

Kingdom of Saudi Arabia
Majmaah University
Ministry of Higher Education
College of Science Al Zulfi

المملكة العربية السعودية
جامعة المجمعة
وزارة التعليم العالي
كلية العلوم بالزلفي



Perfume mixer android app

College of science Al Zulfi
Department of Computer Science and Information

Graduation Project

Submitted in partial fulfillment of the requirements for the award of
Bachelor degree of the Majmaah University
(Semester 2, 2019-20)

Submitted by:

Student: Khaled Alameri **ID:** 351101358

Student: Faisal Almutiri **ID:** 351101574

Under the supervision of:
Dr.Yousef Qawqzeh

Abstract

Perfume mixer is android app for the clients to can choose the perfume mixing by one or more of materials.

Client can choose the type of perfume and choose the materials included after that the system will calculate the price for it and client will pay it .

Do you like playing with scents? Do not know which scent suits you? This app has been created for you. Answer a few simple questions and find out fragrances that perfectly match your personality.

Acknowledgement

Thanks for all, thanks for all staff in college, special thanks for our supervisor. When we need help always we find the person standing by our side, and give us all information and guidance to achieve this project, this person is our supervisor.

Table of Contents

Abstract	ii
Acknowledgement	iii
Tables	2
Introduction	4
CHAPTER ONE: INTRODUCTION	5
1.1 Problem Definition.....	5
1.1.1 Goals	6
1.1.2 Scope.....	7
1.1.3 Objectives.....	7
1.1.4 Critical Success Factors	8
1.2 General rules	8
1.3 Collection of Data	8
1.4 Literature review	8
1.5 Feasibility Study	8
Perfume mixer android app	8
1.5.1 uniquefragrance.....	11
Advantages of Perfume's Club App	12
1.5.2 Comparison between similar systems	12
1.6 Functional and Non-functional Requirements	13
1.6.1 Functional requirements.....	13
1.6.2 Non-functional requirement	13
1.7 Design Model.....	14
1.7.1 Benefits of using Waterfall [5]:	15
1.7.2 Phases of Waterfall Model:.....	15
Chapter Two: System Analysis and Specification	17
2.1 Introduction	17
2.1.1 Overview diagram (level 0)	17
2.2 Data Modeling.....	17
2.2.1 Entity Relationship Diagram (ERD).....	17
2.2.2 Use case Diagram.....	21
2.2.3 Sequence Diagrams.....	22
Chapter 3: System Design.....	22
3.3 Hardware and software requirements.....	23

3.3.1	Hardware Requirements.....	23
3.3.2	Software requirements	24
3.4	Expected interfaces	25
3.4.1	Hardware Requirements.....	32
3.4.2	Software Requirements	33
Chapter 4	34
Introduction	34
4.1	Introduction	34
4.2	Procedures	34
Chapter 5	47
FUTURE ENHANCEMENTS:	47
Conclusion	48
References	49

Table Of Figures

Figure 1	uniquefragrance	11
Figure 2	Waterfall Model.....	15
Figure 3	Dataflow diagram	17
Figure 4	ERD Diagram	19
Figure 5	Database diagram	20
Figure 6	General use case diagram	21
Figure 7	sequence diagram	22
Figure 8	flowchart.....	23
Figure 9	windows 10.....	24
Figure 10	Microsoft Word	24
Figure 11	Android studio	24
Figure 12	PhotoShop.....	25
Figure 13	MySql.....	25
Figure 14	Home Screen.....	26
Figure 15	intro 1	27
Figure 16	Intro 2	28
Figure 17	Login Interface	29
Figure 18	signup interface.....	30
Figure 19	products interface	31
Figure 20	bill interface	32

Tables

Table 1	9
Table 2	9
Table 3	9

Table 4.....	10
Table 5.....	10
Table 6.....	11
Table 7	11
Table 8.....	12
Table 9.....	24
Table 10.....	45
Table 11.....	45
Table 12.....	46
Table 13.....	46
Table 14.....	46

Introduction

Perfume mixer is android app for the clients to can choose the perfume mixing by one or more of materials.

Client can choose the type of perfume and choose the materials included after that the system will calculate the price for it and client will pay it .

Do you like playing with scents? Do not know which scent suits you? This app has been created for you. Answer a few simple questions and find out fragrances that perfectly match your personality.

CHAPTER ONE: INTRODUCTION

1.1 Problem Definition

Undoubtedly when you want buy mixing perfume you have to go to the shop place and choose the materials , you will lose the time and this the problem in time so to provide all this time we create system to buy and sell mixing perfume so the client can login to the app and choose the materials based on the seller materials.

This app solve the problem related by time and the materials provided because maybe you want to buy some of special perfume that is not available or it take time to be in stock, this app give you the answer at this question if the material is available or not.

1.1.1 Goals

The main goal is to develop mobile application.

- The perfume mixer is android app can run to serve only the products need to mixing especially the perfumes.
- It dependent at 2 level of actor's clients, admin.
- Admin is the owner to the shop that he put the price and give each material its features and price.
- Client he is the user that choose the materials and mix it together to buy it.

1.1.2 Scope

This product has great future scope. Perfume mixer android app. This project also provides security with the use of Login-id and Password, so that any unauthorized users can not use your account. The only Authorized that will have proper access authority can access the software.

1.1.3 Objectives

The proposed system aims at achieving the following objectives.

1. The perfume mixer is android app can run to serve only the products need to mixing especially the perfumes.
2. It dependent at 2 level of actors clients, admin
3. Admin is the owner to the shop that he put the price and give each material its features and price.
4. Client he is the user that choose the materials and mix it together to buy it .
5. Enable admin to get notification in case the client has left order.

1.1.4 Critical Success Factors

In the markets of Saudi Arabia, the families come to Shop some day to buy their perfumes. According to the applicable regulations, the people don't know if the martials they need in perfume is available or not before they go.

Computer has become small and small with each passing year, and nowadays it is available inside a smart phone. In addition, Internet connections travel with people anywhere any time.

People use mobile applications in various areas of life, such as education, health, at work, shopping, and entertainment.

According to statista.com, there will be more than 24 million smart phone users in Saudi Arabia by 2020, ranked third in the world in terms of smart phone users, which helps in the

1.2 General rules

This application will be developed for smartphones that run Android operating system. Another platform like IOS (iPhone) will not be supported at the first stage of the project.

The project assumes that both client and Seller have android smart phone with ability to access Internet.

1.3 Collection of Data

Questionnaires and surveys can be efficient ways to collect requirements as long as key stakeholders provide honest and thorough information [1].

A questionnaire is a search tool consisting of a set of questions with the aim of collecting information from the stockholders of the proposed projects. In the questionnaire, the project requirements must be translated into questions that the participant must answer. The questions must be clear and direct to minimize the response error, and it could be distributed as paper or online.

In this project, the questionnaire was designed and distributed on papers to the university students. There were 33 responses to the questionnaire.

From the results of the questionnaire, we found that all the respondents have not dealt with electronic buy system before. 90% of the respondents prefer to use computer system to organize the perfume mixer. 90% of the respondents prefer the system to be on computer, while 90% prefer mobile application. With regards to the language of the system, we found that 42% of the respondents prefer buy perfume online, 45% prefer buy from shop directly,. Finally, all the respondents stated that they will use the proposed application when it is developed. Full questionnaire will be placed in the appendix.

1.4 Literature review

There are some projects like the proposed project. We studied some of them and found the advantages and disadvantages of each project.

1.5 Feasibility Study

Perfume mixer android app

Table 1

1. هل الفكرة قابلة للتنفيذ						نسبة الإجابات	إجمالي الإجابات	
1	نعم					90.91%	30	
2	لا					3.03%	1	
3	لا اعرف					6.06%	2	
تحليل	متوسط:	1.15	انحراف معياري:	0.5	معدل الرضا:	7.58	تم الإجابة عليه	33
	اختلاف:	0.25	خطأ معياري:	0.09			تم تخطيه	0

Table 2

2. هل يساعد التطبيق في ازدهار صناعة العطور في المملكة						نسبة الإجابات	إجمالي الإجابات	
1	نعم					84.85%	28	
2	لا					3.03%	1	
3	لا اعرف					12.12%	4	
تحليل	متوسط:	1.27	انحراف معياري:	0.66	معدل الرضا:	13.64	تم الإجابة عليه	33
	اختلاف:	0.44	خطأ معياري:	0.12			تم تخطيه	0

Table 3

3. ماهى المميزات المقدمة للعمل من خلال التطبيق						نسبة الإجابات	إجمالي الإجابات
1	اسعار منافسة					45.45%	15
2	دقة الاختيار					24.24%	8
3	توفير الوقت والمجهود					48.48%	16
4	تنوع فى العروض					36.36%	12

3. ماهى المميزات المقدمة للعمل من خلال التطبيق

		نسبة الإجابات	إجمالي الإجابات
5	جودة افضل		24.24% 8
6	كل ماسبق		54.55% 18
7	لاشئ مما سبق		3.03% 1
تحليل	متوسط:	8.55	انحراف معياري: 8.08
	اختلاف:	65.34	خطأ معياري: 1.41
		تم الإجابة عليه	33
		تم تخطيه	0

Table 4

4. هل تشهد صناعة العطور فى المملكة ازدهار فى العشر اعوام الماضية

		نسبة الإجابات	إجمالي الإجابات
1	نعم		69.70% 23
2	لا		3.03% 1
3	لا اعرف		27.27% 9
تحليل	متوسط:	1.58	انحراف معياري: 0.89
	اختلاف:	0.79	خطأ معياري: 0.15
		تم الإجابة عليه	33
		تم تخطيه	0

Table 5

5. هل يمكن فتح اسواق جديدة خارج المملكة من خلال التطبيق

		نسبة الإجابات	إجمالي الإجابات
1	نعم		68.75% 22
2	لا		3.13% 1
3	لا اعرف		28.13% 9
تحليل	متوسط:	1.59	انحراف معياري: 0.9
	اختلاف:	0.8	خطأ معياري: 0.16
		تم الإجابة عليه	32
		تم تخطيه	1

Table 6




6. هل تفضل شراء العطور من خلال الانترنت ام الذهاب الى المحل					نسبة الإجابات	إجمالي الإجابات
1	الذهاب الى المحل				45.45%	15
2	الشراء اون لاين				42.42%	14
3	لست مهتم				12.12%	4
تحليل	متوسط:	1.67	انحراف معياري:	0.68	معدل الرضا:	33.33
	اختلاف:	0.46	خطأ معياري:	0.12		
					تم الإجابة عليه	33
					تم تخطيه	0

Table 7

1.5.1 uniquefragrance

<http://www.uniquefragrance.com> is an integrated system that create your own fragrance.



