

# PROJECT DESCRIPTION

## Table of Contents

I.	APPLICATION INTERNAL STRUCTURE.....	3
II.	WELCOME & REGISTERATION .....	4
	➤ Test Cases .....	4
	➤ Code .....	7
	➤ Room Database User Added.....	12
	➤ Firebase Authentication User Added.....	12
III.	LOGIN .....	12
	➤ Test Cases .....	12
	➤ Code .....	15
III.	RESTAURANTS.....	18
	➤ Test Cases .....	18
IV.	DISHES .....	23
	➤ Test Cases .....	23
	➤ Code .....	25
V.	FAVOURITES .....	29
	➤ Test Cases .....	29
	Code .....	31
1.	DISH DETAIL .....	34
	➤ Test Cases .....	34
VII.	CART.....	38
	➤ Test Cases .....	38
	➤ Code .....	41
VIII.	CONFIRM ORDER.....	45
	➤ Test Cases .....	45
	➤ Code .....	50
	➤ Firebase .....	56
IX.	ORDER HISTORY & ORDER HISTORY CONTENT.....	57
	Test Cases .....	57
	➤ Code .....	58
X.	ORDER STATUS.....	64
	Test Cases .....	64
	Code .....	66

XI. PROFILE .....	68
Test Cases .....	68
Code .....	70
XII. UPDATE PROFILE.....	72
Test Cases .....	72
➤ Room Database User Updated.....	73
➤ Firebase Authentication User Updated .....	73
Code .....	74
XIII. EXTRA JAVA CLASSES.....	76
I. WELCOME PAGE.....	79
Test Cases .....	79
Code .....	80
II. REGISTERATION.....	81
Test Cases .....	81
Code .....	82
III. LOGIN .....	84
Test Cases .....	84
Code .....	85
IV. HOMEPAGE.....	87
Test Cases .....	87
Code .....	87
V. ORDER CONFIRMATION .....	88
Test Cases .....	88
Code .....	90
VI. ORDER PROCESSING.....	94
Test Cases .....	94
Code .....	95
VII. ORDER DELIVERY .....	97
Test Cases .....	97
Code .....	97

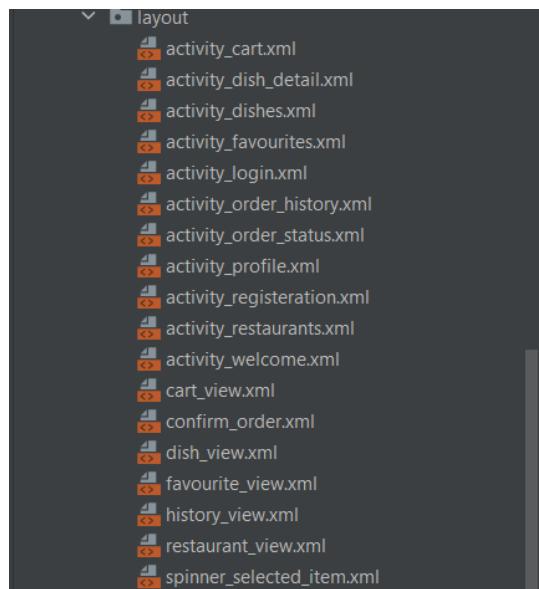
# WAGBA APPLICATION

Important note – to run code in your system, make sure to create firebase project, import the JSON file added in the project folder, then connect all activities with that database

## I. APPLICATION INTERNAL STRUCTURE

The application will contain the following layouts:

- Welcome
- Login
- Registration
- Restaurants list includes recycler view
- Restaurant view that shows the recycler view
- Dishes list includes recycler view
- Dish view that shows the recycler view
- Dish details
- Cart with recycler view
- Cart view that shows the recycler view
- Order layout that tracks the status of current order
- Order history includes a recycler view
- Order history view that shows the recycler view
- Profile
- Favorite dishes that include a recycler view
- Favorite dishes view that shows the recycler view
- Spinner selected items xml for spinner components used



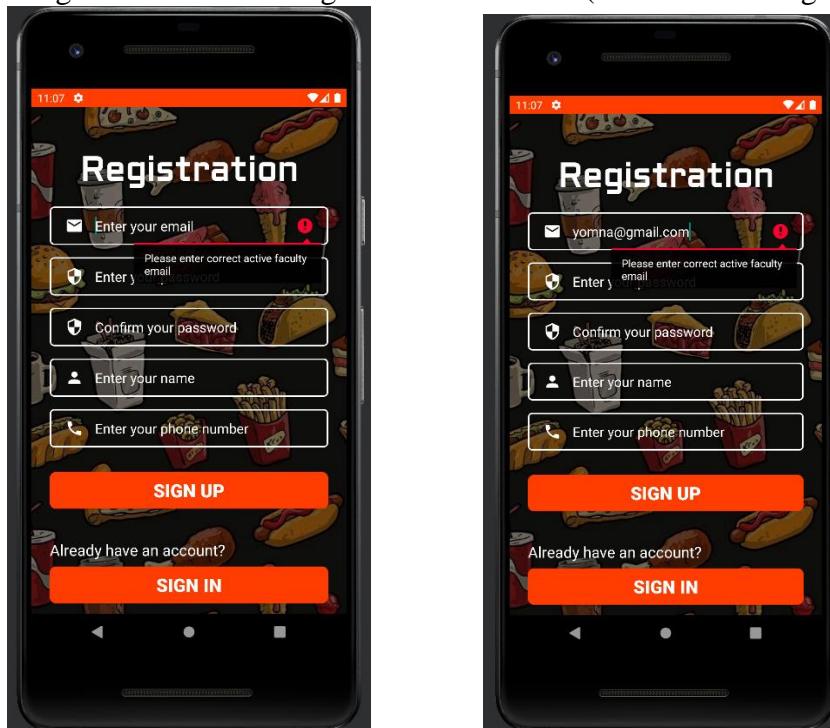
## II. WELCOME & REGISTRATION

### ➤ Test Cases

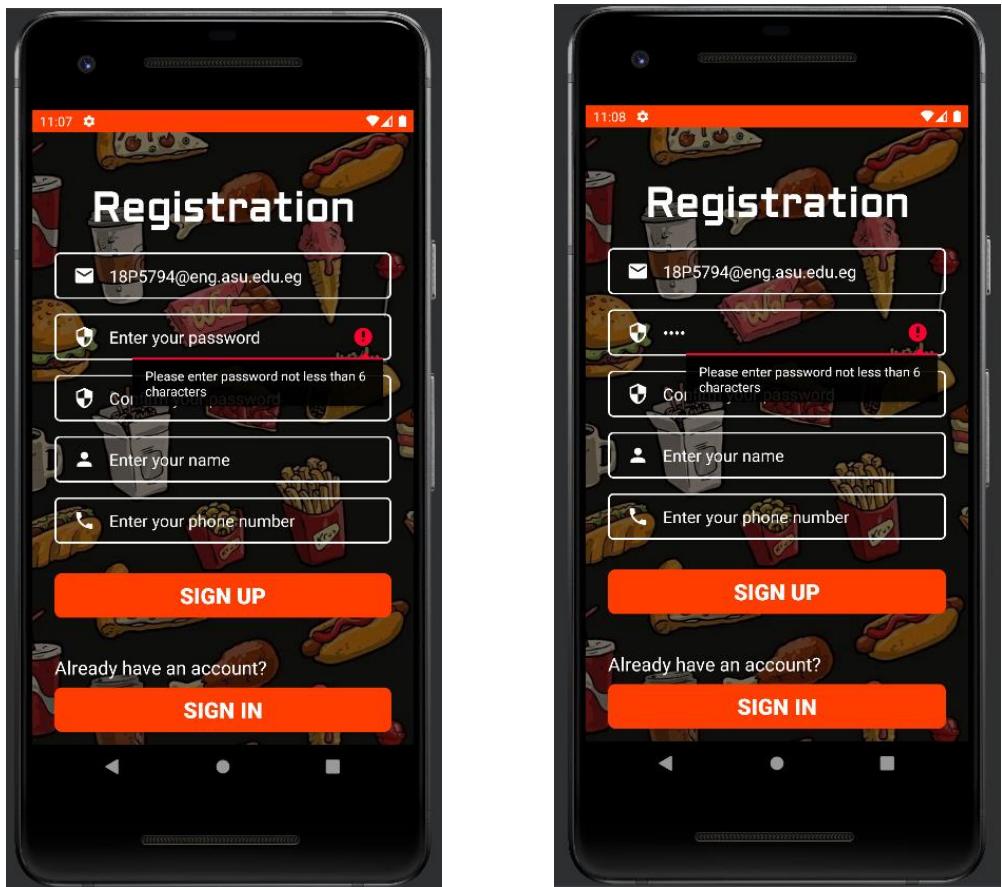
1. The activity without clicking any buttons or entering any values



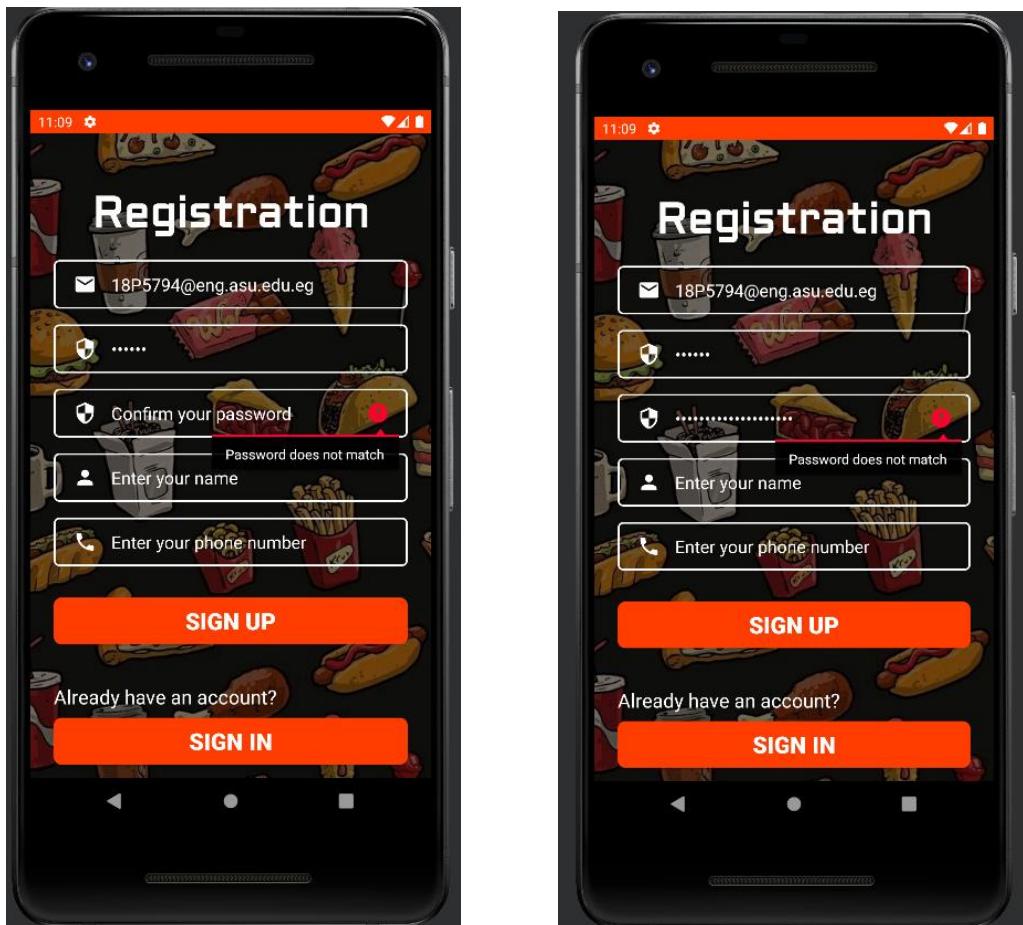
2. Not entering an email or entering an incorrect email (must be an @eng.asu.edu.eg email)



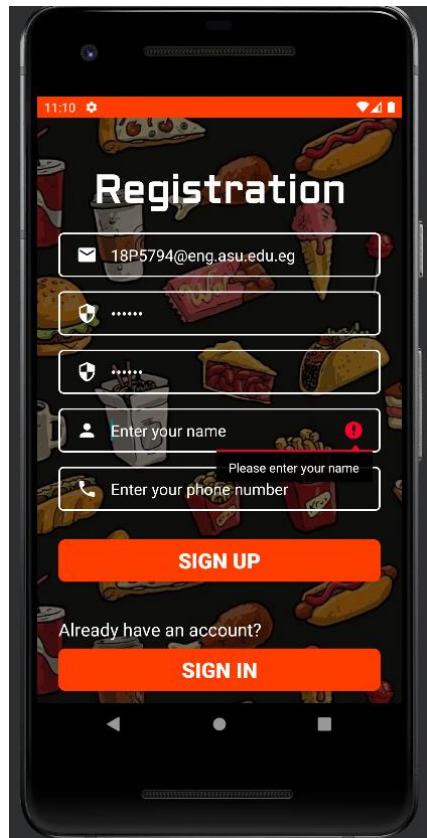
3. Not entering a password or entering password less than 6 characters



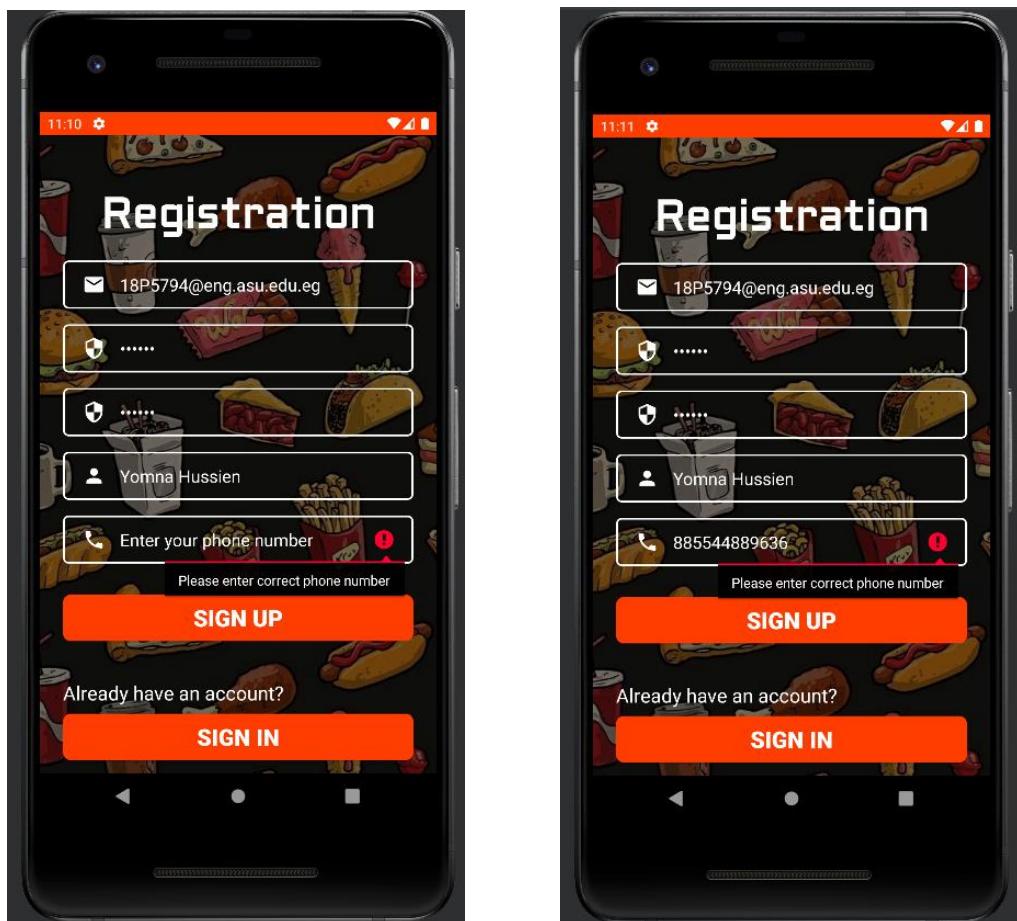
4. Not entering confirm password or entering confirmation password that doesn't match the original.



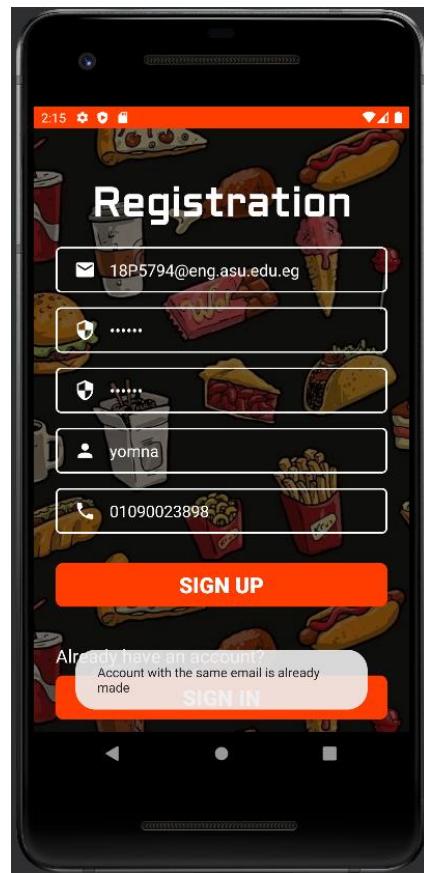
5. Not entering a username



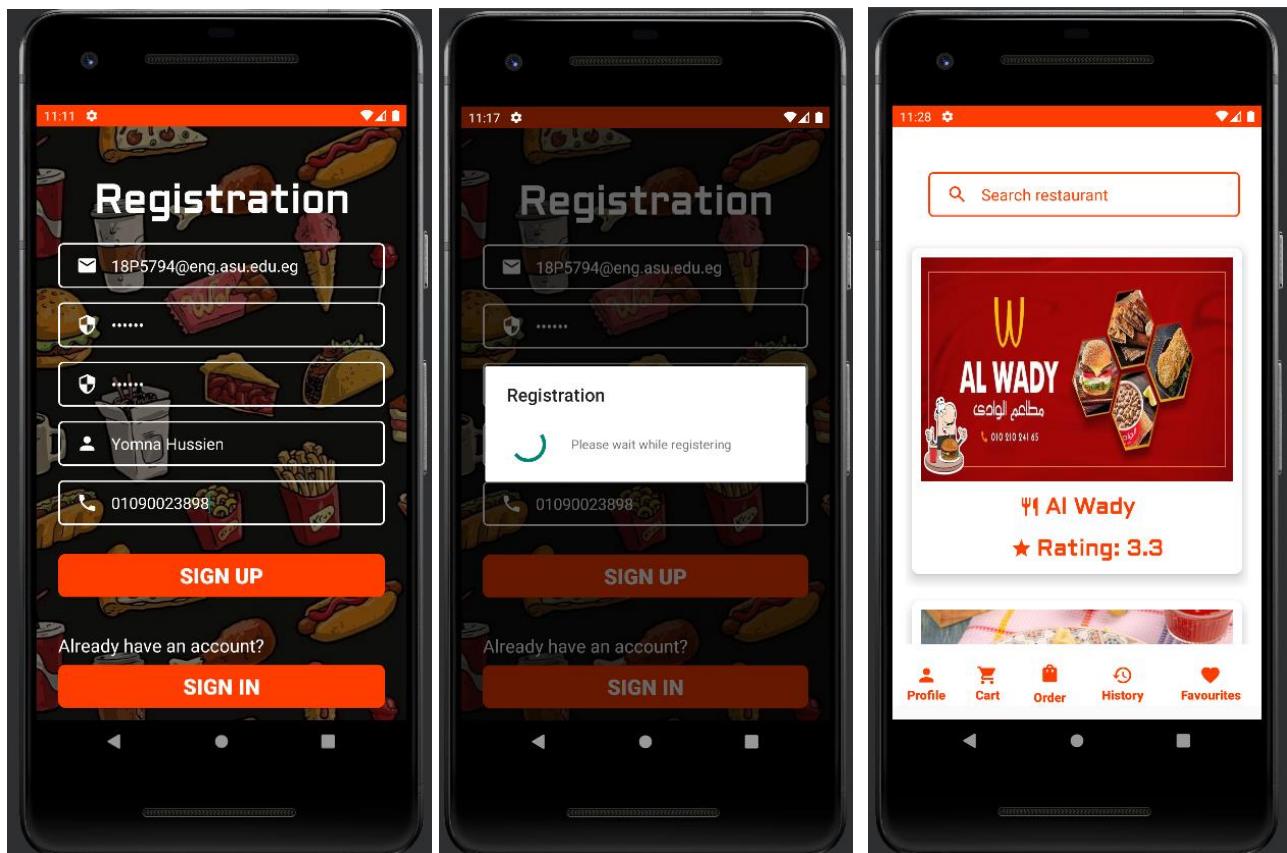
6. Not entering a phone number or incorrect phone number



7. Entering same email that already used (the email already exists in the firebase)



8. Entering everything correct and clicking on Sign Up button



➤ Code

## Welcome Activity

```
package com.example.wagbaapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

import io.paperdb.Paper;

public class Welcome extends AppCompatActivity {

    Button start;
    ProgressDialog progressDialog;
    FirebaseAuth myAuth;
    FirebaseUser myUser;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);

        start = (Button) findViewById(R.id.getStartedButton);
        progressDialog = new ProgressDialog(this);
        myAuth = FirebaseAuth.getInstance();
        myUser = myAuth.getCurrentUser();

        Paper.init(this);

        start.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(Welcome.this, Login.class);
                startActivity(intent);
            }
        });
    }

    String emailKey = Paper.book().read(Prevalent.emailKey);
    String passwordKey = Paper.book().read(Prevalent.passwordKey);

    if(emailKey != "" && passwordKey != "") {
        if(!TextUtils.isEmpty(emailKey) && !TextUtils.isEmpty(passwordKey)) {
            AllowAccess(emailKey, passwordKey);
        }
    }
}

private void AllowAccess(String emailKey, String passwordKey) {
```

```

        progressDialog.setMessage("Please wait while logging");
        progressDialog.setTitle("Already logged in");
        progressDialog.setCanceledOnTouchOutside(false);
        progressDialog.show();

myAuth.signInWithEmailAndPassword(emailKey, passwordKey).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful()){
            progressDialog.dismiss();
            sendToNewActivity();
        }
    }
});

private void sendToNewActivity() {
    Intent intent02 = new Intent(this, Restaurants.class);

intent02.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK|Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent02);
    finish();
}

}

```

## Registration Activity

```

package com.example.wagbaapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.HashMap;

public class Registration extends AppCompatActivity {

    EditText registerEmail, registerPassword, confirmPassword, registerUserName,
registerPhoneNumber;
    Button signUp, signIn;

```

```

//        //normal email pattern
//        String emailPattern = "[a-zA-Z0-9._-]+@[a-z]+\.\+[a-z]+";
//
//        //eng.asu.edu.eg email pattern
String emailPattern = "(\S*)@eng\.asu\.edu\.eg";
String phoneNumberPattern = "^01[0125][0-9]{8}$";
ProgressDialog progressDialog;
FirebaseAuth myAuth;
FirebaseUser myUser;
//        DatabaseReference databaseReference;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_registration);

    registerEmail = (EditText) findViewById(R.id.editTextEmail2);
    registerPassword = (EditText) findViewById(R.id.editTextPassword2);
    confirmPassword = (EditText) findViewById(R.id.editTextConfirmPass);
    registerUserName = (EditText) findViewById(R.id.editTextUserName);
    registerPhoneNumber = (EditText) findViewById(R.id.editTextPhoneNumber);
    signUp = (Button) findViewById(R.id.signUpButton2);
    signIn = (Button) findViewById(R.id.signInButton2);
    progressDialog = new ProgressDialog(this);
    myAuth = FirebaseAuth.getInstance();
    myUser = myAuth.getCurrentUser();
//        databaseReference =
FirebaseDatabase.getInstance().getReference().child("Users");

    Intent intent01 = new Intent(this, Login.class);

    signUp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            register();
        }
    });

    signIn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(intent01);
            finish();
        }
    });
}

private void register() {
    String inputEmail = registerEmail.getText().toString();
    String inputPass = registerPassword.getText().toString();
    String confirmPass = confirmPassword.getText().toString();
    String inputUserName = registerUserName.getText().toString();
    String inputPhoneNumber = registerPhoneNumber.getText().toString();

    if(!inputEmail.matches(emailPattern)){
        registerEmail.setError("Please enter correct active faculty email");
        registerEmail.requestFocus();
    }else if(inputPass.isEmpty() || inputPass.length()<6){
        registerPassword.setError("Please enter password not less than 6
characters");
        registerPassword.requestFocus();
    }
}

```

```

}else if(!inputPass.equals(confirmPass)){
    confirmPassword.setError("Password does not match");
    confirmPassword.requestFocus();
}else if(inputUserName.isEmpty()){
    registerUserName.setError("Please enter your name");
    registerUserName.requestFocus();
}else if(!inputPhoneNumber.matches(phoneNumberPattern)){
    registerPhoneNumber.setError("Please enter correct phone number");
    registerPhoneNumber.requestFocus();
}else{
    progressDialog.setMessage("Please wait while registering");
    progressDialog.setTitle("Registration");
    progressDialog.setCanceledOnTouchOutside(false);
    progressDialog.show();

myAuth.createUserWithEmailAndPassword(inputEmail, inputPass).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful()){
            progressDialog.dismiss();

        ///////////////////////////////////////////////////
        ///////////////////////////////////////////////////
        ///////////////////////////////////////////////////
        ///////////////////////////////////////////////////
        databaseReference.child(myUser.getUid()).updateChildren(userMap);

        //Adding to user info to Room database
        // 1) creating new user
        User user = new User(myUser.getUid(), inputUserName,
inputEmail, inputPhoneNumber);

        // 2) Initializing the database
        FirebaseDatabase userDatabase =
UserDatabase.getUserDatabase(getApplicationContext());

        // 3) Initializing the Dao
        final UserDao userDao = userDatabase.userDao();

        // 4) Inserting to database in a new thread so we don't block
the UI
        new Thread(new Runnable() {
            @Override
            public void run(){
                userDao.insert(user);
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        Toast.makeText(Registeration.this, "You have
registered successfully", Toast.LENGTH_SHORT).show();
                    }
                });
            }
        }).start();

        sendToNewActivity();
    }else{
        progressDialog.dismiss();
        Toast.makeText(Registeration.this, "Account with the same

```

```

        email is already made", Toast.LENGTH_SHORT).show();
    }

}

}

private void sendToNewActivity() {
    Intent intent02 = new Intent(this, Restaurants.class);

    intent02.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK|Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent02);
    finish();
}
}

```

## ➤ Room Database User Added

	user_id	user_name	user_email	user_phone
1	803j4dPv8Zfjuig8oLauECd3ANq1	test name	test123@eng.asu.edu.eg	01090056898
2	e31VogWsqaZde6uTsfA9kZXeJKO2	Yomna Hussien Mohamed	18P5794@eng.asu.edu.eg	01090023898

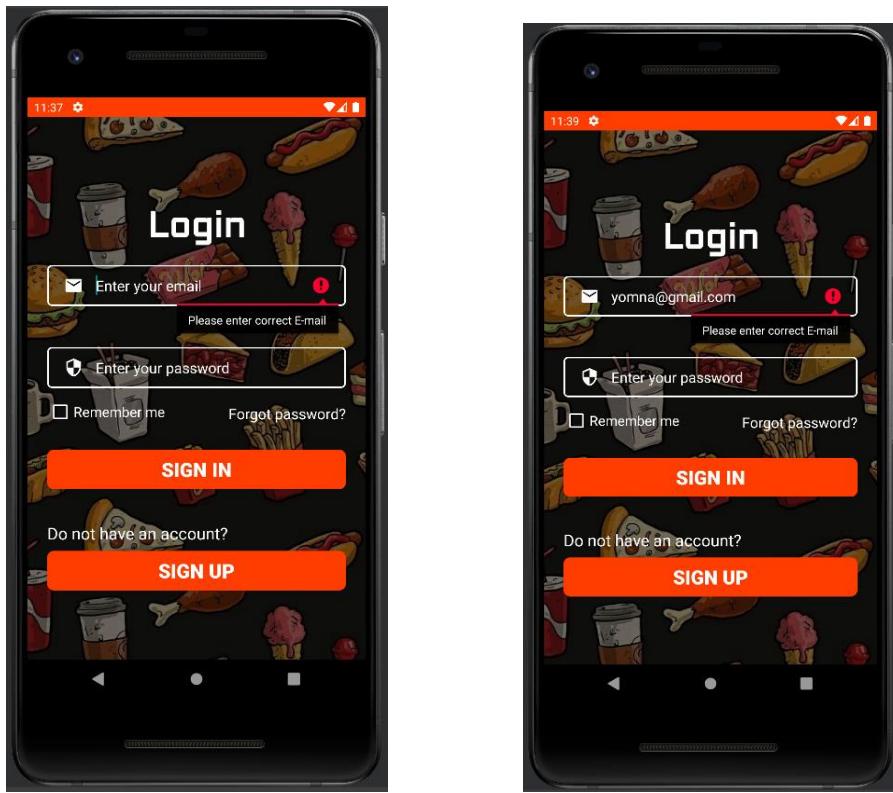
## ➤ Firebase Authentication User Added

Identifier	Providers	Created	Signed in	User UID
18p5794@eng.asu.edu.eg	✉	29 Dec 2022	29 Dec 2022	e31VogWsqaZde6uTsfA9kZXeJKO2
test123@eng.asu.edu.eg	✉	29 Dec 2022	29 Dec 2022	803j4dPv8Zfjuig8oLauECd3ANq1

