جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY
كلية العلوم والدراسات الإنسانية - الحاسب الآلي
College of Science and Humanities - Computer Department

CS 526 – Project Proposal
Term 2 – 2023-2024

Project Proposal Form

✓ Group #:3

✓ Members of the group:

#	Name	ID	Section	Role
1	Khaowlah Zakariya	2200002105	2	Leader
2	Fatimah ALharz	2200004362	2	Member
3	Lama Abdulaziz	2200004324	2	Member
4	Sara Al-Salahi	2200001151	2	Member

Table of Contents

Table of figures

Table of table

1. Introduction of the Project

With the development of technology and our daily use of applications that facilitate daily tasks without the need to complicate things too much or visit many places to perform our tasks, we decided to create an application that serves mothers and children at any time and in more than one region in the Kingdom of Saudi Arabia, as our application will provide the mother with ease. In searches and requests for her child, and providing some fun activities for the child, all of this in just one place without the need to move between applications, search on websites, or visit places.

2. Description of the idea

Initially, the application will provide services to mothers and children.
The mother will be able to open her own account and add her children. We will provide her with a group of services. First, the mother can search for kindergartens in the area where she lives. By specifying the location, a group of kindergartens will appear. She can see information about a specific kindergarten and book an appointment for her child. The reservation can be made at different times, such as by day, hour, or month. The mother can also submit a request for a nanny in her home to help her care for the child. Several options will also appear for her, and she can choose what the mother deems appropriate for her through the information available. Secondly, we will provide a food ordering service with a variety of options for special dishes for the child, as these foods will be suitable for children aged 5 months and above. Third, providing useful and educational activities for the child in an enjoyable way that the mother can present to her child without fear or worry about any immoral content.

3. Problem Statement:

1. The mother goes to many kindergartens to choose the appropriate nursery for the child.
2. Providing the freedom for a child to view educational content, for example, YouTube, may lead to watching clips that are not appropriate for his age.
3. Lack of providing children with home-prepared food with natural ingredients without any additives

4. Objectives (Product Characteristics and Requirements):

1. Providing the service of choosing the appropriate kindergarten for the child from one application without the need for a visit
2. Providing a diverse menu made at home by people who have the skill, as it is suitable for young children.
3. Providing educational games for the child, for example, addition and subtraction operations, and providing various activities

4. Project end user:

Our application will be aimed at parents, especially the mother, as she is the one most concerned with taking care of her child.

5. Summary of Project Deliverables:

The project will culminate in the delivery of a comprehensive mobile application designed to cater to the needs of mothers and children in the Kingdom of Saudi Arabia. The deliverables include A user-friendly interface for mothers to manage their profiles and add their children’s details. A directory of kindergartens within the user's locality, complete with details and the ability to book appointments. A nanny request service, offering various options for in-home childcare assistance. A food ordering feature, specializing in dishes suitable for children aged 5 months and older, focusing on natural and additive-free ingredients. Educational and entertaining activities for children, ensuring content is age-appropriate and engaging.

6. Project Success Criteria:

Success for this project will be measured through:

- User adoption and engagement: A significant number of mothers in the Kingdom of Saudi Arabia downloading and actively using the application.
- Positive feedback: High ratings and favorable reviews on the Google Play Store, reflecting satisfaction with the app’s features and usability.
- Impact on decision-making: A measurable decrease in the time and effort mothers spend on finding kindergartens and nannies, and in preparing suitable meals for their children.
- Educational value: Feedback from users indicating that the educational activities provided are beneficial and enjoyable for their children.

7. Solution Methodology:

We will build our application using the Android Studio application as it can run the app on multiple Android phone types, it provides a flexible design system that enables us to simulate any device's screen. Also, it enables us to watch the design or the output of the screen change as we write the code, and we will be able to test the application before we release it to the end user. Our ultimate reliance will be on Java which is a fast, reliable, and secure programming language. You can program different things using it like desktop apps, mobile apps, and games. Also, it's considered the standard language for programming Android mobile applications. After we are done with the design phase and the testing process, we will publish our application to the end users on Google Play Store.[1][2]

8. Comprehensive List of Application Requirements: Provide a detailed list of requirements for the application, covering functional, non-functional, and technical aspects:

- **Functional Requirements:**

1. User Account Management:

-User Registration and Login: Allow mothers to create and manage their accounts using email, phone number, or social media login.

-Profile Management: Ability to add, edit, and delete children's details and personal information.

-Password Management: Features for password reset and recovery.

2. Kindergarten Directory and Booking:

-Search Functionality: Search for kindergartens based on location, ratings, reviews, and other filters.

-Kindergarten Profiles: Detailed profiles for each kindergarten, including contact information, facilities, reviews, and ratings.

-Booking System: Real-time availability check and appointment booking for various durations (day, hour, month).

-Booking Management: View, edit, and cancel bookings.

3. Nanny Request Service:

-Nanny Search: Search for nannies based on location, experience, ratings, and availability.

-Nanny Profiles: Detailed profiles including experience, qualifications, reviews, and ratings.

-Booking Requests: Submit and manage requests for nanny services.

-Nanny Management: Nannies can create and manage their profiles, view incoming requests, and accept or decline bookings based on their availability.

4. Food Ordering Service:

-Menu Browsing: Browse a variety of meal options suitable for children aged 5 months and older.

5. Educational and Entertaining Activities:

-Activity Library: Access to a library of educational games and activities categorized by age and educational content.

-Safe Content: Assurance that all activities are age-appropriate and free from immoral content.

- **Non-Functional Requirements:**

1. Performance:

-Fast Response Time: Ensure quick loading and response times for all user interactions.

-Efficient Data Handling: Optimize the handling of large datasets for kindergartens and nannies.

2. Scalability:

-Support Growth: Ability to handle an increasing number of users and data without performance degradation.

3. Security:

-Data Encryption: Protect user data by encrypting it.

-User Verification: Ensure users are registered before accessing the app.

4. Usability:

-User-Friendly Interface: Ensure an intuitive and easy-to-navigate interface.

5. Reliability:

-High Availability: Ensure minimal downtime and high availability.

-Data Backups: Regular backups of user data.

-Robust Error Handling: Implement comprehensive error handling and logging.

6. Maintainability:

-Clean Codebase: Maintain a clean and well-documented codebase.

-Regular Maintenance: Schedule regular updates and maintenance.

- **Technical Requirements:**

1. Platform:

-Android Compatibility: Ensure compatibility with Android 5.0 and above.

2. Development Tools:

-Android Studio: Use Android Studio as the primary development environment.

-Java: Develop the app using Java programming language.

3. Database:

-Scalable Database: Use a robust and scalable database system (Firebase).

4. Testing:

Comprehensive Testing: Develop a comprehensive testing plan covering unit test, integration tests, and UI tests.

9. Core Features: Outline the key functionalities of your mobile app:

The following points are the main functionalities of the application:

1. User Account Management:

- Mothers can create and manage their accounts, add their children's details, and personalize their profiles.

2. Kindergarten Directory and Booking:

- A comprehensive directory of local kindergartens with detailed information, including reviews, ratings, and contact details.

- Real-time availability and the ability to book appointments for different durations (day, hour, month).

3. Nanny Request Service:

- A selection of available nannies for in-home childcare assistance, complete with profiles, experience details, and user reviews.

- Nannies can create and manage their own profiles, view incoming requests, and accept or decline bookings based on their availability.

4. Food Ordering Service:

- A variety of menu options featuring homemade, natural, and additive-free dishes suitable for children aged 5 months and older.

- Customizable meal plans and the ability to place orders directly through the app.

5. Educational and Entertaining Activities:

- A library of educational games and activities, such as addition and subtraction games, that are both fun and age-appropriate.

- Assurance that all content is free from any immoral or inappropriate material.

6. User-Friendly Interface:

- Intuitive design ensuring easy navigation and use, tailored to the needs of mothers and their children.

7. Location-Based Services:

- Integration of location services to suggest kindergartens, nannies, and activities nearby.

10. Mobile App User Flow Diagram: Include a diagram that shows the flow of the user through the app, from start to finish:

The flow chart begins with the user creating an account and logging in, which reveals three main sections: searching for a nanny, ordering food for the child, and selecting activities for the child. In the nanny section, the user searches for a nanny in a specific area, fills in the child's personal information, and requests an appointment, receiving notifications of acceptance or rejection. In the food section, the user selects a food category and places an order. In the activities section, the user selects an activity type, chooses the child's age group, and watches a suitable video. The process ensures a streamlined user experience across all three functionalities, concluding when the user's needs are met.

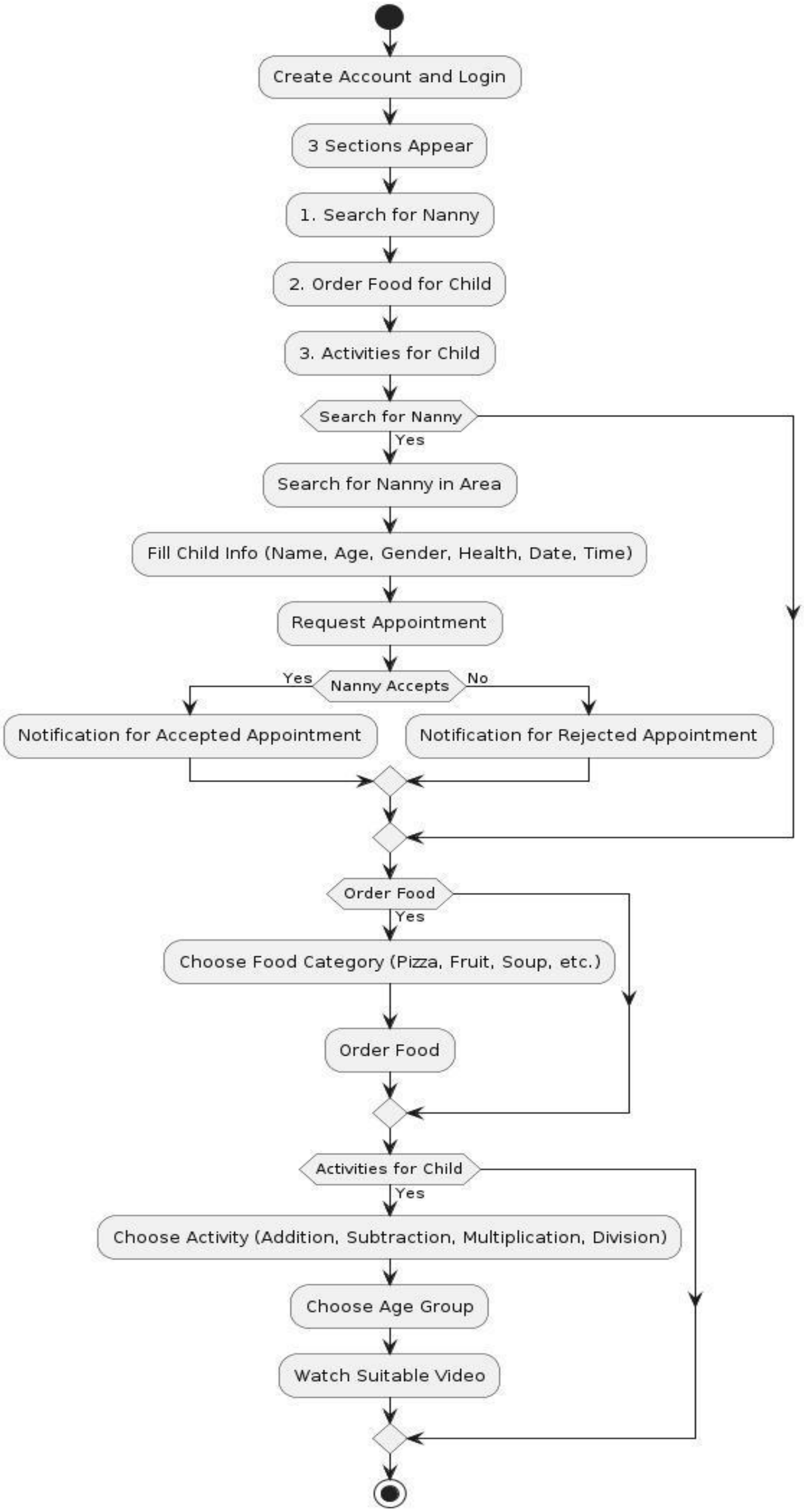


Figure 1Mobile App User Flow Diagram

11. Mobile App Flowchart: Include a flowchart that details the logical flow of the app, including decision points and user interactions:

A flow chart details the logical flow of the application, including decision points and user interactions as it begins with creating accounts for the nanny and the user and explains what each person does when entering the application. It also explains the role of the manager in adding food in the interface assigned to him and how the nanny does it. Add a kindergarten and see the reservations it has received from users

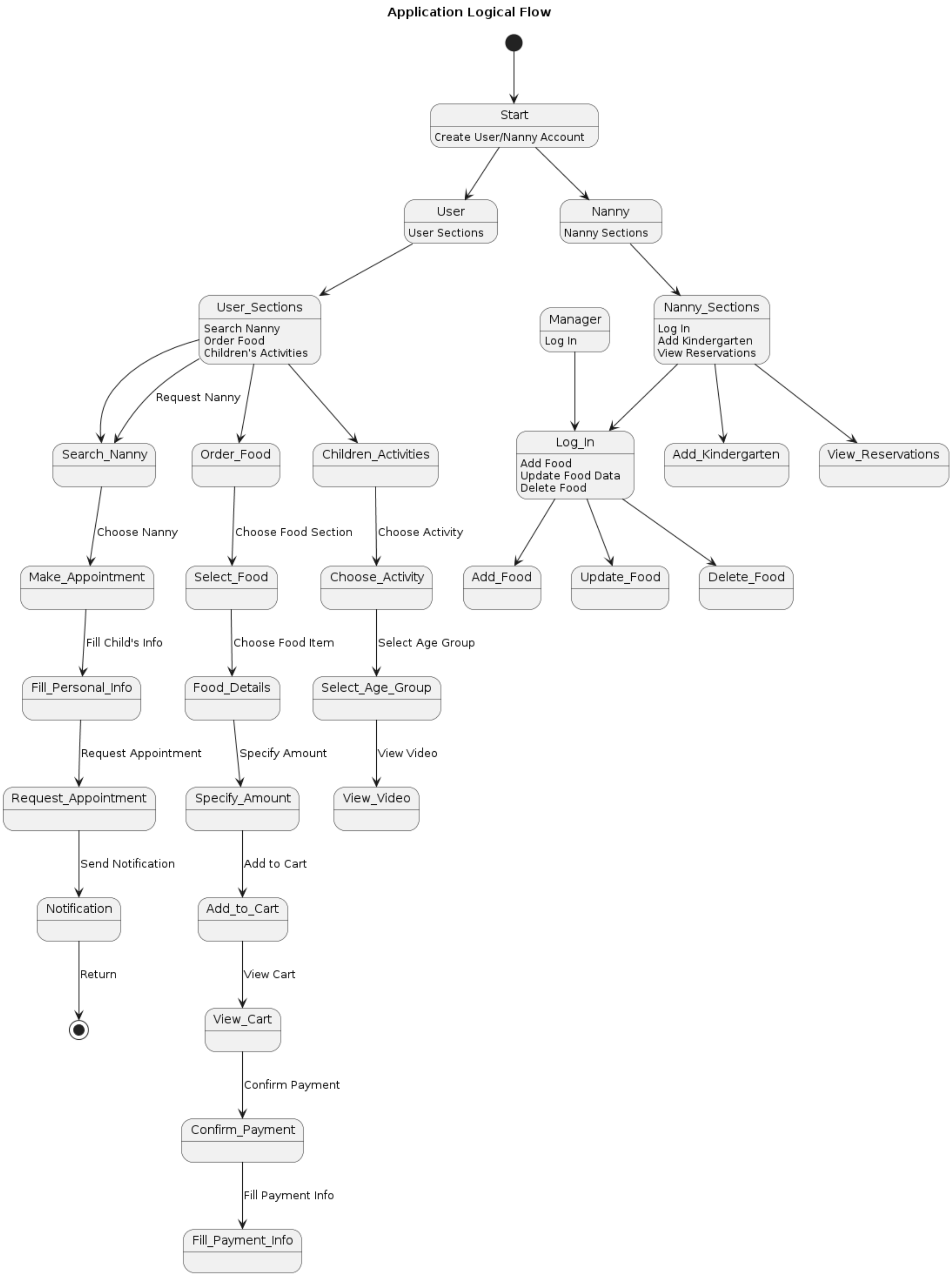


Figure 2Mobile App Flowchart

12.**Prototype and explain each interface:**

12.1 Welcome page

This page appears to the user as soon as he opens the application, and when he presses the let’s Go button, he goes to the account creation page.

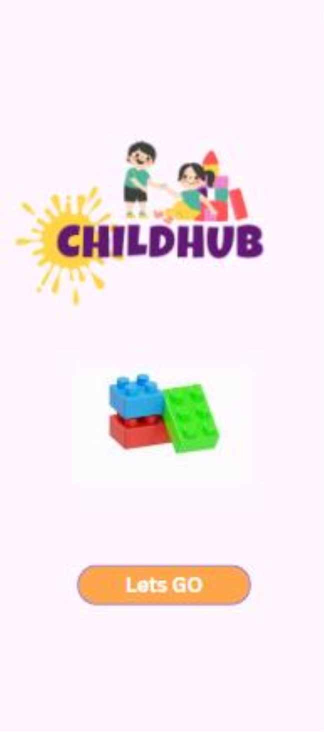
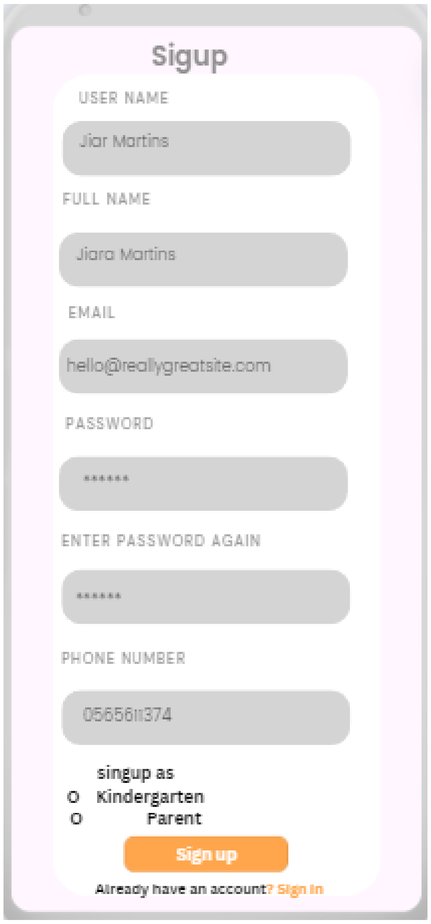


Figure 3Welcome page

12.2 Account creation page

When you enter the application for the first time, you must create an account and fill out the following information, such as the user name, full name, email, password, and mobile number, then choose whether I want to sign up as a parent or as a kindergarten, then press the sin up button.



Sigup

USER NAME

Jiar Martins

FULL NAME

Jiara Martins

EMAIL

hello@reallygreatsite.com

PASSWORD

******

ENTER PASSWORD AGAIN

******

PHONE NUMBER

0565611374

singup as

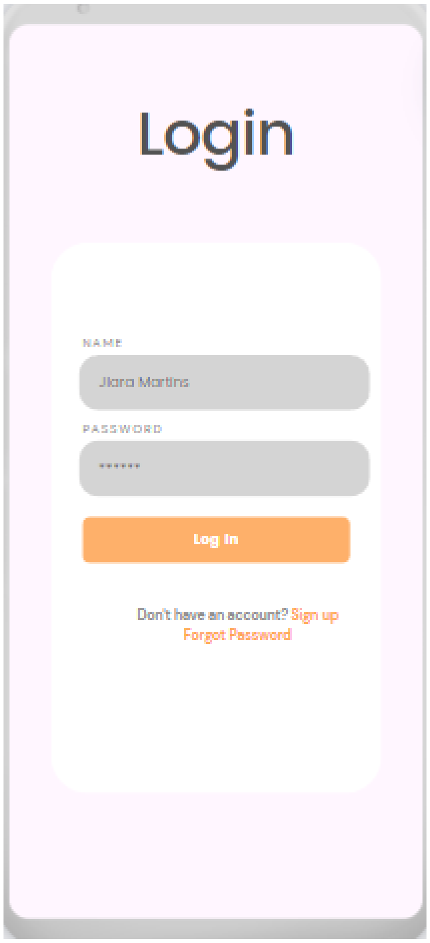
○ Kindergarten

○ Parent

Figure 4Account creation page

12.3 Login page

After creating the account, you can log in by filling in your email and password and clicking the sin in button



NAME

Jlora Martins

PASSWORD

******

Don't have an account? Sign up

Figure 5Login page

12.4 Forgot password page

If you forgot your password, you can reset it by entering your email and clicking done, and you will receive an email to reset it.

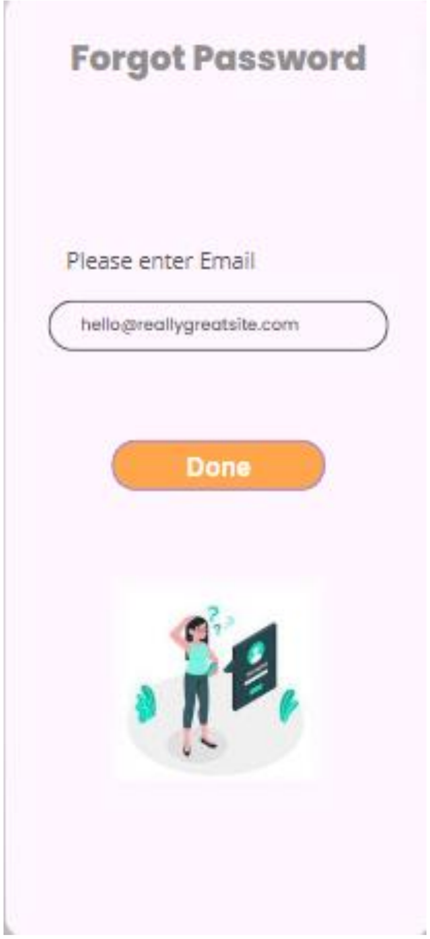
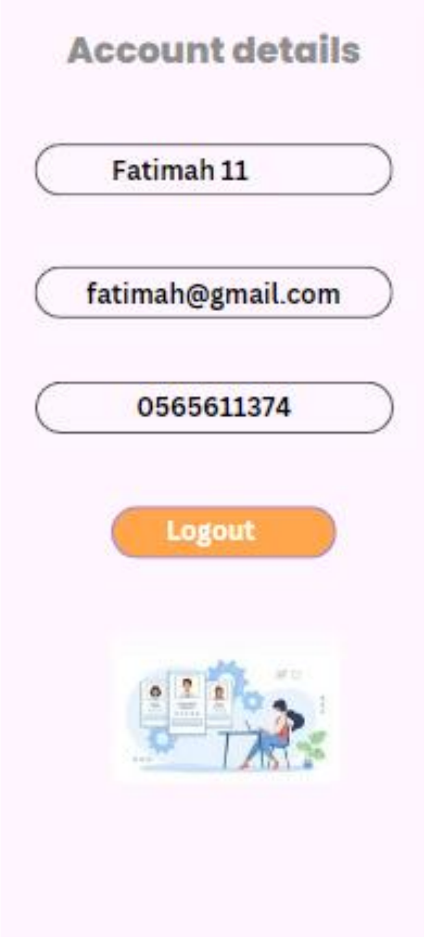


Figure 6 Forgot password page

12.5 User profile page

Here the profile and information for the parents appear, such as email, mobile number, and username. If you want to log out, click the logout button.