Figure 1 uniquefragrance

Advantages of Unique fragrance System

1. Become a perfumer yourself and create your very own fragrance. Choose from 50 exclusive ingredients.
2. Design your own.
3. Saves time.
4. Have our fragrance experts develop your unique fragrance according to your wishes and personality.

5. Launch your own perfume series.

Disadvantages of Unique fragrance System

1. Expensive system.
2. Difficult to install, because it has multiple hardware equipment.

Advantages of Perfume's Club App

1. Scanned Products.
2. Shop.
3. Offers.
4. Direct access to perfumesclub.co.uk shop to buy our products.

Disadvantages of Perfume's Club App

1. It's not specified to shop.
2. Its related to perfumesclub.co.uk.
3. Its not work with mixer perfumes .

1.5.2 Comparison between similar systems

	Unique fragrance System	Perfume 's Club App	Perfume mixer Management System
Support shop mixer perfumes	Yes	No	Yes
Support arabic language	No	No	Yes
Hard to install	Yes	No	No
Acurate	Yes	No	No
Generate repots	Yes	Yes	Yes
Use athinitication	Yes	No	yes
Need to issue card	Yes	No	No
Cost	Heigh	free	Free
Easy to use	No	Yes	Yes

Table 8

1.6 Functional and Non-functional Requirements

System requirements are expressed in a software requirement document. The Software requirement specification (SRS) is the official statement of what is required of the system developers. This requirement document includes the requirements definition and the requirement specification. The software requirement document is not a design document. It should set out what the system should do without specifying how it should be done.

The requirement set out in this document is complete and consistent.

The software specification document satisfies the following:-

- 1 It specifies the external system behaviors.
- 2 It specifies constraints on the implementation.
- 3 It is easy to change.
- 4 It serves as reference tool for system maintainers.
- 5 It record forethought about the life cycle of the system.
- 6 It characterizes acceptable response to undesired events.

1.6.1 Functional requirements

- The System must provide following functionalities—
- Keeping records of admission of customers.
- keeping the records of products.
- keeping the daily sell .
- Storing the feedback given by the customer. keeping details about the product it is delivered or not. etc.
- Storing the items selected by the customer in the temporary storage.

1.6.2 Non-functional requirement

Following Non-functional requirements will be there in the Insurance on internet:

- i). Secure access of confidential data (customer's details).
- ii). 24 X 7 availability.
- iii). Better component design to get better performance at peak time.

iv). Flexible service based architecture will be highly desirable for future extension Non functional requirements define system properties and constraints

It arise through user needs, because of budget constraints or organizational policies, or due to the external factors such as safety regulations, privacy registration and so on.

Various other Non-functional requirements are:

- Security
- Reliability
- Maintainability
- Portability
- Extensibility
- Reusability
- Application Affinity/Compatibility
- Resource Utilization

1.7 Design Model

All projects go through several phases until they are ready to use. These phases include planning, requirements, analysis, design, implementation and testing. These different phases are called software development life cycle (SDLC). SDLC helps the developers design, create and deliver high quality software by defining various tasks that need to take place [2]

System development methodology is a framework that is used to structure, plan and control the process of developing projects [3]. Methodology is the way in which the SDLC phases are implemented.

There are many software methodologies, like Waterfall, Agile, Extreme Programming, Scrum and Spiral. Each methodology has advantages and disadvantages. In this project, the Waterfall methodology will be followed.

Waterfall is one of the simplest and fastest methodologies. It is a sequential design process, used in software development processes. It means it is not an-iterative process [4].

In Waterfall process, the stage of development life cycle is steadily downwards, when leaving one step, it's costly to return to it.

Waterfall is suitable for this project because project requirements are clear from the beginning of the development process, and the project developers (Group Team Members) are the owners of the project (Client). Therefore, there is no importance to communication between the client and the development team.

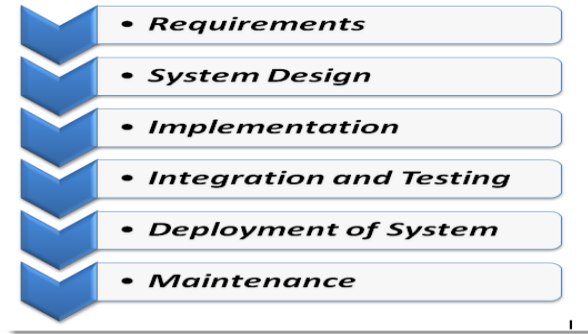


Figure 2 Waterfall Model

1.7.1 Benefits of using Waterfall [5]:

- It is suitable for small business because it does not need iteration.
- Waterfall model is simple to implement, and the amount of resources required for it are minimal.
- It provides a template, into which methods of analysis, design, coding, testing, and maintenance can be placed.
- This methodology is preferred in projects where quality is more important than schedule or cost.
- The work is divided into small projects where the requirements become easy to understand.

1.7.2 Phases of Waterfall Model:

- Feasibility study: In this phase of development, a feasibility study was carried out for the proposed project. The feasibility study includes the technical feasibility, which means there is a possibility to implement the application technically, the requirements of the proposed application are clear and could be implemented for android devices. Economic feasibility is also studied, which means that the system is economically feasible, because the system does not need extra investment to be run after development. Finally, the schedule feasibility was done to ensure that the system could be implemented in the available period.
- Requirements gathering and analysis: This phase focuses on the managers and stakeholders. Meetings with managers, stakeholders and users are held to determine the requirements. In this phase, we will create online questionnaire to collect the system requirements.
- Design: In design phase, defining overall system architecture is made depending on the requirements gathered in the previous analysis phase. In this phase, database of the system will be created, and then the expected interfaces of the proposed system must be created. In this phase, required hardware and software will be specified.
- Coding and unit testing: This phase is the responsibility of the developer, where the coding is starting, and actual application is developed. In this phase, the database must be implemented using MySQL server, then the actual system interfaces will be created using Android studio. It is official IDE from Google for developing

android application. In coding phase, Java programming language will be used. After the completion of coding, unit testing must be carried out. Unit testing includes testing of all modules of the application.

- **Integration and system testing:** After the code is developed, unit of the application is tested. The overall system must be tested to ensure that the application works as expected. In this phase, all proposed application functions will be tested, and output of each function will be verified to compare it with expected output. Also, all objectives of the project must be checked to ensure whether the proposed system achieves the planned goals.
- **Maintenance:** This is the last step of the project development phases. In this phase, the bugs and errors that appear in the testing phases must be fixed. Then the application will be ready for publishing.

Chapter Two: System Analysis and Specification

2.1 Introduction

System analysis is conducted with the aim of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. System analysis includes an understanding of software product limitations, learning system related problems or changes to be done.

2.1.1 Overview diagram (level 0)

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation [6].

2.1.2 General Overview diagram

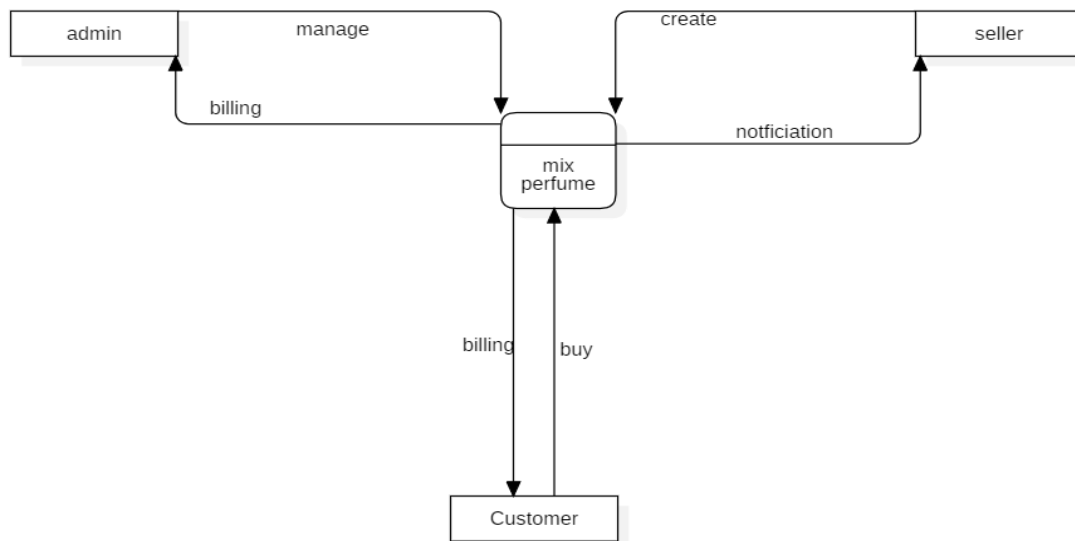


Figure 3 Dataflow diagram

2.2 Data Modeling

2.2.1 Entity Relationship Diagram (ERD)

Entity relationship model is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system [7].

2.2.1.1 Description of Entities

1. Seller:.create materials,manage bills,send the products.
2. admin:.manage users, provide gateways.
3. client:.buy products,search,register.
4. Notification: is the notification from customer ,admin,seller.

