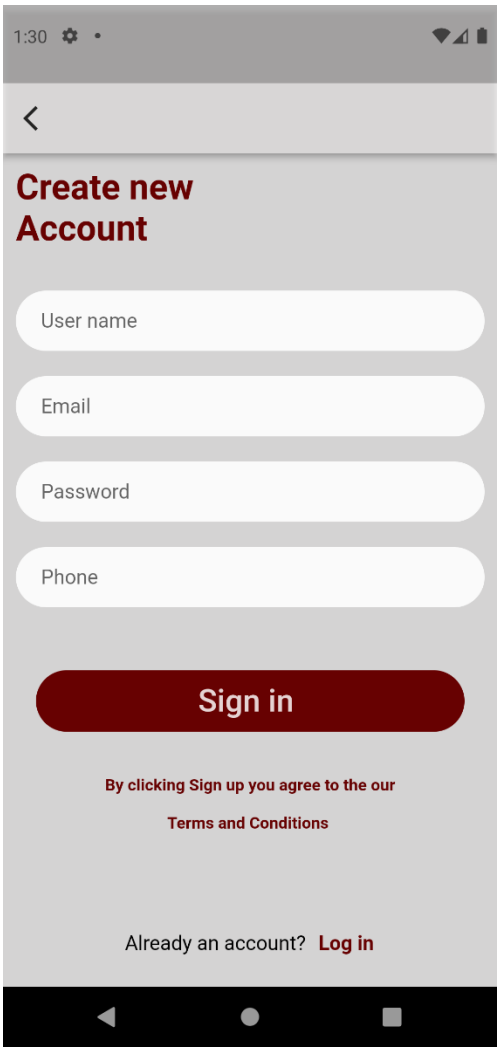


Figure : Sign up Screen

There are some limitations in the information like, Name must be 4 or more characters, Email must be valid and contain @, Password must be more 6, and Correct phone must be more 5 digit.

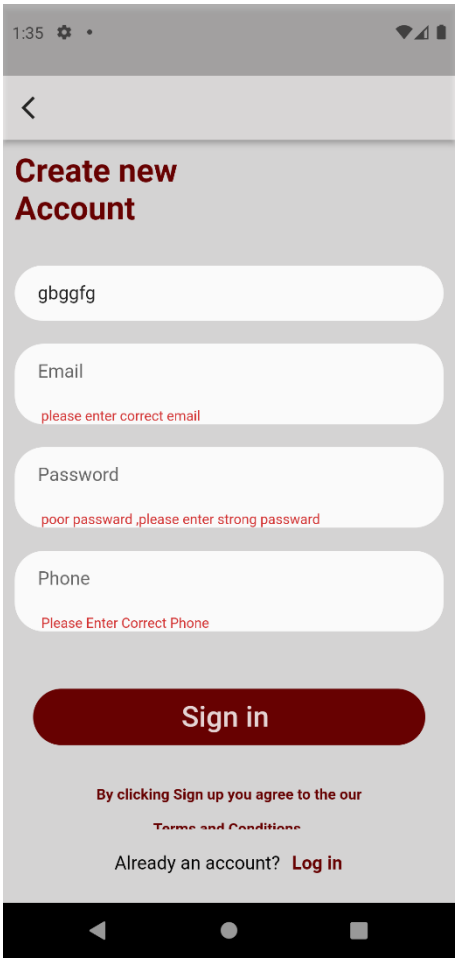


Figure : Sign up Screen

All information is required so when the user presses on the “Sign up” button if any of them is empty or password not match warning message will appear. If everything is alright the user will navigate to his profile screen.

❖ Login Screen

The user writes his Phone and his password on this screen. When he clicks the “login” button, if he writes the wrong phone or password a warning message will appear but if everything is good and his information matches our records, he will move to the home screen.

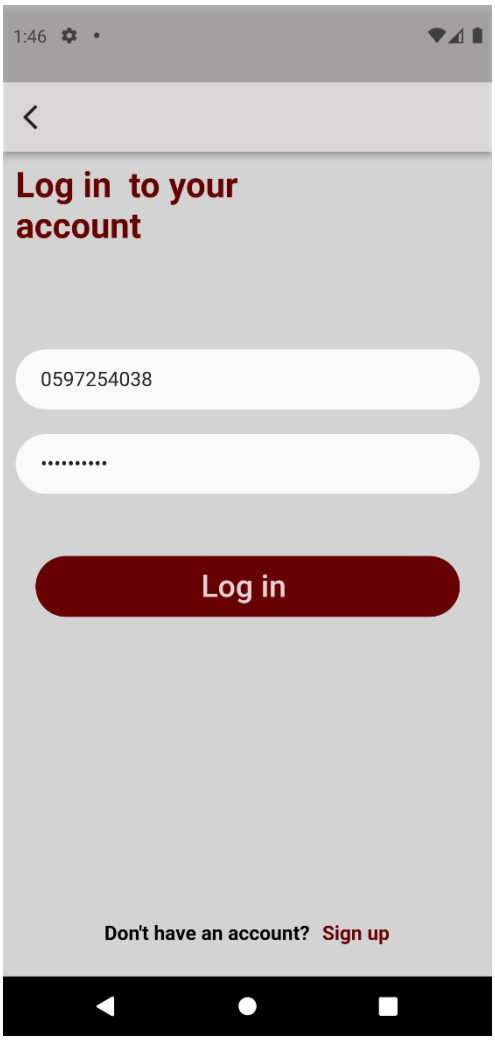


Figure: Login Screen

❖ Home Screen

In an application, an application login process is required. When the customer logs in or creates an account, he is shown this interface consisting of the items at the top, in addition to the presence of a list of offers at the bottom.

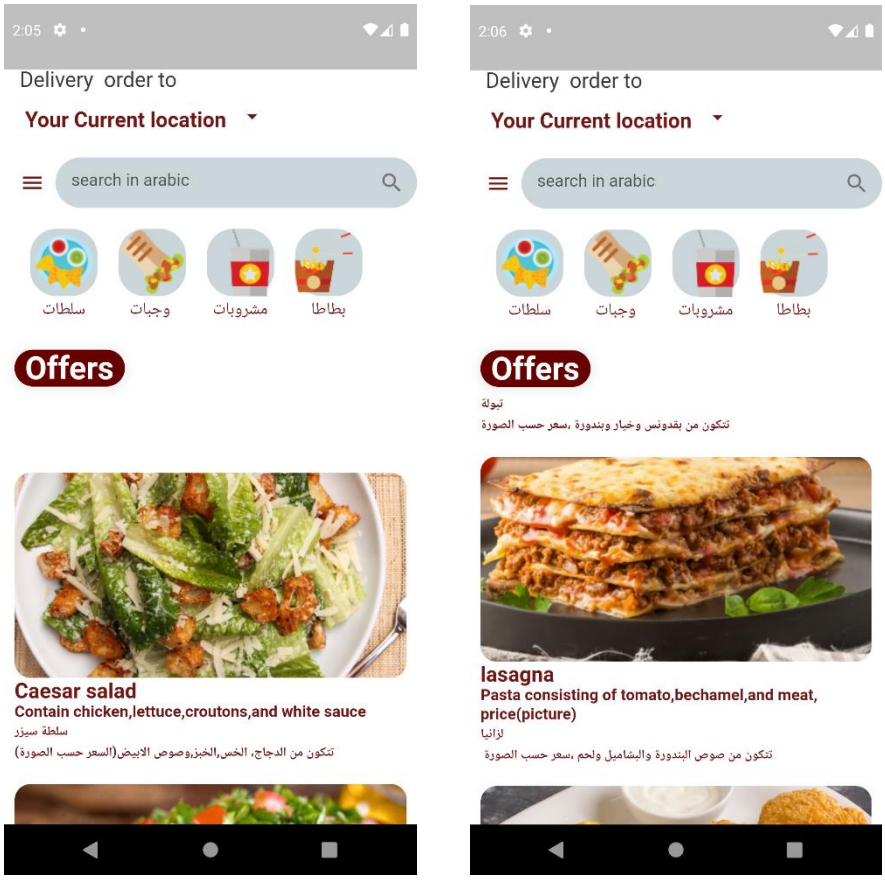


Figure : Home Screen

In this section, it shows the list of categories within the application, so that within each category there is a menu.

Figure : Category Screen

❖ Food Screen

Inside each category, there is a menu that the menu shows the meal, its name, and information about it, and there is a favorite icon if the customer wants to add it to the favorites or not, and there is an Offer icon if there Offer on the meal or not.

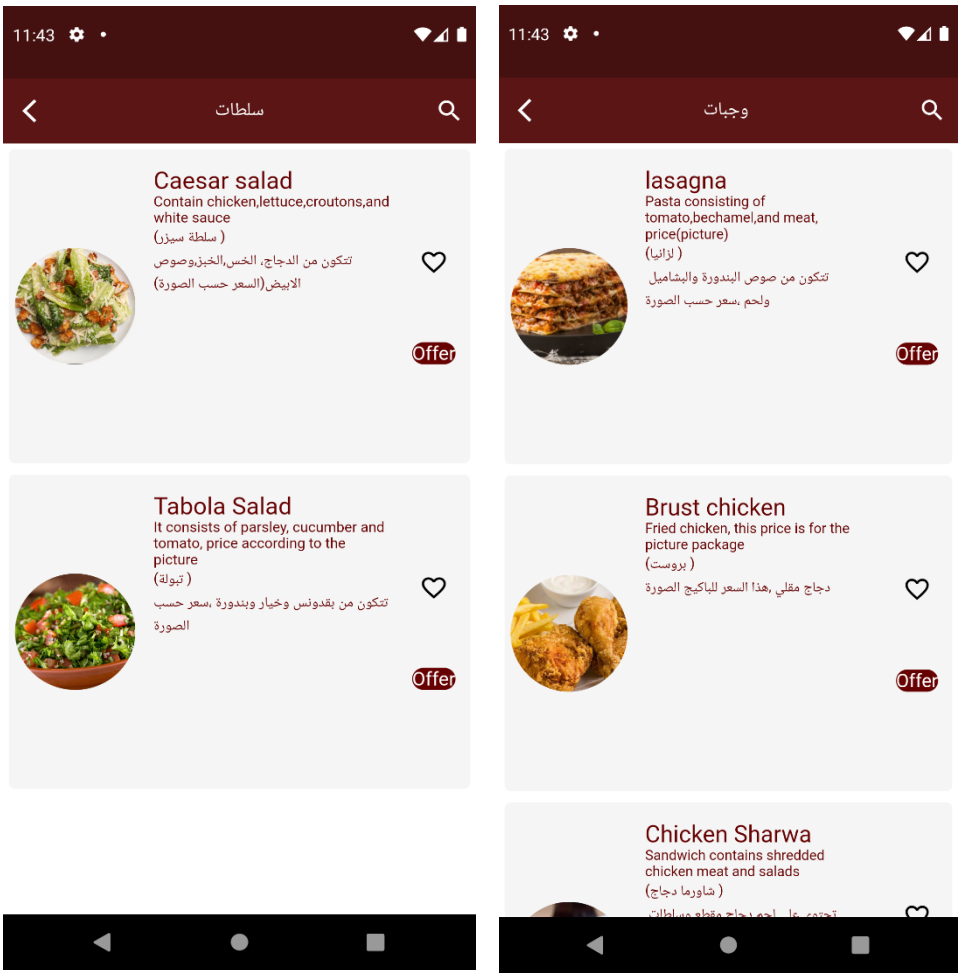


Figure : Food Screen

❖ Drawer Screen

The bedside drawer screen contains Home, Settings, My profile, category, Dark mode, Favorite, My Orders, cart, and tracking me and when you are login additional sections will appear, like cart, wish list, settings, and log out.

