



MUHAMMAD ARSALAN

01-134152-038

ZUBAIR SALEEM

01-134161-101

e-Dukan

Bachelor of Science in Computer Science

Supervisor: Dr.Arif Ur Rahman

Department of Computer Science
Bahria University, Islamabad

February 2020



Bahria University
Islamabad Campus
(Department of Computer Science)

CERTIFICATE

We accept the work contained in the report titled "e-Dukan" as a confirmation to the required standard for the partial fulfillment the degree of BS(CS/IT)

Supervisor

Name: Dr. Arif Ur Rahman

Date: 24/07/2020

Internal Examiner

Name: Romana Talat

Date: 23/07/2020

External Examiner

Name: Dr. Sajid Ali Khan

Date: 23/7/2020

Project Coordinator

Name: _____

Date: _____

Head of Department

Name: _____

Date: _____

Abstract

In the current scenario most of the marketing is done online through different applications and websites but there is still one place which is run manually by people which are local shops. In these shops customer has to wait for the turn to get the order delivered since all the work done here is manual so in order to save customer time these local shops need an application which can help the customers in saving their time. There is one way that the shopkeeper buy an application which will fully automate the shop but it will have its own constraints.

The application named e-Dukan is an Android application used to order product and get time slot for the order to be prepared which can be collected by customer from the shopkeeper at the specified time. This Android application is developed to facilitate customer by saving their time and by keeping their privacy. This application saves customer time by providing them order availability time at which the customer can visit the shop and get the order rather than waiting for his/her side to come and get served. This application keeps customer identity and his/her location private which helps in maintaining privacy related issues of customer which may occur when customer book their order from applications like hummart where customer address is not private since the product has to be delivered to the address and there is another issue that customer has no idea about the quality of product since it is ordered online through applications like hummart but in our application customer gets the order at the shop due to which he/she can have check the product quality before buying that product. Customer can select the shop based on rating as well. We have used three tier architecture to develop this application and we have used Agile SDLC model for this application since this is an application which will have various versions later depending on system compatibility.

Acknowledgments

We want to dedicate this project to all of our teachers and our parents who made us capable enough to face such challenges in life.

We specially want to thank our supervisor Dr. Arif ur Rahman who helped us to tackle the difficulties we faced in this project. He helped us by giving his useful advice and sometimes he also do positive criticism which motivated us to give our best to this project.

MUHAMMAD ARSLAN & ZUBAIR SALEEM
Islamabad, Pakistan

February 2020

“Let your customers be your partners; let your vendors be your employees. What’s necessary in this transformation more than anything else is courage and a willingness to change”

Safra A. Catz

Contents

Abstract	i
1 Introduction	1
1.1 Overview	1
1.2 Problem Description	2
1.3 Project Objective	2
1.4 Project Scope	2
2 Literature Review	5
2.1 Designing an Efficient Scheduling System for Multiple Services	5
2.1.1 SWVL	6
2.1.2 Push Doctor	6
2.1.3 Amazon Shopping	7
2.1.4 Hum Mart	8
2.1.5 K&N	9
3 Requirement Specifications	11
3.1 Application Overview	11
3.2 Existing Systems	11
3.3 Proposed System	11
3.4 Product Function	12
3.5 Requirement Specification	12
3.5.1 Software Specifications	13
3.5.2 Hardware Specifications	13
3.5.3 User Requirements	13
3.5.4 System Requirements	13
3.6 Functional Requirements	14
3.7 Non-Functional Requirements	14
3.8 Use Cases	15
4 Design and Implementation	21
4.1 System Architecture	21
4.1.1 Presentation Tier	21
4.1.2 Application Tier	21
4.1.3 Data Tier	21
4.2 Design Constraints	22
4.3 Design Methodology	22
4.4 High Level Design	24

4.5	Low Level Design	27
4.6	Database Design	30
4.7	Features and Modules	32
4.8	System Architecture	32
4.9	Processing Logic	32
4.10	System Internal Components	33
4.10.1	Login/Sign Up	33
4.10.2	Reset Password	33
4.10.3	Shop Category Selection	33
4.10.4	Shop Selection on Map	33
4.10.5	Notification Generation	33
4.11	Tools and Technology	33
4.11.1	Development Environment and Languages	34
5	System Testing and Evaluation	35
5.1	Graphical user interface testing	35
5.2	Usability Testing	35
5.3	Software Performance Testing	36
5.4	Compatibility Testing	36
5.5	Security Testing	37
5.6	Functional Test Cases	37
6	Conclusions	41
6.1	Limitations	42
6.2	Future Work	42
A	User Manual	43
	References	49

List of Figures

2.1	App Interface of SWVL	6
2.2	App Interface of Push Doctor	7
2.3	Amazon App Interface	8
2.4	Hum Mart App Interface	9
3.1	Use Case Diagram	15
4.1	Three Tier Architecture	22
4.2	Agile SDLC Model	23
4.3	High Level Design	24
4.4	DFD level 0	25
4.5	DFD level 1	26
4.6	Flow Diagram	27
4.7	Sequence Diagram Category Selection	28
4.8	Sequence Diagram Shop Selection	29
4.9	Sequence Diagram Product Selection	30
4.10	Sequence Diagram Product Order	31
4.11	Database Structure	31
4.12	3 Tier Architecture	32
6.1	Advantages of Online Shopping	41
A.1	Login Screen	44
A.2	Main Menu	45
A.3	Home Page	46
A.4	Shop Selection Interface	47
A.5	Order Placement Page	48

List of Tables

2.1	Table of Comparison	9
3.1	Software Specifications	13
3.2	Hardware Specifications	13
3.3	Use Case Registration	16
3.4	Use Case Login	16
3.5	Use Case Select Category	17
3.6	Use Case Shop Selection	18
3.7	Use Case Select Product	18
3.8	Use Case Time Slot	19
3.9	Use Case Add to Cart	19
3.10	Use Case Admin Module Add Products	19
3.11	Use Case Admin Module Delete Products	20
3.12	Use Case Admin Module Update Products	20
5.1	GUI Test Case	35
5.2	Usability Test Case	36
5.3	Software performance Test Case	36
5.4	Compatibility Test Case	36
5.5	Security Test Case	37
5.6	Register Test Case	37
5.7	Login Test Case	38
5.8	Select Category Test Case	38
5.9	Shop Selection Test Case	38
5.10	Select Product Test Case	39
5.11	Time Slot Test Case	40
5.12	Admin Module Test Case	40

Acronyms and Abbreviations

DFD	Data Flow Diagram
e-Books	Electronic Books
e-Dukan	Electronic Dukan
GUI	Graphical User Interface
K&N's	Kandns
SDLC	Software Development Life Cycle
XML	Extensible Markup Language

Chapter 1

Introduction

In this chapter we will look in greater depth of the project and give an overview, problem description, project objective and scope of the project. Time is the most valuable thing in a person's life, each second that passes can be costly. A person loses most of his/her time in waiting in a long line for their turn for example visiting the barber shop, if the shop is empty, then you can be free within 15-20 minutes, but if there is a lot of people present in the shop then no one will let you take their turn or move ahead as time is valuable for everyone. You will wait but if the delay is too long then you will eventually move to a different barber shop in order to get your turn or tell the barber to call you when he gets free. Either way your time has been wasted. It would be helpful for the person if he had an application on his smartphone to locate nearby shops in his/her area and then know what time is the shopkeeper available and can go and visit the shop and not have to waste time in waiting for his turn. There is an application made to cater this need but have a different purpose like 'Hum Mart' is an application in which a user can select the fruits and vegetables and can purchase them online and the desired items will be delivered to their doorstep but these applications only are limited to one category. 'The K&N's Way Club' is another example which is designed for specific K&N's stores only. Through the application the user can become a member of the shop, can place order for the shop but for specific locations. The only issue of such application is that it only targets meat and only the members of K&N's. Requirement of such an application is needed for the user who are new to an area so that they can get the locations of nearby shops located in his/her areas which can help them navigate to the shops and also guide them about the price range and can set a timestamp in which they can visit the shop and no time is wasted. The application would give categories of shops like barber, meat and other shops [1].

1.1 Overview

As technology has a huge impact in the lives of people daily technology is increasing at a rapid rate. In almost every place we are connected to the internet via mobile data or WIFI. Now people

don't search for a location on a paper map or ask directions, they simply look it up on the internet and reach their particular location with ease. Now in this modern era people try to keep up with the modern trend of society. e-Dukan can be very helpful as it locates nearby shops of a particular area regarding the type of shop you are looking for and displays the detail of the shop and user can check the time availability and set a particular time which will not hinder their time. SQLite Database is used as it provides with online functionality as our application is working real time to provide accurate communication between the user and the shopkeeper. [2].

1.2 Problem Description

The problem faced by many people is the wastage of time when they visit the shops, if there is a line at the shop people will often wait 5-10min then move on to the next shop, this not only wastes time of the individual but also creates other issues like fuel wastage. Creating an application can help in resolving these issues and help one in making the right choice. More functionalities can be added to the application to enhance the user's experience which can provide customer satisfaction. All the data between the user and shopkeeper is stored which can be used later on in order to resolve issues. Notifications are sent to the user when he/she has set a timestamp to both the shopkeeper and the user so it will let the user know when the time has started and when a specific task the user has given to the shopkeeper is completed. This is an important application for both user and shopkeeper as not only does it saves time of the customer but also provides customer satisfaction to the shop owner.

1.3 Project Objective

The main objective is to:

- Develop a working android application.
- Deploy the developed application on the playstore.
- Provide regular updates based on customer feedback.

1.4 Project Scope

This application is helpful to people living in a particular area or have shifted to a new location which they are not familiar with. The application requires an internet connection to run the application. The data of both the user and shopkeeper is stored in database, it will provide accessibility and feasibility for both the customer and shopkeeper. The Data stored online is a new way to secure the data on cloud.

This application is only available on the play-store and cannot run on apple and windows platforms. This application will not be available for 24 hours a day instead it will only be available

till the shop timings. The in app language is English so user must understand and be fluent in English to use this application.

Chapter 2

Literature Review

The objective of this chapter is to describe the number of methods and ways previously used by different applications to solve the problem. The main objective of this project is to build a single platform which provides the user functionality of different services in a more reliable way. This project is basically designing of a single platform which includes an efficient scheduling system for number of services. After researching through different research papers related to the project topic and understanding the work done by the researches on different services. We have studied the previously done work on two bases. One is the designing of an Efficient Scheduling System. Secondly the online Android based application for providing number of services in a single interface.

2.1 Designing an Efficient Scheduling System for Multiple Services

The concept of an online shopping is a common trend. There are multiple online stores available in market, which works on selling of their products online, and delivering the product to the doorstep. But there is no such availability of platform that will provide the reliability to help save time of a user to utilize services from local market. There are apps available which works on Scheduling system e.g. online service to schedule for the transport, and Online scheduling system for appointment from Doctor. But this facility is not available in case of services that are available in a local market, like Barber Shop, Meat seller, and other services for domestic and non-domestic purposes. Hence to design an improved version of online shopping, the major analysis on the previously implemented applications has been made in terms of Online shopping, Online Services and Appointment based Services [3].

The major review has been done to produce a single and combined platform to provide multiple services and to produce an efficient scheduling system to save the time of a customer and improve the seller and buyer interaction in local market. The improvised version is derived of previously implemented concept, which has been reviewed on the basis of following categorization.

2.1.1 SWVL

It is an efficient scheduling platform for booking a ride for a bus or mini-van for the desired destination. It provides the reliability to schedule their ride before time in very reasonable cost. This Service has been launched in Kenya, Egypt and three main cities of Pakistan to revolutionize the world in terms of providing a digital application based alternate for economical Public Transport. The system expands its operating circle around a city containing number of bus stops and routes to facilitate the customer. The user needs to install the app and after the registration process and successful two factor Authentication, the user can easily be able to book a ride[4]. The starting point can be either selected as Current Location and can be entered manually and then after choosing destination the available rides for the desired route and their time is displayed to the User. Depending on the choice, a User can select a ride from the list of available ride and later can travel by paying the low wages.