2.2.1.2 ERD Diagram

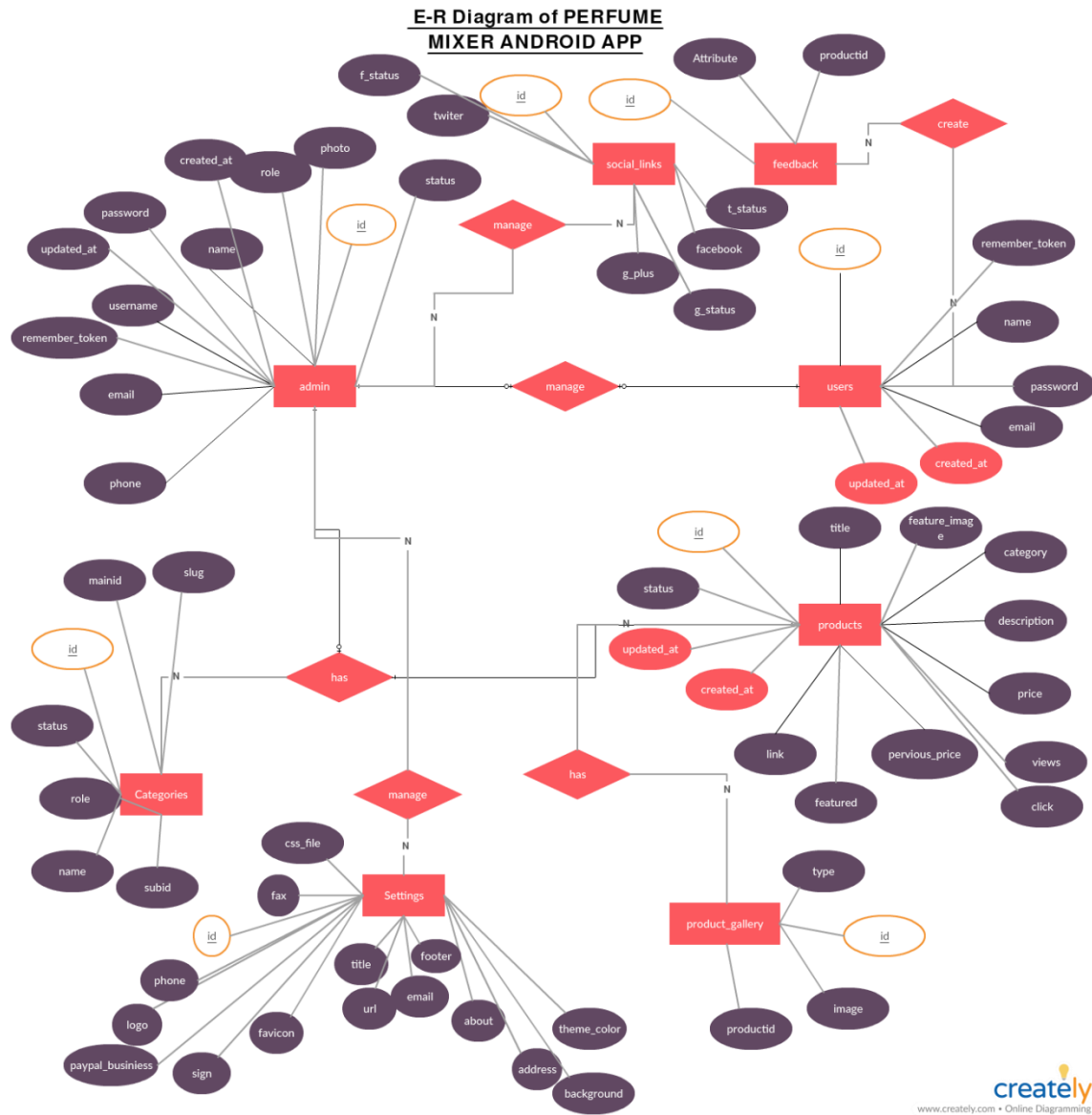


Figure 4 ERD Diagram

2.2.1.3 Data base diagram

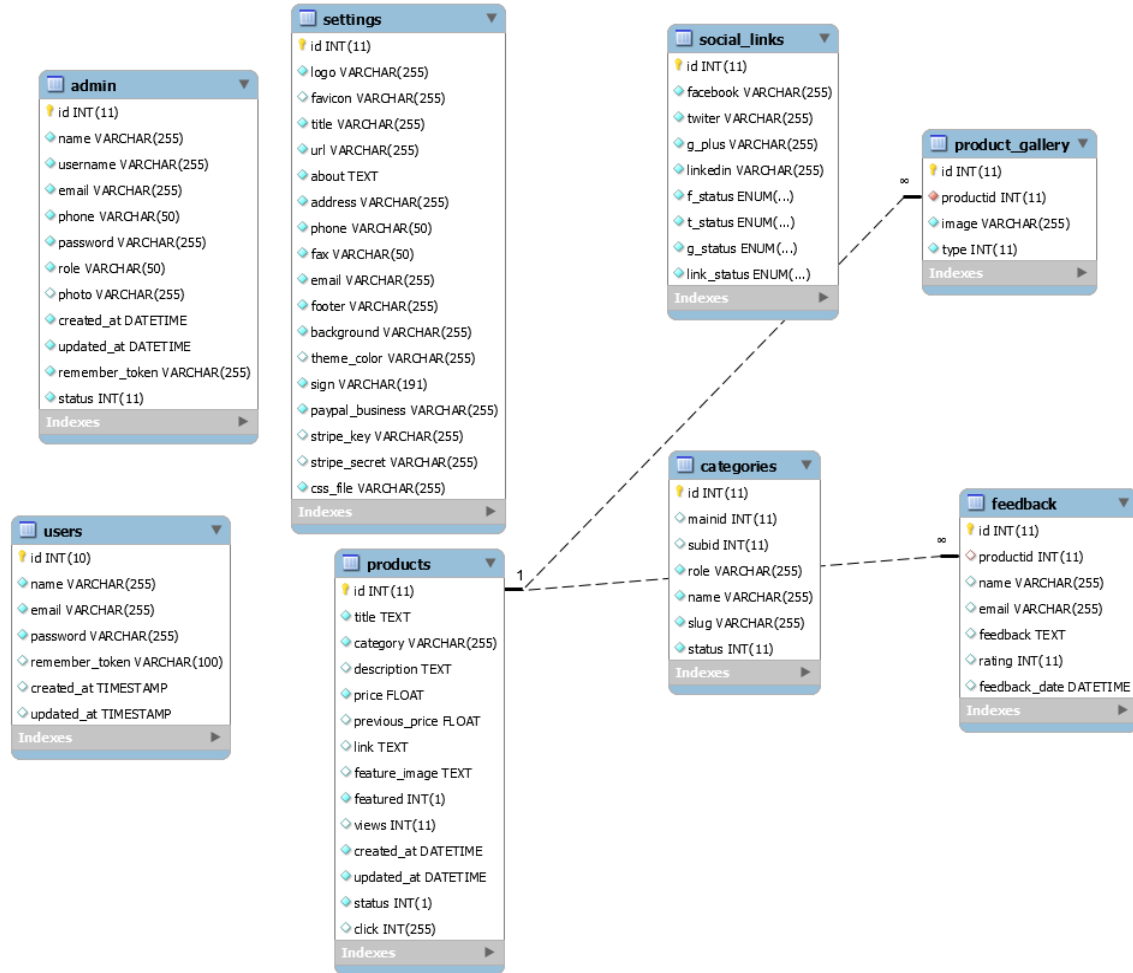


Figure 5 Database diagram

2.2.2 Use case Diagram

2.2.2.1 Definition:

Use case diagram is one of UML diagrams that are created for any software system. It gives a graphic overview of the actors involved in a system and represents different functions done by the actors of the system. It also shows how those different functions interact [8].