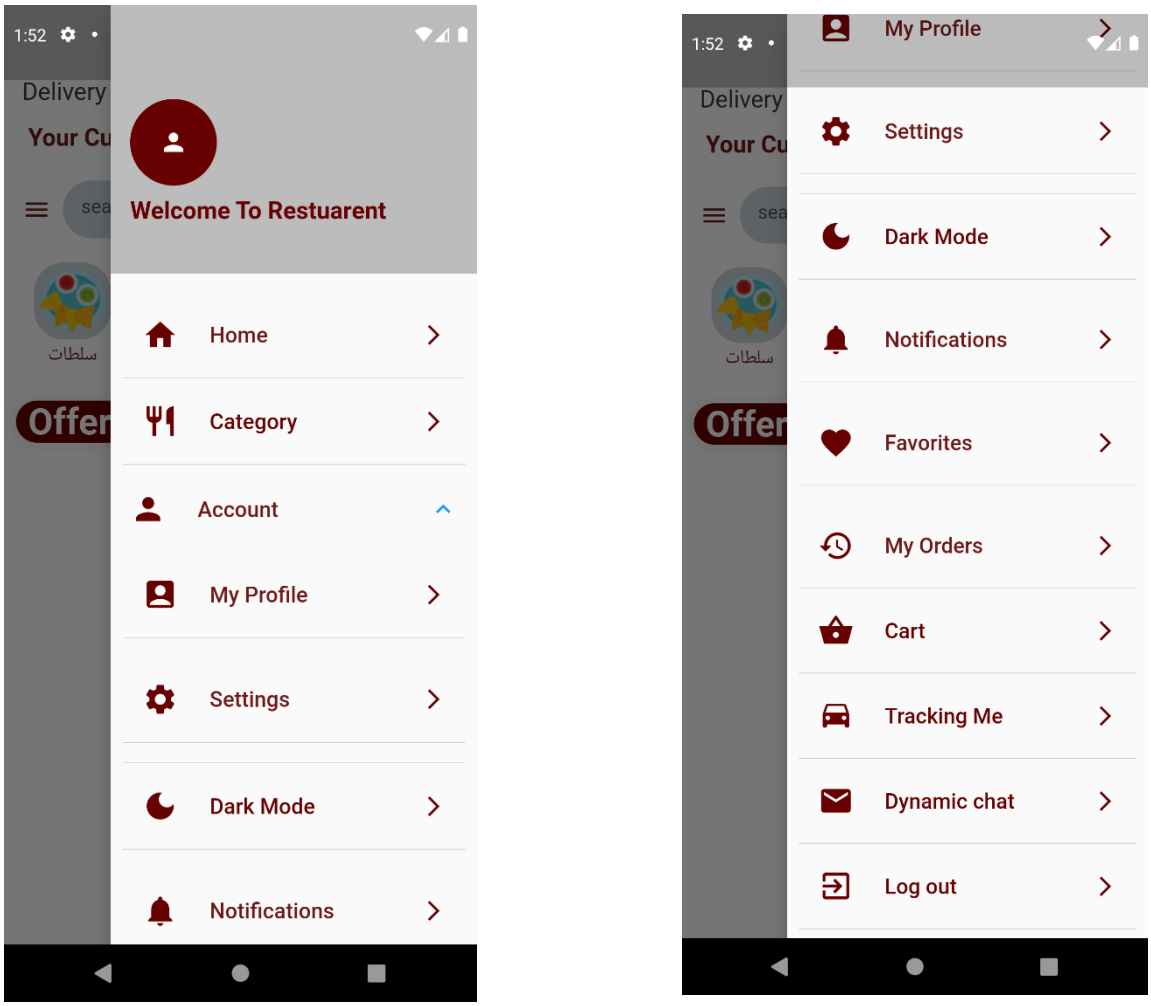


Figure: Draw Screen

❖ Meal Screen

Meal screen has all meals from different categories and stores. It has search button to help customer find his specific meals.

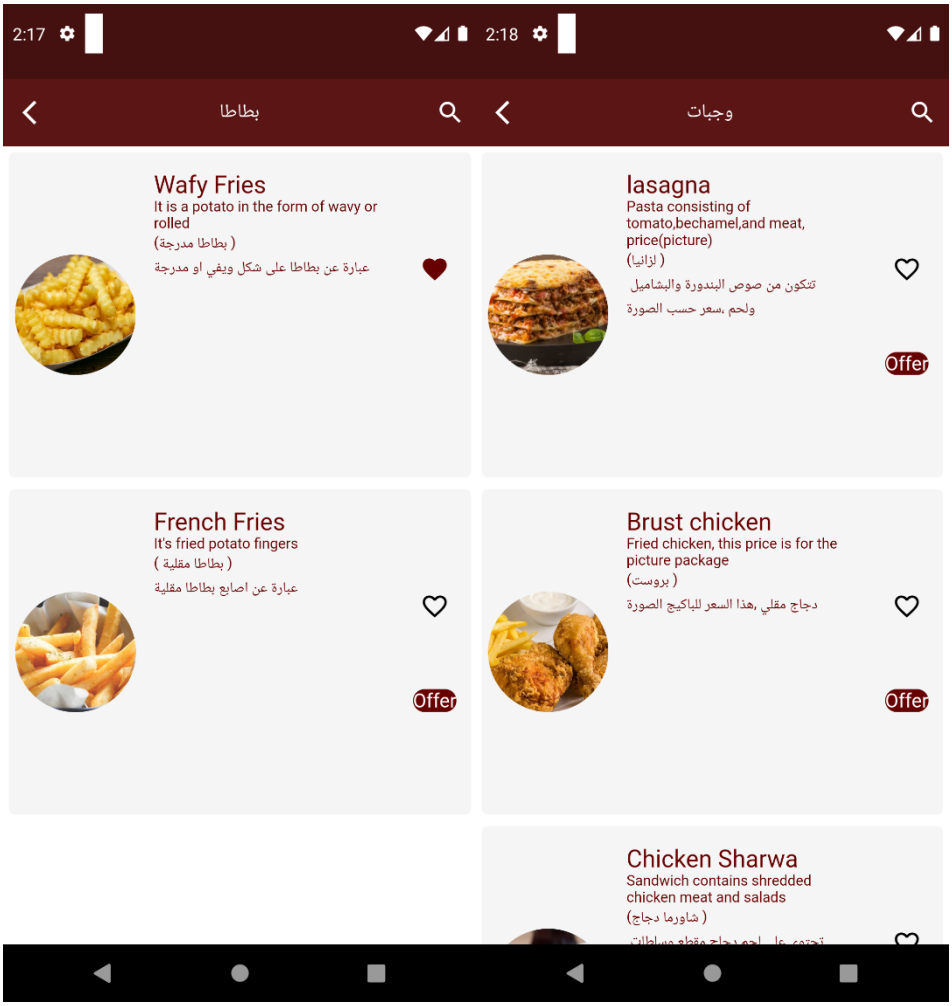


Figure: Meal Screen

❖ Search Screen

On this screen, you can search for a meal you want, on most of the pages the search idea was adopted.

Figure : Search Screen

❖ Dynamic Chat Screen

On this Screen, there is a dynamic chat feature so that the customer is given the ease to answer all his messages directly.

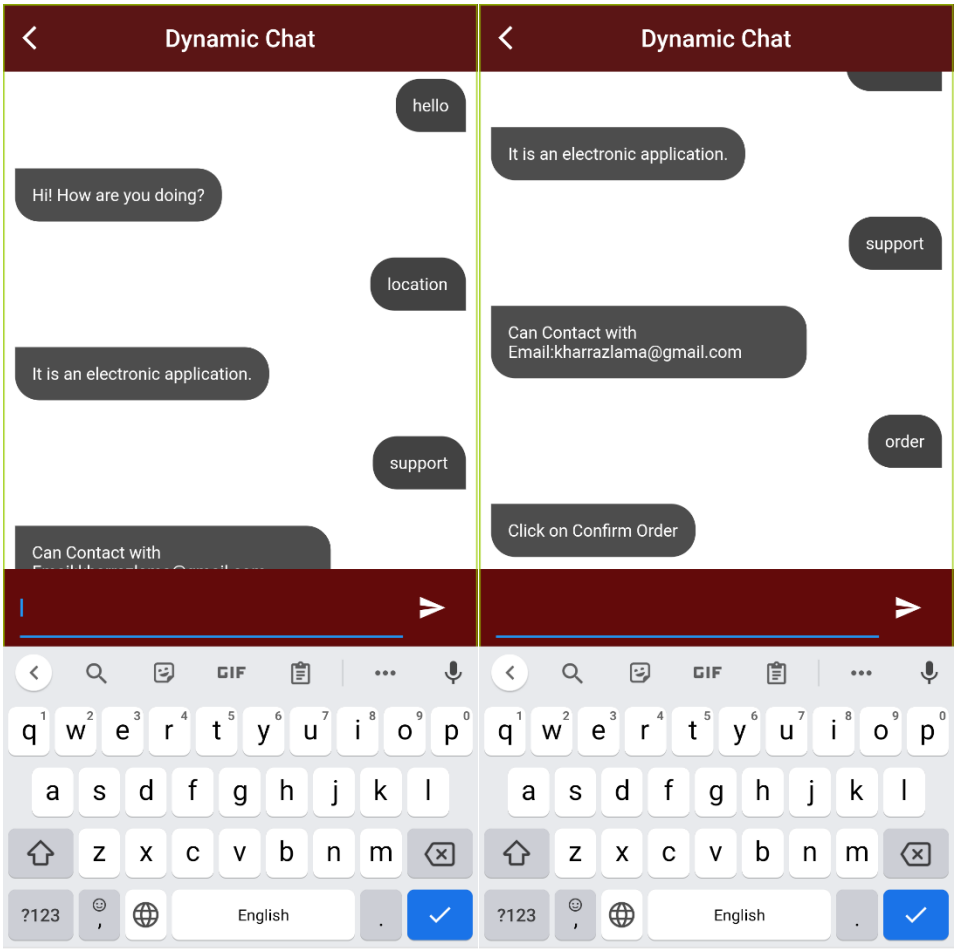


Figure : Dynamic Chat Screen

❖ Tracking Customer Screen

On this Screen, there is a location feature for the customer, so that when the location is specified, it is sent to Delivery.