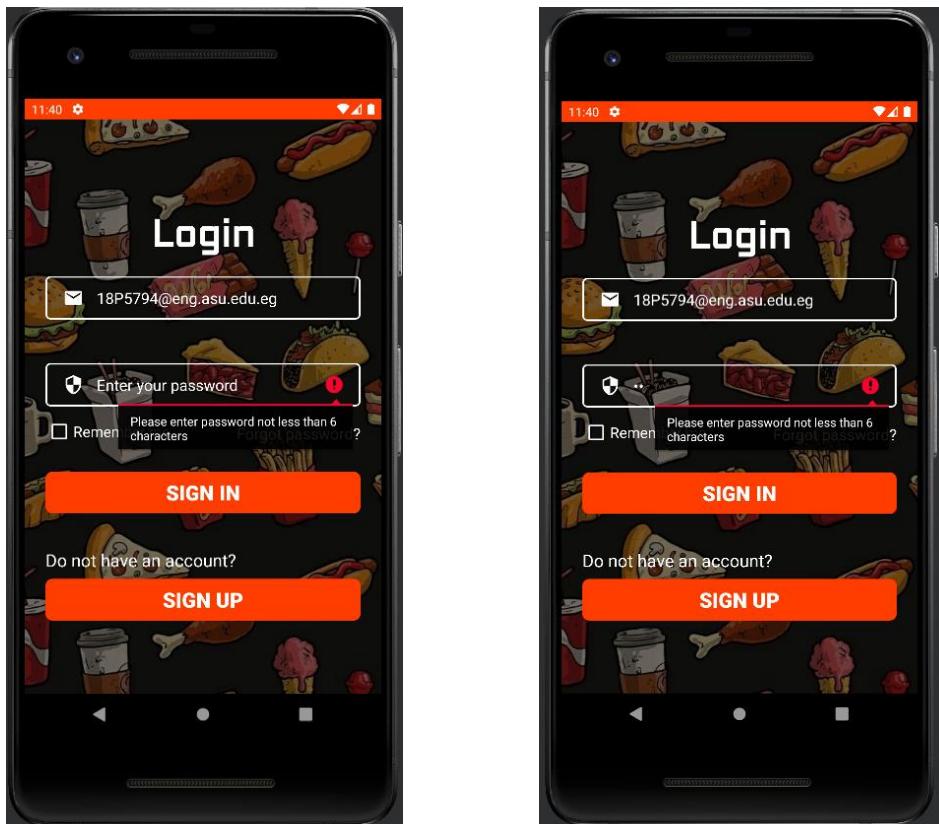
## III. LOGIN

### ➤ Test Cases

1. Not entering an email or entering an incorrect email (must be an @eng.asu.edu.eg email)



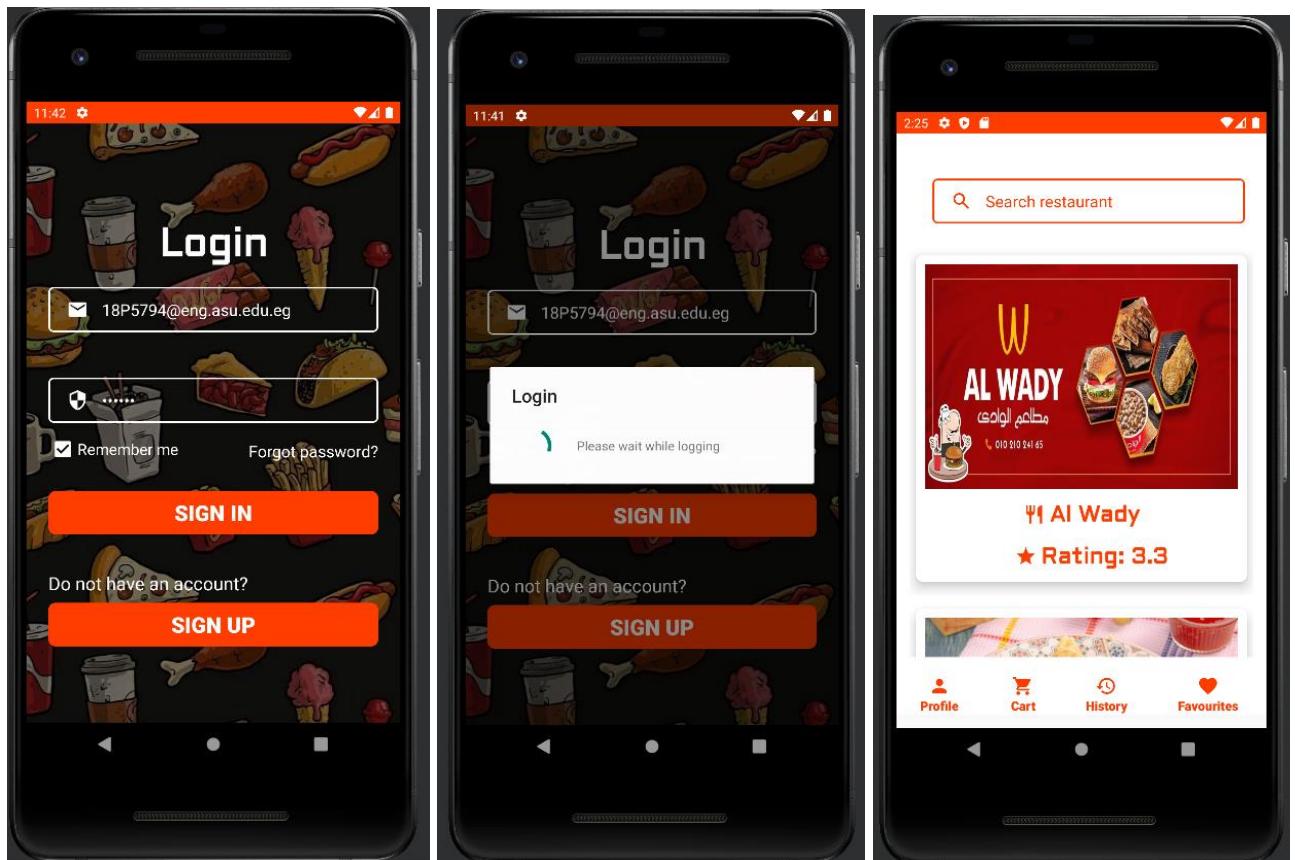
2. Not entering a password or entering password less than 6 characters



3. Entering incorrect email or password that doesn't exist in the database



4. Entering correct email and password that exist in the database



## ➤ Code

### Login Activity

```
package com.example.wagbaapplication;

import 15egistr.annotation.NonNull;
import 15egistr.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

import io.paperdb.Paper;

public class Login extends AppCompatActivity {

    EditText loginEmail, loginPassword;
    Button signUp, signIn;
    CheckBox rememberMe;

    String emailPattern = "(\w*)@eng\.asu\.edu\.eg";
    ProgressDialog progressDialog;
    FirebaseAuth myAuth;
    FirebaseUser myUser;
    //    String UserName, PhoneNumber;
    //    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        loginEmail = (EditText) findViewById(R.id.editTextEmail2);
        loginPassword = (EditText) findViewById(R.id.editTextPassword2);
        signIn = (Button) findViewById(R.id.signInButton);
        signUp = (Button) findViewById(R.id.signUpButton2);
        progressDialog = new ProgressDialog(this);
        myAuth = FirebaseAuth.getInstance();
        myUser = myAuth.getCurrentUser();
        //    databaseReference =
        FirebaseDatabase.getInstance().getReference().child("Users");

        rememberMe = (CheckBox) findViewById(R.id.rememberMe);
        Paper.init(this);

        Intent intent01 = new Intent(this, 15egistration.class);
    }
}
```

```
signIn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        login();
    }
}) ;

signUp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(intent01);
    }
}) ;
}

private void login() {
    String inputEmail = loginEmail.getText().toString();
    String inputPass = loginPassword.getText().toString();

    if(rememberMe.isChecked()){
        Paper.book().write(Prevalent.emailKey,inputEmail);
        Paper.book().write(Prevalent.passwordKey,inputPass);
    }

    if(!inputEmail.matches(emailPattern)){
        loginEmail.setError("Please enter correct E-mail");
        loginEmail.requestFocus();
    }else if(inputPass.isEmpty() || inputPass.length()<6){
        loginPassword.setError("Please enter password not less than 6
characters");
        loginPassword.requestFocus();
    }else {
        progressDialog.setMessage("Please wait while logging");
        progressDialog.setTitle("Login");
        progressDialog.setCanceledOnTouchOutside(false);
        progressDialog.show();
    }

myAuth.signInWithEmailAndPassword(inputEmail, inputPass).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful()){
            progressDialog.dismiss();

            // Read from the database
        }
        databaseReference.child(myUser.getUid()).addValueEventListener(new
ValueEventListener() {
        //                                     @Override
        //                                     public void onDataChange(DataSnapshot dataSnapshot) {
        //                                         UserName =
        dataSnapshot.child("UserName").getValue().toString();
        //                                         PhoneNumber =
        dataSnapshot.child("PhoneNumber").getValue().toString();
        //                                         }
        //                                     @Override
        //                                     public void onCancelled(DatabaseError error) {
        //                                         }
        //                                     });
    }
}
UserDatabase userDatabase =
```

```

UserDatabase.getUserDatabase(getApplicationContext());
//                                              final UserDao userDao = userDatabase.userDao();
//                                              new Thread(new Runnable() {
//                                                 @Override
//                                                 public void run() {
//                                                     if(!userDao.userExists(myUser.getUid())) {
//                                                       //User user = new
//                                                       User(myUser.getUid(),UserName, myUser.getEmail(), PhoneNumber);
//                                                       userDao.insert(user);
//                                                       runOnUiThread(new Runnable() {
//                                                         @Override
//                                                         public void run() {
//                                                             Toast.makeText(Login.this, "Welcome
// Back!", Toast.LENGTH_SHORT).show();
//                                                       }
//                                                       }
//                                                       }
//                                                       }
//                                                       }
//                                                       }

//                                                       sendToNewActivity();
} else{
    progressDialog.dismiss();
    Toast.makeText(Login.this, "Incorrect email or password",
Toast.LENGTH_LONG).show();
}

}
}

private void sendToNewActivity() {
Intent intent02 = new Intent(this, Restaurants.class);

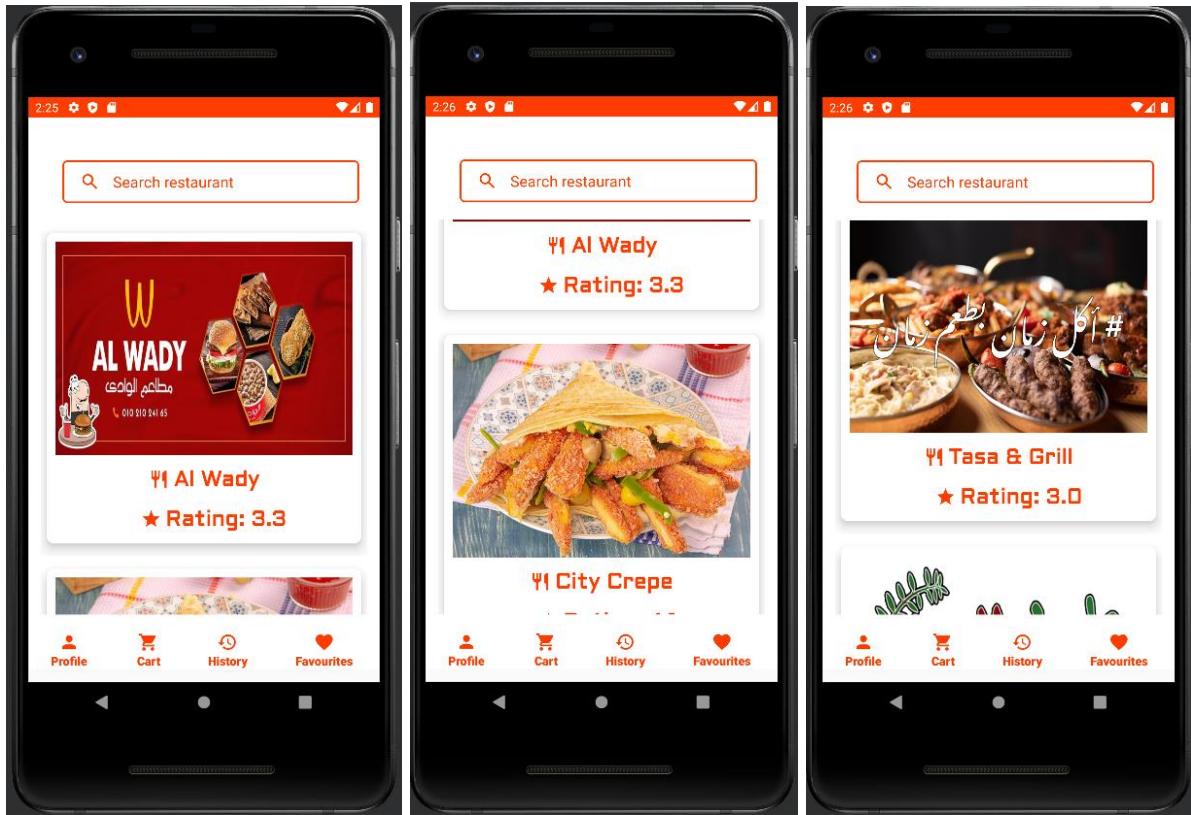
intent02.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK|Intent.FLAG_ACTIVITY_NEW_TASK);
startActivity(intent02);
finish();
}
}

```

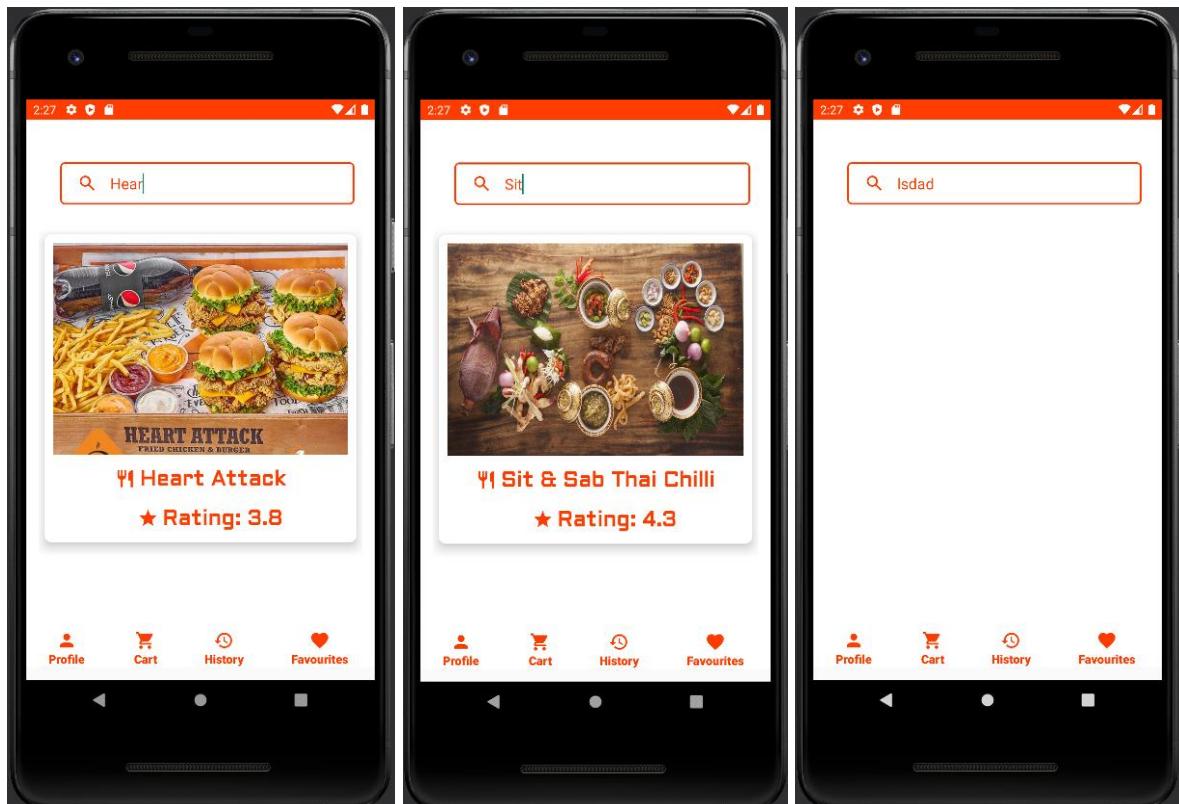
### III. RESTAURANTS

#### ➤ Test Cases

1. Navigating through the restaurant list



2. Searching for specific restaurant



## Restaurants Activity

```
package com.example.wagbaapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.Query;
import com.squareup.picasso.Picasso;

public class Restaurants extends AppCompatActivity {

    EditText searchRestaurant;
    RecyclerView recyclerView;
    TextView profile, cart, orderHistory, favourites;
    FirebaseRecyclerOptions<RestaurantObject> options;
    FirebaseRecyclerAdapter<RestaurantObject, RestaurantViewHolder> adapter;
    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_restaurants);

        searchRestaurant = (EditText) findViewById(R.id.searchRestaurant);
        recyclerView = (RecyclerView) findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new
        LinearLayoutManager(getApplicationContext()));
        recyclerView.setHasFixedSize(true);
        databaseReference =
        FirebaseDatabase.getInstance().getReference().child("Restaurants");

        profile = (TextView) findViewById(R.id.profileButton);
        cart = (TextView) findViewById(R.id.cartButton);
        orderHistory = (TextView) findViewById(R.id.orderHistoryButton);
        favourites = (TextView) findViewById(R.id.favouritesButton);

        orderHistory.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

        Intent intent = new Intent(Restaurants.this,
OrderHistory.class);
        startActivity(intent);
    }
});

cart.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(Restaurants.this, Cart.class);
        startActivity(intent);
    }
});

profile.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(Restaurants.this, Profile.class);
        startActivity(intent);
    }
});

favourites.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(Restaurants.this, Favourites.class);
        startActivity(intent);
    }
});

LoadRestaurants("");

searchRestaurant.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
    }

    @Override
    public void afterTextChanged(Editable editable) {
        if(editable.toString() != null)
            LoadRestaurants(editable.toString());
        else
            LoadRestaurants("");
    }
});

@Override
public void onBackPressed() {
    AlertDialog.Builder builder = new AlertDialog.Builder(Restaurants.this)
        .setMessage("Are you want to exit the application")
        .setPositiveButton("Yes", new DialogInterface.OnClickListener()
{
    @Override

```

```

        public void onClick(DialogInterface dialogInterface, int i)
    {
        finishAffinity();
    }
}).setNegativeButton("No", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i)
} );
builder.show();

}

private void LoadRestaurants(String data) {
    Query query =
databaseReference.orderByChild("Name").startAt(data).endAt(data+"\uf8ff");

    options = new
FirebaseRecyclerOptions.Builder<RestaurantObject>().setQuery(query, RestaurantOb
ject.class).build(); //here we set the query we want to make from the database
    adapter = new FirebaseRecyclerAdapter<RestaurantObject,
RestaurantViewHolder>(options) { //the adapter use the options to know which
objects to bind based on the query you made
        @Override
        protected void onBindViewHolder(@NonNull RestaurantViewHolder
holder, int position, @NonNull RestaurantObject model) {

            holder.name.setText(model.getName());
            holder.rating.setText(model.getRating());
            Picasso.get().load(model.getImage()).into(holder.image);

            holder.itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Intent intent = new Intent(Restaurants.this,
Dishes.class);

                    intent.putExtra("RestaurantId", getRef(holder.getAbsoluteAdapterPosition()).getKey());
                    intent.putExtra("RestaurantName", model.getName());
                    startActivity(intent);
                }
            });
        }

        @NonNull
        @Override
        public RestaurantViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {

            View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.restaurant_view, parent
t, false);
            return new RestaurantViewHolder(view);
        }
    };
    adapter.startListening();
    recyclerView.setAdapter(adapter);
}
}

```

## **RestaurantObject Class**

```
package com.example.wagbaapplication;

public class RestaurantObject {

    private String Name;
    private String Rating;
    private String Image;

    public RestaurantObject(String name, String rating, String image) {
        Name = name;
        Rating = rating;
        Image = image;
    }

    public RestaurantObject() {
    }

    public String getName() {
        return Name;
    }

    public void setName(String name) {
        Name = name;
    }

    public String getRating() {
        return Rating;
    }

    public void setRating(String rating) {
        Rating = rating;
    }

    public String getImage() {
        return Image;
    }

    public void setImage(String image) {
        Image = image;
    }
}
```

## **RestaurantViewHolder Class**

```
package com.example.wagbaapplication;

import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class RestaurantViewHolder extends RecyclerView.ViewHolder {

    ImageView image;
    TextView name, rating;
```

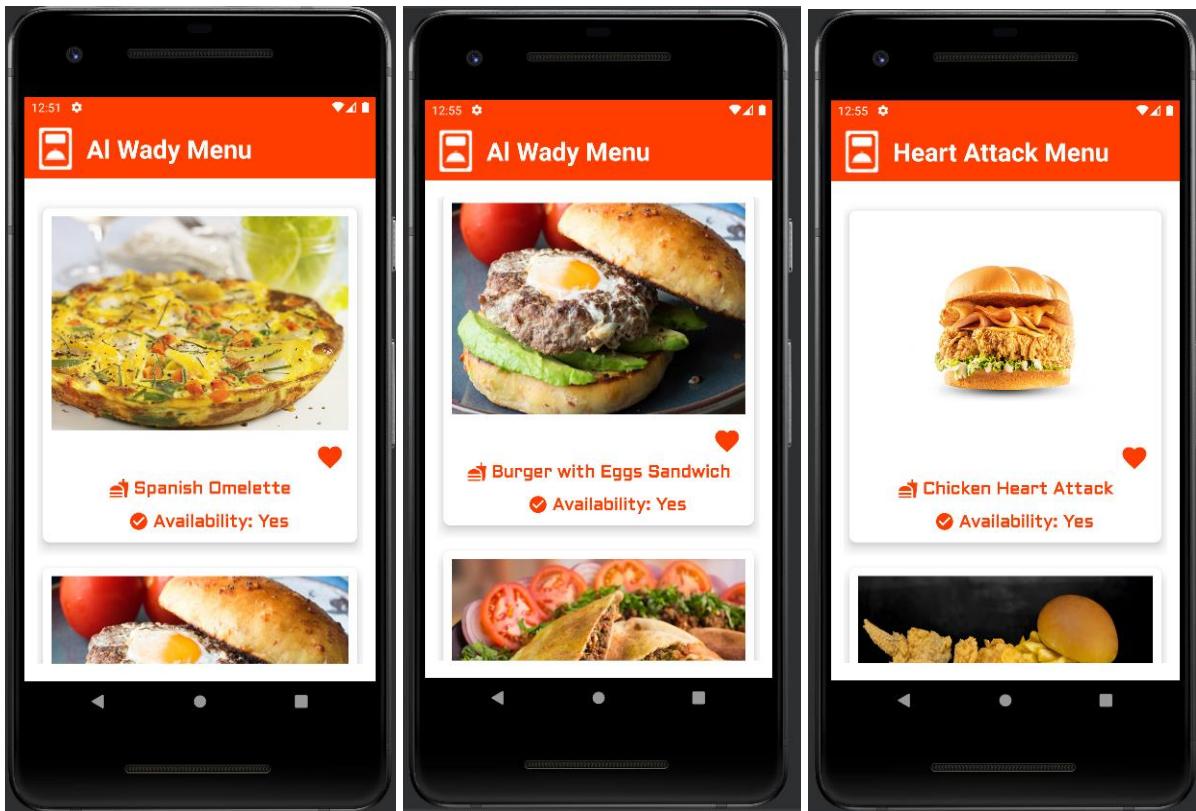
```
public RestaurantViewHolder(@NonNull View itemView) {
    super(itemView);

    image = itemView.findViewById(R.id.restaurantImage);
    name = itemView.findViewById(R.id.restaurantName);
    rating = itemView.findViewById(R.id.restaurantRating);
}
```

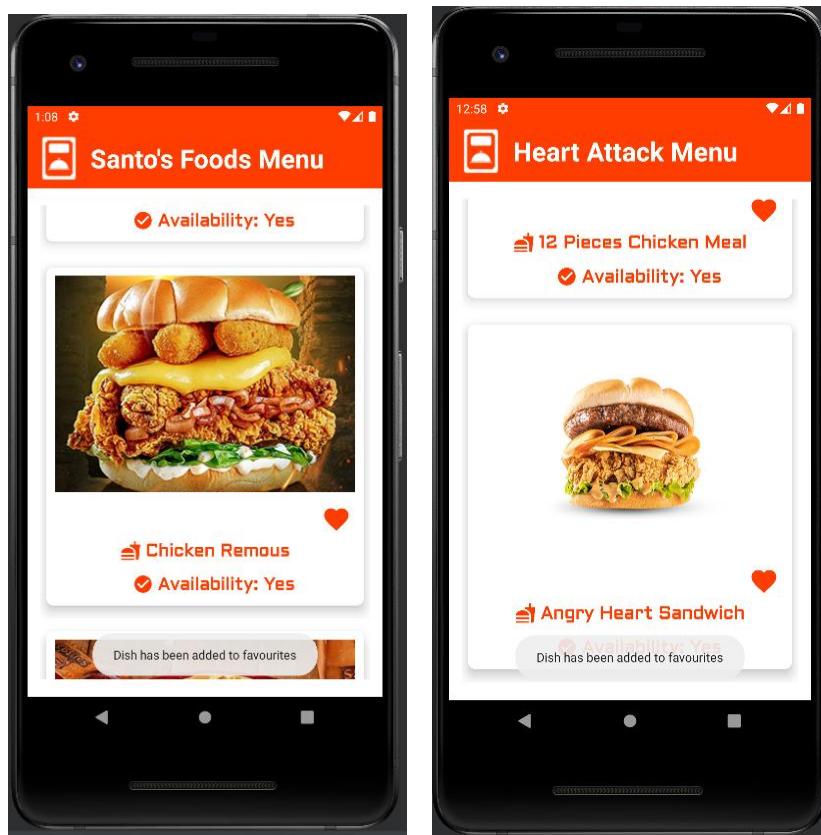
## IV. DISHES

### ➤ Test Cases

1. Navigating through the dishes list



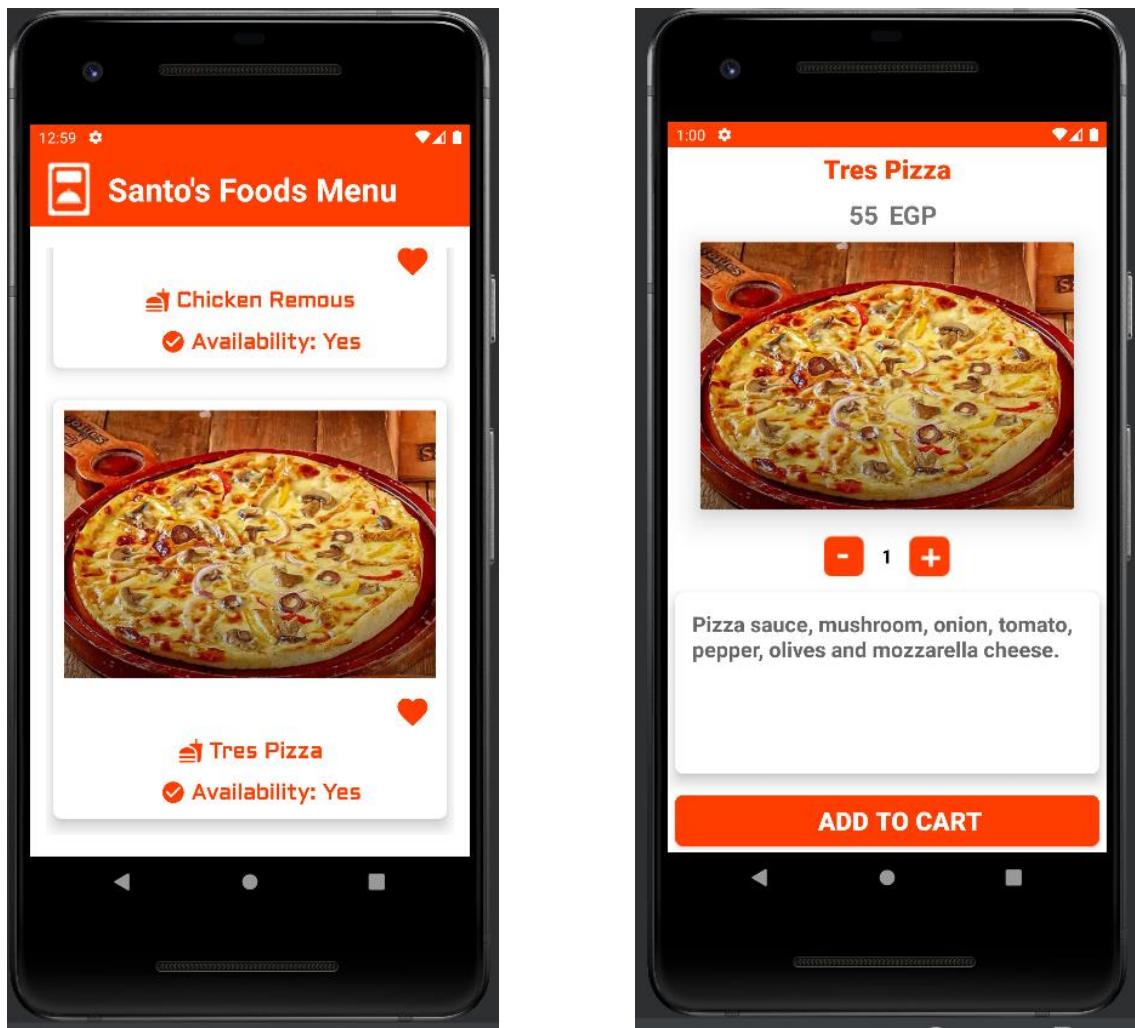
2. Clicking on the favorites button (Heart shaped button)



