THE SUPERIOR COLLEGE LAHORE

Faculty of Computer Science & IT

Final Year Project
PROJECT REPORT

Discount Desk

Project Team

Student Name	Student ID	Program	Contact Number	Email Address
Umair Nazir	Mitm-F18-038	MIT	03358605849	umairnazir068@gmail.com
Noroz Naveed	Mitm-F18-054	MIT	03009471444	Noroznaveed4343@gmail.com
Babar Rafique	Mitm-F18-025	MIT	03404143964	babarrafique1801@gmail.com

Mudasir Dilawar
Lecturer

Project Report

Discount Desk

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

PROJECT MANAGER

Comments: _____

_____

Date:_____ Signature:_____

HEAD OF THE DEPARTMENT

Comments: _____

_____

Date:_____ Signature:_____

Dedication

This work is dedicated to our family, friends and everyone who showered us with their love and support. We are especially thankful to those who presented challenges to our work and we pray for them because without their resistance we would never have been able to achieve what we did.

Acknowledgements

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude we gave to our final year project supervisor Mr. Mudasir Dilawar, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report.

A special thanks goes to our team because we help each other to assemble the parts and gave suggestion about the task “Discount Deck”. We have to appreciate the guidance given by other supervisor, thanks to their comments and advices.

Executive Summary

Main objective of Discount Desk is to develop an application for online shopping. Most of the Customers are still using Manual or visit system for purchase anything. It is difficult for them to get the time for shopping. This project is an attempt to provide the advantages of online shopping of a real shop. It helps buying the products of the shop anywhere through internet by using an android device. Thus, the customer will get the service of online shopping and home delivery from his favorite shop. This system can be implemented to any shop in the locality or Multinational Branded Shops having retail outlet chains. By having this type of application, the shopkeepers won't lose their customer to the trending online shops such as flip cart or eBay. If this application is available in smart phone it is easily accessible and always available.

This application has some new functionality like by reserving a product for short time, location of the outlet and stock of the product. The tool that we are using for developing this application is Android Studio & firebase.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1: Introduction

The purpose of writing this proposal is to describe the complete information about our project named **"Discount Desk"**. In this project we will develop an application which is useful for the customers who always try to wear best stuff clothes at the lowest price. One of the Most Important Feature of this application is to pin the products it means that pinned product will be reserved for the customer for the next 24 hours. In these hours he can visit shop at any time to purchase that thing none of the other customer will allowed to purchase that thing. The idea of mobile application will provide information about all sales of a specific shop. We will give the customers a platform where they can easily get the access to the entire product of that Shop.

1.1. Background

In 1979 it all began when Michael Aldrich invented "Online shopping". Using videotext, a two-way message service. Its revolutionized businesses. Now, we know this as E-commerce. In 1982 Mintel, a videotext online service accessible by telephone could be used to make online purchase. It is considered the most successful pre-WWW online service. In 1984 the first ever Shoppers buy online at Tesco store. In 1995 AMAZON started selling almost anything. Now a day's lot of the websites and application are used for online shopping. But they all are facing some of the problems. We try our best to overcome these problems.

1.2. Motivations and Challenges

This is an online shopping system which provides functionality for both customers as well as shopkeepers, that will help them to connect with each other by using internet. It will be good for customers to having an application that will keep them close to their favorite things which includes clothes Men, Woman accessories etc. Customers will be able to browse through the shop and can check the new arrival and remaining stock. This will reduce wastage of time and make things easy for everyone associated with it. This gave me the perfect motivation to build something for them and offer them some help at their most needed time. I am confident that this product will help them. Development of such system will be a great challenge for us to

understand the complete functionality. Installation in complex environment will create problems in start because changing manual large system to automatic system is not easy to do.

1.3. Goals and Objectives

The first and major goal of our system is to generate detail reports which can help our customer to achieve business intelligence. And it can achieve by efficient record management system and information sharing which will provide by our software system. Our second goal is people will get benefits of using this system by just viewing the detailed information of the shop, like stock, new arrival, and Discounted things. The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information about that shop is coming from the center is accurate.

1.4. Literature Review/Existing Solutions

There are many systems which are developed, but these systems are not self- sufficient. Some record keeping systems do not have efficiency to perform tasks rapidly. If the information or data required from multiple shops that take plenty of time, person has to move shop to shop for the getting their favorite collection, these applications does not have communication interfaces to communicate with the owner. This is a big drawback in information systems. These systems have no integrated information regarding to supply chain Management. If the customers view and analysis can get then it would be a great headache using these systems.

The below given table will show the difference about the existing system with our project:

Applications	Location	Pin Clothes	Direct Contact with Owner	Discount Column
Limelight	No	No	No	Yes
Mod Man	No	Yes	No	No
Outfitters	Yes	Yes	No	Yes
ChenOne	Yes	Yes	No	NO
Discount Desk	Yes	Yes	Yes	Yes

1.5. Gap Analysis

Online shopping is a huge platform, as our goal is to cover all the Stuff in the shop. But in the start, we will cover Men's wear where audience is large as it will help us to grow our system and we will be able to maintain and stabilize our system. Then further on we will add more fields of shop like women wear and other accessories, and then we will enhance our system and meet our desired goals and needs.

1.6. Proposed Solution

In the proposed system customer need not go to the shop for buying the products. He can order the product he wishes to buy through the application in his Smartphone. He/she add the products in wish list which he likes or wants to buy. The shop owner will be admin of the system. The system also recommends a home delivery system for the purchased products. The payment method is on delivery.

1.7. Project Plan

No: of Phases	Description of Work	Time
Phase One	Documentation	4 Months(3rdSemester)
Phase Two	Mobile Application	2 Months(4thSemester)
Phase Three	Testing Phase	2 Months(4thSemester)

Table No 1: Project Plan

1.7.1. Work Breakdown Structure

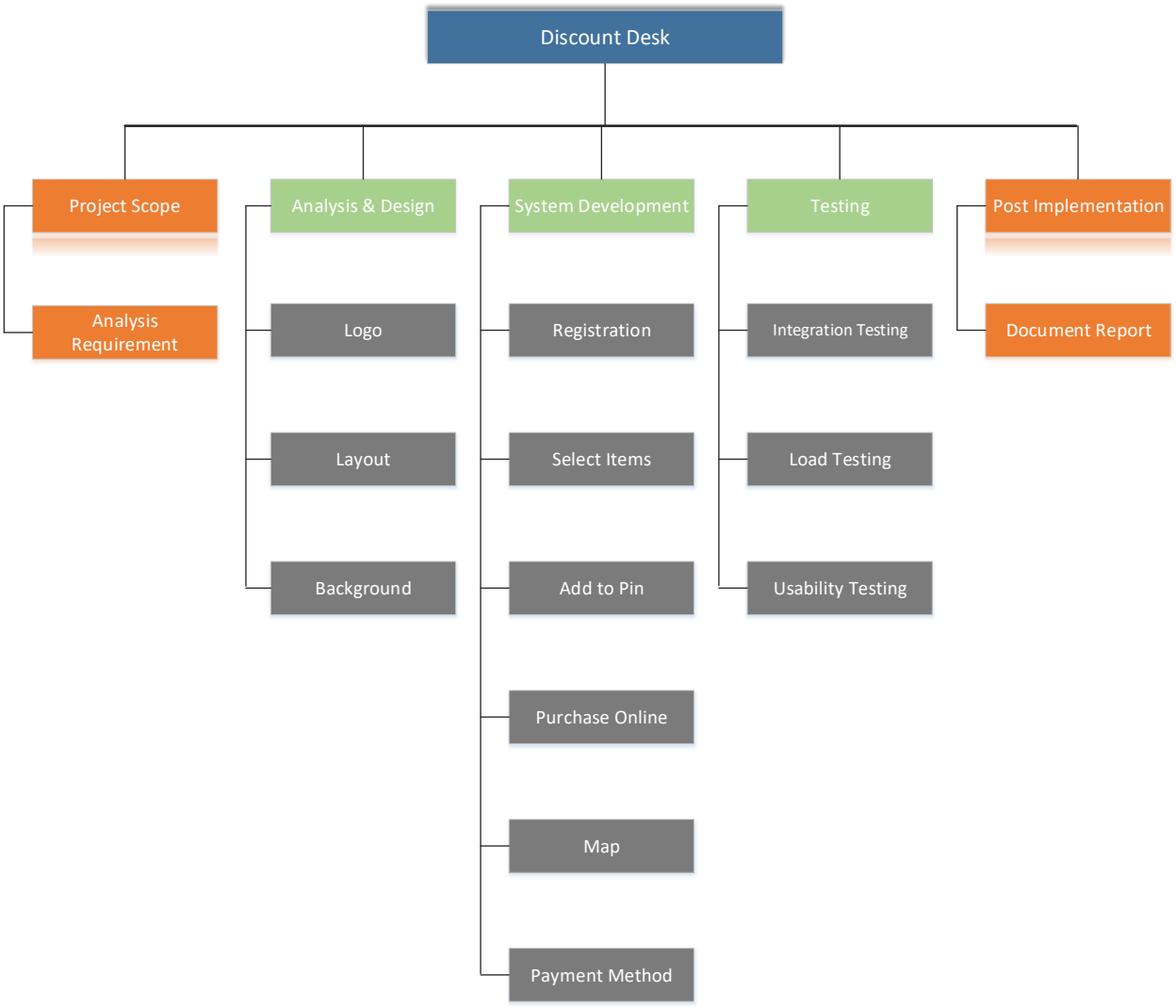


Figure 1: Work Breakdown Structure

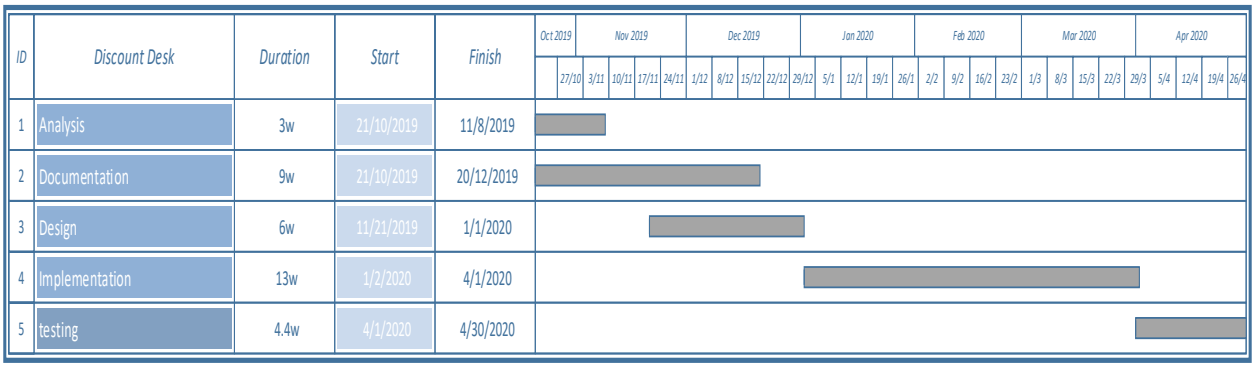
1.7.2. Roles & Responsibility Matrix

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	15-09-2019		Proposal submission	10	All team
2	22-09-2019	2.1	First Documentation	5	Babar
		2.3	Requirement Analysis	15	Noroz
		2.4	System Design	10	Umair
		2.5	Implementation	10	Umair

		2.6	Testing & Performance	5	Babar
		2.7	Conclusion & outlook	2	Babar
3		3.1	Development	7	Umair
		3.1.2	Server	8	Noroz
		3.1.4	Database	4	Noroz
		3.2	Presentation Layer	2	Babar
		3.3	Business Logic layer	8	Noroz, Babar, Umair

Table No 2: Roles & Responsibility Matrix

1.7.3. Gantt Chart



1.8. Report Outline

Purpose of this Project is to provide a better handling about the all relevant data of the shop. This system will update stock on daily basis so that the customer will not face any problem about the unavailability of the clothes. In the pinned column the Clothes will reserve for the customers for the next 24 Hours. It will very helpful for customer and Shopkeepers.

