Princess Noura University

College of Computer & Information Sciences

Computer Science Department

**Garage application**

**(بسطَة)**

**CS 410T**

**A close up of a logo

Description automatically generated**

|  |  |
| --- | --- |
| **Name** | **ID** |
| Reem Ahmed AlRashed | **436000234** |
| Nada Jahz Alharbi | **436000657** |
| Duna Nasser AlZunaytan | **436002628** |
| Meelaf Khaled Alhejeelan | **437004612** |
| Shroog Abdullah Althubiti | **436001125** |

**Under the supervision of:**

**L. Rawan.**

**Date:** April 5, 2020.

Table of Contents

[Abstract 2](#_Toc37019434)

[1. Introduction 2](#_Toc37019435)

[2. Background 3](#_Toc37019436)

[2.1 Pollution: 3](#_Toc37019437)

[2.1.1 Pollution causes 3](#_Toc37019438)

[2.1.2 Affects of pollution 4](#_Toc37019439)

[2.2 Re-using 6](#_Toc37019440)

[2.2.1 How to Reuse 6](#_Toc37019441)

[2.2.2 Buying and selling used products benefits: 6](#_Toc37019442)

[2.3 Mobile Applications: 7](#_Toc37019443)

[2.3.1 Mobile App Functions 7](#_Toc37019444)

[2.3.2 Why Mobile Apps Are Different From Regular Apps 7](#_Toc37019445)

[2.3.3 Finding and Installing Mobile Apps 7](#_Toc37019446)

[2.4 Android Operating system: 8](#_Toc37019447)

[2.4.1 Android Libraries: 8](#_Toc37019448)

[2.4.2 Android Emulator: 8](#_Toc37019449)

[2.4.3 Android versions: 8](#_Toc37019450)

[2.5 Android Studio: 10](#_Toc37019451)

[2.6 Sqlite Database: 11](#_Toc37019452)

[2.6.1 Services Sqlite provides: 11](#_Toc37019453)

[3. Related Work 12](#_Toc37019454)

[3.1 System 1 (swapit): 12](#_Toc37019455)

[3.2 System 2 ( Haraj): 13](#_Toc37019456)

[3.3 Proposed & Similar Systems Comparison: 14](#_Toc37019457)

[4. Methodology 15](#_Toc37019458)

[4.1 Class diagram 15](#_Toc37019459)

[4.2 ER diagram 16](#_Toc37019460)

[4.3 Your code steps in details 17](#_Toc37019461)

[5. Result (output) 60](#_Toc37019462)

[6. Conclusion 72](#_Toc37019463)

[6.1 Future work: 72](#_Toc37019464)

[7. Reference 73](#_Toc37019465)

# Abstract

This project aims overall to support the concept of selling and buying the used product when the owner does not need them anymore—providing customers with a straightforward process of not throwing the unused item in the garbage or keeping them in a warehouse. Throwing in the trash too much, cause pollution, which causes harm to humans, animals and the environment. Therefore, the target is creating an android mobile application that gives customers the ability to sell and buy pre-owned items.

**Keywords**: sell, buy, used products, garbage, warehouse, pollution, android, mobile application, customers.

# Introduction

Due to the absence of adequate recycling practices and support for the reuse idea, waste is typically discarded or burned directly, resulting in critical pollution of the environment [1]. Trash creates pollution, and that is damaging to people, wildlife as well as the atmosphere.

Hence, a garage store is going to be implemented that is workable on the android operating system. It intends to encourage the idea of reusing products, by providing a user-friendly store application. The application support two types of users admin and customer. The admin is responsible for managing and controlling the system. Admin can log in the system and then view all customers information and the sales. Customers, after signing in the system and log in, can do both selling their used items and buy other customer's items as well.

Some of our system limitations are

Users: Admin and Customer.

Language: The application support Arabic language.

Operating System: The application will be developed in an android platform.

The organization of the report is as follows. In Section 2, the background information are given. Section 3 introduces related works survey. In section 4, the proposed methodology for implementing the project. In section 5, results are given . In section 6, conclusion is presented. And finally, in section 7 refences were provided.

# Background

## Pollution:

Pollution is the adding to the atmosphere of any material (solid, liquid, or gas) or any source of energy at a rate higher than it could be spread, filtered, broken down, reused, or stored in any harmless way. Air, water, and land pollution are the main kinds of waste, generally categorized by climate. All forms of pollution may have negative effects on the environment and wildlife, which also impair human safety and well-being [5].

### Pollution causes

1. **Bacteria, insects and vermin thrive from garbage**

A perfect breeding ground for bacteria, insects and vermin are overflowing waste bins. This raises the chance of catching food poisoning, enteric fever, and other serious diseases.

1. **Overflowing waste causes air pollution and respiratory diseases**

Air pollution is one of the results of overflowing garbage which causes various chronic diseases and many other health effects as chemicals are absorbed from the lungs into the other parts of the body. In waste- the poisonous materials contain carbon dioxide, nitrous oxide, and methane.

1. **Garbage contaminates surface waters, which affects all ecosystems**

Garbage and liquid waste that end up in water bodies negatively change the chemical composition of the water. Technically, this type of pollution is called water pollution. It affects all ecosystems existing in the water, including fish and other animals that drink the polluted water.

1. **Direct handling of overflowing waste exposes for health risks**

The dangers of cleaning up and treating excess garbage include injuries, chronic conditions and incidents for waste collection workers. Direct interaction with waste can lead to contamination of the skin and blood through contaminated wounds. This is often dangerous to clean up excess trash because of sharp objects, needles and even nuclear waste [6].

### Affects of pollution

#### Air pollution

Heavy air pollution rates may cause a high risk of developing heart disease, breathing difficulties, skin, nose and throat irritation. Air pollution may also cause cardiovascular issues, allergies and other breathing conditions to increase.

Like people, animals may suffer from a variety of environmental pollution-related health issues, including birth abnormalities, generative failure and diseases. In addition to the impacts on people and wildlife, air pollution is having a variety of environmental consequences.

Acid rain produces elevated amounts of nitric and sulfuric acids produced by oxides and sulfur oxides emitted into the air from fossil fuel combustion. Acid rain kills plants and acidifies lands and bodies of water, rendering the environment too toxic for fish and all organisms in the atmosphere.

Nitrogen oxides released into the air by the burning of fossil fuels also contribute to the nitrogen responsible for toxic algae blooms.

The release of man-made compounds is depleting the ozone. The stratosphere's ozone layer creates a defensive shield that filters toxic ultraviolet radiation back into space which otherwise will kill animal and plant life [2].

#### Water pollution

Pollution from water is a significant danger to human beings, animals and sea life.

The consequences of water contamination are based on where contaminants are being stored. Water sources near to urbanised regions continue to be highly contaminated by both legitimate and unlawful dumping of waste and pollutants from agricultural facilities, health centres, and people.

Loss of aquatic creatures is by far the most significant effect of water pollution and can threaten the whole food chain. Pollutants such as cadmium, arsenic, and lead are consumed by small marine species and are then ingested by fish and shellfish, getting increasingly abundant with each step up the food chain and creating significant human and wildlife problems.

Nitrogen pollution in drinking water supplies may cause toxic algal blooms and produce pollutants and destroy fish and other aquatic animals. Close exposure to this poisonous alga triggers significant human health complications involving psychological symptoms, respiratory disorders, gastrointestinal and liver disease, and rashes [3].

#### Effects of Land & Soil Pollution

Land and soil pollution has major effects on people, wildlife, micro-organisms and aquatic species. Heavy metals and soil may cause different skin issues, breathing problems and even numerous forms of cancers.

Such toxic materials come into close contact with the human body through eating fruits and vegetables produced in polluted fields, consuming them by tainted drinking water, close interaction with the skin and breathing in air filled with pollutants and dust.

Habitat destruction is the main issue of land loss and soil erosion. Clear forest and tree cover removal produces extreme environments that destroy forests and biodiversity.

Climate change often causes an ambient disparity, rising the volume of energy that is removed out of the environment, obviously [4].

## Re-using

Reuse involves using something again instead of throwing it in away. Normally, this implies seeking new usage.

Reuse saves the resources and raw materials used to manufacture new goods, saving money and rising the volume of waste factories emit into the air and water. You will reduce the need to extract, ship and process natural materials and manufacture new goods by recycling or reusing plastic, rubber, or glass pieces [17].

### How to Reuse

1. Buy used items.
2. Look for products that use less packaging. When manufacturers make their products with less packaging, they use less raw material. This reduces waste and costs.
3. Buy reusable over disposable items. Look for items that can be reused; the little things can add up. For example, you can bring your own silverware and cup to work, rather than using disposable items.
4. Rent or sell items that are used infrequently, like decorations, tools or furniture [16].

### Buying and selling used products benefits:

1. Less expensive. Buying anything used is less costly than the new alternative, sometimes up to 90% cheaper, but generally at least 50% cheaper. This application is made for people to buy or sell their used things at a lower price. It benefits the sellers to fulfil their financial need and take advantage of things they don't need anymore.
2. Don’t require energy to create. The farming, harvesting, manufacturing, and shipping of a new product need quite a bit of power from electricity and from oil (from running the vehicles used in farming, harvesting, transportation). This is called "embodied energy". Used goods don't require any energy, except perhaps the gas needed for you to deliver the products. Therefore, used products don't generate pollution. The growing and producing of stuff pump a lot of corruption into the environment – including toxic chemicals, pesticides, and carbon emissions. For example, a new cotton T-shirt is responsible for 1/3 pound of pesticides dumped into the cotton fields.
3. Don't have packaging. When you buy almost any new product, it comes with some kind of packaging – plastic hard casing, shrink wrap, cardboard box, etc. It can be really difficult to find a way to recycle all that packaging, or even more annoying, to have to throw it away. When you buy used products, you don't have any packaging to deal with.
4. Support the local economy. Buying from someone in your country keeps your money your country. Buying from a thrift store supports whatever cause they support – uneducated people, the disabled, the blind, the poor or whatever category they support [7].

## Mobile Applications:

A mobile application often referred to as an app, is a type of software program built to run on a mobile device, such as a smartphone or tablet computer. Mobile applications often serve to provide users with services similar to those that are accessed on Personal computers. Mobile applications are typically have limited functionality, specific purpose device devices [13].

### Mobile App Functions

These apps 'purposes range from functionality, productivity, and navigation to entertainment, sports, fitness, and just about any imaginable others. The social media is one of mobile device most common fields. Facebook was also the most commonly downloaded device across all networks in 2017.

There are several electronic institutions for both web platforms and mobile applications development. In general, the distinction lies in purpose: An app is typically more restricted in scale than a mobile website, has more interactivity and offers more accurate details in a medium that is simple and convenient to navigate on a mobile device.

### Why Mobile Apps Are Different From Regular Apps

Many smartphone applications have similar programs planned to run on desktop computers. However, smartphone applications must function under specific restrictions than their web equivalents. Smartphones have a wide variety of screen sizes, storage limitations, CPU capabilities, user interfaces, buttons and touch controls, and developers need to manage all of these.

### Finding and Installing Mobile Apps

As of 2019, the three major players in the mobile apps space are:

* [Google Play](https://www.lifewire.com/what-is-google-play-1616720): for Android devices
* [Apple's App Store](https://www.lifewire.com/ios-app-store-4154778): for iPads and iPhones
* [Amazon AppStore](https://www.lifewire.com/essential-free-kindle-fire-apps-1357786): for Amazon Fire devices

Many websites also offer corresponding apps and provide download links [14][15].

## Android Operating system:

Android is an operating system developed based on Linux, designed specifically for handheld devices such as smartphones and tablet computers with touch screens. In the last 15 years, the operating system has changed far from black and white phones to new smartphones or minicomputers. The android is a software developed in 2003 in Palo Alto, California.