2.2.2.2 General use case diagram

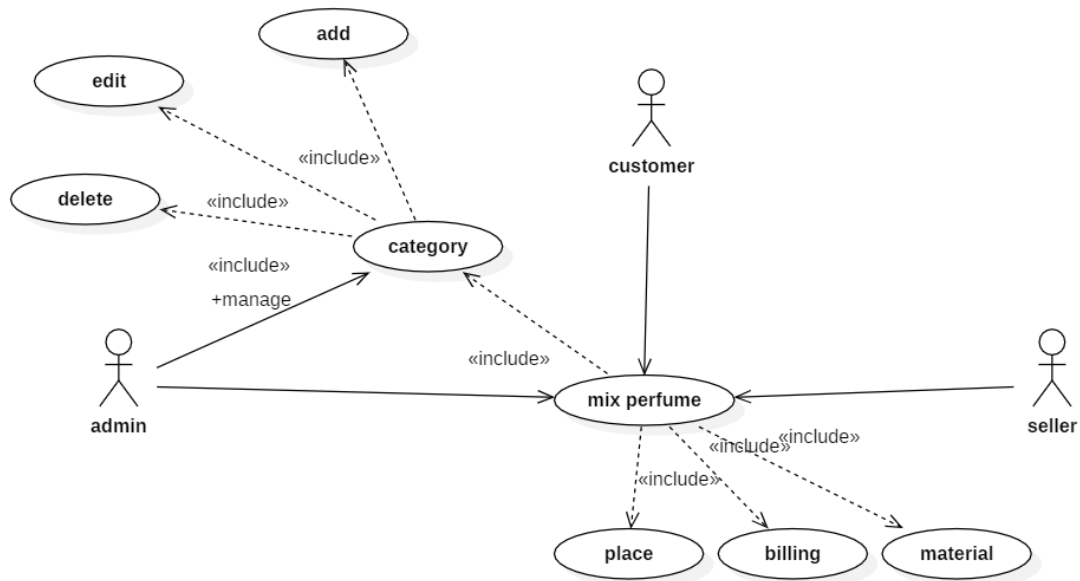


Figure 6 General use case diagram

2.2.3 Sequence Diagrams

2.2.3.1 Definition:

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

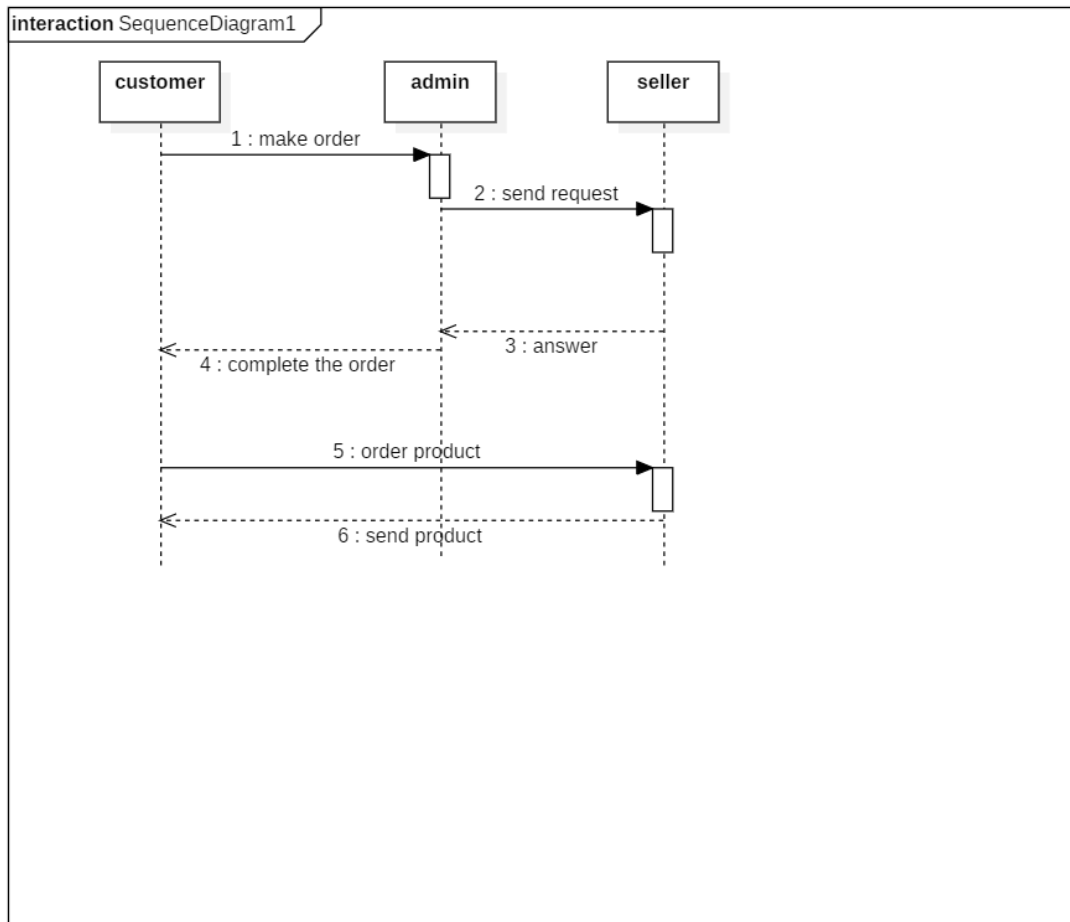


Figure 7 sequence diagram

Chapter 3: System Design

3.1 Description of procedures and functions

1. The seller and buyer create new account.

2. The seller adds his material.
3. The seller can also put offer.
4. When the buyer ask for material not found the admin can send to the seller to create proposal .
5. The system will check the distance between seller and buyer and send it as delivery or ask him to go to shop.
6. The buyer can reject the product.

3.2 Flowchart

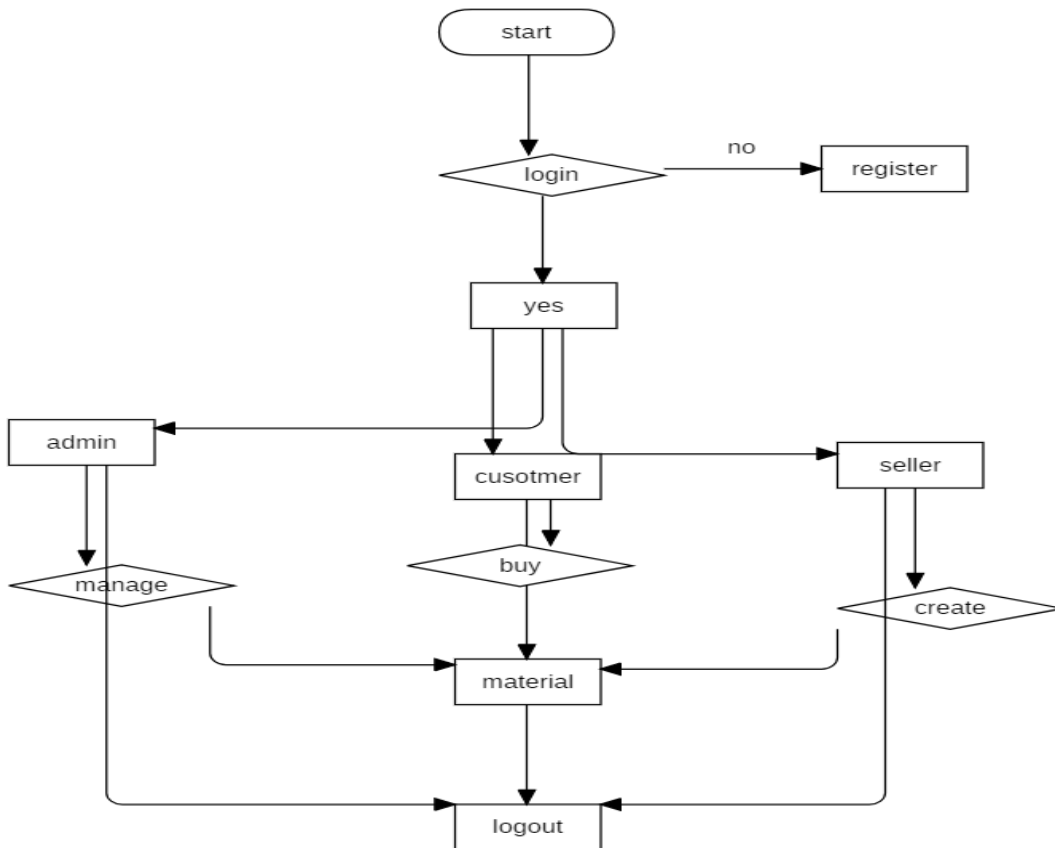


Figure 8 flowchart

3.3 Hardware and software requirements

Application will be developed as client server application for android smartphone. In order to complete this application, we need the following tools and programming languages.

3.3.1 Hardware Requirements

The specifications of the laptop that is needed to complete this project.

CPU	Core I5
RAM	4GB

Monitor	+14"
---------	------

Table 9

3.3.2 Software requirements

To complete this project, several programs are needed.

1. Windows 10: It is the operating system of the laptop that will be used to complete this project.



Figure 9 windows 10

2. Microsoft Word: Microsoft Word is text editor from Microsoft Company. The Word is the most used text editor over the world. It has powerful features and tools. In this project, the Word will be used for writing the report.



Figure 10 Microsoft Word

3. Android Studio: It is an application developing platform that makes it easy for developers to write source code for Android applications and allows developers to preview their application on various screens.



Figure 11 Android studio

4. Photoshop: It is a software program to create and modify photos. It is the most popular software for editing graphics and modifying digital photography.



Figure 12 PhotoShop

5. MySQL: It is a relational database management system that depends on the SQL address. This software is free and open source system. Usually, free software projects that require a database management system are used.