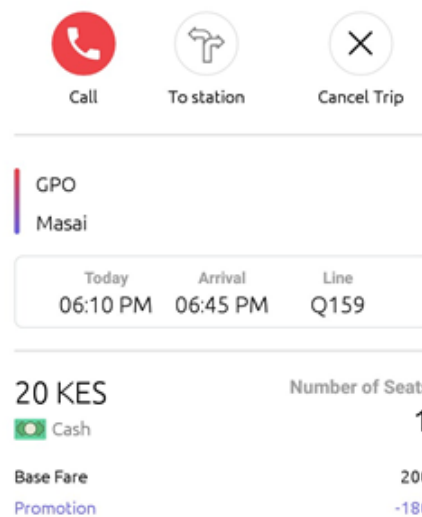


Figure 2.1: App Interface of SWVL

2.1.2 Push Doctor

It is an advanced platform for scheduling the appointment for a specialized doctor. This application is providing 24/7 service around UK. It is an advanced technology based alternate of Doctor Consultation. The service includes the appointment for an Online doctor for consultation through Video call. It also provides services of scheduling online appointment for a visit to a Doctor for desired specialization. First a user needs to register for the Push Doctor online platform. Next the user needs to book an appointment for either online Doctor or 'Push Doctor' registered General Practitioners for a direct visit at suitable time and Place. Choose the desired specialized doctor and select the time slot. In return the doctor can provide you services in term of prescriptions, sick

notes and referrals. Overall the application provides a best solution for saving the time of a patient by providing them quick service for a treatment of ailment using technology.

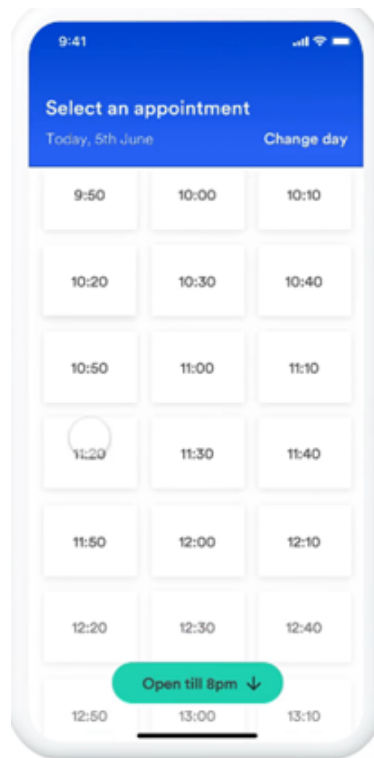


Figure 2.2: App Interface of Push Doctor

2.1.3 Amazon Shopping

Amazon.com is a famous online store which provides services of shopping worldwide. Using this app, a user can directly buy physical items online through using Amazon app. The service also provides convenient delivery service to ship the product at their doorstep. Using the app, a user can manage his/her account credit, previous order and monitor the shipping of the product. The application also provides reliability of purchasing other paid apps and other digital items like games, e-books and subscription for online Streaming services [5].

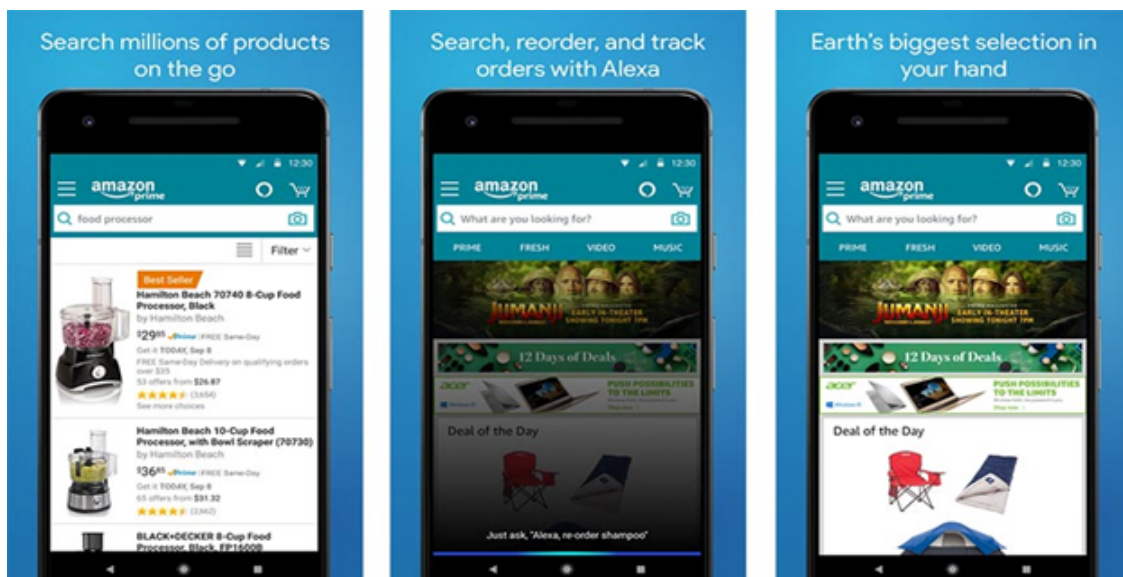


Figure 2.3: Amazon App Interface

2.1.4 Hum Mart

It is an online Grocery Web store to buy necessities and commodities of daily life. It operates only in 'Karachi' city of Pakistan. It provides a comfortable solution for the people of Karachi to avoid hassle of going to market. Hence main aim is to provide the users with a reliable way to shop grocery items and without wasting unwanted time. It provides variety of grocery items which are categorized in Breakfast and dairy, beverages, staples, pulses, and other household items.

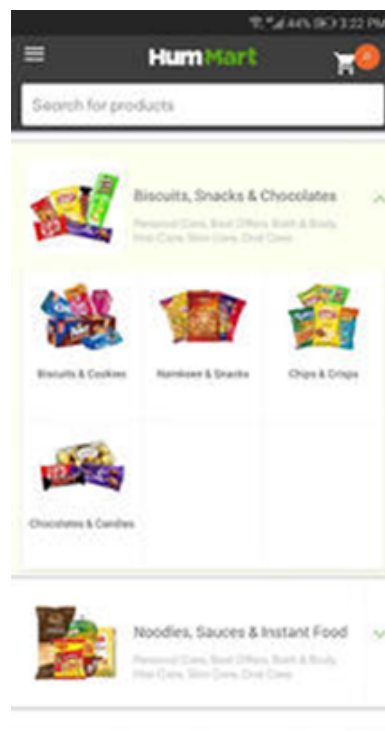


Figure 2.4: Hum Mart App Interface

2.1.5 K&N

K&N's provides an online service to order all K&N's products available at their stores. The desired items will be delivered to the doorstep in a convenient way.

Table 2.1: Table of Comparison

App Name	Publish	Platform	Potential Users	Usability
SWVL	27 Apr 2017	Android,IOS	Travelers	Easy to use
Push Doctor	21 Jan 2016	Android,IOS	Patients	Interactive
Amazon Shopping	10 Dec 2014	Android,IOS	Vendors	Complex
Hum Mart	04 June 2018	Android,IOS	Foodie	Simple
K&N's	15 June 2019	Android,IOS	Chicken lover	Simple

Chapter 3

Requirement Specifications

In this chapter preview of existing systems and requirement specifications of the proposed tool will be discussed.

3.1 Application Overview

The main purpose of the prescribed application is to help in utilizing the free time a user has from his daily schedule. This application is created on Android Studio using Java language and SQLite Database is used to store data.

3.2 Existing Systems

Application like Hum Mart are created to cater this problem like a user can shop online and order the requested things and they are delivered at his doorstep but like all good applications there is a flaw that they don't deliver in the particular area for some reason or their delivery boy is unable to deliver the goods or gets lost or the user gets the opposite of what he wanted.

3.3 Proposed System

The main idea of the application is to utilize the user's time by allocating a time slot to the user and notifying the user when it's his turn or when the order is ready. This also benefits the user who is new to the area by locating nearby shops of different categories. In this application we are using:

- Android Studio
- Java
- XML

- SQLite Database

With the following features:

- User Friendly interface
- Time Stamping
- Locating nearby shops
- Communication between the customer and shopkeeper
- Availability

To develop such an application, we are using Android Studio and using Java Language. In Android Studio we are using the version lollipop as it is easier to code and is compatible with most Android Phones. When a new user will sign in it will be saved in the SQLite Database.

3.4 Product Function

e-Dukan is an Android application which helps in locating nearby shops and to buys products of that shop sitting at home or even you have the option to get the product at the assigneg time from the shop without any wastage of time and privacy. This application has following functionalities:

- Open the e-Dukan Application.
- Sign In/Sign Up.
- User will select the desired category of shop from list.
- Map will display nearby shops of categories selected by user.
- Select a shop and it will display the products and their price available at that shop and there are different time stamps available and also the rating of the shop.
- After the user have selected the product a notification will be sent to the user by the shop keeper about the order delivery time.
- On completion of time another notification will be sent to both the user and the shopkeeper.
- SQLite Database will accurately store the data.

3.5 Requirement Specification

This part of the document describes the features and functionality of the application as well as the criteria to judge the operations of the application.

3.5.1 Software Specifications

Following are the Software requirements of e-Dukan

Table 3.1: Software Specifications

Tools and Technologies	Database
Minimum Api Level	21
Platform Version	Android 5.0
Google Play Services	Enabled
Location Services	Enabled

3.5.2 Hardware Specifications

Table 3.2: Hardware Specifications

Hardware	Requirement
Storage Capacity	Minimum 16GB android phone
Minimum Ram Required	2GB or above
Internet	4 Mbps or above

3.5.3 User Requirements

The app will be used by following users

- **Admin:** The admin will have complete control over the app to verify local vendors and customers and do any modification in the app.
- **Local vendor:** The Local Vendor will be using separate interface to control the appointment of a customer.
- **Customer:** This user will be provided with the interface to schedule an appointment.

3.5.4 System Requirements

The app needs to be efficient in terms of performance, reliability, fast access and availability. To fulfill these a system has two types of requirements:

- Functional Requirements
- Non-Functional Requirements

3.6 Functional Requirements

In this part we have discussed about the functional requirements included in this application which are necessary to run this application.

- User must have registered account in order to place order otherwise the user will only be able to view products.
- Customer can use his/her mobile number to login to the application whereas shopkeeper and admin can sign in to the application by providing other credentials.
- User in case of customer needs to select desired category of shop from category list.
- User can select the desired shop from map and can place the order.
- User in case of shopkeeper will give time stamp to the customer once the customer has ordered the product.
- User in case of admin can add or remove new shops as well as can maintain the product details.

3.7 Non-Functional Requirements

Non-functional requirements play a very important role in the development and testing of the application. Without these requirements this application may have several flaws like if we have not kept in mind about the compatibility of the application then it can not run on most of the devices. Following are the non-Functional requirements we have applied in our application:

- **Maintainability:** In order to maintain this application, we have planned to deploy this application in various builds which will help identify the functionality which needs to be changed or improved through the user feedback.
- **Usability:** This application has an interactive and simple user interface which makes it easier to use. It is designed keeping in mind the types of users.
- **Integrity:** This application is developed in such a way that it maintains previous data. User does not have to worry about finding his/her favorite shop since it is already saved in his/her previous data.
- **Performance:** This application provides a maximum timestamp of 15 minutes for chicken, milk and naan shop whereas it provides a maximum of 60 minutes for barber shop.
- **Scalability:** This application is able to handle large amount of traffic efficiently and can entertain lots of users at the same time.

- **Security:** This application uses two factor authentication for login which makes it way secure than some other applications. To make it more secure we maintain customer and shopkeeper data in our database.