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

2.1.1. Purpose

The main purpose of this document is to describe all the specification of online shopping system "Discount Desk". This system is capable of purchasing clothes, Pinned the things and to give the location of the shop.it will also calculate the bill, give the time of delivery and can generate report of the sales.

2.1.2. Document Conventions

This document is prepared using Microsoft word 2016, font type 'Calibri'. The first heading is 16pt bold, second heading is 14pt bold and paragraph font size is 12pt with 1.5 line spacing. All the pages except cover page are numbered. Standard IEEE template is the template used to organize the appearance of the document and its flow.

2.1.3. Intended Audience and Reading Suggestions

In first section purpose of the document is described where every reader can understand the purpose of the project. In second section we described the product scope where stakeholders can get the idea of the system. In overall description stakeholders can clear whole idea of the systems. Developers can understand the requirements of the system. Designers can understand the design requirements in the user interface requirement sections. Tester can prepare test cases for testing the features of the system. Customers can verify their functional requirements.

2.1.4. Product Scope

This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24*7 and a home delivery system which can make customers happy. The customer can also pin the things which means it will reserve for the customer for next 24 hours. If shop provides an online application where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as flip cart or eBay. Since the application is available in the Smartphone it is easily accessible and always available.

2.2. Overall Description

2.2.1. Product Perspective

The product is supposed to be an open source, under the GNU general Public License. It is a application based system implementing client-server model. The “Discount Desk” provides simple mechanism for customers to purchase clothes online.

- Cross platform support: Offers operating support for most of the known and commercial operating systems
- User account: The system allows the user to create their accounts in the system and provide features of updating and viewing profiles.
- Number of users being supported by the system: Though the number is precisely not mentioned but the system is able to support a large number of online users at a time.
- Search: search is simply local search engine based on key words.

2.2.2. Product Functions

Product functions are given below:

- Search variety.
- Order online.
- Pin the things.
- Check Location of shop.

2.2.3. User Classes and Characteristics

Admin

Admin is the user who interact whole system and manage all activities, so he is also the most privileged user of the system.

Following are activities of admin:

- Check out reports.
- Take the order of customers.
- Deliver Orders.
- Reserve the pinned things.

- Update stock.

Customer

- They can purchase things
- Place order
- Pin the thing.
- Get the location of store.
- Edit profile.

2.2.4. Operating Environment

Operating environment is as:

- Centralized database
- Client/server system
- Operating system: Windows.
- Database: Firebase
- Platform: Android studio

2.2.5. Design and Implementation Constraints

The project intends to follow SCRUM (AGILE METHODOLOGY) for its development. Agility helps in time boxed iterative development, suggests adaptive planning and promote incremental delivery. To deal with changing requirements and yield frequent and small software increments which can be adjusted, documented and built on the project intends to use SCRUM as the process model. Mange the data maybe it will create memory storage limitation trouble for us. User interface should be very simple which can be understandable for non-technical person. We need HTTPS for secure communication. We also have to deal with Firebase as a database.

2.2.6. User Documentation

We will provide the user manuals for the user. After reading manual user will able to use application easily. We will organize seminars and provide tutorials for system users learning.

2.2.7. Assumptions and Dependencies

If new technology or techniques arrive in future and our user would demand for the new technology then we will change the system requirements.

- Each user must have a valid user id and password
- Server must be running for the system to function
- Users must login to the system to access any record.
- Only the Administrator can delete records.

2.3. External Interface Requirements

2.3.1. User Interfaces

Application will be accessed through a Browser Interface. The interface would be viewed best using 1024 x 768 and 800 x 600 pixels' resolution setting. The application would be fully compatible with Google Play store. No user would be able to access the application without logging on to the system.

2.3.2. Hardware Interfaces

- Mobile user
- Customer
- Admin
- Internet
- Security fire wall
- Server
-

2.3.3. Software Interfaces

Following are the software that will be used for our system.

- Language: java
- Tools: Android Studio.
- Database: Firebase

2.3.4. Communications Interfaces

The communication can make through Call or email. The admin shall send confirmation email to the customers that the item they ordered will be delivered to the shipping address along with user identification.

2.4. System Features

Registration

Identifier	SF-2.4.1
Title	System Registration
Requirement	System must Register the User
Rational	To make login with system
Restriction and risk	Valid inputs
Dependencies	NULL
Priority	High

Table 3: Registration

Login

Identifier	SF-2.4.2
Title	System Login
Requirement	System must be login the User
Rational	To interact with system
Restriction and risk	Valid user name and password
Dependencies	**SF-2.4.1**
Priority	High

Search

Identifier	SF-2.4.3
Title	Search
Requirement	System must be able to Search
Rational	To Search New Clothes

Restriction and risk	Valid inputs
Dependencies	**SF-2.4.2**

Place Order

Title	Place order
Requirement	The order must be placed.
Rational	Order online
Restriction and risk	Valid inputs
Dependencies	**SF-2.4.3**
Priority	High

Payment

Identifier	**SF-2.4.5**
Title	payment
Requirement	System must be able to receive payment
Rational	Payment Method
Restriction and risk	Valid inputs
Dependencies	**SF-2.4.4**
Priority	High

2.5. Other Nonfunctional Requirements

2.5.1. Performance Requirements

Performance requirements are:

- Usability
- Reliability
- Performance
- Implementation Requirements

- Interface requirements
- Operations Requirements

2.5.2. Safety Requirements

Security Provider systems SHALL resist unauthorized, accidental or unintended usage and provide access only to users.

- Volume and performance
- Capacity
- Scalability
- Availability
- Recoverability
- Audit & provenance
- Maintainability

2.5.3. Security Requirements

Security Provider systems SHALL resist unauthorized, accidental or unintended usage and provide access only to users.

- Volume and performance
- Capacity
- Scalability
- Availability
- Recoverability
- Audit & provenance
- Maintainability

2.5.4. Software Quality Attributes

- Performance
- Scalability
- Capacity
- Availability

- Reliability
- Recoverability
- Maintainability

2.5.5. Business Rules

- Visibility
- Reliability
- Recoverability
- Maintainability
- Serviceability
- Security
- Manageability
- Data Integrity

Chapter 3

Use Case Analysis

Chapter 3: System Analysis

This chapter describe the system analysis and models of our App. This chapter includes all the main and sub functions. These models tell us how the user interacts with the App and what is the sequence of activities that occurs as the App used. It includes use case models which helps in the development of this App.