Android is a powerful operating system which supports a massive number of smartphone applications. Some features are more user-friendly and professional. The hardware that serves android applications is based on a framework for the ARM architecture. The Android is an open-source operating system which means it is free, and everyone can use it. The Android has millions of applications available that can help you handle your life one way or another, and that's why Android is so typical with low price prices.

### Android Libraries:

It is a range of libraries on top of a Linux kennel, like open source web browsers like the webkit, library libc. Such libraries are used for audio and video storage and filming. The SQLite is a database that is useful for storing and exchanging app records. Web encryption etc. is the responsibility of the SSL libraries.

### Android Emulator:

The Emulator is a new feature in Android OS. The Emulator is a modern prototype for the development and testing of android software without requiring any hardware device.

The android simulator includes all the functionality of the hardware and software such as mobile phones except for phone calls. It provides a set of navigation and control buttons. It even shows the application via a screen. Emulators use simulated system settings for android users. When the code is working on you, it will use android framework tools to support specific applications, access the network, play audio, video, store and retrieve the data [9].

### Android versions:

* **Android 1.5, Cupcake:**April 27, 2009
* **Android 1.6, Donut:** September 15, 2009
* **Android 2.0-2.1, Eclair:** October 26, 2009 (initial release)
* **Android 2.2-2.2.3, Froyo:**May 20, 2010 (initial release)
* **Android 2.3-2.3.7, Gingerbread:**December 6, 2010 (initial release)
* **Android 3.0-3.2.6, Honeycomb**: February 22, 2011 (initial release)
* **Android 4.0-4.0.4, Ice Cream Sandwich:**October 18, 2011 (initial release)
* **Android 4.1-4.3.1, Jellybean:**July 9, 2012 (initial release)
* **Android 4.4-4.4.4, KitKat:** October 31, 2013 (initial release)
* **Android 5.0-5.1.1, Lollipop:**November 12, 2014 (initial release)
* **Android 6.0-6.0.1, Marshmallow:**October 5, 2015 (initial release)
* **Android 7.0-7.1.2, Nougat:**August 22, 2016 (initial release)
* **Android 8.0-8.1, Oreo:**August 21, 2017 (initial release)
* **Android 9.0, Pie:**August 6, 2018
* **Android 10.0**: September 3, 2019 [10]

## Android Studio:

Android Studio is the certified integrated development environment (IDE) for Android application development. It's built to accelerate applications development and help developers create the best quality applications on any Android device. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.

Android Studio uses a Gradle-based build framework, debugger, project samples, and collaboration with Github to facilitate the creation of software within the Android operating system. Every Android Studio project includes one or more modalities including source code and resource data. Such modalities include modules for the Android app, Library modules, and modules for the Google Device System.

Android Studio uses an Immediate Push feature to transfer code and resource updates to an application that is running. A code editor lets developers create code and provide the delivery, refraction and evaluation of code. Applications created in Android Studio for submission to the Google Play Store are then compiled into the APK file.

The platform was first revealed at Google I / O in May 2013, with the introduction of the first stable version in December 2014. Android Studio is designed for mobile systems running Ios, Windows, and Linux. It replaced Eclipse Android Development Tools (ADT) as the primary IDE for developing Android applications. You will access Android Studio and the Application Development Kit directly from Google+ [11][12].

## Sqlite Database:

SQLite is an in-process library that implements a transactional SQL database system that is self-contained, serverless, zero-configure. The SQLite code is in the public domain and is thus open for use on any reason, whether commercial or private. SQLite is the most commonly used database in the world with more implementations, including many high-profile initiatives, than we can count.

The SQLite project began on the year 2000-05-09. The future is still challenging to foresee but the developers 'goal is to help SQLite until 2050. Having this purpose in mind, concept choices are made.

### Services Sqlite provides:

1. Tables, indexes, triggers, and views in unlimited quantity
2. Up to 32K columns in a table and unlimited rows
3. Indexes can use DESC and COLLATE
4. CHECK, UNIQUE, NOT NULL, and FOREIGN KEY constraints.
5. ACID transactions using BEGIN, COMMIT, and ROLLBACK
6. Nested transactions using SAVEPOINT, RELEASE, and ROLLBACK TO
7. Subqueries, including correlated subqueries
8. Up to 64-way joins
9. LEFT JOIN
10. DISTINCT, ORDER BY, GROUP BY, HAVING, LIMIT, and OFFSET
11. UNION, UNION ALL, INTERSECT, and EXCEPT
12. A rich library of standard SQL functions
13. Aggregate functions including DISTINCT aggregates
14. Window functions
15. UPDATE, DELETE, and INSERT
16. REPLACE INTO
17. The ON CONFLICT clause
18. The INDEXED BY clause
19. Virtual tables
20. Multiple databases on the same database connection using ATTACH DATABASE

And so many more services. SQLite has a full-featured SQL implementation [8].

# Related Work

## System 1 (swapit):

A screen shot of a person

Description automatically generated A picture containing photo, small, different, sitting

Description automatically generated A screenshot of a cell phone

Description automatically generated

Figure swapit application Figure swapit application. Figure swapit application.

Table swapit related sytem

|  |  |
| --- | --- |
| Application name | swapit |
| Advantages | **Disadvantages** |
| * Free. * Categorized products. * Unlimited posts. * Offers chatting capability. | * Free features are limited in term of number of photos and characters to descripe the product. * Limited reigon supported. * Badly designed. |

## System 2 ( Haraj):

A screenshot of a cell phone

Description automatically generated A screen shot of a smart phone

Description automatically generated

Figure Haraj application Figure Haraj application

Table Haraj related system

|  |  |
| --- | --- |
| Application name | Haraj (حراج) |
| Advantages | **Disadvantages** |
| * Fully Free. * Categorized products. * Unlimited posts. | * No check out. * Prices determined after negotiation. |

## Proposed & Similar Systems Comparison:

Table Proposed and similar systems comparsion

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Our system** | **System (1)**  swapit | **System (2)**  Haraj |
| **Free** | Fully | Partially | Fully |
| **Users** | Admin, customers. | Cutomers. | Cutomers. |
| **Domain** | Online stores. | Online stores. | Online stores. |
| **Language** | Arabic | English | Arabic |
| **Sign in** | Required | Required | Required |
| **Selling and buying at the same time** | Available | Available | Available |
| **Add selling post** | Unlimited posts | Unlimited posts | Unlimited posts |
| **Products pictures** | Unlimited number of pictures | Limited number of pictures | Unlimited number of pictures |
| **Reigion supported** | Supports any place | Only nearby | Supports any place |
| **Chat** | Not available | Available | Not Available |
| **Check out** | Available | Not available | Not available |

# Methodology

## A screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedClass diagram

## ER diagram

## Your code steps in details

**activity\_login.xml:**

A login is a set of credentials used to gain access to an area requiring proper authorization. Login grant access to and control of  [Bastah](https://www.computerhope.com/jargon/c/computer.htm) application, login page consist of both a [username](https://www.computerhope.com/jargon/u/username.htm) and [password](https://www.computerhope.com/jargon/p/password.htm). When a login fails (the username and password combination does not match a user account), the user is disallowed access. There is a signup button for the user who does not have an account. And we have one admin with username: admin, and password: 123.

The attributes are:

* ImageView contains an application image.
* First TextView contains login text.
* The second and third TextView contains username and password texts.
* First and second EditText allows entering username and password.
* The first button to allow login.
* The second button to allow sin up.
* The fourth TextView contains if you do not have an account text.
* <?xml version="1.0" encoding="utf-8"?>  
  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   xmlns:tools="http://schemas.android.com/tools"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   app:theme="@style/AppTheme">  
    
   <ImageView  
   android:id="@+id/imageView4"  
   android:layout\_width="149dp"  
   android:layout\_height="162dp"  
   android:layout\_marginStart="130dp"  
   android:layout\_marginTop="2dp"  
   android:layout\_marginEnd="130dp"  
   android:layout\_marginBottom="14dp"  
   android:background="@drawable/pngbastah"  
   app:layout\_constraintBottom\_toTopOf="@+id/textView"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="1.0"  
   app:layout\_constraintStart\_toEndOf="@+id/signupBtn"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent" />  
    
   <TextView  
   android:id="@+id/textView"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginTop="168dp"  
   android:gravity="center"  
   android:text=" تسجيل الدخول"  
   android:textColor="@color/colorPrimaryDark"  
   android:textSize="30sp"  
   android:textStyle="bold"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.0"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent" />  
    
   <TextView  
   android:id="@+id/textView3"  
   android:layout\_width="380dp"  
   android:layout\_height="wrap\_content"  
   android:layout\_gravity="right"  
   android:layout\_marginStart="17dp"  
   android:layout\_marginTop="286dp"  
   android:layout\_marginEnd="336dp"  
   android:layout\_marginBottom="426dp"  
   android:text="اسم المستخدم"  
   android:textSize="24sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.003"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.446"  
   tools:ignore="RtlCompat" />  
    
   <TextView  
   android:id="@+id/textView4"  
   android:layout\_width="380dp"  
   android:layout\_height="36dp"  
   android:layout\_marginStart="25dp"  
   android:layout\_marginTop="73dp"  
   android:layout\_marginEnd="336dp"  
   android:layout\_marginBottom="335dp"  
   android:text="كلمة المرور"  
   android:textAlignment="textStart"  
   android:textColor="#5A585E"  
   android:textSize="24sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.023"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toBottomOf="@+id/textView3"  
   app:layout\_constraintVertical\_bias="0.375"  
   tools:ignore="RtlCompat" />  
    
   <EditText  
   android:id="@+id/username"  
   android:layout\_width="380dp"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginStart="20dp"  
   android:layout\_marginEnd="183dp"  
   android:layout\_marginBottom="13dp"  
   android:ems="10"  
   android:inputType="textPersonName"  
   android:textCursorDrawable="@color/colorPrimary"  
   app:layout\_constraintBottom\_toTopOf="@+id/textView4"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.0"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toBottomOf="@+id/textView3"  
   app:layout\_constraintVertical\_bias="0.0" />  
    
   <com.google.android.material.textfield.TextInputLayout  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginTop="27dp"  
   android:layoutDirection="rtl"  
   app:errorEnabled="true"  
   app:layout\_constraintBottom\_toTopOf="@+id/signupBtn"  
   app:layout\_constraintTop\_toTopOf="@+id/textView4"  
   app:passwordToggleEnabled="true">  
    
   <EditText  
   android:id="@+id/pass"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginLeft="20dp"  
   android:layout\_marginTop="10dp"  
   android:layout\_marginRight="20dp"  
   android:ems="10"  
   android:inputType="textPassword"  
   android:singleLine="true"  
   android:textAlignment="viewStart"  
   android:textColor="@color/black"  
   android:textSize="18sp" />  
   </com.google.android.material.textfield.TextInputLayout>  
    
    
   <Button  
   android:id="@+id/loginBtn"  
   android:layout\_width="200dp"  
   android:layout\_height="56dp"  
   android:layout\_marginBottom="40dp"  
   android:background="@drawable/button"  
   android:text="تسجيل الدخول"  
   android:textSize="24sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.497"  
   app:layout\_constraintStart\_toStartOf="parent" />  
    
   <Button  
   android:id="@+id/signupBtn"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginTop="432dp"  
   android:layout\_marginEnd="19dp"  
   android:background="@color/colorPrimary"  
   android:text="انضم إلينا"  
   android:textSize="18dp"  
   android:textStyle="bold|italic"  
   app:layout\_constraintEnd\_toStartOf="@+id/accountTV"  
   app:layout\_constraintTop\_toTopOf="parent" />  
    
   <TextView  
   android:id="@+id/accountTV"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginEnd="24dp"  
   android:text=" ليس لديك حساب؟"  
   android:textSize="17sp"  
   android:textStyle="italic|bold"  
   app:layout\_constraintBottom\_toBottomOf="@+id/signupBtn"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintTop\_toTopOf="@+id/signupBtn" />  
  </androidx.constraintlayout.widget.ConstraintLayout>

**LoginActivity.java:**

LoginActivity page extends AppCompatActivity. In the onCreate method, there is a findViewById() method that finds the view from the layout resource file that is attached with current Activity, ProgressDialog() class is used to show the progress of a task contain methods such as setTitle(), setMessage(), setCancelable(), setIndeterminate(), show(), and hide().