3.8 Use Cases

The following use case is presented to explain the requirements. The figure 3.1 shows the processes and actor involved in this application.

- The first process shows that the customer has to select the shop category in order to proceed further.
- After the customer has selected the shop category the next process is to selected the desired shop from the shown on google map.
- Once the customer has selected the desired shop the next process is to place the order by selecting the desired product and its quantity.
- The next process is to be done by the shopkeeper which is to give time slot to the customer which will be received by the customer in the form of notification.
- The other processes are performed by the admin which are to add and delete the shops as well as to update the products.

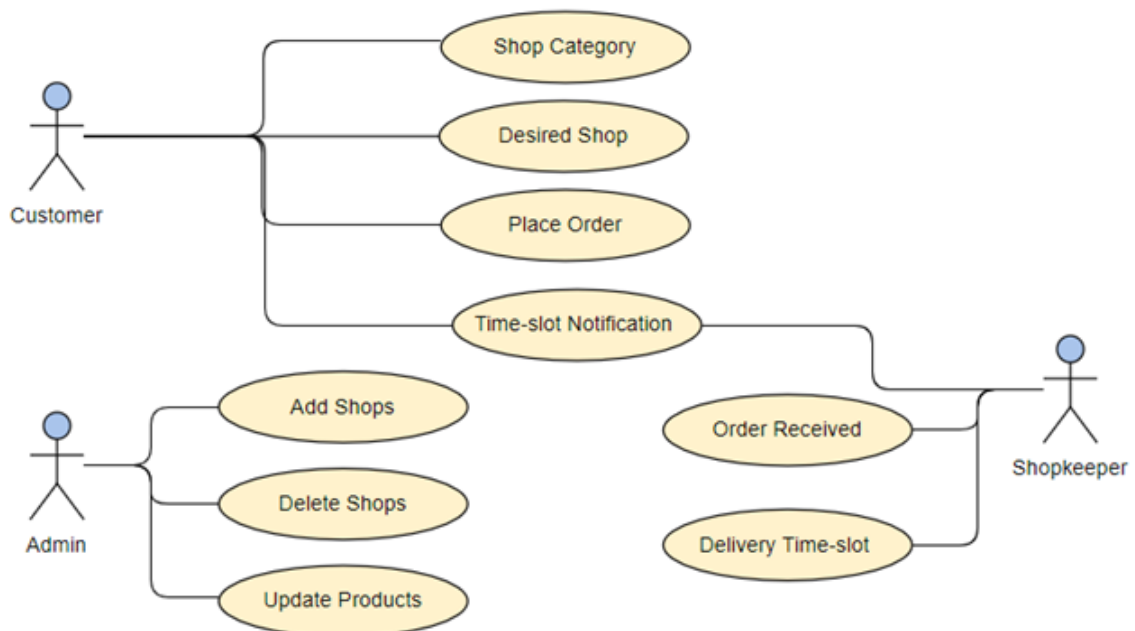


Figure 3.1: Use Case Diagram

This below Use Case diagram explains the actors involved in this application which are customer, admin and shopkeeper. Here the processes are also linked with actors to show the connection that who performs which task.

In table 3.3 Registration Use Case is explained where the actors, pre-condition, post-condition and response of the application is explained. Here it is mentioned that customer need to enter the credentials in order to get the account registered.

Table 3.3: Use Case Registration

Use Case	Details
ID	1
Application	e-Dukan
Description	Customer will provide the required credentials in order to get the account registered.
Primary Actor	Customer,Shopkeeper
Pre-condition	Customer should have our application installed. Customer should be connected to internet connection.
Post-condition	Customer registered sucessfully.
Response	Customer can proceed to login activity.

In table 3.4 Login Use Case is explained through the actors, pre-condition, post-condition and response. In this Use Case it is mentioned that customer and shopkeeper can login to the application using phone number through two factor authentication.

Table 3.4: Use Case Login

Use Case	Details
ID	2
Application	e-Dukan
Description	Customer and shopkeeper will provide the phone number in order to login to the application using two factor authentication. Whereas admin can login using the email and password
Primary Actor	Customer,Shopkeeper and Admin
Pre-condition	Customer should have our application installed. Customer should be connected to internet connection.
Post-condition	Customer logged in sucessfully.
Response	Customer can proceed to shop category selection.

In table 3.5 Select Category Use Case is explained where actors, pre and post conditions are defined. This use case is for the customer where customer as an actor needs to select shop category. The pre condition is that customer needs to have our application installed in the device and should have stable internet connection. The response of this will be that the customer will be directed to the google map activity where the customer can select desired shop.

Table 3.5: Use Case Select Category

Use Case	Details
ID	3
Application	e-Dukan
Description	Customer will select shop category from given list.
Primary Actor	Customer
Pre-condition	Customer should have our application installed. Customer should be connected to internet connection.
Post-condition	Shop category selected
Response	Customer can proceed to shop selection.

In the table 3.6 Shop Selection Use Case is discussed where the actor is again the customer. In this use case customer will select the desired shop from where he/she wants to book an order. The pre condition here is that customer must have first selected the shop category and in response of this activity customer can now place the order.

The table 3.7 represents Select Product Use Case where the actor again is the customer. In this use case customer selects the desired product after selecting the shop from where he/she wants to place the order. In response of placing the order customer will get a notification from shopkeeper about time slot for the order to be ready. The pre condition is that the customer must have selected the desired shop.

In the table 3.8 Time Slot Use Case is discussed where the actors are customer and shopkeeper. In this use case after the customer have selected the desired shop to book an order and has selected the desired product then the customer will receive notification from shopkeeper about the time slot. The pre condition here is that customer must have placed the order.

Table 3.6: Use Case Shop Selection

Use Case	Details
ID	4
Application	e-Dukan
Description	Customer will select the desired shop from google map.
Primary Actor	Customer
Pre-condition	Customer have selected shop category.
Post-condition	Desired shop selected from google map.
Response	Customer can proceed to order placement.

Table 3.7: Use Case Select Product

Use Case	Details
ID	5
Application	e-Dukan
Description	Customer will select the desired product.
Primary Actor	Customer
Pre-condition	Customer have selected desired shop.
Post-condition	Customer order have been placed.
Response	Customer will receive notification from shopkeeper.

The table 3.9 represents Add to Cart use case where the actor is the shopkeeper. In this use case customer has added the product to the cart and order is received by the shopkeeper which was placed by the customer. In response since the order is ready the customer will visit that shop at the allotted time to collect the order. The pre condition is that customer should have received the time slot notification.

In the table 3.10 Admin Module Add Products Use Case is discussed where the actor is admin. In this use case the role of the admin is defined clearly that the admin can add new shops. The pre condition here is that the user must have to be admin in order to perform such tasks.

In the table 3.11 Admin Module Delete Products Use Case is discussed where the actor is admin. In this use case the role of the admin is defined clearly that the admin can delete the shops.

Table 3.8: Use Case Time Slot

Use Case	Details
ID	6
Application	e-Dukan
Description	After customer has selected the product he/she will receive notification about time-slot from shopkeeper.
Primary Actor	Customer, Shopkeeper
Pre-condition	Customer have placed order.
Post-condition	Customer have received time-slot notification.
Response	-

Table 3.9: Use Case Add to Cart

Use Case	Details
ID	7
Application	e-Dukan
Description	Order is received by the shopkeeper.
Primary Actor	Shopkeeper
Pre-condition	Customer have received time-slot notification.
Post-condition	Order is ready to be delivered.
Response	Customer will go to the shop to collect the order.

Table 3.10: Use Case Admin Module Add Products

Use Case	Details
ID	8
Application	e-Dukan
Description	Admin can add new products in the application.
Primary Actor	Admin
Pre-condition	User must be admin
Post-condition	New shop will be added.
Response	-

The pre condition here is that the user must have to be admin in order to perform such tasks whereas post condition is that the shop will be deleted.

Table 3.11: Use Case Admin Module Delete Products

Use Case	Details
ID	9
Application	e-Dukan
Description	Admin can delete shops in the application.
Primary Actor	Admin
Pre-condition	User must be admin
Post-condition	Shop will be deleted.
Response	-

In the table 3.12 Admin Module Update Products Use Case is discussed where the actor is admin. In this use case the role of the admin is defined clearly that the admin can update products within the application. The pre condition here is that the user must have to be admin in order to perform such tasks. The post condition is that the products data will be updated.

Table 3.12: Use Case Admin Module Update Products

Use Case	Details
ID	10
Application	e-Dukan
Description	Admin can update products in the application.
Primary Actor	Admin
Pre-condition	User must be admin
Post-condition	Products will be updated
Response	-

Chapter 4

Design and Implementation

This chapter describes the GUI design of the application e-Dukan. The interface of app is properly explained in the form of block Diagram. In this chapter a comprehensive detail of all the developing tools, algorithms, and Database that are utilized while developing this application.

4.1 System Architecture

In order to build this application, we have used three tier architecture with the help of which we can create link between design, logic and database layers of the application. This will help improve the performance of the application.

4.1.1 Presentation Tier

In this layer of the architecture design part is taken into consideration using XML. This designed layer of the application interacts with other layer through API's.

4.1.2 Application Tier

In this part of architecture, the development phase is done using Java language. This layer takes care of the functionality of the system.

4.1.3 Data Tier

This tier stores the data and provides access to the data maintained within the application. This layer also has a program which allows read and write access to the data of the application.

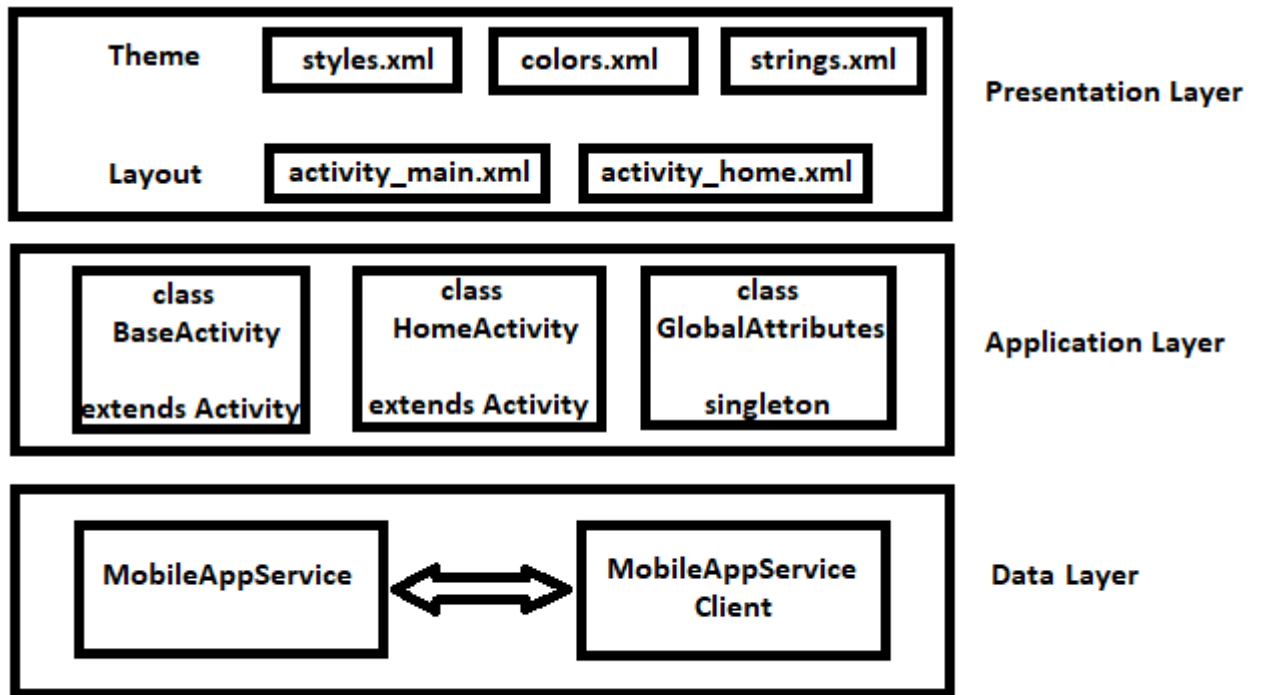


Figure 4.1: Three Tier Architecture

4.2 Design Constraints

- The application requires Android platform above 5.0 to be installed
- The device should have a working internet connection.
- The user needs to have an account to access the application
- The application is highly secured as it provides two-factor authentication to verify a user.
- The application is having an access to Maps to search for shops and local vendors.
- User should have an access to Smartphone and also educated enough to operate the application.