This screenshot shows the Firebase Realtime Database interface. The URL is https://wagbaapplication-9691c-default-firebaseio.com/. The database structure under the root is as follows:

- AllFavourites**: This node contains two child nodes, 12 and 23, which represent the IDs of the dishes added to favorites.
- 12**:
  - DishName: "Angry Heart Sandwich"
  - Image: "https://s3-eu-west-1.amazonaws.com/elmenusv5-stg/Normal/8a088238-4d7c-4e10-98fa-b9a156ef7e22.jpg"
  - Price: "85"
- 23**:
  - DishName: "Chicken Remous"
  - Image: "https://s3-eu-west-1.amazonaws.com/elmenusv5-stg/Normal/ec6f6b32-9c5f-4d46-bcc1-63975ddf3b76.jpg"
  - Price: "62"

3. Clicking on specific dish



## ➤ Code

### Dishes Activity

```
package com.example.wagbaapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebaseio.ui.database.FirebaseRecyclerOptions;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
```

```

import com.google.firebaseio.database.ValueEventListener;
import com.squareup.picasso.Picasso;

public class Dishes extends AppCompatActivity {

    RecyclerView recyclerView;
    FirebaseRecyclerOptions<DishObject> options;
    FirebaseRecyclerAdapter<DishObject, DishViewHolder> adapter;
    DatabaseReference databaseReference;
    String RestaurantId = "";
    String RestaurantName = "";
    TextView RestaurantTitle;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dishes);

        RestaurantTitle = (TextView) findViewById(R.id.dishesListTitle);
        recyclerView = (RecyclerView) findViewById(R.id.recyclerView2);
        recyclerView.setLayoutManager(new
        LinearLayoutManager(getApplicationContext()));
        recyclerView.setHasFixedSize(true);
        databaseReference = FirebaseDatabase.getInstance().getReference();

        //Getting the intent from restaurant page
        if (getIntent() != null) {
            RestaurantId = getIntent().getStringExtra("RestaurantId");
            RestaurantName = getIntent().getStringExtra("RestaurantName");
        }

        if (!RestaurantId.isEmpty() && RestaurantId != null)
            LoadDishes(RestaurantId);
    }

    private void LoadDishes(String restaurantId) {
        RestaurantTitle.setText(RestaurantName+ " Menu");

        // Select * from dishes where DishId == Restaurant Id
        options = new
        FirebaseRecyclerOptions.Builder<DishObject>().setQuery(databaseReference.child("Dishe
        s").orderByChild("DishId").equalTo(restaurantId), DishObject.class).build();
        adapter = new FirebaseRecyclerAdapter<DishObject, DishViewHolder>(options) {
            @Override
            protected void onBindViewHolder(@NonNull DishViewHolder holder, int
            position, @NonNull DishObject model) {

                holder.name.setText(model.getName());
                holder.availability.setText(model.getAvailability());
                Picasso.get().load(model.getImage()).into(holder.image);

                holder.itemView.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        Intent intent = new Intent(Dishes.this, DishDetail.class);
                        intent.putExtra("DishId", getRef(holder.getAbsoluteAdapterPosition()).getKey());
                        startActivity(intent);
                    }
                });
            }
        };
    }
}

```

```

        holder.favourite.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                DatabaseReference myRef =
databaseReference.child("AllFavourites").child(FirebaseAuth.getInstance().getCurrentUser().getUid()).child(holder.getAbsoluteAdapterPosition()).getKey();

                myRef.child("Image").setValue(model.getImage());
                myRef.child("DishName").setValue(model.getName());
                myRef.child("Price").setValue(model.getPrice());

                Toast.makeText(Dishes.this, "Dish has been added to
favourites", Toast.LENGTH_SHORT).show();
            }
        });
    }

    @NonNull
    @Override
    public DishViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.dish_view,parent,false);
        return new DishViewHolder(view);
    }
};

adapter.startListening();
recyclerView.setAdapter(adapter);
}
}

```

## DishObject Class

```

package com.example.wagbaapplication;

public class DishObject {

    private String Name;
    private String Price;
    private String Description;
    private String Image;
    private String DishId;
    private String Availability;

    public DishObject(String name, String price, String description, String image,
String dishId, String availability) {
        Name = name;
        Price = price;
        Description = description;
        Image = image;
        DishId = dishId;
        Availability = availability;
    }

    public DishObject() {
    }

    public String getName() {
        return Name;
    }
}

```

```

public void setName(String name) {
    Name = name;
}

public String getPrice() {
    return Price;
}

public void setPrice(String price) {
    Price = price;
}

public String getDescription() {
    return Description;
}

public void setDescription(String description) {
    Description = description;
}

public String getImage() {
    return Image;
}

public void setImage(String image) {
    Image = image;
}

public String getDishId() {
    return DishId;
}

public void setDishId(String dishId) {
    DishId = dishId;
}

public String getAvailability() {
    return Availability;
}

public void setAvailability(String availability) {
    Availability = availability;
}
}

```

## DishViewHolder Class

```

package com.example.wagbaapplication;

import android.view.View;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class DishViewHolder extends RecyclerView.ViewHolder {

```

```
ImageView image;
TextView name,availability;
ImageButton favourite;

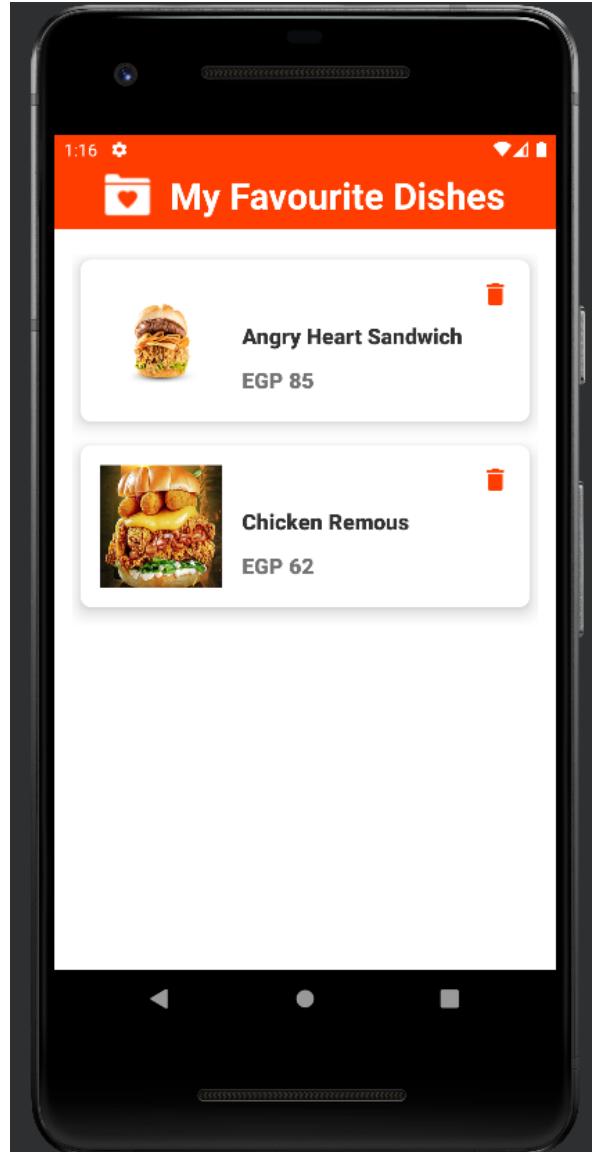
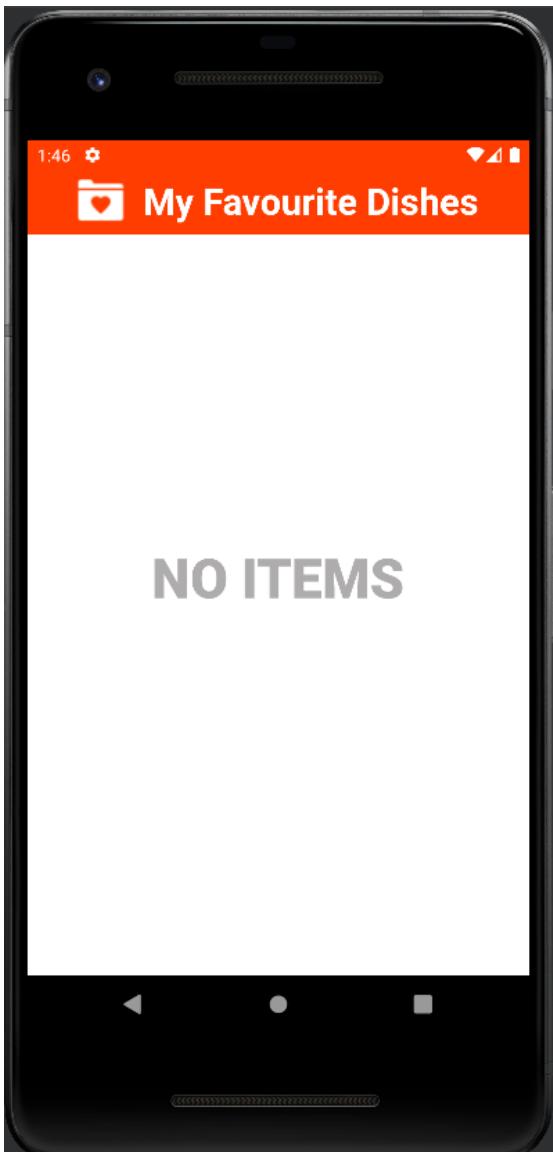
public DishViewHolder(@NonNull View itemView) {
    super(itemView);

    image = itemView.findViewById(R.id.dishImage);
    name = itemView.findViewById(R.id.dishName);
    availability = itemView.findViewById(R.id.dishAvailability);
    favourite = itemView.findViewById(R.id.heartButton);
}
```

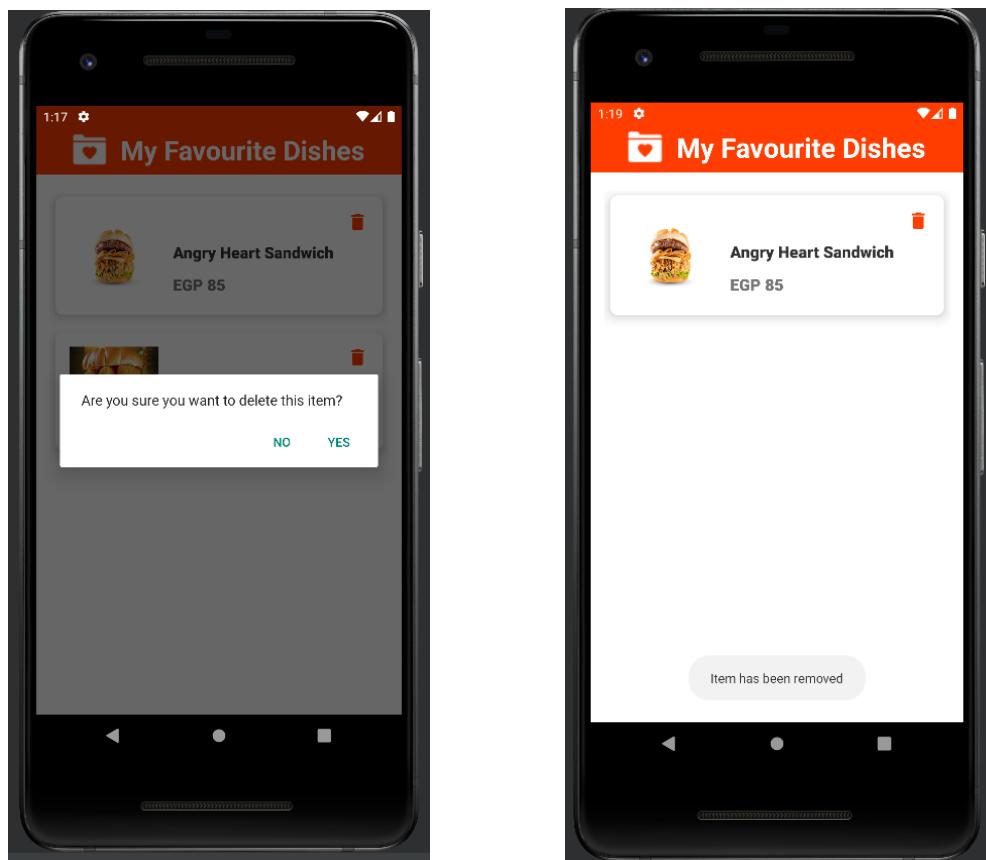
## V. FAVOURITES

### ➤ Test Cases

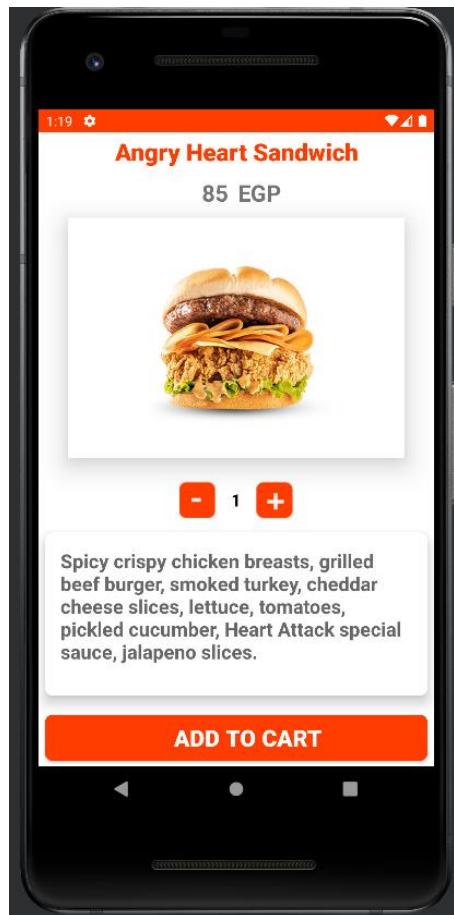
1. Favorite dishes list is Empty / Navigating through the favorite dishes list



2. Clicking on the delete button (bin shaped button)



3. Clicking on specific favorite dish



## Code

### Favourites Activity

```
package com.example.wagbaapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.firebaseio.ui.database.FirebaseRecyclerAdapter;
import com.firebaseio.ui.database.FirebaseRecyclerOptions;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.Picasso;

public class Favourites extends AppCompatActivity {

    RecyclerView recyclerView;
    TextView noItems;
    FirebaseRecyclerOptions<OrderObject> options;
    FirebaseRecyclerAdapter<OrderObject, FavouriteViewHolder> adapter;
    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_favourites);

        noItems = (TextView) findViewById(R.id.noItems);
        recyclerView = (RecyclerView) findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(getApplicationContext()));
        recyclerView.setHasFixedSize(true);
        databaseReference = FirebaseDatabase.getInstance().getReference()

.child("AllFavourites").child(FirebaseAuth.getInstance().getCurrentUser().getUid());

        LoadFavouritesList();
        CheckIfFavouritesIsEmpty();
    }

    private void CheckIfFavouritesIsEmpty() {
        databaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
```

```

        if (!dataSnapshot.hasChildren()) {
            noItems.setVisibility(View.VISIBLE);
        }
    }
    @Override
    public void onCancelled(DatabaseError error) {
        // Failed to read value
    }
}) ;
}

private void LoadFavouritesList() {
    options = new
FirebaseRecyclerOptions.Builder<OrderObject>().setQuery(databaseReference, OrderObject
.class).build();
    adapter = new FirebaseRecyclerAdapter<OrderObject,
FavouriteViewHolder>(options) {
        @Override
        protected void onBindViewHolder(@NonNull FavouriteViewHolder holder, int
position, @NonNull OrderObject model) {

            holder.name.setText(model.getDishName());
            holder.price.setText("EGP "+model.getPrice());
            Picasso.get().load(model.getImage()).into(holder.image);

            holder.itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Intent intent = new Intent(Favourites.this,
DishDetail.class);
                    intent.putExtra("DishId",getRef(holder.getAbsoluteAdapterPosition()).getKey());
                    startActivity(intent);
                }
            });
            holder.delete.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    AlertDialog.Builder builder = new
AlertDialog.Builder(Favourites.this)
                        .setMessage("Are you sure you want to delete this
item?")
                        .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface
dialogInterface, int i) {
                                databaseReference.child(getRef(holder.getAbsoluteAdapterPosition()).getKey())
                                    .removeValue().addOnCompleteListener(new OnCompleteListener<Void>() {
                                        @Override
                                        public void onComplete(@NonNull
Task<Void> task) {
                                            if(task.isSuccessful()){
                                                Toast.makeText(Favourites.this, "Item has been removed", Toast.LENGTH_LONG).show();
                                            }
                                        }
                                    });
                            }
                        });
                }
            });
        }
    }
}

```

```

                }).setNegativeButton("No", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface
dialogInterface, int i) {
        //do nothing
    }
});
builder.show();
}
);

}

@NonNull
@Override
public FavouriteViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
    View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.favourite_view,parent,false
);
    return new FavouriteViewHolder(view);
}
;
adapter.startListening();
recyclerView.setAdapter(adapter);
}
}
}

```

## FavouriteViewHolder Class

```

package com.example.wagbaapplication;

import android.view.View;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class FavouriteViewHolder extends RecyclerView.ViewHolder{

    ImageView image;
    TextView name, price;
    ImageButton delete;

    public FavouriteViewHolder(@NonNull View itemView) {
        super(itemView);

        image = itemView.findViewById(R.id.orderImageFavourite);
        name = itemView.findViewById(R.id.orderNameFavourite);
        price = itemView.findViewById(R.id.orderPriceFavourite);
        delete = itemView.findViewById(R.id.deleteFavouriteButton);

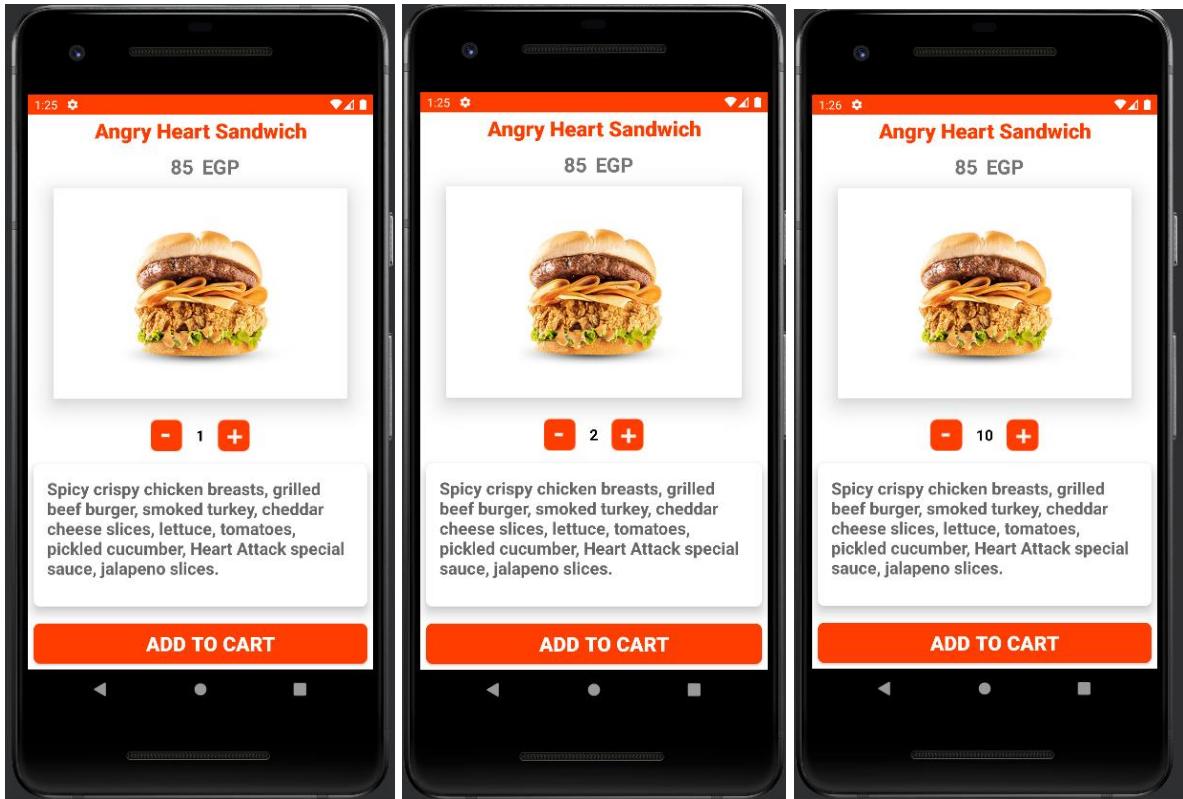
    }
}

```

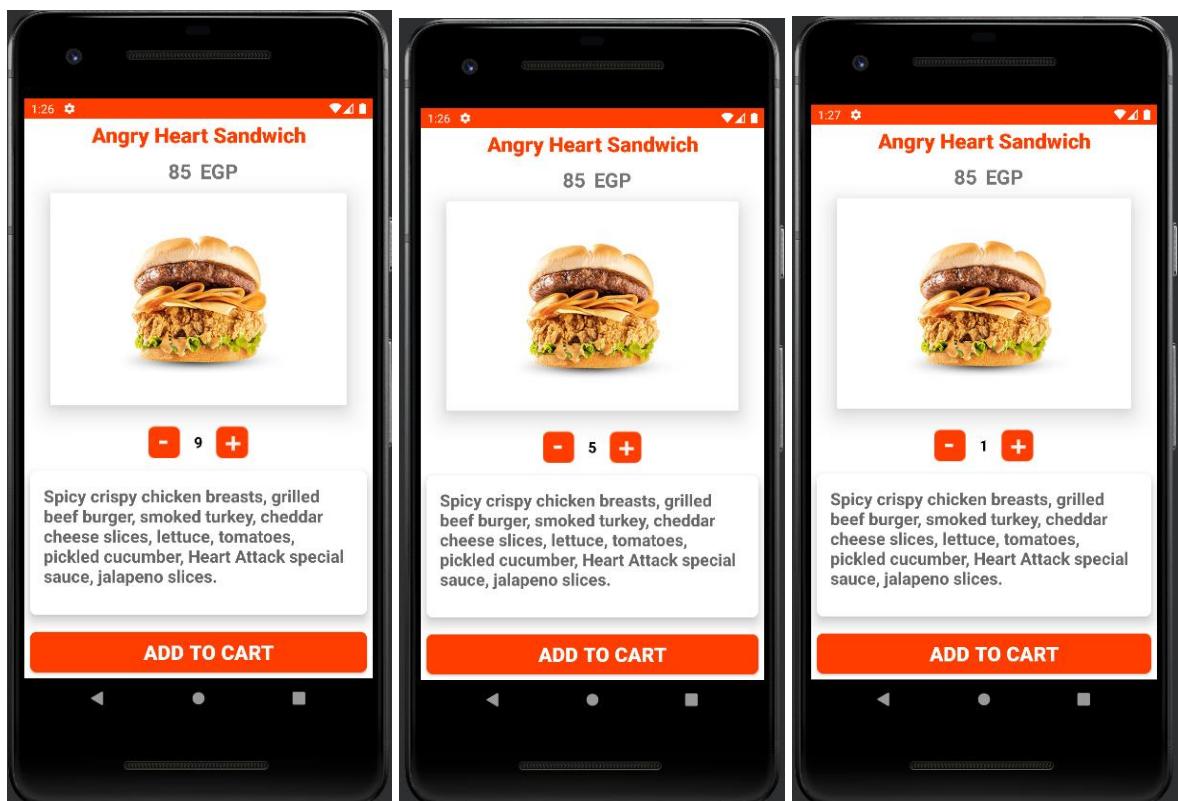
# 1.DISH DETAIL

## ➤ Test Cases

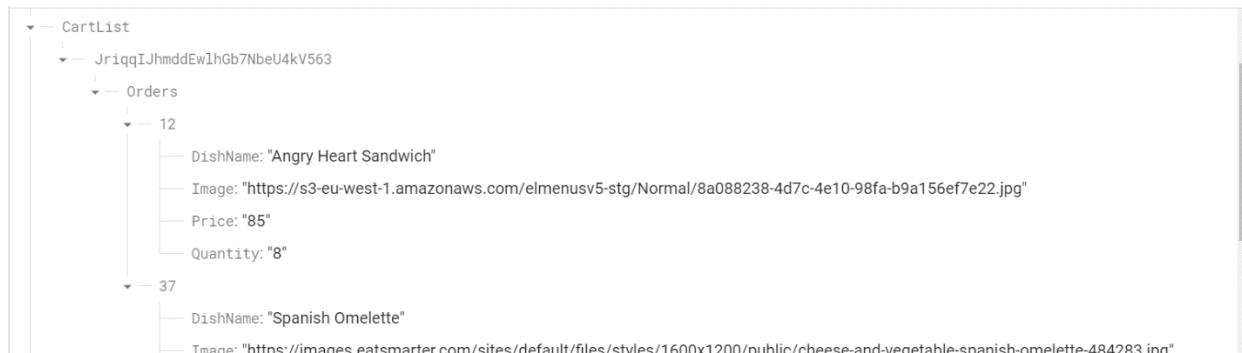
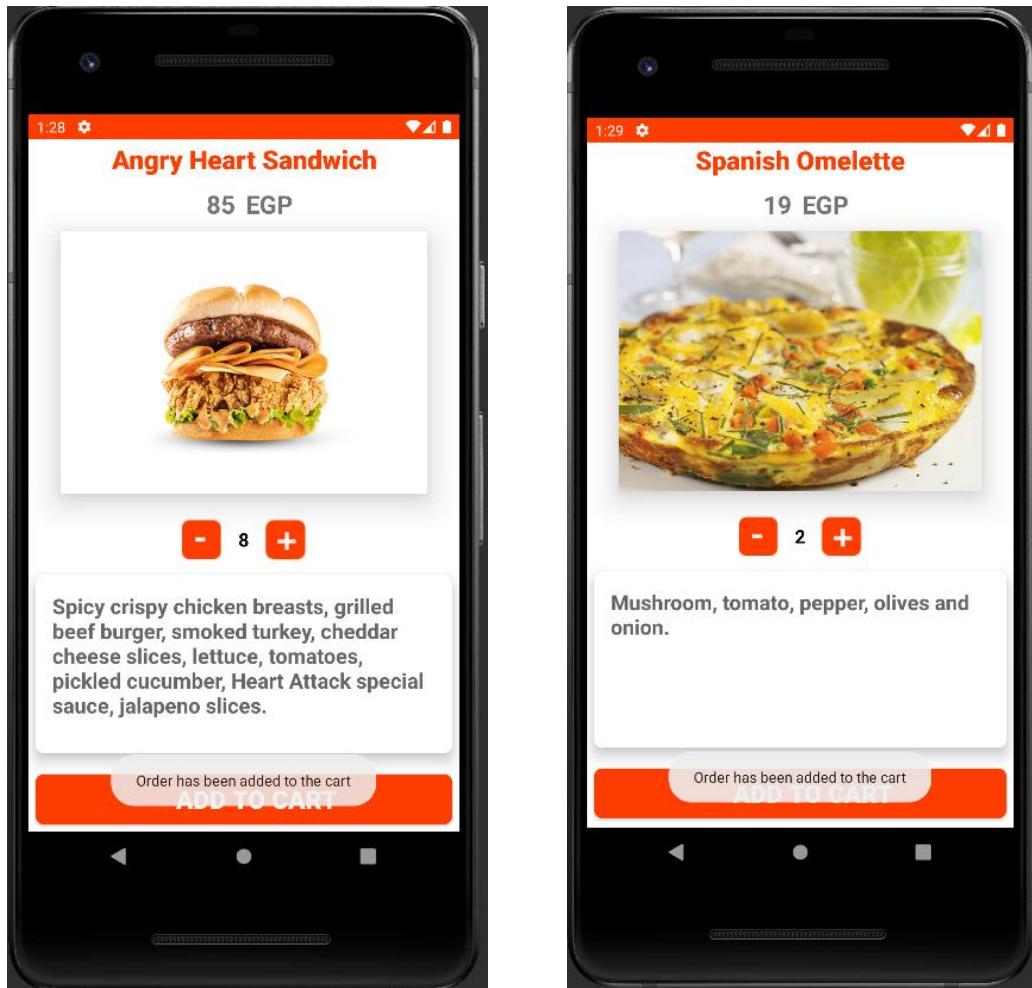
1. Clicking on add button