When the user presses the login button, the isExist() method will take username and password that user entered and check the result if it is true the customer will be login, and these preferences will automatically save to SharedPreferences as the user interacts with them, to retrieve an instance of SharedPreferences call getDefaultSharedPreference. And if the result is false will check the username is admin and password is 123 then the admin will be login and if it is neither then setError() contain login failed, username or password wrong text. Lastly, if the person presses the signup button then the signup page will display.

package com.example.project;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
  
public class LoginActivity extends AppCompatActivity {  
 Button btnLogin, signupBtn;  
 EditText edtUsernme, edtPassword;  
 DBHelper databaseHelper;  
 // DBHelper db;  
 boolean isExist;  
 ProgressDialog mProgress;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
 btnLogin = (Button) findViewById(R.id.*loginBtn*);  
 signupBtn = (Button) findViewById(R.id.*signupBtn*);  
 edtUsernme = (EditText) findViewById(R.id.*username*);  
 edtPassword = (EditText) findViewById(R.id.*pass*);  
  
 databaseHelper = new DBHelper(LoginActivity.this);  
 mProgress = new ProgressDialog(this);  
 mProgress.setTitle("جاري تسجيل الدخول...");  
 mProgress.setMessage("فضلًا انتظر...");  
 mProgress.setCancelable(false);  
 mProgress.setIndeterminate(true);  
 mProgress.hide();  
  
 btnLogin.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 mProgress.show();  
 isExist = databaseHelper.checkUserExist(edtUsernme.getText().toString(), edtPassword.getText().toString());  
 if (isExist) {  
 Intent intent = new Intent(LoginActivity.this, MainActivity.class);  
 intent.putExtra("username", edtUsernme.getText().toString());  
 SharedPreferences sharedPreferences = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 SharedPreferences.Editor editor = sharedPreferences.edit();  
 editor.putString("username", edtUsernme.getText().toString());  
 editor.commit();  
 startActivity(intent);  
 } else if (edtUsernme.getText().toString().equals("admin") && edtPassword.getText().toString().equals("123")) {  
 startActivity(new Intent(LoginActivity.this, AdminHome.class));  
 } else {  
 mProgress.hide();  
 edtUsernme.setError("فشل تسجيل الدخول. اسم المستخدم أو كلمة المرور خاطئة!");  
 }  
 }  
 });  
 signupBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(LoginActivity.this, CustomerSignUp.class));  
 }  
 });  
  
 }//end of onCreate()  
  
}//end of class

**admin\_home.xml:**

Home page for the admin after success login, he will have three buttons each button will take him to a specific page to continue his tasks.

The attributes are:

* ImageView contains an application image.
* TextView will appear hello admin text.
* The first button will take the admin to view the sales page.
* The second button will take the admin to view the customers info page.
* The third button allows logout.
* <?xml version="1.0" encoding="utf-8"?>  
  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   xmlns:tools="http://schemas.android.com/tools"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:background="@color/colorAccent"  
   tools:context=".AdminHome">  
    
   <ImageView  
   android:id="@+id/imageView3"  
   android:layout\_width="400dp"  
   android:layout\_height="258dp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.498"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.077"  
   app:srcCompat="@drawable/pngbastah" />  
    
   <TextView  
   android:id="@+id/helloUser"  
   android:layout\_width="0dp"  
   android:layout\_height="109dp"  
   android:textAppearance="?android:attr/textAppearanceLarge"  
   android:textColor="#1A721E"  
   android:textSize="36sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.0"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.464" />  
    
   <Button  
   android:id="@+id/Admin\_logoutBtn"  
   android:layout\_width="171dp"  
   android:layout\_height="80dp"  
   android:background="@drawable/button"  
   android:text="تسجيل الخروج"  
   android:textColor="#F9F9F9"  
   android:textSize="25sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.913" />  
    
   <Button  
   android:id="@+id/admin\_salesBtn"  
   android:layout\_width="131dp"  
   android:layout\_height="69dp"  
   android:background="@drawable/button"  
   android:text="المبيعات"  
   android:textColor="#F9F9F9"  
   android:textSize="25sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.566" />  
    
   <Button  
   android:id="@+id/admin\_customerBtn"  
   android:layout\_width="131dp"  
   android:layout\_height="69dp"  
   android:background="@drawable/button"  
   android:text="العملاء"  
   android:textColor="#F9F9F9"  
   android:textSize="25sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="@+id/helloUser"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.733" />  
    
  </androidx.constraintlayout.widget.ConstraintLayout>

**AdminHome.java:**

AdminHome page extends AppCompatActivity, getExtra() is used to get values from the intent that are stored in bundle and getIntent() is used to switch between activities. In the onCreate method, there is a getString() take username and setText() will present hello admin in the TextView. If the admin presses the sales button then the admin will view all sales information page, if the admin presses the costumers button then the admin will view all costumers information page, and if the admin presses the logout button then the admin will logout from the application successfully.

package com.example.project;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class AdminHome extends AppCompatActivity {  
 TextView helloUser;  
 Button salesBtn, customerBtn, AdminLogoutBtn;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*admin\_home*);  
 Bundle extras = getIntent().getExtras();  
  
 salesBtn = (Button) findViewById(R.id.*admin\_salesBtn*);  
 customerBtn = (Button) findViewById(R.id.*admin\_customerBtn*);  
 AdminLogoutBtn = (Button) findViewById(R.id.*Admin\_logoutBtn*);  
  
 String username = null;  
 if (extras != null) {  
 username = extras.getString("username");  
 helloUser.setText(" .." + username + "مرحبا بالمشرف ");  
 }  
 salesBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 viewAllSales();  
 }  
 });  
 customerBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 viewAllCustomer();  
 }  
 });  
 AdminLogoutBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(AdminHome.this, SplashActivity.class));  
 }  
 });  
  
 }  
  
  
 public void viewAllCustomer() {  
 Intent intent = new Intent(AdminHome.this, ViewAllCustomersInfo.class);  
 startActivity(intent);  
  
 }  
  
 public void viewAllSales() {  
 Intent intent = new Intent(AdminHome.this, ViewAllSales.class);  
 startActivity(intent);  
  
 }  
}//end of class

**customer\_signup.xml:**

Sign up page will allow the customer to create his account easily by entering their information then save it to the database so they can log in to the application.

The attributes are:

* ImageView contains an application image.
* The first TextView will appear create account text.
* Five TextView to tell the customer which the information to enter such as username, password, name, email, and phone.
* Five EditText to allow the customer to enter his information.
* Button to save all the customer information.
* <?xml version="1.0" encoding="utf-8"?>  
  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   xmlns:tools="http://schemas.android.com/tools"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:theme="@style/AppTheme">  
    
   <LinearLayout  
   android:id="@+id/linearLayout2"  
   android:layout\_width="0dp"  
   android:layout\_height="0dp"  
   android:layout\_marginStart="4dp"  
   android:layout\_marginLeft="4dp"  
   android:layout\_marginTop="2dp"  
   android:layout\_marginEnd="4dp"  
   android:layout\_marginRight="4dp"  
   android:layout\_marginBottom="2dp"  
   android:orientation="vertical"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent">  
    
   <ImageView  
   android:id="@+id/imageView2"  
   android:layout\_width="180dp"  
   android:layout\_height="100dp"  
   android:layout\_gravity="center"  
   android:background="@drawable/pngbastah2"  
   app:layout\_constraintBottom\_toTopOf="@+id/textView3"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="1.0"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.0" />  
    
   <TextView  
   android:id="@+id/textView"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:text="إنشاء حساب"  
   android:textAlignment="center"  
   android:textColor="@color/colorPrimaryDark"  
   android:textSize="26sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toTopOf="@+id/textView3"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toBottomOf="@+id/imageView2" />  
    
   <TextView  
   android:id="@+id/usernameTV"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_gravity="right"  
   android:layout\_marginRight="20dp"  
   android:text="اسم المستخدم"  
   android:textSize="19sp"  
   android:textStyle="bold" />  
    
   <EditText  
   android:id="@+id/username"  
   android:layout\_width="365dp"  
   android:layout\_height="40dp"  
   android:layout\_gravity="center"  
   android:ems="10"  
   android:inputType="textPersonName"  
   android:textStyle="italic" />  
    
   <TextView  
   android:id="@+id/passwordTV"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_gravity="right"  
   android:layout\_marginRight="20dp"  
   android:text="كلمة المرور"  
   android:textSize="19sp"  
   android:textStyle="bold"  
   tools:layout\_editor\_absoluteX="0dp" />  
    
   <com.google.android.material.textfield.TextInputLayout  
   android:layout\_width="match\_parent"  
   android:layout\_height="50dp"  
   android:layoutDirection="rtl"  
   app:errorEnabled="true"  
   app:passwordToggleEnabled="true">  
    
   <EditText  
   android:id="@+id/pass"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginLeft="20dp"  
   android:layout\_marginRight="20dp"  
   android:ems="10"  
   android:inputType="textPassword"  
   android:singleLine="true"  
   android:textColor="@color/black"  
   android:textSize="18sp" />  
   </com.google.android.material.textfield.TextInputLayout>  
    
   <TextView  
   android:id="@+id/nameTV"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_gravity="right"  
   android:layout\_marginRight="20dp"  
   android:text="الاسم"  
   android:textSize="19dp"  
   android:textStyle="bold" />  
    
   <EditText  
   android:id="@+id/name"  
   android:layout\_width="365dp"  
   android:layout\_height="40dp"  
   android:layout\_gravity="center"  
   android:ems="10"  
   android:inputType="textPersonName"  
   android:textStyle="italic" />  
    
   <TextView  
   android:id="@+id/emailTV"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_gravity="right"  
   android:layout\_marginRight="20dp"  
   android:text="البريد الإلكتروني"  
   android:textSize="19sp"  
   android:textStyle="bold" />  
    
   <EditText  
   android:id="@+id/email"  
   android:layout\_width="365dp"  
   android:layout\_height="40dp"  
   android:layout\_gravity="center"  
   android:ems="10"  
   android:inputType="textEmailAddress"  
   android:textStyle="italic" />  
    
    
   <TextView  
   android:id="@+id/phoneTV"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_gravity="right"  
   android:layout\_marginRight="20dp"  
   android:text="رقم الجوال"  
   android:textSize="19dp"  
   android:textStyle="bold"  
   tools:ignore="RtlCompat" />  
    
    
   <EditText  
   android:id="@+id/phone"  
   android:layout\_width="365dp"  
   android:layout\_height="40dp"  
   android:layout\_gravity="right"  
   android:layout\_marginRight="20dp"  
   android:ems="10"  
   android:inputType="phone"  
   android:textAlignment="textEnd"  
   android:textStyle="italic"  
   tools:ignore="RtlCompat" />  
    
   <Button  
   android:id="@+id/createAccountBtn"  
   android:layout\_width="130dp"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginLeft="135dp"  
   android:background="@drawable/button"  
   android:layout\_marginTop="5dp"  
   android:foregroundTint="@color/colorPrimary"  
   android:text="إنشاء"  
   android:textSize="23sp"  
   android:textStyle="bold" />  
    