3.1. Use Case Model

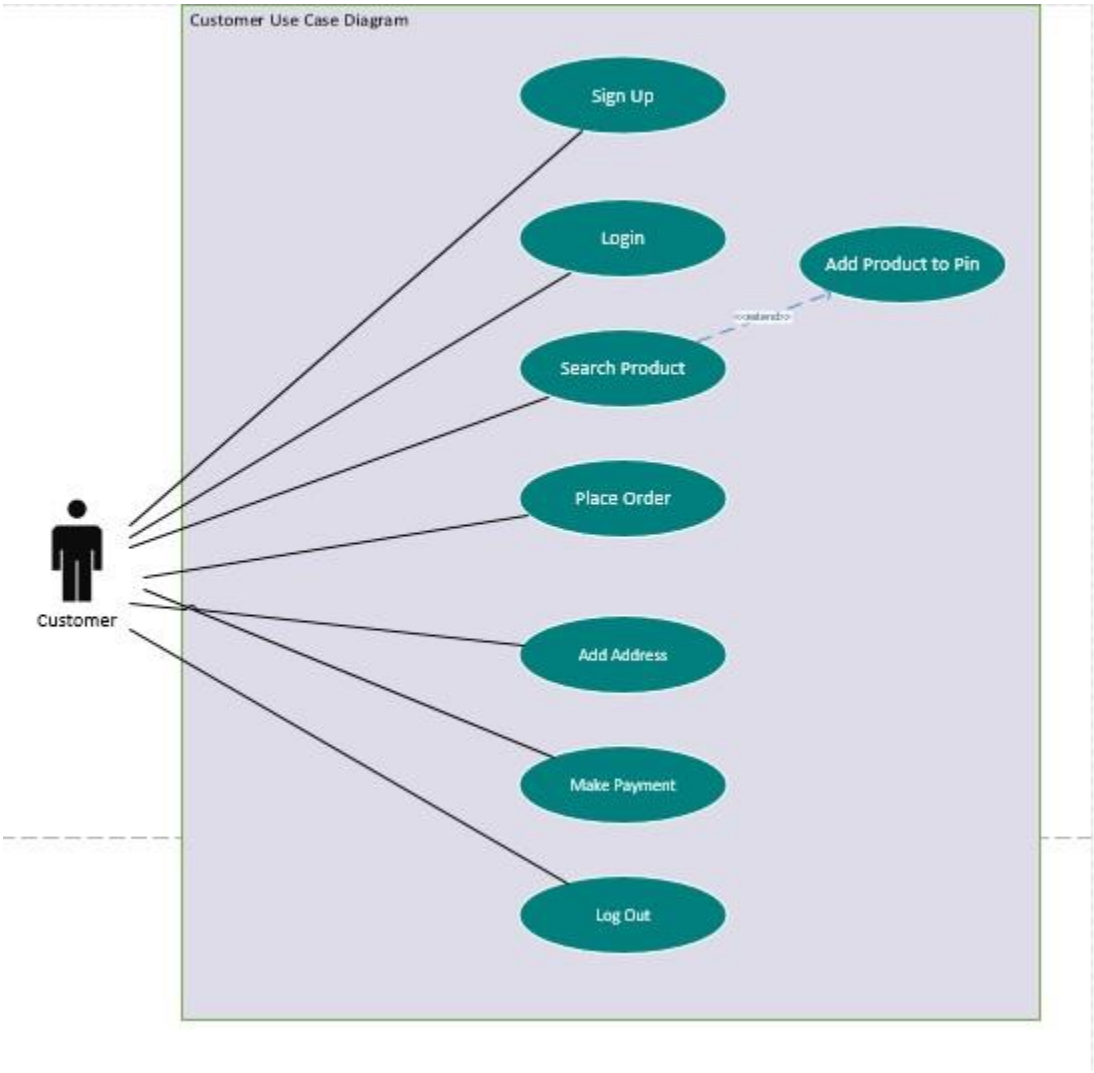


Figure 2: Customer use case

3.1.1. Use Case Admin

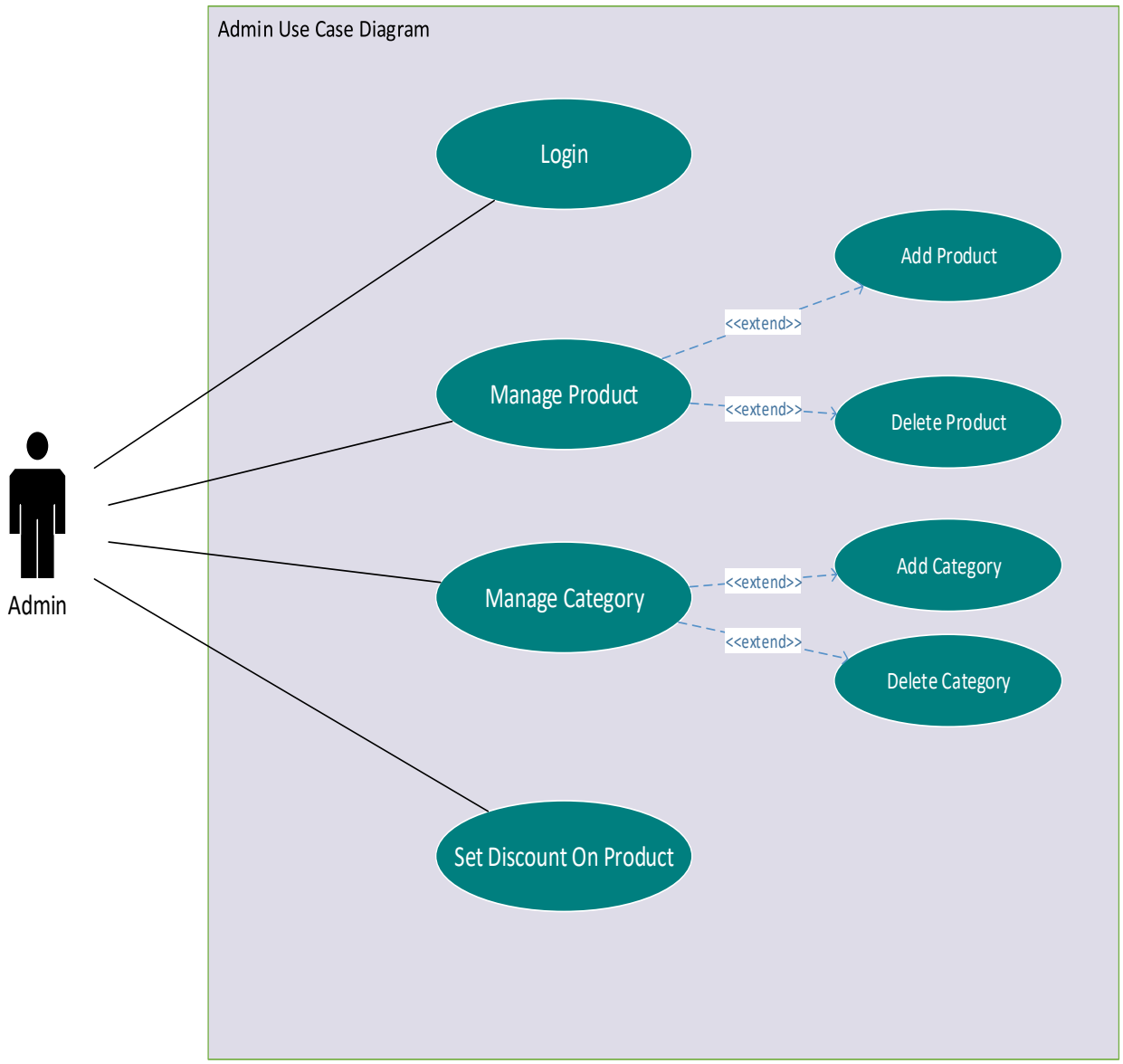


Figure 3: Admin Use Case

3.1.2. Use Case Discount Desk

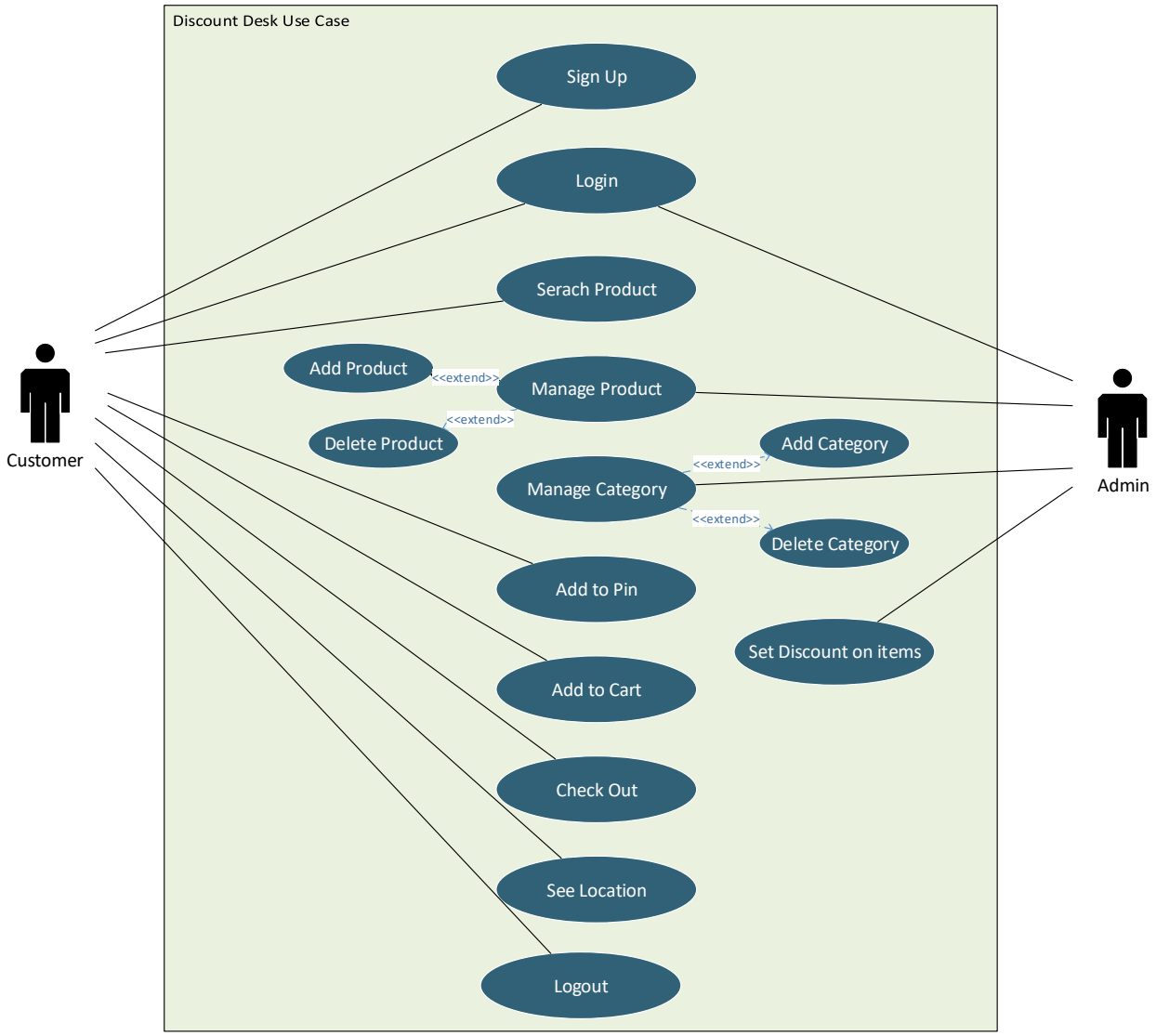


Figure 4: Use Case

3.2. Use Case Descriptions

Use Case Name	Login
Scope	Compulsory for the user to login first then start working on the Application.
Primary Actors	Customer
Secondary Actors	Admin
Pre-conditions	The internet connection must be working. User already login for the application.
Post-Condition	Customer successfully login.
Main Scenario	When the user enters correct E-mail and password then get logged in successfully.
Extension	If the required E-mail and password is not correct.
Special Requirement	The user must provide valid E-mail and correct password

Use Case Name	Add item to Pin
Scope	Shopping pin facility is available in our project.
Primary Actors	customers
Secondary Actors	
Pre-conditions	Select a product to pin
Post-Condition	Product is added in pin successfully.
Main Scenario	If the product is not pin the reason behind this must be the poor internet connection.
Special Requirement	User must provide proper quality of product.

Use Case Name	Buy Product
Scope	Customer can buy the product inserted in to cart.
Primary Actors	Customer
Secondary Actors	
Pre-conditions	Login into the system. Added a product to pin or cart. Product ready for the shipping process.
Post-Condition	Product is delivered to the customer.
Main Scenario	When the customer gets their parcel.
Extension	If the product is not delivered then it must be due to slow delivery service.
Special Requirement	User must provide correct address or phone number tom get the product on time.