2. Clicking on minus button



### 3. Clicking on add to cart button



### DishDetail Activity

```
package com.example.wagbaapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
```

```

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;
import com.squareup.picasso.Picasso;

import java.util.HashMap;

public class DishDetail extends AppCompatActivity {

    TextView dishTitle, dishPrice, dishQuantity, dishDescription;
    ImageView dishImage, minus, plus;
    Button addToCart;
    DatabaseReference databaseReference;
    String DishId = "";
    String imageURL;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dish_detail);

        databaseReference = FirebaseDatabase.getInstance().getReference();
        dishTitle = (TextView) findViewById(R.id.dishDetailTitle);
        dishPrice = (TextView) findViewById(R.id.dishDetailPrice);
        dishQuantity = (TextView) findViewById(R.id.orderQuantity);
        dishDescription = (TextView) findViewById(R.id.dishDescription);
        dishImage = (ImageView) findViewById(R.id.dishDetailImage);
        minus = (ImageView) findViewById(R.id.minus);
        plus = (ImageView) findViewById(R.id.plus);
        addToCart = (Button) findViewById(R.id.addCart);

        //activating the plus and minus buttons
        minus.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String quantity = dishQuantity.getText().toString();
                if(!quantity.equals("1")){
                    int quantityNumber = Integer.parseInt(quantity);
                    quantityNumber = quantityNumber - 1;
                    dishQuantity.setText(String.valueOf(quantityNumber));
                }
            }
        });

        plus.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String quantity = dishQuantity.getText().toString();
                int quantityNumber = Integer.parseInt(quantity);
                quantityNumber = quantityNumber + 1;
                dishQuantity.setText(String.valueOf(quantityNumber));
            }
        });

        //Getting the intent from dish page
        if (getIntent() != null)
    }
}

```

```

        DishId = getIntent().getStringExtra("DishId");

        if (!DishId.isEmpty() && DishId != null)
            LoadDishesDetails(DishId);

databaseReference.child("Dishes").child(DishId).child("Image").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        // This method is called once with the initial value and again
        // whenever data at this location is updated.
        imageURL = dataSnapshot.getValue(String.class);
    }
    @Override
    public void onCancelled(DatabaseError error) {
        // Failed to read value
    }
}) ;

addToCart.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (!DishId.isEmpty() && DishId != null)
            addToCartListFireBase(DishId);
    }
}) ;

private void addToCartListFireBase(String dishId) {
    final HashMap<String, Object> cartMap = new HashMap<>();
    cartMap.put("DishName", dishTitle.getText().toString());
    cartMap.put("Quantity", dishQuantity.getText().toString());
    cartMap.put("Price", dishPrice.getText().toString());
    cartMap.put("Image", imageURL);

databaseReference.child("CartList").child(FirebaseAuth.getInstance().getCurrentUser().getUid())
    .child("Orders")
    .child(dishId)
    .updateChildren(cartMap)
    .addOnCompleteListener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if(task.isSuccessful()){
                Toast.makeText(DishDetail.this, "Order has been added to the cart", Toast.LENGTH_LONG).show();
            }
        }
    });
}

private void LoadDishesDetails(String dishId) {

databaseReference.child("Dishes").child(dishId).addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        // This method is called once with the initial value and again
        // whenever data at this location is updated.
        //DishObject contains same key values of each dish that in the
    }
});
}

```

```

firebase

        DishObject dishObject =
dataSnapshot.getValue(DishObject.class);

        dishTitle.setText(dishObject.getName());
        dishPrice.setText(dishObject.getPrice());
        dishDescription.setText(dishObject.getDescription());
        Picasso.get().load(dishObject.getImage()).into(dishImage);
    }

    @Override
    public void onCancelled(DatabaseError error) {
        // Failed to read value
    }
});;

}

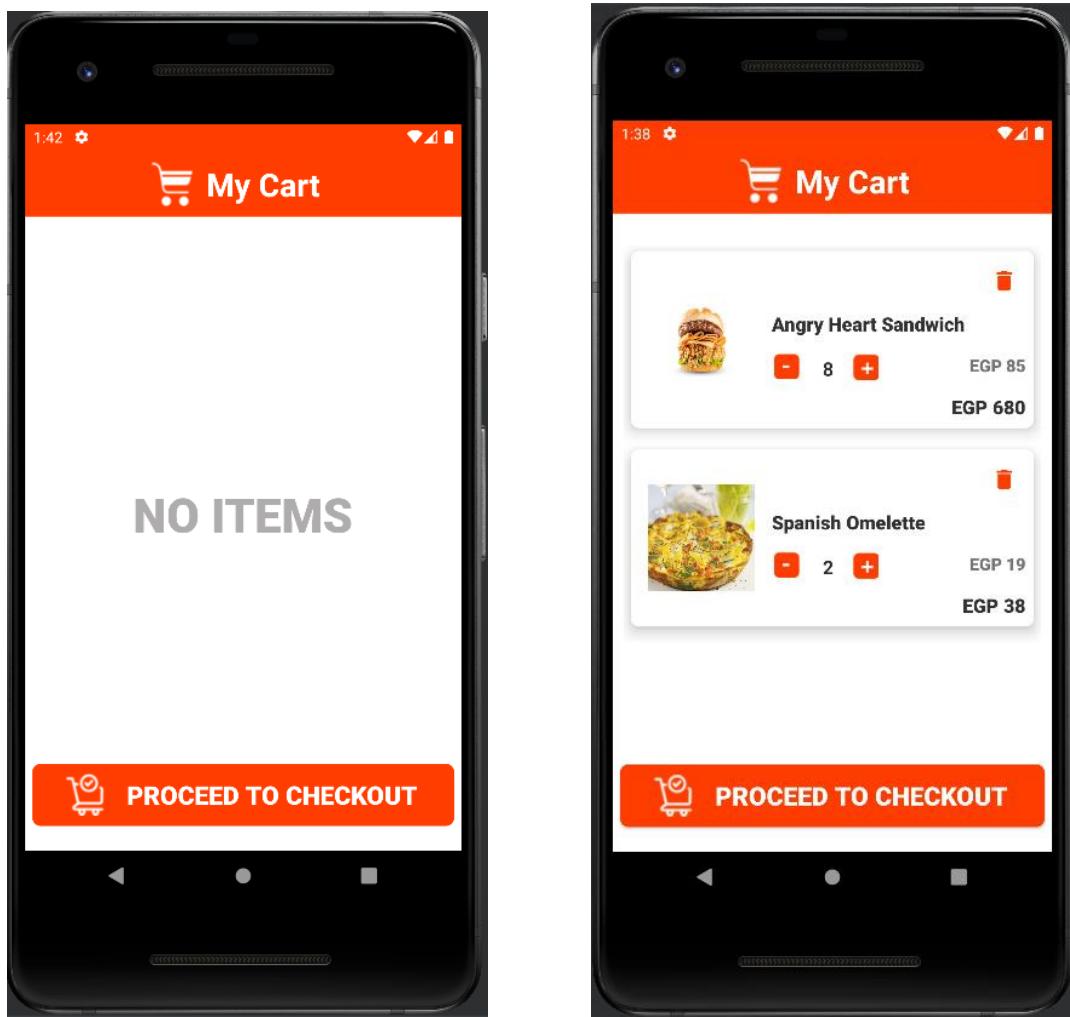
}

```

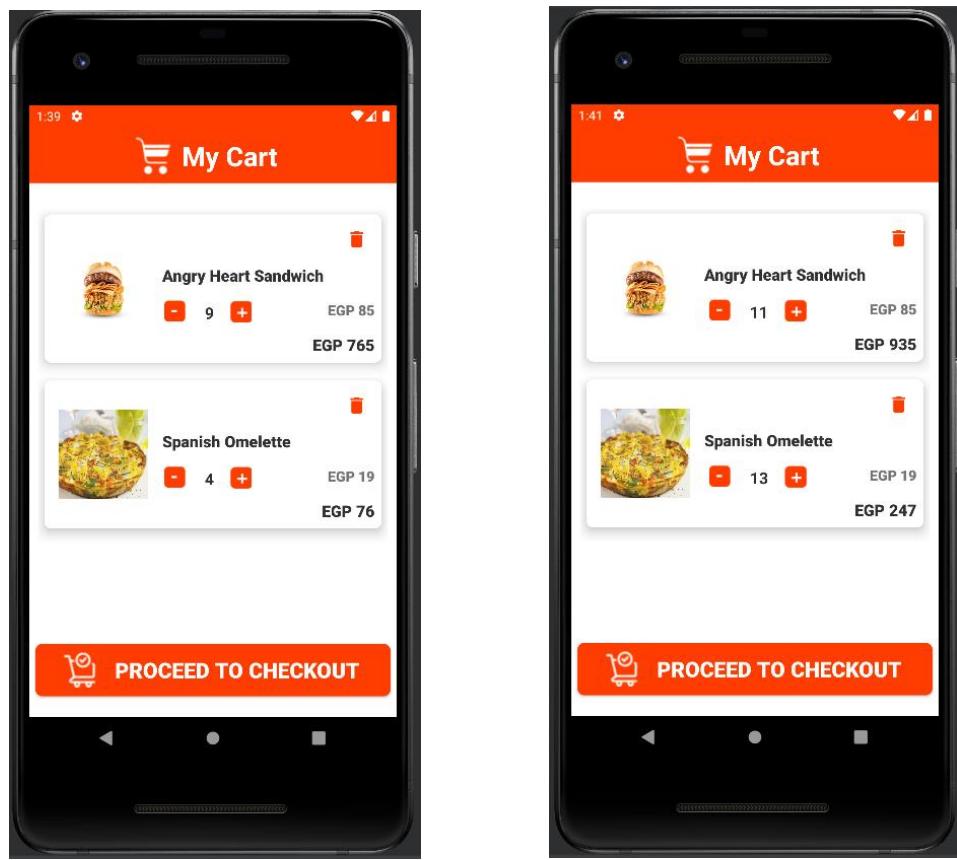
## VII. CART

### ➤ Test Cases

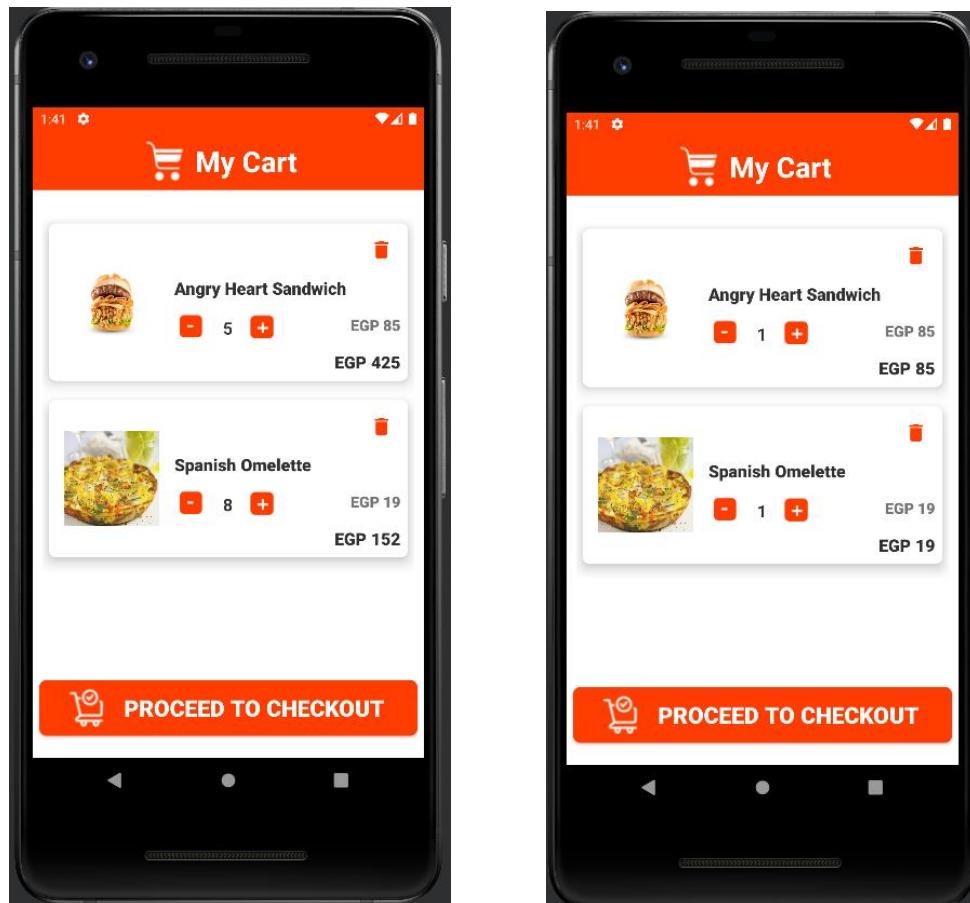
1. Cart is Empty / Navigating through the cart list when it is not empty (note: proceed to checkout button is disabled if there are no items in the cart)



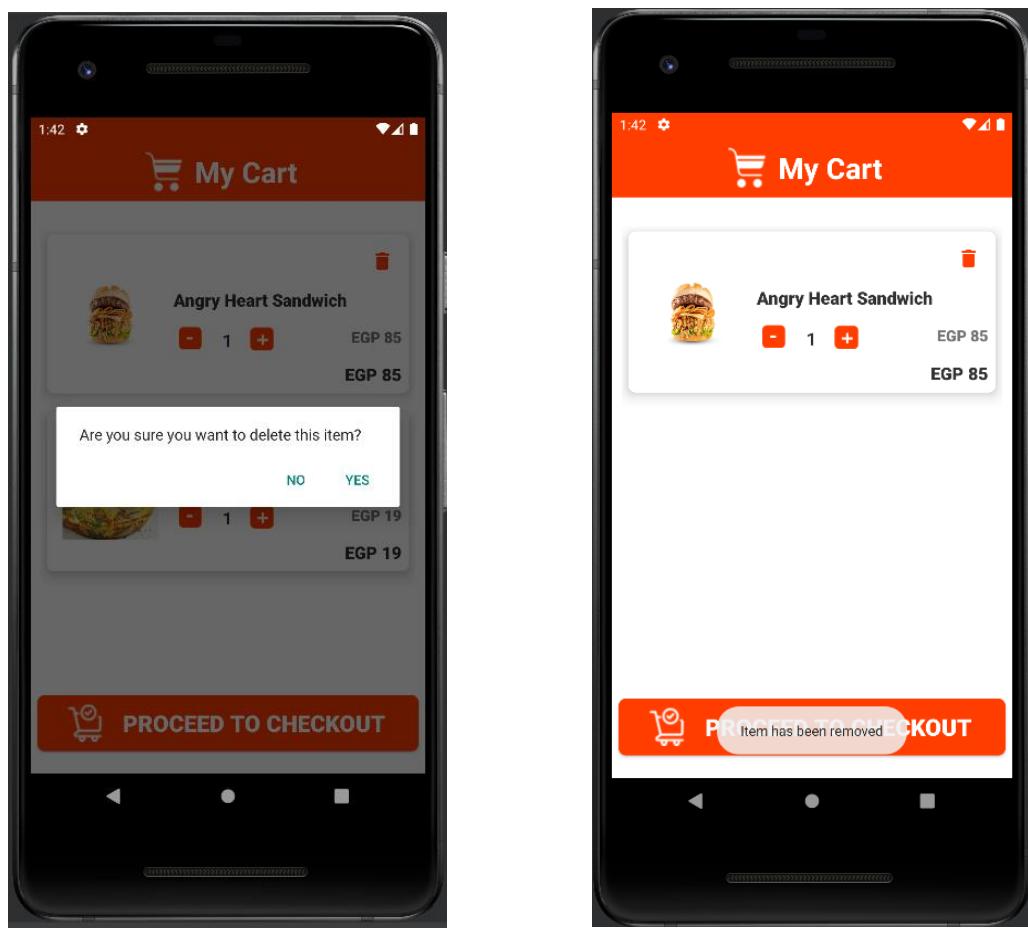
2. Clicking on add button



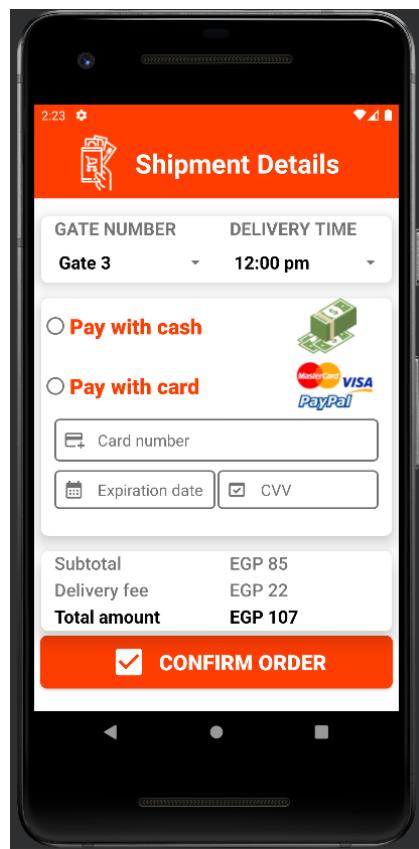
3. Clicking on minus button



4. Clicking on the delete button (bin shaped button)



5. Clicking on proceed to checkout button



## ➤ Code

### Cart Activity

```
package com.example.wagbaapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.firebaseio.ui.database.FirebaseRecyclerAdapter;
import com.firebaseio.ui.database.FirebaseRecyclerOptions;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.Picasso;

public class Cart extends AppCompatActivity {

    RecyclerView recyclerView;
    Button proceedButton;
    TextView noItems;
    FirebaseRecyclerOptions<OrderObject> options;
    FirebaseRecyclerAdapter<OrderObject, OrderViewHolder> adapter;
    DatabaseReference databaseReference;
    int totalPrice = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cart);

        proceedButton = (Button) findViewById(R.id.proceedToCheckout);
        noItems = (TextView) findViewById(R.id.noItems);
        recyclerView = (RecyclerView) findViewById(R.id.cartRecyclerView);
        recyclerView.setLayoutManager(new
        LinearLayoutManager(getApplicationContext()));
        recyclerView.setHasFixedSize(true);
        databaseReference = FirebaseDatabase.getInstance().getReference()

        .child("CartList").child(FirebaseAuth.getInstance().getCurrentUser().getUid()).child(
        "Orders");
    }
}
```

```

        LoadCartList();
        CheckIfCartIsEmpty();

        proceedButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(Cart.this, ConfirmOrder.class);
                startActivity(intent);
            }
        });
    }

    private void CheckIfCartIsEmpty() {
        databaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if(!dataSnapshot.hasChildren()) {
                    noItems.setVisibility(View.VISIBLE);
                    proceedButton.setEnabled(false);
                }
            }
            @Override
            public void onCancelled(DatabaseError error) {
                // Failed to read value
            }
        });
    }

    private void LoadCartList() {
        options = new FirebaseRecyclerOptions.Builder<OrderObject>()
            .setQuery(databaseReference, OrderObject.class)
            .build();
        adapter = new FirebaseRecyclerAdapter<OrderObject, OrderViewHolder>(options) {
            @Override
            protected void onBindViewHolder(@NonNull OrderViewHolder holder, int position, @NonNull OrderObject model) {
                holder.name.setText(model.getDishName());
                holder.price.setText("EGP "+model.getPrice());
                holder.quantity.setText(model.getQuantity());
                Picasso.get().load(model.getImage()).into(holder.image);

                //This is for each separate item on cart
                int price_of_one_item = Integer.valueOf(model.getPrice());
                int final_price_perItem =
                    price_of_one_item*Integer.valueOf(model.getQuantity());
                holder.totalPrice.setText("EGP
                    "+String.valueOf(final_price_perItem));

                holder.minus.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        if(!model.getQuantity().equals("1")){
                            int currentQuantity =
                                Integer.parseInt(model.getQuantity());
                            currentQuantity = currentQuantity - 1;
                            String current_quantity_string =
                                String.valueOf(currentQuantity);
                            databaseReference
                                .child(getRef(holder.getAbsoluteAdapterPosition()).getKey())

```

```

        .child("Quantity").setValue(current_quantity_string);
    }
}
} );

holder.plus.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        int currentQuantity = Integer.parseInt(model.getQuantity());
        currentQuantity = currentQuantity + 1;
        String current_quantity_string =
String.valueOf(currentQuantity);
        databaseReference
.child(getRef(holder.getAbsoluteAdapterPosition()).getKey())
        .child("Quantity").setValue(current_quantity_string);
    }
} );

holder.delete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        AlertDialog.Builder builder = new
AlertDialog.Builder(Cart.this)
        .setMessage("Are you sure you want to delete this
item?")
        .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
dialogInterface, int i) {
                databaseReference.child(getRef(holder.getAbsoluteAdapterPosition()).getKey())
                    .removeValue().addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull
Task<Void> task) {
                            if(task.isSuccessful()){
                                Toast.makeText(Cart.this,
                                "Item has been removed", Toast.LENGTH_LONG).show();
                            }
                        }
                    });
                builder.setNegativeButton("No", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface
dialogInterface, int i) {
                        //do nothing
                    }
                });
                builder.show();
            }
        } );
    }
} );

@NonNull
@Override
public OrderViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int

```

```

viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.cart_view, parent, false);
        return new OrderViewHolder(view);
    }
    adapter.startListening();
recyclerView.setAdapter(adapter);
}
}

```

## OrderObject Class

```

package com.example.wagbaapplication;

public class OrderObject {
    private String DishName;
    private String Quantity;
    private String Price;
    private String Image;

    public OrderObject(String dishName, String quantity, String price, String image)
{
    DishName = dishName;
    Quantity = quantity;
    Price = price;
    Image = image;
}

    public OrderObject() {
}

    public String getDishName() {
        return DishName;
    }

    public void setDishName(String dishName) {
        DishName = dishName;
    }

    public String getQuantity() {
        return Quantity;
    }

    public void setQuantity(String quantity) {
        Quantity = quantity;
    }

    public String getPrice() {
        return Price;
    }

    public void setPrice(String price) {
        Price = price;
    }

    public String getImage() {
        return Image;
    }
}

```

```
    public void setImage(String image) {
        Image = image;
    }
}
```

## OrderViewHolder Class

```
package com.example.wagbaapplication;

import android.view.View;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class OrderViewHolder extends RecyclerView.ViewHolder {

    ImageView image, minus, plus;
    TextView name, price, quantity, totalPrice;
    ImageButton delete;

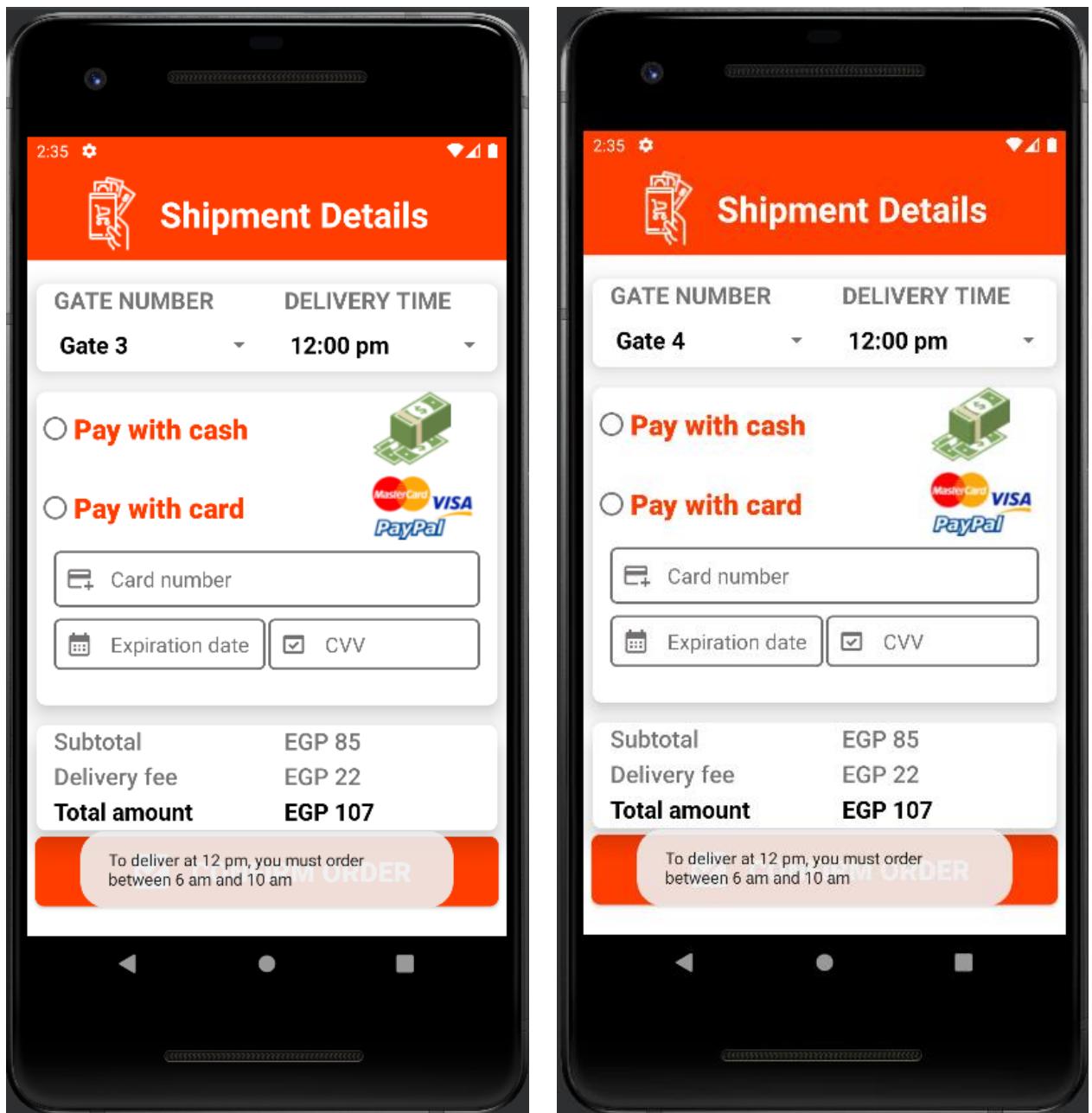
    public OrderViewHolder(@NonNull View itemView) {
        super(itemView);

        image = itemView.findViewById(R.id.orderImage);
        name = itemView.findViewById(R.id.orderName);
        price = itemView.findViewById(R.id.orderPrice);
        quantity = itemView.findViewById(R.id.orderQuantity);
        totalPrice = itemView.findViewById(R.id.orderTotal);
        minus = itemView.findViewById(R.id.minus2);
        plus = itemView.findViewById(R.id.plus2);
        delete = itemView.findViewById(R.id.deleteButton);
    }
}
```