Account details

Fatimah 11

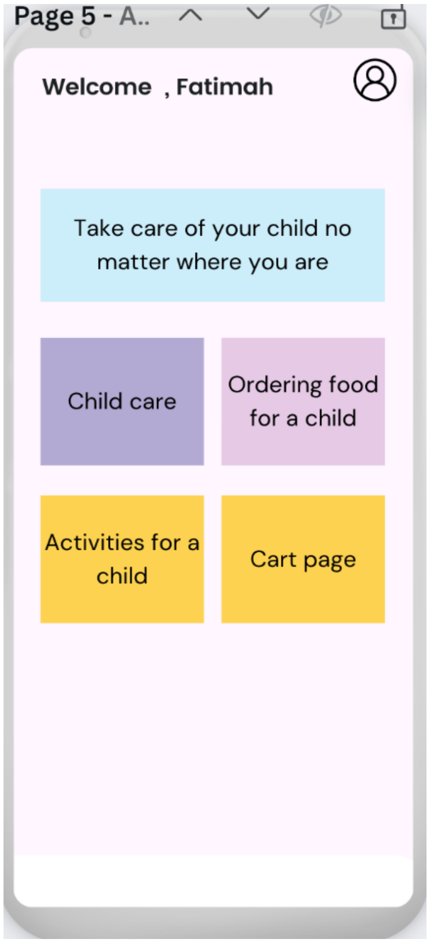
fatimah@gmail.com

0565611374

Figure 7User profile page

12.6 The user's home page after logging in

When the user logs in, the main page will appear to him containing the displayed sections, where he can, through these buttons, access the pages he wants.



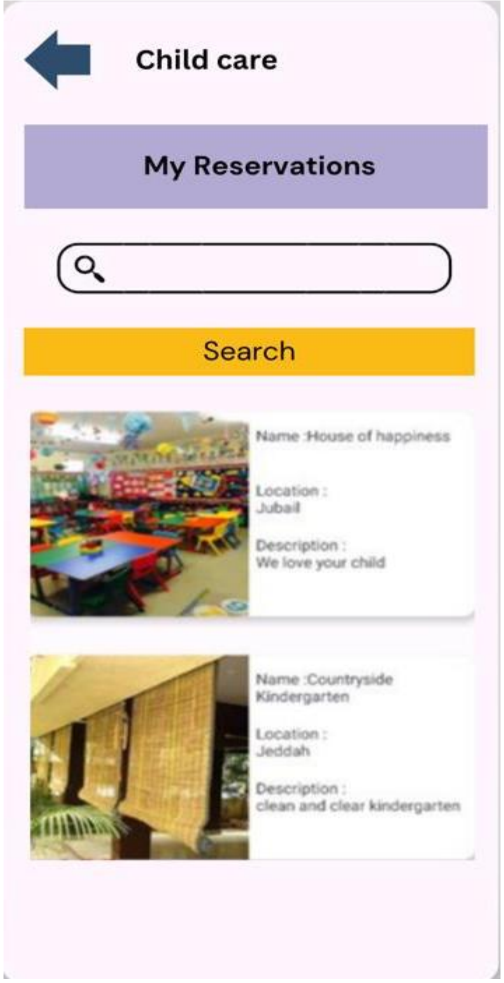
Welcome , Fatimah

Take care of your child no matter where you are

Figure 8 home page

12.7 Child care

The interface displays a childcare reservation page. It includes a back arrow for navigation, the title "Child care," and a "My Reservations" button to view current reservations. There is a search bar with a magnifying glass icon and a yellow "Search" button. Below, the page lists childcare facilities with images, names, locations, and brief descriptions. This interface allows users to search for and view details about various childcare options. And when selected one of Child care the details page will appear.



Name :House of happiness

Location :
Jubail

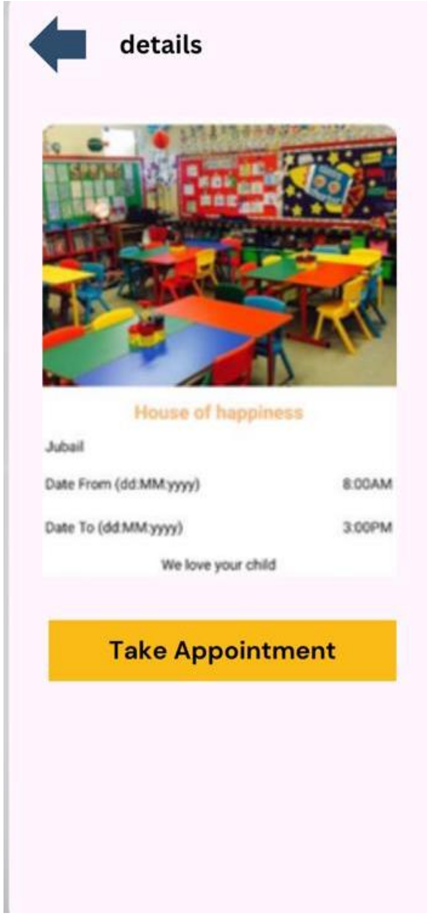
Description :
We love your child

Name :Countryside Kindergarten

Location :
Jeddah

Description :
clean and clear kindergarten

Figure 9 Child care 1



House of happiness

Jubail

Date From (dd:MM:yyyy) 8:00AM

Date To (dd:MM:yyyy) 3:00PM

We love your child

Figure 10 Child care 2

After choosing the nanny and clicking on the Take Appointment button, a page will appear to fill out the child’s information

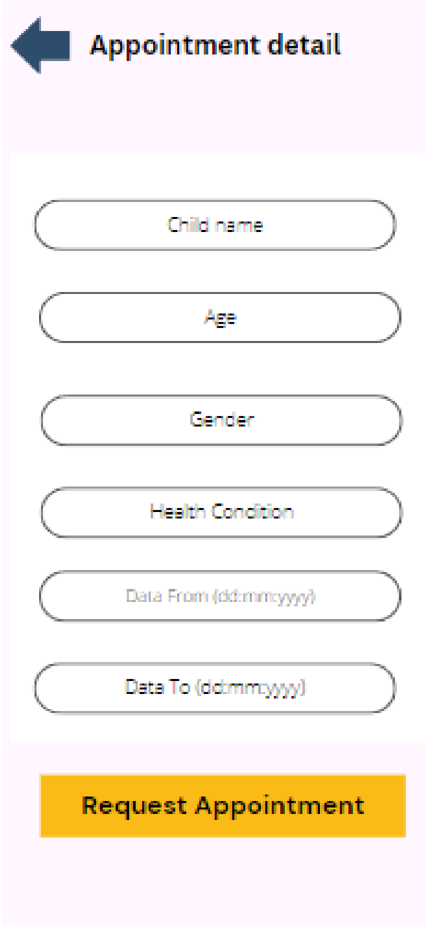


Figure 11 Appointment detail

12.8 The nanny's home page

This is the home page for the nanny. She can choose appointment request to view reservations or my kindergarten to put a description of her, her location, and the date of availability.

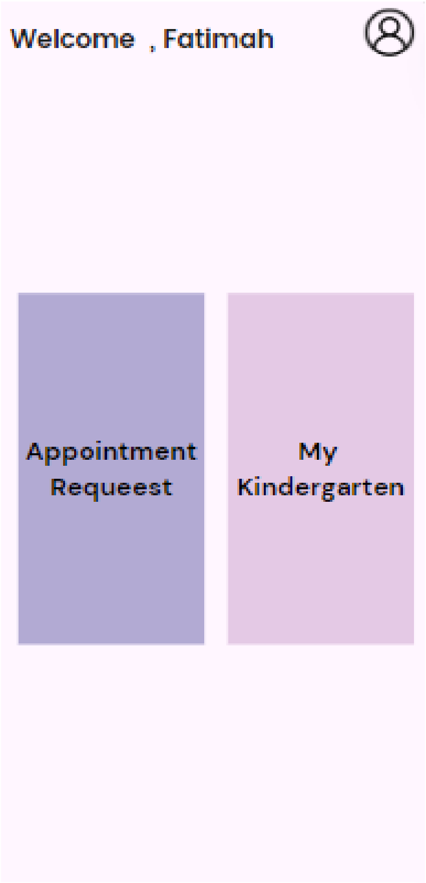
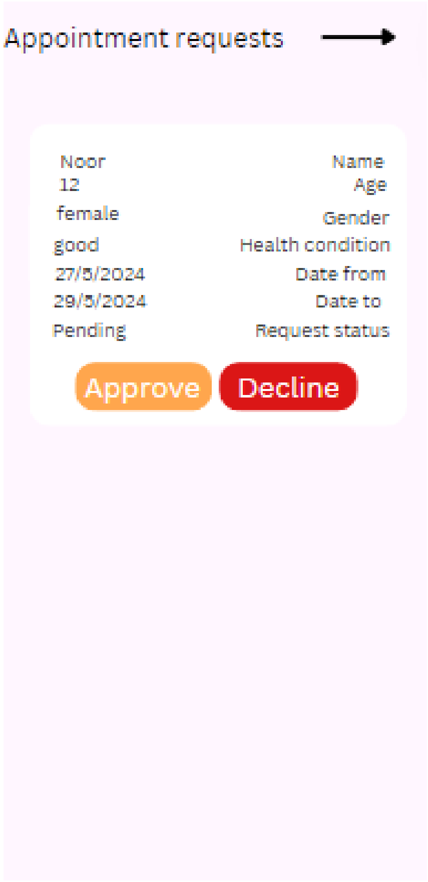


Figure 12 The nanny's home page

12.9 Appointment request

Here the nanny can accept or reject the reservation



Appointment requests

Noor	Name
12	Age
female	Gender
good	Health condition
27/5/2024	Date from
29/5/2024	Date to
Pending	Request status

Figure 13 Appointment request

12.10 My kindergarten

Here you can fill out the nanny's information, description, and location

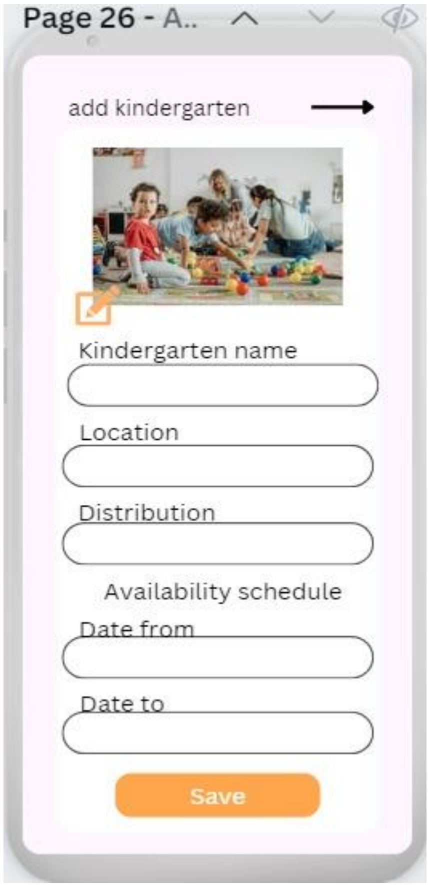
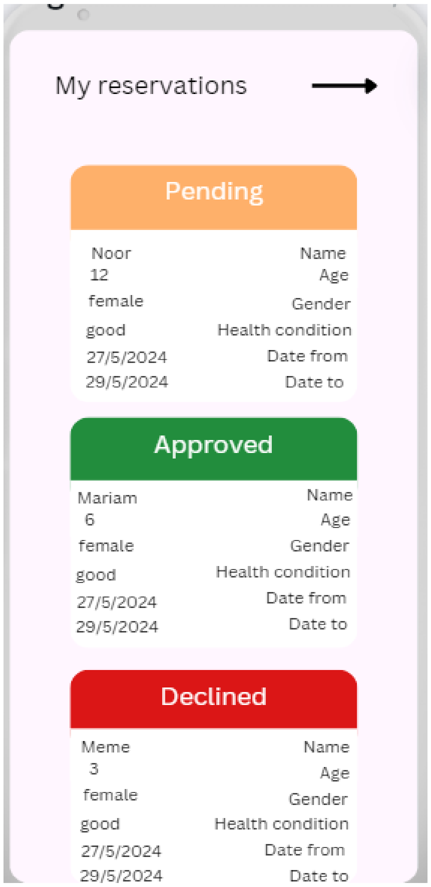


Figure 14 My kindergarten

12.11Nanny reservation’s view page for the manager

In this page, the nanny can view all the reservations along with the status of each reservation.



My reservations →

Pending

Noor	Name
12	Age
female	Gender
good	Health condition
27/5/2024	Date from
29/5/2024	Date to

Approved

Mariam	Name
6	Age
female	Gender
good	Health condition
27/5/2024	Date from
29/5/2024	Date to

Declined

Meme	Name
3	Age
female	Gender
good	Health condition
27/5/2024	Date from
29/5/2024	Date to

Figure 15 Nanny reservation’s view page for the manager

12.12Activities page for children

When parents access the activities page for the child, they will see multiple options for activities that they can access, for example, among the available activities are mathematical operations.

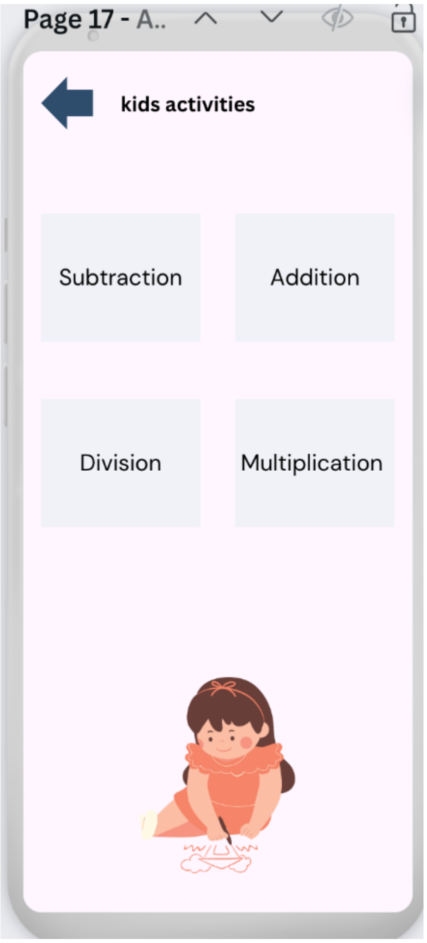


Figure 16 Activities page for children1

When you enter one of the sections, this page will appear, from which the appropriate section for the child’s age will be selected, and then a video of mathematical operations will be displayed.

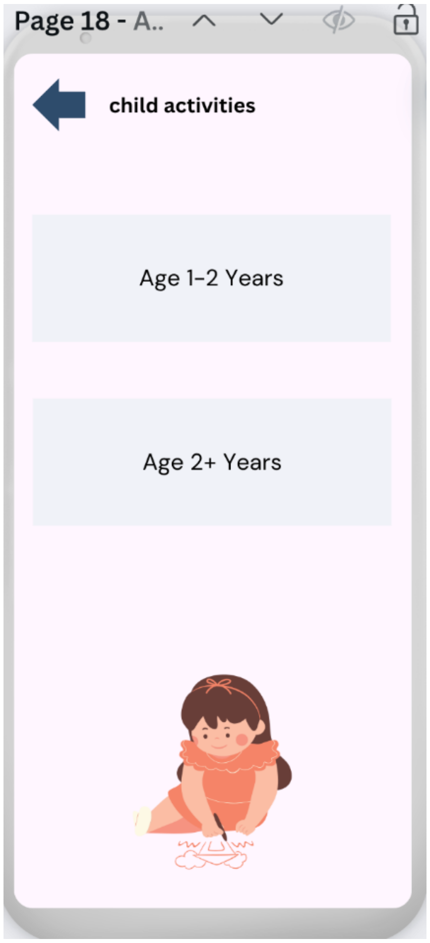


Figure 17Activities page for children 2

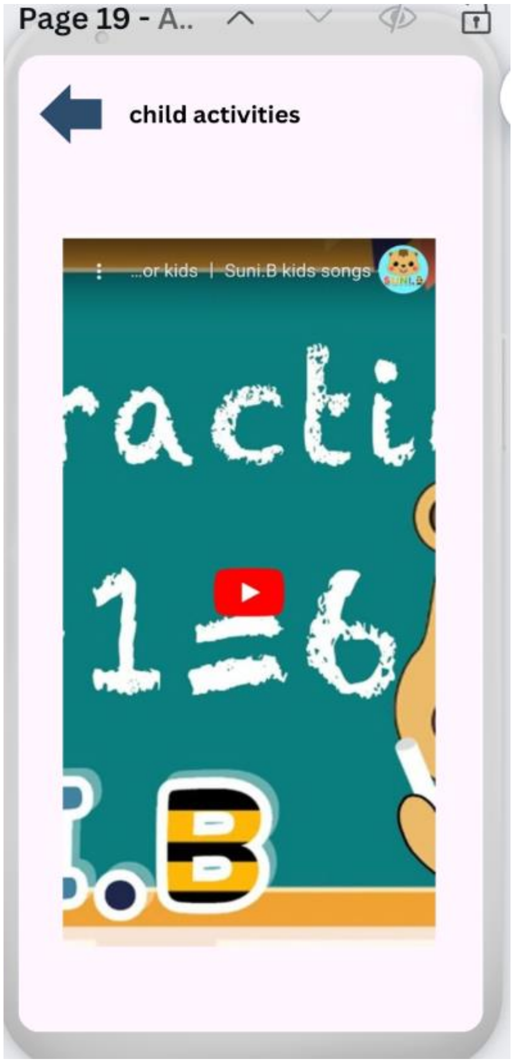


Figure 18 Activities page for children 3

12.13Food order home page for the user

The interface shown displays a food ordering page. It includes a search bar, an order button, food categories like pizza and fruits, and popular food items. At the bottom, there are icons for navigating to the user profile, shopping cart, and home page.

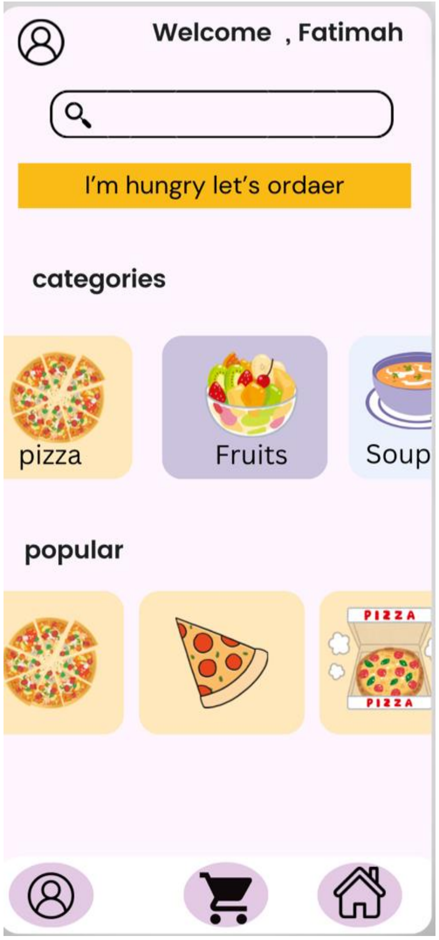


Figure 19 Food order home page

12.14The sections inside the food

When the user enters one of the food sections, the menu will appear to him, and he can enter the food plate

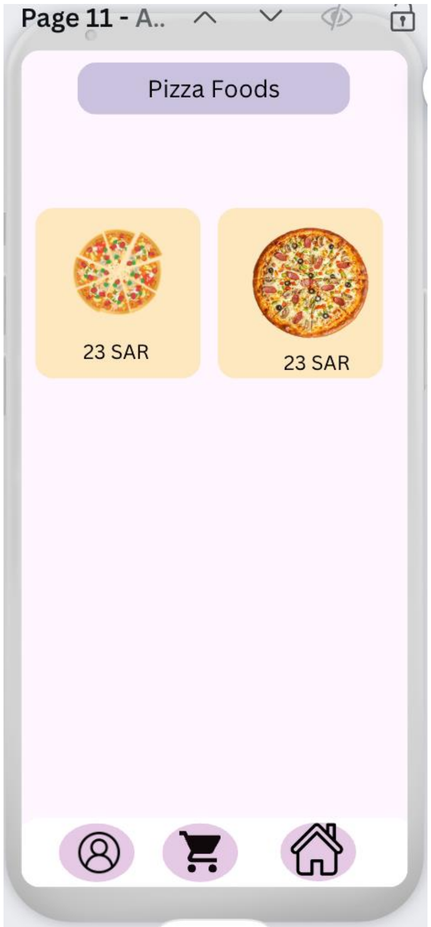


Figure 20 The sections inside the food

When the user enters a specific dish that he wants to buy, some information about this food will appear to him, with the condition that he specify the quantity he wants so that he can purchase

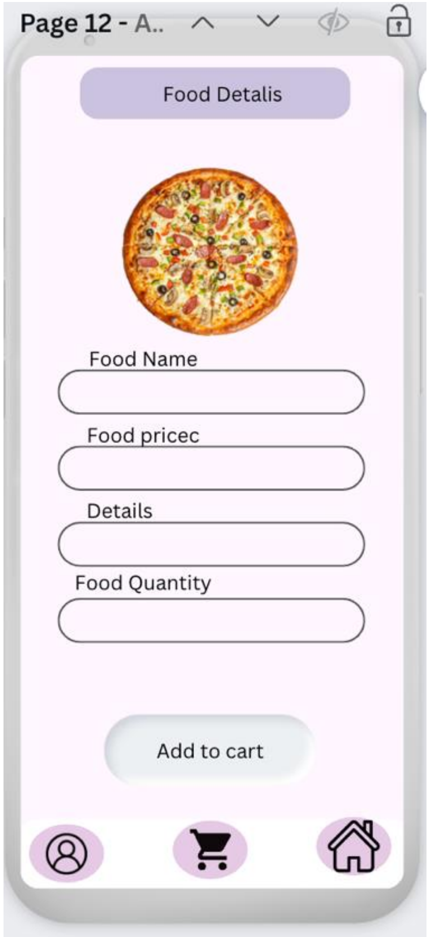
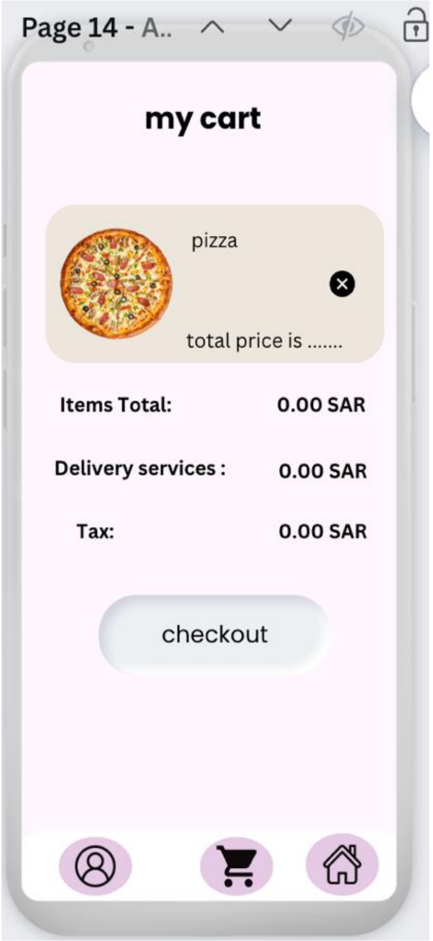


Figure 21 Food Detalis

The user can see the food on the cart page before completing the payment process



Items Total: 0.00 SAR

Delivery services : 0.00 SAR

Tax: 0.00 SAR

Figure 22 my cart

12.15Payment on the food page

The interface shown displays a checkout page with fields for the user to enter their name, card number, phone number, and address. It includes buttons to cancel or confirm the checkout process. At the bottom, there are icons for navigating to the user profile, shopping cart, and home page.