Use Case Name	Manage category
Scope	Admin can manage the category of the product /items added in to the system.
Primary Actors	
Secondary Actors	Admin
Pre-conditions	User must login in to the system. User select the category from menu bar.
Post-Condition	User successfully add category of a product.
Main Scenario	When an admin can update the category section of the product.
Extension	The manage category option is only available to the admin. Admin can add or delete the product category.

Special Requirement	Many products can be added inside one category.

Use Case Name	**Check out**
Scope	This is the selection for product to go for shipping process.
Primary Actors	Customer
Secondary Actors	Admin
Pre-conditions	A product is ready for shipment process. Check out option is selected.
Post-Condition	Admin approve the order to be delivered to the customer.
Main Scenario	When an order is ready to deliver.
Extension	
Special Requirement	User should provide proper quality they want to buy.

Use case name	**Logout**
Scope	For privacy concern a user must be logout from the system.
Primary Actors	Customer
Secondary Actors	Admin
Pre-conditions	User must be logged in first.
Post-Condition	User is successfully logged out.
Main Scenario	When the user is logged in. after processing for the product and order the user is logged out from the system.

Extension	The logout button is present inside the application.
Special Requirement	After login the user must click the logout button.

Table No 4: Fully Use Case

Chapter 4

System Design

Chapter 4: System Design

This chapter describes the system design of our App. It comprises of different system diagrams such as system architecture diagram, domain model, ER diagram, class diagram, sequence diagram, component diagram and deployment model. These diagram shows and give an understanding of the architecture flow of different components, sequence and working of different functions of our Application.