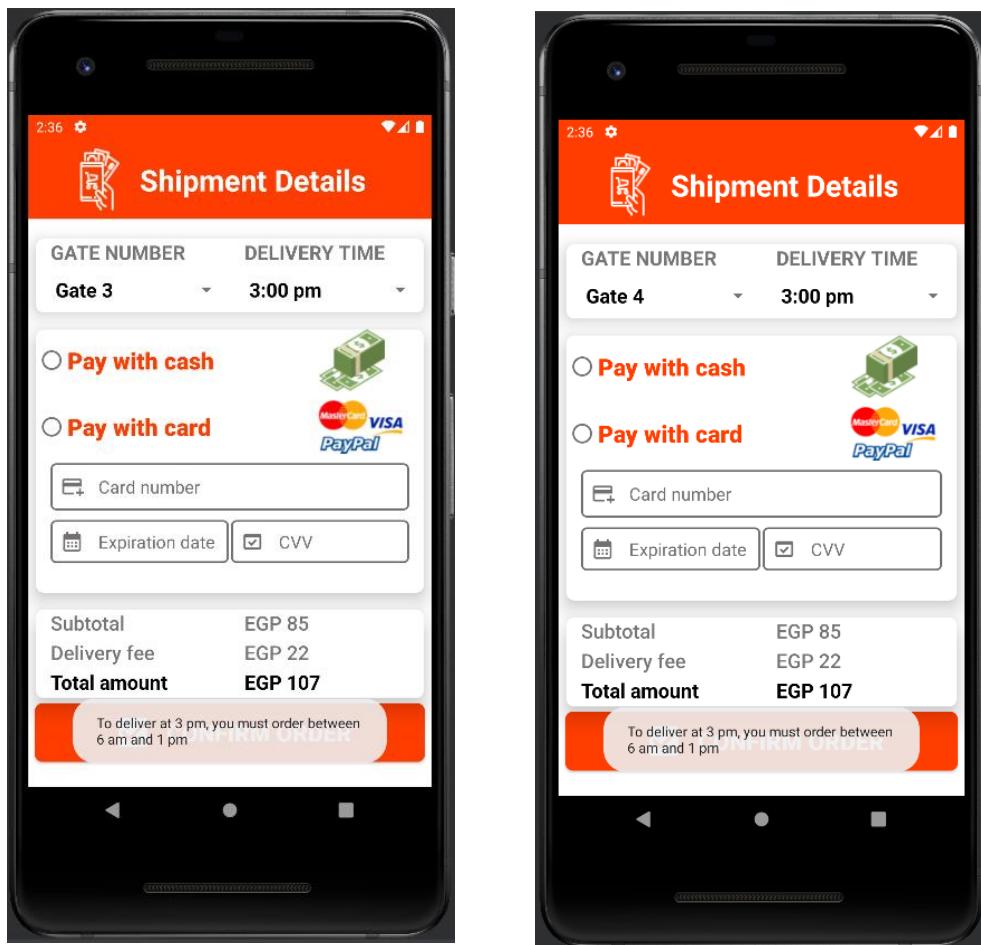
## VIII. CONFIRM ORDER

### ➤ Test Cases

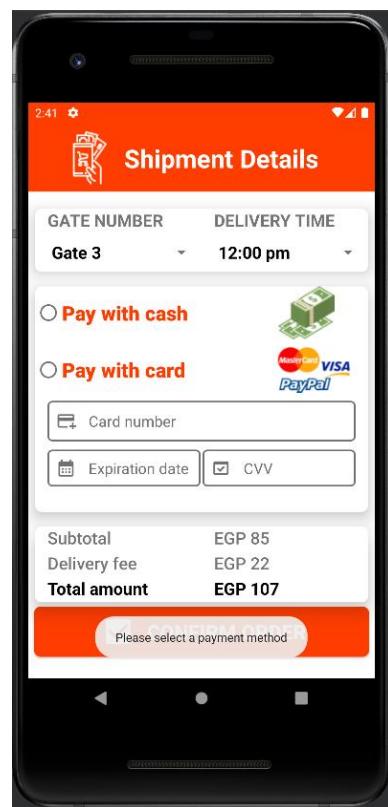
1. Clicking confirm order button with delivery time set at 12 pm but your current time is not between 6 am and 10 am



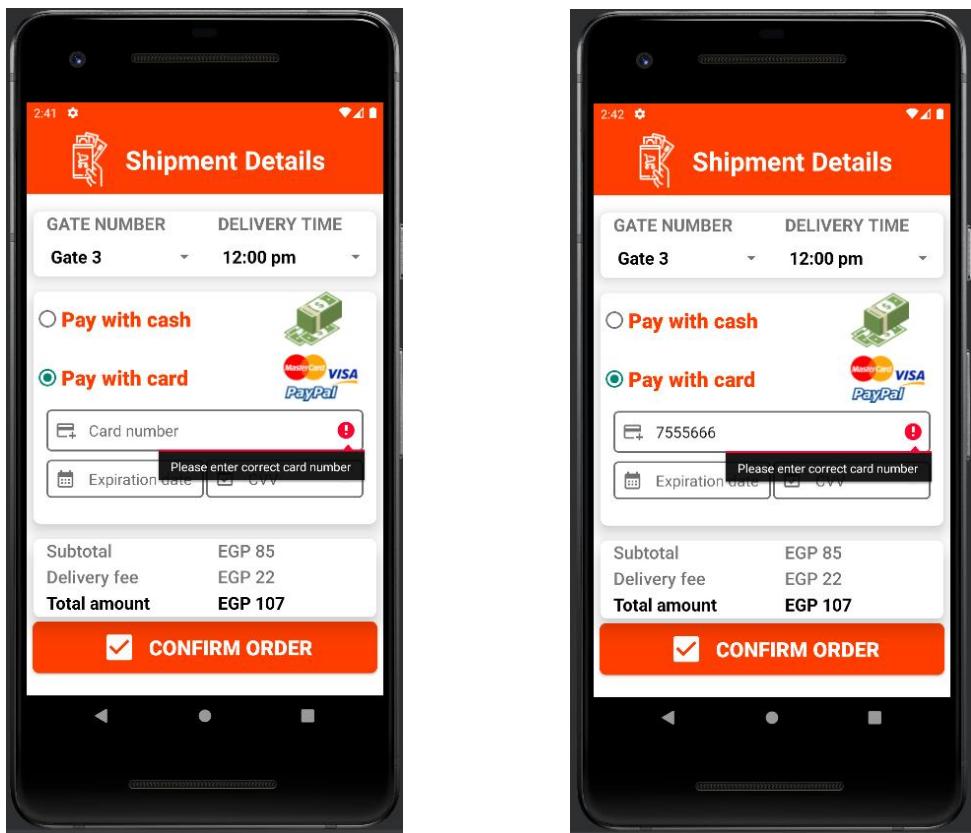
- Clicking confirm order button with delivery time set at 3 pm but your current time is not between 6 am and 1 pm



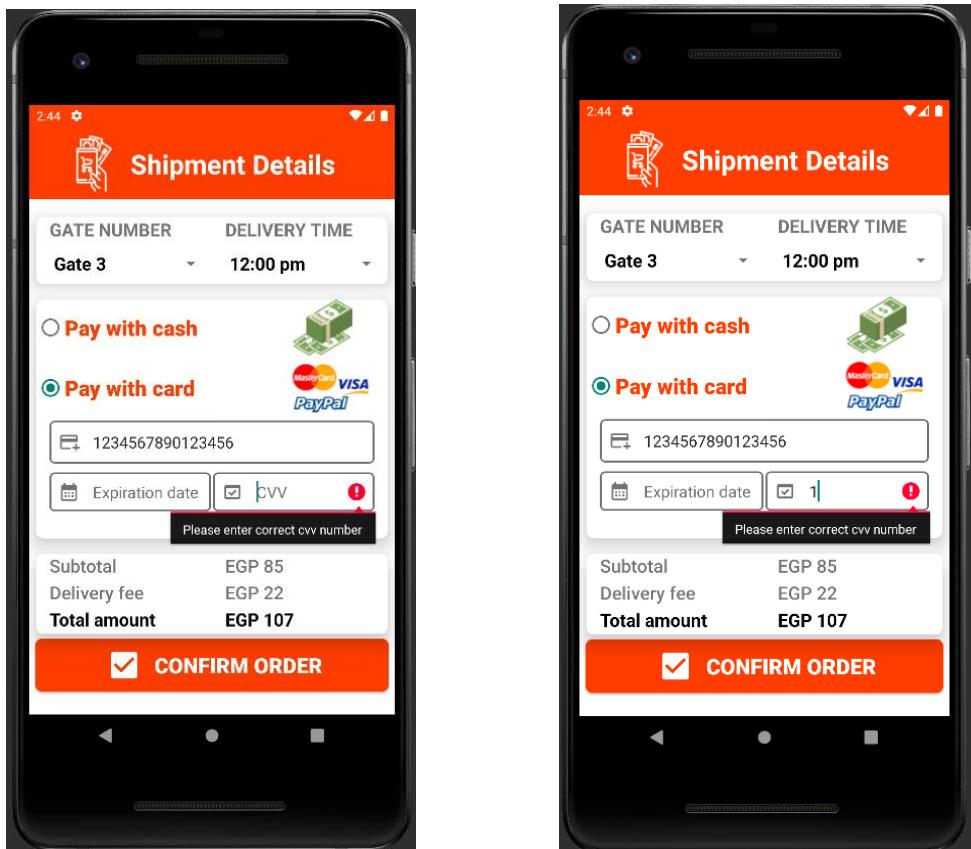
3. Clicking confirm order button without picking payment method



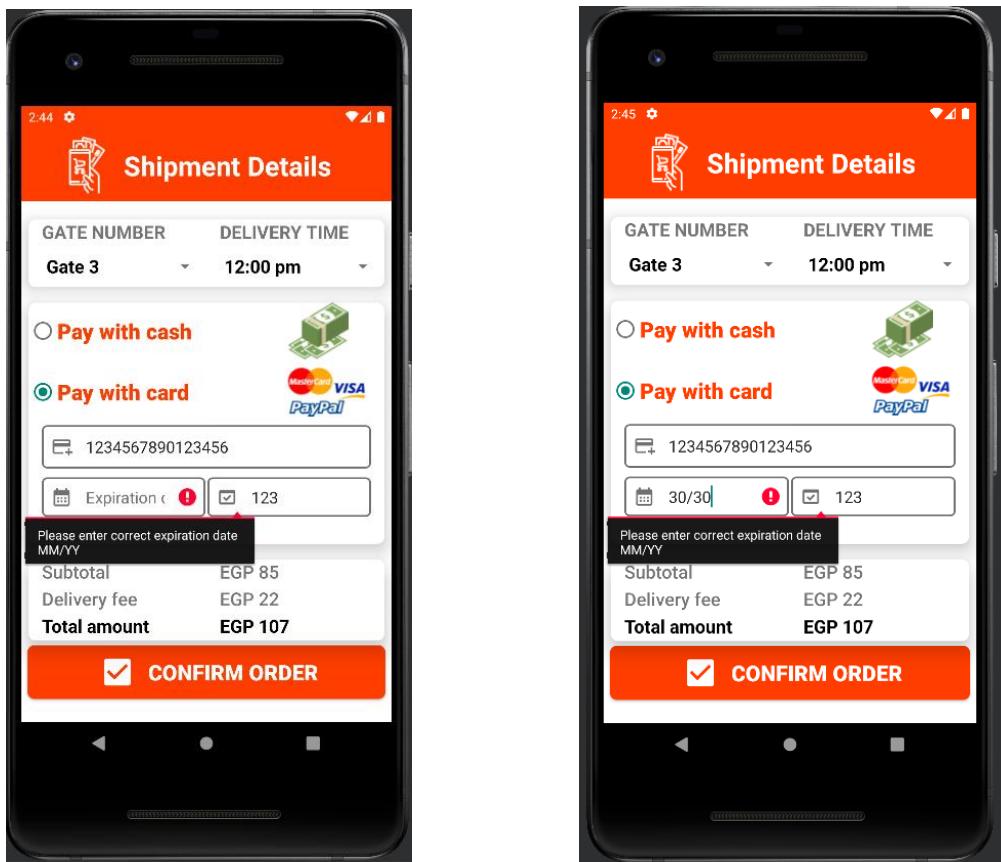
4. Clicking confirm order button with picking card payment method but without filling card number or filling it wrong



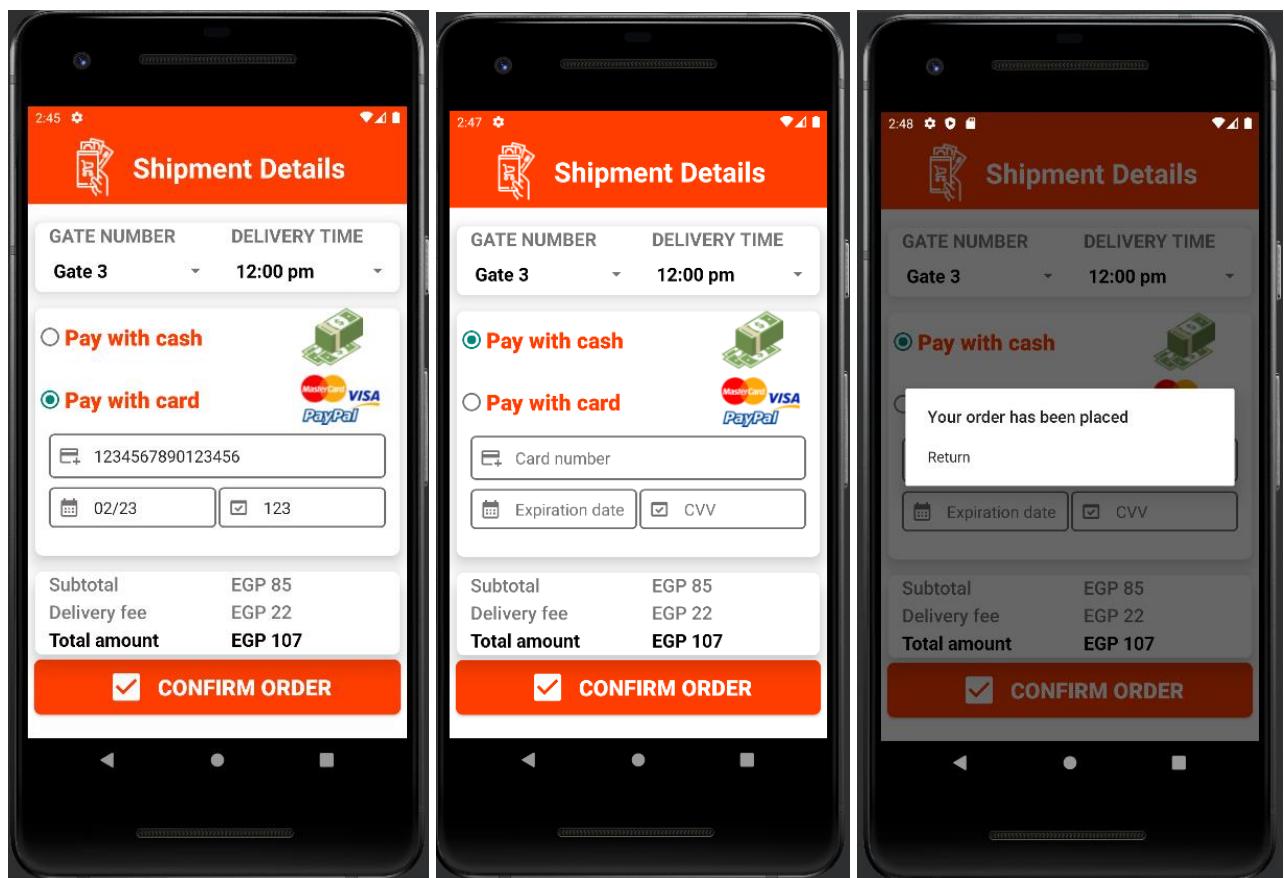
- Clicking confirm order button with picking card payment method but without filling CVV number or filling it wrong



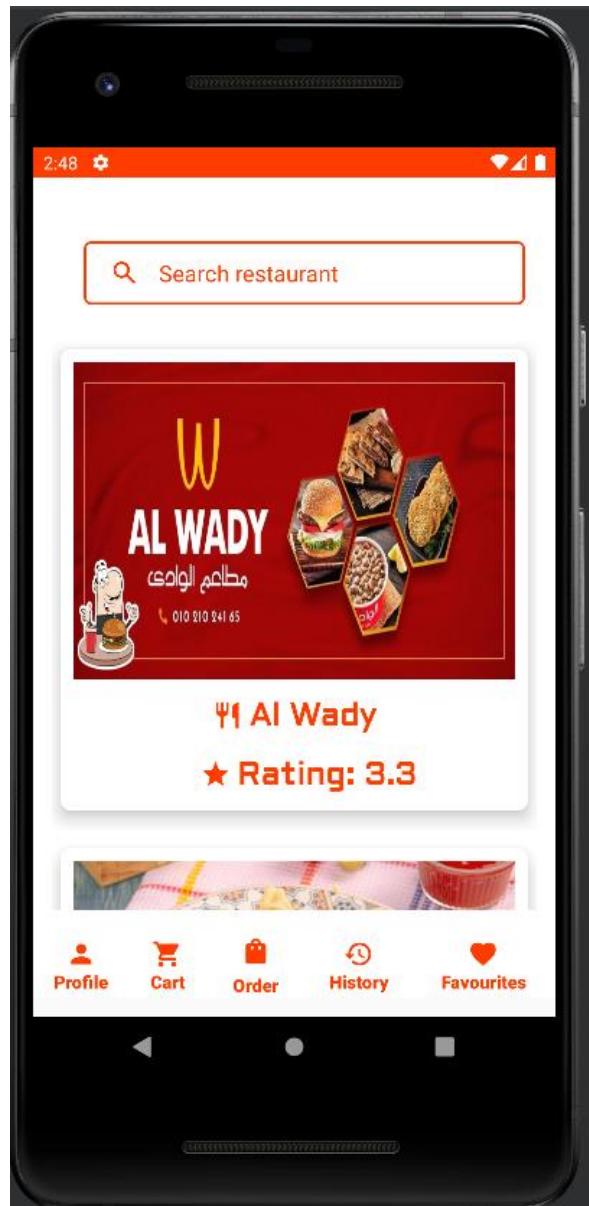
- Clicking confirm order button with picking card payment method but without filling expiration date or filling it wrong



- Clicking confirm order button with everything correct



- Clicking confirm order button then chose return



## ➤ Code

### ConfirmOrder Activity

```
package com.example.wagbaapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
```

```

import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.format.DateTimeFormatter;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;

public class ConfirmOrder extends AppCompatActivity {

    Spinner gateNumber, deliveryTime;
    RadioGroup paymentMethod;
    RadioButton paymentCash;
    RadioButton paymentCard;
    String selectedRadioButton;
    EditText cardNumber, cardExpirationDate, cardCVV;
    TextView subTotal, deliveryFee, orderTotalPrice;
    Button confirmOrder;
    DatabaseReference databaseReference;
    final int deliveryCost = 22;
    String expirationDatePattern = "^(0[1-9]|1[0-2])\\/(0[1-9]|1[0-2])$";
    String gate, timeToDeliver, orderTime;
    int Total;
    long maxId = 0;
    int CartTotal = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.confirm_order);

        databaseReference = FirebaseDatabase.getInstance().getReference();
        paymentMethod = (RadioGroup) findViewById(R.id.radioGroup);
        paymentCash = (RadioButton) findViewById(R.id.payCash);
        paymentCard = (RadioButton) findViewById(R.id.payCard);
        cardNumber = (EditText) findViewById(R.id.cardNumber);
        cardExpirationDate = (EditText) findViewById(R.id.cardExpirationDate);
        cardCVV = (EditText) findViewById(R.id.cardCVV);
        subTotal = (TextView) findViewById(R.id.subTotal);
        deliveryFee = (TextView) findViewById(R.id.deliveryFee);
        orderTotalPrice = (TextView) findViewById(R.id.orderTotalPrice);
        confirmOrder = (Button) findViewById(R.id.confirmOrderButton);

        // Read from the database

        databaseReference.child("CartList").child(FirebaseAuth.getInstance().getCurrentUser()
                .getUid())
                .child("Orders").addValueEventListener(new ValueEventListener() {
                    @Override

```

```

        public void onDataChange(DataSnapshot dataSnapshot) {
            for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
                String priceString =
String.valueOf(snapshot.child("Price").getValue());
                String quantityString =
String.valueOf(snapshot.child("Quantity").getValue());
                CartTotal = CartTotal
+(Integer.valueOf(priceString)*Integer.valueOf(quantityString));

            }
            subTotal.setText("EGP "+ String.valueOf(CartTotal));
            orderTotalPrice.setText("EGP "+
String.valueOf(deliveryCost+CartTotal));
        }
    }

    gateNumber = findViewById(R.id.gateNumber);
    ArrayAdapter<CharSequence> adapter1 = ArrayAdapter.createFromResource(this,
R.array.gate, R.layout.spinner_selected_item);

adapter1.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    gateNumber.setAdapter(adapter1);

    deliveryTime = findViewById(R.id.deliveryTime);
    ArrayAdapter<CharSequence> adapter2 = ArrayAdapter.createFromResource(this,
R.array.timeDelivery, R.layout.spinner_selected_item);

adapter2.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    deliveryTime.setAdapter(adapter2);

    // Read from the database
    databaseReference.child("AllOrders").addValueEventListener(new
ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if(dataSnapshot.exists()){
                maxId=(dataSnapshot.getChildrenCount());
            }
        }
        @Override
        public void onCancelled(DatabaseError error) {
            // Failed to read value
        }
    });
}

confirmOrder.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        CheckOrderConfirmation();
    }
});

cardNumber.setEnabled(false);
cardExpirationDate.setEnabled(false);

```

```

        cardCVV.setEnabled(false);

        paymentCard.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                cardNumber.setEnabled(true);
                cardExpirationDate.setEnabled(true);
                cardCVV.setEnabled(true);
            }
        });

        paymentCash.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                cardNumber.setEnabled(false);
                cardExpirationDate.setEnabled(false);
                cardCVV.setEnabled(false);
            }
        });
    }

    private void CheckOrderConfirmation() {
        gate = gateNumber.getSelectedItem().toString();
        timeToDeliver = deliveryTime.getSelectedItem().toString();

        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("dd-M-yyyy
hh:mm:ss");
        orderTime = simpleDateFormat.format(new Date());

        //Here we get the current time in the 24 hours format, while HH:mm a gives
you format of 12:00 pm
        SimpleDateFormat simpleDateFormat2 = new SimpleDateFormat("Hm");
        String currentTime = simpleDateFormat2.format(new Date());
        String time1 = "600";
        String time2 = "1000";
        String time3 = "1300";

        String testTime1 = "100";
        String testTime2 = "500";

        if((timeToDeliver.equals("12:00 pm")) && (Integer.valueOf(currentTime) <
Integer.valueOf(time1) || Integer.valueOf(currentTime) > Integer.valueOf(time2))){
            Toast.makeText(this,"To deliver at 12 pm, you must order between 6 am and
10 am", Toast.LENGTH_LONG).show();
        }else if((timeToDeliver.equals("3:00 pm")) && (Integer.valueOf(currentTime) <
Integer.valueOf(time1) || Integer.valueOf(currentTime) > Integer.valueOf(time3))){
            Toast.makeText(this,"To deliver at 3 pm, you must order between 6 am and
1 pm", Toast.LENGTH_LONG).show();
        }else if(paymentMethod.getCheckedRadioButtonId() == -1){
            paymentMethod.requestFocus();
            Toast.makeText(this,"Please select a payment method",
Toast.LENGTH_LONG).show();
        }else if(paymentCard.isChecked() &&
(cardNumber.getText().toString().length()<16)){
            cardNumber.setError("Please enter correct card number");
            cardNumber.requestFocus();
        }else if(paymentCard.isChecked() &&
(cardCVV.getText().toString().length()<3)){
    }
}

```

```

        cardCVV.setError("Please enter correct cvv number");
        cardCVV.requestFocus();
    }else if(paymentCard.isChecked() &&
(!cardExpirationDate.getText().toString().matches(expirationDatePattern))){
        cardExpirationDate.setError("Please enter correct expiration date
MM/YY");
        cardExpirationDate.requestFocus();
    }else{ //now all the validations are over, here we will add data to firebase
        addDataToAllOrders();
    }

}

private void addDataToAllOrders() {
    maxId += 1;
    copyOrders();

    if(paymentCard.isChecked()){
        selectedRadioButton = "Payment with card";
    }else if(paymentCash.isChecked()){
        selectedRadioButton = "Payment with cash";
    }

    final HashMap<String, Object> orderMap = new HashMap<>();
    orderMap.put("OrderUserId",
    FirebaseAuth.getInstance().getCurrentUser().getUid());
    orderMap.put("GateNumber", gate);
    orderMap.put("DeliveryTime", timeToDeliver);
    orderMap.put("PaymentMethod", selectedRadioButton);
    orderMap.put("Status", "Placed");
    orderMap.put("OrderTime", orderTime);
    orderMap.put("TotalAmount", String.valueOf(deliveryCost+CartTotal));

    databaseReference.child("AllOrders").
        child(String.valueOf(maxId))
        .updateChildren(orderMap).addOnCompleteListener(new
OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {

databaseReference.child("CartList").child(FirebaseAuth.getInstance().getCurrentUser()
.getUid())
        .child("Orders").removeValue();

        CharSequence options[] = new CharSequence[]
        {
            "Return"
        };
        AlertDialog.Builder builder = new
AlertDialog.Builder(ConfirmOrder.this);
        builder.setTitle("Your order has been placed");

        builder.setItems(options, new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int
i) {
                if(i==0){
                    Intent intent = new Intent(ConfirmOrder.this,
                    Restaurants.class);
                    intent.putExtra("OrderId", maxId);
                }
            }
        });
    }
}

```

```

                startActivity(intent);
                finish();
            }
        }
    );
    builder.show();
}
}

private void copyOrders() {
    DatabaseReference cartListOrderNode =
databaseReference.child("CartList").child(FirebaseAuth.getInstance().getCurrentUser()
.getUid())
    .child("Orders");
    DatabaseReference AllOrdersContentNode =
databaseReference.child("AllOrdersContent").child(String.valueOf(maxId));
    // Read from the database
    cartListOrderNode.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            for(DataSnapshot cartCode: dataSnapshot.getChildren()) {
                String cartCodeKey = cartCode.getKey();
                String DishName =
cartCode.child("DishName").getValue(String.class);
                String Image = cartCode.child("Image").getValue(String.class);
                String Price = cartCode.child("Price").getValue(String.class);
                String Quantity =
cartCode.child("Quantity").getValue(String.class);

                AllOrdersContentNode.child(cartCodeKey).child("DishName").setValue(DishName);
                AllOrdersContentNode.child(cartCodeKey).child("Image").setValue(Image);
                AllOrdersContentNode.child(cartCodeKey).child("Price").setValue(Price);
                AllOrdersContentNode.child(cartCodeKey).child("Quantity").setValue(Quantity);
            }
        }
        @Override
        public void onCancelled(DatabaseError error) {
            // Failed to read value
        }
    });
}
}

```

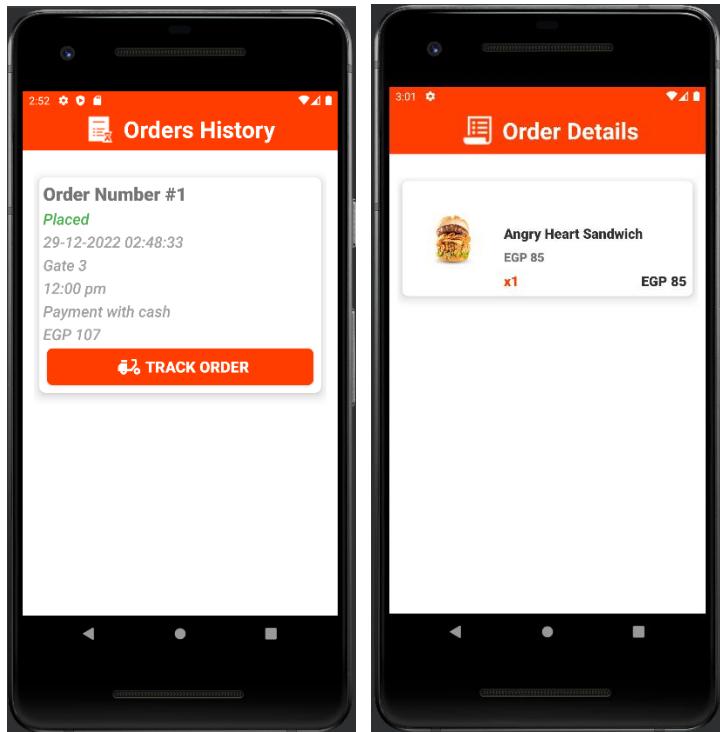
## ➤ Firebase

```
- AllOrders
  - 1
    - DeliveryTime: "12:00 pm" ✖
    - GateNumber: "Gate 3"
    - OrderTime: "29-12-2022 02:48:33"
    - OrderUserId: "e31VogWsqaZde6uTsfA9kZXEjKO2"
    - PaymentMethod: "Payment with cash"
    - Status: "Placed"
    - TotalAmount: "107"
  - AllOrdersContent
    - 1
      - 12
        - DishName: "Angry Heart Sandwich"
        - Image: "https://s3-eu-west-1.amazonaws.com/elmenusv5-stg/Normal/8a088238-4d7c-4e10-98fa-b9a156ef7e22.jpg"
        - Price: "85"
        - Quantity: "1"
```

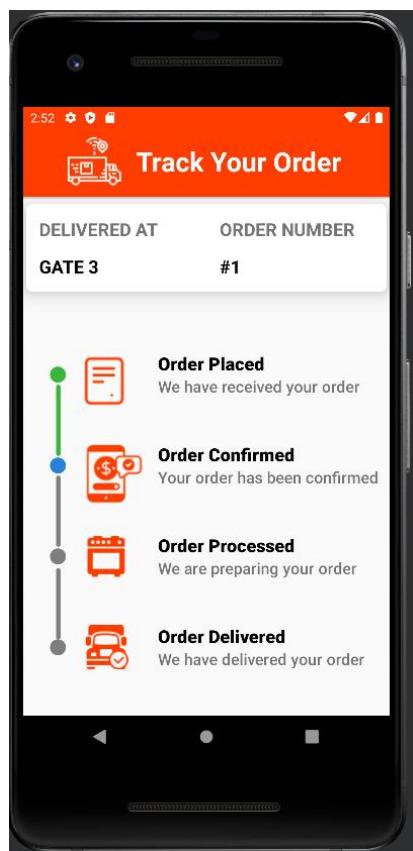
## IX. ORDER HISTORY & ORDER HISTORY CONTENT