Figure 13 MySq

3.4 Expected interfaces

Home Screen



Figure 14 Home Screen

Intro 1

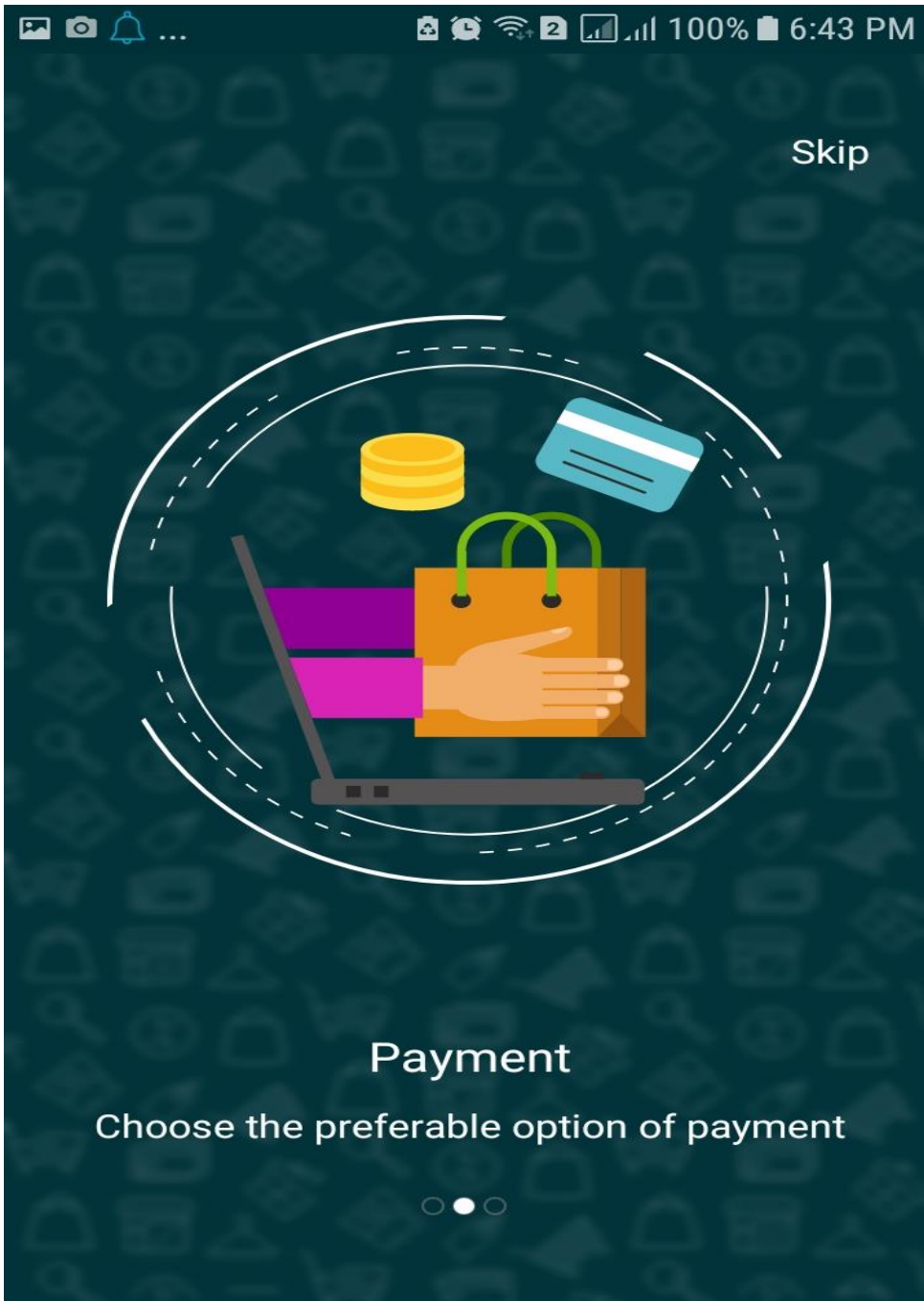


Figure 15 intro 1

Intro 2

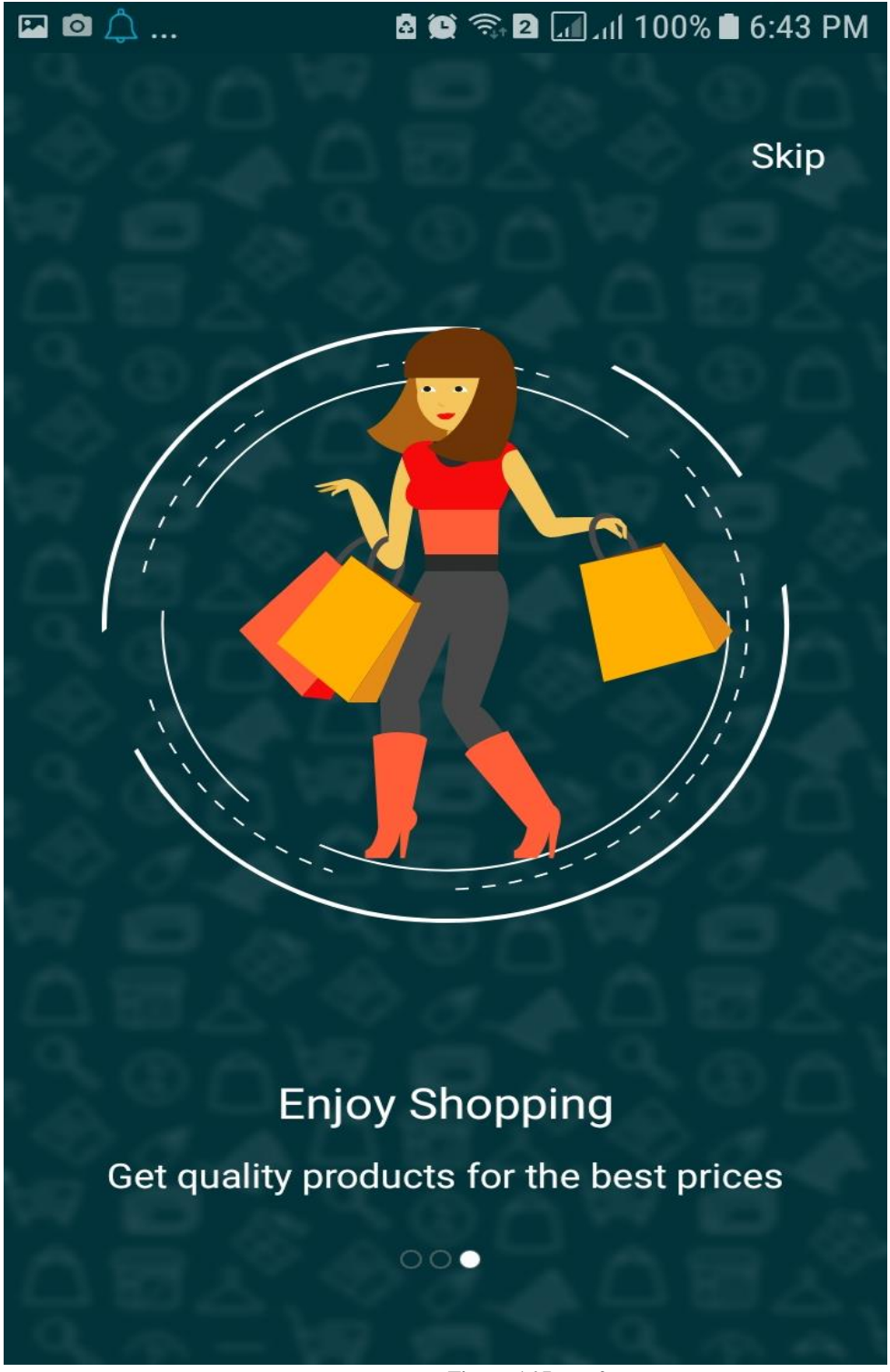


Figure 16 Intro 2

Login Interface

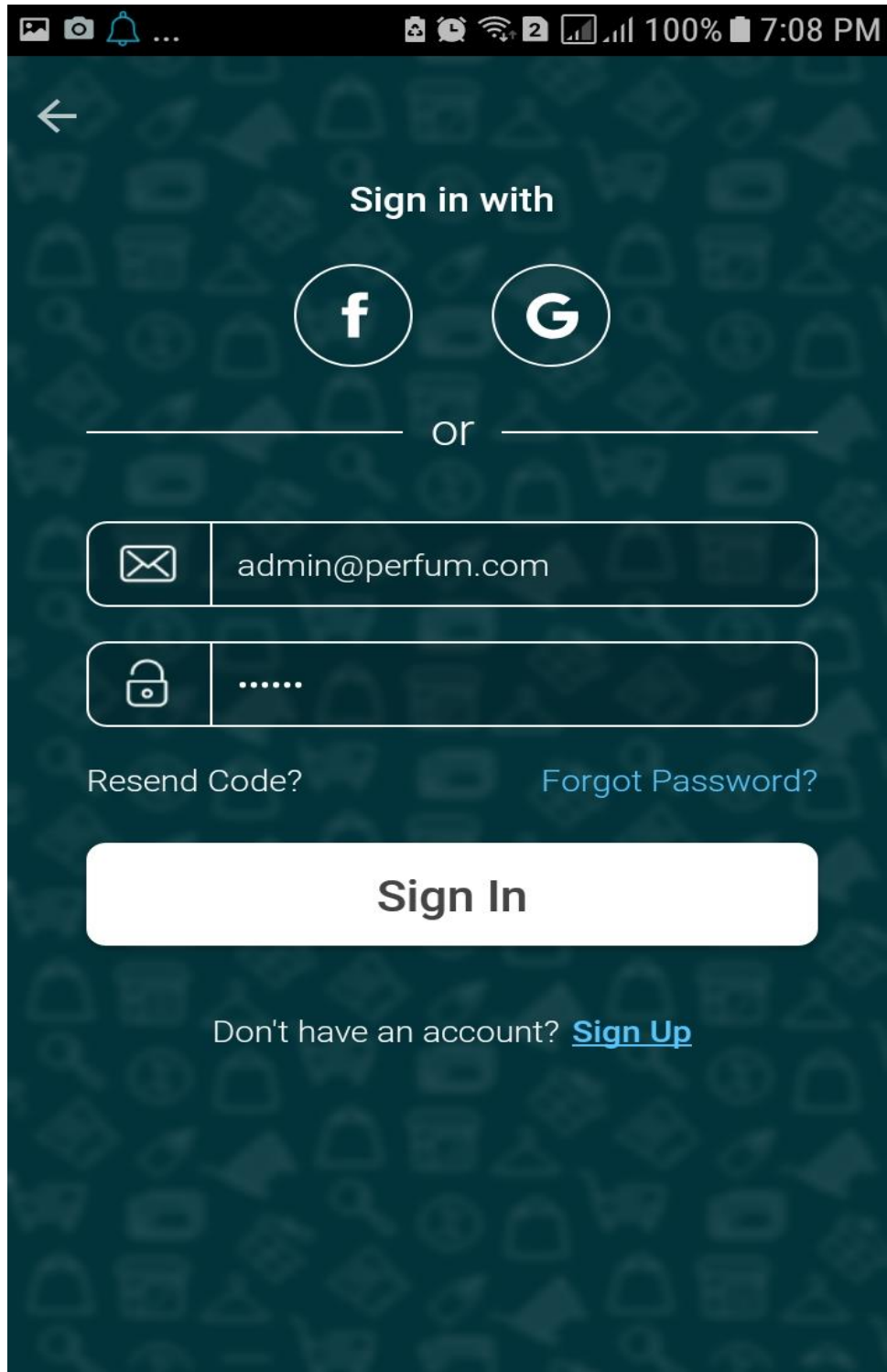


Figure 17 Login Interface

Sign Up Interface