4.1. Architecture Diagram

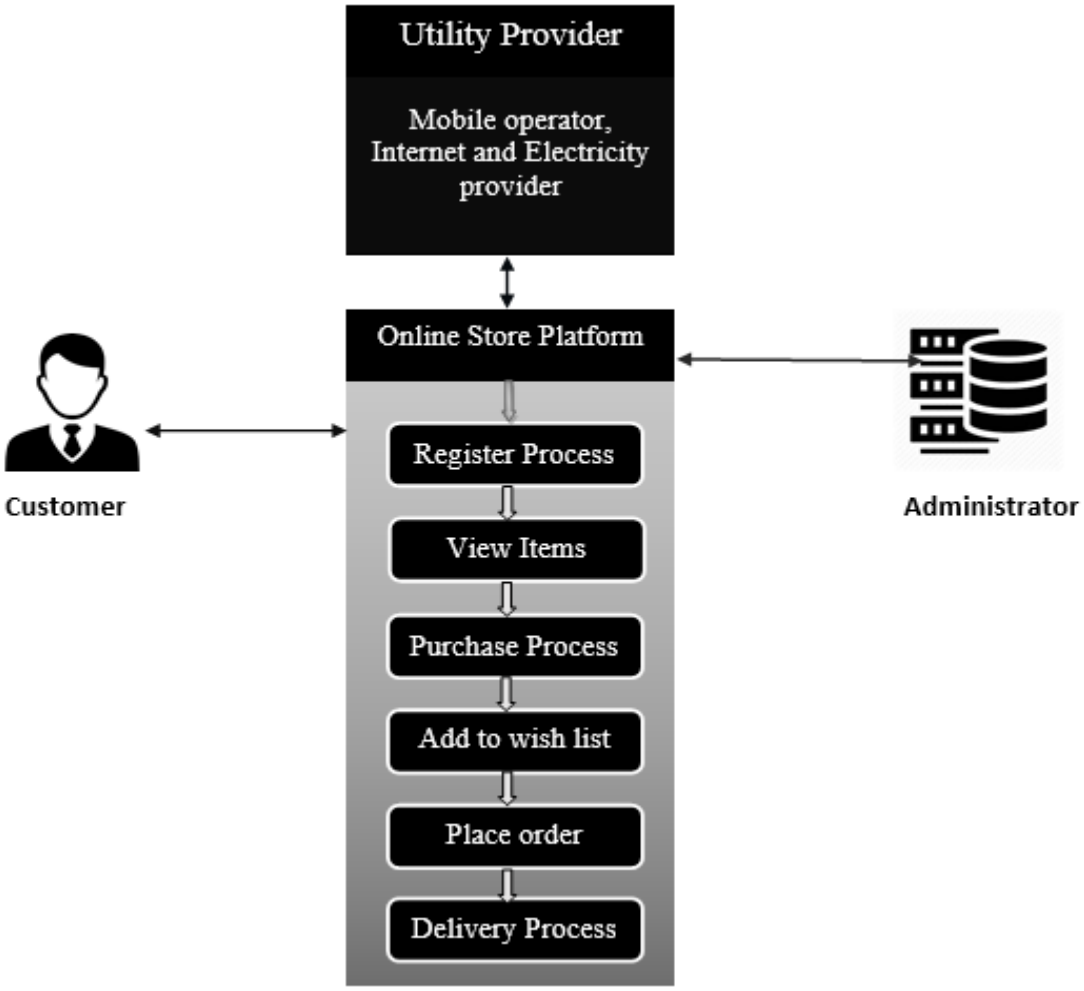


Figure 5: Architecture Diagram

4.2. Domain Model

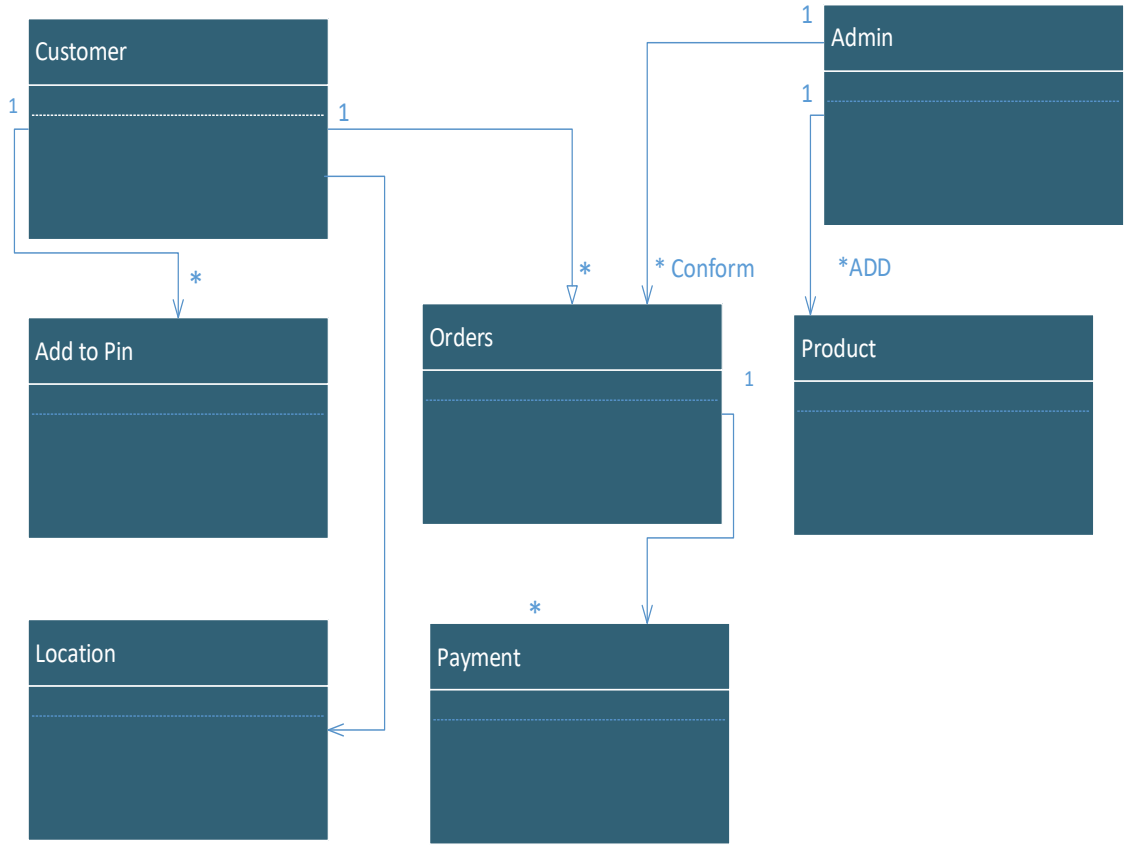


Figure 6: Domain Model

4.3. Entity Relationship Diagram with data dictionary

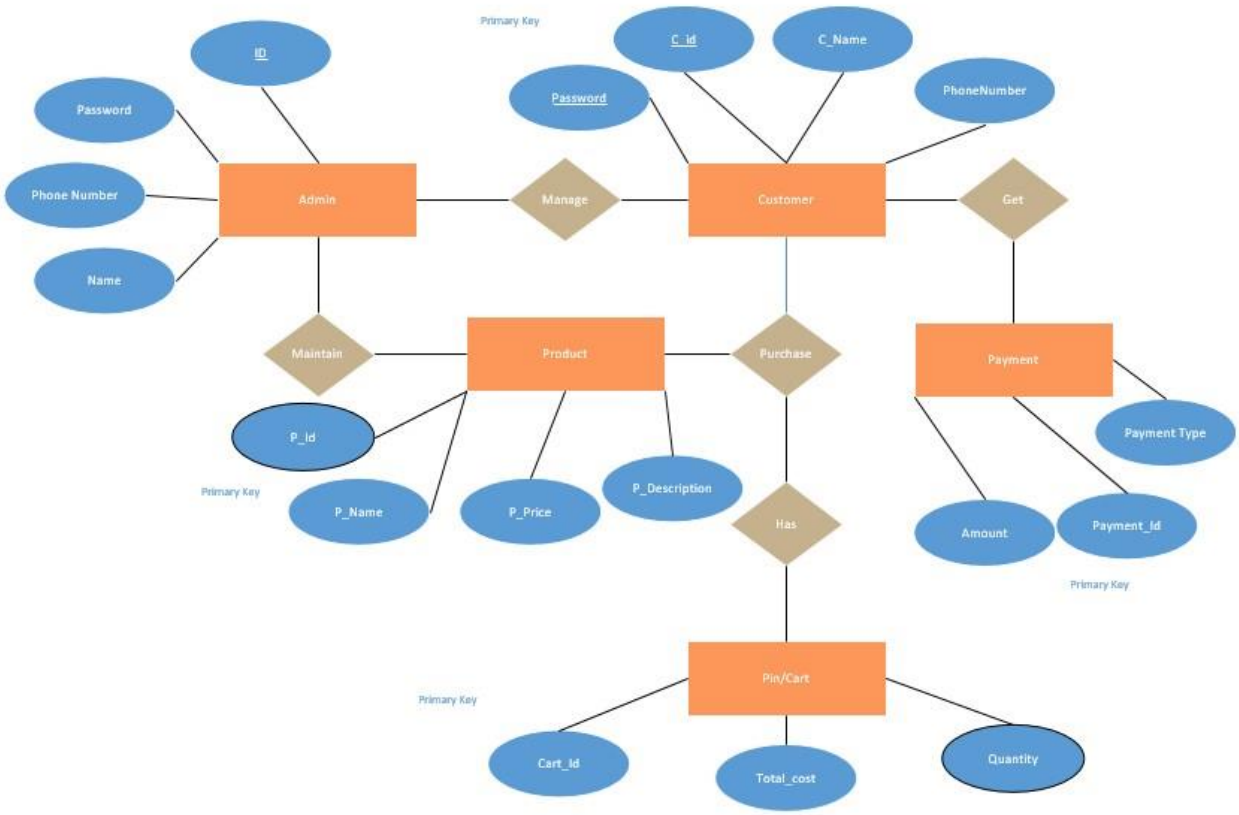


Figure 7: ERD

4.4. Class Diagram

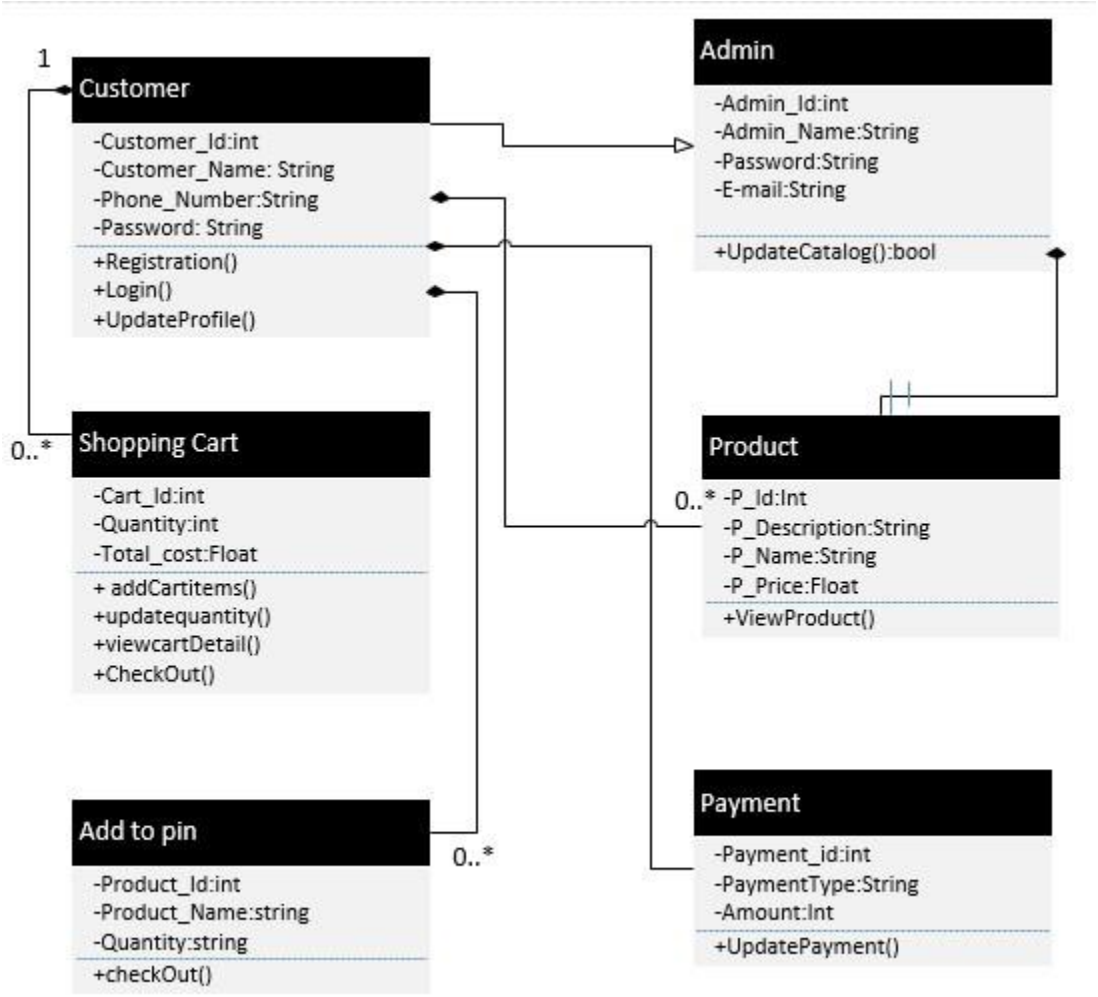


Figure 8: Class Diagram

4.5. Sequence / Collaboration Diagram

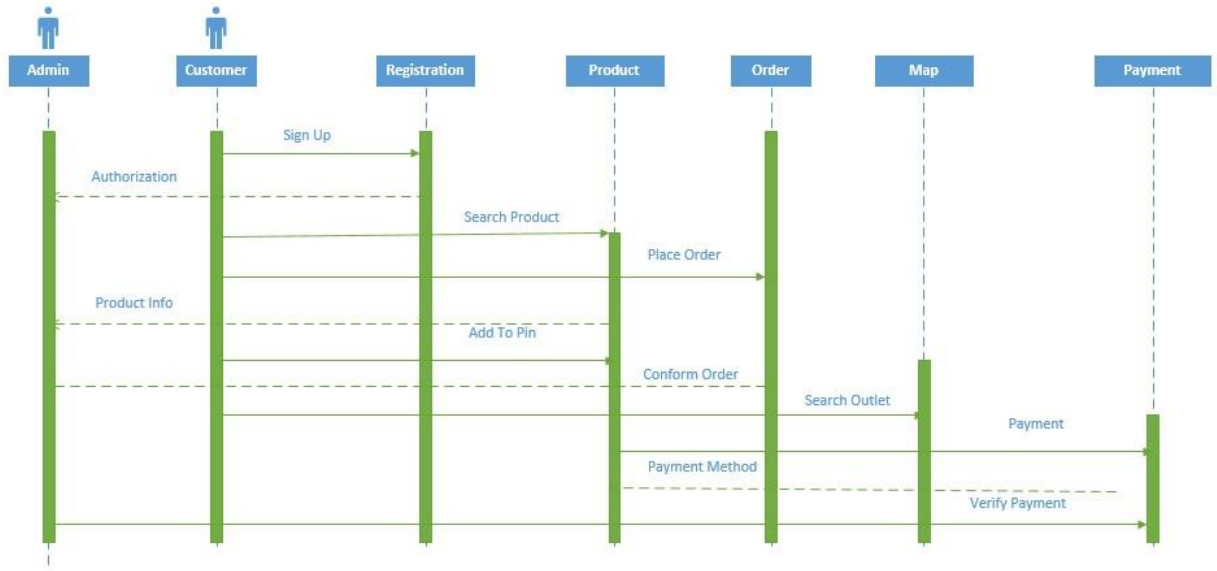


Figure 9: Sequence Diagram

4.6. Operation contracts

Operation Contract for View:

Operation: View items (Cloth)
Cross Reference:
Use case: view item
Scenario: view item
Pre-Condition:

User registration

Post Condition

Order conform or done

Operation Contract for Order:

Operation: Make order (Jeans, Shirt)
Cross Reference:
Use case: Make order
Scenario: Make order
Pre-Condition:

View items

Post Condition:

Payment

Operation Contract for Payment:

Operation: Make Payment (Cash on delivery)
Cross Reference:
Use case: Make payment
Scenario: Make payment
Pre-Condition:

Registration, Make order

Post Condition:

Receive item

Operation Contract for Admin Update:

Operation: Make update (Cloth, price, size)
Cross Reference:
Use case: Make updating
Scenario: Make updating
Pre-Condition:

Registration

Post Condition:

None

4.7. Activity Diagram

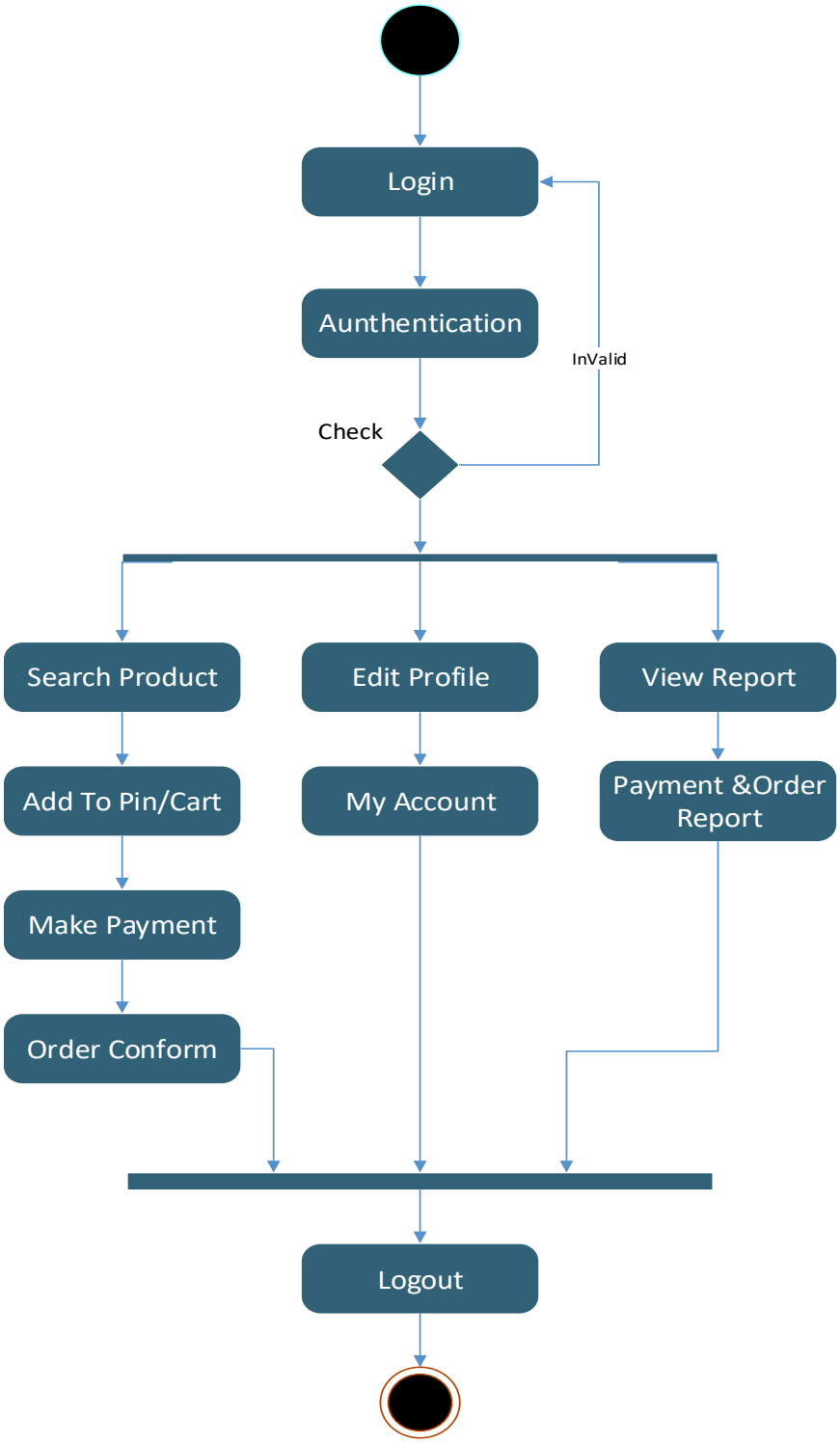


Figure 10: Activity Diagram Login

Admin Activity Diagram

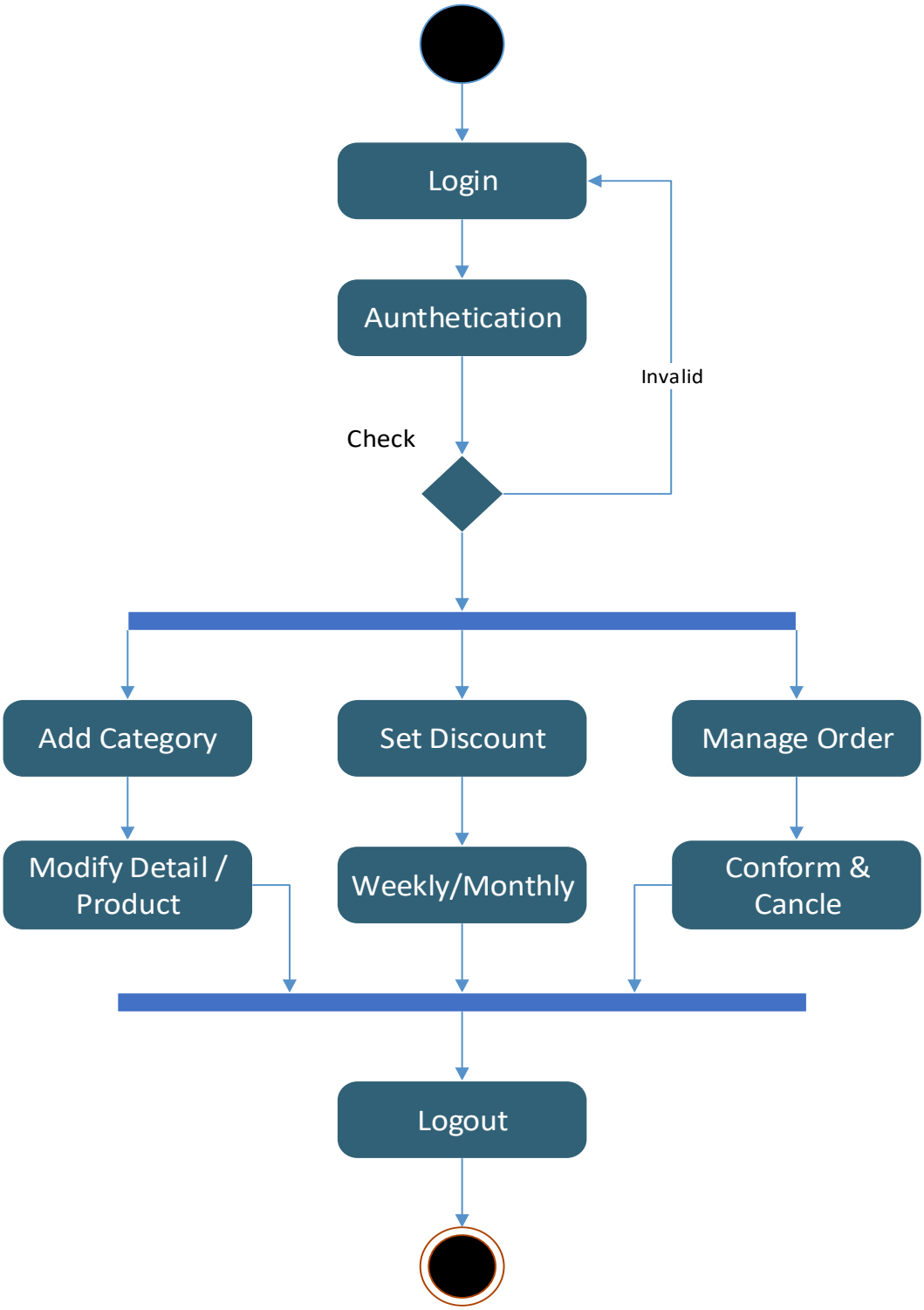


Figure 11: Admin Activity Diagram

4.8. State Transition Diagram

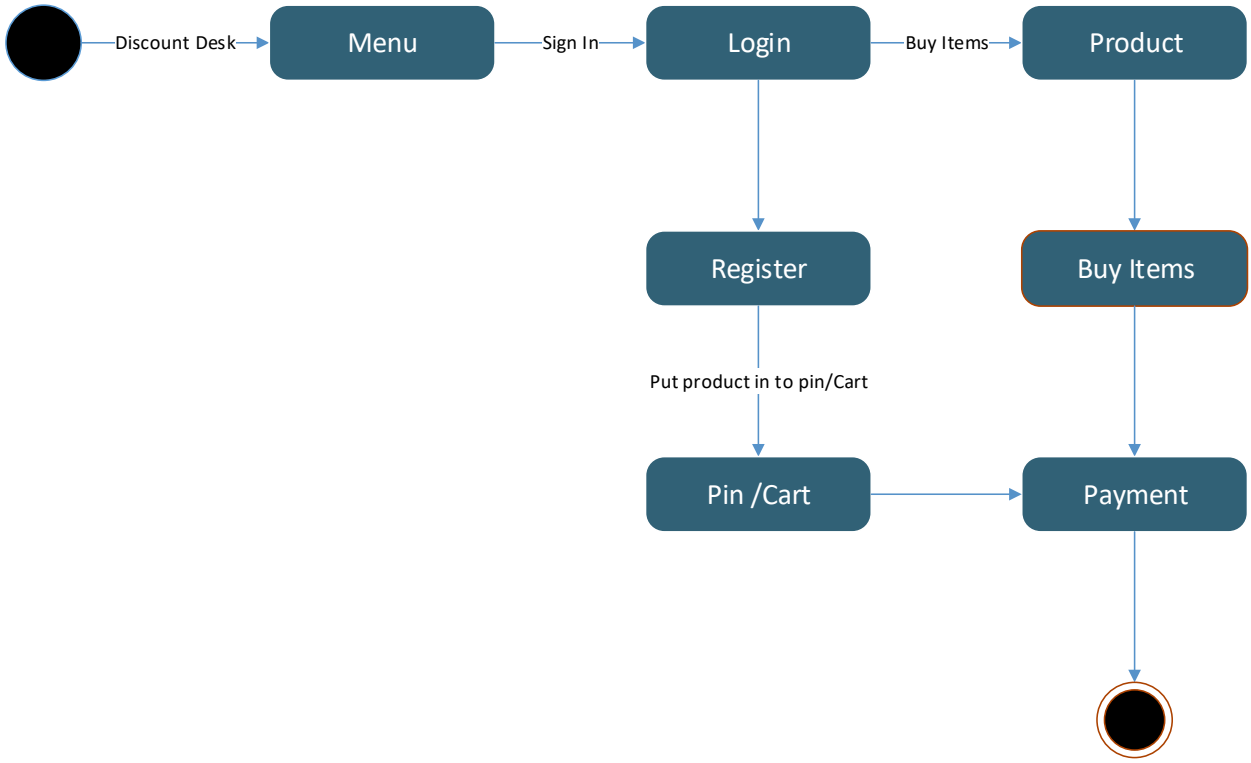


Figure 12: State Diagram

4.9. Component Diagram

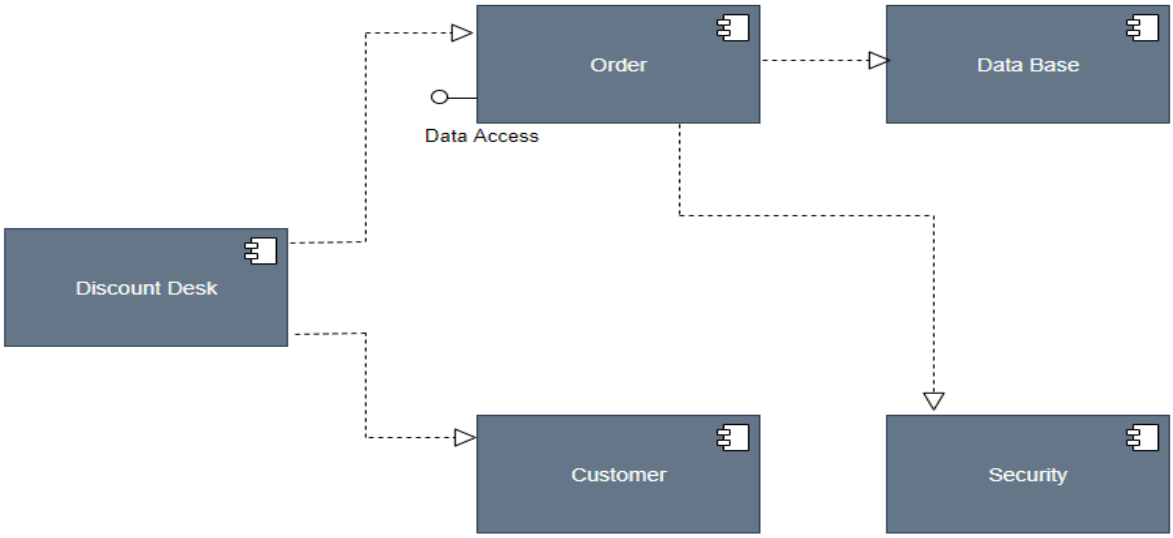


Figure 13: Component Diagram

4.10. Deployment Diagram

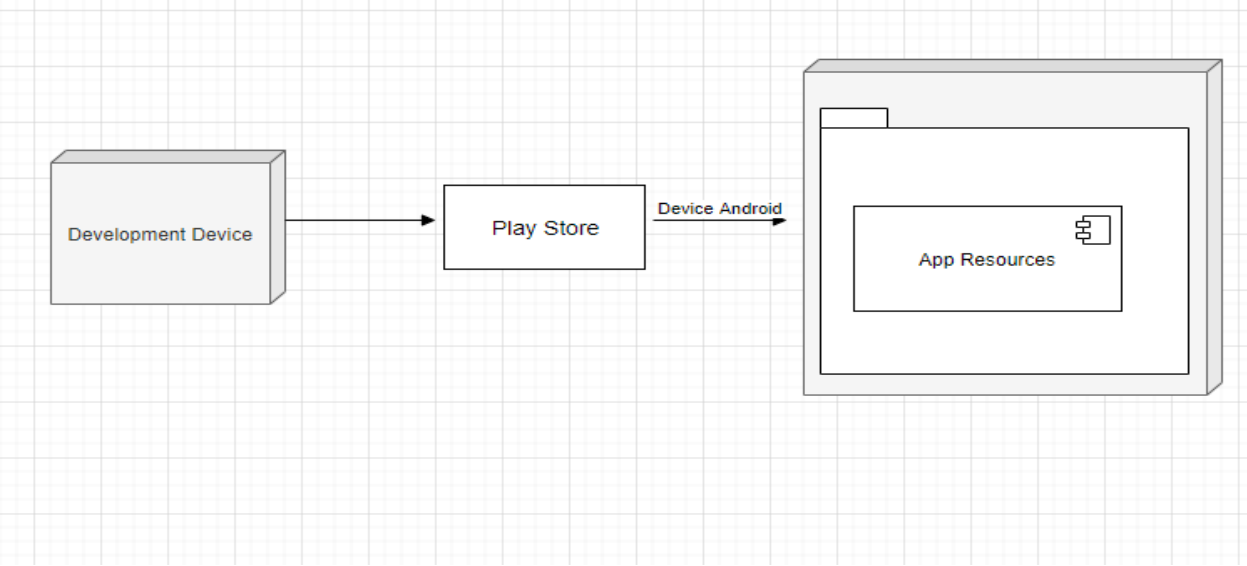


Figure 14: Deployment Diagram

Chapter 5

Implementation

Chapter 5: Implementation

This chapter covers the implementation of our project features and implementation flow control of our App. This chapter also describe the components, libraries and web services used in our project. The development environment, tools and techniques and coding standards are explained in this chapter that we are using in our project.

5.1. Important Flow Control/Pseudo codes

Main Activity

```
package com.example.discountdesk;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import Model.Users;
import Prevalent.Prevalent;
import io.paperdb.Paper;

public class MainActivity extends AppCompatActivity {
   private Button joinNowButton, loginButton;
```

```
private ProgressDialog loadingBar;

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

    joinNowButton = (Button) findViewById(R.id.main_join_now_btn);
    loginButton = (Button) findViewById(R.id.main_login_btn);
    loadingBar = new ProgressDialog(this);
    Paper.init(this);
    loginButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, LoginActivity.class);
            startActivity(intent);
        }
    });
    joinNowButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(MainActivity.this, RegisterActivity.class);
            startActivity(intent);
        }
    });
    String UserPhoneKey = Paper.book().read(Prevalent.UserPhoneKey);
    String UserPasswordKey = Paper.book().read(Prevalent.UserPasswordKey);

    if (UserPhoneKey != "" && UserPasswordKey != "") {
```

```
        if (!TextUtils.isEmpty(UserPhoneKey) && !TextUtils.isEmpty(UserPasswordKey)) {
            AllowAccess(UserPhoneKey, UserPasswordKey);

            loadingBar.setTitle("Already Logged in");
            loadingBar.setMessage("Please wait.....");
            loadingBar.setCanceledOnTouchOutside(false);
            loadingBar.show();
        }
    }
}
private void AllowAccess(final String phone, final String password) {
    final DatabaseReference RootRef;
    RootRef = FirebaseDatabase.getInstance().getReference();
    RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (dataSnapshot.child("Users").child(phone).exists()) {
                Users usersData = dataSnapshot.child("Users").child(phone).getValue(Users.class);

                assert usersData != null;
                if (usersData.getPhone().equals(phone)) {
                    if (usersData.getPassword().equals(password)) {
                        Toast.makeText(MainActivity.this, "Please wait, you are already logged in...",
Toast.LENGTH_SHORT).show();
                        loadingBar.dismiss();

                        Intent intent = new Intent(MainActivity.this, HomeActivity.class);
                        Prevalent.currentOnlineUser = usersData;
                        startActivity(intent);
```

```
                } else {
                    loadingBar.dismiss();
                    Toast.makeText(MainActivity.this, "Password is incorrect.",
Toast.LENGTH_SHORT).show();
                }
            }
        } else {
            Toast.makeText(MainActivity.this, "Account with this " + phone + " number do not
exists.", Toast.LENGTH_SHORT).show();
            loadingBar.dismiss();
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
    }
  });
 }
}
```

Home

```
package com.example.discountdesk;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.Menu;
import android.view.ViewGroup;
```

```
import android.widget.TextView;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.squareup.picasso.Picasso;
import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.core.view.GravityCompat;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import Model.Products;
import Prevalent.Prevalent;
import ViewHolder.ProductViewHolder;