    
   </LinearLayout>  
    
  </androidx.constraintlayout.widget.ConstraintLayout>

**CustomerSignUp.java:**

CustomerSignUp extends AppCompatActivity and implements view.OnClickListner. In the onCreate method, check if getId() of the [android.view.View](https://www.programcreek.com/java-api-examples/index.php?api=android.view.View) class equal with R.id.createAccountBtn if it is true then getWritableDatabase() will Create and/or open a database that will be used for reading and writing and beginTransaction() will automatically make the transaction explicit and holds a lock on the table until the transaction is either committed or rolled back. Try to insert value to customer table, [setTransactionSuccessful](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase#setTransactionSuccessful())() will mark the current transaction as successful, Toast.makeText() will present you sign up successfully text, then the login page will appear. Or finally, Toast.makeText() will present Failed text, and [endTransaction](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase#endTransaction())() to end a transaction.

package com.example.project;  
  
import android.content.Intent;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import com.example.project.DBHelper;  
import com.example.project.LoginActivity;  
import com.example.project.R;  
  
public class CustomerSignUp extends AppCompatActivity implements View.OnClickListener {  
 DBHelper db = new DBHelper(this);  
  
 SQLiteDatabase database;  
 EditText name, username, password, phone, address;  
 Button createAccountBtn;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*customer\_signup*);  
 name = (EditText) findViewById(R.id.*name*);  
 phone = (EditText) findViewById(R.id.*phone*);  
 username = (EditText) findViewById(R.id.*username*);  
 password = (EditText) findViewById(R.id.*pass*);  
 address = (EditText) findViewById(R.id.*email*);  
  
 createAccountBtn = (Button) findViewById(R.id.*createAccountBtn*);  
 createAccountBtn.setOnClickListener(this);  
 }  
  
 @Override  
 public void onClick(View v) {  
 if (v.getId() == R.id.*createAccountBtn*) {  
 database = db.getWritableDatabase();  
 database.beginTransaction();  
  
 try {  
 //database.execSQL("INSERT INTO user(username,password,userType) VALUES('" + username.getText() + "','" + password.getText() + "','" + "customer" + "')");  
 database.execSQL("INSERT INTO customer(name, username, password, phone, email) VALUES('" + name.getText() + "','" + username.getText() + "','" + password.getText() + "','" + phone.getText() + "','" + address.getText() + "')");  
 database.setTransactionSuccessful();  
 Toast.*makeText*(this, " لقد تم تسجيل حسابك بنجاح", Toast.*LENGTH\_LONG*).show();  
 Intent intent = new Intent(CustomerSignUp.this, LoginActivity.class);  
 startActivity(intent);  
 } finally {  
 Toast.*makeText*(this, " failed", Toast.*LENGTH\_LONG*).show();  
 database.endTransaction();  
 }  
  
 }  
 }  
}

**view\_sales.xml:**

A view sales page is created to view all sales information and will allow keeping a history of their previous orders with a specific purpose, to secure sales for the product. The product selling on a page can differ depending on the industry. However, the purpose of the sales page remains constant – getting more customers.

The attributes are:

* The first TextView will appear the sales information text.
* The second, third, and fourth TextView contain user\_id, product\_id, and price of the sales.
* ListView allows displaying a list of items.
* <?xml version="1.0" encoding="utf-8"?>  
  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent">  
    
    
   <TextView  
   android:id="@+id/result"  
   android:layout\_width="411dp"  
   android:layout\_height="55dp"  
   android:layout\_marginTop="60dp"  
   android:gravity="center"  
   android:text="معلومات المبيعات"  
   android:textAppearance="@style/TextAppearance.AppCompat.Large"  
   android:textSize="40sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toTopOf="@+id/textView11"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.0"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent" />  
    
    
   <TextView  
   android:id="@+id/textView10"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_alignParentBottom="true"  
   android:layout\_marginStart="50dp"  
   android:layout\_marginLeft="50dp"  
   android:layout\_marginEnd="300dp"  
   android:layout\_marginRight="350dp"  
   android:layout\_marginBottom="8dp"  
   android:text="هوية المستخدم"  
   android:textColor="#FF5A585E"  
   android:textSize="24sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toTopOf="@+id/linearLayout"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.393"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toBottomOf="@+id/result"  
   app:layout\_constraintVertical\_bias="0.0" />  
    
   <TextView  
   android:id="@+id/textView11"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_alignParentBottom="true"  
   android:layout\_marginStart="30dp"  
   android:layout\_marginLeft="65dp"  
   android:layout\_marginEnd="187dp"  
   android:layout\_marginRight="187dp"  
   android:layout\_marginBottom="8dp"  
   android:layout\_toRightOf="@+id/textView10"  
   android:text="معرف المنتج"  
   android:textSize="24sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toTopOf="@+id/linearLayout"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.0"  
   app:layout\_constraintStart\_toEndOf="@+id/textView10"  
   app:layout\_constraintTop\_toBottomOf="@+id/result" />  
    
   <TextView  
   android:id="@+id/textView12"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginStart="50dp"  
   android:layout\_marginLeft="85dp"  
   android:layout\_marginTop="7dp"  
   android:layout\_marginEnd="37dp"  
   android:layout\_marginRight="60dp"  
   android:layout\_marginBottom="17dp"  
   android:text="السعر"  
   android:textSize="24sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toTopOf="@+id/linearLayout"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="1.0"  
   app:layout\_constraintStart\_toEndOf="@+id/textView11"  
   app:layout\_constraintTop\_toBottomOf="@+id/result"  
   app:layout\_constraintVertical\_bias="0.375" />  
    
   <LinearLayout  
   android:id="@+id/linearLayout"  
   android:layout\_width="fill\_parent"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginBottom="6dp"  
   android:orientation="horizontal"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toBottomOf="@+id/textView11">  
    
   <ListView  
   android:id="@+id/list"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="564dp" />  
   </LinearLayout>  
  </androidx.constraintlayout.widget.ConstraintLayout>

**view\_customers\_info.xml:**

A view customer page is created to view all customer information and that allows to track customers, after filling out the customers information.

The attributes are:

* The first TextView will appear the customers information text.
* The second, third, and fourth TextView contain a name, phone, and email of the customers.
* ListView allows displaying a list of items.
* <?xml version="1.0" encoding="utf-8"?>  
  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   xmlns:tools="http://schemas.android.com/tools"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   tools:context=".ViewAllCustomersInfo">  
    
    
   <TextView  
   android:id="@+id/result"  
   android:layout\_width="411dp"  
   android:layout\_height="55dp"  
   android:layout\_marginTop="80dp"  
   android:gravity="center"  
   android:text="معلومات المستخدمين"  
   android:textAppearance="@style/TextAppearance.AppCompat.Large"  
   android:textSize="35sp"  
   app:layout\_constraintBottom\_toTopOf="@+id/textView11"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.0"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent" />  
    
   <TextView  
   android:id="@+id/textView10"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_alignParentBottom="true"  
   android:layout\_marginStart="14dp"  
   android:layout\_marginLeft="14dp"  
   android:layout\_marginEnd="333dp"  
   android:layout\_marginRight="333dp"  
   android:layout\_marginBottom="8dp"  
   android:text="الاسم"  
   android:textColor="#FF5A585E"  
   android:textSize="20sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toTopOf="@+id/linearLayout"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.0"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toBottomOf="@+id/result"  
   app:layout\_constraintVertical\_bias="0.0" />  
    
   <TextView  
   android:id="@+id/textView11"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_alignParentBottom="true"  
   android:layout\_marginStart="65dp"  
   android:layout\_marginLeft="65dp"  
   android:layout\_marginEnd="187dp"  
   android:layout\_marginRight="187dp"  
   android:layout\_marginBottom="8dp"  
   android:layout\_toRightOf="@+id/textView10"  
   android:text="رقم الجوال"  
   android:textColor="#FF5A585E"  
   android:textSize="20sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toTopOf="@+id/linearLayout"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.016"  
   app:layout\_constraintStart\_toEndOf="@+id/textView10"  
   app:layout\_constraintTop\_toBottomOf="@+id/result" />  
    
   <TextView  
   android:id="@+id/textView12"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginStart="50dp"  
   android:layout\_marginLeft="85dp"  
   android:layout\_marginTop="7dp"  
   android:layout\_marginEnd="37dp"  
   android:layout\_marginRight="60dp"  
   android:layout\_marginBottom="17dp"  
   android:text="البريد الالكتروني"  
   android:textColor="#FF5A585E"  
   android:textSize="20sp"  
   android:textStyle="bold"  
   app:layout\_constraintBottom\_toTopOf="@+id/linearLayout"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.358"  
   app:layout\_constraintStart\_toEndOf="@+id/textView11"  
   app:layout\_constraintTop\_toBottomOf="@+id/result"  
   app:layout\_constraintVertical\_bias="0.375" />  
    
   <LinearLayout  
   android:id="@+id/linearLayout"  
   android:layout\_width="fill\_parent"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginBottom="6dp"  
   android:orientation="horizontal"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toBottomOf="@+id/textView11">  
    
   <ListView  
   android:id="@+id/list"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="564dp" />  
    
    
   </LinearLayout>  
    
    
  </androidx.constraintlayout.widget.ConstraintLayout>

**layout.xml:**

layout page for the list and to be used in CustomAdapter page.

The attributes are:

* Three TextView contains a name, phone, and email of the customers.
* <?xml version="1.0" encoding="utf-8"?>  
  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent">  
    
   <TextView  
   android:layout\_width="100dp"  
   android:layout\_height="wrap\_content"  
   android:text="New Text"  
   android:textSize="15dp"  
   android:id ="@+id/tvname"  
   android:layout\_marginLeft="15dp"  
   android:layout\_marginRight="15dp"/>  
   <TextView  
   android:layout\_width="100dp"  
   android:layout\_height="wrap\_content"  
   android:text="New Text"  
   android:textSize="15dp"  
   android:id ="@+id/tvphone"  
   android:layout\_marginLeft="15dp"  
   android:layout\_marginRight="15dp"/>  
    
   <TextView  
   android:id="@+id/tvemail"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginLeft="15dp"  
   android:layout\_marginRight="15dp"  
   android:text="New Text"  
   android:textSize="15dp" />  
  </LinearLayout>

**ViewAllCostomersInfo.java:**

ViewAllCostomersInfo extends AppCompatActivity and imports all necessary classes, SQLiteDatabase has methods to create, delete, execute SQL commands, and perform other common database management tasks. In the displayData method, there is a Cursors are what contains the result set of a query made against a database and rawQuery() method of SQLiteDatabase class to use a select query for retrieving data from the customer table. moveToFirst() method moves it to the first row of customer table and moveToNext() move the cursor to the next row.

package com.example.project;  
  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.widget.ListView;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.ArrayList;  
  
public class ViewAllCustomersInfo extends AppCompatActivity {  
  
 DBHelper DatabaseController = new DBHelper(this);  
 SQLiteDatabase db;  
  
 private ArrayList<String> phone = new ArrayList<String>();  
 private ArrayList<String> name =new ArrayList<String>();  
 private ArrayList<String> email =new ArrayList<String>();  
  
 ListView customerslistview;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*view\_customers\_info*);  
 customerslistview = (ListView) findViewById(R.id.*list*);  
 }  
  
 @Override  
 protected void onResume() {  
 displayData();  
 super.onResume();  
  
 }  
  
 private void displayData (){  
 db = DatabaseController.getReadableDatabase();  
 Cursor cursor = db.rawQuery("select \* from customer",null);  
 name.clear();  
 phone.clear();  
 email.clear();  
 if (cursor.moveToFirst()) {// if the cursor points to something  
  
 do {  
 name.add(cursor.getString(cursor.getColumnIndex("name")));  
 email.add(cursor.getString(cursor.getColumnIndex("email")));  
 phone.add(cursor.getString(cursor.getColumnIndex("phone")));  
  
 }while ( cursor.moveToNext() ); // the loop will keep going to next until pointing to null  
  
 }  
 CustomAdapter c = new CustomAdapter(ViewAllCustomersInfo.this,name,email,phone);  
 customerslistview.setAdapter(c);  
 cursor.close();  
  
 }  
}

**layout2.xml:**

layout page for the list and to be used in CustomAdapter2 page.