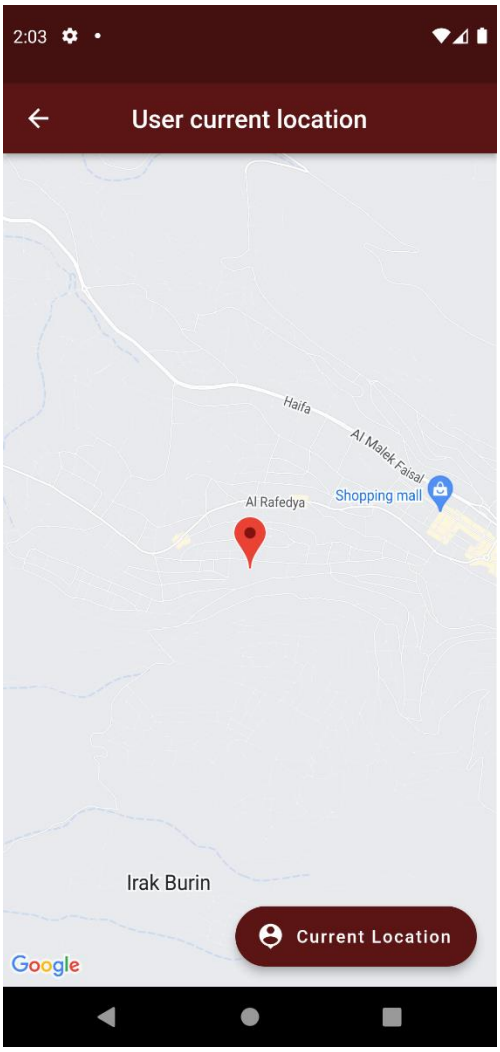


Figure : Tracking Location Customer

❖ Profile Screen

This screen contains name of customer, customer details.

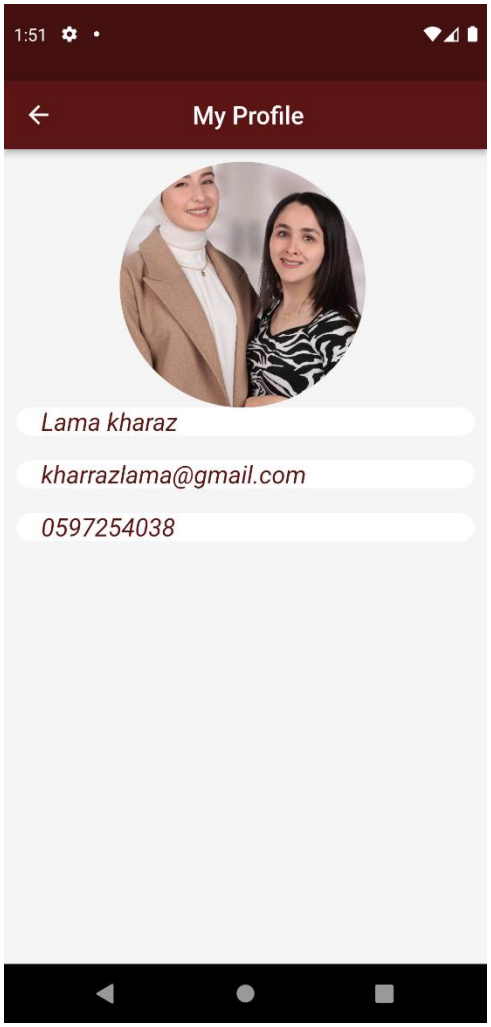


Figure : Profile Screen

❖ Customer Settings Screen

In this screen, the Customer can update any of them and then save the changes by clicking on the save button.

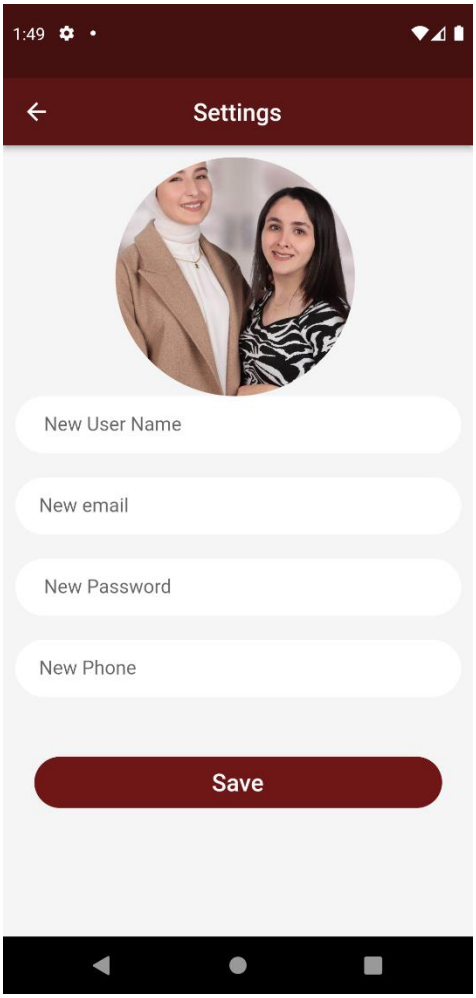


Figure: Customer information Setting Screen

❖ Food detail Screen

This screen contains all information about a specific Meal. Its name, image, price, and description.
It contains also a favorite icon to put the meal on your wish list to see it later and Add to Cart button to store the meal in your cart to buy it later after checkout all meals in your cart.

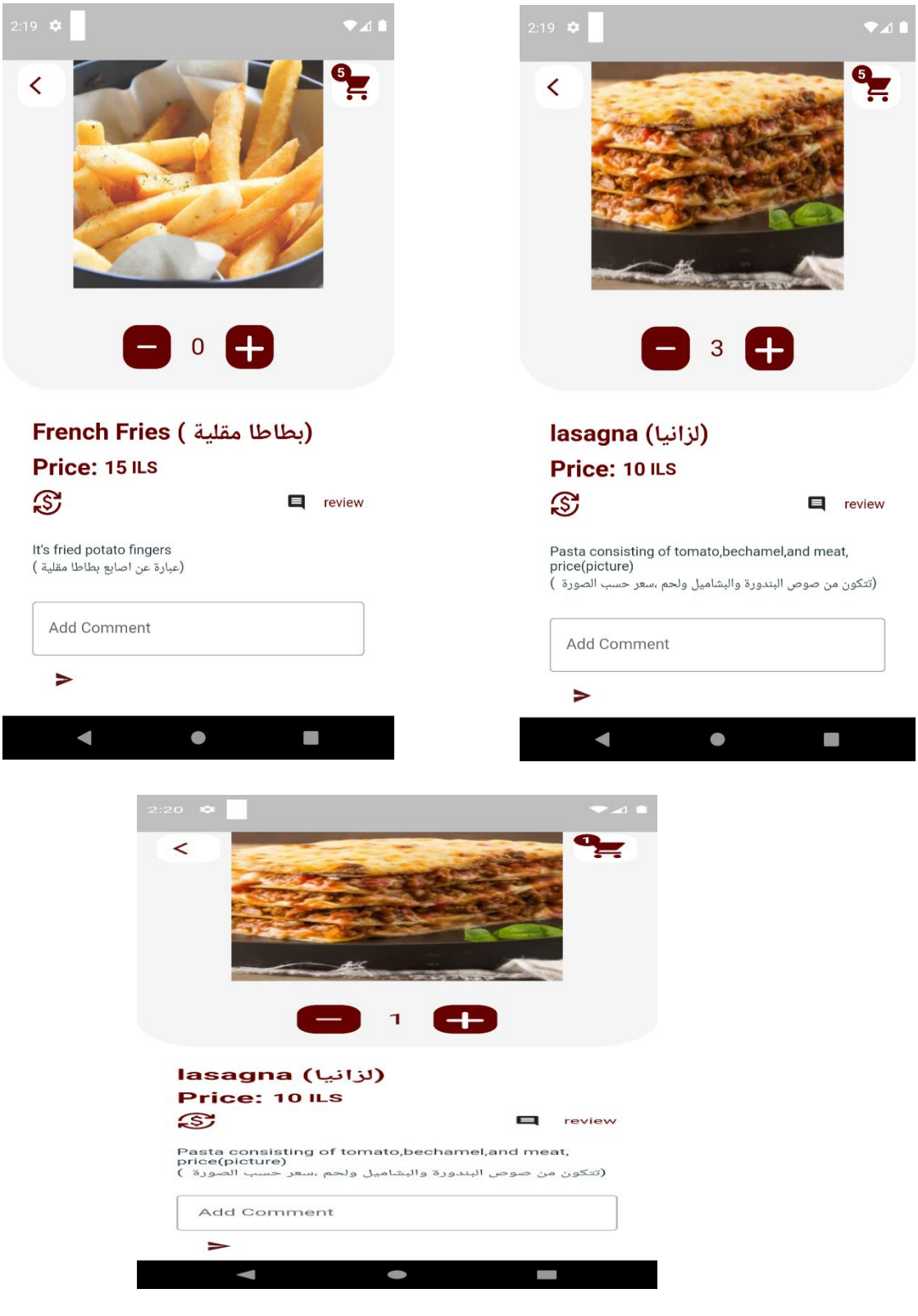


Figure: Food Details Screen

❖ Favorite Screen

The wish list screen contains all Meals added to the wish list by customers from the Meal details screen, by clicking on the favorite button icon to make its color red. Customers can delete any meal and remove it from their wish list by clicking another time on their favorite icon to make its color transparent.