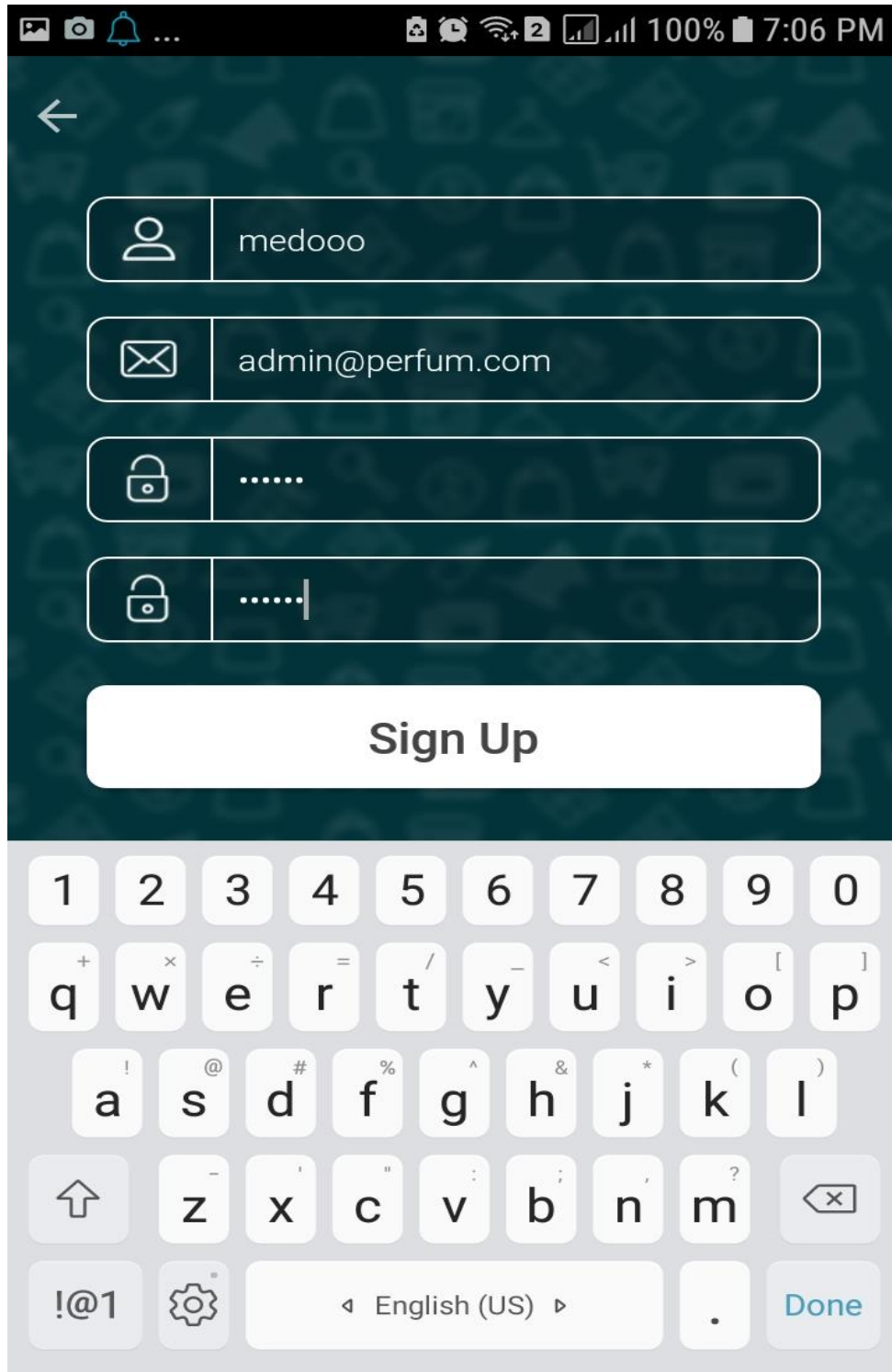


Figure 18 signup interface

Products Interface

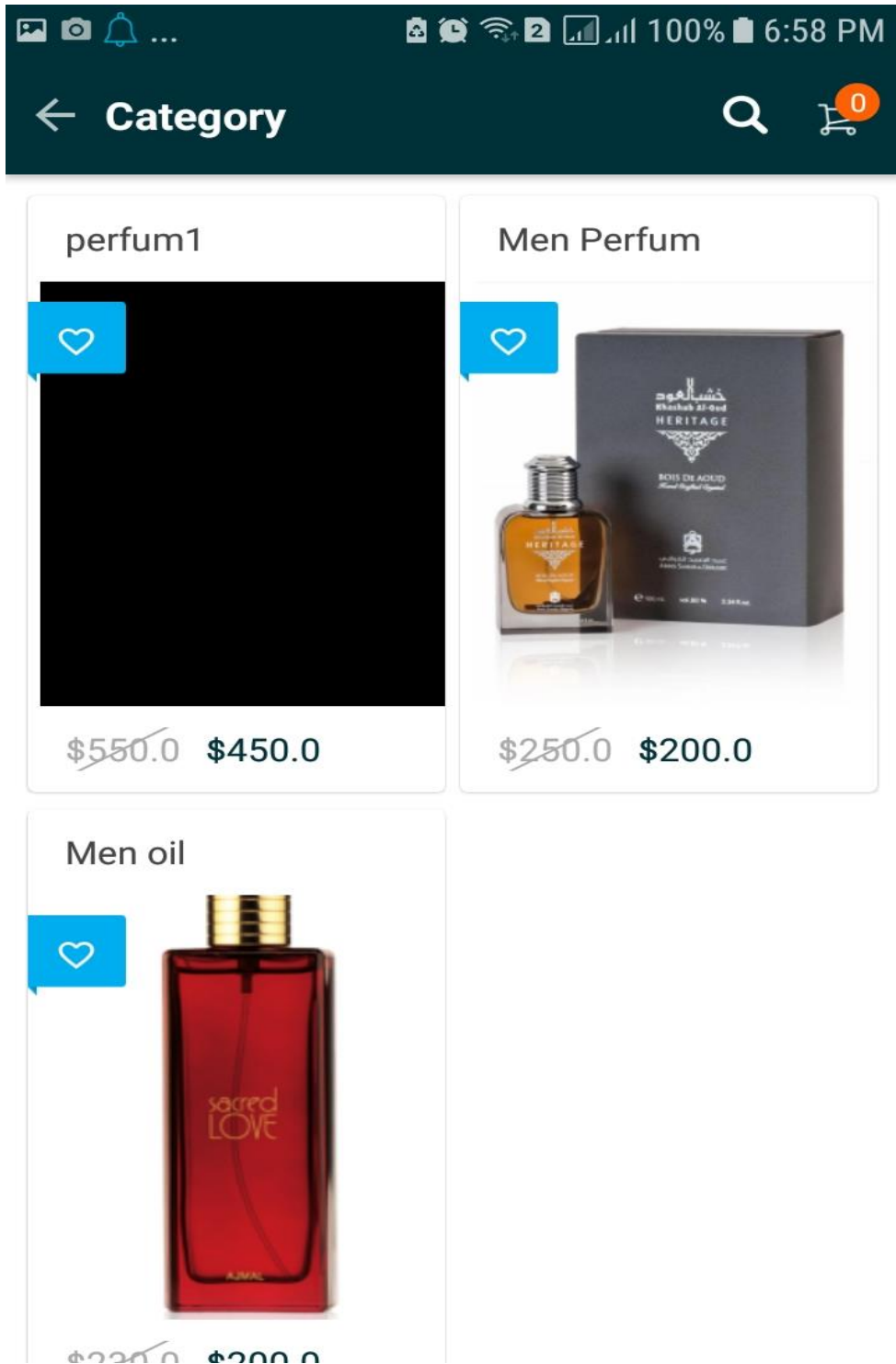


Figure 19 products interface

Bill Interface

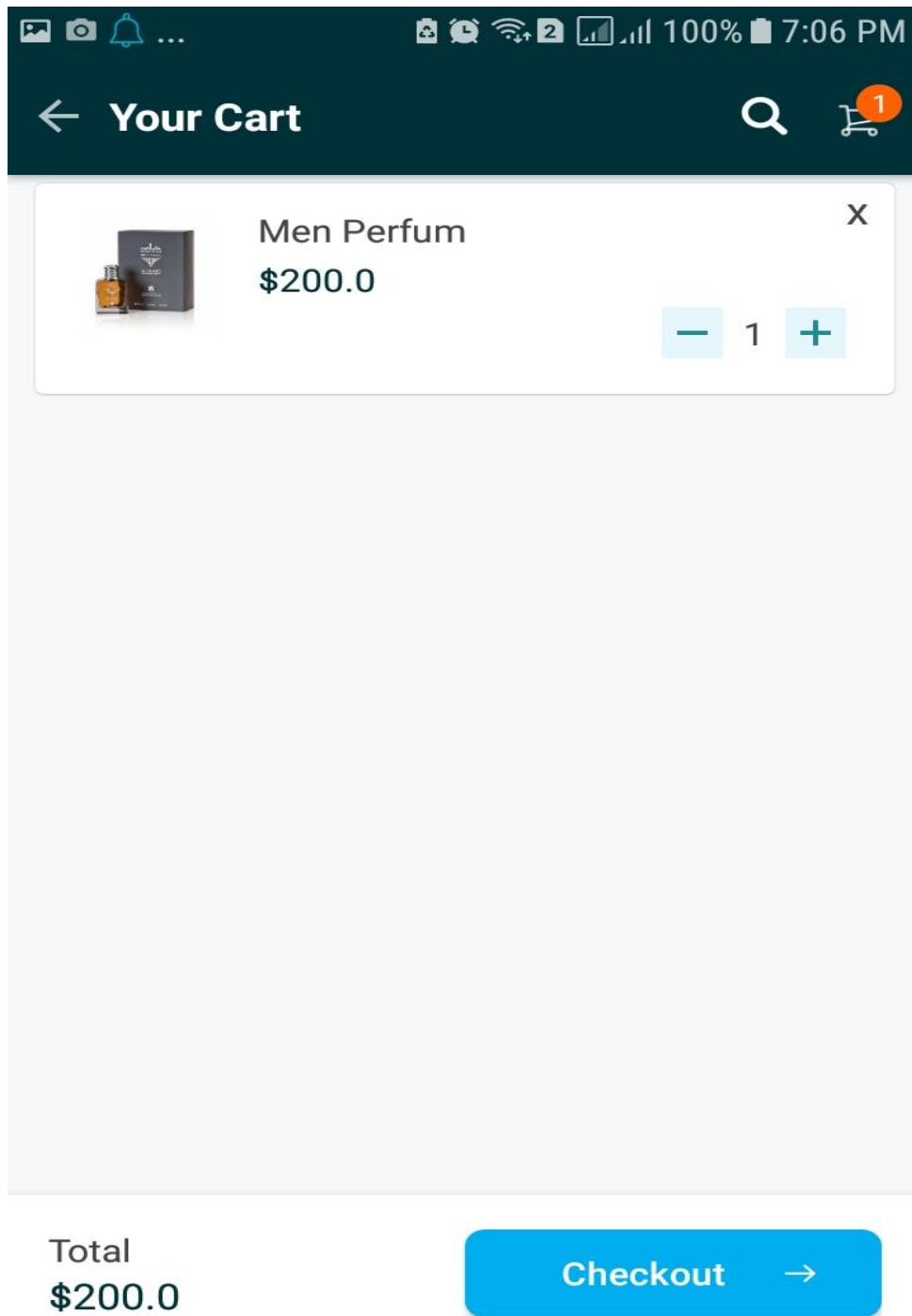


Figure 20 bill interface

3.4.1 Hardware Requirements

- Computer
 - 2GHz Processor with three cores.