The attributes are:

* Three TextView contains user\_id, product\_id, and price of the sales.
* <?xml version="1.0" encoding="utf-8"?>  
  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent">  
    
   <TextView  
   android:layout\_width="100dp"  
   android:layout\_height="wrap\_content"  
   android:text="New Text"  
   android:textSize="15dp"  
   android:id ="@+id/tvuser\_id"  
   android:layout\_marginLeft="15dp"  
   android:layout\_marginRight="15dp"/>  
   <TextView  
   android:layout\_width="100dp"  
   android:layout\_height="wrap\_content"  
   android:text="New Text"  
   android:textSize="15dp"  
   android:id ="@+id/tvproduct\_id"  
   android:layout\_marginLeft="15dp"  
   android:layout\_marginRight="15dp"/>  
    
   <TextView  
   android:id="@+id/tvprice"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginLeft="15dp"  
   android:layout\_marginRight="15dp"  
   android:text="New Text"  
   android:textSize="15dp" />  
  </LinearLayout>

**ViewAllSales.java:**

ViewAllSales extends AppCompatActivity and imports all necessary classes. In the displayData method, there is a [getString](https://developer.android.com/reference/android/database/Cursor#getString(int))() returns the value of the column as a String, [getColumnIndex](https://developer.android.com/reference/android/database/Cursor#getColumnIndex(java.lang.String))() returns the zero-based index for the column name, or -1 if the column doesn't exist, [getAdapter](https://developer.android.com/reference/android/widget/AdapterView#getAdapter())() returns the adapter currently associated with this widget, and [close](https://developer.android.com/reference/android/database/Cursor#close())()releasing all resources and making it completely invalid.

package com.example.project;  
  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.widget.ListView;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.ArrayList;  
  
public class ViewAllSales extends AppCompatActivity {  
  
 DBHelper DatabaseController = new DBHelper(this);  
 SQLiteDatabase db;  
  
 private ArrayList<String> user\_id = new ArrayList<String>();  
 private ArrayList<String> product\_id =new ArrayList<String>();  
 private ArrayList<String> price =new ArrayList<String>();  
  
 ListView customerslistview;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*view\_sales*);  
 customerslistview = (ListView) findViewById(R.id.*list*);  
 }  
  
 @Override  
 protected void onResume() {  
 displayData();  
 super.onResume();  
  
 }  
  
 private void displayData (){  
 db = DatabaseController.getReadableDatabase();  
 Cursor cursor = db.rawQuery("select \* from sales",null);  
 user\_id.clear();  
 product\_id.clear();  
 price.clear();  
 if (cursor.moveToFirst()) {// if the cursor points to something  
  
 do {  
 user\_id.add(cursor.getString(cursor.getColumnIndex("user\_id")));  
 product\_id.add(cursor.getString(cursor.getColumnIndex("product\_id")));  
 price.add(cursor.getString(cursor.getColumnIndex("price")));  
 }while ( cursor.moveToNext() ); // the loop will keep going to next until pointing to null  
  
 }  
 CustomAdapter c = new CustomAdapter(ViewAllSales.this,user\_id,product\_id,price);  
 customerslistview.setAdapter(c);  
 cursor.close();  
  
 }  
}

**Controllerdb.java:**

Controllerdb extends SQLiteOpenHelper class and create variable to store database name. In the onCreate method, it will create a customer table with all it is attributes if it is not exit and [execSQL](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase#execSQL(java.lang.String))() execute a single SQL statement that is NOT a SELECT or any other SQL statement that returns data. And the onUpgrade method will drop the table if the customer table does not exist then send it to the onCreate method.

package com.example.project;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
public class Controllerdb extends SQLiteOpenHelper {  
  
 private static final String *DATABASE\_NAME*="data2.db";  
  
 public Controllerdb(Context applicationcontext){  
 super(applicationcontext, *DATABASE\_NAME*, null, 2);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 String query;  
 query= "CREATE TABLE IF NOT EXISTS customer(userid INTEGER PRIMARY KEY AUTOINCREMENT, name VARCHAR,username VARCHAR,password VARCHAR,phone NUMBER, email VARCHAR );";  
 db.execSQL(query);  
  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){  
 String query;  
 query = "DROP TABLE IF EXISTS customer";  
 db.execSQL(query);  
 onCreate(db);  
 }  
}

**CustomAdapter.java:**

CustomAdapter extends BaseAdapter class that can be used in customer ListView, ArrayList is a dynamic data structure in which you can add or remove any number of elements and those elements are stored in an ordered sequence. In the getView method, there is a [getSystemServiceName](https://developer.android.com/reference/android/content/Context#getSystemServiceName(java.lang.Class%3C?%3E))() gets the name of the system-level service that is represented by the specified class and Inflate() is the process of adding a view (.xml) to activity on runtime. When we create a listView we inflate its every item dynamically.

package com.example.project;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.TextView;  
import java.util.ArrayList;  
  
public class CustomAdapter extends BaseAdapter {  
 private Context mContext;  
 Controllerdb controldb;  
 SQLiteDatabase db;  
  
 private ArrayList<String> name =new ArrayList<String>();  
 private ArrayList<String> email =new ArrayList<String>();  
 private ArrayList<String> phone =new ArrayList<String>();  
  
 public CustomAdapter(Context context,ArrayList<String> name, ArrayList<String> email, ArrayList<String> phone){  
 this.mContext = context;  
 this.name=name;  
 this.email=email;  
 this.phone=phone;  
  
 }  
 @Override  
 public int getCount() {  
 return name.size();  
 }  
  
 @Override  
 public Object getItem(int position) {  
 return null ;  
 }  
  
 @Override  
 public long getItemId(int position) {  
 return 0;  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
  
 final viewHolder holder;  
 controldb = new Controllerdb(mContext);  
 LayoutInflater layoutInflater;  
  
 if(convertView == null){  
  
 layoutInflater = (LayoutInflater) mContext.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 convertView = layoutInflater.inflate(R.layout.*layout*, null);  
 holder = new viewHolder();  
  
 holder.name = (TextView) convertView.findViewById(R.id.*tvname*);  
 holder.email = (TextView) convertView.findViewById(R.id.*tvemail*);  
 holder.phone = (TextView) convertView.findViewById(R.id.*tvphone*);  
 convertView.setTag(holder);  
 }  
  
 else  
 {  
 holder = (viewHolder) convertView.getTag(); }  
  
 holder.name.setText(name.get(position));  
 holder.email.setText(email.get(position));  
 holder.phone.setText(phone.get(position));  
 return convertView;  
 }  
  
 public class viewHolder {  
  
 TextView name;  
 TextView email;  
 TextView phone;  
  
 }  
}

**CustomAdapter2.java:**

CustomAdapter2 extends BaseAdapter class that can be used in sales ListView. In the getView method, there are setTag() and getTag() to set and get custom objects as per our requirement. The setTag() method takes an argument of type Object, and getTag() returns an Object. A viewHolder method to create TextView variables.

package com.example.project;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.TextView;  
  
import java.util.ArrayList;  
  
public class CustomAdapter2 extends BaseAdapter {  
 private Context mContext;  
 SalesTableDatabaseController controldb;  
 SQLiteDatabase db;  
  
 private ArrayList<String> user\_id = new ArrayList<String>();  
 private ArrayList<String> product\_id = new ArrayList<String>();  
 private ArrayList<String> price = new ArrayList<String>();  
  
 public CustomAdapter2(Context context, ArrayList<String> user\_id, ArrayList<String> product\_id, ArrayList<String> price) {  
 this.mContext = context;  
 this.user\_id = user\_id;  
 this.product\_id = product\_id;  
 this.price = price;  
 }  
  
 @Override  
 public int getCount() {  
 return user\_id.size();  
 }  
  
 @Override  
 public Object getItem(int position) {  
 return null;  
 }  
  
 @Override  
 public long getItemId(int position) {  
 return 0;  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
  
 final CustomAdapter2.viewHolder holder;  
 controldb = new SalesTableDatabaseController(mContext);  
 LayoutInflater layoutInflater;  
  
 if (convertView == null) {  
  
 layoutInflater = (LayoutInflater) mContext.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 convertView = layoutInflater.inflate(R.layout.*layout2*, null);  
 holder = new CustomAdapter2.viewHolder();  
  
 holder.user\_id = (TextView) convertView.findViewById(R.id.*tvuser\_id*);  
 holder.product\_id = (TextView) convertView.findViewById(R.id.*tvproduct\_id*);  
 holder.price = (TextView) convertView.findViewById(R.id.*tvprice*);  
 convertView.setTag(holder);  
 } else {  
 holder = (CustomAdapter2.viewHolder) convertView.getTag();  
 }  
  
 holder.user\_id.setText(user\_id.get(position));  
 holder.product\_id.setText(product\_id.get(position));  
 holder.price.setText(price.get(position));  
 return convertView;  
 }  
  
 public class viewHolder {  
  
 TextView user\_id;  
 TextView product\_id;  
 TextView price;  
 }  
}

**Customer.java:**

Customer class contains a constructor is a special method that is called whenever an object is created using the new keyword. It contains a block of statements that are used to initialize instance variables of an object before the reference of this object. A set() method and a get() method with return value for every variable.

package com.example.project;  
  
public class Customer {  
  
 private int id;  
 private String name;  
 private String username;  
 private String password;  
 private String email;  
 private int phoneNumber;  
  
  
 public Customer() {  
  
 }  
  
 public Customer(int id, String name, String username, String password, String email, int phoneNumber) {  
 this.id = id;  
 this.name = name;  
 this.username = username;  
 this.password = password;  
 this.email = email;  
 this.phoneNumber = phoneNumber;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public void setUsername(String username) {  
 this.username = username;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
  
 public void setPhoneNumber(int phoneNumber) {  
 this.phoneNumber = phoneNumber;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getUsername() {  
 return username;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public int getPhoneNumber() {  
 return phoneNumber;  
 }  
}//end of class

**Product.java:**

Product class takes the same attributes as the database, it contains set() and get() methods.

package com.example.project;  
  
public class Product {  
 private int id;  
 private String productName;  
 private String productDesc;  
 private String productPrice;  
 private byte[] productPic;  
 private String categoryName;  
  
  
 public Product(int id, String productName,String productDesc, String productPrice, byte[] productPic, String categoryName) {  
 this.id = id;  
 this.productName = productName;  
 this.productDesc = productDesc;  
 this.productPrice = productPrice;  
 this.productPic = productPic;  
 this.categoryName = categoryName;  
 }  
  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public void setProductName(String productName) {  
 this.productName = productName;  
 }  
  
 public void setProductPrice(String productPrice) {  
 this.productPrice = productPrice;  
 }  
  
 public void setProductPic(byte[] productPic) {  
 this.productPic = productPic;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setProductDesc(String productDesc) {  
 this.productDesc = productDesc;  
 }  
  
 public String getProductDesc() {  
 return productDesc;  
 }  
  
 public String getProductName() {  
 return productName;  
 }  
  
 public String getProductPrice() {  
 return productPrice;  
 }  
  
 public byte[] getProductPic() {  
 return productPic;  
 }  
  
 public void setCategoryName(String categoryName) {  
 this.categoryName = categoryName;  
 }  
  
 public String getCategoryName() {  
 return categoryName;  
 }  
}//end of class

**ProductListAdapter.java:**

ProductListAdapter extends BaseAdapter and contains a **constructor** is a special method that is used to initialize objects.