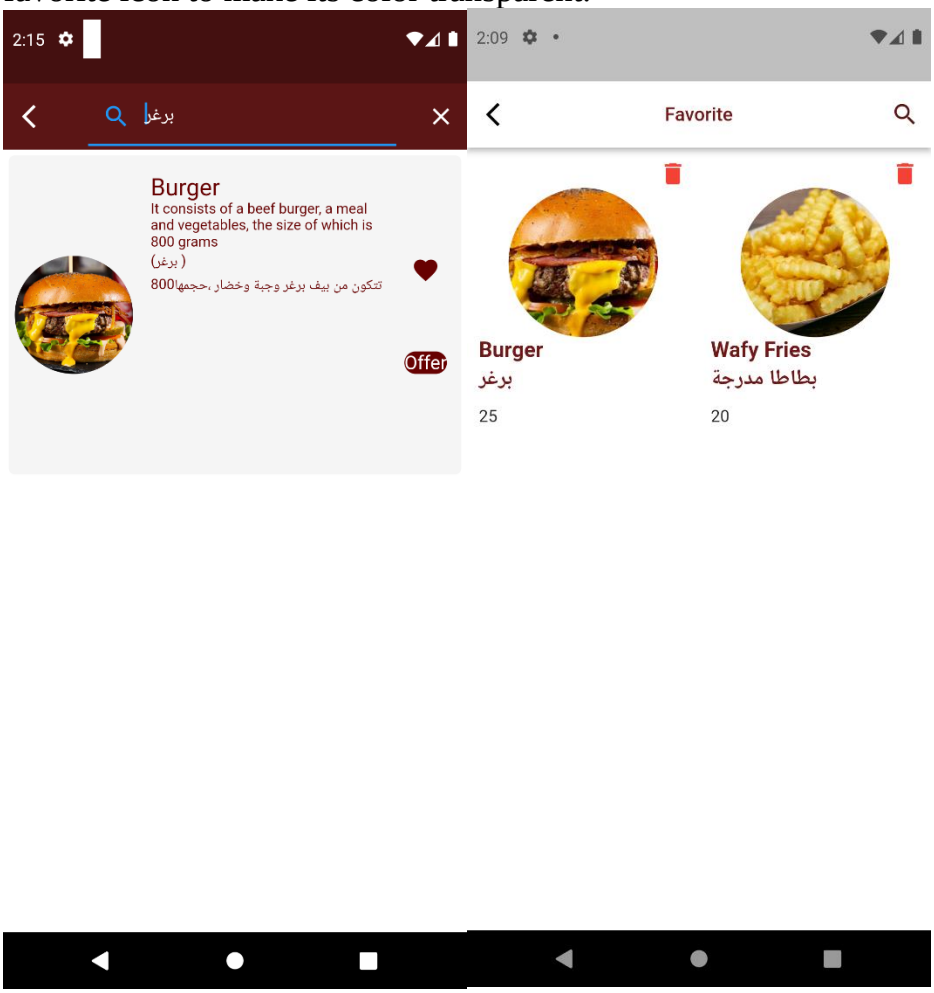


Figure: Favorite Screen

❖ Cart Screen

The cart screen contains all Meals added by the customer from the meal details screen, with the price for each meal. Customer can change the quantity of each meal by selecting add or minus icon or deleting any meal and removing it from the cart by swapping it to the left. At the bottom of the screen, there are the total price of all meals in the cart and the checkout button that navigate the customer to the checkout screen to place his order.

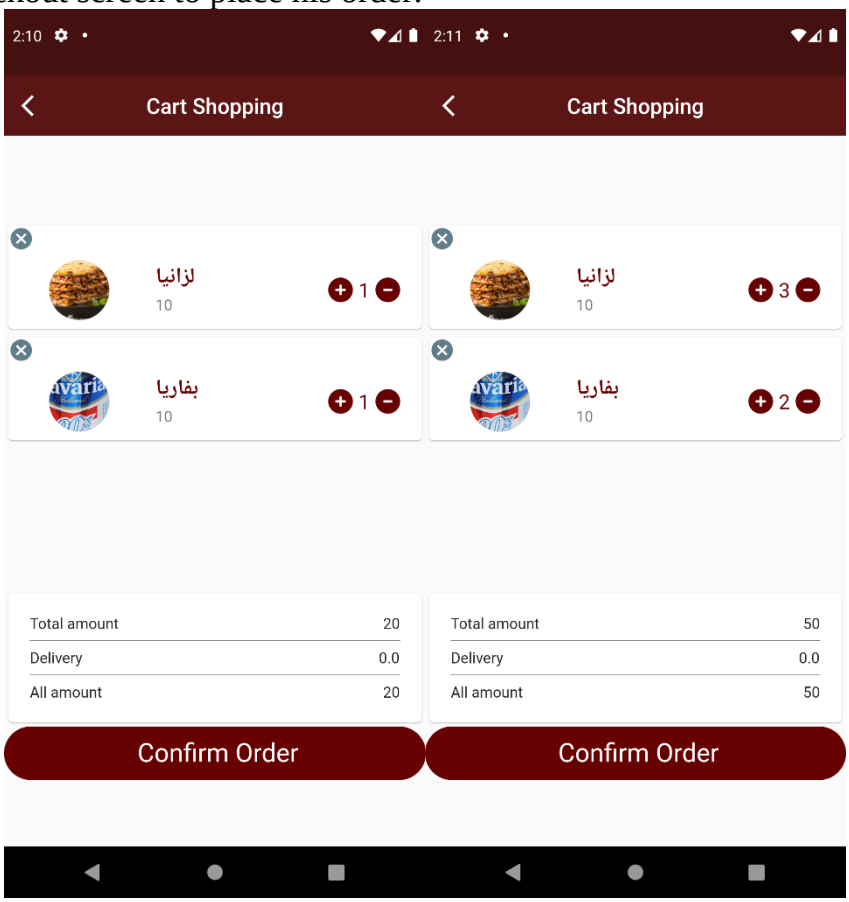


Figure :Cart Screen

❖ Confirm Order Screen

This page contains all the items in the shopping cart, and after the customer is sure of the orders he wants, the order is confirmed when he clicks on the order confirmation icon, and the customer can see his order through the My Order interface in the interface of his orders,

and through that interface, he can see all .

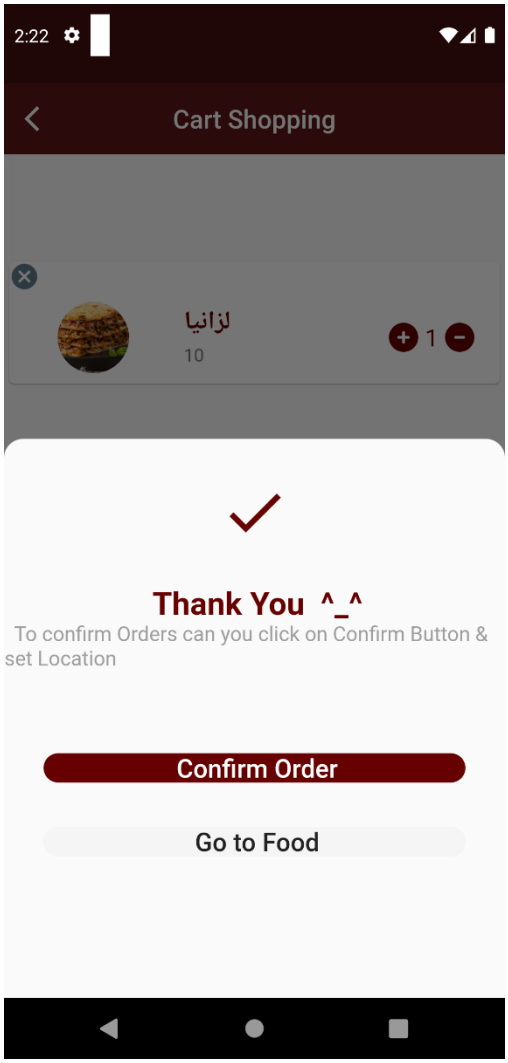


Figure :Confirm Order Screen

Figure : Order Screen

On this screen, you can see the details of the Order that you have previously placed

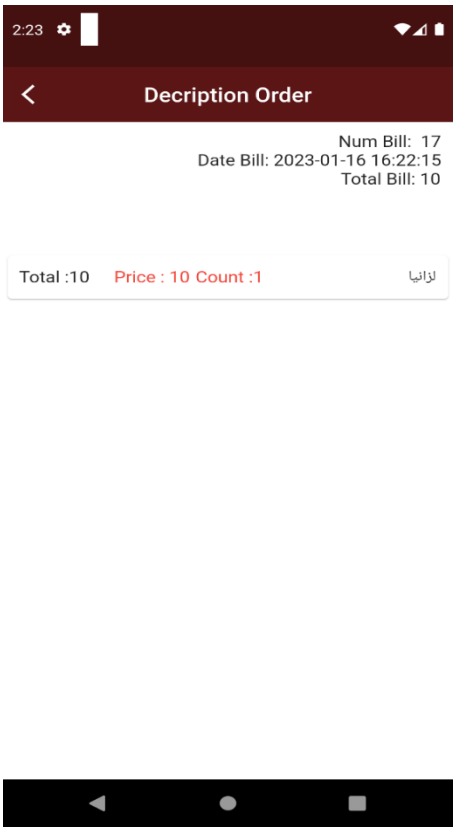


Figure : Decription Order Screen

❖ Notifications Screen

The Notification system pushes notifications if the admin adds a new Food.