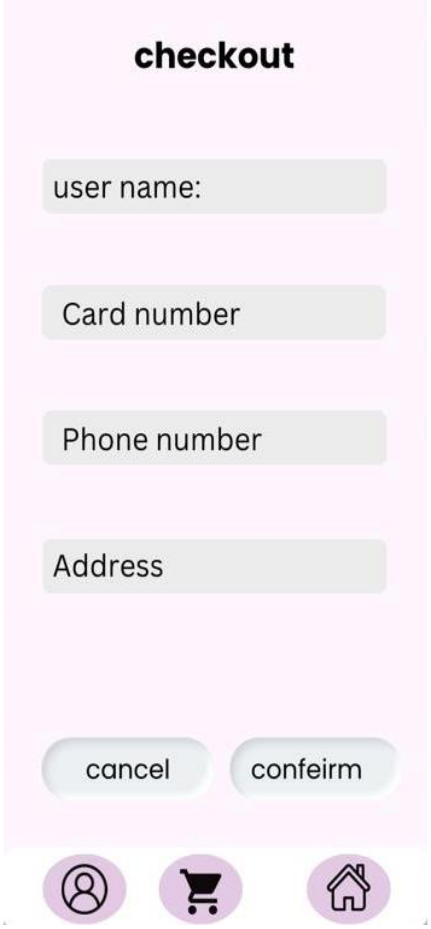


Figure 23checkout

12.16 Food order page for the manager

When the manager logs in with his email and password, he can enter and add new dishes of food

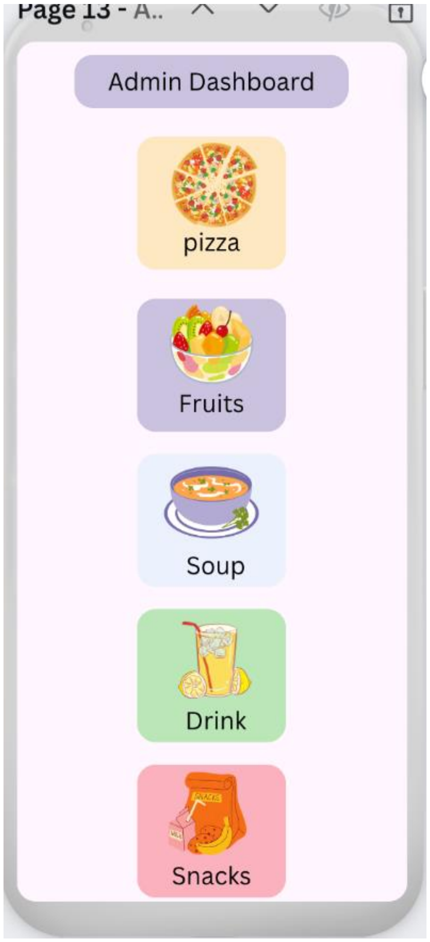


Figure 24 Admin Dashboard

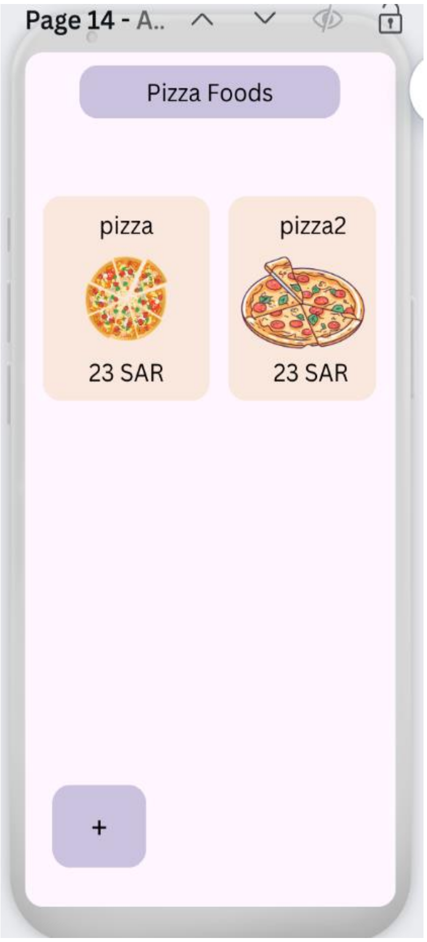


Figure 25 Pizza Foods

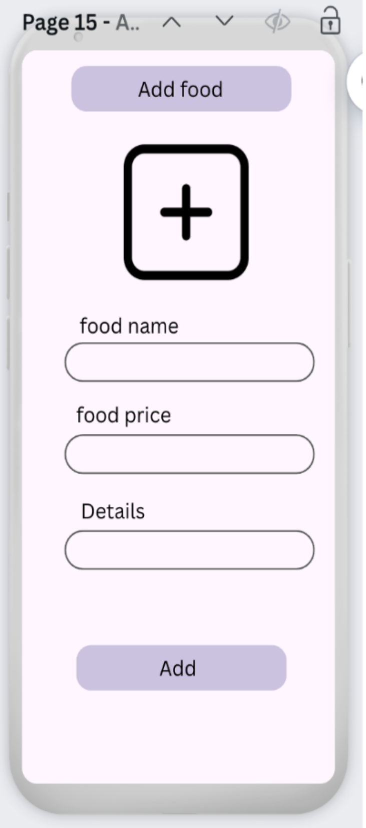
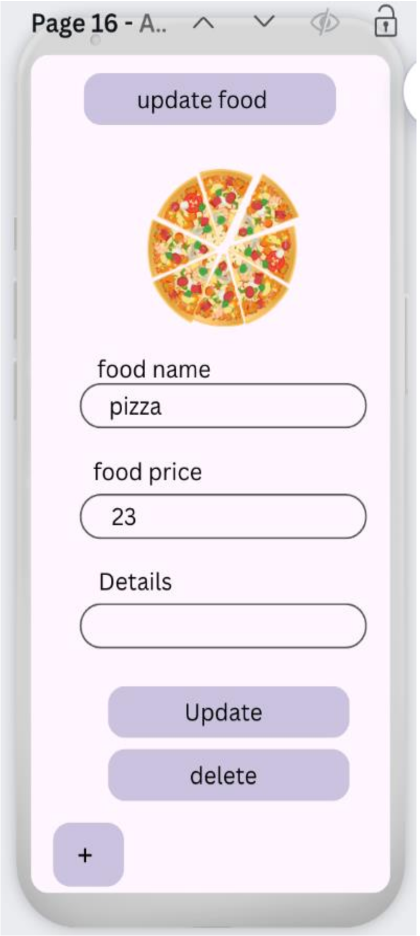


Figure 26 Add food

The manager can also update or delete the food page



food name
pizza

food price
23

Details

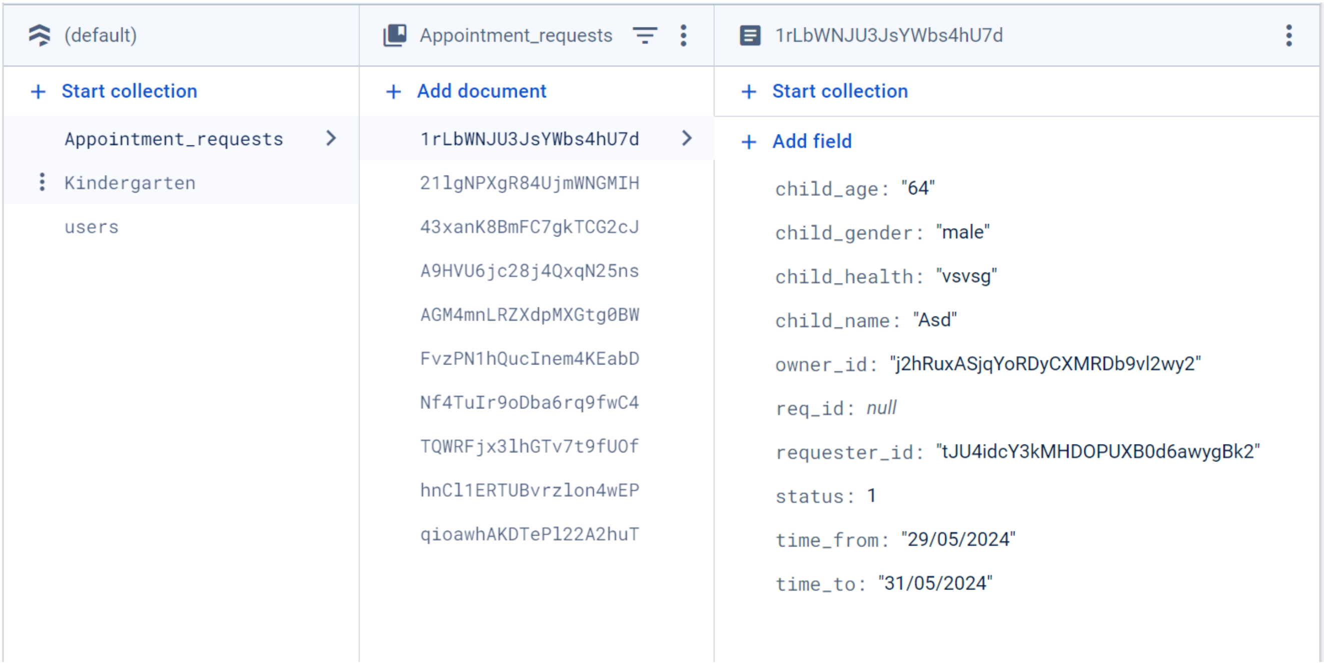
Figure 27update food

13.Database Design

The database was created using Firebase platform which is google supported platform that could be connected to mobile applications or websites.

13.1 appointment requests table

This table is responsible for holding data related to new appointments’ requests. Every time a mother requests a new appointment a new record will be created in this table.



Appointment_requests
Kindergarten
users

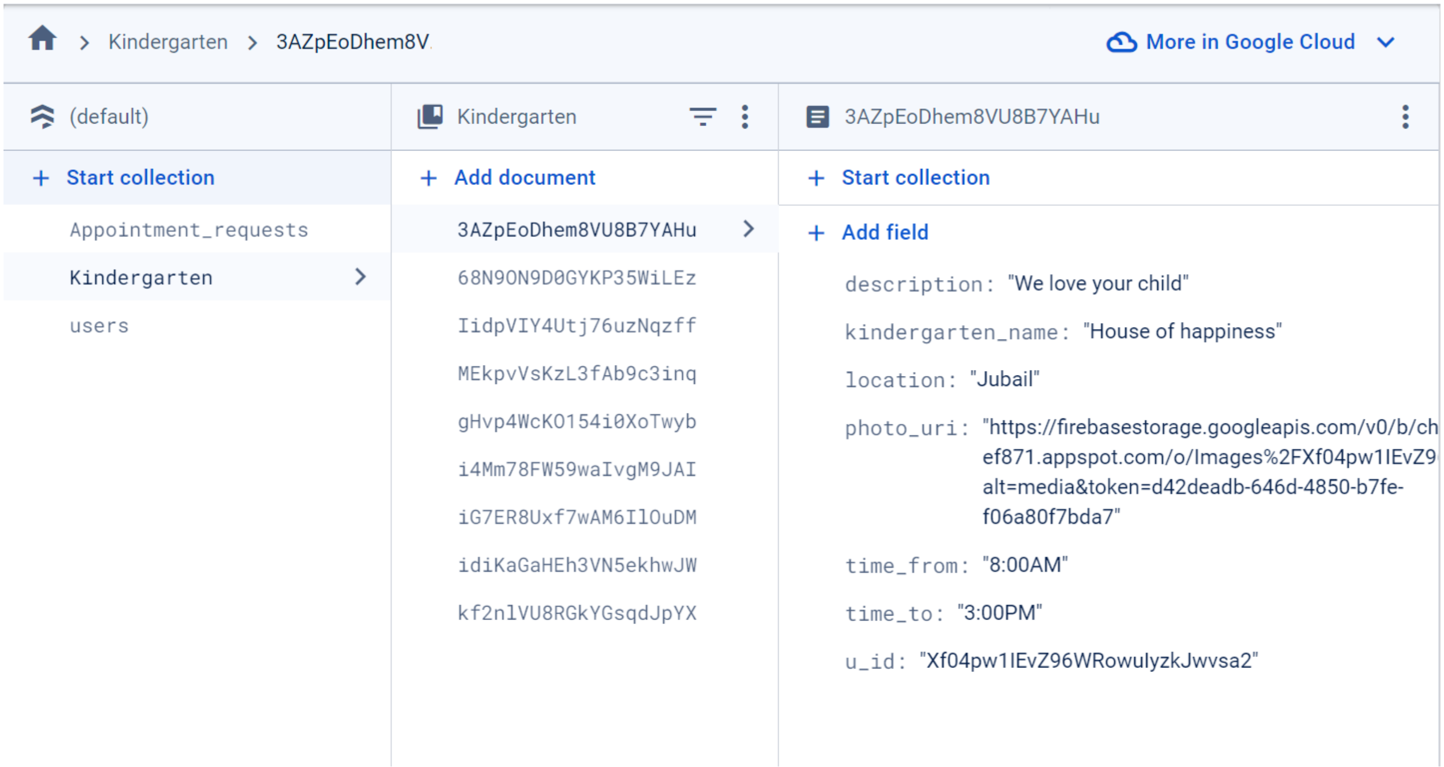
1rLbWNJU3JsYWbs4hU7d
21lgNPXgR84UjmWNGMIH
43xanK8BmFC7gkTCG2cJ
A9HVU6jc28j4QxqN25ns
AGM4mnLRZXdpMXGtg0BW
FvzPN1hQucInem4KEabD
Nf4TuIr9oDba6rq9fwC4
TQWRFjx3lhGTv7t9fUOf
hnCl1ERTUBvrzlon4wEP
qioawhAKDTePl22A2huT

child_age: "64"
child_gender: "male"
child_health: "vsvsg"
child_name: "Asd"
owner_id: "j2hRuxASjqYoRDyCXMRDb9vl2wy2"
req_id: *null*
requester_id: "tJU4idcY3kMHDOPUXB0d6awygBk2"
status: 1
time_from: "29/05/2024"
time_to: "31/05/2024"

Figure 28 appointment requests table

13.2 Kindergarten table

This table is responsible for holding the information of each kindergarten.



Appointment_requests
Kindergarten
users

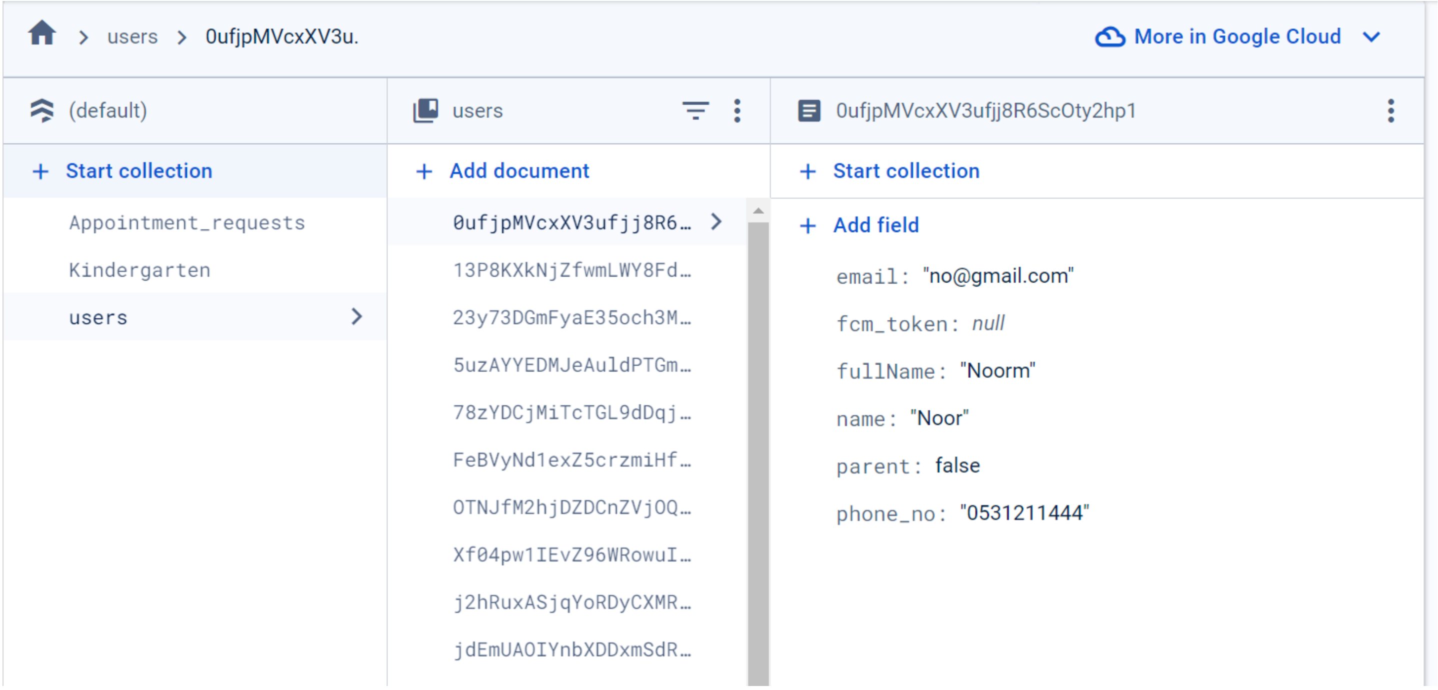
3AZpEoDhem8VU8B7YAHu
68N9ON9D0GYKP35WiLEz
IidpVIY4Utj76uzNqzff
MEkpvVsKzL3fAb9c3inq
gHvp4WcK0154i0XoTwyb
i4Mm78FW59waIvgM9JAI
iG7ER8Uxf7wAM6Il0uDM
idiKaGaHEh3VN5ekhwJW
kf2nlVU8RGkYGsqdJpYX

description: "We love your child"
kindergarten_name: "House of happiness"
location: "Jubail"
photo_uri: "https://firebasestorage.googleapis.com/v0/b/ch
ef871.appspot.com/o/Images%2FXf04pw1IEvZ9
alt=media&token=d42deadb-646d-4850-b7fe-
f06a80f7bda7"
time_from: "8:00AM"
time_to: "3:00PM"
u_id: "Xf04pw1IEvZ96WRowulyzkJwvsa2"

Figure 29 Kindergarten table

13.3 Users table

The users table was created to hold user's information.



Appointment_requests
Kindergarten
users

0ufjpMVcxXV3ufjj8R6…
13P8KXkNjZfwmLWY8Fd…
23y73DGmFyaE35och3M…
5uzAYYEDMJeAuldPTGm…
78zYDCjMiTcTGL9dDqj…
FeBVyNd1exZ5crzmiHf…
OTNJfM2hjDZDCnZVjOQ…
Xf04pw1IEvZ96WRowuI…
j2hRuxASjqYoRDyCXMR…
jdEmUAOIYnbXDDxmSdR…

email: "no@gmail.com"
fcm_token: *null*
fullName: "Noorm"
name: "Noor"
parent: false
phone_no: "0531211444"

Figure 30 users table

13.3 Cart table

In the Firebase database, cart items are organized with several key fields to describe each product added to the cart. These fields include:

categoryName: Specifies the category of the product (e.g., "Pizza").

description: Provides a detailed description of the product.

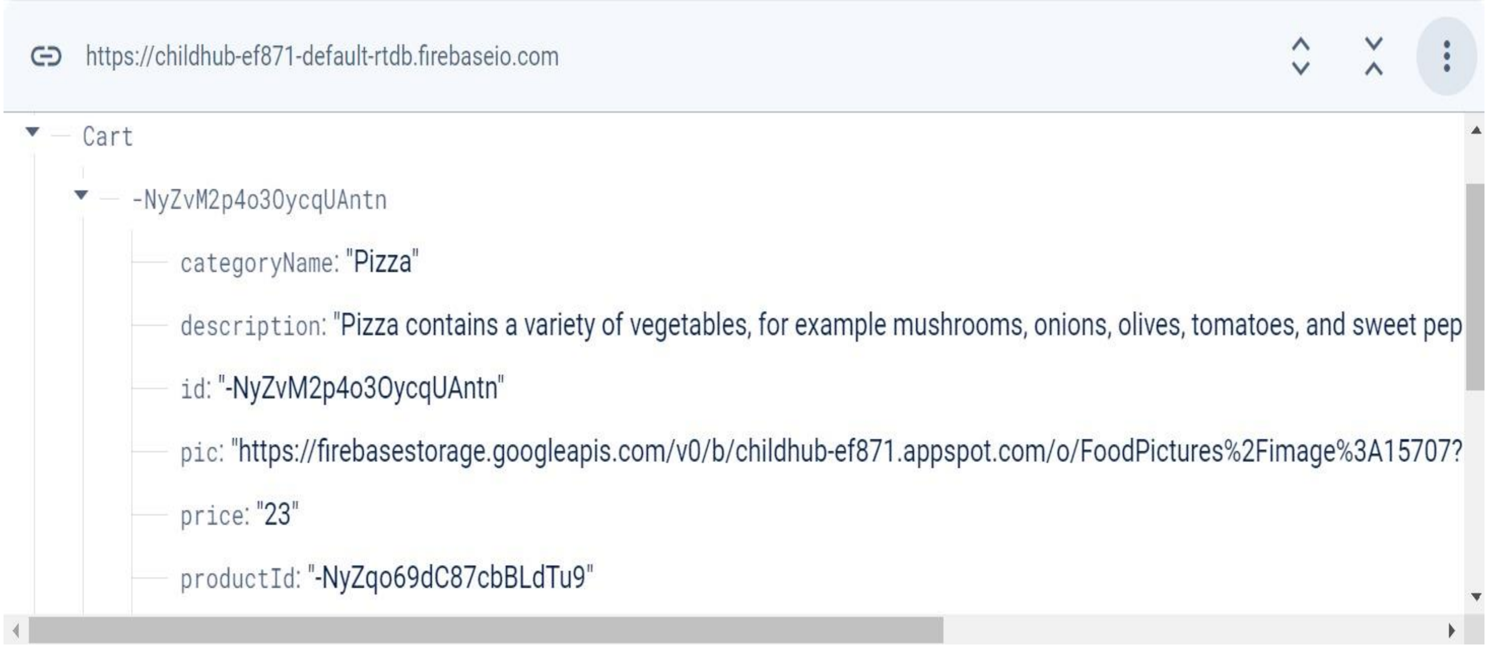
id: A unique identifier for the cart item in the database.

pic: A URL link to the product image stored in Firebase Storage.

price: The price of the product.

productId: A unique identifier for the product itself, distinct from the cart item ID.