### Test Cases

- Navigating through history list and clicking on history view item



- Clicking on track order button



## ➤ Code

### OrderHistory Activity

```
package com.example.wagbaapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.firebaseio.ui.database.FirebaseRecyclerAdapter;
import com.firebaseio.ui.database.FirebaseRecyclerOptions;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

public class OrderHistory extends AppCompatActivity {

    RecyclerView recyclerView;
    TextView noItems;
    FirebaseRecyclerOptions<OrderHistoryObject> options;
    FirebaseRecyclerAdapter<OrderHistoryObject, OrderHistoryViewHolder> adapter;
    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_order_history);

        noItems = (TextView) findViewById(R.id.noItems);
        recyclerView = (RecyclerView) findViewById(R.id.recyclerView);
        LinearLayoutManager linearLayoutManager = new
        LinearLayoutManager(getApplicationContext());
        linearLayoutManager.setReverseLayout(true);
        linearLayoutManager.setStackFromEnd(true);
        recyclerView.setLayoutManager(linearLayoutManager);
        recyclerView.setHasFixedSize(true);
        databaseReference =
        FirebaseDatabase.getInstance().getReference().child("AllOrders");

        LoadHistory();
        CheckIfHistoryIsEmpty();

    }

    private void CheckIfHistoryIsEmpty() {
        databaseReference.orderByChild("OrderId")
    }
}
```

```

        .equalTo(FirebaseAuth.getInstance().getCurrentUser().getUid()).addValueEventListener(
    new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if(!dataSnapshot.exists()){
                noItems.setVisibility(View.VISIBLE);
            }
        }
        @Override
        public void onCancelled(DatabaseError error) {
        }
    });
}

private void LoadHistory() {
    options = new FirebaseRecyclerOptions.Builder<OrderHistoryObject>()
        .setQuery(databaseReference.orderByChild("OrderUserId")
        .equalTo(FirebaseAuth.getInstance().getCurrentUser().getUid()),
    OrderHistoryObject.class).build();

    adapter = new FirebaseRecyclerAdapter<OrderHistoryObject,
    OrderHistoryViewHolder>(options) {
        @Override
        protected void onBindViewHolder(@NonNull OrderHistoryViewHolder holder,
int position, @NonNull OrderHistoryObject model) {

            holder.orderId.setText("Order Number #" +
getRef(holder.getAbsoluteAdapterPosition()).getKey());
            holder.status.setText(model.getStatus());
            holder.orderTime.setText(model.getOrderTime());
            holder.gateNumber.setText(model.getGateNumber());
            holder.deliveryTime.setText(model.getDeliveryTime());
            holder.paymentMethod.setText(model.getPaymentMethod());
            holder.totalAmount.setText("EGP " + model.getTotalAmount());

            holder.itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Intent intent = new Intent(OrderHistory.this,
OrderHistoryContent.class);

                    intent.putExtra("OrderId", getRef(holder.getAbsoluteAdapterPosition()).getKey());
                    startActivity(intent);
                }
            });

            holder.trackOrder.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Intent intent = new Intent(OrderHistory.this,
OrderStatus.class);

                    intent.putExtra("OrderId", getRef(holder.getAbsoluteAdapterPosition()).getKey());
                    intent.putExtra("OrderGate", model.getGateNumber());
                    startActivity(intent);
                }
            });
        }
    };
}

```

```

        @Override
        public OrderHistoryViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
            View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.history_view, parent,
false);
            return new OrderHistoryViewHolder(view);
        }
    };
    adapter.startListening();
    recyclerView.setAdapter(adapter);
}
}

```

## OrderHistoryObject Class

```

package com.example.wagbaapplication;

public class OrderHistoryObject {
    String DeliveryTime;
    String GateNumber;
    String OrderTime;
    String OrderUserId;
    String Status;
    String TotalAmount;
    String PaymentMethod;

    public OrderHistoryObject(String deliveryTime, String gateNumber, String
orderTime, String orderUserId, String status, String totalAmount, String
paymentMethod) {
        DeliveryTime = deliveryTime;
        GateNumber = gateNumber;
        OrderTime = orderTime;
        OrderUserId = orderUserId;
        Status = status;
        TotalAmount = totalAmount;
        PaymentMethod = paymentMethod;
    }

    public OrderHistoryObject() {
    }

    public String getDeliveryTime() {
        return DeliveryTime;
    }

    public void setDeliveryTime(String deliveryTime) {
        DeliveryTime = deliveryTime;
    }

    public String getGateNumber() {
        return GateNumber;
    }

    public void setGateNumber(String gateNumber) {
        GateNumber = gateNumber;
    }
}

```

```

public String getOrderTime() {
    return OrderTime;
}

public void setOrderTime(String orderTime) {
    OrderTime = orderTime;
}

public String getOrderUserId() {
    return OrderUserId;
}

public void setOrderUserId(String orderUserId) {
    OrderUserId = orderUserId;
}

public String getStatus() {
    return Status;
}

public void setStatus(String status) {
    Status = status;
}

public String getTotalAmount() {
    return TotalAmount;
}

public void setTotalAmount(String totalAmount) {
    TotalAmount = totalAmount;
}

public String getPaymentMethod() {
    return PaymentMethod;
}

public void setPaymentMethod(String paymentMethod) {
    PaymentMethod = paymentMethod;
}
}

```

## OrderHistoryViewHolder Class

```

package com.example.wagbaapplication;

import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class OrderHistoryViewHolder extends RecyclerView.ViewHolder{

    TextView orderId, deliveryTime, gateNumber, orderTime, status, totalAmount,
    paymentMethod;
    Button trackOrder;

    public OrderHistoryViewHolder(@NonNull View itemView) {
        super(itemView);
    }
}

```

```

        orderId = itemView.findViewById(R.id.orderHistoryNumber);
        deliveryTime = itemView.findViewById(R.id.orderHistoryTime);
        gateNumber = itemView.findViewById(R.id.orderHistoryGate);
        orderTime = itemView.findViewById(R.id.orderHistoryDate);
        status = itemView.findViewById(R.id.orderHistoryStatus);
        totalAmount = itemView.findViewById(R.id.orderHistoryTotal);
        paymentMethod = itemView.findViewById(R.id.orderHistoryPaymentMethod);
        trackOrder = itemView.findViewById(R.id.trackOrderButton);
    }
}

```

## OrderHistoryContent Activity

```

package com.example.wagbaapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;

import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebaseio.ui.database.FirebaseRecyclerOptions;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.squareup.picasso.Picasso;

public class OrderHistoryContent extends AppCompatActivity {

    RecyclerView recyclerView;
    FirebaseRecyclerOptions<OrderObject> options;
    FirebaseRecyclerAdapter<OrderObject, OrderHistoryContentViewHolder> adapter;
    DatabaseReference databaseReference;
    String OrderId = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_order_history_content);

        recyclerView = (RecyclerView) findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(getApplicationContext()));
        recyclerView.setHasFixedSize(true);
        databaseReference =
                FirebaseDatabase.getInstance().getReference().child("AllOrdersContent");

        //Getting the intent from order history page
        if (getIntent() != null)
            OrderId = getIntent().getStringExtra("OrderId");

        if (!OrderId.isEmpty() && OrderId != null)
            LoadOrderHistoryDetails(OrderId);
    }
}

```

```

    }

    private void LoadOrderHistoryDetails(String orderId) {
        options = new
        FirebaseRecyclerOptions.Builder<OrderObject>().setQuery(databaseReference.child(order
        Id), OrderObject.class).build();
        adapter = new FirebaseRecyclerAdapter<OrderObject,
        OrderHistoryContentViewHolder>(options) {
            @Override
            protected void onBindViewHolder(@NonNull OrderHistoryContentViewHolder
            holder, int position, @NonNull OrderObject model) {

                holder.name.setText(model.getDishName());
                holder.price.setText("EGP "+model.getPrice());
                holder.quantity.setText("x"+model.getQuantity());
                Picasso.get().load(model.getImage()).into(holder.image);
                holder.totalPrice.setText("EGP
"+String.valueOf(Integer.valueOf(model.getPrice())*Integer.valueOf(model.getQuantity(
))))));
            }

            @NonNull
            @Override
            public OrderHistoryContentViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
                View view =
                LayoutInflater.from(parent.getContext()).inflate(R.layout.history_content_view, parent
                ,false);
                return new OrderHistoryContentViewHolder(view);
            }
        };
        adapter.startListening();
        recyclerView.setAdapter(adapter);
    }
}

```

## OrderHistoryContentViewHolder Class

```

package com.example.wagbaapplication;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
public class OrderHistoryContentViewHolder extends RecyclerView.ViewHolder{
    ImageView image;
    TextView name, price, quantity, totalPrice;
    public OrderHistoryContentViewHolder(@NonNull View itemView) {
        super(itemView);

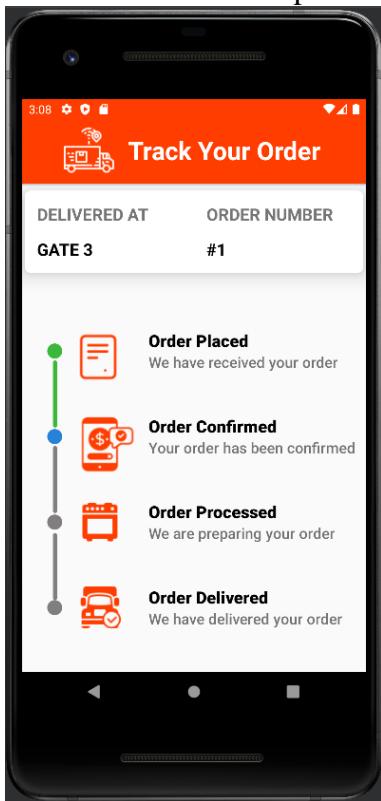
        image = itemView.findViewById(R.id.orderHistoryContentImage);
        name = itemView.findViewById(R.id.orderHistoryContentName);
        price = itemView.findViewById(R.id.orderHistoryContentPrice);
        quantity = itemView.findViewById(R.id.orderHistoryContentQuantity);
        totalPrice = itemView.findViewById(R.id.orderHistoryContentTotal);
    }
}

```

## X. ORDER STATUS

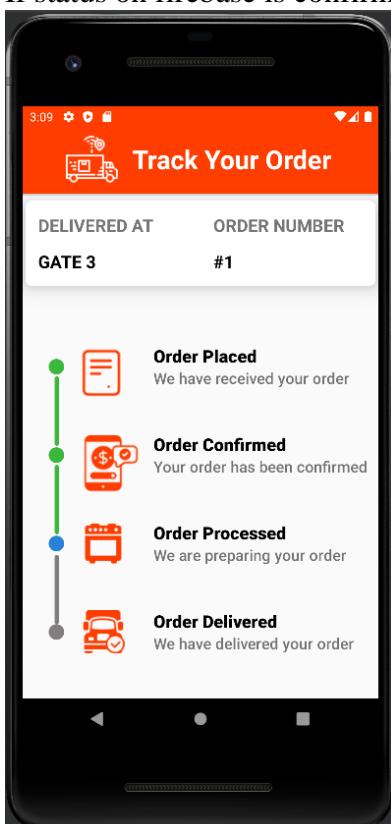
### Test Cases

1. If status on firebase is placed



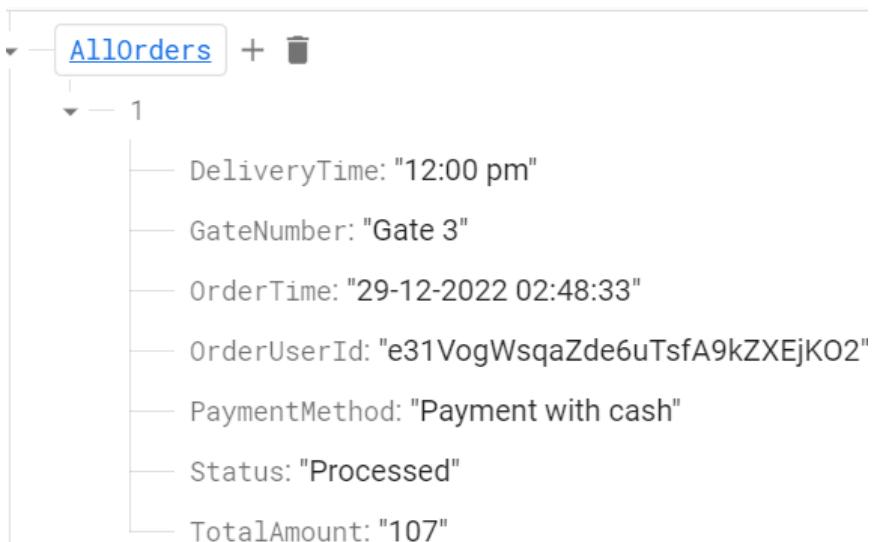
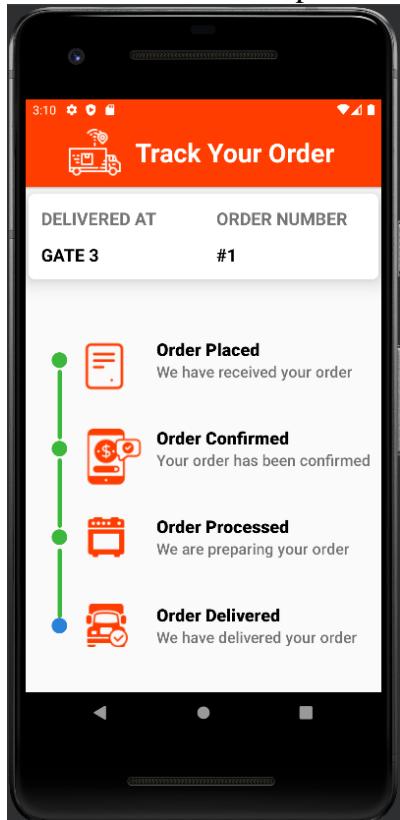
```
  AllOrders
    1
      DeliveryTime: "12:00 pm"
      GateNumber: "Gate 3"
      OrderTime: "29-12-2022 02:48:33"
      OrderUserId: "e31VogWsqaZde6uTsfA9kZXEjKO2"
      PaymentMethod: "Payment with cash"
      Status: "Placed"
      TotalAmount: "107"
```

2. If status on firebase is confirmed

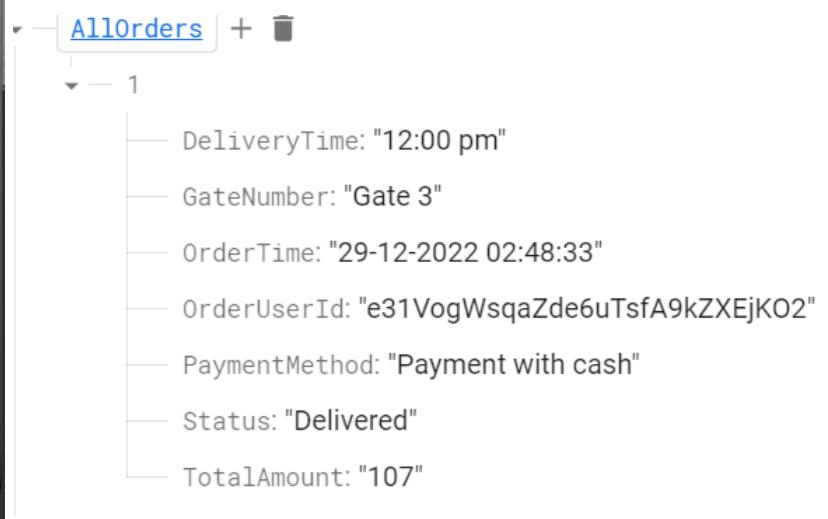
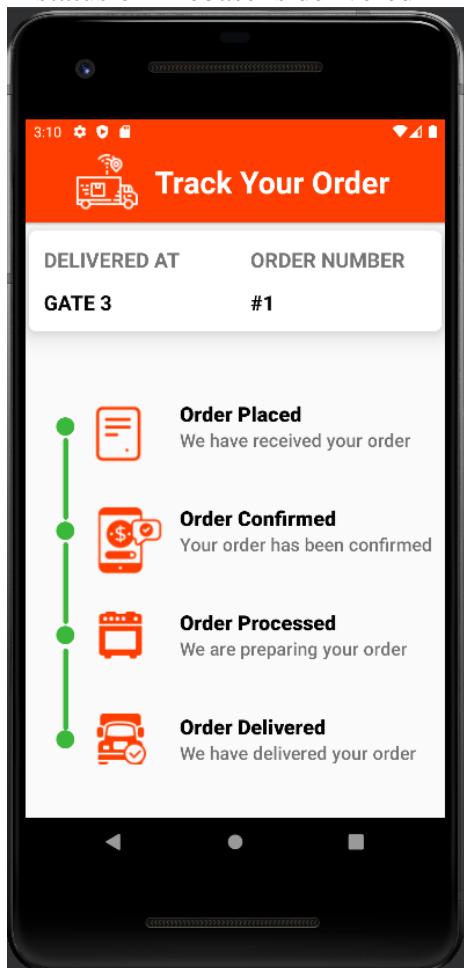


```
  AllOrders
    1
      DeliveryTime: "12:00 pm"
      GateNumber: "Gate 3"
      OrderTime: "29-12-2022 02:48:33"
      OrderUserId: "e31VogWsqaZde6uTsfA9kZXEjKO2"
      PaymentMethod: "Payment with cash"
      Status: "Confirmed"
      TotalAmount: "107"
```

3. If status on firebase is processed



4. If status on firebase is delivered



## Code

### OrderStatus Activity

```
package com.example.wagbaapplication;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

public class OrderStatus extends AppCompatActivity {

    TextView orderGateNumber, orderNumber;
    TextView orderConfirmed, orderConfirmed2, orderProcessed, orderProcessed2,
orderDelivered;
    DatabaseReference databaseReference;
    String OrderId = "";
    String OrderGate = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_order_status);

        orderGateNumber = (TextView) findViewById(R.id.orderGate);
        orderNumber = (TextView) findViewById(R.id.orderNumber);
        orderConfirmed = (TextView) findViewById(R.id.orderConfirmed);
        orderConfirmed2 = (TextView) findViewById(R.id.orderConfirmed2);
        orderProcessed = (TextView) findViewById(R.id.orderProcessed);
        orderProcessed2 = (TextView) findViewById(R.id.orderProcessed2);
        orderDelivered = (TextView) findViewById(R.id.orderDelivered);
        databaseReference =
        FirebaseDatabase.getInstance().getReference().child("AllOrders");

        //Getting the intent from order history page
        if (getIntent() != null){
            OrderId = getIntent().getStringExtra("OrderId");
            OrderGate = getIntent().getStringExtra("OrderGate");
        }

        if (!OrderId.isEmpty() && OrderId != null)
            LoadOrderTrackingDetails(OrderId, OrderGate);

    }

    private void LoadOrderTrackingDetails(String orderId, String orderGate) {

        orderGateNumber.setText(orderGate);
        orderNumber.setText("#" + orderId);

        // Read from the database
        databaseReference.child(orderId).addValueEventListener(new
ValueEventListener() {

            @Override
```

```

        public void onDataChange(DataSnapshot dataSnapshot) {

if(dataSnapshot.child("Status").getValue().toString().equals("Placed")){
    orderConfirmed.setBackground(getResources().getDrawable(R.drawable.shape_status_current));
    orderConfirmed2.setBackground(getResources().getDrawable(R.drawable.shape_bar_remaining));
    orderProcessed.setBackground(getResources().getDrawable(R.drawable.shape_status_remaining));
    orderProcessed2.setBackground(getResources().getDrawable(R.drawable.shape_bar_remaining));
    orderDelivered.setBackground(getResources().getDrawable(R.drawable.shape_status_remaining));
} else if(dataSnapshot.child("Status").getValue().toString().equals("Confirmed")){
    orderConfirmed.setBackground(getResources().getDrawable(R.drawable.shape_status_completed));
    orderConfirmed2.setBackground(getResources().getDrawable(R.drawable.shape_bar_completed));
    orderProcessed.setBackground(getResources().getDrawable(R.drawable.shape_status_current));
    orderProcessed2.setBackground(getResources().getDrawable(R.drawable.shape_bar_remaining));
    orderDelivered.setBackground(getResources().getDrawable(R.drawable.shape_status_remaining));
} else if(dataSnapshot.child("Status").getValue().toString().equals("Processed")){
    orderConfirmed.setBackground(getResources().getDrawable(R.drawable.shape_status_completed));
    orderConfirmed2.setBackground(getResources().getDrawable(R.drawable.shape_bar_completed));
    orderProcessed.setBackground(getResources().getDrawable(R.drawable.shape_status_completed));
    orderProcessed2.setBackground(getResources().getDrawable(R.drawable.shape_bar_completed));
    orderDelivered.setBackground(getResources().getDrawable(R.drawable.shape_status_current));
} else if(dataSnapshot.child("Status").getValue().toString().equals("Delivered")){
    orderConfirmed.setBackground(getResources().getDrawable(R.drawable.shape_status_completed));
    orderConfirmed2.setBackground(getResources().getDrawable(R.drawable.shape_bar_completed));
}
}

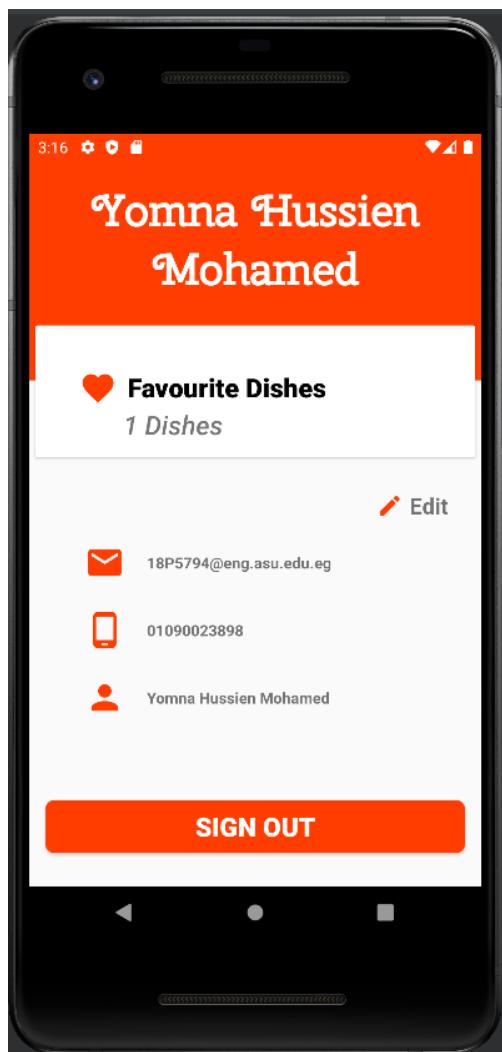
```

```
ed) );  
  
orderProcessed.setBackground(getResources().getDrawable(R.drawable.shape_status_compl  
eted));  
  
orderProcessed2.setBackground(getResources().getDrawable(R.drawable.shape_bar_complet  
ed));  
  
orderDelivered.setBackground(getResources().getDrawable(R.drawable.shape_status_compl  
eted));  
        }  
    }  
    @Override  
    public void onCancelled(DatabaseError error) {  
        // Failed to read value  
    }  
});  
}  
}
```

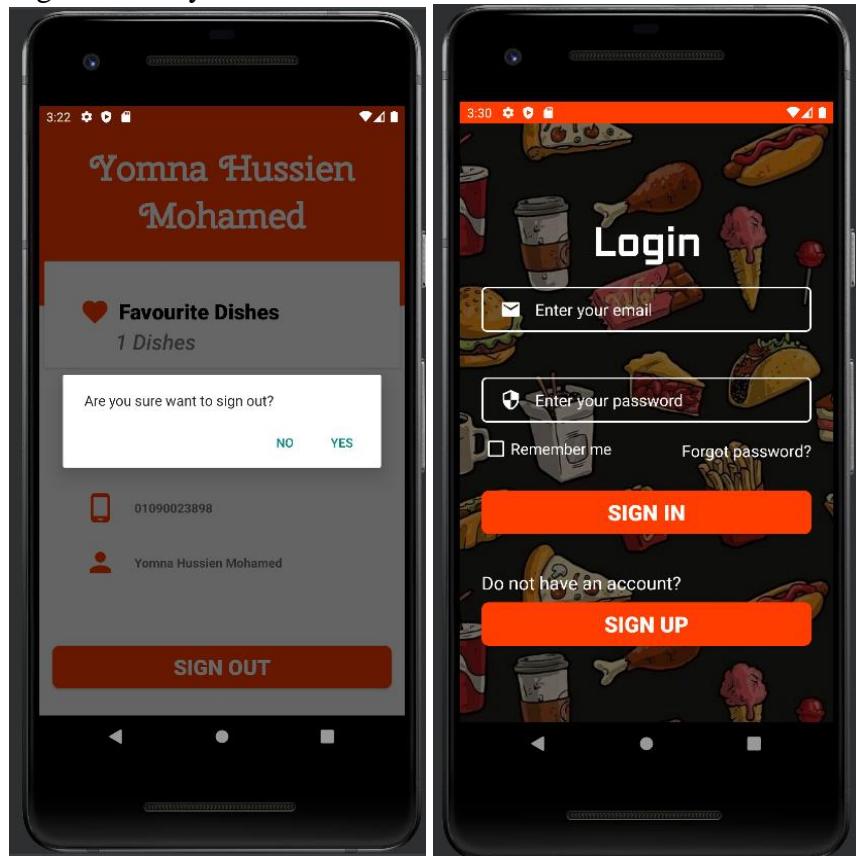
## XI. PROFILE

## Test Cases

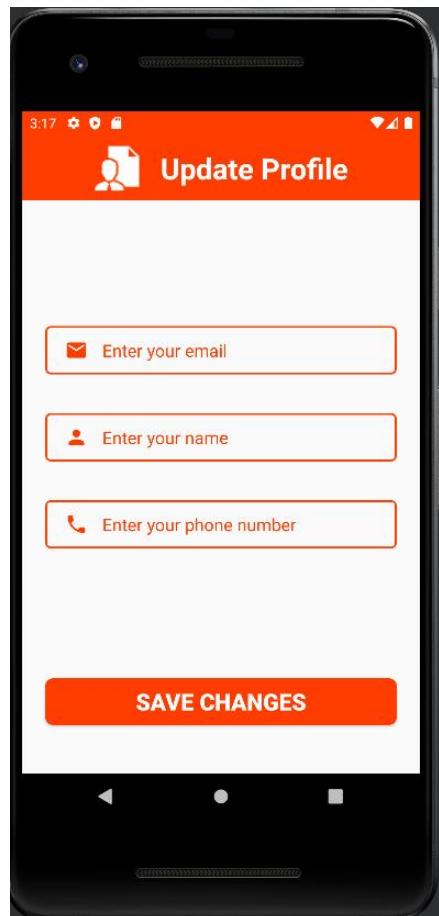
- ## 1. Profile layout



2. Clicking on sign out then yes



3. Clicking on edit profile button



## Code

### Profile Activity

```
package com.example.wagbaapplication;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

import io.paperdb.Paper;

public class Profile extends AppCompatActivity {

    Button signOutButton;
    TextView numberOffavouriteDishes, profileNameTitle, profileMail, profileNumber,
    profileName, editProfile;
    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile);

        signOutButton = (Button) findViewById(R.id.signOutButton);
        numberOffavouriteDishes = (TextView)
        findViewById(R.id.numberOffavouriteDishes);
        profileNameTitle = (TextView) findViewById(R.id.profileNameTitle);
        profileMail = (TextView) findViewById(R.id.profileMail);
        profileNumber = (TextView) findViewById(R.id.profileNumber);
        profileName = (TextView) findViewById(R.id.profileName);
        editProfile = (TextView) findViewById(R.id.editProfile);
        databaseReference =
        FirebaseDatabase.getInstance().getReference().child("AllFavourites");

        //Getting favorites from Firebase
        // Read from the database
        databaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String favoritesTotal =
                String.valueOf(dataSnapshot.child(FirebaseAuth.getInstance().getCurrentUser().getUid()
                )).getChildrenCount();
                numberOffavouriteDishes.setText(favoritesTotal+ " Dishes");
            }
            @Override
            public void onCancelled(DatabaseError error) {
                // Failed to read value
            }
        });
    }
}
```

```

        }

    });

    //Performing Room database query
    UserDatabase userDatabase =
UserDatabase.getUserDatabase(getApplicationContext());
    final UserDao userDao = userDatabase.userDao();
    new Thread(new Runnable() {
        @Override
        public void run() {
            User user =
userDao.getUser(FirebaseAuth.getInstance().getCurrentUser().getUid());
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    profileNameTitle.setText(user.getUserName());
                    profileName.setText(user.getUserName());
                    profileMail.setText(user.getUserEmail());
                    profileNumber.setText(user.getUserPhone());
                }
            });
        }
    }).start();

    signOutButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            AlertDialog.Builder builder = new AlertDialog.Builder(Profile.this)
                .setMessage("Are you sure want to sign out?")
                .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface, int
i) {
                        Intent intent = new Intent(Profile.this,
Login.class);
                        startActivity(intent);
                    }
                }).setNegativeButton("No", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface, int
i) {}});
                builder.show();
        }
    });

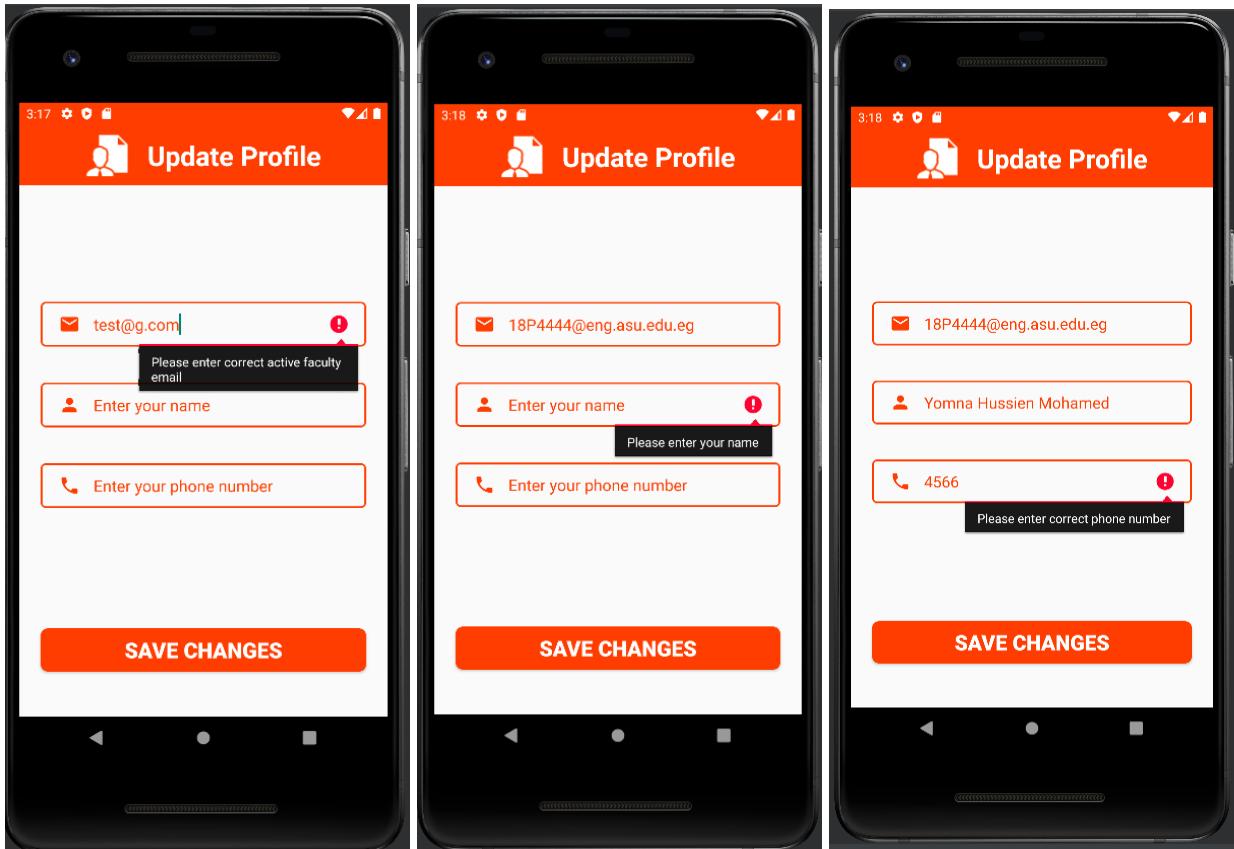
    editProfile.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(Profile.this, UpdateProfile.class);
            startActivity(intent);
            finish();
        }
    });
}
}