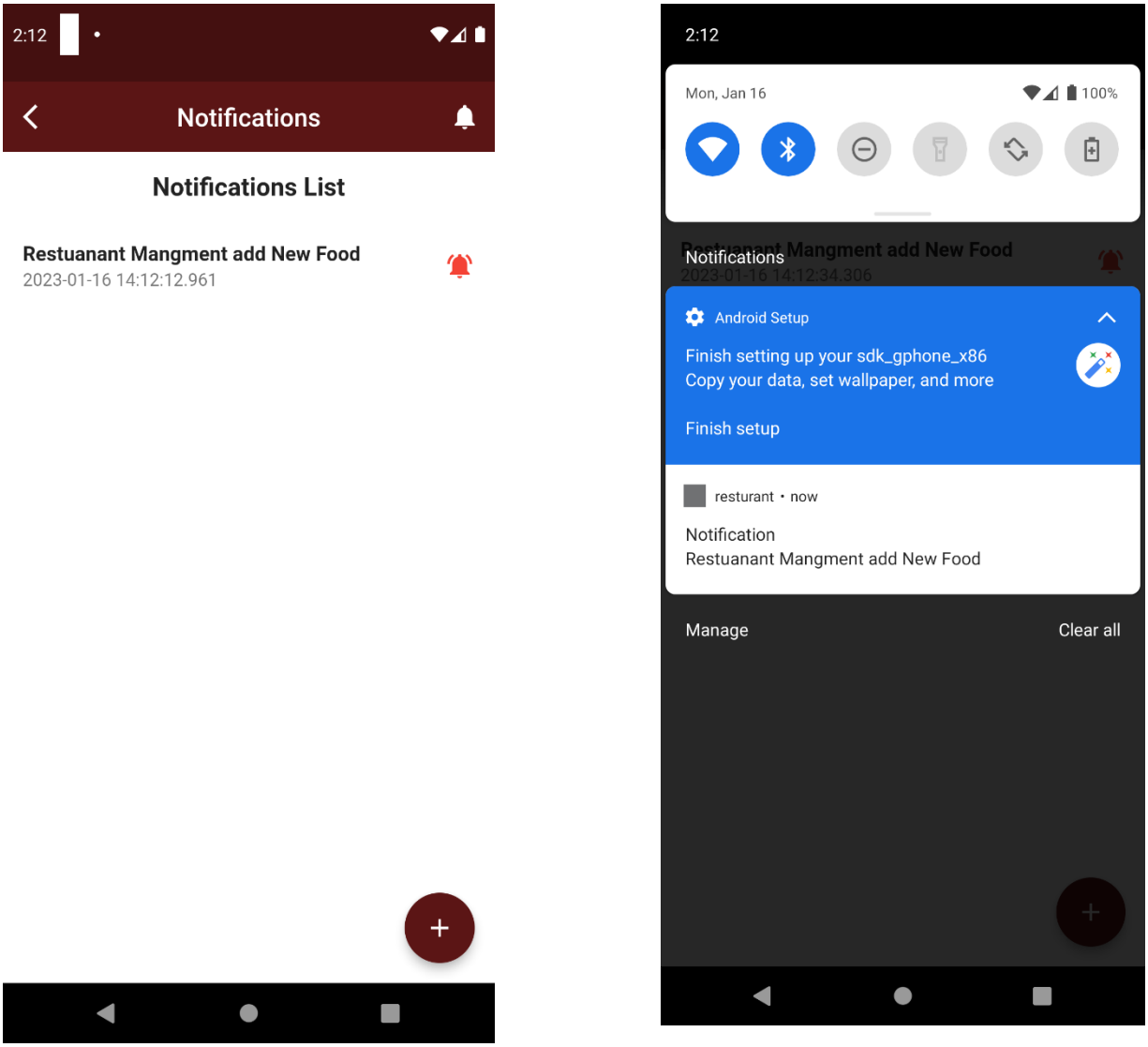


Figure : Notification Screen

❖ Dual Mode

When you click the dual mode button, if the interface is bright, it turns into a dark interface, if the interface is dark, it turns into a bright interface.

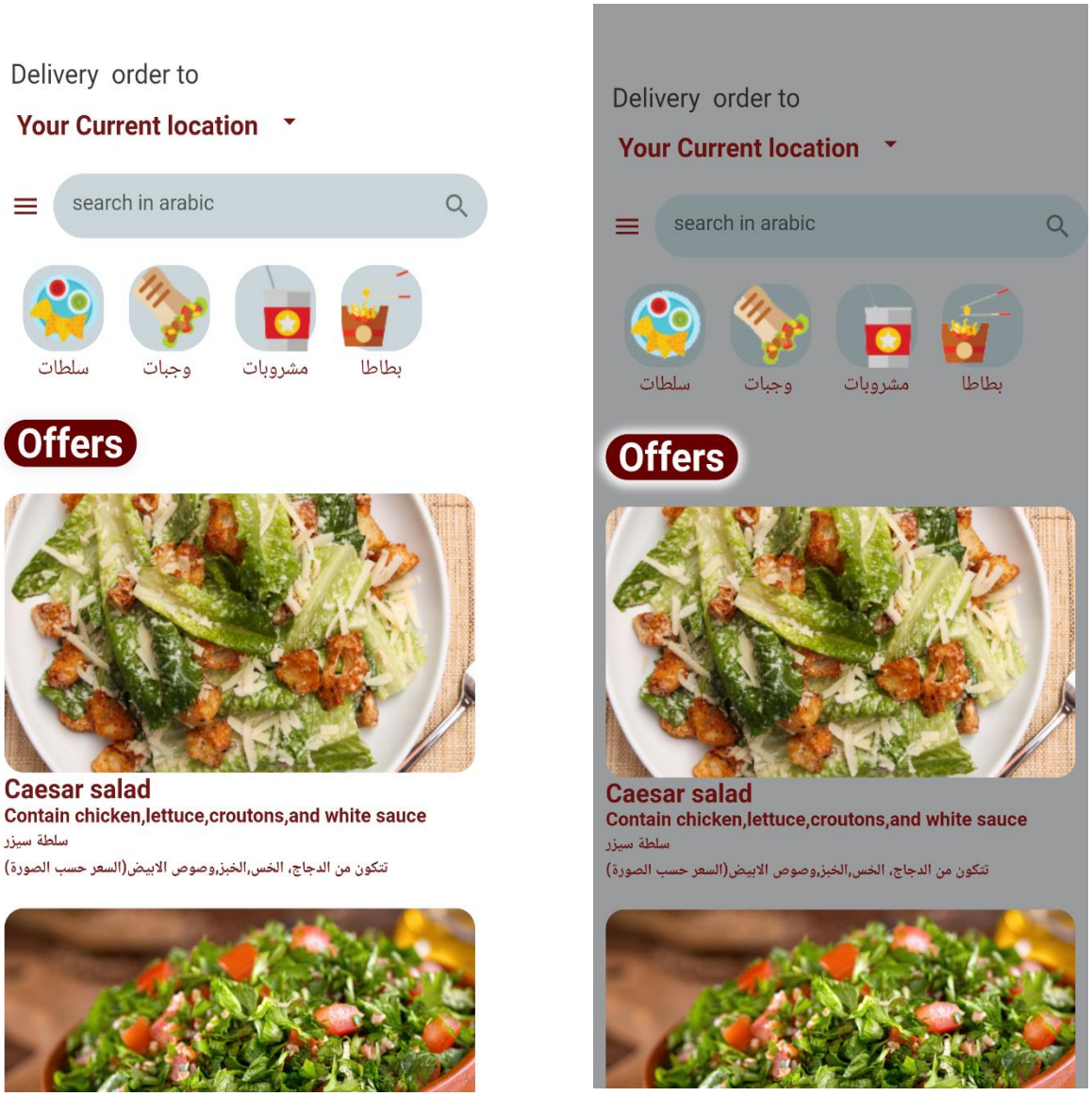


Figure : Dual Mode Screen

❖ Simple Comment

The screen contains a simple comment, the customer can add a comment on a specific meal or food without showing the name, in addition to that, the customer can view all comments on all meals.