import de.hdodenhof.circleimageview.CircleImageView;
import io.paperdb.Paper;
```

```
public class HomeActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {
   private AppBarConfiguration mAppBarConfiguration;
   private DatabaseReference ProductsRef;
   private RecyclerView recyclerView;
   RecyclerView.LayoutManager layoutManager;

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

      ProductsRef = FirebaseDatabase.getInstance().getReference().child("Products");
      Paper.init(this);
      Toolbar toolbar = findViewById(R.id.toolbar);
      toolbar.setTitle("Home");
      setSupportActionBar(toolbar);

      FloatingActionButton fab = findViewById(R.id.fab);
      fab.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view)
         {
            Intent intent = new Intent(HomeActivity.this, PinActivity.class);
            startActivity(intent);
}
      });
```

```
        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();
        NavigationView navigationView = findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
        View headerView = navigationView.getHeaderView(0);
        TextView userNameTextView = headerView.findViewById(R.id.user_profile_name);
        CircleImageView profileImageView = headerView.findViewById(R.id.user_profile_image);

        userNameTextView.setText(Prevalent.currentOnlineUser.getName());

Picasso.get().load(Prevalent.currentOnlineUser.getImage()).placeholder(R.drawable.profile).int
o(profileImageView);
        recyclerView = findViewById(R.id.recycler_menu);
        recyclerView.setHasFixedSize(true);
        layoutManager = new LinearLayoutManager(this);
        recyclerView.setLayoutManager(layoutManager);

        mAppBarConfiguration = new AppBarConfiguration.Builder(
                R.id.nav_home, R.id.nav_gallery, R.id.nav_slideshow)
                .setDrawerLayout(drawer)
                .build();
    }

    @Override
    protected void onStart()
```

```
{
    super.onStart();

    FirebaseRecyclerOptions<Products> options =
        new FirebaseRecyclerOptions.Builder<Products>()
            .setQuery(ProductsRef, Products.class)
            .build();

    FirebaseRecyclerAdapter<Products, ProductViewHolder> adapter =
        new FirebaseRecyclerAdapter<Products, ProductViewHolder>(options) {
          @Override
          protected void onBindViewHolder(@NonNull ProductViewHolder holder, int i,
@NonNull final Products model)
          {
            holder.txtProductName.setText(model.getPname());
            holder.txtProductDescription.setText(model.getDescription());
            holder.txtProductPrice.setText("Price = " + model.getPrice() + "$");
            Picasso.get().load(model.getImage()).into(holder.imageView);
            holder.imageView.setOnClickListener(new View.OnClickListener() {
              @Override
              public void onClick(View v)
              {
                Intent intent = new Intent(HomeActivity.this, ProductDetailsActivity.class);
                intent.putExtra("pid", model.getPid());
                startActivity(intent);
              }
            });
          }
```

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

   @Override
   public void onBackPressed()
   {
      DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
      if (drawer.isDrawerOpen(GravityCompat.START)) {
         drawer.closeDrawer(GravityCompat.START);
      } else {
         super.onBackPressed();
      }
   }
   @Override
   public boolean onCreateOptionsMenu(Menu menu) {
      getMenuInflater().inflate(R.menu.home, menu);
```

```
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    int id = item.getItemId();
    return super.onOptionsItemSelected(item);
}

@Override
public boolean onNavigationItemSelected(@NonNull MenuItem item)
{
    int id = item.getItemId();
    if (id == R.id.nav_cart)
    {
        Intent intent = new Intent(HomeActivity.this, PinActivity.class);
        startActivity(intent);
    }
    else if (id == R.id.nav_orders)
    {}
    else if (id == R.id.nav_categories)
    {       }
    else if (id == R.id.nav_location)
    {       }
    else if (id == R.id.nav_settings)
    {
        Intent intent = new Intent(HomeActivity.this, SettingsActivity.class);
         startActivity(intent);
    }
```