4.3 Design Methodology

This application can have more features in future depending on customers' needs which will be collected from the feedback section. Since we need a system which can accommodate changes in future, we need such a model which can have such a characteristic. In this application customers will register themselves and select the shop and will place the order to get a time slot. Shopkeepers will be registered and must log into the application using two factor authentications to get customer order served. All this process will be monitored, and data will be kept into the database with the

help of admin panel. Admin panel will review the rating of the shopkeepers and will also keep an eye on customer feedback. The model which suits our application is Agile SDLC model. For this application Agile SDLC model is used since we have system which can have various changes after some time depending on user requirements. So, keeping it user friendly this system can get more feature later due to which we have used this model. Agile SDLC model is a blend of iterative and increment procedure with aim to get procedure flexibility and consumer contentment through fast conveyance of the working system[6]. Agile SDLC model breaks the product into iteration and there are specific teams assigned to do those tasks in the following phases.

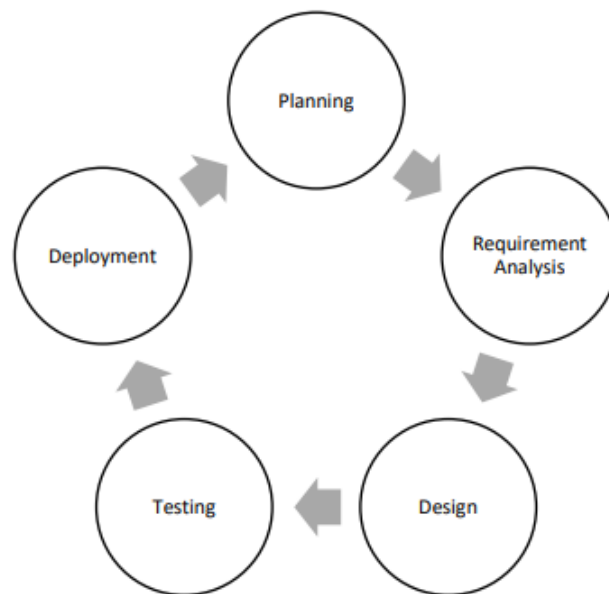


Figure 4.2: Agile SDLC Model

- **Planning Phase:** Starting phase of the system in which ideas are discussed and opinions are made to whether work on the selected plan or not and discussion held about resources.
- **Requirement Analysis Phase:** In this phase consumer needs are considered, and further changes are made through the feedback of the consumer and all the requirements are analyzed by the knowledge workers.
- **Design Phase:** In designing phase, the main concern is about designing an interactive user interface prototype by applying different methods and techniques to get desired output.
- **Coding Phase:** In this phase we will convert the prototype of the user interface into application by coding and by following specific rules.
- **Unit Testing Phase:** In this phase we will have different build of our application depending on the testing done on this application.

- **Acceptance Testing Phase:** In this phase this application will be provided to the consumer for testing to see if the application can be made operational or not. Consumers will test the features and will tell the team about their satisfaction level for this application.

4.4 High Level Design

This high level design mainly tells us how our application is going to be built using the architecture shown below in the diagram. This figure 4.3 gives an overview of the application and its main components. This figure displays the structure where from the front end of the application to the operations performed on the database all this process is interlinked.

In figure 4.3 the high level design of the application is shown. Here we have an android device which shows the front end of the application to the user and is using Google Maps API for selection shops on map activity inside the application. The application sends request to the server and expects a response from the back-end of the application which is the fire base database in our case where products can be added, deleted and updated and queries can be written in order to fetch specific data.

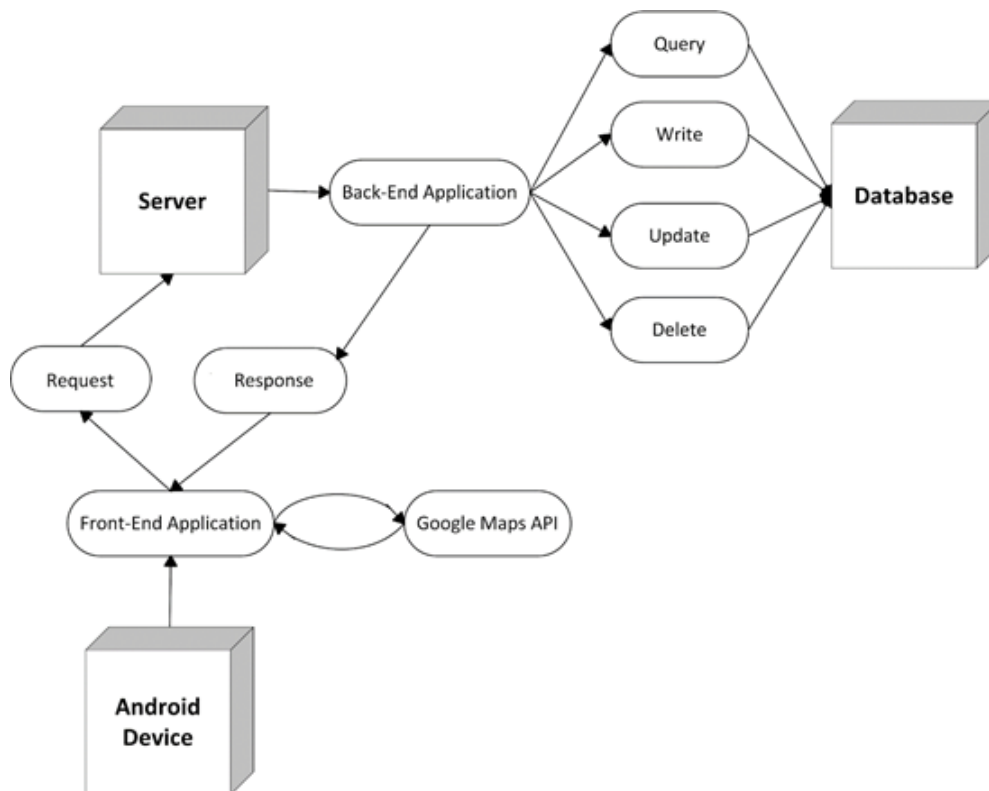


Figure 4.3: High Level Design

Below is the context diagram for the application also called level 0 DFD. In figure 4.4 we have shown the main processes performed among the actors. All these processes get linked to only one main process placed at the center. In this figure all the processes performed by the customer, admin and the shopkeeper are shown through flow lines. This figure clearly shows the boundary of the application. The whole application is displayed through single process.

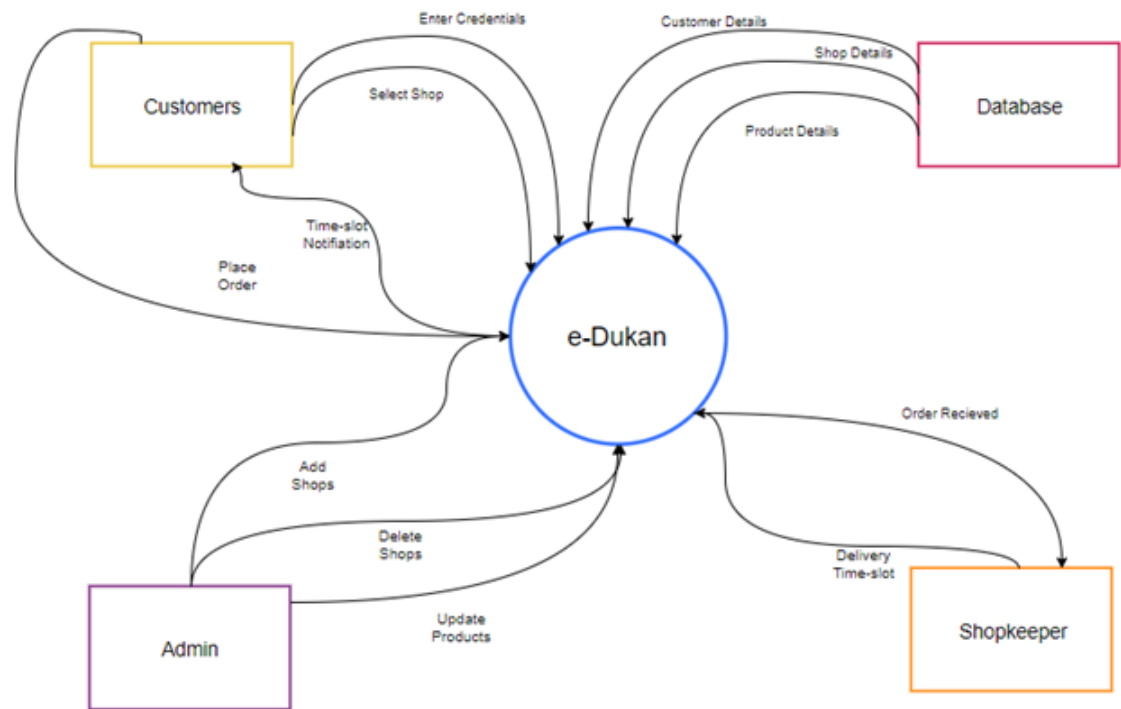


Figure 4.4: DFD level 0

In figure 4.5 the above DFD figure is expanded and is explained in details by splitting one process into many. This is our level 1 DFD diagram which shows the detailed view of data flow.

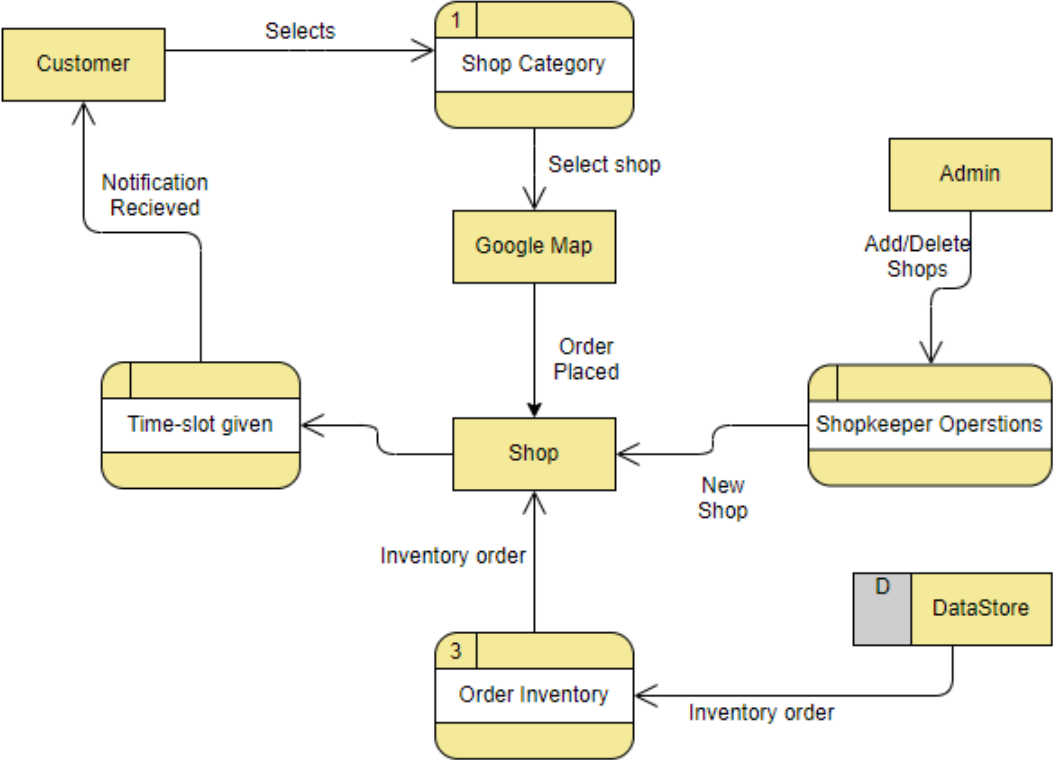


Figure 4.5: DFD level 1

4.5 Low Level Design

The figure 4.6 is a flow diagram showing the flow of the processes performed by this application. This design level helps in explaining components of the application on small level. This design goes deep into the sub components of the application and gives extra details which high level cannot provide to the developers. This diagram shows the flow of processes as well as the conditions applied on those processes.