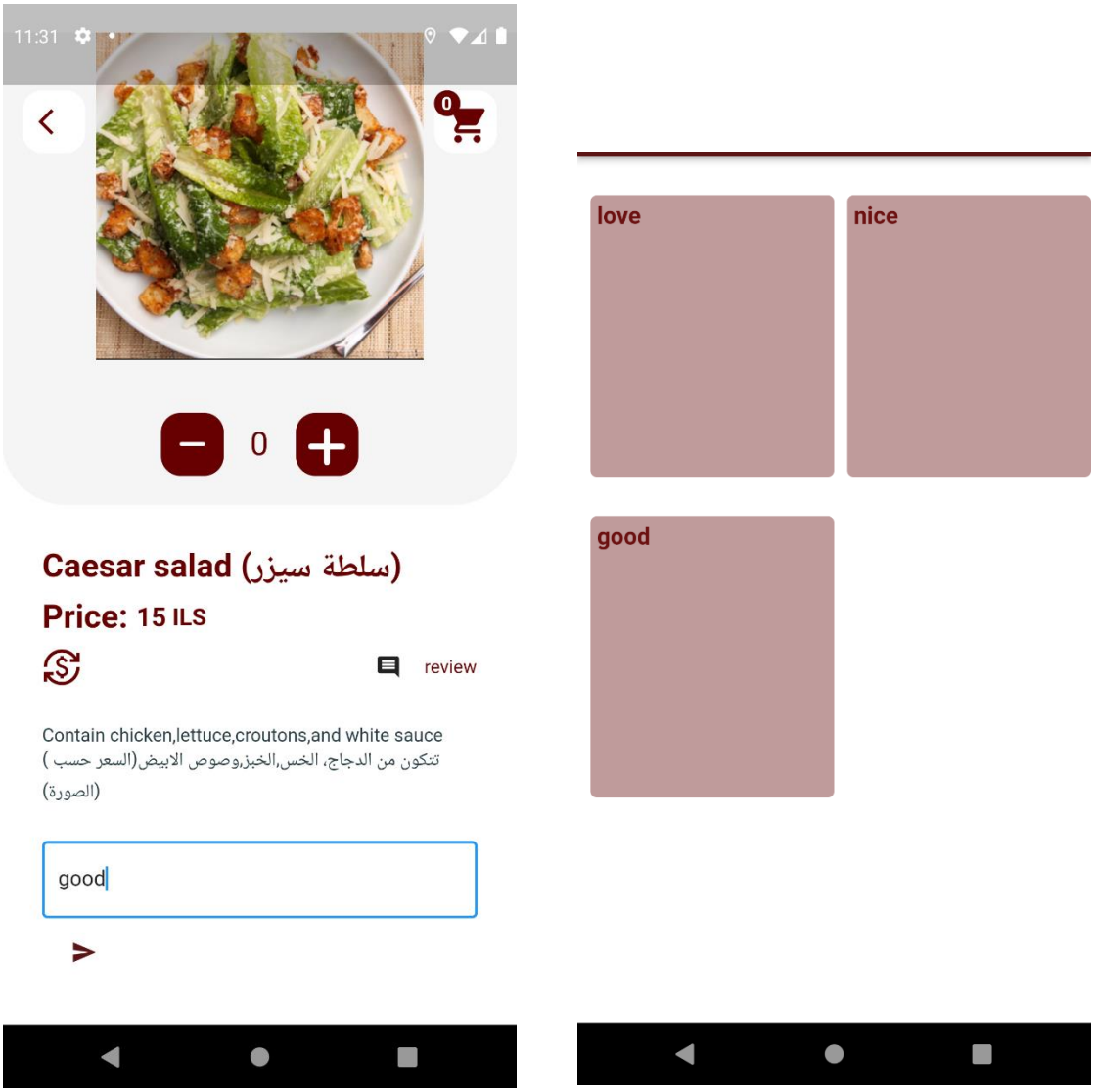


Figure : Simple comment Screen

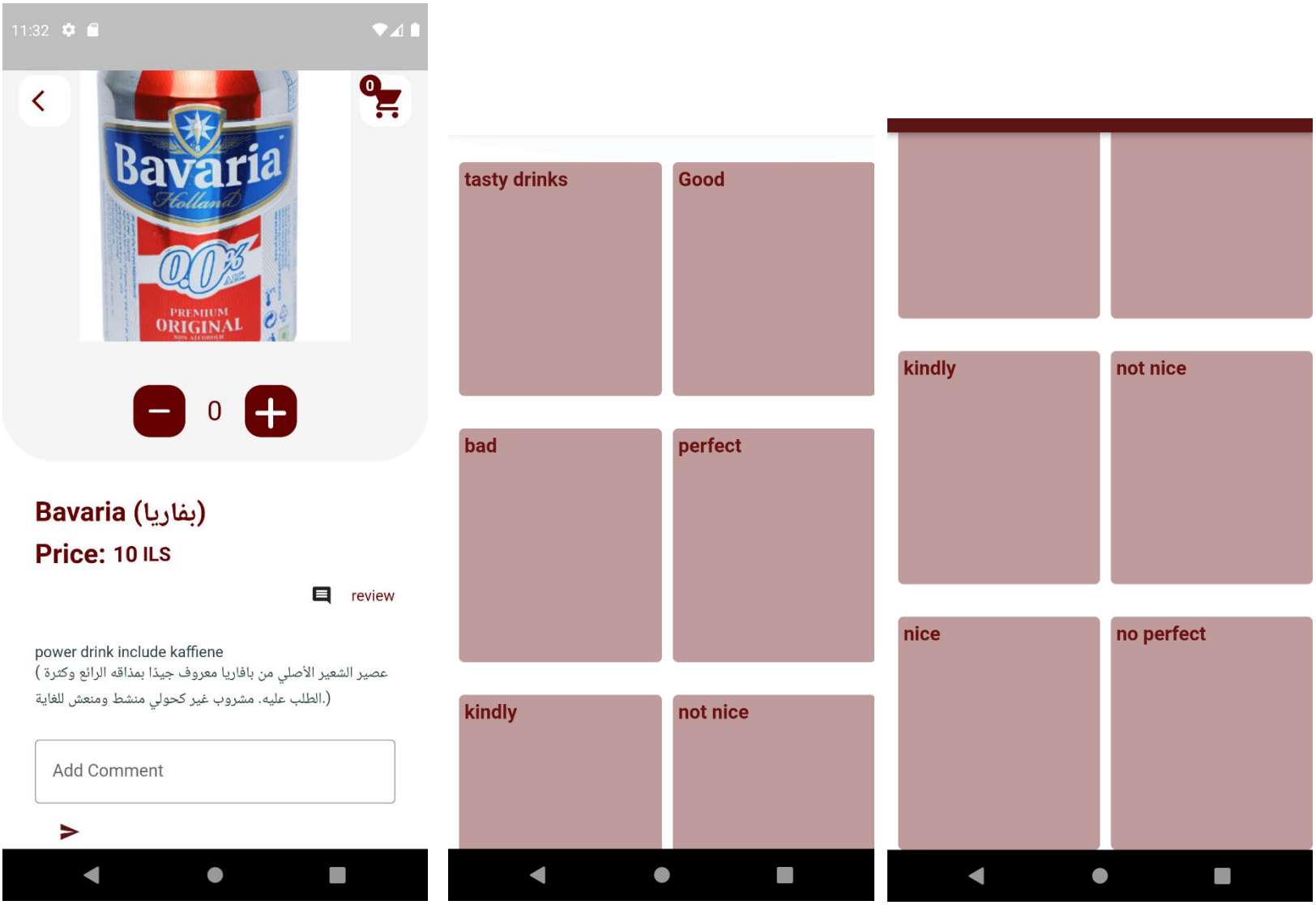


Figure : Simple comment Screen

5.3.2. Management Dashboard Application

This application is for the owner of the restaurant and the management team of the Restaurant application.

❖ Login

This is a simple login page where the Admin only writes their phone and password and the Application will redirect him to the corresponding dashboard.

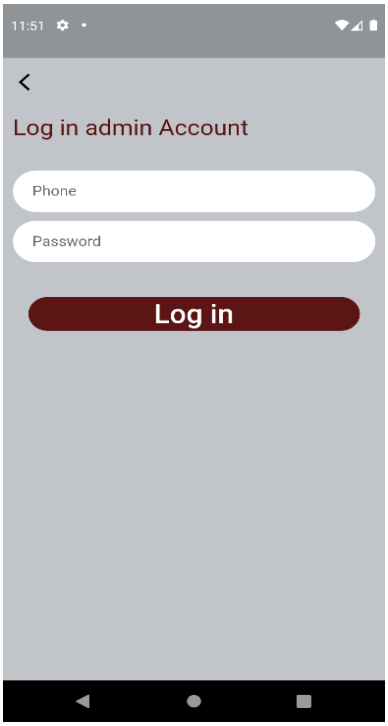


Figure : Log in Admin Screen

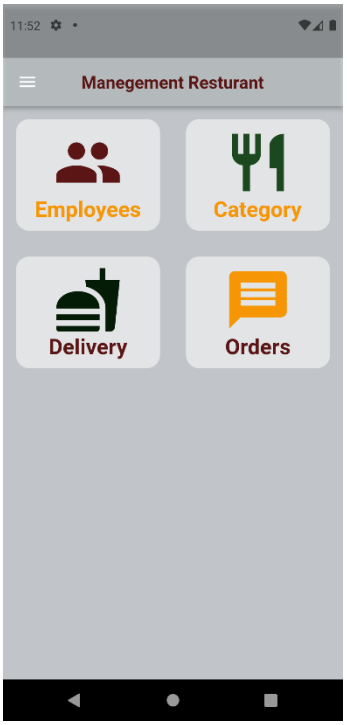


Figure : Management Rest Screen

❖ Employee Management

In this section, there is a Employee administration, which is a record of workers, which provides an opportunity for the admin to add a new worker through the application, modify the worker's information, delete the worker, or search for a specific worker.

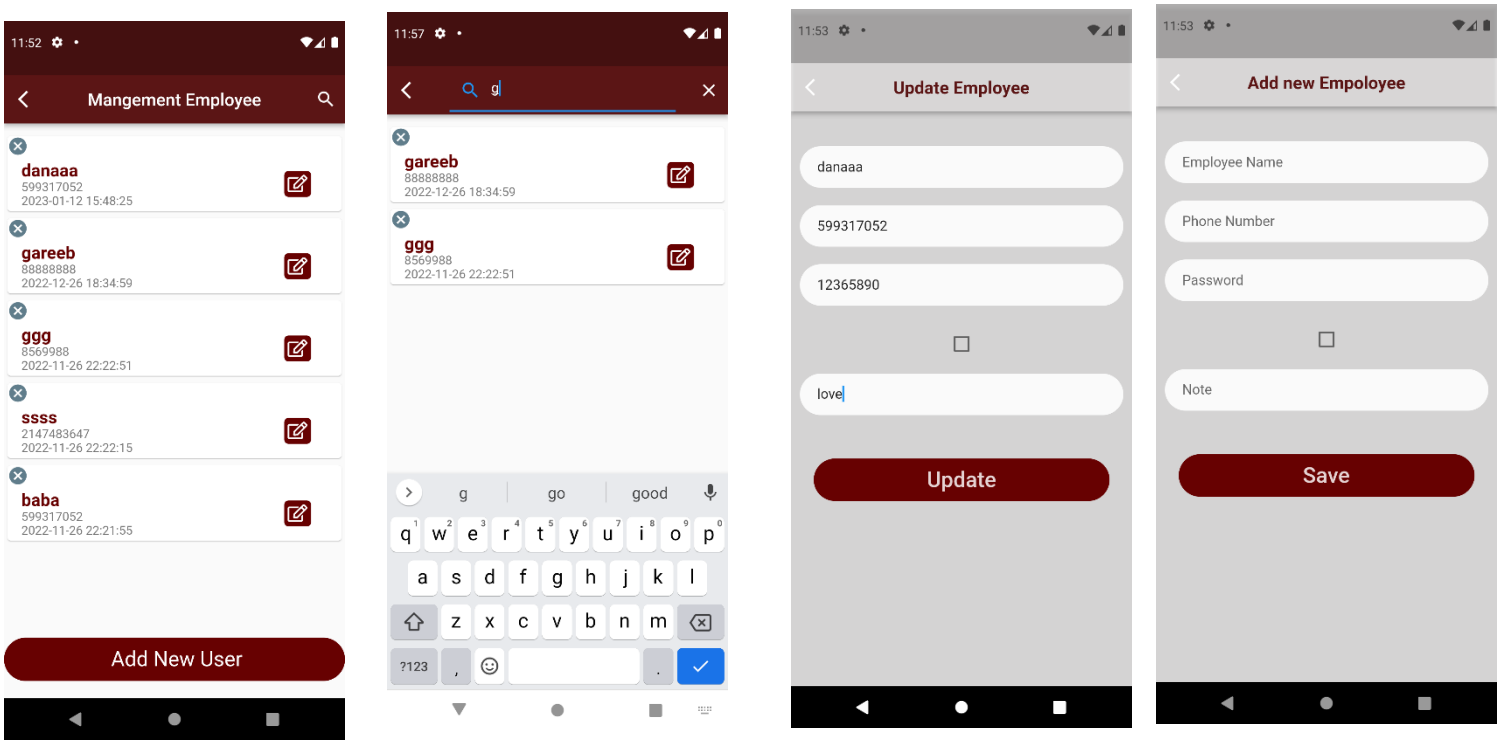


Figure : : Employee Management Screen

❖ Category Management

In this section, there is Category management, which provides an opportunity for the administrator to add a new category through the application, modify the category data, or delete the category.