package com.example.project;  
  
import android.content.Context;  
import android.content.Intent;  
import android.database.sqlite.SQLiteDatabase;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import java.util.ArrayList;  
  
public class ProductListAdapter extends BaseAdapter {  
 private Context context;  
 private int layout;  
 SQLiteDatabase database;  
 private ArrayList<Product> productList;  
  
 public ProductListAdapter(Context context, int layout, ArrayList<Product> productList) {  
 this.context = context;  
 this.layout = layout;  
 this.productList = productList;  
 }  
  
 @Override  
 public int getCount() {  
 return productList.size();  
 }  
  
 @Override  
 public Object getItem(int position) {  
 return productList.get(position);  
 }  
  
 @Override  
 public long getItemId(int position) {  
 return position;  
 }  
  
 private class ViewHolder {  
 ImageView imageView;  
 TextView name, price, desc;  
 Button add\_to\_cart;  
 }  
  
 @Override  
 public View getView(final int position, View convertView, ViewGroup parent) {  
 View row = convertView;  
 ViewHolder holder = new ViewHolder();  
  
 if (row == null) {  
 LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 row = inflater.inflate(layout, null);  
  
 holder.name = (TextView) row.findViewById(R.id.*product\_nameTV*);  
 holder.desc = (TextView) row.findViewById(R.id.*product\_descTV*);  
 holder.price = (TextView) row.findViewById(R.id.*product\_priceTV*);  
 holder.imageView = (ImageView) row.findViewById(R.id.*product\_pic*);  
 holder.add\_to\_cart = (Button) row.findViewById(R.id.*add\_to\_cart*);  
 row.setTag(holder);  
 } else {  
 holder = (ViewHolder) row.getTag();  
 }  
  
 Product product = productList.get(position);  
 holder.name.setText(product.getProductName());  
 holder.desc.setText(product.getProductDesc());  
 holder.price.setText(product.getProductPrice() + "SR");  
  
 byte[] image = product.getProductPic();  
 Bitmap bitmap = BitmapFactory.*decodeByteArray*(image, 0, image.length);  
 holder.imageView.setImageBitmap(bitmap);  
 holder.add\_to\_cart.setOnClickListener(new View.OnClickListener() {  
  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent(view.getContext(), AddToCart.class);  
 intent.putExtra("name", productList.get(position).getProductName());  
 intent.putExtra("price", productList.get(position).getProductPrice());  
 intent.putExtra("pic", productList.get(position).getProductPic());  
 view.getContext().startActivity(intent);  
 }  
 });  
  
 return row;  
 }  
}

**We have three fragments (Setting,Dashboard and Profile): profile fragement which get customer information and show it in Edit text with ability to update any of his information, email password..etc:**  
  
package com.example.project.ui.profile;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.LinearLayout;  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import com.example.project.Customer;  
import com.example.project.DBHelper;  
import com.example.project.R;  
import com.google.android.material.snackbar.Snackbar;  
public class ProfileFragment extends Fragment {  
LinearLayout coordinatorLayout;  
Button saveBtn;  
EditText name, username, password, phone, email;  
String CurrentUsername;  
DBHelper db;  
Customer customer;  
public int id;  
public View onCreateView(@NonNull LayoutInflater inflater,  
ViewGroup container, Bundle savedInstanceState) {  
View view = inflater.inflate(R.layout.fragment\_profile, container, false);  
db = new DBHelper(getContext());  
customer = new Customer();  
coordinatorLayout = view.findViewById(R.id.profileLL);  
name = view.findViewById(R.id.namemyaccount);  
username = view.findViewById(R.id.usernamemyaccount);  
password = view.findViewById(R.id.passmyaccout);  
email = view.findViewById(R.id.emailmyaccount);  
phone = view.findViewById(R.id.myaccountphoneET);  
saveBtn = view.findViewById(R.id.saveBtn);  
fillData();//get user information from database  
saveBtn.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View v) {  
updateData();  
}  
});  
return view;  
}//end of onCreateView()  
private void fillData() {  
Bundle extras = getActivity().getIntent().getExtras();  
assert extras != null;  
CurrentUsername = extras.getString("username");  
customer = db.getCustomer(CurrentUsername);  
id=customer.getId();  
name.setText(customer.getName());  
username.setText(customer.getUsername());  
password.setText(customer.getPassword());  
email.setText(customer.getEmail());  
phone.setText(Integer.toString(customer.getPhoneNumber()));  
}//end fillData()  
private void updateData() {  
try {  
db.updateCustomer(name.getText().toString(), username.getText().toString(),  
password.getText().toString(), Integer.parseInt(phone.getText().toString()),  
email.getText().toString(),id);  
Snackbar snackbar = Snackbar  
.make(coordinatorLayout, "تم حفظ التغييرات بنجاح! ", Snackbar.LENGTH\_LONG);  
snackbar.show();  
} catch (Exception e) {  
e.printStackTrace();  
}  
}//end of updateData()  
}//end of fragment  
  
  
  
package com.example.project.ui.setting;  
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.Button;  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AlertDialog;  
import androidx.fragment.app.Fragment;  
import com.example.project.R;  
import com.example.project.SplashActivity;  
public class SettingFragment extends Fragment {  
Button logoutBtn;  
AlertDialog.Builder builder;  
public View onCreateView(@NonNull LayoutInflater inflater,  
ViewGroup container, Bundle savedInstanceState) {  
View view = inflater.inflate(R.layout.fragment\_setting, container, false);  
builder = new AlertDialog.Builder(getContext());  
logoutBtn = (Button) view.findViewById(R.id.logoutBtn);  
logoutBtn.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View v) {  
showLogoutDialogMessage();  
}  
});  
return view;  
}  
public void showLogoutDialogMessage() {  
builder.setMessage("هل أنت متأكد تريد تسجيل الخروج؟").setTitle("تسجيل الخروج")  
.setIcon(R.drawable.logout)  
.setCancelable(false)  
.setPositiveButton("نعم", new DialogInterface.OnClickListener() {  
@Override  
public void onClick(DialogInterface dialog, int which) {  
startActivity(new Intent(getContext(), SplashActivity.class));  
}  
}).setNegativeButton("إلغاء", new DialogInterface.OnClickListener() {  
public void onClick(DialogInterface dialog, int id) {  
dialog.dismiss();  
}  
});  
//Creating dialog box  
AlertDialog alert = builder.create();  
alert.show();  
}  
}//end of fragment  
  
**DashboardFragment which show the four categories we have, with navigation to each category page:**  
  
package com.example.project.ui.dashboard;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.ImageView;  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import com.example.project.AddToCart;  
import com.example.project.BooksCategory;  
import com.example.project.ElectronicCategory;  
import com.example.project.HomeAccessoriesCategory;  
import com.example.project.HomeApplianceCategory;  
import com.example.project.MainActivity;  
import com.example.project.R;  
public class DashboardFragment extends Fragment {  
private ImageView booksCategory, homeAccessoriceCategory, electronicCategory, homeApplianceCategory;  
private Button cart\_page;  
public View onCreateView(@NonNull LayoutInflater inflater,  
ViewGroup container, Bundle savedInstanceState) {  
View view = inflater.inflate(R.layout.fragment\_dashboard, container, false);  
booksCategory = (ImageView) view.findViewById(R.id.categoryBooks);  
homeAccessoriceCategory = (ImageView) view.findViewById(R.id.categoryHome);  
electronicCategory = (ImageView) view.findViewById(R.id.categoryElectronic);  
homeApplianceCategory = (ImageView) view.findViewById(R.id.categoryFurnitures);  
  
booksCategory.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View v) {  
startActivity(new Intent(getContext(), BooksCategory.class));  
}  
});  
homeAccessoriceCategory.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View v) {  
startActivity(new Intent(getContext(), HomeAccessoriesCategory.class));  
}  
});  
electronicCategory.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View v) {  
startActivity(new Intent(getContext(), ElectronicCategory.class));  
}  
});  
homeApplianceCategory.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View v) {  
startActivity(new Intent(getContext(), HomeApplianceCategory.class));  
}  
});  
cart\_page = view.findViewById(R.id.cart\_page);  
cart\_page.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View v) {  
startActivity(new Intent(getContext(), AddToCart.class));  
}  
});  
return view;  
}//end of onCreate()  
}//end of fragment

**PhotoUploaded:**

is the Activity where user can take image to upload his product with product name, description, price and category name. It has showPictureDialog() method to show two choices (from Photo, from camera)  
it has requestMultiplePermissions() method to ask for permission of accessing device photos or camera  
  