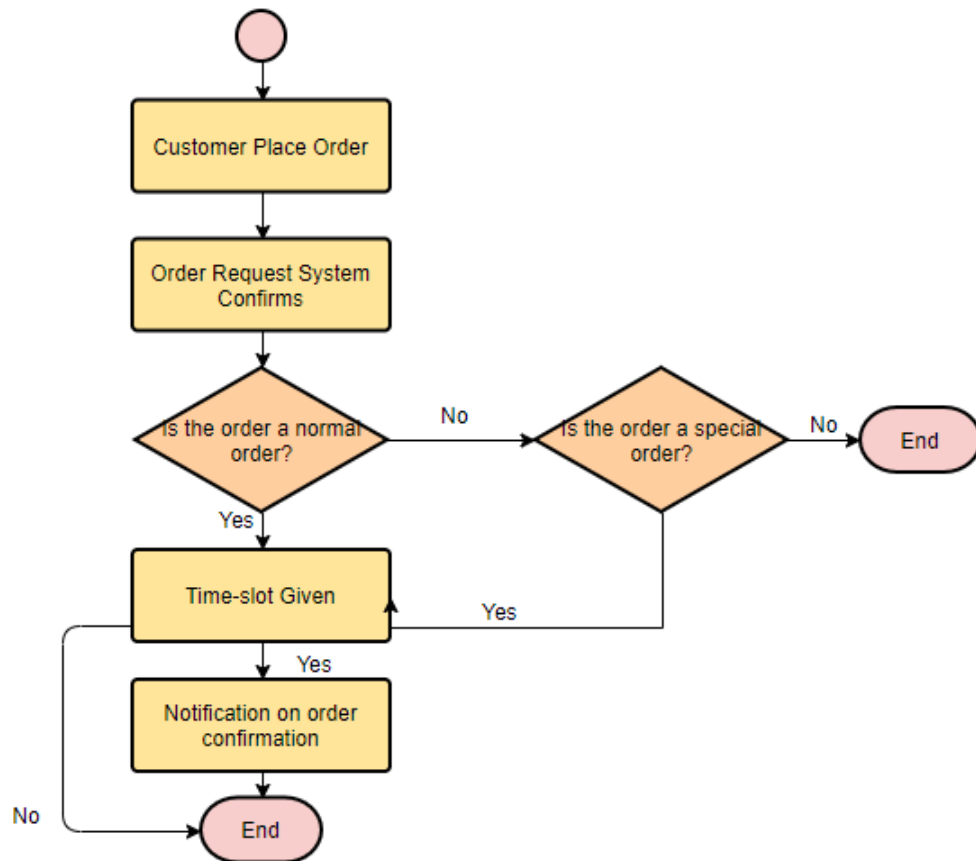


Figure 4.6: Flow Diagram

The figure 4.7 shows the sequence diagram which shows the interaction between the objects of the application. The objects which are shown in the figure are the customer, category and the map. The flow will go in this way that first of all customer will select the shop category and then desired shop will be selected from the map.

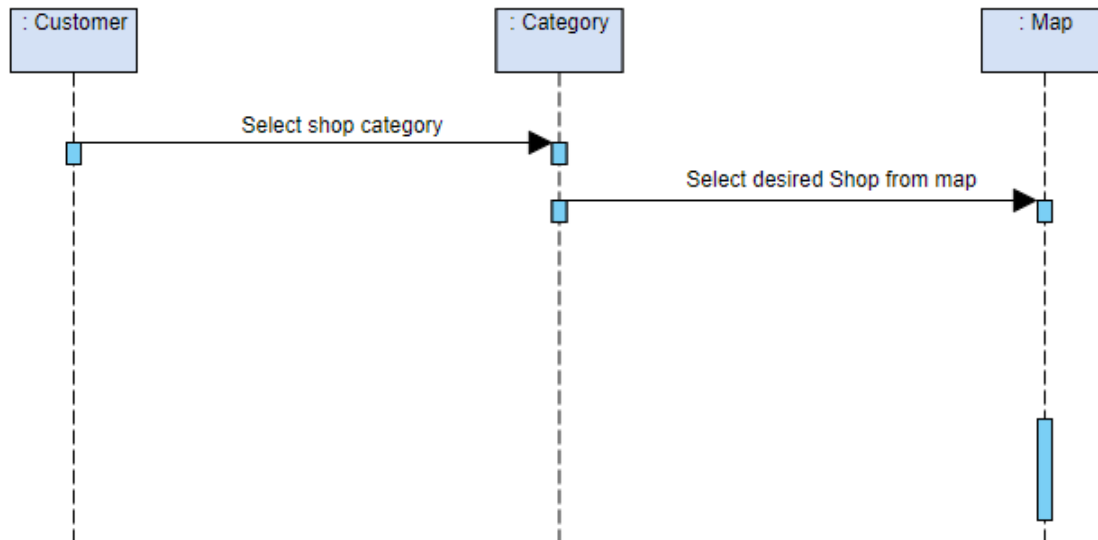


Figure 4.7: Sequence Diagram Category Selection

In the figure 4.8 the next objects of the application are shown and the processes are shown in the forward and backward manner. The object introduced here are the product and the shop. The initial process in this figure is that the customer will order the product and then the shopkeeper will be notified about the order and in the end customer will get confirmation notification from shopkeeper.

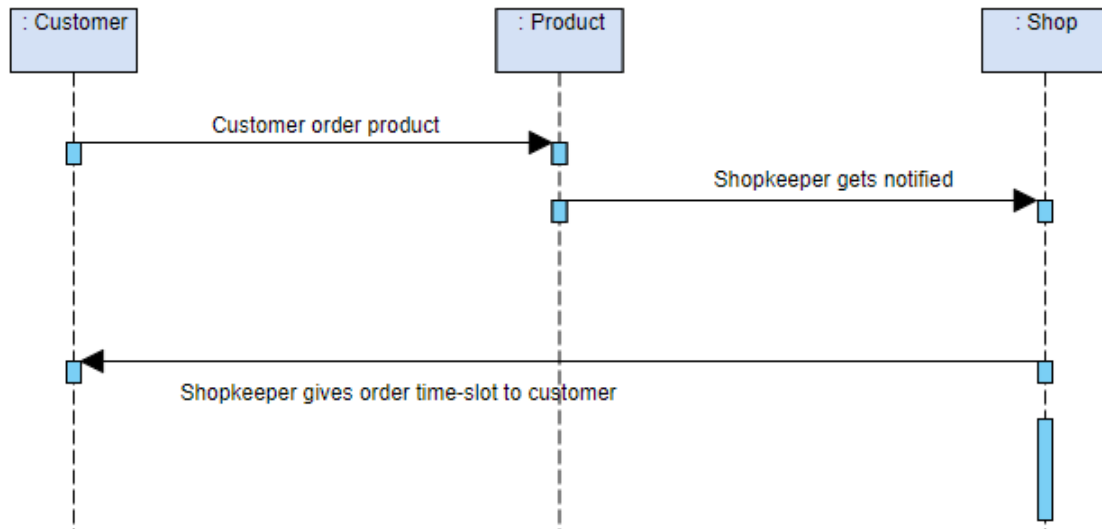


Figure 4.8: Sequence Diagram Shop Selection

In the figure 4.9 the detailed view of the the main objects like product and order and actors like shopkeeper were involved. The figure shows the flow of activities when the customer orders a product.

In the figure 4.10 the detailed view of the sequence diagram is displayed where the main objects and the processes performed by them are shown in a certain manner. The figure shows the main objects like product and shop as well as the actors involved like shopkeeper. The figure shows the link between actors and the objects as well as the processes performed by them.

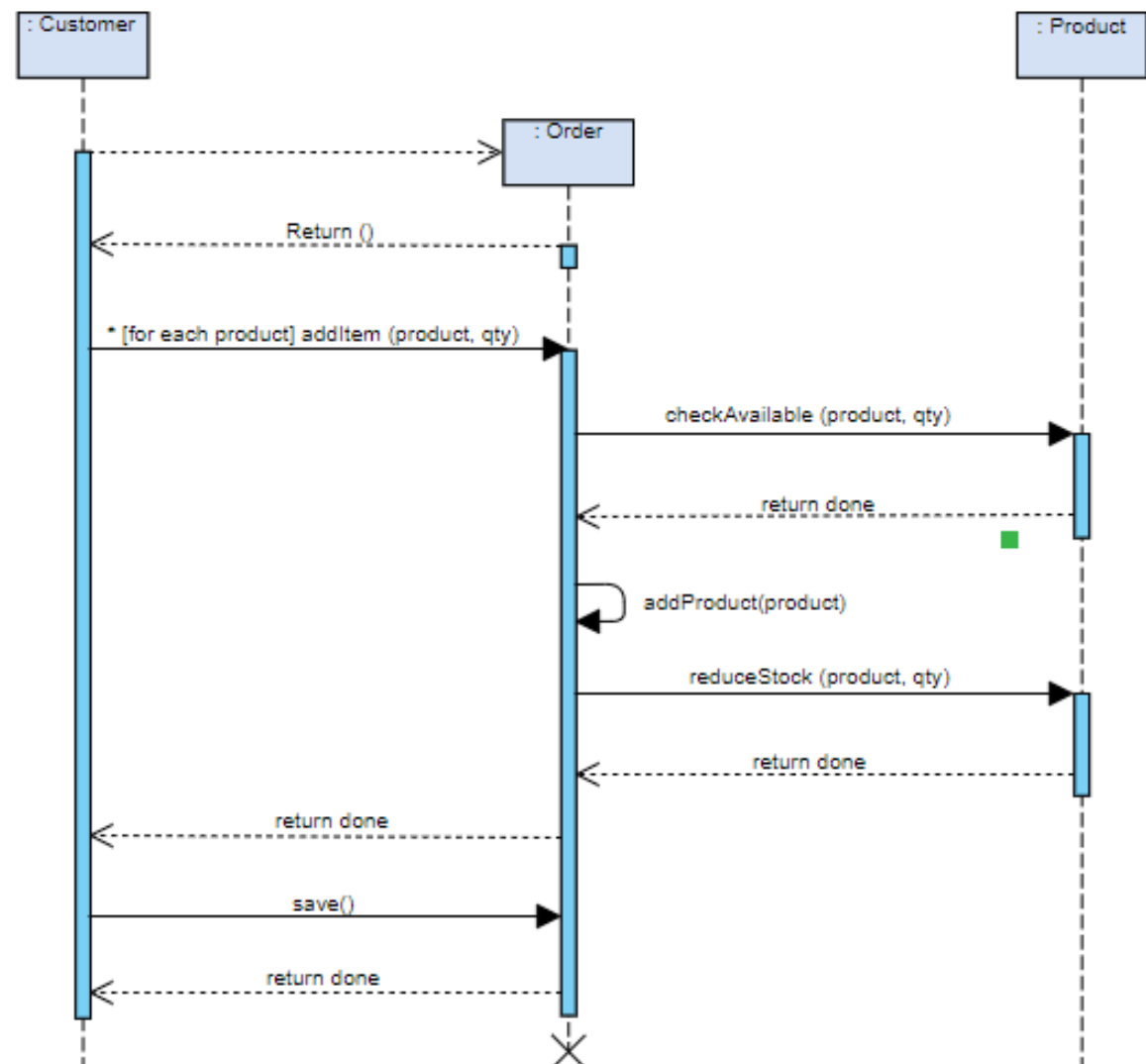


Figure 4.9: Sequence Diagram Product Selection

4.6 Database Design

In this part we use our data model in order to consolidate our data. It tells us that which data we need to store in our db. The figure 4.11 shows the database structure where the whole flow of data within the application is shown in this diagram. In this diagram we have shown the main functions of the system which will be performed in a certain manner.