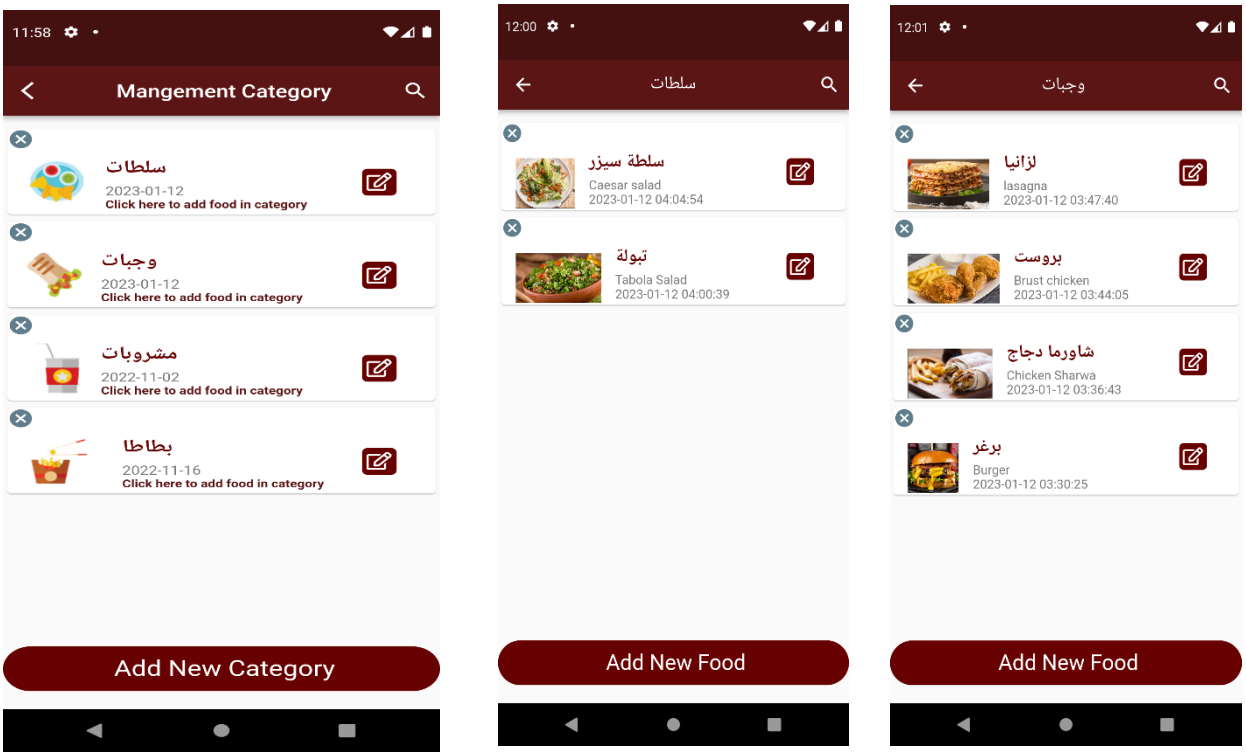


Figure : Category Management Screen

❖ **Food Management**

In this section, there is Food management, which provides an opportunity for the administrator to add a new food through the application, modify the food data, or delete the food.

This interface shows the modification of a specific meal, by entering the required fields, for example, the name in English and Arabic, the price, and is there an offer or not? Description in English and Arabic.
to complete the meal modification process.

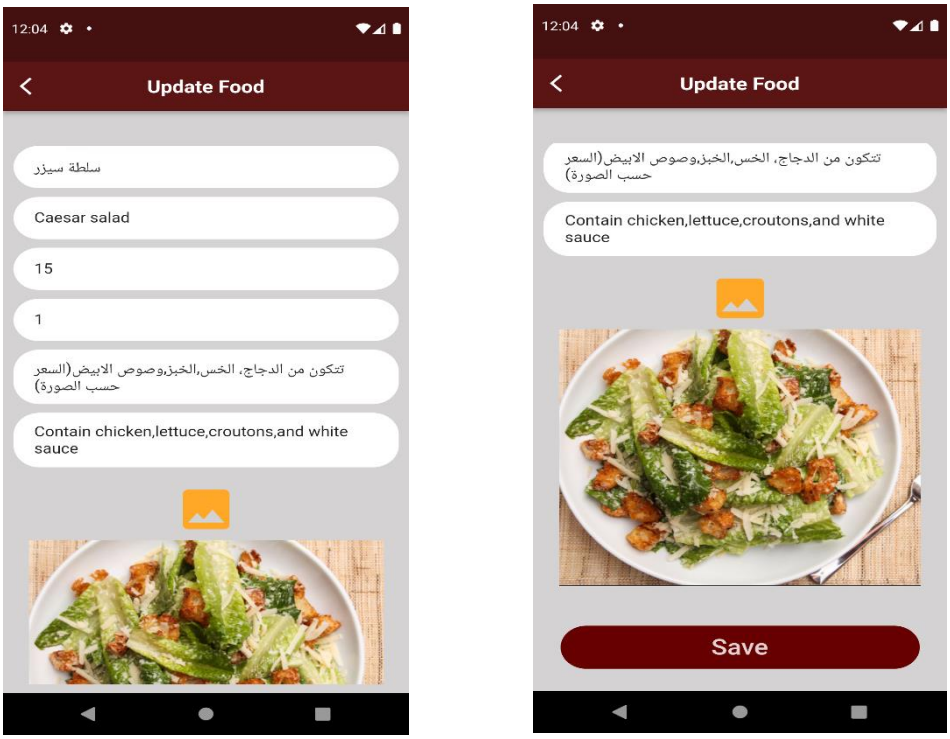


Figure: Update Food Screen

❖ Add New Category

This section shows the addition of a new category, so that a name and a picture are added to a category.

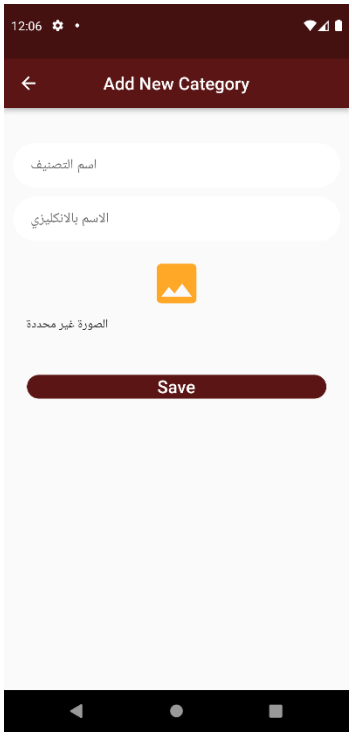


Figure: Add New Category Screen

❖ Add new Food

This section explains adding a new meal so that the process of adding a meal is done by selecting a category and then pressing the Add a new food button within that category, so that a name and a picture are added to the food.

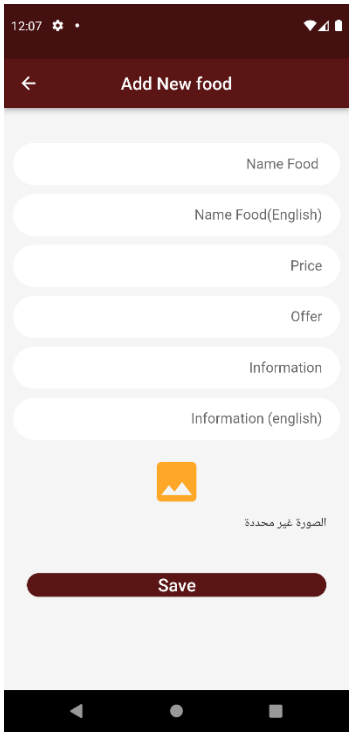


Figure: add new Food screen

❖ Received orders screen

Its monitor is responsible for keeping track of all new orders that have not been sent to the delivery company for delivery to the customer. When the store owner submits the order for delivery, all he has to do is click on the appropriate delivery option, so that when choosing, all customer orders are sent to a personal delivery account.

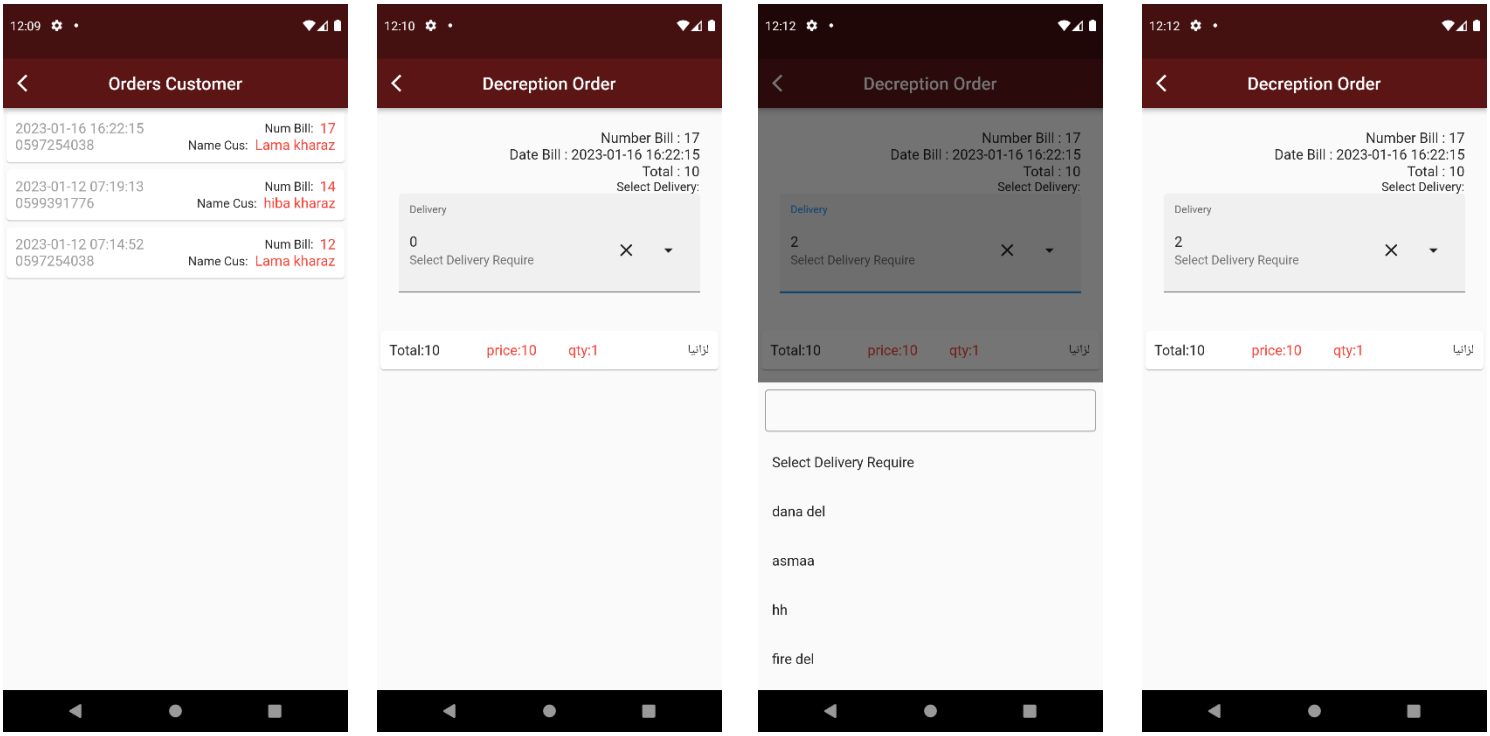


Figure: Decreption Order Management Screen

❖ Delivery Management

In this section, there is a Delivery administration, which is a record of Delivery, which provides an opportunity for the admin to add a new delivery through the application, modify the delivery information, delete the delivery, or search for a specific delivery.