This structured approach ensures that each cart item is well-documented with all necessary details for easy retrieval and display in the shopping cart functionality of the application.



```
https://childhub-ef871-default-rtdb.firebaseio.com
Cart
  -NyZvM2p4o3OycqUAntn
    categoryName: "Pizza"
    description: "Pizza contains a variety of vegetables, for example mushrooms, onions, olives, tomatoes, and sweet pep
    id: "-NyZvM2p4o3OycqUAntn"
    pic: "https://firebasestorage.googleapis.com/v0/b/childhub-ef871.appspot.com/o/FoodPictures%2Fimage%3A15707?
    price: "23"
    productId: "-NyZqo69dC87cbBLdTu9"
```

Figure 31 Cart table

14.5 Food table

In the Firebase database, food items are organized with several key fields to describe each product. These fields include:

categoryName: Specifies the category of the product (e.g., "Pizza").

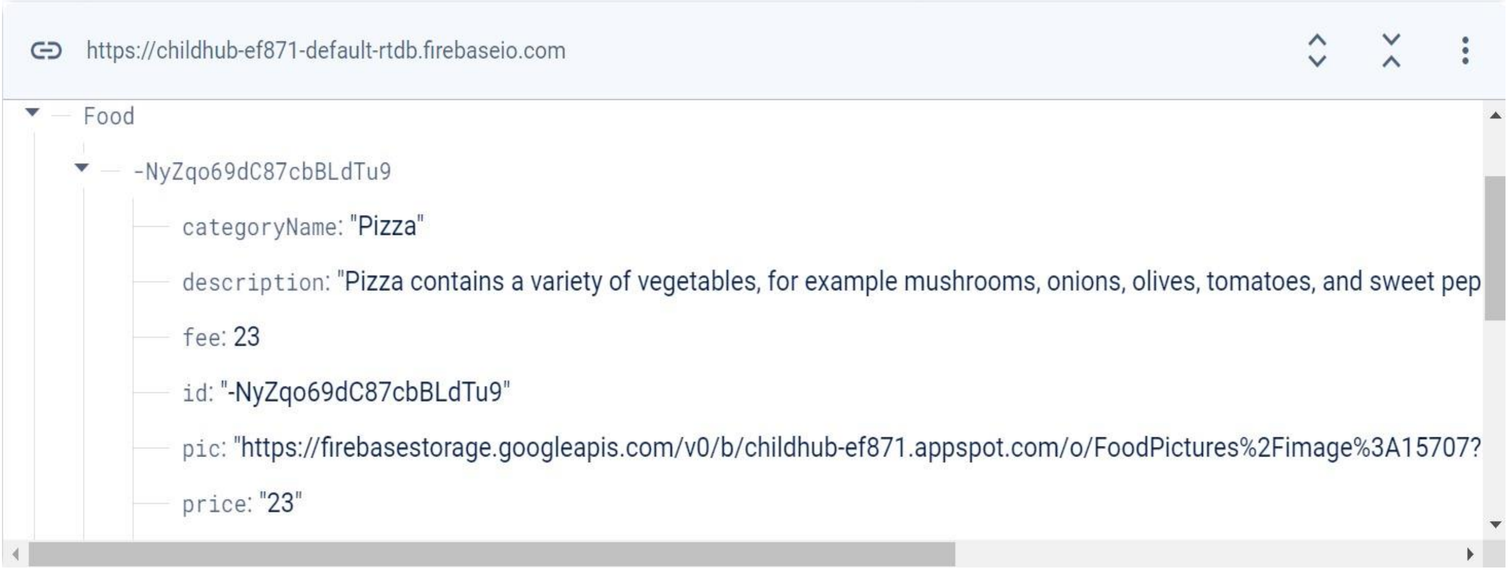
description: Provides a detailed description of the product.

fee: Indicates the fee associated with the product.

id: A unique identifier for the food item in the database.

pic: A URL link to the product image stored in Firebase Storage.

price: The price of the product.



```
https://childhub-ef871-default-rtdb.firebaseio.com
Food
  -NyZqo69dC87cbBLdTu9
    categoryName: "Pizza"
    description: "Pizza contains a variety of vegetables, for example mushrooms, onions, olives, tomatoes, and sweet pep
    fee: 23
    id: "-NyZqo69dC87cbBLdTu9"
    pic: "https://firebasestorage.googleapis.com/v0/b/childhub-ef871.appspot.com/o/FoodPictures%2Fimage%3A15707?
    price: "23"
```

Figure 32 Food table

14.5 Order table

The Orders table contains a collection of orders, with each order having the following information:

categoryName: The category of the food item. In this example, the category is "Pizza."

description: A description of the food item, indicating that the pizza contains a variety of vegetables such as mushrooms, onions, olives, tomatoes, and sweet peppers.

id: A unique identifier for the order.

pic: A link to the image of the food item. The link points to Firebase Storage where images are stored.

price: The price of the order. In this example, the price is "23."

productId: A unique identifier for the product.

quantity: The quantity of the product ordered. In this example, the quantity is "1."

title: The title of the order, here it is "vegetable pizza."

userAddress: The address of the user, in this example, it is "csvsgsgsg."

userCardNumber: The user's card number, a placeholder number "6666 6644 4664 4446" is used here.

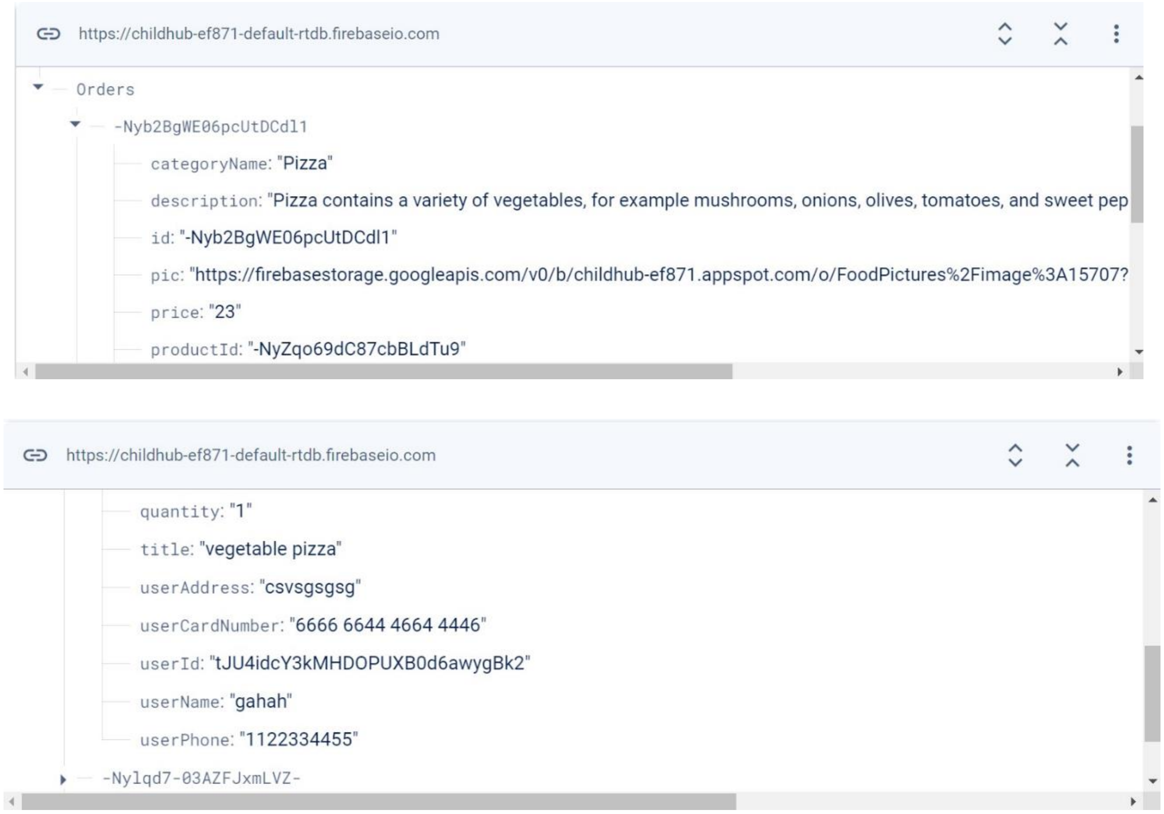
userId: A unique identifier for the user.

userName: The user's name, here it is "gahah."

userPhone: The user's phone number, here it is "1122334455."

Each order is identified by a unique ID, which helps distinguish between different orders.

User details include name, address, phone number, and user ID, facilitating order tracking and user communication.



```
https://childhub-ef871-default-rtdb.firebaseio.com

Orders
  -Nyb2BgWE06pcUtDCdl1
    categoryName: "Pizza"
    description: "Pizza contains a variety of vegetables, for example mushrooms, onions, olives, tomatoes, and sweet pep
    id: "-Nyb2BgWE06pcUtDCdl1"
    pic: "https://firebasestorage.googleapis.com/v0/b/childhub-ef871.appspot.com/o/FoodPictures%2Fimage%3A15707?
    price: "23"
    productId: "-NyZqo69dC87cbBLdTu9"
```

```
https://childhub-ef871-default-rtdb.firebaseio.com

    quantity: "1"
    title: "vegetable pizza"
    userAddress: "csvsgsgsg"
    userCardNumber: "6666 6644 4664 4446"
    userId: "tJU4idcY3kMHDOPUXB0d6awygBk2"
    userName: "gahah"
    userPhone: "1122334455"
  -Nylqd7-03AZFJxmLVZ-
```

Figure 33 Order table

14.A new concept was implemented in the project

In our ongoing efforts to improve user experience and security, a new concept was introduced into the project to enhance password management. This innovative feature allows users to easily set a new password by clicking on a provided link and entering their desired new password directly.

Moreover, we have streamlined the process for users who forget their passwords. By simply entering their email address and clicking "Done," users will receive an email with instructions to reset their password. This approach ensures a smooth and secure user experience, reinforcing the project's commitment to robust security measures and user-friendly design.Additionally, a new concept was implemented to enhance the educational experience for children. When users click on the Children’s Activities section, they are directed to the main page for children’s activities, which includes sections on addition, subtraction, multiplication, and division.

Upon selecting any of these sections, the activities are further categorized according to different age groups, ensuring that the content is appropriate and engaging for each developmental stage. This structured approach makes learning more accessible and enjoyable for children, aligning with the project's goal of fostering educational growth through innovative and user-friendly design.

Furthermore, a new feature has been integrated into the project: the inclusion of YouTube videos. These videos provide interactive and visual learning experiences, enhancing the understanding and retention of mathematical concepts. This addition enriches the educational content, making learning more dynamic and effective for children.

15.Implementing the interfaces and explaining it step by step:

15.1 Welcome page

This page appears to the user as soon as he opens the application, and when he presses the let’s Go button, he goes to the account creation page.

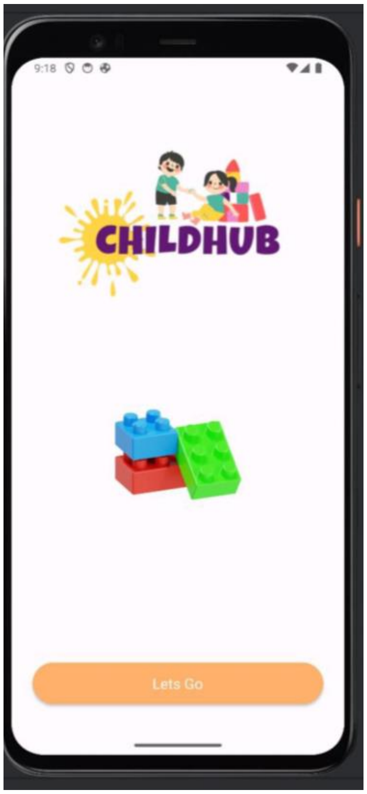
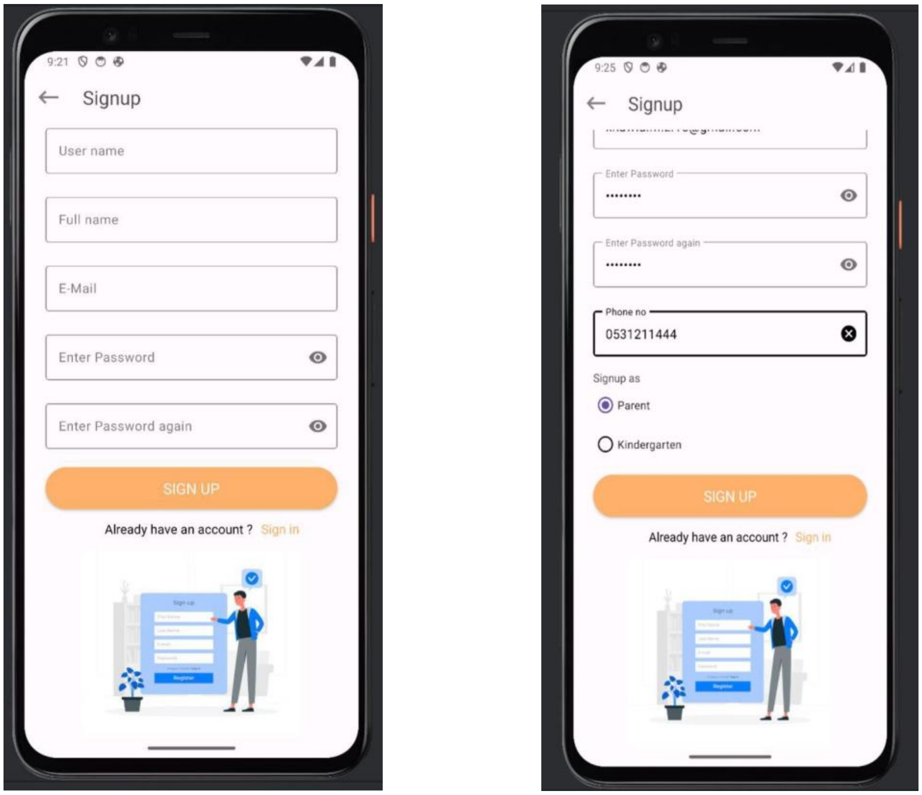


Figure 34 Welcome page

15.2 Create an account

When you enter the application for the first time, you must create an account and fill out the following information, such as the user name, full name, email, password, and mobile number, then choose whether I want to sign up as a parent or as a kindergarten, then press the sin up button.



9:21

← Signup

User name

Full name

E-Mail

Enter Password

Enter Password again

SIGN UP

Already have an account ? Sign in

9:25

← Signup

Enter Password: ••••••••

Enter Password again: ••••••••

Phone no: 0531211444

Signup as

● Parent

○ Kindergarten

SIGN UP

Already have an account ? Sign in

Figure 35Create an account

The most important codes used in this part are:

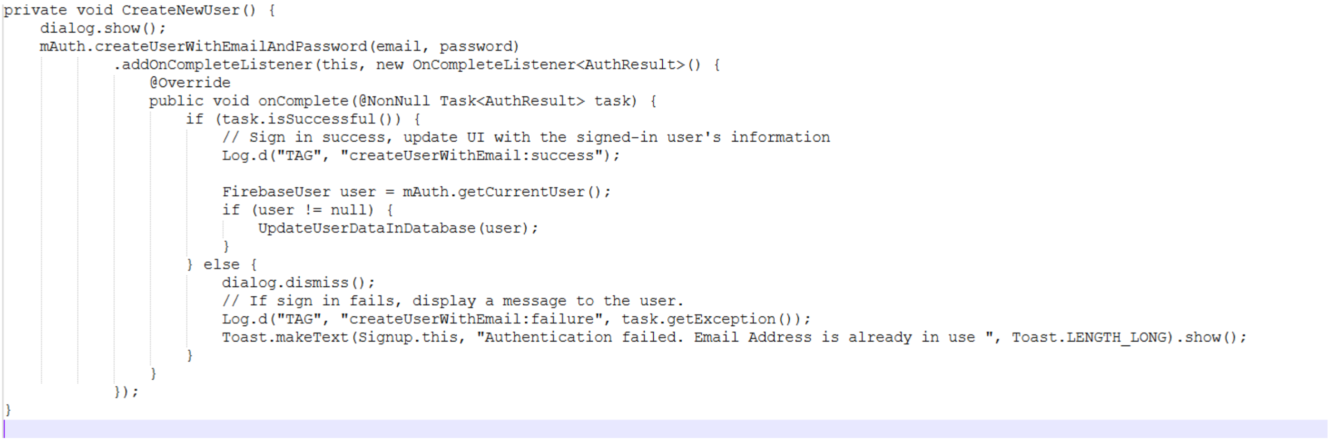
The CreateNewUser() function is crucial because it:

- Manages the user registration process.

- Interacts with Firebase Authentication to create a new user account.

- Calls UpdateUserDataInDatabase(user) to store additional user information in the Firestore database.

- Provides feedback to the user about the registration status, handling both success and failure scenarios.



```
private void CreateNewUser() {
    dialog.show();
    mAuth.createUserWithEmailAndPassword(email, password)
            .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        // Sign in success, update UI with the signed-in user's information
                        Log.d("TAG", "createUserWithEmail:success");

                        FirebaseUser user = mAuth.getCurrentUser();
                        if (user != null) {
                            UpdateUserDataInDatabase(user);
                        }
                    } else {
                        dialog.dismiss();
                        // If sign in fails, display a message to the user.
                        Log.d("TAG", "createUserWithEmail:failure", task.getException());
                        Toast.makeText(Signup.this, "Authentication failed. Email Address is already in use ", Toast.LENGTH_LONG).show();
                    }
                }
            });
}
```

15.3 sign in

After creating the account, you can log in by filling in your email and password and clicking the sin in button

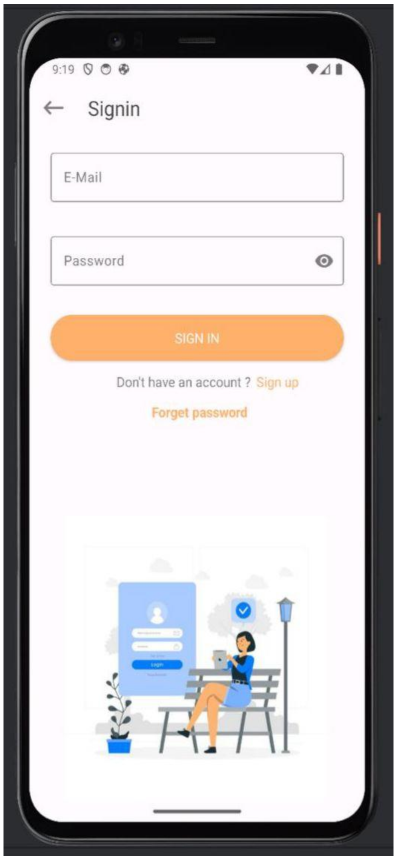


Figure 36 sign in

The most important codes used in this part are:

This function is essential because it:

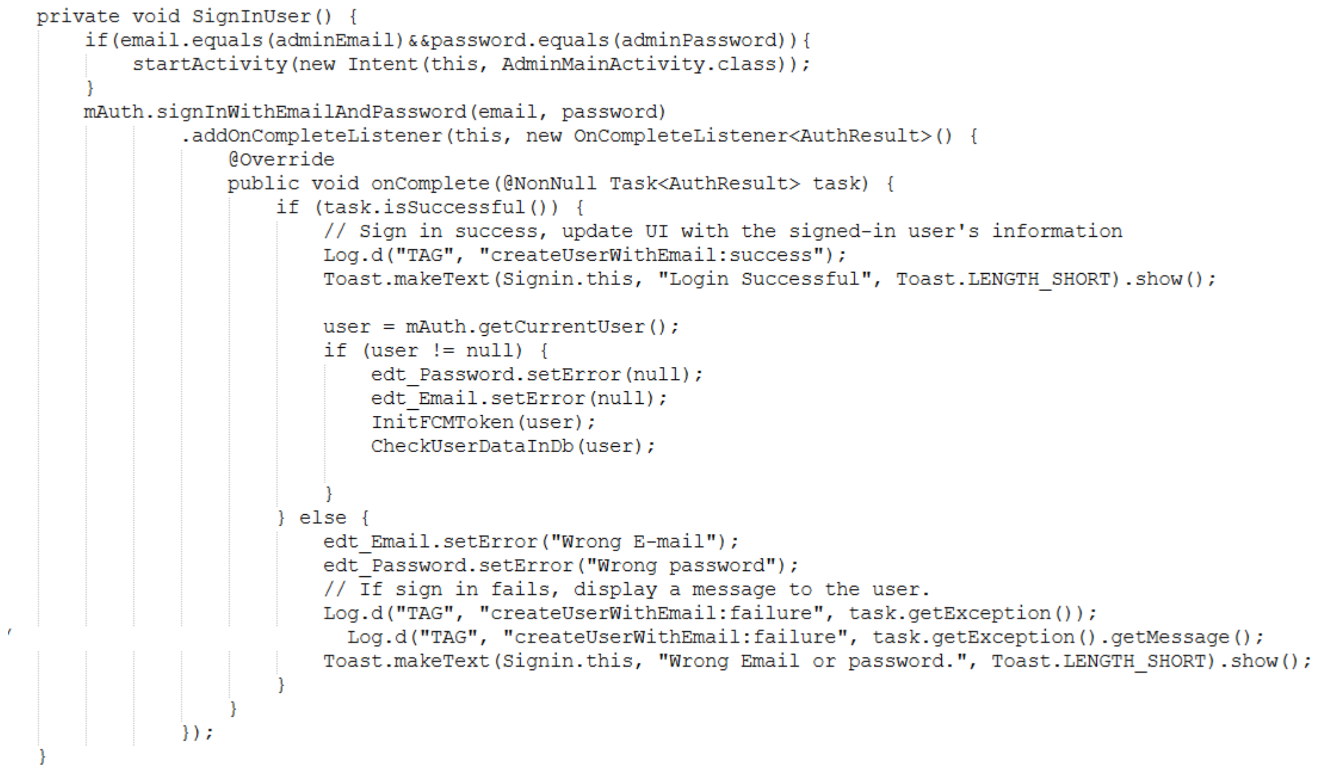
- Validates the user's email and password.

- Authenticates the user using Firebase.

- Handles user role-based redirection (e.g., admin or regular user).

- Initializes the FCM token and checks user data in the database on successful sign-in.