- 8 GB RAM.
- 500 GB HDD space.
- Monitor with good resolution.

3.4.2 Software Requirements

- Java SDK and android studio IDE (with SDK bundle).
- Java programming language.
- MYSQL
- Android Studio

❖ Summary

In this chapter, we described the procedure of functions and write the tables with attributes, also we determined the hardware and software requirements.

Chapter 4 Introduction

4.1 Introduction

We will describe here the code implementation for the project.

4.2 Procedures

Connection to database procedures :

```
package com.perfum.data.util;

public class Constants {

    public interface SharedPrefCredential {
        public String SHARED_PREF_USER = "sharedPreferencesUserBootic";
        public String SHARED_PREF_LOGGEDIN_USER =
"sharedPreferencesLoggedInUserBootic";
        public String SHARED_PREF_CURRENCY =
"sharedPreferencesCurrencyBootic";
        public String SHARED_PREF_CURRENCY_IN =
"sharedPreferencesCurrencyInBootic";
        public String SHARED_PREF_EMPTY_CART =
"sharedPreferencesEmptyCartBootic";
        public String SHARED_PREF_EMPTY_CART_IN =
"sharedPreferencesEmptyCartInBootic";
        public String SHARED_PREF_PROFILEIMAGE =
"sharedPreferencesProfileImageBootic";
        public String SHARED_PREF_PROFILEIMAGE_IN =
"sharedPreferencesProfileImageInBootic";
        public String PRODUCT_DETAIL_INTENT = "productDetailIntentBootic";
        public String INTENT_CATEGORY_ID = "category_id";
    }

    public interface Preferences {
        String USER_REGISTRATION = "registered";

        String CURRENCY = "currency";
        String CURRENCY_FONT = "font";

        String BANNER_STATUS = "banner_status";
        String BANNER_UNIT_ID = "banner_unit_id";
        String BANNER_APP_ID = "banner_app_id";
        String INTERSTITIAL_STATUS = "interstitial_status";
        String INTERSTITIAL_UNIT_ID = "interstitial_unit_id";
        String INTERSTITIAL_APP_ID = "interstitial_app_id";
        String STATUS_ON = "1";
        String USER_ADDRESS = "address";
        String USER_PROFILE_IMAGE = "pro_image";
        String TAX = "tax";
        String ENVIRONMENT = "envo";
        String MERCHANT_ID = "m_id";
    }
}
```

```

String PUBLIC_KEY = "pub_id";
String PRIVATE_KEY = "pri_id";
}

public interface IntentKey {
String PAYMENT_RESPONSE = "pay";
String TOTAL_AMOUNT = "amount";
String PAYMENT_METHOD = "payment_method";
String FLAG_ORDER = "success";
String FLAG_REVIEW_GIVEN="review_given";
public String INTENT_SLIDER_ID = "category_id";
String ITEM_ID = "item_id";
String IS_ORDERED = "ordered";
}

public interface ServerUrl {
String API_TOKEN = "www";
String MAIN_URL =
"https://api.perfum.ofertasbooking.com/perfumadmin/public/";
String ROOT_URL = MAIN_URL + "api/";
String FULL_IMAGE_URL = MAIN_URL + "uploads/";
String THUMB_IMAGE_URL = MAIN_URL + "uploads/thumb/";

//end url
String LOGIN_URL = "user/login.php";
String SIGNUP_URL = "user/register.php";
String EMAIL_VERIFICATION = "user/email-verification.php";
String RESEND_CODE = "user/send-code.php";
String UPDATE_PASSWORD = "user/update-password.php";
String ADMOB_URL = "admob/admob.php";
String SETTINGS_URL = "settings/setting.php";
String MAIN_PRODUCT_URL = "product/home.php";
String PRODUCT_DETAILS_URL = "product/one.php";
String CATEGORY_URL = "category/all.php";
String PRODUCT_GRID_URL = "product/by-category.php";
String USER_ADDRESS = "address/add.php";
String UPLOAD_PROFILE_IMAGE = "user/upload-profile-image.php";
String ADD_REVIEW = "review/add.php";
String SHOW_REVIEW = "review/by-item.php";

String BRAIN_TREE = "order/braintree.php";
String ORDER_URL = "order/add.php";
String GET_ALL_ADDRESS_URL = "address/by-user.php";
String REMOVE_ADDRESS_URL = "address/remove.php";
}

public interface DefaultValue {
int WELCOME_DELAY = 1000;
String REMEMBER_ME = "yes";
long DELAY_MS = 500;
long PERIOD_MS = 5000;
int DELAY_INTERVAL_VISIBILITY = 60000;
int DELAY_INTERVAL = 1000;
}

```



```

        initView();
        mLoader = new Loader(this);
    }

    /**
     * initializing view with data
     */
    private void initView() {
        editTextName = findViewById(R.id.edit_text_name);
        editTextEmail = findViewById(R.id.edit_text_email);
        editTextPassword = findViewById(R.id.edit_text_password);
        editTextConfirmPassword =
findViewById(R.id.edit_text_confirm_password);

        textViewNameError = findViewById(R.id.text_view_name_error);
        textViewEmailError = findViewById(R.id.text_view_email_error);
        textViewPasswordError = findViewById(R.id.text_view_password_error);
        textViewConfirmPasswordError =
findViewById(R.id.text_view_confirm_password_error);
        textViewSignUp = findViewById(R.id.text_view_sign_up);

        buttonSignUp = findViewById(R.id.button_sign_in);
        buttonSignUp.setOnClickListener(this);
        textViewSignUp.setOnClickListener(this);
        checkFieldValidity();
    }

    /**
     * init toolbar
     */
    private void initToolbar() {
        toolbar = findViewById(R.id.toolbar);
        toolbar.setTitle("");
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
    }

    @Override
    protected void stopUI() {
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {

            case R.id.button_sign_in:
                getUserInput();
                break;
            case R.id.text_view_sign_up:
                Intent intent = new Intent(RegisterActivity.this,
LoginActivity.class);

```



```

        startActivity(intent);
        finish();
        break;
    }
}

@Override
protected RegisterPresenter initPresenter() {
    return new RegisterPresenter();
}

/**
 * this method check validation of input fields
 */
private void checkFieldValidity() {
    editTextName.setOnFocusChangeListener(new View.OnFocusChangeListener()
    {
        @Override
        public void onFocusChange(View v, boolean hasFocus) {
            if (!hasFocus) {
                if (editTextName.getText().toString().equals("")
                ||
                !presenter.nameValidity(editTextName.getText().toString())) {
                    textViewNameError.setVisibility(View.VISIBLE);

                textViewNameError.setText(getString(R.string.need_2_char));
                    aBoolean = false;
                } else {
                    aBoolean = true;
                    textViewNameError.setVisibility(View.INVISIBLE);
                }
            } else {
                aBoolean = !editTextName.getText().toString().equals("")
                &&
                presenter.nameValidity(editTextName.getText().toString());
                textViewNameError.setVisibility(View.INVISIBLE);
            }
        }
    });
    editTextEmail.setOnFocusChangeListener(new
    View.OnFocusChangeListener() {
        @Override
        public void onFocusChange(View v, boolean hasFocus) {
            if (!hasFocus) {
                if (editTextEmail.getText().toString().equals("")
                ||
                !presenter.emailValidity(editTextEmail.getText().toString())) {
                    textViewEmailError.setVisibility(View.VISIBLE);

                textViewEmailError.setText(getString(R.string.val_id_email));
                    aBoolean1 = false;
                } else {
                    aBoolean1 = true;
                    textViewEmailError.setVisibility(View.INVISIBLE);
                }
            }
        }
    });
}

```

```

        }
    } else {
        aBoolean1 = !editTextEmail.getText().toString().equals("")
        &&
presenter.emailValidity(editTextEmail.getText().toString());
        textViewEmailError.setVisibility(View.INVISIBLE);
    }
}
});
editTextPassword.setOnFocusChangeListener(new
View.OnFocusChangeListener() {
    @Override
    public void onFocusChange(View v, boolean hasFocus) {
        if (!aBoolean5) {
            if (!hasFocus) {
                if (editTextPassword.getText().toString().equals("")
                    ||
!presenter.passwordValidity(editTextPassword.getText().toString())) {
                    textViewPasswordError.setVisibility(View.VISIBLE);

textViewPasswordError.setText(getString(R.string.need_6_char));
                    aBoolean2 = false;
                } else {
                    aBoolean2 = true;

textViewPasswordError.setVisibility(View.INVISIBLE);
                }
            } else {
                aBoolean2 =
!editTextPassword.getText().toString().equals("")
                &&
presenter.passwordValidity(editTextPassword.getText().toString());
                textViewPasswordError.setVisibility(View.INVISIBLE);
            }
        }
    }
});

editTextName.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int
count, int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before,
int count) {

    }

    @Override
    public void afterTextChanged(Editable s) {
        if((editTextPassword.getText().toString().equals("")
            ||