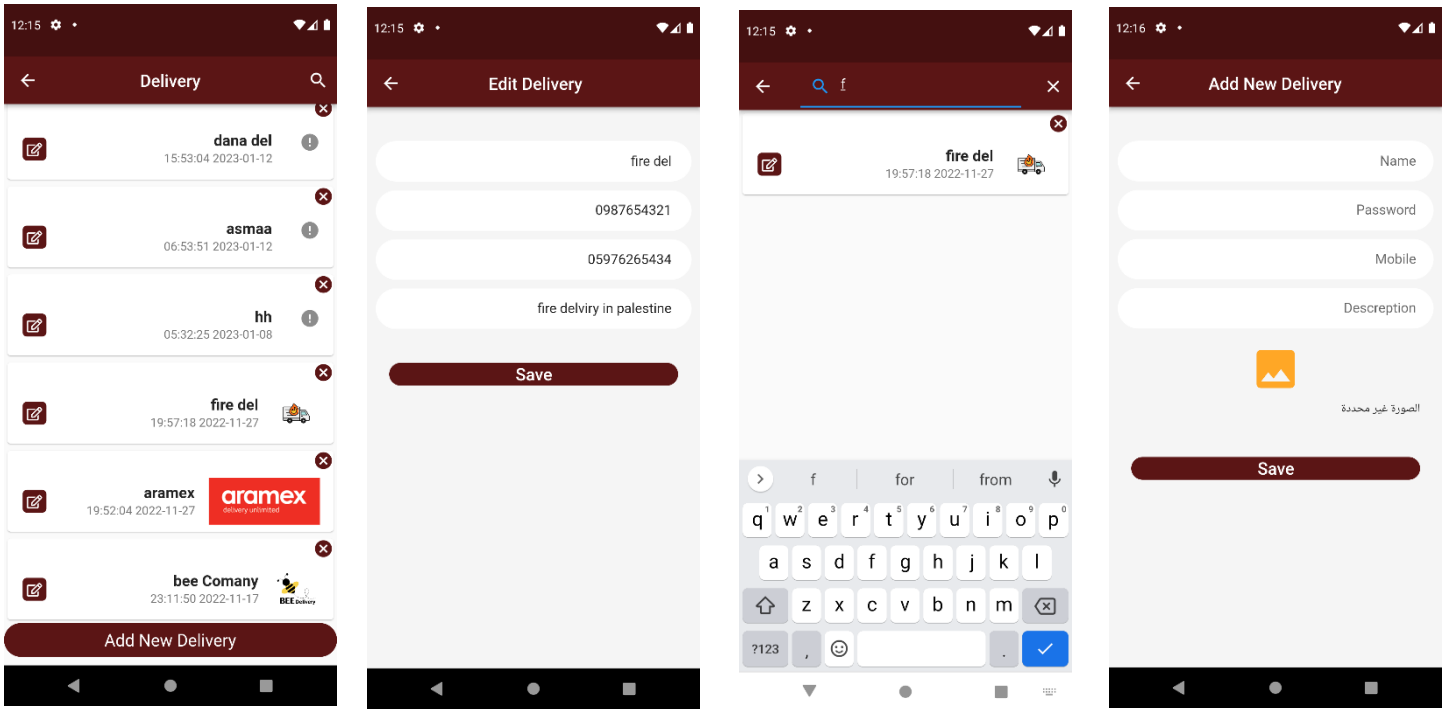
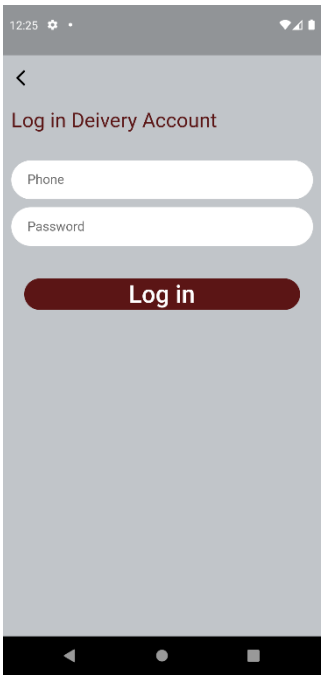


Figure: Delivery Setting Screen

5.3.3. Delivery Dashboard Application

❖ **Log In**

The process of logging in to the delivery account, through the phone number and the password, the process of adding a new delivery through the Management application, no one can create an account except through the Management.



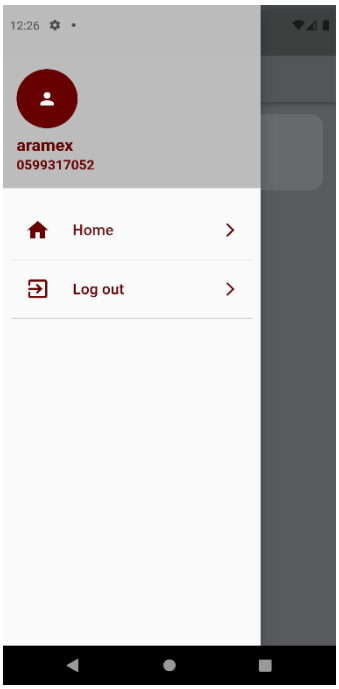
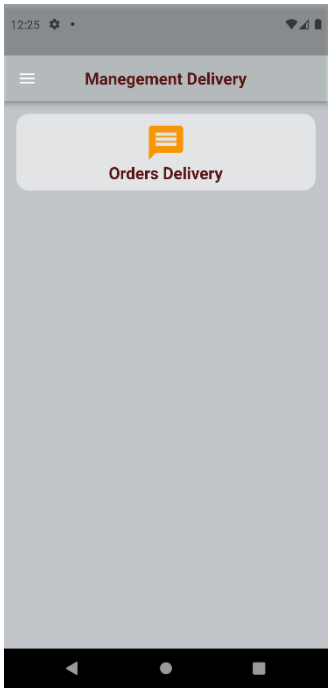


Figure: Log in Delivery Account Screen



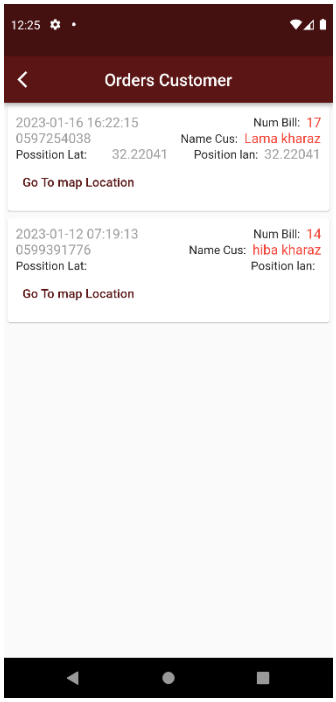


Figure: Orders Delivery Screen

❖ Orders delivery

In this section, all delivery orders are linked to his personal account, so that each order shows the information of the customer who made the order in terms of his name, phone number, date and time of the order, and tracking the customer's location using Google Maps.
It shows all the details of the order and the bill.

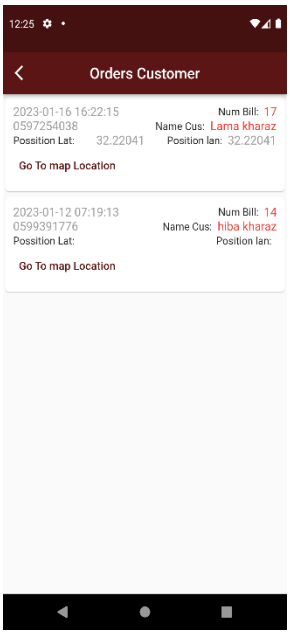


Figure: Order Customer Screen

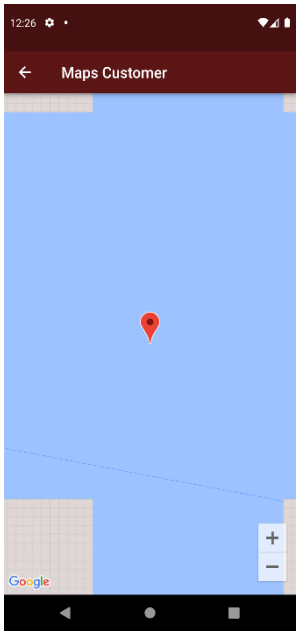


Figure: Tracking Location Customer Screen

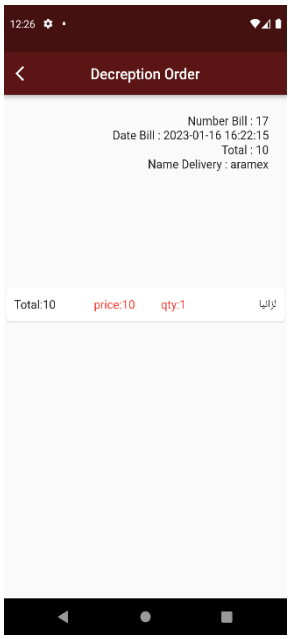


Figure: Decr Order Customer

6. Results and Discussion

I have built an application that helps customers view meals from the restaurant, giving them the advantage of choosing meals more easily instead of going to the restaurant.

Also, it can keep track of all meals without you having to keep checking.

The application also provides great assistance to the restaurant owner in terms of managing the restaurant, managing the delivery of their orders, and tracking the customer's location, in addition to staying abreast of all the new technologies used at the present time.

6.1. Learning

Learning something new takes patience, effort, and a lot of research. I learned a lot from this project and used a lot of new techniques that I didn't know before and had to learn from scratch. I choose to learn and use flutter and dart for client and server side. The internet is full of perfect educational content on all topics. In addition to the huge and massive documentation of any technology, all this helps to plan, develop and implement the idea more easily.

6.2. Challenges

I had a problem with my project's dependencies versions, some of the features we wanted to use were outdated but eventually we found alternatives for them.

7. Conclusions and Recommendation

7.1. Conclusion

I managed to create a restaurant application as a solution for many problems faces online Restaurant

Recommendations

In order to improve the application we can add an advanced recommendation system using AI and Tracking **delivery**.

7.2. What we learned

- Flutter framework with dart language to implement the project.
- On the Server side we used PHP language.

8. References

[1] yummy App – app on google play

[2] https://www.w3schools.com/php/