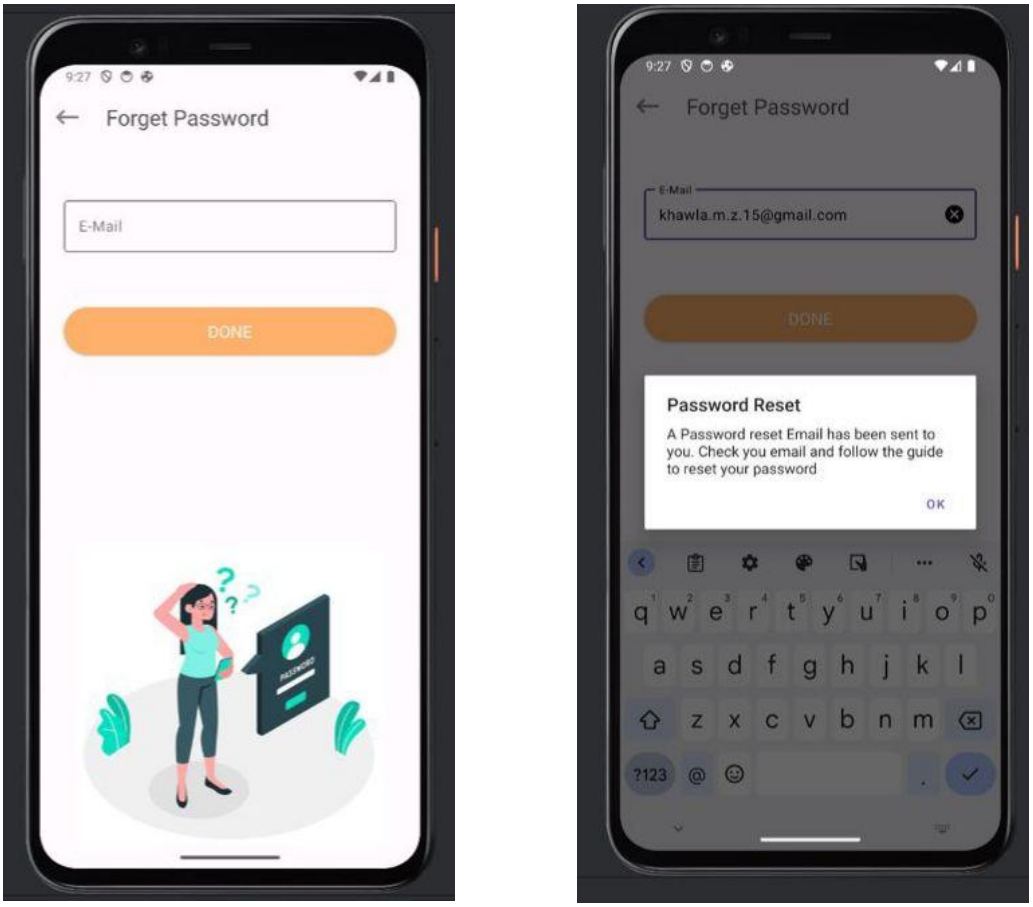
- Provides error handling for failed sign-in attempts.



```
private void SignInUser() {
    if(email.equals(adminEmail)&&password.equals(adminPassword)){
        startActivity(new Intent(this, AdminMainActivity.class));
    }
    mAuth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        // Sign in success, update UI with the signed-in user's information
                        Log.d("TAG", "createUserWithEmail:success");
                        Toast.makeText(Signin.this, "Login Successful", Toast.LENGTH_SHORT).show();

                        user = mAuth.getCurrentUser();
                        if (user != null) {
                            edt_Password.setError(null);
                            edt_Email.setError(null);
                            InitFCMToken(user);
                            CheckUserDataInDb(user);

                        }
                    } else {
                        edt_Email.setError("Wrong E-mail");
                        edt_Password.setError("Wrong password");
                        // If sign in fails, display a message to the user.
                        Log.d("TAG", "createUserWithEmail:failure", task.getException());
                          Log.d("TAG", "createUserWithEmail:failure", task.getException().getMessage();
                        Toast.makeText(Signin.this, "Wrong Email or password.", Toast.LENGTH_SHORT).show();
                    }
                }
            });
}
```

15.4 Forgot password page

If you forgot your password, you can reset it by entering your email and clicking done, and you will receive an email to reset it.



E-Mail: khawla.m.z.15@gmail.com

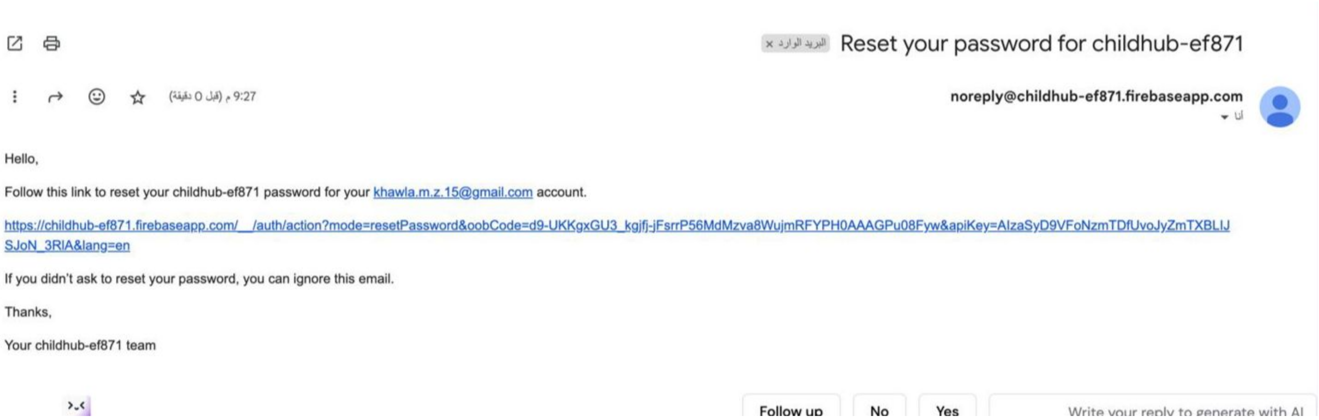
Password Reset

A Password reset Email has been sent to you. Check you email and follow the guide to reset your password

Figure 37Forgot password page1

After clicking on the link, you will enter the new password you want

This part of the project is considered a new concept



Reset your password for childhub-ef871

noreply@childhub-ef871.firebaseapp.com

9:27 م (قبل 0 دقيقة)

Hello,

Follow this link to reset your childhub-ef871 password for your khawla.m.z.15@gmail.com account.

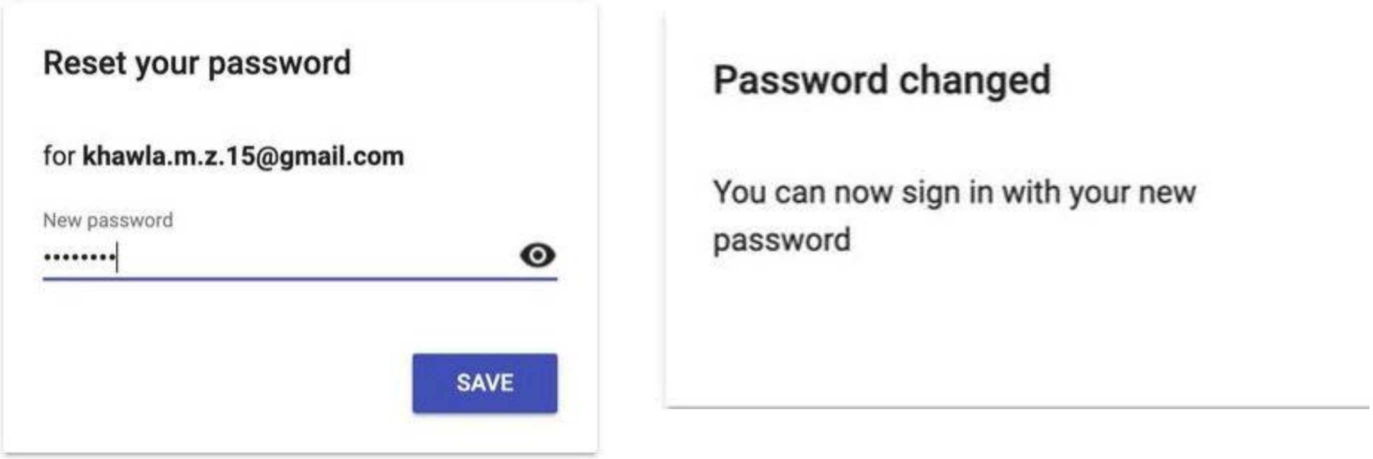
https://childhub-ef871.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=d9-UKKgxGU3_kgjfj-jFsrrP56MdMzva8WujmRFYPH0AAAGPu08Fyw&apiKey=AIzaSyD9VFoNzmTDfUvoJyZmTXBLIJSJoN_3RIA&lang=en

If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your childhub-ef871 team

Figure 38 Forgot password page2



Reset your password

for khawla.m.z.15@gmail.com

New password: ••••••••

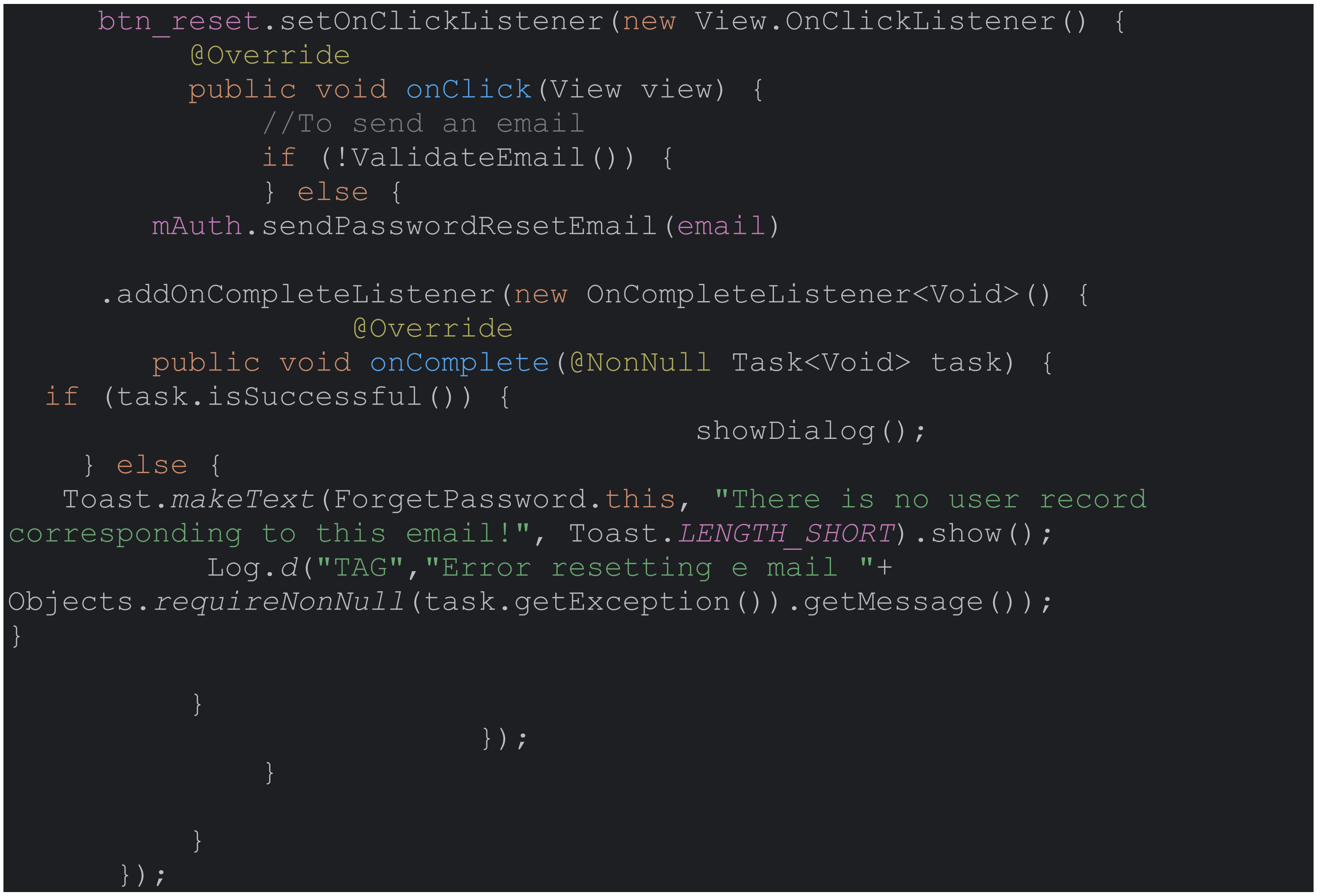
Password changed

You can now sign in with your new password

Figure 39 Forgot password page3

The most important codes used in this part are:

- If the email is valid, the application sends a password reset message to the user's email using mAuth.sendPasswordResetEmail(email).
- This function sends a request to Firebase Auth to send a password reset message to the email provided.
- If the submission is successful, the showDialog() function will be called to display a dialog window informing the user that a password reset message has been sent.
- If sending fails, an error message will be displayed using Toast.makeText() and the message will be printed



```
     btn_reset.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View view) {
              //To send an email
              if (!ValidateEmail()) {
              } else {
        mAuth.sendPasswordResetEmail(email)

     .addOnCompleteListener(new OnCompleteListener<Void>() {
                   @Override
        public void onComplete(@NonNull Task<Void> task) {
  if (task.isSuccessful()) {
                                      showDialog();
    } else {
   Toast.makeText(ForgetPassword.this, "There is no user record
corresponding to this email!", Toast.LENGTH_SHORT).show();
           Log.d("TAG","Error resetting e mail "+
Objects.requireNonNull(task.getException()).getMessage());
}

          }
                          });
              }

          }
      });
```

15.5 Home page

After logging in as a parent, the home page will appear, which contains the services that the mother needs for her child, such as child care by requesting a nanny, order food for child, or activities for child.

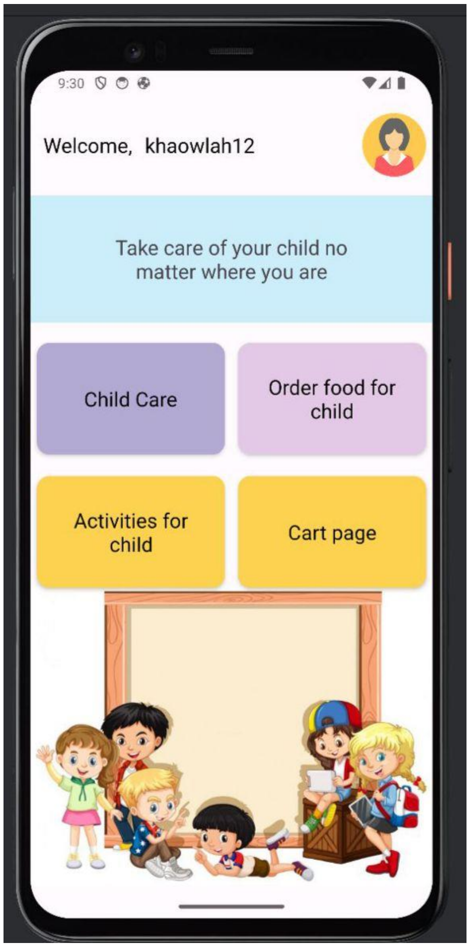
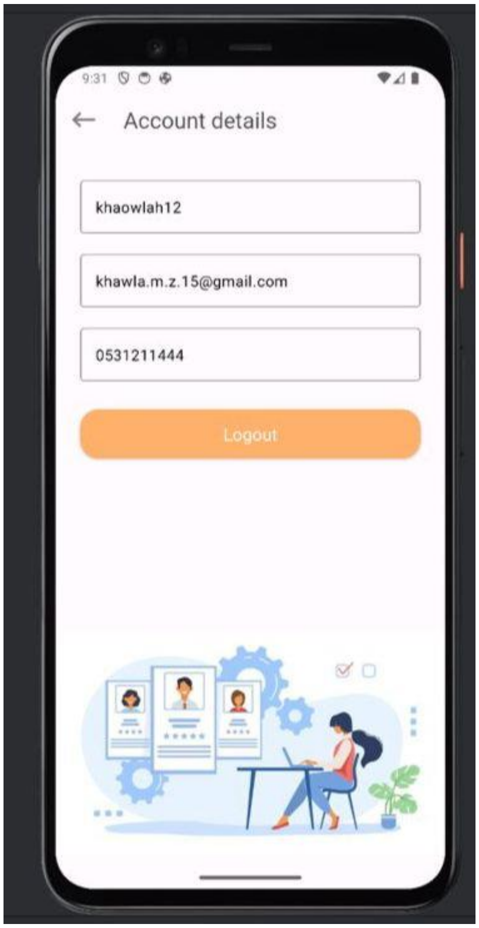


Figure 40 home page

15.6 Profile page

Here the profile and information for the parents appear, such as email, mobile number, and user name. If you want to log out, click the logout button.



khaowlah12

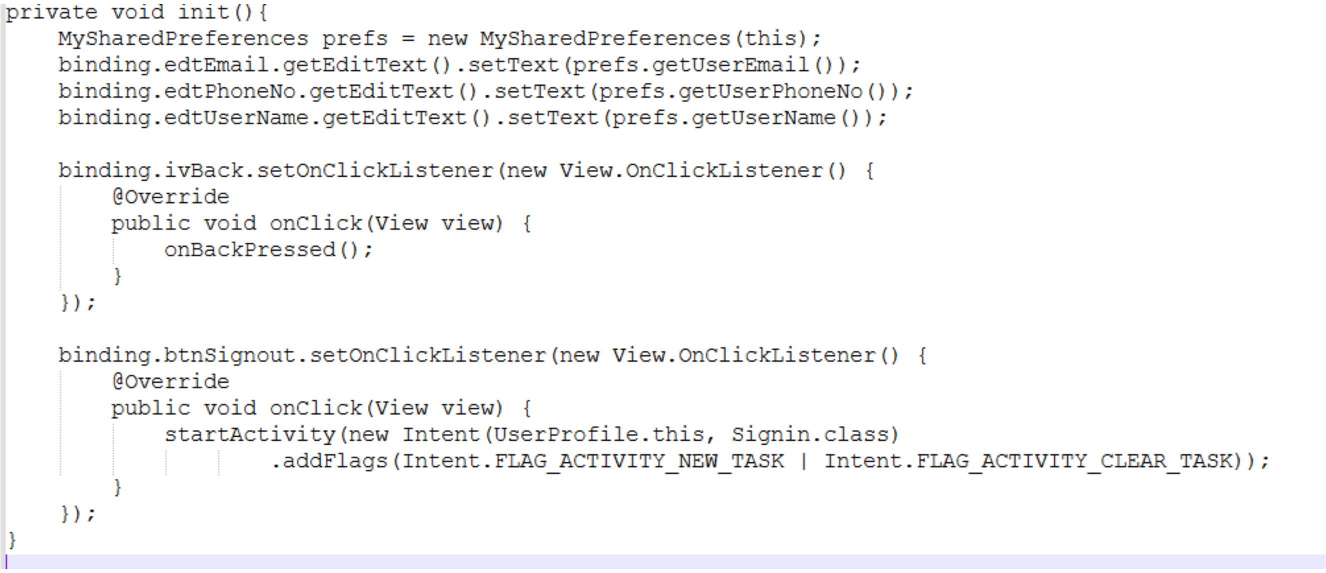
khawla.m.z.15@gmail.com

0531211444

Figure 41 User profile page

The most important codes used in this part are:

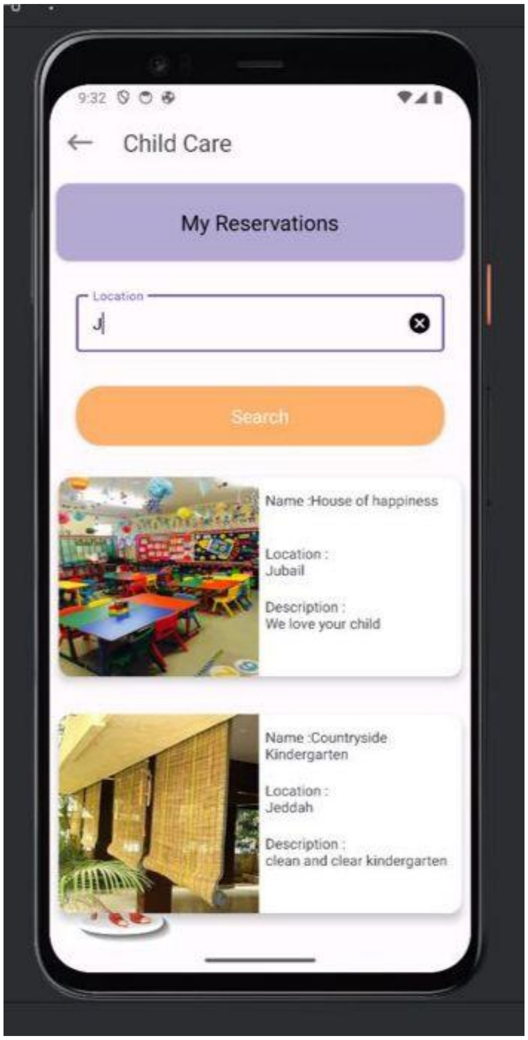
This function is key to ensuring the user data is loaded correctly and that the user can navigate back or sign out from the activity.



```
private void init(){
    MySharedPreferences prefs = new MySharedPreferences(this);
    binding.edtEmail.getEditText().setText(prefs.getUserEmail());
    binding.edtPhoneNo.getEditText().setText(prefs.getUserPhoneNo());
    binding.edtUserName.getEditText().setText(prefs.getUserName());

    binding.ivBack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            onBackPressed();
        }
    });

    binding.btnSignout.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(UserProfile.this, Signin.class)
                    .addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK));
        }
    });
}
```

15.7 Child Care Page

On the child care page, a parent can book a nanny for her child by searching for the area she wants, then showing her the available nannies in her area, choosing the nanny she wants, then pressing the “Take Appointment” button.