```

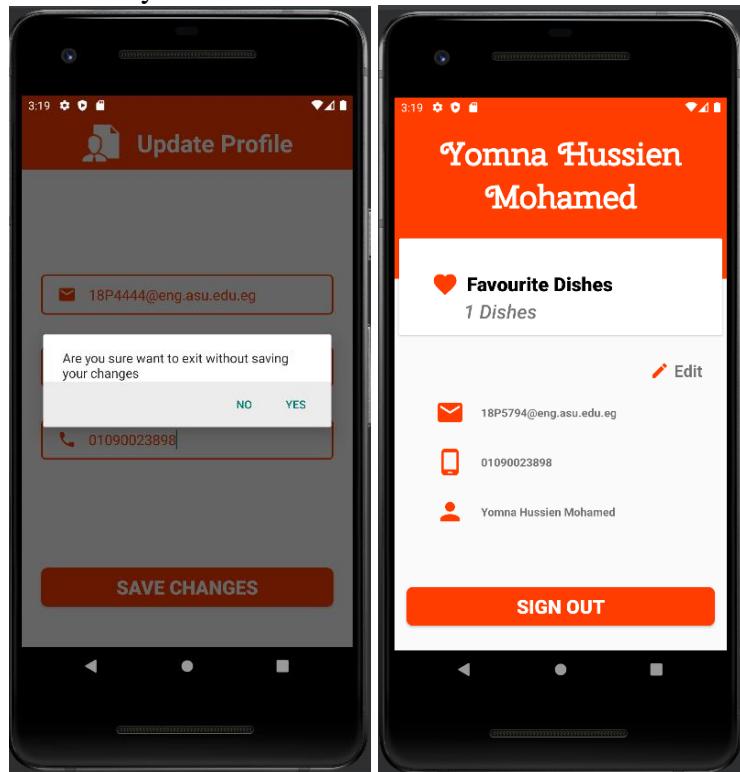
## XII. UPDATE PROFILE

### Test Cases

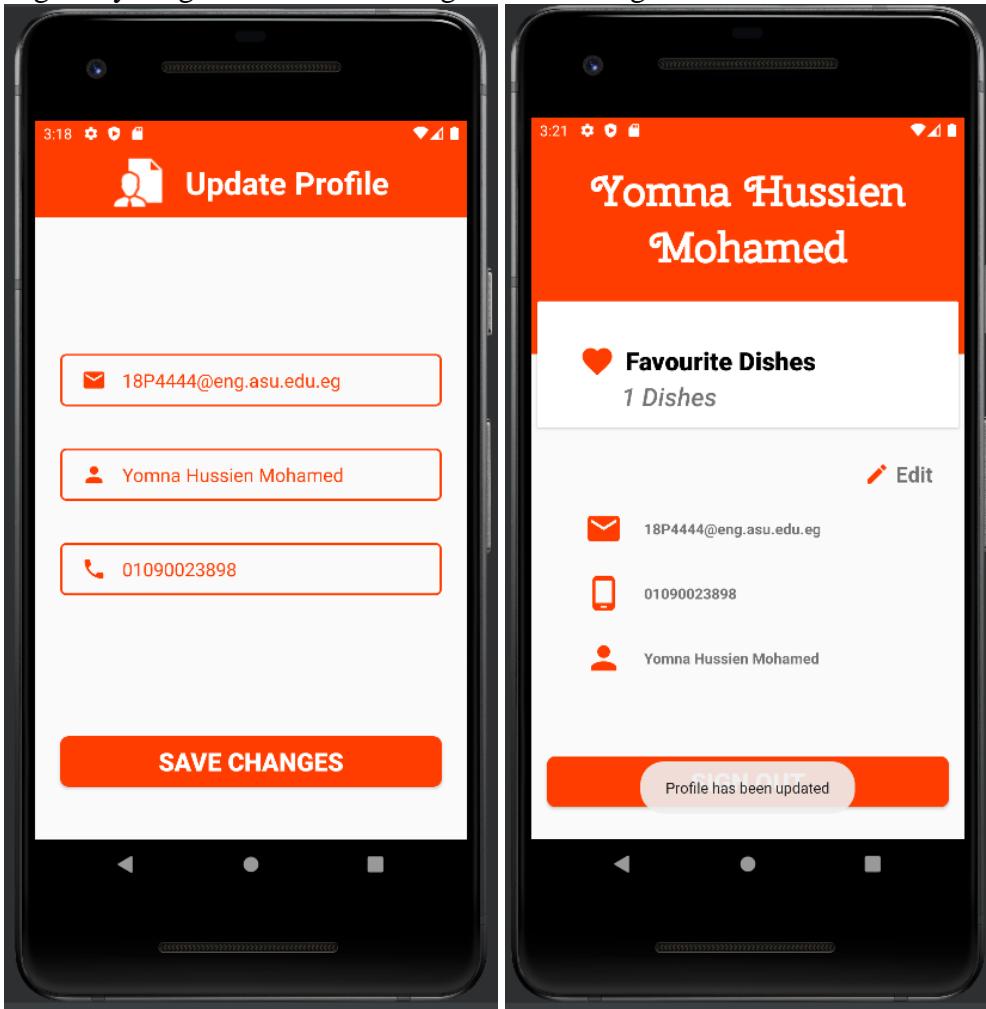
1. Not entering correct email or entering incorrect email / Not entering correct name or entering incorrect name / Not entering correct phone number or entering incorrect phone number



2. Clicking on back button then yes



3. Entering everything correct and clicking on save changes button



## ➤ Room Database User Updated

App Inspection					
Pixel 2 API 30 > com.example.wagbaapplication					
Database Inspector		Network Inspector		Background Task Inspector	
Databases	Tables	Rows	Columns	Operations	
user_table	user_id, user_name, user_email, user_phone	2	user_id, user_name, user_email, user_phone	Live updates	50
room_master_table					
user_table					

The screenshot shows the Room Database Inspector in Android Studio. It displays the 'user\_table' with two rows. Row 1 has user\_id 1, user\_name 'test name', user\_email 'test123@eng.asu.edu.eg', and user\_phone '01090056898'. Row 2 has user\_id 2, user\_name 'Yomna Hussien Mohamed', user\_email '18P4444@eng.asu.edu.eg', and user\_phone '01090023898'.

## ➤ Firebase Authentication User Updated

Search by email address, phone number or user UID					Add user	⋮	
Identifier	Providers	Created	Signed in	User UID			
18p4444@eng.asu.edu.eg	✉	29 Dec 2022	29 Dec 2022	e31VogWsqaZde6uTsfA9kZXEjKO2			
test123@eng.asu.edu.eg	✉	29 Dec 2022	29 Dec 2022	803j4dPv8Zfjuig8oLauECd3ANq1			
			Rows per page	50	1 – 2 of 2	◀ ▶	

## Code

### UpdateProfile Activity

```
package com.example.wagbaapplication;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;

import io.paperdb.Paper;

public class UpdateProfile extends AppCompatActivity {

    EditText updateProfileEmail, updateProfileName, updateProfilePhone;
    Button saveChangesButton;
    String emailPattern = "(\w+@\w+\.\w+\.\w+)";
    String phoneNumberPattern = "^\d{10}$";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update_profile);

        updateProfileEmail = (EditText) findViewById(R.id.updateProfileEmail);
        updateProfileName = (EditText) findViewById(R.id.updateProfileName);
        updateProfilePhone = (EditText) findViewById(R.id.updateProfilePhone);
        saveChangesButton = (Button) findViewById(R.id.saveChangesButton);

        saveChangesButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Paper.book().destroy();
                update();
            }
        });
    }

    private void update() {
        String inputEmail = updateProfileEmail.getText().toString();
        String inputUserName = updateProfileName.getText().toString();
        String inputPhoneNumber = updateProfilePhone.getText().toString();

        if (!inputEmail.matches(emailPattern)) {
            updateProfileEmail.setError("Please enter correct active faculty email");
            updateProfileEmail.requestFocus();
        } else if (inputUserName.isEmpty()) {
            updateProfileName.setError("Please enter your name");
            updateProfileName.requestFocus();
        } else if (!inputPhoneNumber.matches(phoneNumberPattern)) {
            updateProfilePhone.setError("Please enter correct phone number");
            updateProfilePhone.requestFocus();
        } else {
    
```

```

        //update email in Firebase
        FirebaseAuth.getInstance().getCurrentUser().updateEmail(inputEmail);

        //update in Room database
        // 1) creating new user
        User user = new
User(FirebaseAuth.getInstance().getCurrentUser().getUid(), inputUserName, inputEmail,
inputPhoneNumber);

        // 2) Initializing the database
        FirebaseDatabase userDatabase =
UserDatabase.getUserDatabase(getApplicationContext());

        // 3) Initializing the Dao
        final UserDao userDao = userDatabase.userDao();

        // 4) Inserting to database in a new thread so we don't block the UI
        new Thread(new Runnable() {
            @Override
            public void run() {
                userDao.update(user);
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        Toast.makeText(UpdateProfile.this, "Profile has been
updated", Toast.LENGTH_SHORT).show();
                    }
                });
            }
        }).start();

        Intent intent = new Intent(UpdateProfile.this, Profile.class);
        startActivity(intent);
        finish();
    }

}

@Override
public void onBackPressed() {
    AlertDialog.Builder builder = new AlertDialog.Builder(UpdateProfile.this)
        .setMessage("Are you sure want to exit without saving your changes")
        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                Intent intent = new Intent(UpdateProfile.this,
Profile.class);
                startActivity(intent);
                finish();
            }
        })
        .setNegativeButton("No", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {}
        });
    builder.show();
}
}

```

### XIII. EXTRA JAVA CLASSES

#### Prevalent Class

```
package com.example.wagbaapplication;

public class Prevalent {
    public static String emailKey = "UserEmail";
    public static String passwordKey= "UserPassword";
}
```

#### User Class (Entity)

```
package com.example.wagbaapplication;

import androidx.annotation.NonNull;
import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.PrimaryKey;

@Entity(tableName = "user_table")
public class User {

    @PrimaryKey
    @NonNull
    @ColumnInfo(name = "user_id")
    private String uid;

    @ColumnInfo(name = "user_name")
    private String userName;

    @ColumnInfo(name = "user_email")
    private String userEmail;

    @ColumnInfo(name = "user_phone")
    private String userPhone;

    public User(@NonNull String uid, String userName, String userEmail, String userPhone) {
        this.uid = uid;
        this.userName = userName;
        this.userEmail = userEmail;
        this.userPhone = userPhone;
    }

    @NonNull
    public String getUid() {
        return uid;
    }

    public void setUid(@NonNull String uid) {
        this.uid = uid;
    }

    public String getUserName() {
        return userName;
    }
}
```

```

public void setUserName(String userName) {
    this.userName = userName;
}

public String getUserEmail() {
    return userEmail;
}

public void setUserEmail(String userEmail) {
    this.userEmail = userEmail;
}

public String getUserPhone() {
    return userPhone;
}

public void setUserPhone(String userPhone) {
    this.userPhone = userPhone;
}
}

```

## UserDao Class

```

package com.example.wagbaapplication;

import androidx.room.Dao;
import androidx.room.Insert;
import androidx.room.Query;
import androidx.room.Update;

@Dao
public interface UserDao {

    @Insert
    void insert(User user);

    @Query("SELECT * FROM user_table WHERE user_id=:uid")
    User getUser(String uid);

    @Update
    void update(User user);
}

```

## UserDatabase Class

```

package com.example.wagbaapplication;

import android.content.Context;

import androidx.room.Database;
import androidx.room.Room;
import androidx.room.RoomDatabase;

@Database(entities = {User.class}, version = 1)
public abstract class UserDatabase extends RoomDatabase {

    private static final String dbName = "user_table";
    private static UserDatabase usersDatabase;
}

```

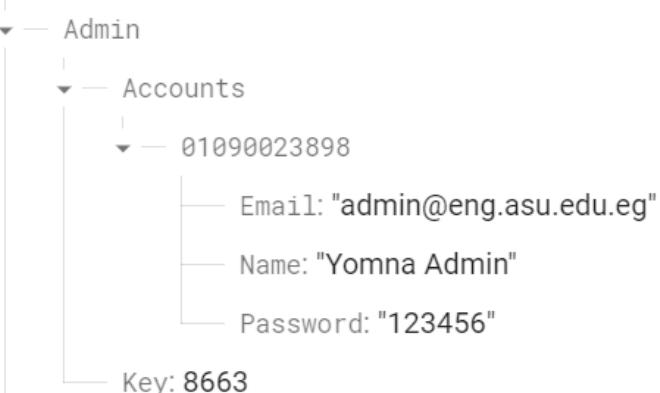
```
public static synchronized UserDatabase getUserDatabase(Context context) {  
    if(usersDatabase == null){  
        usersDatabase = Room.databaseBuilder(context, UserDatabase.class, dbName)  
            .fallbackToDestructiveMigration()  
            .build();  
    }  
    return usersDatabase;  
}  
public abstract UserDao userDao();  
}
```

# **WAGBA ADMIN APPLICATION**

## **I. WELCOME PAGE**

### **Test Cases**

1. Entering wrong code number (number that the admin has so that he can access the application)



2. Entering correct code number



## Code

### Welcome Activity

```
package com.example.wagbaadminstrator;

import androidx.appcompat.app.AppCompatActivity;
import android.app.AlertDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class Welcome extends AppCompatActivity {

    EditText adminCode;
    Button getStartedButton;
    AlertDialog progressDialog;
    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);

        adminCode = (EditText) findViewById(R.id.adminCode);
        getStartedButton = (Button) findViewById(R.id.getStartedButton);
        progressDialog = new AlertDialog(this);
        databaseReference =
            FirebaseDatabase.getInstance().getReference().child("Admin");

        getStartedButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String inputCode = adminCode.getText().toString();

                progressDialog.setMessage("Please wait checking the admin code");
                progressDialog.setTitle("Admin code");
                progressDialog.setCancelable(false);
                progressDialog.show();

                databaseReference.addValueEventListener(new ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {
                        // This method is called once with the initial value and
                        again
                        // whenever data at this location is updated.
                        String value =
                            dataSnapshot.child("Key").getValue().toString();
                            if(inputCode.equals(value)) {

```

```
        progressDialog.dismiss();
        Intent intent = new Intent(Welcome.this, Login.class);
        startActivity(intent);

    }else {
        progressDialog.dismiss();
        Toast.makeText(Welcome.this, "Incorrect admin code",
        Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onCancelled(DatabaseError error) {

}

}) ;

}

}
}

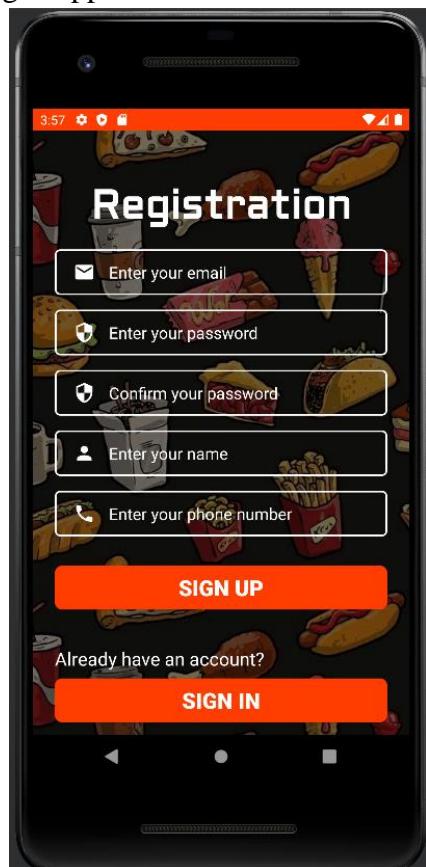
@Override
public void onBackPressed() {
    super.onBackPressed();
    finishAffinity();
}

}
```

## **II. REGISTRATION**

## Test Cases

Same test cases as registration for wagba application



## Code

### Registration activity

```
package com.example.wagbaadminstrator;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.HashMap;

public class Registration extends AppCompatActivity {

    EditText registerEmail, registerPassword, confirmPassword, registerUserName,
registerPhoneNumber;
    Button signUp, signIn;
    String emailPattern = "(\w+@\w+\.\w+\.\w+\.\w+)";
    String phoneNumberPattern = "^\d{10}$";
    ProgressDialog progressDialog;
    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);

        registerEmail = (EditText) findViewById(R.id.editTextEmail2);
        registerPassword = (EditText) findViewById(R.id.editTextPassword2);
        confirmPassword = (EditText) findViewById(R.id.editTextConfirmPass);
        registerUserName = (EditText) findViewById(R.id.editTextUserName);
        registerPhoneNumber = (EditText) findViewById(R.id.editTextPhoneNumber);
        signUp = (Button) findViewById(R.id.signUpButton2);
        signIn = (Button) findViewById(R.id.signInButton2);
        progressDialog = new ProgressDialog(this);
        databaseReference =
        FirebaseDatabase.getInstance().getReference().child("Admin").child("Accounts");

        signUp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                register();
            }
        });
        signIn.setOnClickListener(new View.OnClickListener() {
```

```

        @Override
        public void onClick(View view) {
            Intent intent = new Intent(Registration.this, Login.class);
            startActivity(intent);
            finish();
        }
    });

}

private void register() {
    String inputEmail = registerEmail.getText().toString();
    String inputPass = registerPassword.getText().toString();
    String confirmPass = confirmPassword.getText().toString();
    String inputUserName = registerUserName.getText().toString();
    String inputPhoneNumber = registerPhoneNumber.getText().toString();

    if (!inputEmail.matches(emailPattern)) {
        registerEmail.setError("Please enter correct active faculty email");
        registerEmail.requestFocus();
    } else if (inputPass.isEmpty() || inputPass.length() < 6) {
        registerPassword.setError("Please enter password not less than 6
characters");
        registerPassword.requestFocus();
    } else if (!inputPass.equals(confirmPass)) {
        confirmPassword.setError("Password does not match");
        confirmPassword.requestFocus();
    } else if (inputUserName.isEmpty()) {
        registerUserName.setError("Please enter your name");
        registerUserName.requestFocus();
    } else if (!inputPhoneNumber.matches(phoneNumberPattern)) {
        registerPhoneNumber.setError("Please enter correct phone number");
        registerPhoneNumber.requestFocus();
    } else{
        progressDialog.setMessage("Please wait while registering");
        progressDialog.setTitle("Registration");
        progressDialog.setCanceledOnTouchOutside(false);
        progressDialog.show();

        databaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if (!(dataSnapshot.hasChild(inputPhoneNumber))) {
                    HashMap<String, Object> adminAccount = new HashMap<>();
                    adminAccount.put("Password", inputPass);
                    adminAccount.put("Name", inputUserName);
                    adminAccount.put("Email", inputEmail);

                    databaseReference.child(inputPhoneNumber).updateChildren(adminAccount).addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if (task.isSuccessful()){
                                progressDialog.dismiss();
                                sendToNewActivity();
                            }
                        }
                    });
                } else{
                    progressDialog.dismiss();
                }
            }
        });
    }
}

```

```

        @Override
        public void onCancelled(DatabaseError error) {
            // Failed to read value
        }
    });

}

private void sendToNewActivity() {
    Intent intent02 = new Intent(this, HomePage.class);

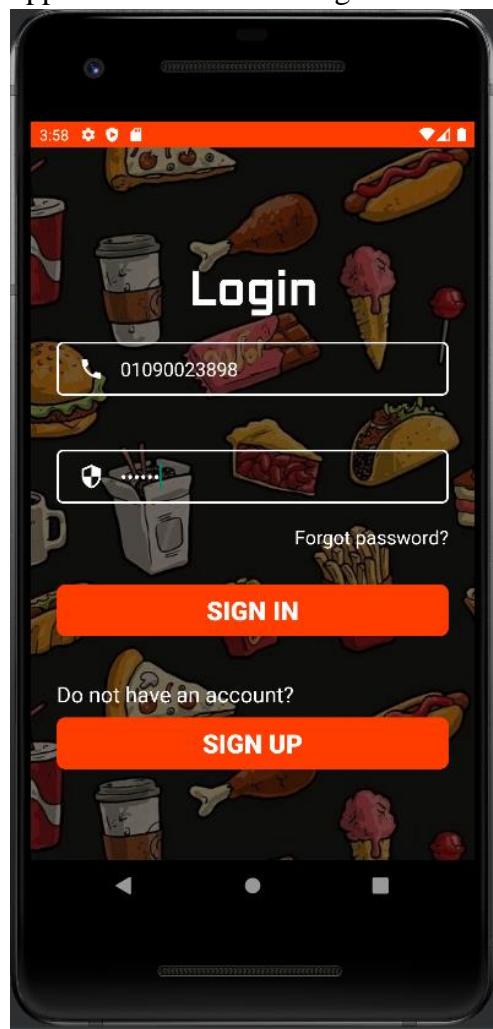
    intent02.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK|Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent02);
    finish();
}
}

```

### III. LOGIN

#### Test Cases

Same test cases as login for wagba application. After entering correct info, it goes to the home page



## Code

### Login Activity

```
package com.example.wagbaadminstrator;

import androidx.appcompat.app.AppCompatActivity;
import android.app.AlertDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class Login extends AppCompatActivity {

    EditText loginNumber, loginPassword;
    Button signUp, signIn;

    String phoneNumberPattern = "^(01[0125][0-9]{8})$";
    ProgressDialog progressDialog;
    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        loginNumber = (EditText) findViewById(R.id.editTextEmail);
        loginPassword = (EditText) findViewById(R.id.editTextPassword);
        signIn = (Button) findViewById(R.id.signInButton);
        signUp = (Button) findViewById(R.id.signUpButton);
        progressDialog = new ProgressDialog(this);
        databaseReference =
            FirebaseDatabase.getInstance().getReference().child("Admin").child("Accounts");

        signIn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                login();
            }
        });

        signUp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent01 = new Intent(Login.this, Registration.class);
                startActivity(intent01);
            }
        });
    }

    private void login() {
        if (loginNumber.getText().toString().matches(phoneNumberPattern)) {
            progressDialog.show();
            databaseReference.addValueEventListener(new ValueEventListener() {
                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    if (dataSnapshot.exists()) {
                        String email = dataSnapshot.child(loginNumber.getText().toString()).child("Email").getValue();
                        String password = dataSnapshot.child(loginNumber.getText().toString()).child("Password").getValue();
                        if (email.equals(loginNumber.getText().toString()) & password.equals(loginPassword.getText().toString())) {
                            Intent intent02 = new Intent(Login.this, Home.class);
                            startActivity(intent02);
                            progressDialog.dismiss();
                        } else {
                            Toast.makeText(Login.this, "Email or Password is incorrect", Toast.LENGTH_SHORT).show();
                            progressDialog.dismiss();
                        }
                    } else {
                        Toast.makeText(Login.this, "User does not exist", Toast.LENGTH_SHORT).show();
                        progressDialog.dismiss();
                    }
                }

                @Override
                public void onCancelled(DatabaseError databaseError) {
                    Toast.makeText(Login.this, "Database Error", Toast.LENGTH_SHORT).show();
                    progressDialog.dismiss();
                }
            });
        } else {
            Toast.makeText(Login.this, "Enter valid phone number", Toast.LENGTH_SHORT).show();
        }
    }
}
```

```

private void login() {
    String inputNumber = loginNumber.getText().toString();
    String inputPass = loginPassword.getText().toString();

    if(!inputNumber.matches(phoneNumberPattern)){
        loginNumber.setError("Please enter correct phone number");
        loginNumber.requestFocus();
    }else if(inputPass.isEmpty() || inputPass.length()<6){
        loginPassword.setError("Please enter password not less than 6
characters");
        loginPassword.requestFocus();
    }else {
        progressDialog.setMessage("Please wait while logging");
        progressDialog.setTitle("Login");
        progressDialog.setCanceledOnTouchOutside(false);
        progressDialog.show();

        // Read from the database
        databaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if(dataSnapshot.child(inputNumber).exists()){

                    if(inputPass.equals(dataSnapshot.child(inputNumber).child("Password").getValue().toString())){
                        progressDialog.dismiss();
                        sendToNewActivity();
                    }else{
                        progressDialog.dismiss();
                        Toast.makeText(Login.this, "Incorrect email or password",
Toast.LENGTH_LONG).show();
                    }
                }else{
                    progressDialog.dismiss();
                    Toast.makeText(Login.this, "Incorrect email or password",
Toast.LENGTH_LONG).show();
                }
            }

            @Override
            public void onCancelled(DatabaseError error) {
                // Failed to read value
            }
        });
    }
}

private void sendToNewActivity() {
    Intent intent02 = new Intent(this, HomePage.class);

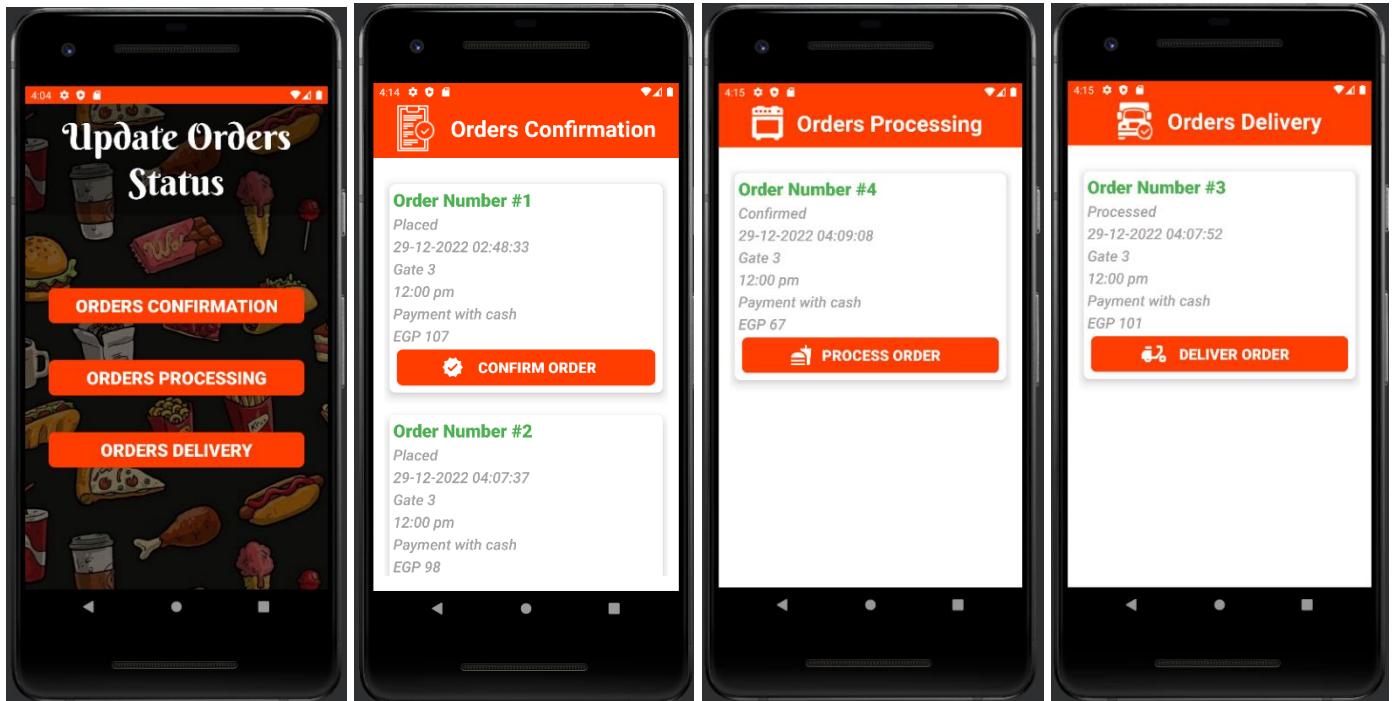
    intent02.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK|Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent02);
    finish();
}
}

```

## IV. HOMEPAGE

### Test Cases

Clicking on orders confirmation button will go to order confirmation activity, clicking on orders processing button will go to order processing activity, clicking on orders delivery button will go to order delivery activity.