package com.example.project;  
import android.Manifest;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.database.sqlite.SQLiteDatabase;  
import android.graphics.Bitmap;  
import android.graphics.drawable.BitmapDrawable;  
import android.media.MediaScannerConnection;  
import android.net.Uri;  
import android.os.Bundle;  
import android.os.Environment;  
import android.provider.MediaStore;  
import android.util.Log;  
import android.view.View;  
import android.view.Window;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.RadioGroup;  
import android.widget.Toast;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.constraintlayout.widget.ConstraintLayout;  
import androidx.core.content.ContextCompat;  
import com.google.android.material.snackbar.Snackbar;  
import com.karumi.dexter.Dexter;  
import com.karumi.dexter.MultiplePermissionsReport;  
import com.karumi.dexter.PermissionToken;  
import com.karumi.dexter.listener.DexterError;  
import com.karumi.dexter.listener.PermissionRequest;  
import com.karumi.dexter.listener.PermissionRequestErrorListener;  
import com.karumi.dexter.listener.multi.MultiplePermissionsListener;  
import java.io.ByteArrayOutputStream;  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.util.Calendar;  
import java.util.List;  
public class PhotoUploaded extends AppCompatActivity {  
Toolbar toolbar;  
EditText productName,productDesc, productPrice;  
Button submitProduct;  
RadioGroup radioGroup;  
String categoryName = "";  
ConstraintLayout constraintLayout;  
//database  
SQLiteDatabase database;  
Controllerdb db = new Controllerdb(this);  
DBHelper dbHelper = new DBHelper(this);  
//fot image uploading  
ImageView imageUploaded;  
private static final String IMAGE\_DIRECTORY = "/demonuts";//path to save uploaded photo  
private int GALLERY = 1, CAMERA = 2;  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
super.onCreate(savedInstanceState);  
setContentView(R.layout.photo\_uploaded);  
imageUploaded = (ImageView) findViewById(R.id.image\_uploaded);  
productName = (EditText) findViewById(R.id.product\_name);  
productDesc = (EditText) findViewById(R.id.product\_desc);  
productPrice = (EditText) findViewById(R.id.product\_price);  
radioGroup = (RadioGroup) findViewById(R.id.radioGroup);  
constraintLayout = (ConstraintLayout) findViewById(R.id.constraintLayout);  
submitProduct = (Button) findViewById(R.id.submit\_productBtn);  
setToolbar();  
imageUploaded.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View v) {  
requestMultiplePermissions();  
showPictureDialog();  
}  
});  
submitProduct.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View v) {  
if (radioGroup.getCheckedRadioButtonId() == R.id.radioBooks) {  
categoryName = "books";  
} else if (radioGroup.getCheckedRadioButtonId() == R.id.radioHomeAccess) {  
categoryName = "home\_accessories";  
} else if (radioGroup.getCheckedRadioButtonId() == R.id.radioElectronic) {  
categoryName = "electronic";  
} else if (radioGroup.getCheckedRadioButtonId() == R.id.radioHomeAppliance) {  
categoryName = "home\_appliance";  
} else {  
Snackbar snackbar = Snackbar  
.make(constraintLayout, "لم تقم باختيار القسم لمنتجك!", Snackbar.LENGTH\_LONG);  
snackbar.show();  
return;  
}  
boolean isInserted = dbHelper.insertProduct(productName.getText().toString(),productDesc.getText().toString(), productPrice.getText().toString(), imageViewByte(imageUploaded)  
, categoryName);  
if (isInserted) {  
Toast.makeText(PhotoUploaded.this, "تم رفع المنتج بنجاح!", Toast.LENGTH\_LONG).show();  
} else {  
Toast.makeText(PhotoUploaded.this, "حدث خطأ!", Toast.LENGTH\_LONG).show();  
}  
}  
});  
}//end of onCreate()  
private byte[] imageViewByte(ImageView image) {  
Bitmap bitmap = ((BitmapDrawable) image.getDrawable()).getBitmap();  
ByteArrayOutputStream stream = new ByteArrayOutputStream();  
bitmap.compress(Bitmap.CompressFormat.PNG, 100, stream);  
byte[] byteArray = stream.toByteArray();  
return byteArray;  
}  
private void showPictureDialog() {  
AlertDialog.Builder pictureDialog = new AlertDialog.Builder(this);  
pictureDialog.setTitle("رفع الصورة");  
String[] pictureDialogItems = {  
"اختيار صورة من استديو الصور",  
"اختيار صورة من الكاميرا"};  
pictureDialog.setItems(pictureDialogItems,  
new DialogInterface.OnClickListener() {  
@Override  
public void onClick(DialogInterface dialog, int which) {  
switch (which) {  
case 0:  
choosePhotoFromGallary();  
break;  
case 1:  
takePhotoFromCamera();  
break;  
}  
}  
});  
pictureDialog.show();  
}  
public void choosePhotoFromGallary() {  
Intent galleryIntent = new Intent(Intent.ACTION\_PICK,  
MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI);  
startActivityForResult(galleryIntent, GALLERY);  
}  
private void takePhotoFromCamera() {  
Intent intent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);  
startActivityForResult(intent, CAMERA);  
}  
@Override  
public void onActivityResult(int requestCode, int resultCode, Intent data) {  
super.onActivityResult(requestCode, resultCode, data);  
if (resultCode == this.RESULT\_CANCELED) {  
return;  
}  
if (requestCode == GALLERY) {  
if (data != null) {  
Uri contentURI = data.getData();  
try {  
Bitmap bitmap = MediaStore.Images.Media.getBitmap(this.getContentResolver(), contentURI);  
String path = saveImage(bitmap);  
Toast.makeText(PhotoUploaded.this, "تم حفظ الصورة", Toast.LENGTH\_SHORT).show();  
imageUploaded.setImageBitmap(bitmap);  
} catch (IOException e) {  
e.printStackTrace();  
Toast.makeText(PhotoUploaded.this, "فشل حفظ الصورة!", Toast.LENGTH\_SHORT).show();  
}  
}  
} else if (requestCode == CAMERA) {  
Bitmap thumbnail = (Bitmap) data.getExtras().get("data");  
imageUploaded.setImageBitmap(thumbnail);  
saveImage(thumbnail);  
Toast.makeText(PhotoUploaded.this, " تم حفظ الصورة!", Toast.LENGTH\_SHORT).show();  
}  
}  
public String saveImage(Bitmap myBitmap) {  
ByteArrayOutputStream bytes = new ByteArrayOutputStream();  
myBitmap.compress(Bitmap.CompressFormat.JPEG, 90, bytes);  
File wallpaperDirectory = new File(  
Environment.getExternalStorageDirectory() + IMAGE\_DIRECTORY);  
if (!wallpaperDirectory.exists()) {  
wallpaperDirectory.mkdirs();  
}  
try {  
File f = new File(wallpaperDirectory, Calendar.getInstance()  
.getTimeInMillis() + ".jpg");  
f.createNewFile();  
FileOutputStream fo = new FileOutputStream(f);  
fo.write(bytes.toByteArray());  
MediaScannerConnection.scanFile(this,  
new String[]{f.getPath()},  
new String[]{"image/jpeg"}, null);  
fo.close();  
Log.d("TAG", "File Saved::---&gt;" + f.getAbsolutePath());  
return f.getAbsolutePath();  
} catch (IOException e1) {  
e1.printStackTrace();  
}  
return "";  
}  
private void requestMultiplePermissions() {  
Dexter.withActivity(this)  
.withPermissions(  
Manifest.permission.CAMERA,  
Manifest.permission.WRITE\_EXTERNAL\_STORAGE,  
Manifest.permission.READ\_EXTERNAL\_STORAGE)  
.withListener(new MultiplePermissionsListener() {  
@Override  
public void onPermissionsChecked(MultiplePermissionsReport report) {  
//check if all permissions are granted  
if (report.areAllPermissionsGranted()) {  
Toast.makeText(getApplicationContext(), "All permissions are granted by user!", Toast.LENGTH\_SHORT).show();  
}  
}  
@Override  
public void onPermissionRationaleShouldBeShown(List<PermissionRequest> permissions, PermissionToken token) {  
token.continuePermissionRequest();  
}  
}).  
withErrorListener(new PermissionRequestErrorListener() {  
@Override  
public void onError(DexterError error) {  
Toast.makeText(getApplicationContext(), "Error! ", Toast.LENGTH\_SHORT).show();  
}  
})  
.onSameThread()  
.check();  
}//end of requestMultiplePermissions()  
public void setToolbar() {  
toolbar = findViewById(R.id.add\_product\_toolbar);  
setSupportActionBar(toolbar);  
//setting back button  
Window window = this.getWindow();  
window.setStatusBarColor(ContextCompat.getColor(this, R.color.colorPrimaryDark));  
getSupportActionBar().setDisplayHomeAsUpEnabled(true);//setting the tool bar  
getSupportActionBar().setDisplayShowHomeEnabled(true);//making it showing  
toolbar.setNavigationOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View v) {  
onBackPressed();//going automatically to the previous page  
}  
});  
}//end of setToolbar  
}//end of class

Cart.java:

The cart class is configured variables such as product name and price and image and setter and getter methods to access the variables later.

public class Cart {  
 private String productName;  
 private String productPrice;  
 private byte[] productPic;  
 public Cart( String productName, String productPrice, byte[] productPic) {  
 this.productName = productName;  
 this.productPrice = productPrice;  
 this.productPic = productPic; }  
 public void setProductName(String productName) {  
 this.productName = productName;  
 }  
  
 public void setProductPrice(String productPrice) {  
 this.productPrice = productPrice;  
 }  
  
 public void setProductPic(byte[] productPic) {  
 this.productPic = productPic;  
 }  
 public String getProductName() {  
 return productName;  
 }  
  
 public String getProductPrice() {  
 return productPrice;  
 }  
  
 public byte[] getProductPic() {  
 return productPic;  
 }}

AddToCart.java:

1. Three variables define ArrayList<Cart> shopping\_list and Listview list and CartListAdapter Adapter.
2. get values from ProductListAdapter using getExtras.
3. Add values to shopping\_list.
4. Create a new adapter of type CartListAdapter and gives cart\_item layout and shopping\_list.
5. Finally, pass the adapter to listView.
6. import android.content.Intent;  
   import android.os.Bundle;  
   import android.widget.ListView;  
   import android.widget.Toast;  
   import androidx.appcompat.app.AppCompatActivity;  
   import java.util.ArrayList;  
     
   public class AddToCart extends AppCompatActivity {  
    ArrayList<Cart>shopping\_list ;  
    ListView list;  
    CartListAdapter Adapter;  
     
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
     
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.*activity\_cart*);  
    shopping\_list = new ArrayList<>();  
    list = (ListView) findViewById(R.id.*list*);  
     
     
    Intent intent =getIntent();  
    if(intent.getExtras()!=null){  
    String name =intent.getStringExtra("name");  
    String price = intent.getStringExtra("price");  
    byte [] pic=intent.getExtras().getByteArray("pic");  
    shopping\_list.add(new Cart(name,price,pic));  
    Toast.*makeText*(this,name+price,Toast.*LENGTH\_LONG*).show();  
    Adapter = new CartListAdapter(this, R.layout.*cart\_item*, shopping\_list);  
    list.setAdapter(Adapter); }}}

CartListAdabter.java:

1. create adapter for listview .
2. Three variables define ArrayList<Cart> product\_list and layout and context .
3. Creating a contractor takes three variables.
4. Create a ViewHolder class that takes the values in cart\_item.xml.

import android.content.Context;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
import java.util.ArrayList;  
  
public class CartListAdapter extends BaseAdapter {  
  
 private Context context;  
 private int layout;  
 private ArrayList<Cart> productList;  
  
 public CartListAdapter(Context context, int layout, ArrayList<Cart> productList) {  
 this.context = context;  
 this.layout = layout;  
 this.productList = productList; }  
  
 @Override  
 public int getCount() {  
 return productList.size();  
 }  
  
 @Override  
 public Object getItem(int position) {  
 return productList.get(position);  
 }  
  
 @Override  
 public long getItemId(int position) {  
 return position;  
 }  
  
 private class ViewHolder {  
 ImageView imageView;  
 TextView name, price;  
 Button add\_to\_cart;  
 }  
  
 @Override  
 public View getView(final int position, View convertView, ViewGroup parent) {  
 View row = convertView;  
 ViewHolder holder = new ViewHolder();  
  
 if (row == null) {  
 LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 row = inflater.inflate(R.layout.*cart\_item*, null);  
  
 holder.name = (TextView) row.findViewById(R.id.*product\_name*);  
 holder.price = (TextView) row.findViewById(R.id.*product\_price*);  
 holder.imageView = (ImageView) row.findViewById(R.id.*product\_picImg*);  
 row.setTag(holder);  
 } else {  
 holder = (CartListAdapter.ViewHolder)row.getTag();  
 }  
  
 Cart product = productList.get(position);  
 holder.name.setText(product.getProductName());  
 holder.price.setText(product.getProductPrice() + "SR");  
  
 byte[] image = product.getProductPic();  
 Bitmap bitmap = BitmapFactory.*decodeByteArray*(image, 0, image.length);  
 holder.imageView.setImageBitmap(bitmap);  
  
 return row; }}

activtiy\_cart.xml:

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:padding="10dp"  
 tools:context=".AddToCart"  
 app:theme="@style/Theme.Design.Light.NoActionBar">  
  
  
 <ListView  
 android:id="@+id/list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="fill\_parent"  
 android:layout\_above="@+id/bill"  
 android:gravity="center"  
 android:layout\_below="@+id/book\_category\_toolbar"  
 />  
  
  
 <Button  
 android:id="@+id/bill"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginStart="0dp"  
 android:layout\_marginBottom="1dp"  
 android:background="@color/colorPrimaryDark"  
 android:text="فاتورتي"  
 android:textAllCaps="true"  
 android:textColor="@android:color/white"  
 android:textSize="25sp"  
 android:textStyle="bold" />  
  
  
</RelativeLayout>

cart\_item.xml :

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:orientation="vertical">  
  
  
 <ImageView  
 android:id="@+id/product\_picImg"  
 android:layout\_width="199dp"  
 android:layout\_height="182dp"  
 android:layout\_marginTop="20dp"  
 app:srcCompat="@drawable/pngbastah" />  
  
 <TextView  
 android:id="@+id/product\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="TextView"  
 android:layout\_marginTop="20dp"  
 android:textColor="@color/black"  
 android:textSize="20sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/product\_price"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="textView"  
 android:layout\_marginTop="20dp"  
 android:textColor="@color/colorAccent"  
 android:textSize="20sp" />

*Sale.xml:*

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 >  
 <ImageView  
 android:id="@+id/imageView4"  
 android:layout\_width="149dp"  
 android:layout\_height="162dp"  
 android:layout\_marginStart="130dp"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginEnd="130dp"  
 android:layout\_marginBottom="14dp"  
 android:background="@drawable/pngbastah"  
 />  
  
  
 <TextView  
 android:id="@+id/thankstext"  
 android:layout\_width="409dp"  
 android:layout\_height="69dp"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginTop="200dp"  
 android:layout\_marginEnd="0dp"  
 android:text="شكرًا على ثقتك في بسطة!"  
 android:textAlignment="center"  
 android:textColor="@color/colorAccent"  
 android:textSize="35sp" />  
  