Location
J

Name :House of happiness

Location :
Jubail

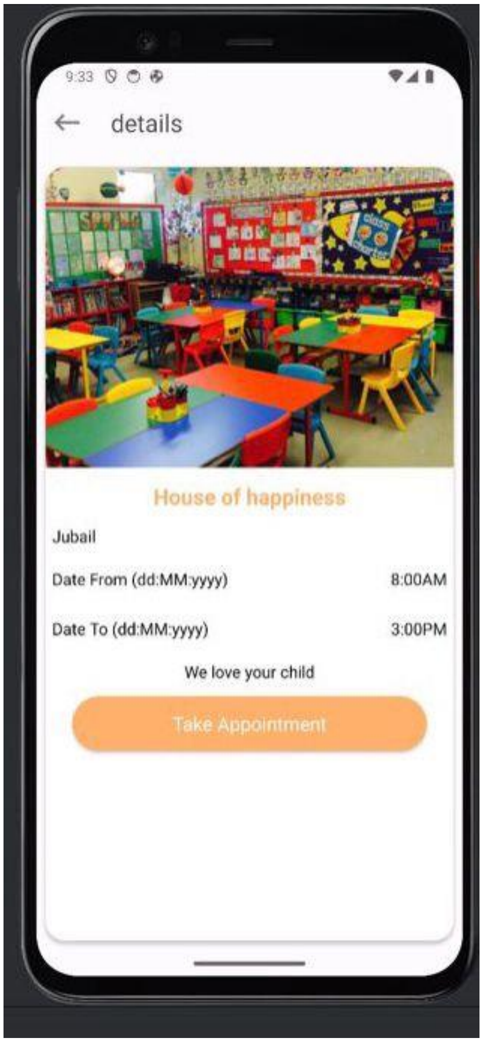
Description :
We love your child

Name :Countryside Kindergarten

Location :
Jeddah

Description :
clean and clear kindergarten

Figure 42 Child care



details

House of happiness

Jubail

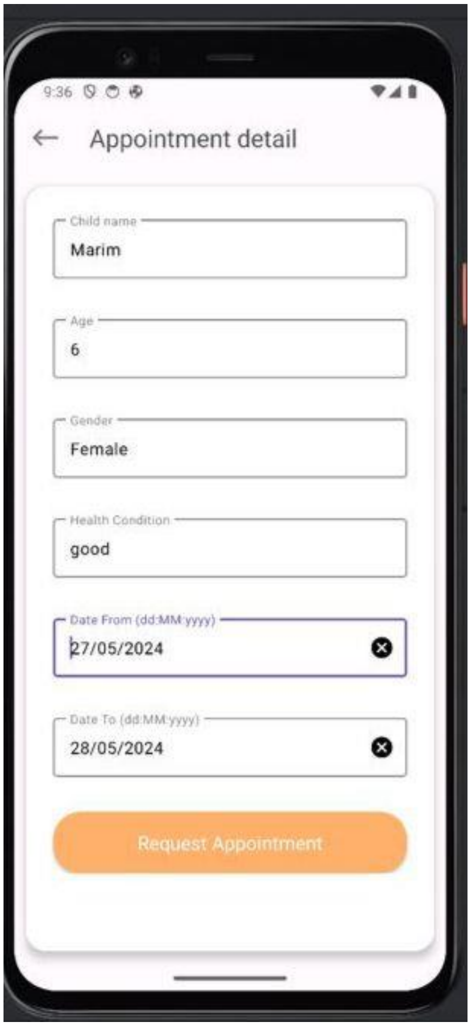
Date From (dd:MM:yyyy) 8:00AM

Date To (dd:MM:yyyy) 3:00PM

We love your child

Figure 43 Take Appointment

After clicking on the Take Appointment button, a page will appear for her to fill in the information about her child, such as the child’s name, age, gender, health condition, date from to, and time from to, then click on the Request Appointment button to book the nanny.



Child name: Marim

Age: 6

Gender: Female

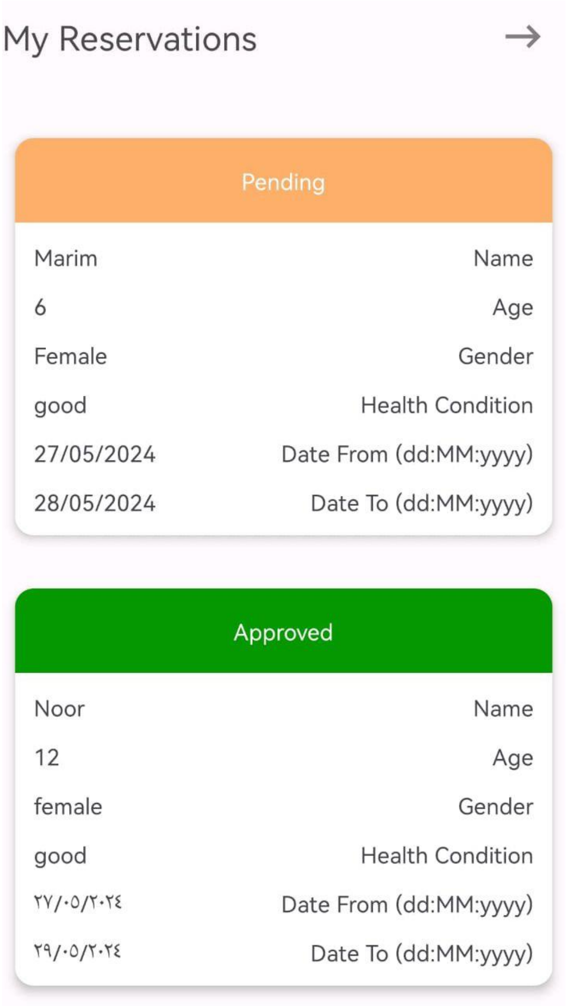
Health Condition: good

Date From (dd:MM:yyyy): 27/05/2024

Date To (dd:MM:yyyy): 28/05/2024

Figure 44 Appointment detail

The parent can track the status of the nanny’s acceptance or rejection of the request through the My reservation page



My Reservations →

Pending

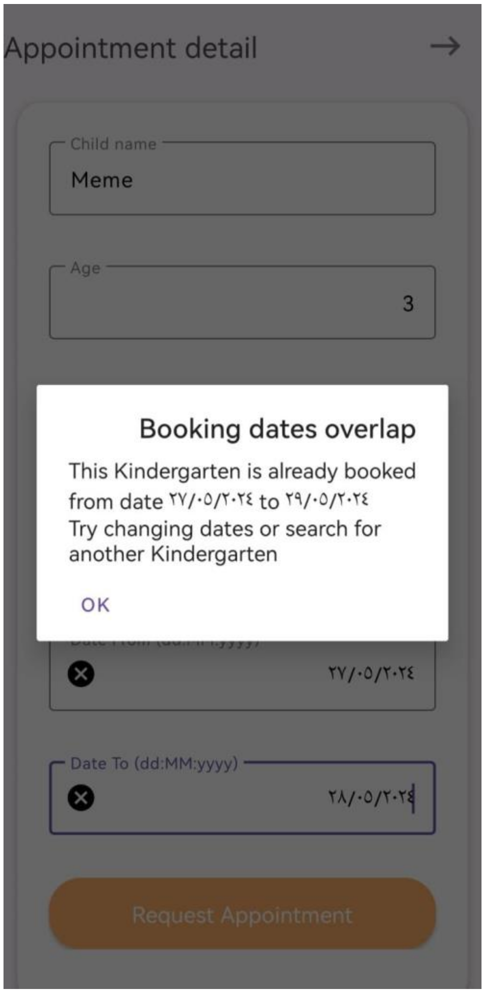
Marim	Name
6	Age
Female	Gender
good	Health Condition
27/05/2024	Date From (dd:MM:yyyy)
28/05/2024	Date To (dd:MM:yyyy)

Approved

Noor	Name
12	Age
female	Gender
good	Health Condition
٢٧/٠٥/٢٠٢٤	Date From (dd:MM:yyyy)
٢٩/٠٥/٢٠٢٤	Date To (dd:MM:yyyy)

Figure 45 My Reservations for user1

If there is a time conflict, it will appear to the parent that there is a time conflict, that another user has booked at the same time, so he can look for another nanny
or another time.



Child name: Meme

Age: 3

Booking dates overlap

This Kindergarten is already booked from date ٢٧/٠٥/٢٠٢٤ to ٢٩/٠٥/٢٠٢٤
Try changing dates or search for another Kindergarten

OK

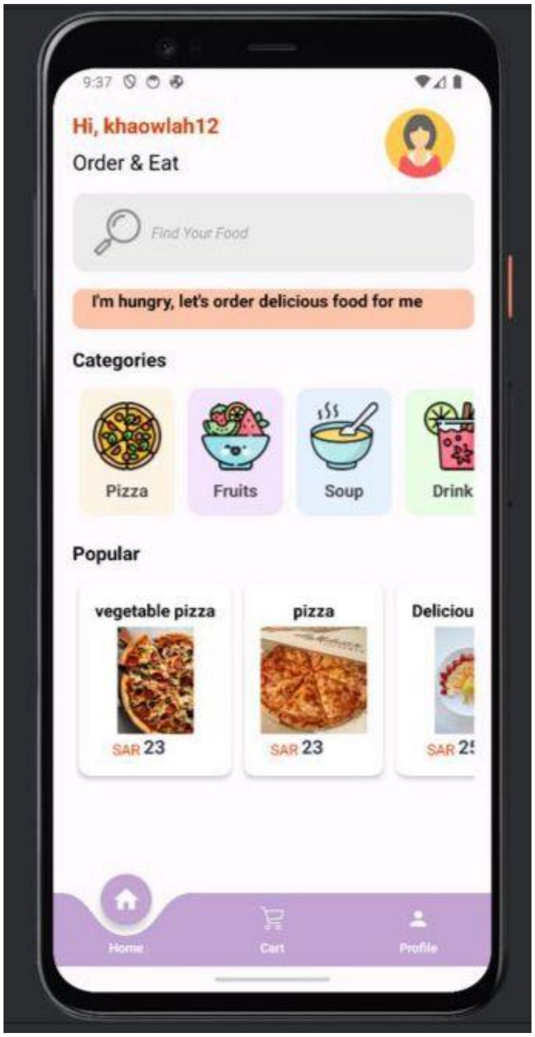
Date From (dd:MM:yyyy): ٢٧/٠٥/٢٠٢٤

Date To (dd:MM:yyyy): ٢٨/٠٥/٢٠٢٤

Figure 46 My Reservations for user2

15.8 Food order page for children

When you click on the order food for children section, the main page for ordering food will appear


Hi, khaowlah12

Order & Eat

I'm hungry, let's order delicious food for me

Figure 47Food order page

There are different sections of food suitable for children, such as pizza, fruits, soups, juices, and snacks. The parent can choose any section he wants and order food for his child.

- **This is the pizza section**

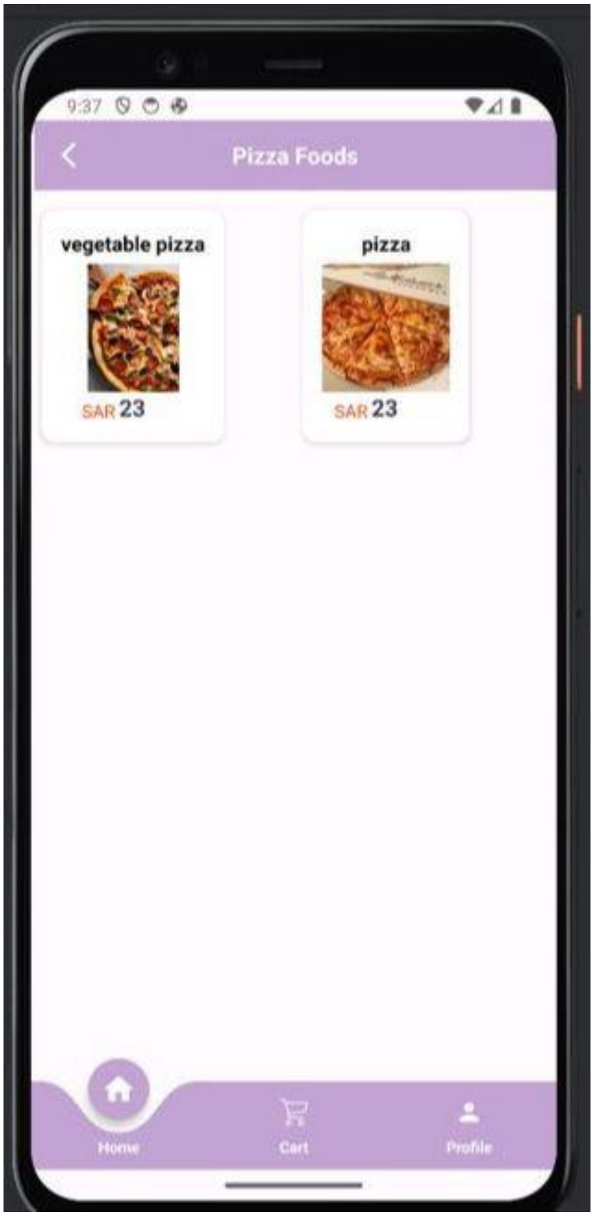


Figure 48 pizza section

- **This is the fruit section**

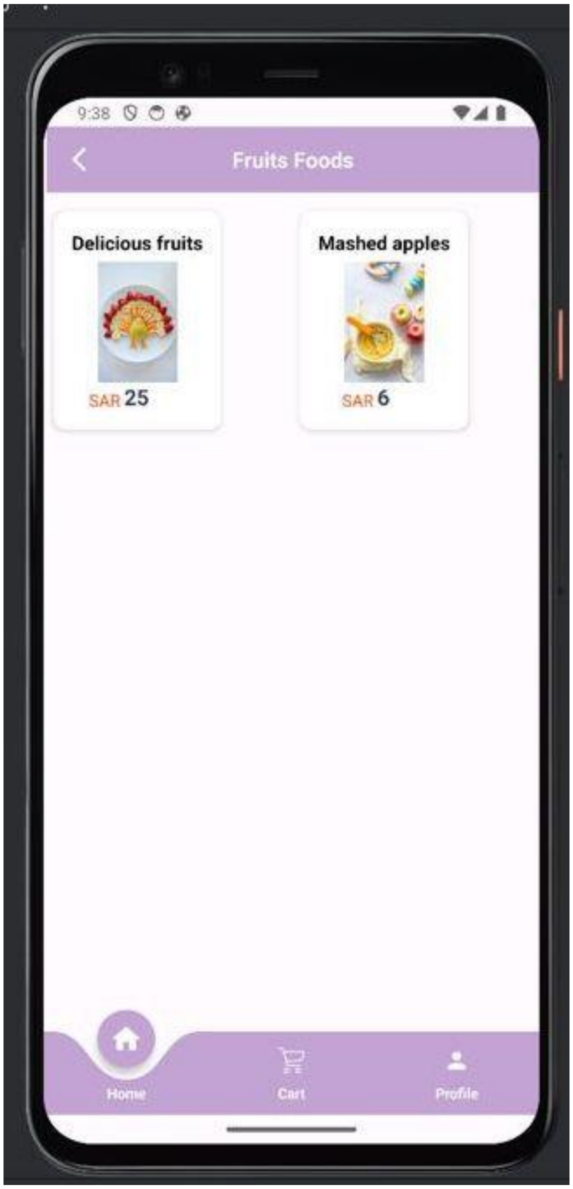


Figure 49 fruit section

- **This is the soups section**

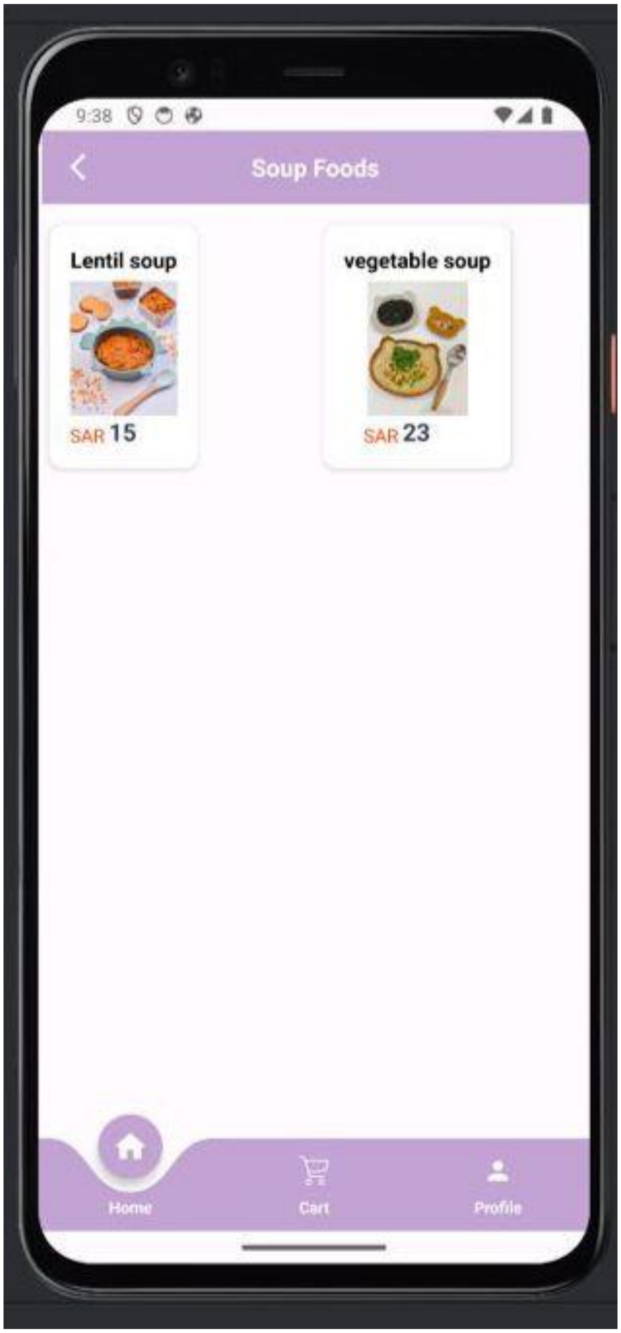
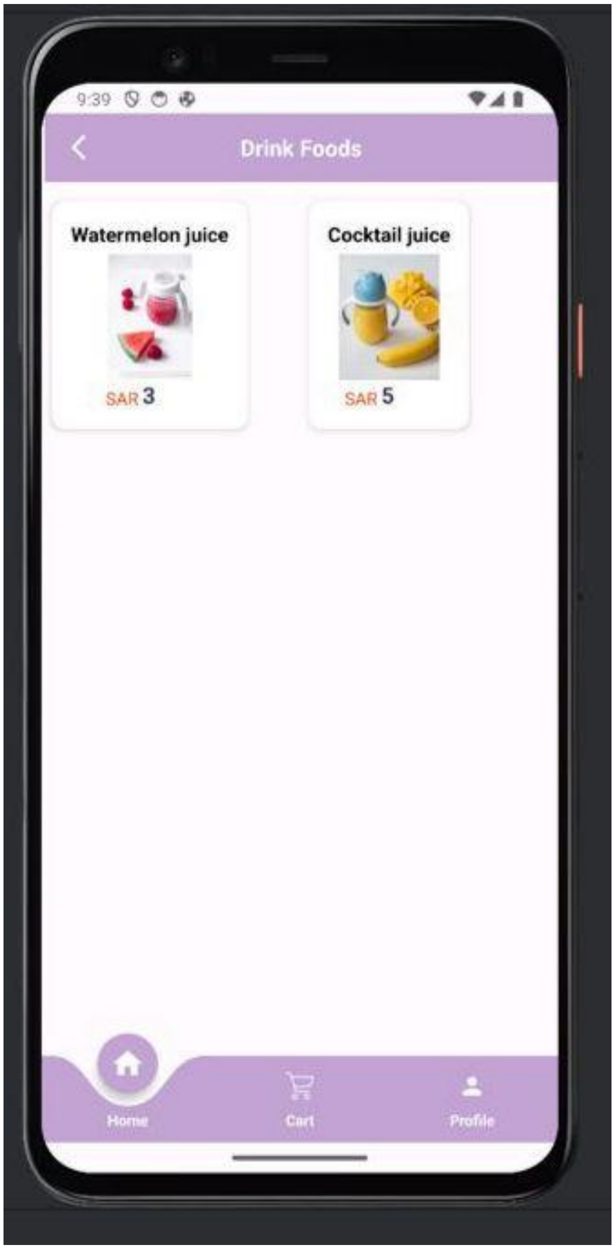


Figure 50 soups section

- **This is the drinks section**



9:39

Drink Foods

Watermelon juice

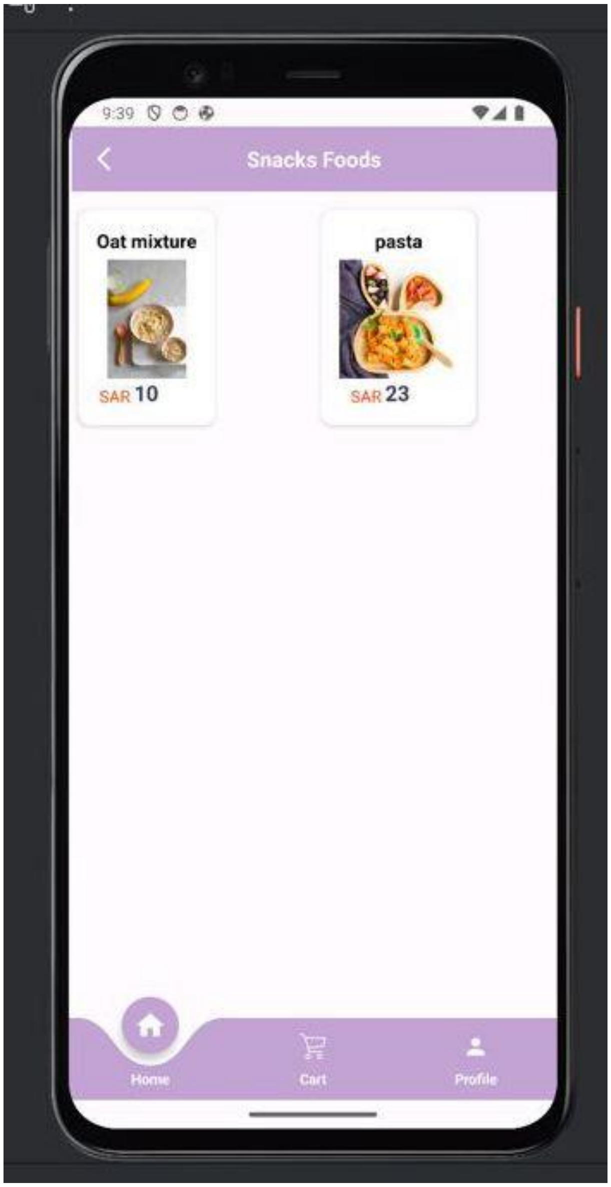
SAR 3

Cocktail juice

SAR 5

Figure 51 drinks section

- **This is the snack section**



9:39

Snacks Foods

Oat mixture

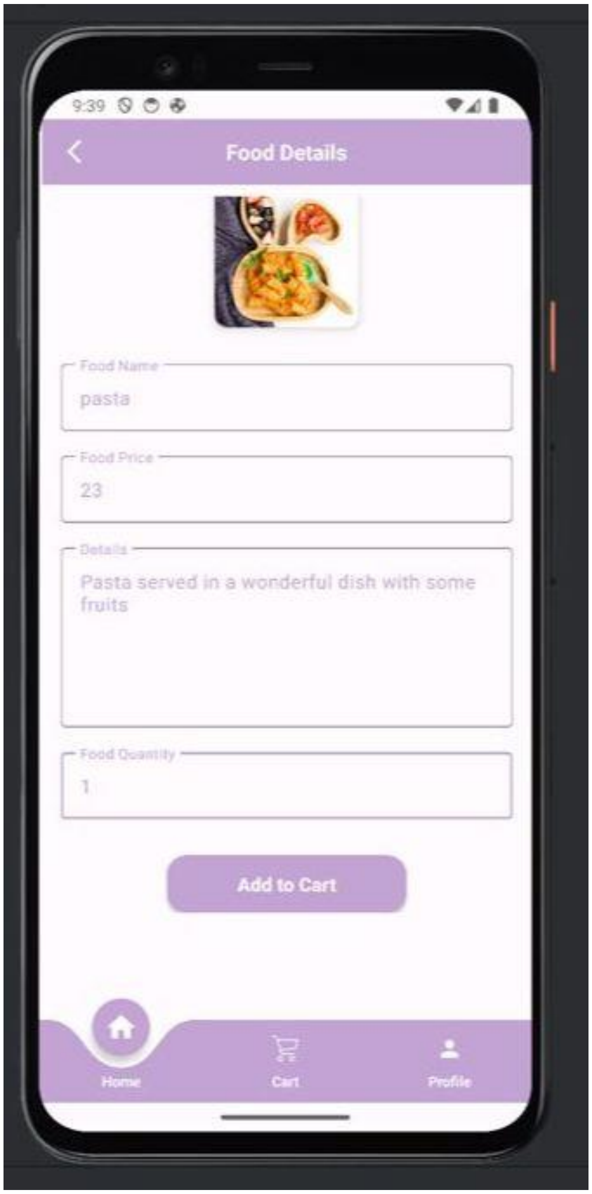
SAR 10

pasta

SAR 23

Figure 52 snack section

If the parent wants to order food for his child, he chooses which section he wants, then clicks on the dish he wants and specifies the quantity, and then adds it to the cart. For example, choose the snacks section, choose the dish is pasta, specify the quantity he wants, then press the add to cart button.