### Code

#### HomePage Activity

```
package com.example.wagbaadministrator;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class HomePage extends AppCompatActivity {

    Button orderConfirmation, orderProcessing, orderDelivery;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home_page);

        orderConfirmation = (Button) findViewById(R.id.orderConfirmation);
        orderProcessing = (Button) findViewById(R.id.orderProcessing);
        orderDelivery = (Button) findViewById(R.id.orderDelivery);

        orderConfirmation.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(HomePage.this, OrderConfirmation.class);
                startActivity(intent);
            }
        });
    }
}
```

```
        }
    });
}

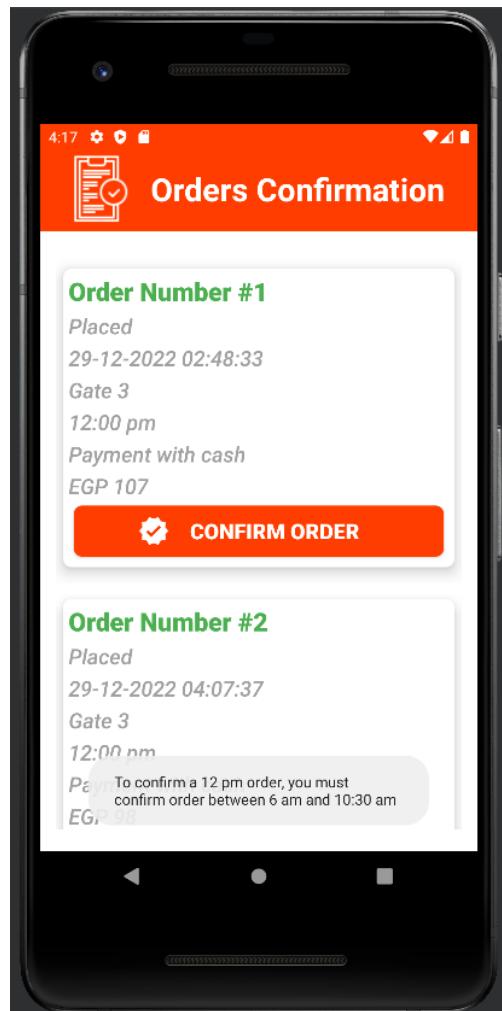
orderProcessing.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(HomePage.this, OrderProcessing.class);
        startActivity(intent);
    }
});

orderDelivery.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(HomePage.this, OrderDelivery.class);
        startActivity(intent);
    }
});
}
```

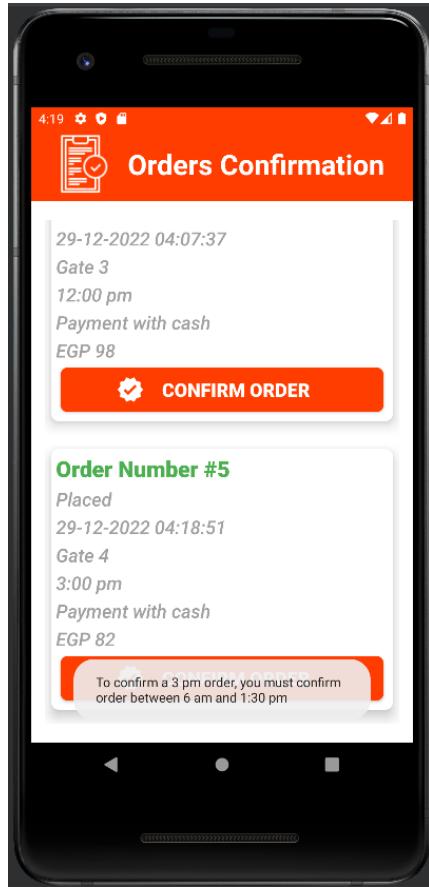
## V. ORDER CONFIRMATION

## Test Cases

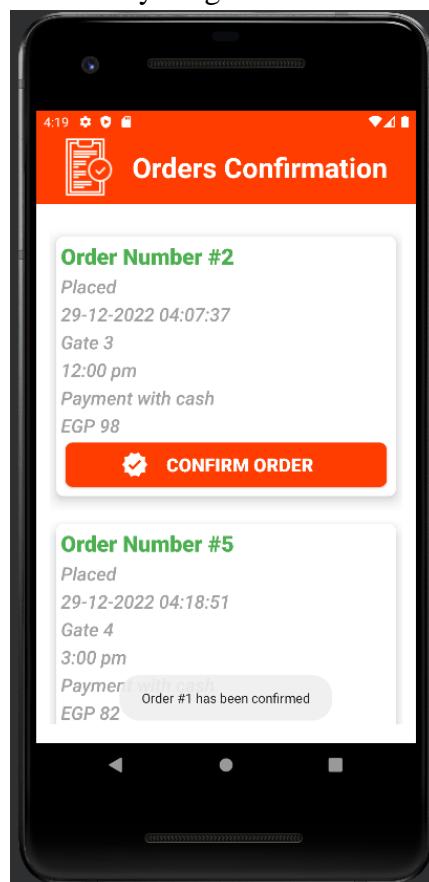
1. Clicking confirm order button with order whose delivery time set at 12 pm but your current time is not between 6 am and 10:30 am



2. Clicking confirm order button with order whose delivery time set at 3 pm but your current time is not between 6 am and 1:30 pm



3. Clicking confirm order button with everything correct



## Code

### OrderConfirmation Activity

```
package com.example.wagbaadminstrator;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.firebaseio.ui.database.FirebaseRecyclerAdapter;
import com.firebaseio.ui.database.FirebaseRecyclerOptions;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

import java.text.SimpleDateFormat;
import java.util.Date;

public class OrderConfirmation extends AppCompatActivity {

    RecyclerView recyclerView;
    TextView noItems;
    FirebaseRecyclerOptions<OrderObject> options;
    FirebaseRecyclerAdapter<OrderObject, OrderViewHolder> adapter;
    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_order_confirmation);

        noItems = (TextView) findViewById(R.id.noItems);
        recyclerView = (RecyclerView) findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new
        LinearLayoutManager(getApplicationContext()));
        recyclerView.setHasFixedSize(true);
        databaseReference =
        FirebaseDatabase.getInstance().getReference().child("AllOrders");

        LoadOrders();
        CheckIfOrdersIsEmpty();
    }

    private void CheckIfOrdersIsEmpty() {
        databaseReference.orderByChild("Status")
            .equalTo("Placed").addValueEventListener(new ValueEventListener() {
                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    if(!dataSnapshot.exists()){
                        noItems.setVisibility(View.VISIBLE);
                    }
                }
            });
    }
}
```

```

        }
        @Override
        public void onCancelled(DatabaseError error) {
        }
    });

private void LoadOrders() {
    options = new FirebaseRecyclerOptions.Builder<OrderObject>()
        .setQuery(databaseReference.orderByChild("Status")
            .equalTo("Placed"), OrderObject.class).build();

    adapter = new FirebaseRecyclerAdapter<OrderObject, OrderViewHolder>(options)
    {
        @Override
        protected void onBindViewHolder(@NonNull OrderViewHolder holder, int position, @NonNull OrderObject model) {

            holder.orderId.setText("Order Number #"
                + getRef(holder.getAbsoluteAdapterPosition()).getKey());
            holder.status.setText(model.getStatus());
            holder.orderTime.setText(model.getOrderTime());
            holder.gateNumber.setText(model.getGateNumber());
            holder.deliveryTime.setText(model.getDeliveryTime());
            holder.paymentMethod.setText(model.getPaymentMethod());
            holder.totalAmount.setText("EGP " + model.getTotalAmount());

            holder.confirmOrder.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {

                    //Order must be confirmed before 10:30 for noon delivery and
                    before 1:30 for 3:00 pm delivery.
                    SimpleDateFormat simpleDateFormat2 = new
                    SimpleDateFormat("Hmm");
                    String currentTime = simpleDateFormat2.format(new Date());
                    String time1 = "600";
                    String time2 = "1030";
                    String time3 = "1330";

                    String testTime1 = "100";
                    String testTime2 = "500";

                    if((model.getDeliveryTime().equals("12:00 pm")) &&
                    (Integer.valueOf(currentTime) < Integer.valueOf(time1) ||
                    Integer.valueOf(currentTime) > Integer.valueOf(time2))){
                        Toast.makeText(OrderConfirmation.this, "To confirm a 12
                    pm order, you must confirm order between 6 am and 10:30 am",
                    Toast.LENGTH_SHORT).show();
                    }else if((model.getDeliveryTime().equals("3:00 pm")) &&
                    (Integer.valueOf(currentTime) < Integer.valueOf(time1) ||
                    Integer.valueOf(currentTime) > Integer.valueOf(time3))){
                        Toast.makeText(OrderConfirmation.this, "To confirm a 3 pm
                    order, you must confirm order between 6 am and 1:30 pm",
                    Toast.LENGTH_SHORT).show();
                    }else{

databaseReference.child(getRef(holder.getAbsoluteAdapterPosition()).getKey())
                .child("Status").setValue("Confirmed");
                    Toast.makeText(OrderConfirmation.this, "Order
                    #" + getRef(holder.getAbsoluteAdapterPosition()).getKey().toString() + " has been
                    confirmed", Toast.LENGTH_SHORT).show();
                }
            });
        }
    };
}

```

```

        }
    }

    @NotNull
    @Override
    public OrderViewHolder onCreateViewHolder(@NotNull ViewGroup parent, int
viewType) {
    View view =
LayoutInflator.from(parent.getContext()).inflate(R.layout.confirmed_view, parent,
false);
    return new OrderViewHolder(view);
}
};

adapter.startListening();
recyclerView.setAdapter(adapter);
}

}

```

## OrderObject Class

```

package com.example.wagbaadminstrator;

public class OrderObject {
    private String DeliveryTime;
    private String GateNumber;
    private String OrderTime;
    private String OrderUserId;
    private String Status;
    private String TotalAmount;
    private String PaymentMethod;

    public OrderObject(String deliveryTime, String gateNumber, String orderTime,
String orderUserId, String status, String totalAmount, String paymentMethod) {
        DeliveryTime = deliveryTime;
        GateNumber = gateNumber;
        OrderTime = orderTime;
        OrderUserId = orderUserId;
        Status = status;
        TotalAmount = totalAmount;
        PaymentMethod = paymentMethod;
    }

    public OrderObject() {
    }

    public String getDeliveryTime() {
        return DeliveryTime;
    }

    public void setDeliveryTime(String deliveryTime) {
        DeliveryTime = deliveryTime;
    }

    public String getGateNumber() {
        return GateNumber;
    }

    public void setGateNumber(String gateNumber) {

```

```

        GateNumber = gateNumber;
    }

    public String getOrderTime() {
        return OrderTime;
    }

    public void setOrderTime(String orderTime) {
        OrderTime = orderTime;
    }

    public String getOrderUserId() {
        return OrderUserId;
    }

    public void setOrderUserId(String orderUserId) {
        OrderUserId = orderUserId;
    }

    public String getStatus() {
        return Status;
    }

    public void setStatus(String status) {
        Status = status;
    }

    public String getTotalAmount() {
        return TotalAmount;
    }

    public void setTotalAmount(String totalAmount) {
        TotalAmount = totalAmount;
    }

    public String getPaymentMethod() {
        return PaymentMethod;
    }

    public void setPaymentMethod(String paymentMethod) {
        PaymentMethod = paymentMethod;
    }
}

```

## OrderViewHolder Class

```

package com.example.wagbaadministrator;

import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class OrderViewHolder extends RecyclerView.ViewHolder{

    TextView orderId, deliveryTime, gateNumber, orderTime, status, totalAmount,
    paymentMethod;
    Button confirmOrder,processOrder, deliverOrder;
}

```

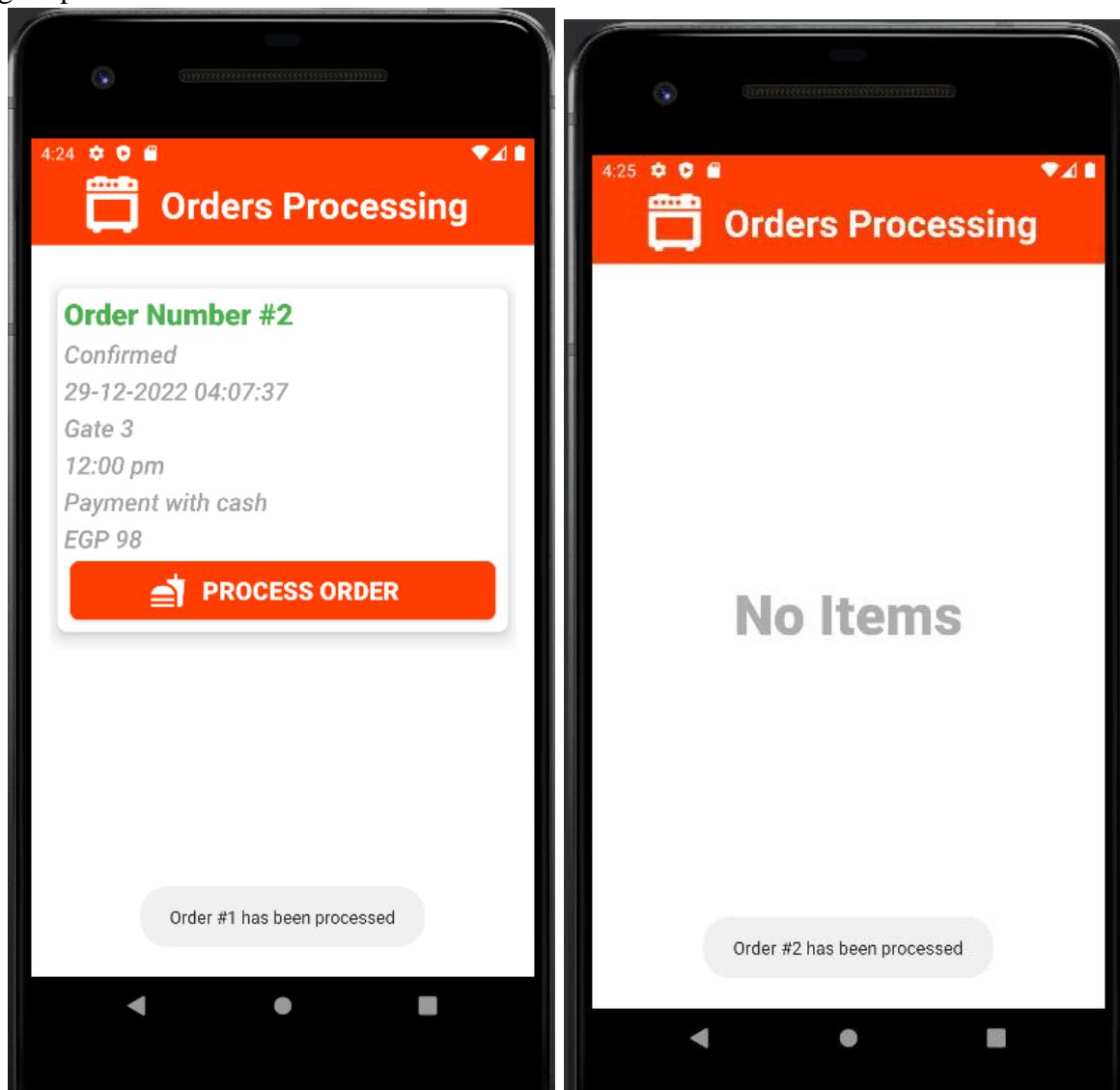
```
public OrderViewHolder(@NonNull View itemView) {
    super(itemView);

    orderId = itemView.findViewById(R.id.orderNumber);
    deliveryTime = itemView.findViewById(R.id.orderTime);
    gateNumber = itemView.findViewById(R.id.orderGate);
    orderTime = itemView.findViewById(R.id.orderDate);
    status = itemView.findViewById(R.id.orderStatus);
    totalAmount = itemView.findViewById(R.id.orderTotal);
    paymentMethod = itemView.findViewById(R.id.orderPaymentMethod);
    confirmOrder = itemView.findViewById(R.id.confirmButton);
    processOrder = itemView.findViewById(R.id.processButton);
    deliverOrder = itemView.findViewById(R.id.deliverButton);
}
```

## VI. ORDER PROCESSING

### Test Cases

Clicking on process order button



## Code

### OrderProcessing Activity

```
package com.example.wagbaadminstrator;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.firebaseio.ui.database.FirebaseRecyclerAdapter;
import com.firebaseio.ui.database.FirebaseRecyclerOptions;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

import java.text.SimpleDateFormat;
import java.util.Date;

public class OrderProcessing extends AppCompatActivity {

    RecyclerView recyclerView;
    TextView noItems;
    FirebaseRecyclerOptions<OrderObject> options;
    FirebaseRecyclerAdapter<OrderObject, OrderViewHolder> adapter;
    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_order_processing);

        noItems = (TextView) findViewById(R.id.noItems);
        recyclerView = (RecyclerView) findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new
        LinearLayoutManager(getApplicationContext()));
        recyclerView.setHasFixedSize(true);
        databaseReference =
        FirebaseDatabase.getInstance().getReference().child("AllOrders");

        LoadOrders();
        CheckIfOrdersIsEmpty();
    }

    private void CheckIfOrdersIsEmpty() {
        databaseReference.orderByChild("Status")
            .equalTo("Confirmed").addValueEventListener(new ValueEventListener()
    {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if(!dataSnapshot.exists()){
                noItems.setVisibility(View.VISIBLE);
            }
        }
    });
}
}
```

```

        }
    }
    @Override
    public void onCancelled(DatabaseError error) {
    }
}) ;
}

private void LoadOrders() {
    options = new FirebaseRecyclerOptions.Builder<OrderObject>()
        .setQuery(databaseReference.orderByChild("Status")
            .equalTo("Confirmed"), OrderObject.class).build();

    adapter = new FirebaseRecyclerAdapter<OrderObject, OrderViewHolder>(options)
    {
        @Override
        protected void onBindViewHolder(@NonNull OrderViewHolder holder, int position, @NonNull OrderObject model) {

            holder.orderId.setText("Order Number #"
                + getRef(holder.getAbsoluteAdapterPosition()).getKey());
            holder.status.setText(model.getStatus());
            holder.orderTime.setText(model.getOrderTime());
            holder.gateNumber.setText(model.getGateNumber());
            holder.deliveryTime.setText(model.getDeliveryTime());
            holder.paymentMethod.setText(model.getPaymentMethod());
            holder.totalAmount.setText("EGP " + model.getTotalAmount());

            holder.processOrder.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {

                    DatabaseReference.child(getRef(holder.getAbsoluteAdapterPosition()).getKey())
                        .child("Status").setValue("Processed");
                    Toast.makeText(OrderProcessing.this, "Order
#" + getRef(holder.getAbsoluteAdapterPosition()).getKey().toString() + " has been
processed", Toast.LENGTH_SHORT).show();
                }
            });
        }

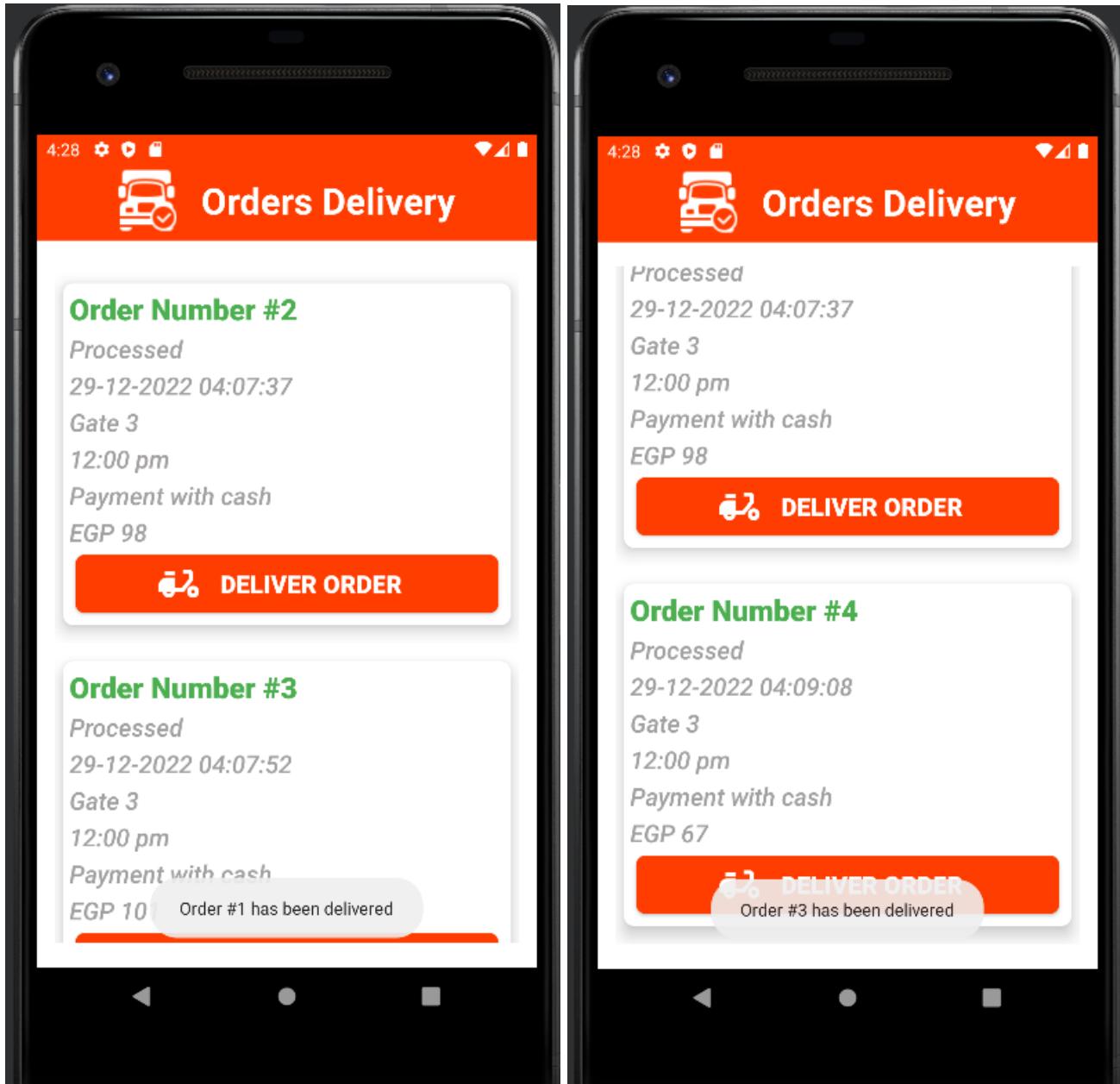
        @NonNull
        @Override
        public OrderViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
            View view =
LayoutInflator.from(parent.getContext()).inflate(R.layout.processing_view, parent,
false);
            return new OrderViewHolder(view);
        }
    };
    adapter.startListening();
    recyclerView.setAdapter(adapter);
}
}

```

## VII. ORDER DELIVERY

### Test Cases

Clicking on deliver order button



### Code

#### OrderDelivery Activity

```
package com.example.wagbaadminstrator;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;
import android.view.LayoutInflater;
```

```

import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.firebaseio.ui.database.FirebaseRecyclerAdapter;
import com.firebaseio.ui.database.FirebaseRecyclerOptions;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

public class OrderDelivery extends AppCompatActivity {

    RecyclerView recyclerView;
    TextView noItems;
    FirebaseRecyclerOptions<OrderObject> options;
    FirebaseRecyclerAdapter<OrderObject, OrderViewHolder> adapter;
    DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_order_delivery);

        noItems = (TextView) findViewById(R.id.noItems);
        recyclerView = (RecyclerView) findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new
        LinearLayoutManager(getApplicationContext()));
        recyclerView.setHasFixedSize(true);
        databaseReference =
        FirebaseDatabase.getInstance().getReference().child("AllOrders");

        LoadOrders();
        CheckIfOrdersIsEmpty();
    }

    private void CheckIfOrdersIsEmpty() {
        databaseReference.orderByChild("Status")
            .equalTo("Processed").addValueEventListener(new ValueEventListener()
    {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if(!dataSnapshot.exists()){
                noItems.setVisibility(View.VISIBLE);
            }
        }
        @Override
        public void onCancelled(DatabaseError error) {
        }
    });
    }

    private void LoadOrders() {
        options = new FirebaseRecyclerOptions.Builder<OrderObject>()
            .setQuery(databaseReference.orderByChild("Status")
                .equalTo("Processed"), OrderObject.class).build();

        adapter = new FirebaseRecyclerAdapter<OrderObject, OrderViewHolder>(options)
    }
}

```

```
    protected void onBindViewHolder(@NonNull OrderViewHolder holder, int position, @NonNull OrderObject model) {

        holder.orderId.setText("Order Number #" +
getRef(holder.getAbsoluteAdapterPosition()).getKey());
        holder.status.setText(model.getStatus());
        holder.orderTime.setText(model.getOrderTime());
        holder.gateNumber.setText(model.getGateNumber());
        holder.deliveryTime.setText(model.getDeliveryTime());
        holder.paymentMethod.setText(model.getPaymentMethod());
        holder.totalAmount.setText("EGP " + model.getTotalAmount());

        holder.deliverOrder.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

databaseReference.child(getRef(holder.getAbsoluteAdapterPosition()).getKey())
                .child("Status").setValue("Delivered");
                Toast.makeText(OrderDelivery.this, "Order
#" +getRef(holder.getAbsoluteAdapterPosition()).getKey().toString() +" has been
delivered", Toast.LENGTH_SHORT).show();
            }
        });
    }

    @NonNull
    @Override
    public OrderViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.delivery_view, parent,
false);
        return new OrderViewHolder(view);
    }
};

adapter.startListening();
recyclerView.setAdapter(adapter);
}
}
```