In this figure the structure of database is explained where we have three classes/actors/entities which have certain attributes and these actors or entities are stored in our SQLite database through their unique id which is primary key in our case so to access some data we use those primary keys.

The three boxes in this figure shows the users of this application whose data is stored in the SQLite database and can be accessed easily.

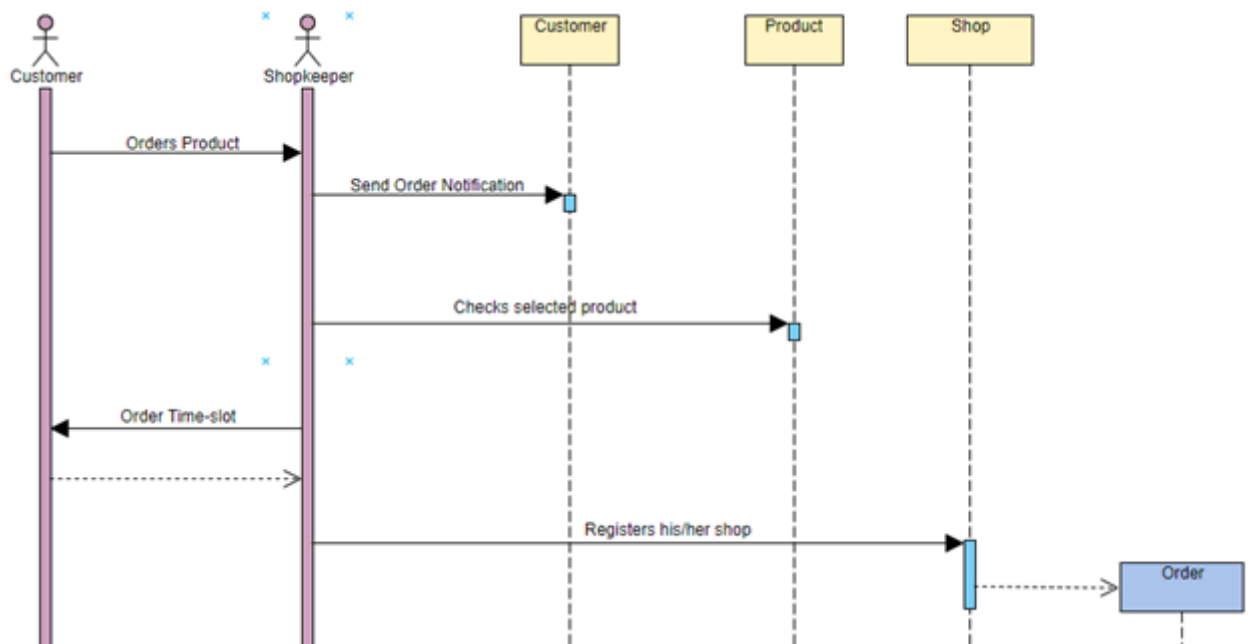


Figure 4.10: Sequence Diagram Product Order

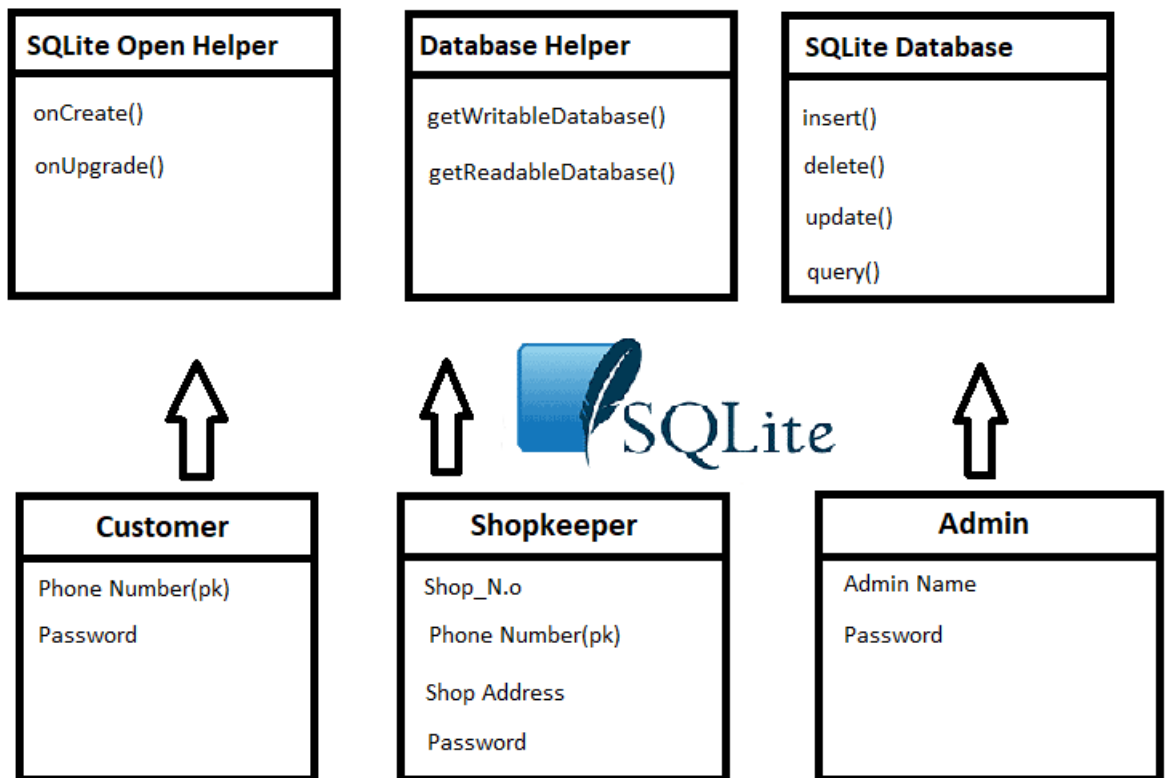


Figure 4.11: Database Structure

4.7 Features and Modules

Following are the features and modules of this application:

- Admin Panel
- User Module
- Shopkeeper Module

4.8 System Architecture

In this part we have discussed the System internal components, Functionality of the components and Communication between the components as well as the architecture of the system which we have discussed earlier will be discussed in terms of the internal features of the system.

So to build this application we have used three tier architecture in which the first tier represents the graphical user interface having the login, sign up and main menu pages in it and is called presentation layer. The second tier includes the functionality used in the application to perform specific tasks like to search nearby shops we have used Google Map API and using the Brute Force Technique and through user reviews we recommend shops to different users and it is application layer. The third tier is mainly used for storing, retrieving, updating and deleting the data or placing the data offline or using real time database[7]. For this we have used SQLite Database and this layer is Data layer at the bottom.

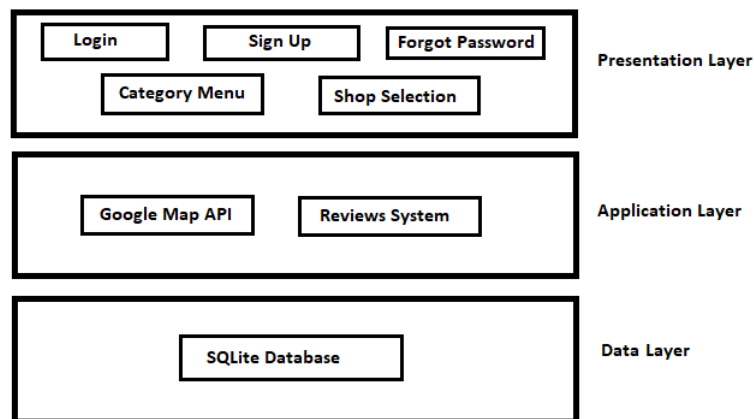


Figure 4.12: 3 Tier Architecture

4.9 Processing Logic

For this application we have used SQLite database to perform CRUD operations on user data. We have used this database because it does not connect app to SQLite through http request rather it

connects through Websockets which are much faster than http. In terms of security it way much secure than other databases since its GCloud is the best solution to security issues that might occur[8].

Authentication is also not an issue while using this database because it provides in app email and password authentication system. In terms of hosting SQLite uses global CDN with HTTP/2 which is very easy to use service.

4.10 System Internal Components

4.10.1 Login/Sign Up

Here the user will first see the login user interface. If the user is not registered then he/she has to click on Sign Up option through which user will get the Sign Up interface where he/she can put his/her credentials and get registered.

4.10.2 Reset Password

This part is very helpful for user if they forgot their previous password they can enter their email to get reset password link which will be generated afterwards.

4.10.3 Shop Category Selection

After the user has successfully logged in to the account the next thing they need to do is to select the category of shop from the given list like barber, milk or chicken shop etc.

4.10.4 Shop Selection on Map

Since the user has selected the shop the next thing is to select the shop from the map based on user choice or based on the recommended location fetched by the application through user reviews.

4.10.5 Notification Generation

After the user has selected the shop and placed the order the shopkeeper will receive a notification about the order and the shopkeeper will give a time-slot to the user through the notification message generated by notification manager.

4.11 Tools and Technology

The tools and technologies used to complete this application are:

Hardware:

- Android Phone

Software:

- Android Studio
- Windows 10
- Java
- SQLite Database

4.11.1 Development Environment and Languages

The least requirements needed for the system are:

- i3 5th generation processor
- 2gb RAM
- Windows 10 64-bit

Languages Used:

- Java
- Sql

Chapter 5

System Testing and Evaluation

In this chapter we will discuss about the types of testing performed on this application and the evaluation of the different components and features of the system. We will also discuss about the performance of the system and the test cases which we implemented on the system.

5.1 Graphical user interface testing

We applied GUI testing by comparing our user requirements defined in the project proposal with the interface of our application and tried to find differences to make sure all functionalities are implemented.

Table 5.1: GUI Test Case

Test Case ID	TC-1
Description	In this test case we have discussed about graphical user interface and its issues
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	Check the login and sign up interfaces
3	Check the map activity and the main menu if it is correctly working

5.2 Usability Testing

In this part we discuss about the user interfaces and the issues which might occur and see if the user required interface it being designed or not.

Table 5.2: Usability Test Case

Test Case ID	TC-2
Description	In this test case we have discussed about the usability issue that may occur in our application
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	Check the interfaces to see if user requirements are met
3	Compare the interface with the required one mentioned in the project proposal

5.3 Software Performance Testing

In this testing we checked the application responsiveness and the load time of our application.

Table 5.3: Software performance Test Case

Test Case ID	TC-3
Description	In this test case we have discussed about the performance of the application
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	Check if the application launch time is long or not
3	Check if the application response time is better or not

5.4 Compatibility Testing

Here in this testing we checked whether our application runs on low end devices or not by checking the build version.

Table 5.4: Compatibility Test Case

Test Case ID	TC-4
Description	Here we have discussed about the compatibility of the application
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	Check if the app can run on minimum of build versions
3	Check if the application crashes on old android phones or not.

5.5 Security Testing

In this part we checked if our application is free from the risk of unauthorized access or not and checked user privacy level.