Food Details

Food Name: pasta

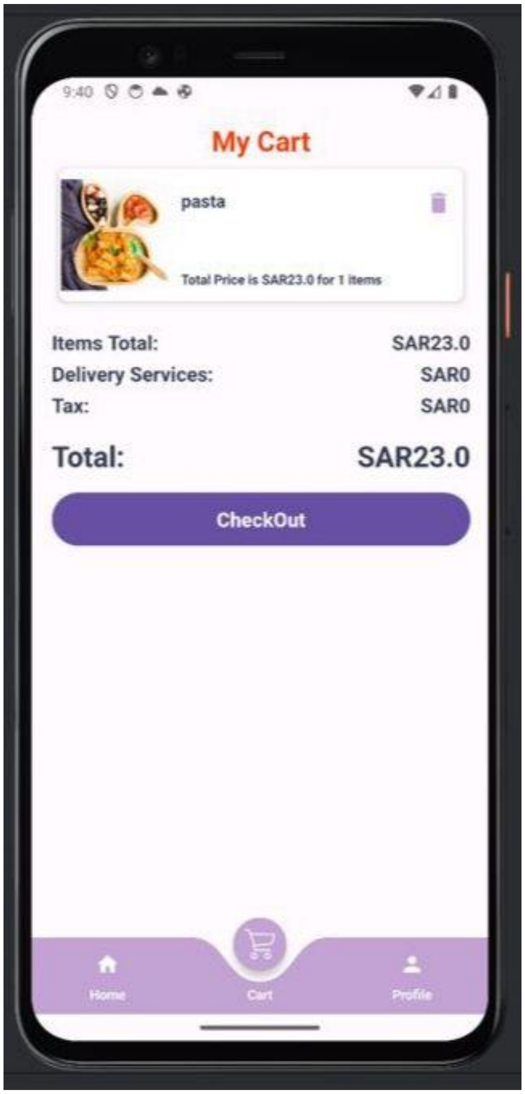
Food Price: 23

Details: Pasta served in a wonderful dish with some fruits

Food Quantity: 1

Figure 53 Food Details

Then go to the cart page to check out



My Cart

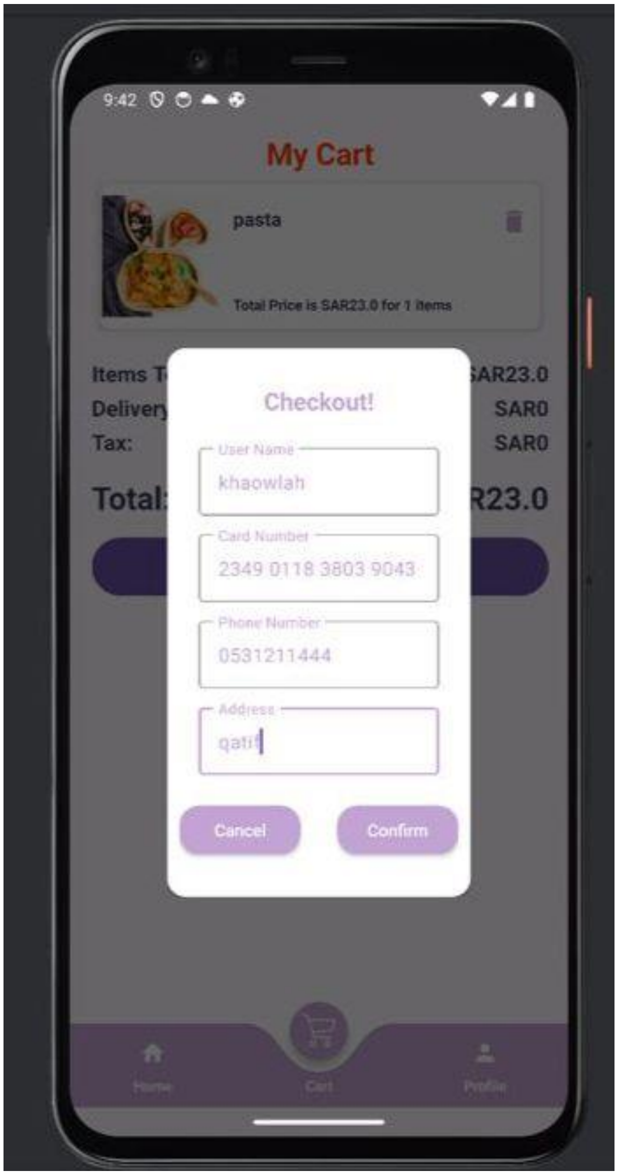
pasta

Total Price is SAR23.0 for 1 items

Items Total:	SAR23.0
Delivery Services:	SAR0
Tax:	SAR0
Total:	SAR23.0

Figure 54 My Cart

When you click on the check-out button, this page will appear to fill in the card information for payment, the address for food delivery, and the mobile number to communicate when the food arrives.



My Cart

pasta

Total Price is SAR23.0 for 1 items

Items T… SAR23.0
Delivery SAR0
Tax: SAR0

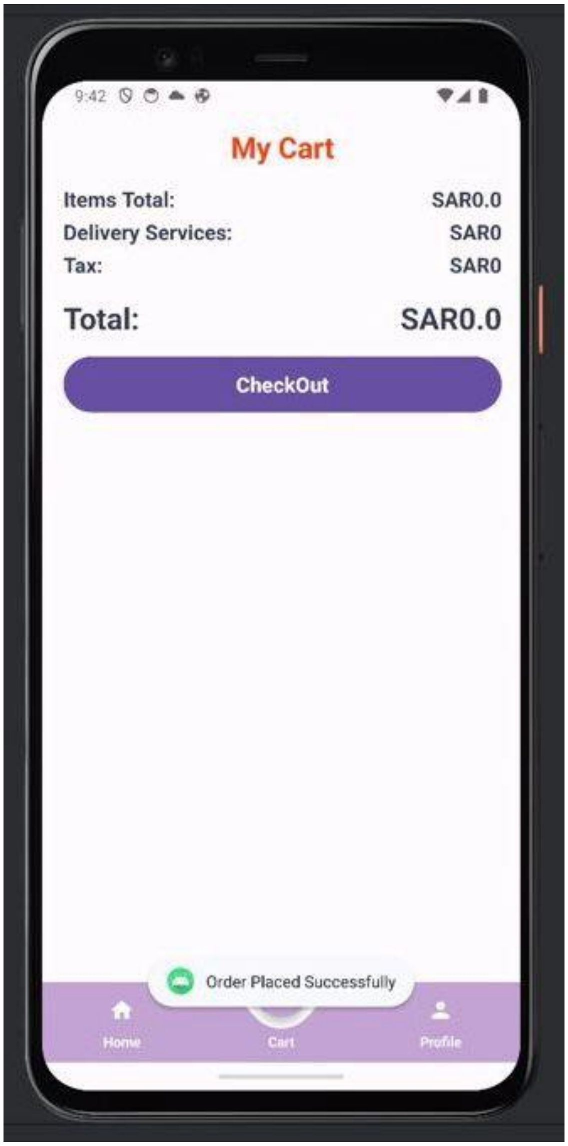
Total: R23.0

Checkout!

User Name: khaowlah
Card Number: 2349 0118 3803 9043
Phone Number: 0531211444
Address: qatif

Figure 55 Checkout

After pressing the confirm button, a message will appear stating “order place successfully”



9:42

My Cart

Items Total: SAR0.0

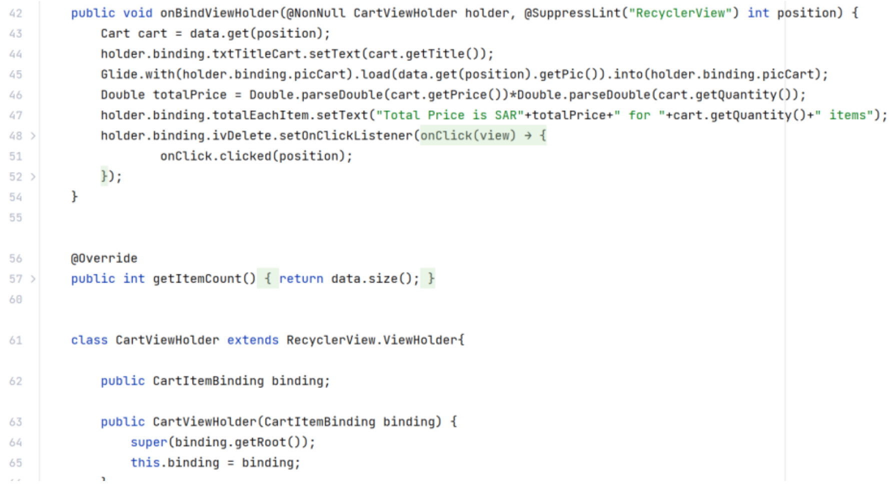
Delivery Services: SAR0

Tax: SAR0

Total: SAR0.0

Order Placed Successfully

Figure 56 My Cart



```
public void onBindViewHolder(@NonNull CartViewHolder holder, @SuppressLint("RecyclerView") int position) {
    Cart cart = data.get(position);
    holder.binding.txtTitleCart.setText(cart.getTitle());
    Glide.with(holder.binding.picCart).load(data.get(position).getPic()).into(holder.binding.picCart);
    Double totalPrice = Double.parseDouble(cart.getPrice())*Double.parseDouble(cart.getQuantity());
    holder.binding.totalEachItem.setText("Total Price is SAR"+totalPrice+" for "+cart.getQuantity()+" items");
    holder.binding.ivDelete.setOnClickListener(onClick(view) → {
            onClick.clicked(position);
    });
}


@Override
public int getItemCount() { return data.size(); }


class CartViewHolder extends RecyclerView.ViewHolder{

    public CartItemBinding binding;

    public CartViewHolder(CartItemBinding binding) {
        super(binding.getRoot());
        this.binding = binding;
```

Here is the code for the previous page where it shows the cart and what it contains and return the number of items in it

15.9 Activities page for children

When you click on the children’s activities section, the main page for children’s activities will appear, which contains the following sections: addition, subtraction, multiplication, and division.

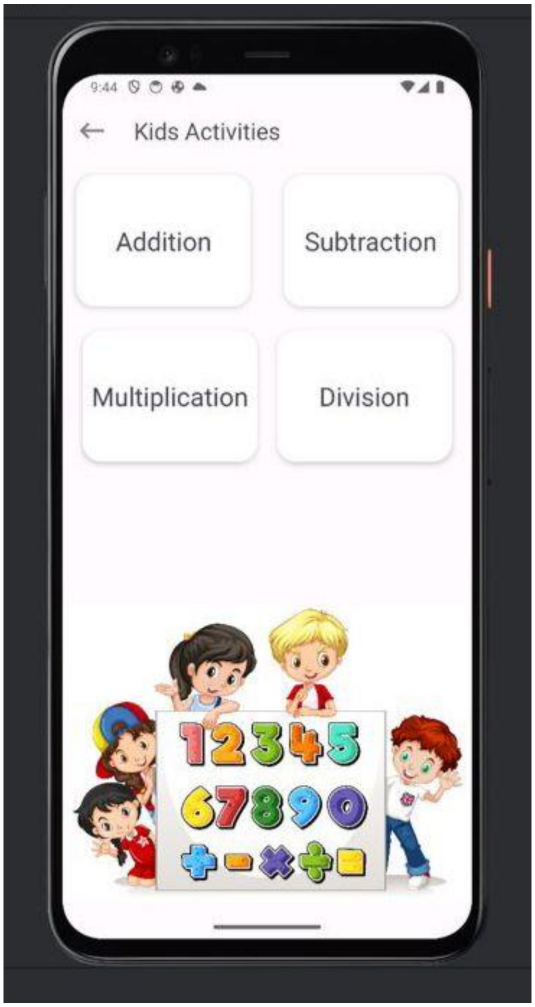


Figure 57Activities page for children1

When you choose any section, whether addition, subtraction, multiplication, or division, classifications will appear according to the age group

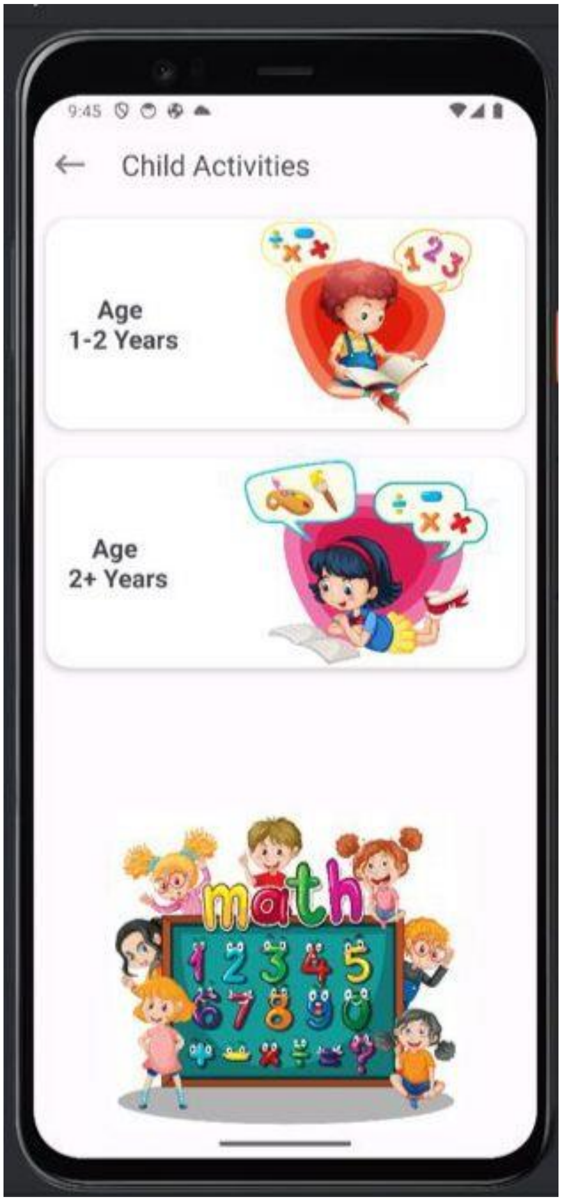


Figure 58 Activities page for children2

The most important codes used in this part are:

```
package com.example.application.KidsInterface;

import android.os.Bundle;
import android.view.View;

import androidx.appcompat.app.AppCompatActivity;

import com.example.application.Utils.YoutubeVideoUtils;
import com.example.application.databinding.ActivityYoutubePlayerBinding;

public class Multiplication1to2Years extends AppCompatActivity {
    ActivityYoutubePlayerBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding=ActivityYoutubePlayerBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
    init();
    }
    private void init(){
        binding.ivBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                onBackPressed();
            }
        });
        new
YoutubeVideoUtils().embedYouTubeVideo(binding.youtubeWebView,"oPINS561Des?si=V6nmN
UmVDPPs4jci");

    }
}
```

When you choose the age group, the appropriate video for the child will appear

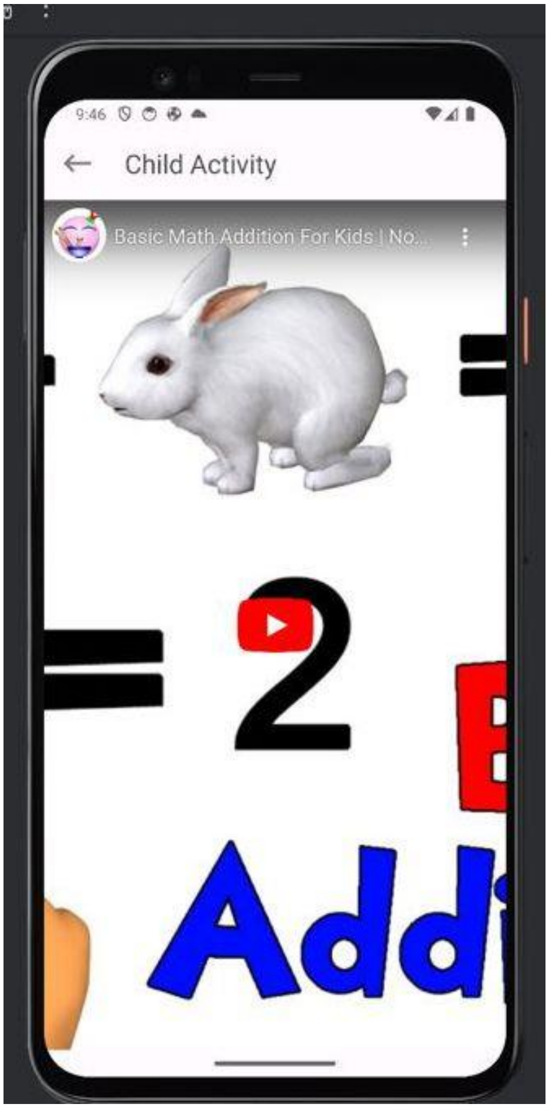
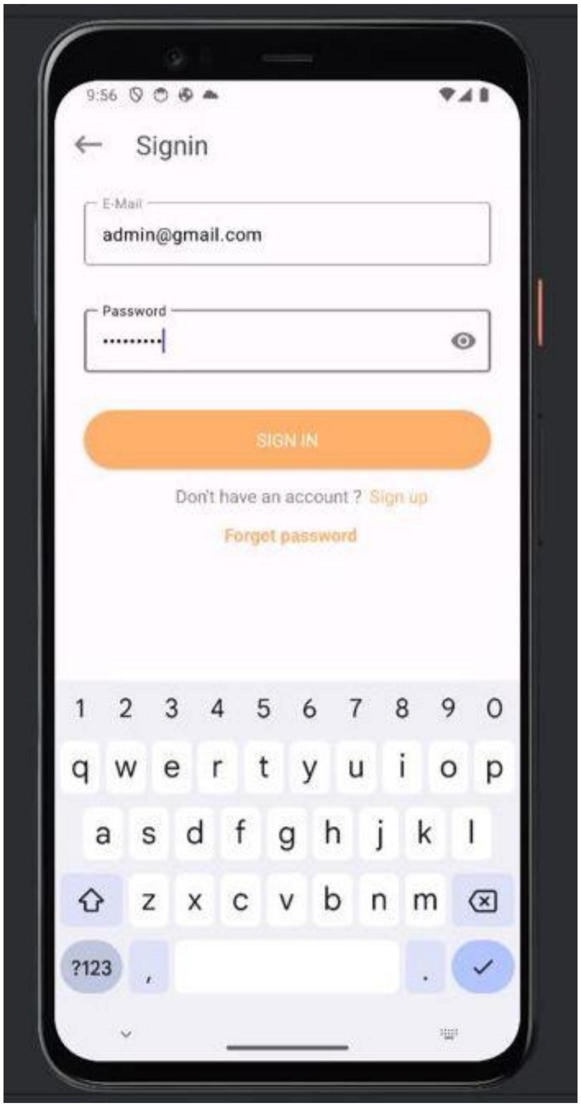


Figure 59 Activities page for children3

15.10Admin page to manage food orders

Log in as admin to manage the food page request, noting that the email and password for the admin to manage the food request is as follows:
sign in email: admin@gmail.com and password: admin@123



Signin

E-Mail: admin@gmail.com

Password: ·········

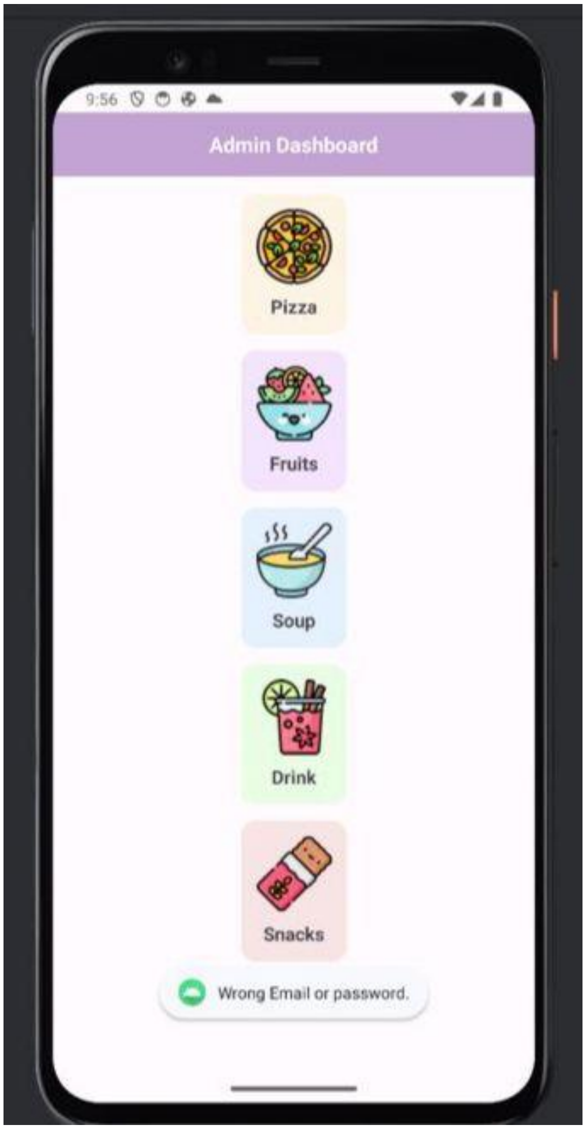
Figure 60 Sing-in admin foods

After logging in as a manager, the manager's home page will appear to manage the food order

The most important codes used in this part are:

```
package com.example.application.orderfood.admin;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;


import com.example.application.orderfood.adapter.CategoryAdapter;
import com.example.application.orderfood.model.Category;
import com.example.application.orderfood.model.OnClick;
import com.example.application.R;
import com.example.application.databinding.ActivityAdminMainBinding;

import java.util.ArrayList;

public class AdminMainActivity extends AppCompatActivity {
    ActivityAdminMainBinding binding;

    @SuppressLint("SetTextI18n")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityAdminMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        binding.rlTop.tvLabel.setText("Admin Dashboard");

        ArrayList<Category> arrayList = new ArrayList();
        arrayList.add(new Category("Pizza", R.drawable.pizzam));
        arrayList.add(new Category("Fruits", R.drawable.fruit));
        arrayList.add(new Category("Soup", R.drawable.soup));
        arrayList.add(new Category("Drink", R.drawable.cocktail));
        arrayList.add(new Category("Snacks", R.drawable.snacks));
        CategoryAdapter adapter = new CategoryAdapter(this, arrayList, new
OnClick() {
            @Override
            public void clicked(int pos) {
                Category category = arrayList.get(pos);
                Intent intent = new Intent(AdminMainActivity.this,
ProductsActivity.class);
                intent.putExtra("name", category.getTitle());
                startActivity(intent);
            }
        });
        binding.rvCategories.setLayoutManager(new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false));
        binding.rvCategories.setAdapter(adapter);

    }
}
```

He can choose any section he wants, add a dish, update a dish, or delete a dish



Wrong Email or password.