 <TextView  
 android:id="@+id/pilltext"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:textColor="@color/colorPrimaryDark"  
 android:textSize="30sp"  
 android:textStyle="bold"  
 android:text="فاتورة"  
 android:layout\_below="@id/thankstext"  
 />  
<RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginHorizontal="15dp"  
 android:layout\_below="@id/pilltext">  
  
 <LinearLayout  
 android:id="@+id/userlayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="right"  
 android:orientation="horizontal" >  
  
 <TextView  
 android:id="@+id/username"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/colorPrimaryDark"  
 android:textSize="25sp"  
 android:textStyle="normal"  
  
 />  
 <TextView  
 android:id="@+id/usernamelabel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="المشتري: "  
 android:textColor="@color/colorPrimaryDark"  
 android:textSize="25sp"  
 android:textStyle="bold"  
  
 />  
  
 </LinearLayout>  
 <LinearLayout  
 android:id="@+id/datelayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/userlayout"  
 android:gravity="right"  
 android:orientation="horizontal" >  
  
 <TextView  
 android:id="@+id/saledate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/colorPrimaryDark"  
 android:textSize="25sp"  
 android:textStyle="normal"  
 />  
 <TextView  
 android:id="@+id/datelabel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="تاريخ الشراء: "  
 android:textColor="@color/colorPrimaryDark"  
 android:textSize="25sp"  
 android:textStyle="bold"  
  
 />  
  
 </LinearLayout>  
 <TextView  
 android:id="@+id/itemstext"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/datelayout"  
 android:gravity="right"  
 android:textColor="@color/colorPrimaryDark"  
 android:text="الأغراض:"  
 android:textSize="25sp"  
 android:textStyle="bold"  
  
 />  
 <RelativeLayout  
 android:id="@+id/itemlayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/itemstext"  
 android:gravity="right" >  
  
 <TextView  
 android:id="@+id/productname"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentRight="true"  
 android:textColor="@color/black"  
 android:textSize="25sp"  
 android:textStyle="normal"  
  
 />  
 <TextView  
 android:id="@+id/saleprice"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:textColor="@color/black"  
 android:textSize="25sp"  
 android:textStyle="normal"  
  
 />  
 </RelativeLayout>  
  
 <View  
 android:layout\_width="match\_parent"  
 android:id="@+id/line5"  
 android:layout\_height="1dp"  
 android:layout\_below="@id/itemlayout"  
 android:background="@color/colorPrimaryDark" />  
  
 <RelativeLayout  
 android:id="@+id/totallayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/line5" >  
 <TextView  
 android:id="@+id/totalprice"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/colorPrimaryDark"  
 android:layout\_alignParentLeft="true"  
 android:textSize="25sp"  
 android:textStyle="bold"  
 />  
 <TextView  
 android:id="@+id/totallabel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="المجموع: "  
 android:layout\_alignParentRight="true"  
 android:textColor="@color/colorPrimaryDark"  
 android:textSize="25sp"  
 android:textStyle="bold"  
  
 />  
  
 </RelativeLayout>  
</RelativeLayout>  
</RelativeLayout>

Saleactivity.java:

First we using SharedPreferences with user name to allow access to username , and extract product information from the intent, then use it to add to the edit text.

package com.example.project;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Locale;  
  
public class saleactivity extends AppCompatActivity {  
 TextView usernameact;  
 TextView productnameact;  
 TextView salepriceact;  
 TextView saledateact;  
 TextView totalprice;  
  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*sale*);  
 usernameact = findViewById(R.id.*username*);  
 productnameact= findViewById(R.id.*productname*);  
 salepriceact= findViewById(R.id.*saleprice*);  
 saledateact= findViewById(R.id.*saledate*);  
 totalprice = findViewById(R.id.*totalprice*);  
  
 //aread user name  
 SharedPreferences sharedPreferences = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 String username = sharedPreferences.getString("username", "default");  
  
 String name = "";  
 String price = "";  
 String date = "";  
 Intent intent =getIntent();  
 if(intent.getExtras()!=null){  
 name =intent.getStringExtra("name");  
 price = intent.getStringExtra("price");  
 date = new SimpleDateFormat("yyyy-MM-dd", Locale.*getDefault*()).format(new Date());  
  
 }  
  
 usernameact.setText(username);  
 productnameact.setText(name);  
 salepriceact.setText(price+"SR");  
 totalprice.setText(price+"SR");  
 saledateact.setText(date);  
  
  
  
  
  
  
  
  
 }  
}

# Result (output)

A close up of a logo

Description automatically generated

Splash activity with timer set to three seconds then go to next activity or once the user click anywhere in the screen

A screenshot of a cell phone

Description automatically generated

We have only one admin already singed in the application with username: **admin**, and password: **123**

A screenshot of a cell phone

Description automatically generated

**Admin Home page**: it has three navigation buttons one for viewing all sales, the second one is for viewing all sales, the third is for logging out.

A screenshot of a cell phone

Description automatically generated

Admin can view all customers information as a listView, and this is for communication with the customer if needed.

Then moving to the **customer**

A screenshot of a cell phone

Description automatically generated

Customer Sign in page, after entering information successfully it will show a toast message to inform the user that account is signed in successfully.

A screenshot of a cell phone

Description automatically generated

**Login page:** Error warning with explanation if the username or the password is wrong.

A picture containing clock

Description automatically generated

**Customer home page:** it contains a welcoming textView, 4 buttons that transfer the customer to the needed category and a menu below the page for easy transformation for profile, home page and settings.

The plus button is for adding new product to sell

A screenshot of a cell phone

Description automatically generated

**Customer profile page:** it contains a form to get customers info.

A screenshot of a cell phone

Description automatically generated

**Customer profile page:** Toast message that information editted successfully.

A screenshot of a cell phone

Description automatically generated

**Customer settings page:** Notifications settings and logout buttom.

A screenshot of a cell phone

Description automatically generated

**Customer settings page:** alertDialog to ensure that the user want to logout.

A screenshot of a cell phone

Description automatically generated

**Add product page:** it allows uploading image with two way, from photo saved in device or from camera

A screenshot of a cell phone

Description automatically generated

**Add product page:** ask user for permission before acess camera, or other media.

A picture containing flower

Description automatically generated

**Add product page:** camera permission

A picture containing indoor, ceiling, room, sitting

Description automatically generated

**Add product page:** taking picture through emulator’s camera looks like this.

A screenshot of a cell phone

Description automatically generated

**Add product page:** if the user chose picture from gallery.

With snack bar warning if the user didn’t choose category

A screenshot of a cell phone

Description automatically generated

**Customer Electronics category:** shows the products that are related to specific category and allow the user to add in cart.

A screenshot of a cell phone

Description automatically generated

**Customer cart page:** shows the products added in cart.

**Customer Add product:** Customer add product pircture, product name , product price , and product section.

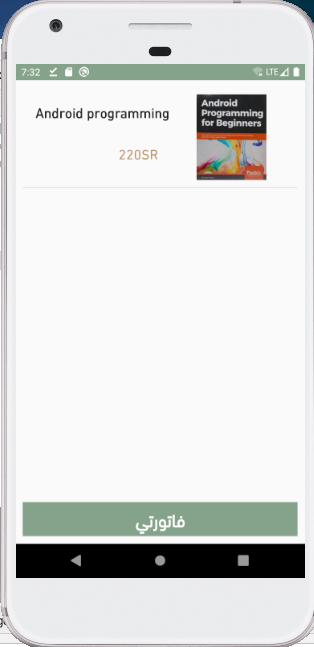
A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Customer added product:** After add product in the book section



**Customer choose product:** After customer added one product in the Cart.



**Customer Bill:** Which include Customer name, date , products , and price.

# Conclusion

Smartphones applications are growing fast. These digital devices represent the formation of technologies that can be used to make our lives much easier. While there are thousands of applications stores available today, not many if they thought of saving our planet and supporting the re-using concept. Therefore, choosing the store application is confusing for users and challenging for mobile applications developers, offering such as (بسطة) application with our native language will be such an advantage.

In the design phase, we took into consideration that all ages should be benefited from (بسطة) application, so the design was simple, clear and comforts the eyes.

So, we conclude that we have a user-friendly application which System allows customers to sell their unneeded used products and buy things they need from other customers as well.

## Future work:

* Add chatting capability.
* Transulate the application to more languages.

# Reference

1. Fan, C., An, H., Du, J. and Luo, Y., 2019. High‐performance printable paper‐like composites derived from plastic flexible film wastes. *Polymer International*, 69(2), pp.184-191.
2. *Environmental Pollution Series A, Ecological and Biological*, 1984. Air pollution—Physiological effects. 34(2), p.194.
3. Clark, R., 1970. Biological effects of pollution in the sea. *Marine Pollution Bulletin*, 1(6), p.95.
4. Bell, J., 1997. Pollution: Causes, effects and control. *Environmental Pollution*, 96(2), pp.276-277.
5. Encyclopedia Britannica. 2020. *Pollution | Definition, History, & Facts*. [online] Available at: <https://www.britannica.com/science/pollution-environment> [Accessed 27 March 2020].
6. Ecube Labs. 2020. *Overflowing Garbage Bins: 5 Impacts On Health And Environment, And How To Prevent | Ecube Labs*. [online] Available at: <https://www.ecubelabs.com/overflowing-garbage-bins-5-impacts-on-health-and-environment-and-how-to-prevent/> [Accessed 27 March 2020].
7. Inspiration, A., 2020. *Top 7 Reasons To Buy Used Goods - Resilience*. [online] Resilience. Available at: <https://www.resilience.org/stories/2008-11-20/top-7-reasons-buy-used-goods/> [Accessed 27 March 2020].
8. Sqlite.org. 2020. *Full-Featured SQL*. [online] Available at: <https://www.sqlite.org/fullsql.html> [Accessed 27 March 2020].
9. ElProCus - Electronic Projects for Engineering Students. 2020. *What Is Android? Introduction Of Android OS & It's Applications*. [online] Available at: <https://www.elprocus.com/what-is-android-introduction-features-applications/> [Accessed 27 March 2020].
10. How-To Geek. 2020. *What’S The Latest Version Of Android?*. [online] Available at: <https://www.howtogeek.com/345250/whats-the-latest-version-of-android/> [Accessed 27 March 2020].
11. SearchMobileComputing. 2020. *What Is Android Studio? - Definition From Whatis.Com*. [online] Available at: <https://searchmobilecomputing.techtarget.com/definition/Android-Studio> [Accessed 27 March 2020].
12. Android Developers. 2020. *Download Android Studio And SDK Tools  |  Android Developers*. [online] Available at: <https://developer.android.com/studio> [Accessed 27 March 2020].
13. Techopedia.com. 2020. *What Is A Mobile Application? - Definition From Techopedia*. [online] Available at: <https://www.techopedia.com/definition/2953/mobile-application-mobile-app> [Accessed 28 March 2020].
14. Lifewire. 2020. *What's A Mobile App?*. [online] Available at: <https://www.lifewire.com/what-is-a-mobile-application-2373354> [Accessed 28 March 2020].
15. Sciencedirect.com. 2020. *Mobile Application - An Overview | Sciencedirect Topics*. [online] Available at: <https://www.sciencedirect.com/topics/computer-science/mobile-application> [Accessed 28 March 2020].
16. US EPA. 2020. *Reducing And Reusing Basics | US EPA*. [online] Available at: <https://www.epa.gov/recycle/reducing-and-reusing-basics> [Accessed 28 March 2020].
17. Department of Environmental Protection. 2020. *What About Reuse*. [online] Available at: <https://www.dep.pa.gov/Citizens/JustForKids/Recycling/Pages/Reuse.aspx> [Accessed 28 March 2020].