Table 5.5: Security Test Case

Test Case ID	TC-5
Description	Here we have discussed about the security of the application
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	Check if the app allows invalid users to use the app
3	Check if the encryption is applied on the password

5.6 Functional Test Cases

In table 5.6 we have discussed the Register Test Case in which the process of registration is tested to check if this functionality of the application is working fine or not.

Table 5.6: Register Test Case

Test Case ID	TC-6
Description	Here we have discussed about the registration process of the application
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	Provide the required credentials.
3	Check if the account gets registered or not.

In table 5.7 we have discussed the Login Test Case in which login functionality is tested to check if it is working or not. To check this functionality in case of customer we guided the customer to use the mobile number in order to login to this application using two factor authentication.

In table 5.8 we have discussed the Select Category Test Case in which the shop category selection functionality is tested to check if it is working or not. To check this functionality we asked the customer to select different categories of shop and hence there were no issues in that part of the application.

Table 5.7: Login Test Case

Test Case ID	TC-7
Description	Here we have discussed about the login functionality of the application
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	Provide the required credentials in order to login.
3	User in case of customer and shopkeeper should enter the mobile number to login.
4	User in case of admin should enter the email and password to login.
5	Check if the user logged in successfully or not.

Table 5.8: Select Category Test Case

Test Case ID	TC-8
Description	Here we have discussed about the category selection functionality of the application
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	Select the desired category of shop.
3	Check if this functionality is working properly by ensuring that the application moves to the next activity.

In table 5.9 we have discussed the Shop Selection Test Case in which the shop selection functionality is tested to check if it is working or not. In this part of the application user has to select the shop from a map activity so we checked if the map activity is working fine or is it taking time but it was working fine.

Table 5.9: Shop Selection Test Case

Test Case ID	TC-9
Description	Here we have discussed about the shop selection functionality of the application
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	Select the desired shop from the map activity which will take you to the e-store of that shop.
3	Check if this functionality is working by making sure that e-store of that shop is opened in the application.

In table 5.10 we have discussed the Product Selection Test Case in which the product selection functionality is tested to check if it is working or not. In this part of the application user has to select the product from the e-store of the online shop. Here we checked the e-store of the shop as well as the add to cart button and quantity button within the interface.

Table 5.10: Select Product Test Case

Test Case ID	TC-10
Description	Here we have discussed about the product selection functionality of the application
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	Select the desired product from the e-store of the shop.
3	Enter the quantity in which you want that product.
3	Check if all these functionalities are working fine.

In table 5.11 we have discussed the Time Slot Test Case in which the time slot notification functionality is tested to check if it is working or not. In this part of the application after the customer has selected the product and has added it to the cart a notification generation occurs which is tested at both customer and shopkeeper ends.

Table 5.11: Time Slot Test Case

Test Case ID	TC-11
Description	Here we have discussed about the time slot assigning functionality of the application
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	After the product is added to cart user must receive a pop up notification.
3	The pop up notification will provide the details of the product and delivery time.

In table 5.12 we have discussed the Admin Module Test Case in which the admin module of the application is tested to check if it is working or not. In this part of the application an admin is allowed to add, delete and update the products of different shops at any time they want to.

Table 5.12: Admin Module Test Case

Test Case ID	TC-12
Description	Here we have discussed about the admin module functionality of the application
Minimum API Level	Android version 4.4: KitKat
Conditions	Feasible environment must be available
Task No.	Results
1	Launch the application
2	Admin should be able to add new products to the e-store of the shop.
3	Admin should be able to delete products from the e-store of the shop.
2	Admin should be able to update product details to the e-store of the shop.

Chapter 6

Conclusions

These days people do not have much time to waste going to shops and waiting for their turn so to ease them this application is the solution. It helps the user to get their order on time that too without having to tell their address they just have to get their order from shop on allocated time which is very easy and convenient.

People who are using other food delivery apps may find them easy to use but their privacy is risked since they have to tell their complete address to get the order delivered whereas in our application it is totally dependent on customer will whether he/she wants to get the order delivered or they want to get the order from shopkeeper.

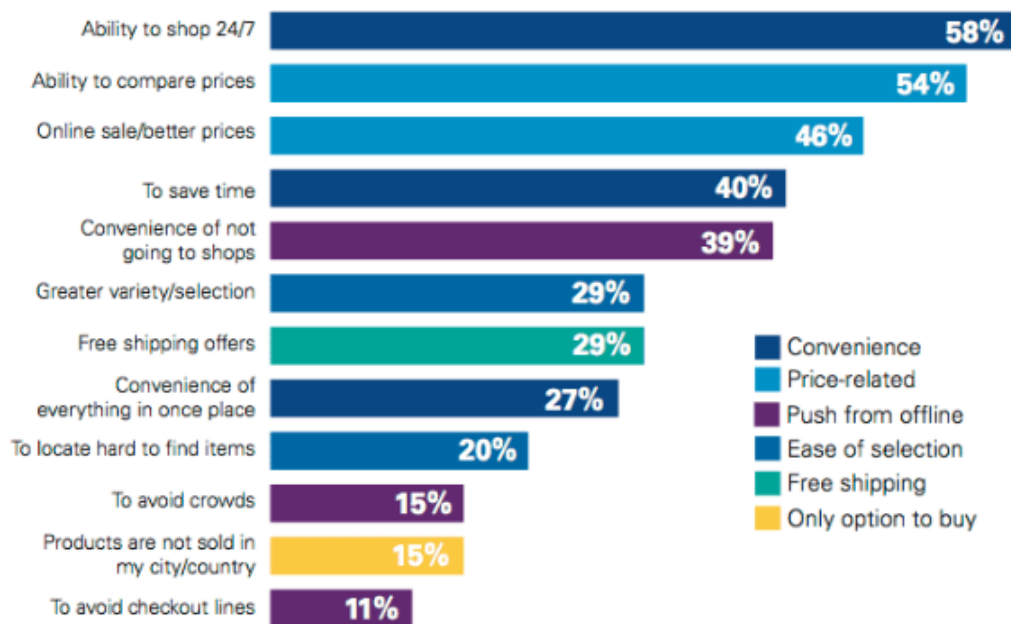


Figure 6.1: Advantages of Online Shopping

This application also helps the customers who are new to that place or people who have no idea about the nearby shops so he/she can select the category of shop and can get the order delivered or can take the items from the shop.

6.1 Limitations

Even though this application is user friendly but there still remains some issue which might occur:

- The first issue we might face is lack of discount on products so we will try to give offers on different products through this application.
- The second issue is that there is less interaction between customer and shopkeeper due to which there is no chance of bargain. To cope up with this issue we will try to bring more features to our app to improve interaction between them.
- There is another issue which may occur that sometimes customer did not like the product and want to return it but in order to return the product customer has to bear all the cost of returning the product.

6.2 Future Work

- We will try to update this app on regular basis based on user requirements
- We will try to bring new features to this application like use of microphone for communication.
- In order to make the app more secure we will try to add more features like two factor authentication.

Appendix A

User Manual

e-Dukan is an easy to use android application which helps in providing time slot for ordered product by the customers as well as helps the shopkeeper in taking care of its customers.

The interfaces for this application includes three parts. One is for the user who is customer and they can login to the application just by their phone number and same is the case for shopkeeper. But for admin we have seperate module in which admin has to enter certain credentials to get access to the application.

The figure A.1 shows the login interface of this e-dukan application in which user, admin and shopkeeper can login by just using their phone number through two factor authentication.

Login

Customer Shop Owner

Phone Number or Admin Username

Password

LOGIN

Don't have account? Register Now >

Figure A.1: Login Screen

In figure A.2 the interface shows the main menu of the application which consists of a navigation bar which consists of three sections including home, categories and exit.

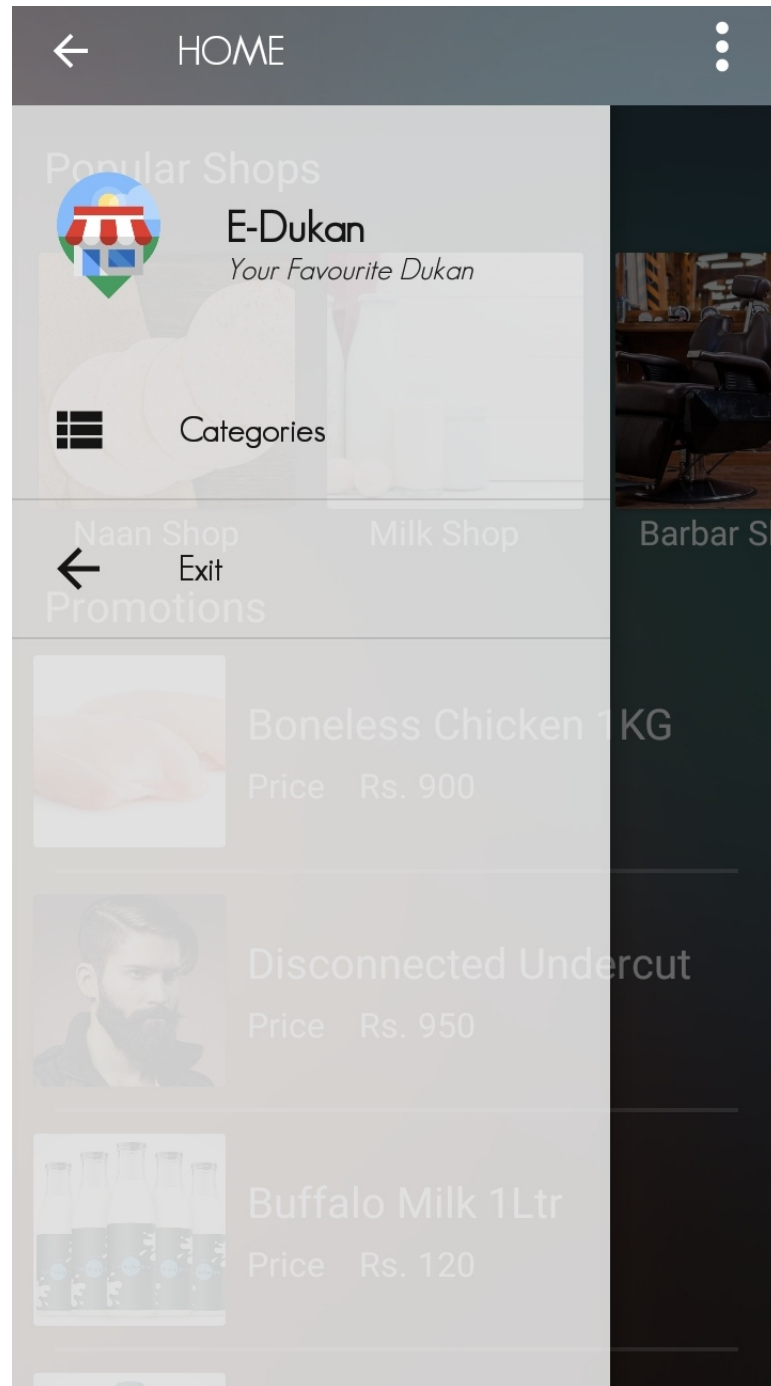


Figure A.2: Main Menu

In figure A.3 the interface shows the home page of the application which consists of a navigation bar which consists of two sections including popular shops and promotions.