Figure 61 Admin Dashboard

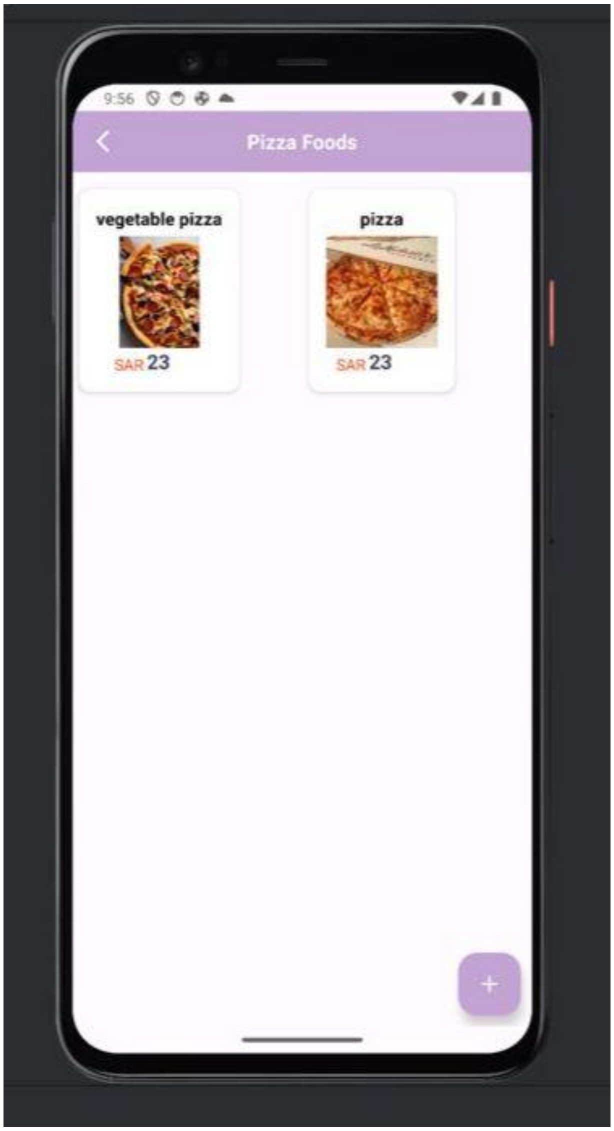


Figure 62 Pizza Foods

The most important codes used in this part are:

```
binding.ivMachine.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_OPEN_DOCUMENT);
        startActivityForResult(Intent.createChooser(intent, "Select Machine
Image"), 123);
    }
}
```

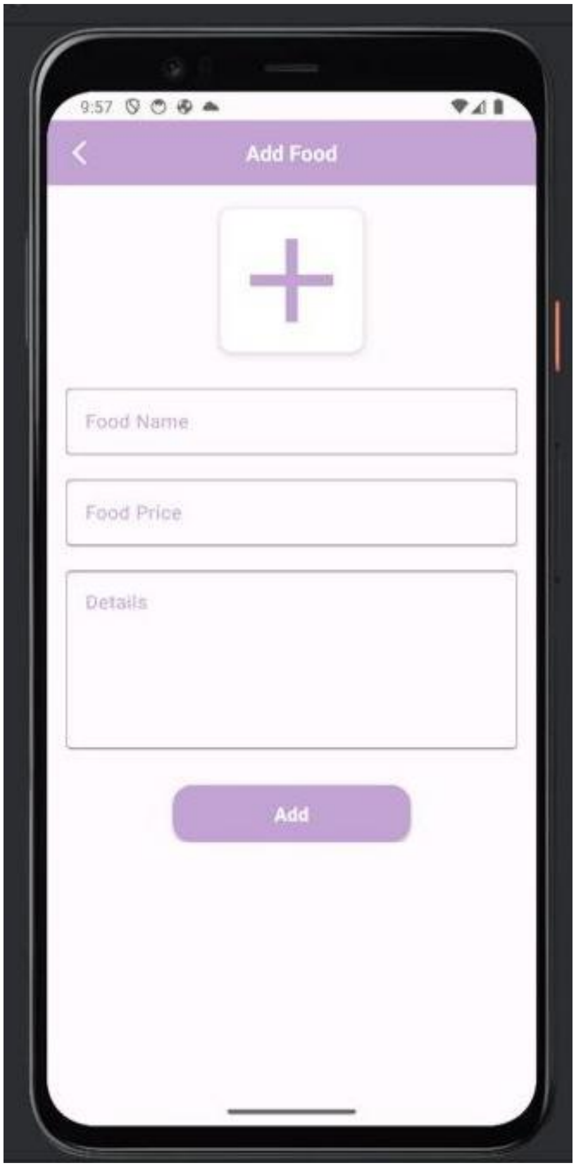
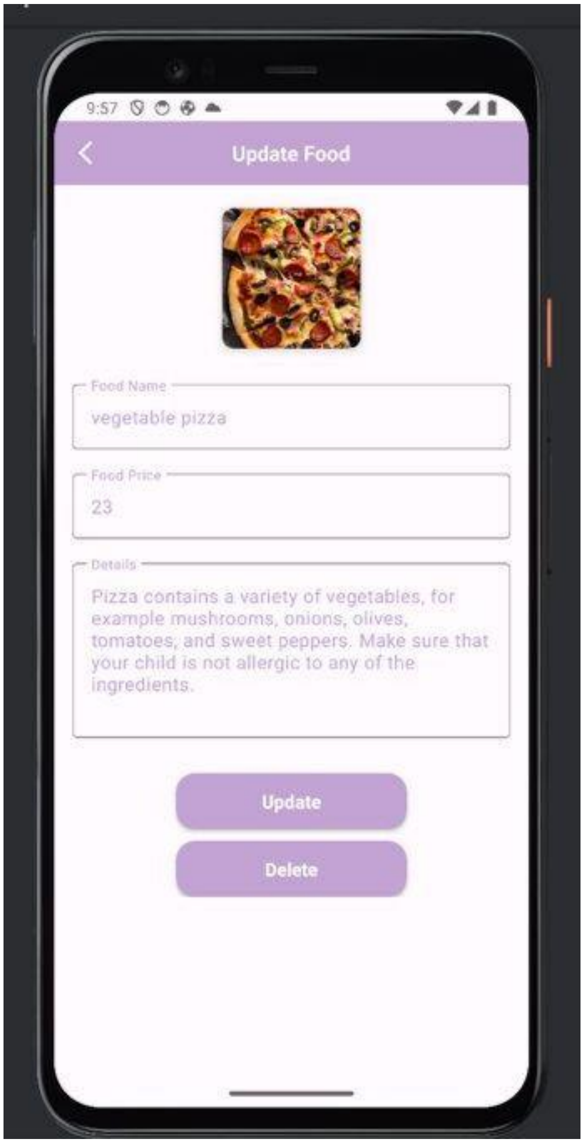


Figure 63Add Food

The most important codes used in this part are:

```
if (foodModel != null){
    binding.rlTop.tvLabel.setText("Update Food");
}else{
    binding.rlTop.tvLabel.setText("Add Food");
}
```



Food Name
vegetable pizza

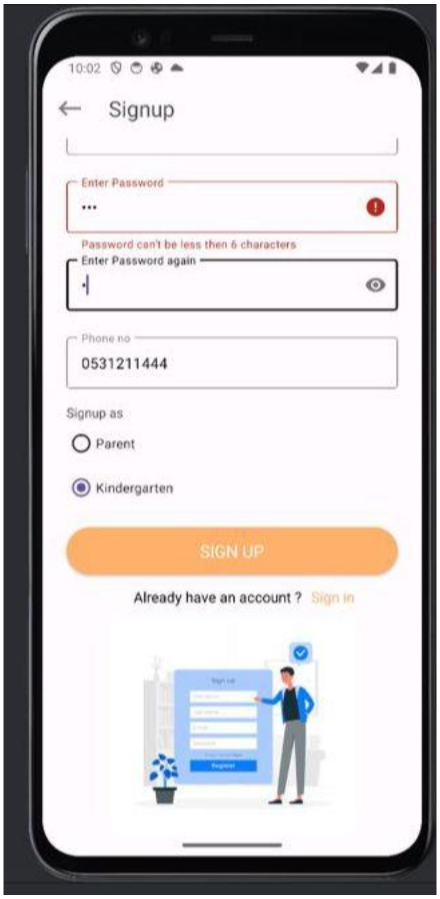
Food Price
23

Details
Pizza contains a variety of vegetables, for example mushrooms, onions, olives, tomatoes, and sweet peppers. Make sure that your child is not allergic to any of the ingredients.

Figure 64 Update Food and delete

15.11The manager's page for the nanny

If you want to care for children, you can create an account as a nanny



10:02

Signup

Enter Password
•••

Password can't be less then 6 characters

Enter Password again
•

Phone no
0531211444

Signup as

○ Parent

● Kindergarten

Already have an account ? Sign in

After creating an account, nanny's home page will appear

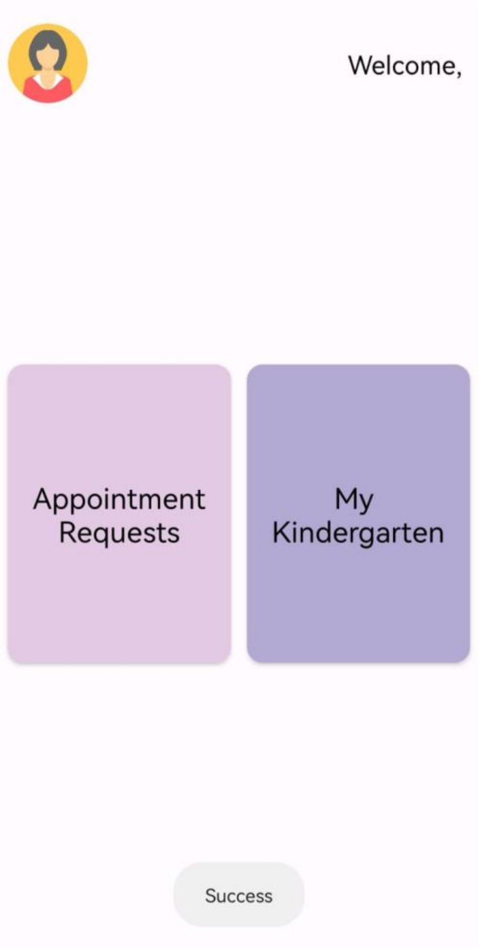
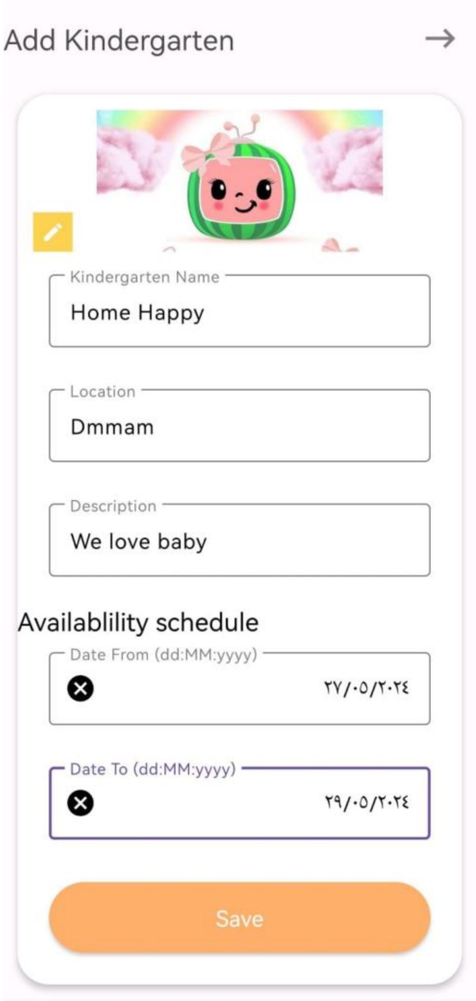

Welcome,

Figure 65 Home page for nanny

When you click on my kindergarten, the nanny can add her information, a description, her area, the date she is available, and a special picture of her so that parents can reserve her, and she can modify the information whenever she wants.



Add Kindergarten

Kindergarten Name: Home Happy

Location: Dmmam

Description: We love baby

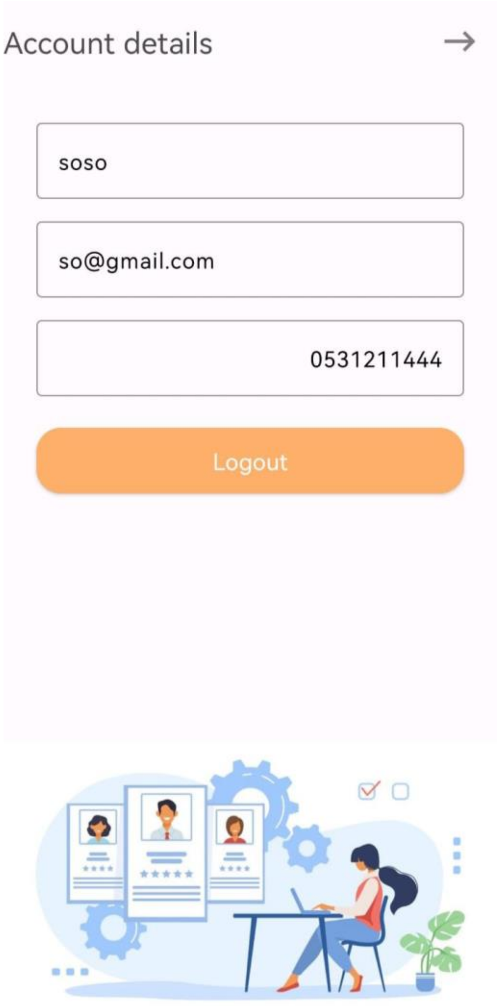
Availablility schedule

Date From (dd:MM:yyyy): ٢٧/٠٥/٢٠٢٤

Date To (dd:MM:yyyy): ٢٩/٠٥/٢٠٢٤

Figure 66 Add Kindergarten

Here is the nanny's profile page



Account details →

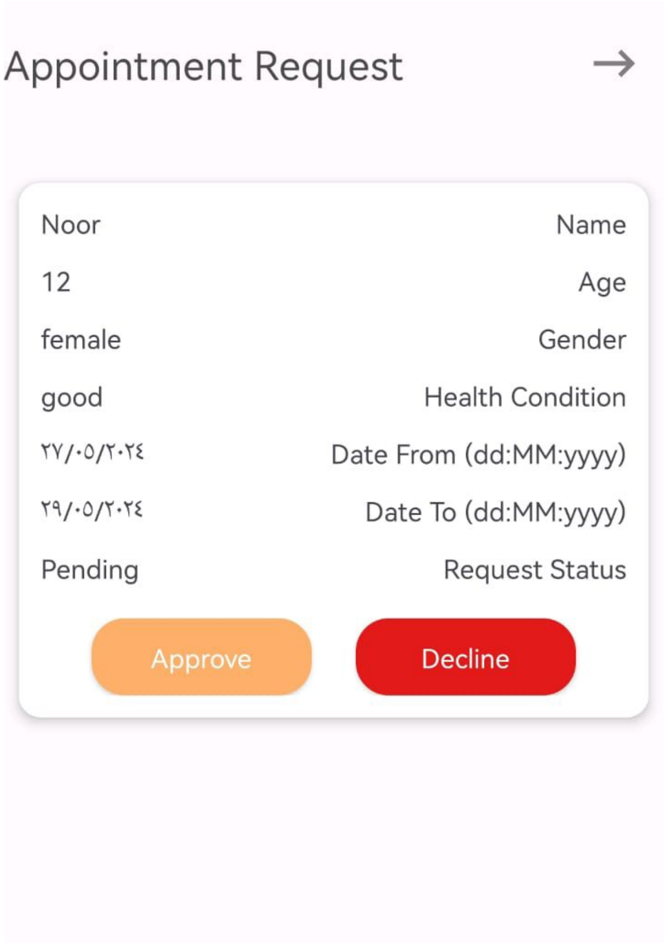
soso

so@gmail.com

0531211444

Figure 67 Account details for Nanny

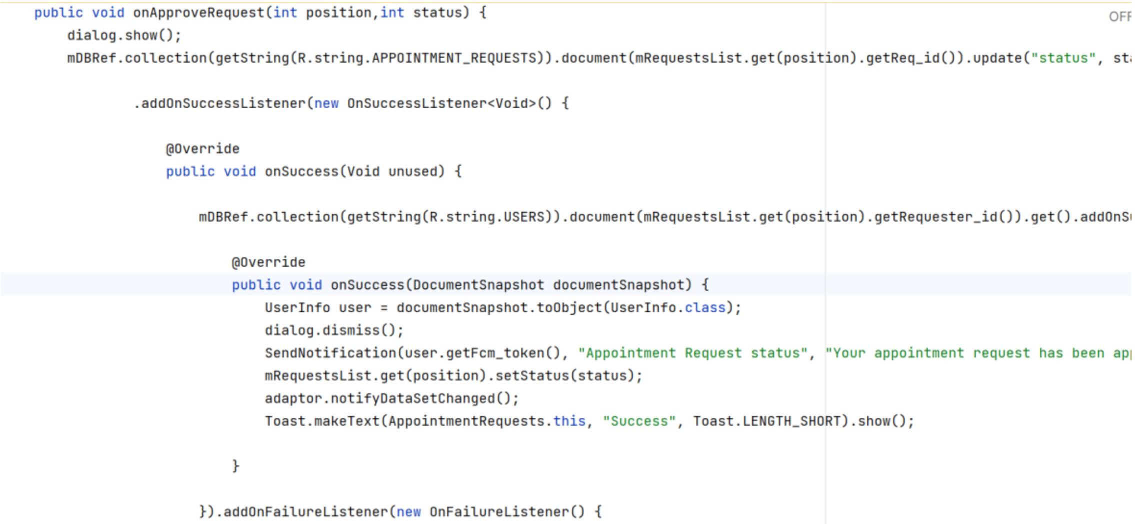
In the appointment request section, the nanny can accept or reject the reservation if one of the parents requests her to care for her child



Appointment Request

Noor	Name
12	Age
female	Gender
good	Health Condition
٢٧/٠٥/٢٠٢٤	Date From (dd:MM:yyyy)
٢٩/٠٥/٢٠٢٤	Date To (dd:MM:yyyy)
Pending	Request Status

Figure 68 Appointment Request



```
public void onApproveRequest(int position,int status) {
    dialog.show();
    mDBRef.collection(getString(R.string.APPOINTMENT_REQUESTS)).document(mRequestsList.get(position).getReq_id()).update("status", st

            .addOnSuccessListener(new OnSuccessListener<Void>() {

                @Override
                public void onSuccess(Void unused) {

                    mDBRef.collection(getString(R.string.USERS)).document(mRequestsList.get(position).getRequester_id()).get().addOnS

                        @Override
                        public void onSuccess(DocumentSnapshot documentSnapshot) {
                            UserInfo user = documentSnapshot.toObject(UserInfo.class);
                            dialog.dismiss();
                            SendNotification(user.getFcm_token(), "Appointment Request status", "Your appointment request has been ap
                            mRequestsList.get(position).setStatus(status);
                            adaptor.notifyDataSetChanged();
                            Toast.makeText(AppointmentRequests.this, "Success", Toast.LENGTH_SHORT).show();

                        }

                    }).addOnFailureListener(new OnFailureListener() {
```

Here is the code to approve the request of the appointment when the approve button is clicked

16.Tasks of team members

Name	Task
Khaowlah Zakariya	• **project proposal:** - Introduction of the Project - Description of the idea • **Prototype** - Welcome page - Account creation page - Login page - Forget password - Activities page for children • **Report** - Mobile App Flowchart: Include a flowchart that details the logical flow of the app, including decision points and user interactions - Arranging the file and modifying some paragraphs • **Implement the code** - Add food for admin - Create an account - Sign in - The sections inside the food - Checkout page - My reservation for user
Fatimah ALharz	• **project proposal:** - Problem Statement - Objectives • **Prototype** -user page profile - The user's home page after logging in -child care - Appointment detail - Food order home page for the user • **Report** - Mobile App User Flow Diagram: Include a diagram that shows the flow of the user through the app, from start to finish: • **Implement the code** - Forgot password page - Nanny reservation's view page

	- Profile page for user - Activities page for children(Addition, Subtraction, Multiplication, Division) - my cart for order foods - update and delete foods
Lama Abdulaziz	• **project proposal:** - Summary of Project Deliverables - Project Success Criteria • **Prototype** - The nanny's home page - Appointment requests - My kindergarten - Nanny reservation's view page for the manager - The sections inside the food • **Report** - Core Features: Outline the key functionalities of your mobile app: - A New Concept was Implemented in the Project • **Implement the code:** - Child care page - Appointment detail page - The nanny home page - Food order home page for the user - Welcome page - Activities home page for children
Sara Al-Salahi	• **project proposal:** - Solution Methodology - Logo for the project • **Prototype** - Food Detalis - my cart - checkout - Food order page for the manager - Add food page and update and delete • **Report** - Comprehensive List of Application Requirements: Provide a detailed list of requirements for the

	application, covering functional, non functional, and technical aspects: • **Implement the code:** - Requests to appoint a nanny - My Kindergarten page - Home page for user - Food order home page for the manager - Profile page for nannay - Activities page for children(Age 1-2 Years, Age 2+ Years)

table 1 Tasks of team members

Note: In writing the description of the interfaces and the prototype, each student wrote about the interface she created in the assignment table.
Note: The database we created together.

References

[1] “تنزيل ‘استوديو Android’ وأدوات التطبيقات - مطوّرو تطبيقات Android | Android Studio | Android Developers.” Accessed: Feb. 02, 2024. [Online]. Available: https://developer.android.com/studio?hl=ar

[2] “Introduction to Java.” Accessed: Feb. 02, 2024. [Online]. Available: https://www.w3schools.com/java/java_intro.asp