```
        else if (id == R.id.nav_logout)
        {
            Paper.book().destroy();
            Intent intent = new Intent(HomeActivity.this, MainActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
            startActivity(intent);
            finish();
        }
        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        drawer.closeDrawer(GravityCompat.START);
        return true;
    }
}
```

5.2. Components, Libraries, Web Services and stubs

Components

Components define and control the behavior of App Objects they are attached to. Following are the components of android studio that will be used in our App.

- Activities
- Content Providers
- Fragments
- Views
- Services
- View pagers
- SDK
- JDK
- Action Bars

5.2.2. Libraries

The android libraries that we will use in our app project are listed below.

- Java libraries (JAR).

5.3. Deployment Environment

Since our project Discount Desk is a mobile application, hence in deployment environment there will be.

- Android smart phone or android tablets
- User will download it from Play Store.

5.4. Tools and Techniques

TOOLS:

- Android Studio
- Corel draw
- Visio

5.5. Best Practices / Coding Standards

- Comments and documentation make the code easy to understand.
- Code must be readable.
- Code must be reusable which saves a lot of time.
- Avoid the hard-code that cannot be understandable.
- Conform to the coding standards and user requirements.
- Pair programming and code review is also an efficient way.
- Backup and save the data. So, there is no fear of loss of data.

5.1. Version Control

Team Foundation Server (TFS) is used for version control which is the Development software by **Microsoft** which includes revision control.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

Following strategies are used for testing.

- Unit Testing
- Integrated Testing
- System Testing
- Recovery Testing
- Security Testing
- Stress Testing
- Performance Testing

6.1. Use Case Testing

Login

Main Success Scenario	**Step**	**Description**
A = Actor	1	A: Enter phone and password
S = System	2	S: Valid Password
	3	Allow Account Access
Extension	2a	S: Invalid Password Message Display on the Screen asks for Re-enter password.

Logout

Main Success Scenario	Step	Description
A = Actor	1	A: Enter Logout
S=System	2	S: System Show Message to User logout
Extension	2a	S: User must login again

View Product

Main Success Scenario	Step	Description
A = Actor	1	A: View products
S = System	2	S: System Show all products to user/customer
Extension	2a	S: User/customer must login to view products

Add to Pin

Main Success Scenario	Step	Description
A = Actor	1	A: Add products to Pin list
S = System	2	S: System Show the pinned products in Pin layout
Extension	2a	S: User must login to add the products in Pin list

6.2. Boundary value analysis

Login

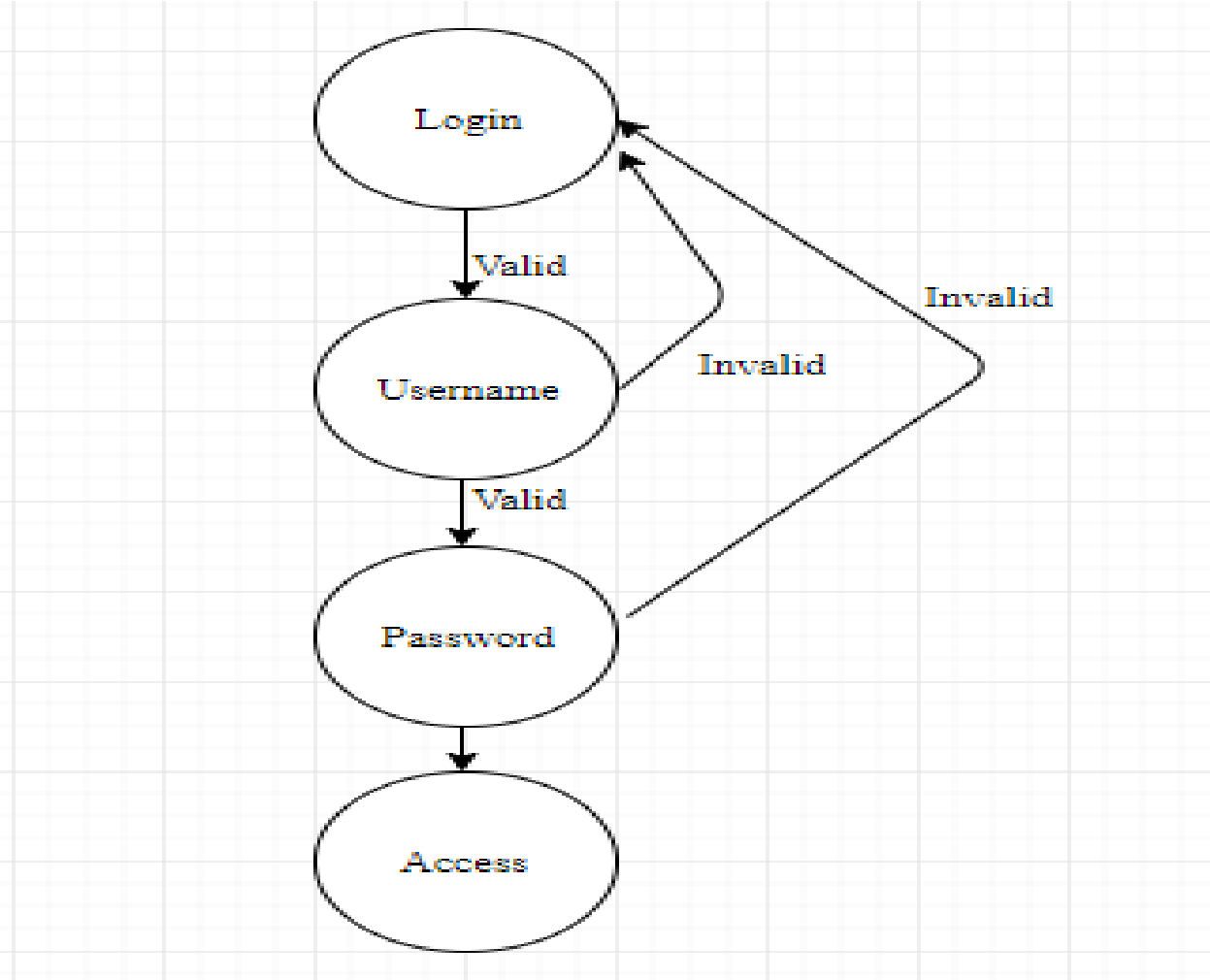
Phone

Invalid (min-1)	**Valid (min, +max, -min, max)**	**Invalid (max+1)**
3	3 to 50	30

Password

Invalid (min-1)	**Valid (min, +max, -min, max)**	**Invalid (max+1)**
8	8 to 25	25

6.3. Data flow testing



6.4. Unit testing

Unit testing is a white-box testing technique. The main consideration in this test is verification of all modules of the software system. Each module is unit tested, as follows:

- Unit testing for the module User Management is tested by valid user or invalid user.
- Unit testing for the module User Account management is provided by testing Login, Logout and creating new or deleting updating existing user.

- Unit testing for the module Detail of each Product
- Unit testing for the module Add/update/delete product, price quantity and description detail.

6.5. Integration testing

Integration testing is the technique for testing the interfaces of application components. Each application components in unit tested, and all the components are integrated to perform together. The tests are conducted to ensure that the components are working properly after interfacing.

All modules are integrated by an incremental approach, and integration testing of the system is performed as follows:

- Integrate login module and test that the application is properly connected to the database.
- Integrate all modules and test by inserting updating, and deleting records.

6.6. Performance testing

Performance testing uses criteria to check that the system functions according to the specifications. All unit modules have been tested successfully. The integration of these unit modules produces reports in the required format.

6.7. Stress Testing

A system evaluated under normal conditions in all previous testing techniques, and no conditions where system can fail are tested. Stress testing evaluates the system under abnormal conditions. The system is tested for various costs and number to check accuracy and to ensure that the system performs accurately under all input conditions.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

The summary of this project is to describe the complete information about our project named "Discount Desk". In this project we will develop an application which is useful for the customers who always try to wear best stuff clothes at the lowest price. One of the Most Important Feature of this application is to pin the products it means that pinned product will be reserved for the customer for the next 24 hours. In these hours he can visit shop at any time to purchase that thing none of the other customer will allowed to purchase that thing. The idea of mobile application will provide information about all sales of a specific shop. We will give the customers a platform where they can easily get the access to the entire product of that Shop.

7.2. Achievements and Improvements

We have presented this project discount desk to our seniors and our teachers. They Guide us at different stages. We have improved many things like first we are working on it like a university project but know we take it as our dream project. As we are students so we are trying to improve it day by day.

7.3. Critical Review

As we are students some are not that much of expert and we did not have a proper environment and we did not have that much of sources so we cannot it publicly.

7.4. Lessons Learnt

A Semester ago we did not know anything about it but now we learned a lot about making an application.

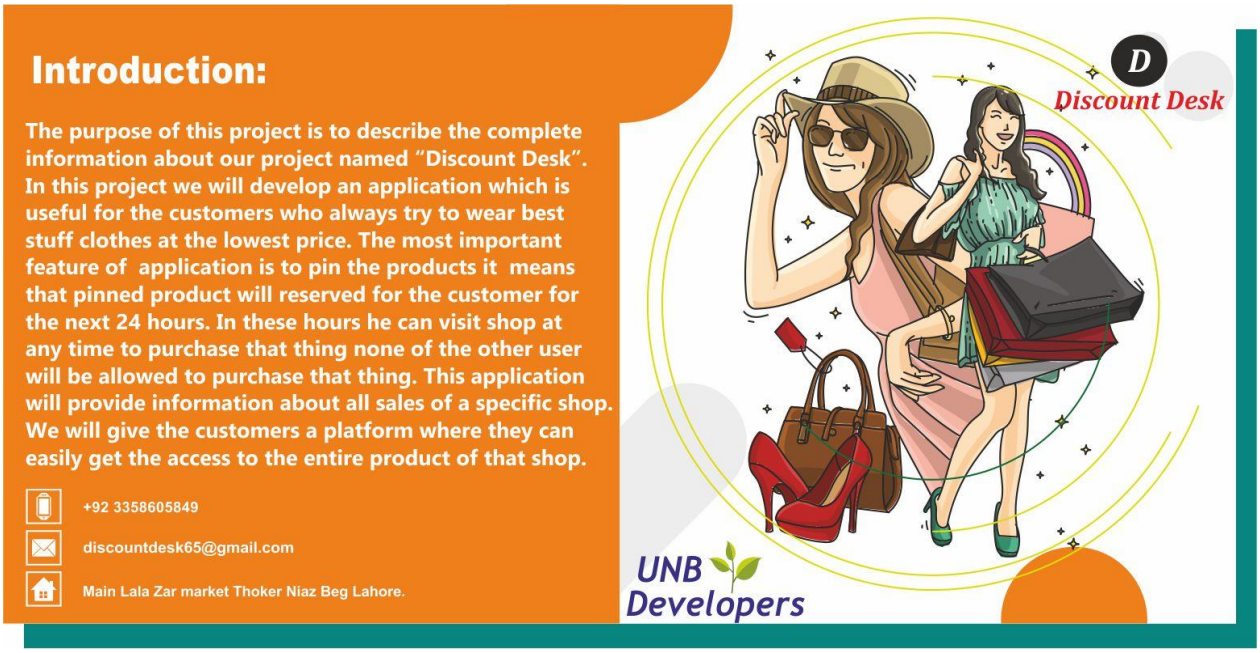
7.5. Future Enhancements/Recommendations

In future we will try issue our application to general public.

Appendices

Appendix C: Information / Promotional Material

C.1. Broacher



C.2. Flyer


UNB
Developers
Discount Desk
+92 3358605849
discountdesk65@gmail.com
Main Lala Zar market Thoker Niaz Beg Lahore.

Reference and Bibliography

Reference and Bibliography

[1] M. Sher, M. Rehman, "*Title of the Paper*" Conference name/Journal Name, Edition, Volume, Issue, ISBN/ISSN, PP, Publisher/City-Country, Year.

[2] …