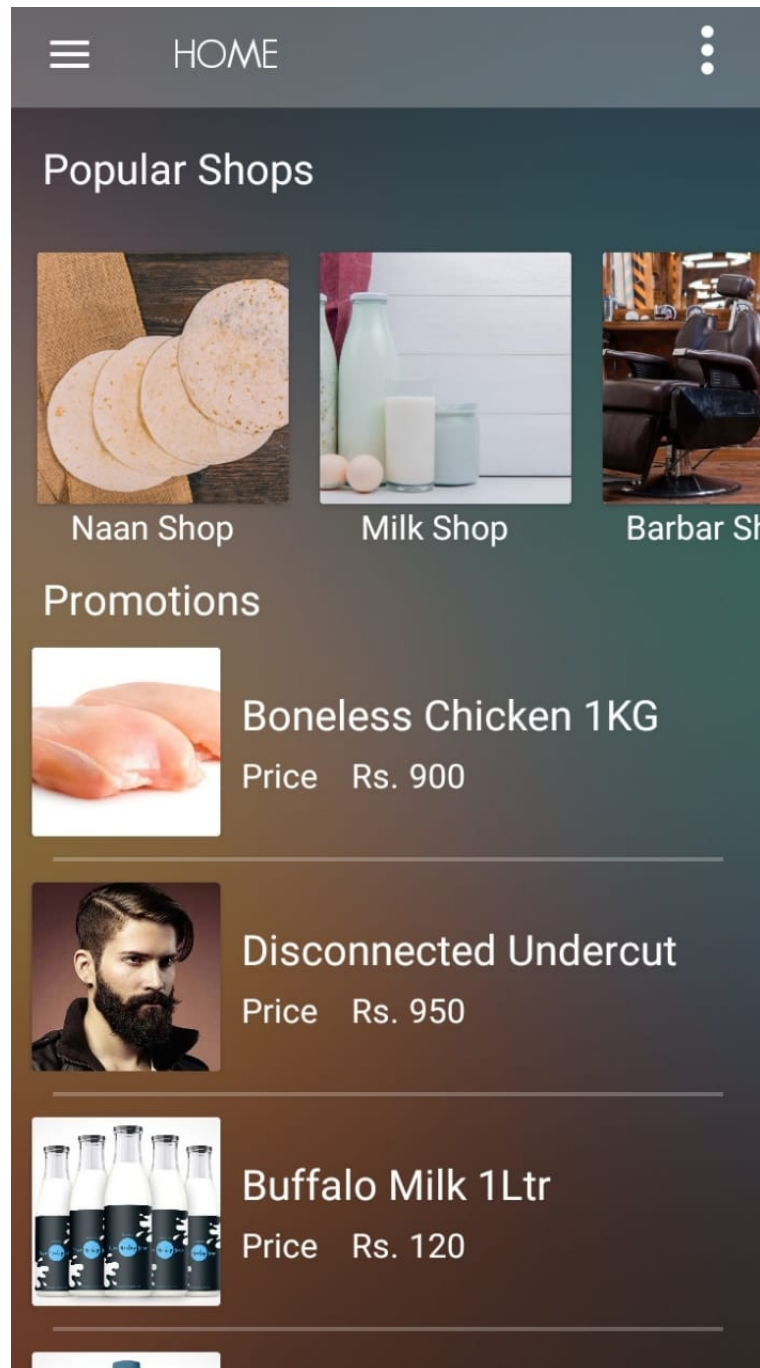


Figure A.3: Home Page

In figure A.4 the interface shows the map activity of the application from where user can see their selected shop and order different products. The map is currently showing the location of user.

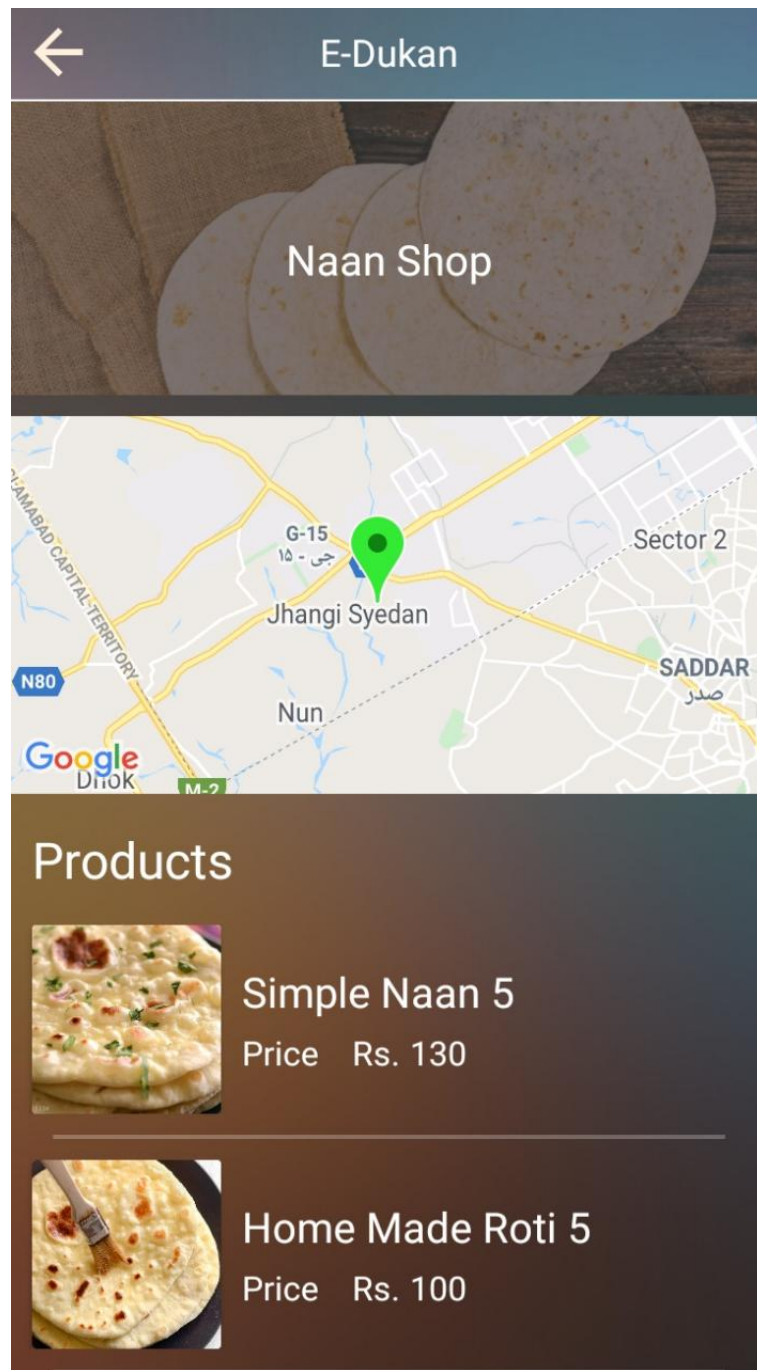


Figure A.4: Shop Selection Interface

In figure A.5 the interface shows the order placement page of the application which consists of product price, rating and comments given by the user. From here user can add the product to cart.

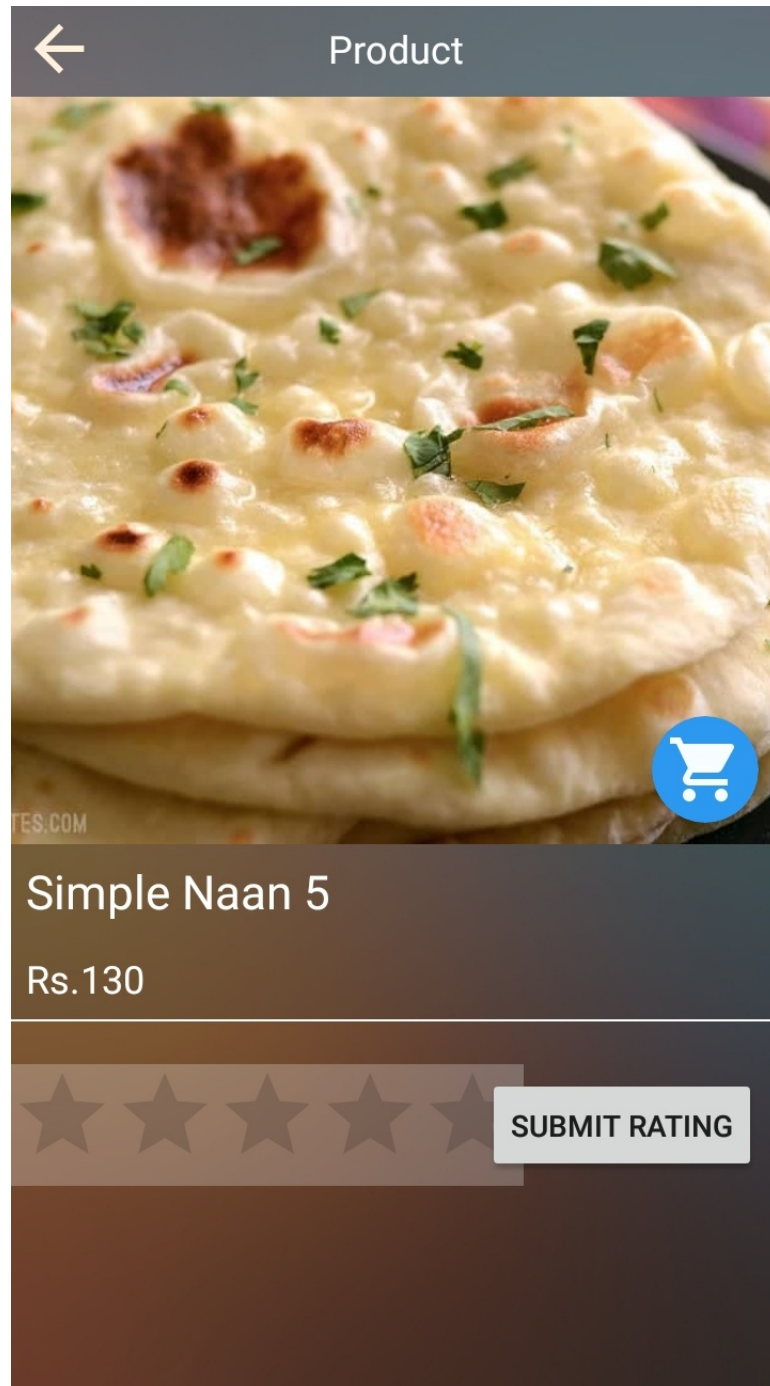


Figure A.5: Order Placement Page

References

- [1] Rebecca Jen-Hui Wang, Edward C Malthouse, and Lakshman Krishnamurthi. How mobile shopping affects customer purchase behavior: A retailer's perspective. In *Let's Get Engaged! Crossing the Threshold of Marketing's Engagement Era*, pages 703–704. Springer, 2016. Cited on p. 1.
- [2] Nilanjan Chatterjee, Souvik Chakraborty, Aakash Decosta, and Asoke Nath. Real-time communication application based on android using google firebase. *International Journal of Advance Research in Computer Science and Management Studies*, 6(4), 2018. Cited on p. 2.
- [3] O Osunade, A Osho, and S Oyebamiji. Android appointment manager application development with google apis. *Transnational Journal of Science and Technology*, 4(2):80–92, 2014. Cited on p. 5.
- [4] Joydeep Sarkar, Yadnesh Khode, Shubham Jadhav, and Akshata Laddha. Car rental system for maharashtra (android app). 2019. Cited on p. 6.
- [5] Mingyung Kim, Jeeyeon Kim, Jeonghye Choi, and Minakshi Trivedi. Mobile shopping through applications: Understanding application possession and mobile purchase. *Journal of Interactive Marketing*, 39:55 – 68, 2017. Cited on p. 7.
- [6] Marian Stoica, Marinela Mircea, and Bogdan Ghilic-Micu. Software development: Agile vs. traditional. *Informatica Economica*, 17(4), 2013. Cited on p. 23.
- [7] Karina Sokolova, Marc Lemercier, and Ludovic Garcia. Android passive mvc: a novel architecture model for the android application development. In *International Conference on Pervasive Patterns and Applications*, pages 7–12, 2013. Cited on p. 32.
- [8] Margaretha Ohyver, Jurike V Moniaga, Iwa Sungkawa, Bonifasius Edwin Subagyo, and Ian Argus Chandra. The comparison firebase realtime database and mysql database performance using wilcoxon signed-rank test. *Procedia Computer Science*, 157:396–405, 2019. Cited on p. 33.