```

```

!presenter.passwordValidity(editTextPassword.getText().toString())){
    aBoolean = false;
}else {
    aBoolean = true;
}
}
});
editTextEmail.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int
count, int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before,
int count) {

    }

    @Override
    public void afterTextChanged(Editable s) {
        if((editTextPassword.getText().toString().equals("")
||
!presenter.passwordValidity(editTextPassword.getText().toString())){
            aBoolean1 = false;
        }else {
            aBoolean1 = true;
        }
    }
});
editTextPassword.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int
count, int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before,
int count) {

    }

    @Override
    public void afterTextChanged(Editable s) {
        if (s.toString().contains(" ")) {
            aBoolean3 = false;
            aBoolean5 = true;
            textViewPasswordError.setVisibility(View.VISIBLE);

textViewPasswordError.setText(getString(R.string.no_space));
        } else {
            aBoolean3 = true;
            aBoolean5 = false;

```

```

        textViewPasswordError.setVisibility(View.INVISIBLE);
    }
    if
    (presenter.isConfirmPasswordMatch(editTextPassword.getText().toString(),
        editTextConfirmPassword.getText().toString())
        ||
    editTextConfirmPassword.getText().toString().equals("")) {
    textViewConfirmPasswordError.setVisibility(View.INVISIBLE);
        aBoolean4 = true;
    } else {
        aBoolean4 = false;
        textViewConfirmPasswordError.setVisibility(View.VISIBLE);
    }
    textViewConfirmPasswordError.setText(getString(R.string.password_mismatch));
    }
    });

    editTextConfirmPassword.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int
count, int after) {
        }

        @Override
        public void onTextChanged(CharSequence s, int start, int before,
int count) {
        }

        @Override
        public void afterTextChanged(Editable s) {
            if
            (!presenter.isConfirmPasswordMatch(editTextPassword.getText().toString()
                , s.toString())) {
                aBoolean4 = false;
                textViewConfirmPasswordError.setVisibility(View.VISIBLE);
            }
            textViewConfirmPasswordError.setText(getString(R.string.password_mismatch));
        } else {
            aBoolean4 = true;
        }
        textViewConfirmPasswordError.setVisibility(View.INVISIBLE);
    }
    }
    });
}

/**
 * getdata from input field
 */
private void getUserInput() {
    String name = editTextName.getText().toString();
    String email = editTextEmail.getText().toString();
    String password = editTextPassword.getText().toString();
}

```

```

        String confirmPassword = editTextConfirmPassword.getText().toString();

        if (!name.equals("") && !email.equals("") && !password.equals("") &&
!confirmPassword.equals("")) {
            if (aBoolean && aBoolean1 && aBoolean2 && aBoolean3 && aBoolean4
                && presenter.emailValidity(email) &&
presenter.nameValidity(name)
                && presenter.passwordValidity(password)) {
                    mLoader.show();
                    presenter.userRegistration(name, email, password, this);
            } else {
                Toast.makeText(this, getString(R.string.check_input_field),
Toast.LENGTH_LONG).show();
            }
        } else {
            Toast.makeText(this, getString(R.string.fill_field),
Toast.LENGTH_LONG).show();
        }
    }

    @Override
    public void onSignUpSuccess(UserRegistrationResponse user) {
        if (user != null) {
            if (user.statusCode == 200) {
                mLoader.stopLoader();
                Intent intent = new Intent(RegisterActivity.this,
EmailVerificationActivity.class);
                intent.putExtra(EMAIL_INTENT_KEY,
user.userRegistrationInfo.email);
                startActivity(intent);
            } else {
                Toast.makeText(this, user.message, Toast.LENGTH_LONG).show();
            }
        }
        mLoader.stopLoader();
    }

    @Override
    public void onSignUpError(String errorMessage) {
        Toast.makeText(this, errorMessage, Toast.LENGTH_LONG).show();
        mLoader.stopLoader();
    }
}

```

Login procedures

```

private void signIn() {
    Intent intent = Auth.GoogleSignInApi.getSignInIntent(googleApiClient);
    startActivityForResult(intent, REQ_CODE);
}

public void signout() {
    Auth.GoogleSignInApi.signOut(googleApiClient).setResultCallback(new
ResultCallback<Status>() {

```



```

}

@Override
protected CategoryPresenter initPresenter() {
    return new CategoryPresenter();
}

@Override
public void onCategoryListSuccess(AllCategoryResponse categoryResponse) {
    mCategoryResponse = categoryResponse;
    if (mCategoryResponse.dataList != null && mCategoryResponse.statusCode ==
    HttpURLConnection.HTTP_OK) {
        mAdapter.addItem(mCategoryResponse.dataList);
        isFirstTime = true;
        linearLayout.setVisibility(View.GONE);
    } else {
        hasMore = false;
        if (!isFirstTime) {
            linearLayout.setVisibility(View.VISIBLE);
        }
    }
    progressBar.setVisibility(View.GONE);
    mLoader.stopLoader();
}

```

4.3 Reports

User Table

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
SELECT * FROM `user`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

	id	username	email	password	type	social_id	verification_token	status	image_name	image
<input type="checkbox"/>	5	medooo	admin@perfum.com	\$2y\$10\$R3J5VxcVCLYd.kawV8rDN.w3rmQ5rEjvKwYjQFibB...	3	NULL	1593	1	profile_default.jpg	500.50

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Table 10

Admin Table

Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

```
SELECT * FROM `admin`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

	id	username	email	password
<input type="checkbox"/>	1	admin	admin@gmail.com	\$2y\$10\$nvuKEjEupX3W065ewlKhO1g61e0291XleNYYnDyXB...

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Table 11

Product Table

Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

SELECT * FROM `product`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	id	status	admin_id	category	title	image_name	image_resolution	description	tags	purchase_price	prev_price	current_price	featured
<input type="checkbox"/>	8	1	1	4	Men oil	7idrr1lf0ggks4.png	283.465	Lorem Ipsum is simply dummy text of the printing a...	new, slider_3, banner_1, recent	100	230	200	1
<input type="checkbox"/>	9	1	1	4	Men Perfum	71338.png	1000.1000	Lorem Ipsum is simply dummy text of the printing a...	new, recent, slider_2, banner_2	150	250	200	1
<input type="checkbox"/>	10	1	1	4	perfum1	71203.jpg	400.400	Lorem Ipsum is simply dummy text of the printing a...	new, banner_1, slider_3, recent	400	550	450	2

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Table 12

Category Table

Showing rows 0 - 1 (2 total, Query took 0.0007 seconds.)

SELECT * FROM `category`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	id	admin_id	status	title	image_name	image_resolution
<input type="checkbox"/>	3	1	1	Women	kd615i0fkg8ko40.jpg	680.680
<input type="checkbox"/>	4	1	1	Men	8e2nsqakjk8408s.jpg	500.500

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Table 13

Payment Table

Showing rows 0 - 0 (1 total, Query took 0.0017 seconds.)

SELECT * FROM `payment`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

	id	admin_id	environment	merchant_id	public_key	private_key
<input type="checkbox"/>	1	1	sandbox	k2dk75vhbmd8wthg	ktb9336dmpq6c266	47afb07fa413fb9a6d12d8e5dd209

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Table 14

Chapter 5

FUTURE ENHANCEMENTS:

This application avoids the manual work and the problems concern with it. It is an easy way to obtain the information regarding the perfum mixing .

Well I and my team members have worked hard in order to present an improved app better than the existing one's regarding the information about the various activities. Still ,we found out that the project can be done in a better way.

The next enhancement that we can add the question option. Patient can directly know his case if he open questions.These are the two enhancements that we could think of at present.

Conclusion

The idea of this project is to simplify the process of mixer perfume.

We finished the analysis and design of the system including finding functional and nonfunctional requirements, drawing UML diagrams and creating the prototype interface.

Next step is developing the real application using Android studio and MySql database .

References

- [1] K. Schwalbe, Information Technology Project Management, 2013.
- [2] "SDLC," [Online]. Available: <https://www.testingexcellence.com/software-development-life-cycle-sdlc-phases/>. [Accessed 24 10 2018].
- [3] "software development methodologies," [Online]. Available: <http://www.itinfo.am/eng/software-development-methodologies/>. [Accessed 25 10 2018].
- [4] "Waterfall," [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm. [Accessed 24 10 2018].
- [5] "waterfall," [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm.
- [6] "data flow diagram," [Online]. Available: <https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/>.
- [7] "ERD," [Online]. Available: <https://searchdatamanagement.techtarget.com/definition/entity-relationship-diagram-ERD>.
- [8] "use case diagram," [Online]. Available: <https://whatis.techtarget.com/definition/use-case-diagram>.
- [9] "class diagram," [Online]. Available: https://www.tutorialspoint.com/uml/uml_class_diagram.htm.
- [10] "activity diagram," [Online]. Available: https://www.tutorialspoint.com/uml/uml_activity_diagram.htm.
- [11] "Number of smartphone users in Saudi Arabia," [Online]. Available: <https://www.statista.com/statistics/494616/smartphone-users-in-saudi-arabia/>. [Accessed 1 11 2018].
- [12] "context diagram," [Online]. Available: <https://study.com/academy/lesson/system-context-diagram-description-examples.html>.
- [13] "Windows 10," [Online]. Available: https://en.wikipedia.org/wiki/Windows_10.
- [14] "Word," [Online]. Available: <https://office.live.com/start/Word.aspx>.
- [15] "android studio," [Online]. Available: <https://developer.android.com/studio/preview/>.
- [16] "photoshop," [Online]. Available: <https://www.photoshop.com/>.
- [17] "mysql," [Online]. Available: <https://www.mysql.com/>.
- [18] "clearvision," [Online]. Available: <http://www.clearvision.com.sa>.
- [19] "attendance management system," [Online]. Available: <https://bizom.in/sfa/attendance-management-system.php>.
- [20] "Employee Leave Management System," [Online]. Available: https://play.google.com/store/apps/details?id=com.vvs.android.vvslms&hl=en_US.

