

GRAMMO

BY

Abu Al Ibna Hossain

ID: 183-16-392

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computing and Information System

Supervised By

Abdullah Bin Kashem Bhuiyan

Lecturer

Department of CIS

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
21 NOVEMBER, 2022

APPROVAL

This Project titled “Grammo”, Submitted by Abu Al Ibna Hossain, ID No: 183-16-392 to the Department of Computing & Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information Systems and approved as to its style and contents. The presentation has been held on- 21-11-2022.

BOARD OF EXAMINERS



Mr. Md Sarwar Hossain Mollah

Chairman

Associate Professor and Head

Department of Computing & Information Systems

Faculty of Science & Information Technology

Daffodil International University



Mr. Abdullah Bin Kasem Bhuiyan

Internal Examiner

Lecturer

Department of Computing & Information Systems

Faculty of Science & Information Technology

Daffodil International University



Mr. Md. Mehedi Hasan

Internal Examiner

Lecturer

Department of Computing & Information Systems

Faculty of Science & Information Technology

Daffodil International University



Dr. Saifuddin Md. Tareeq

External Examiner

Professor & Chairman

Department of Computer Science and Engineering

University of Dhaka, Dhaka

DECLARATION

We hereby declare that, this project has been done by me under the supervision of **Abdullah Bin Kashem Bhuiyan, Lecturer, Department of CIS** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

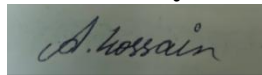
Supervised by:



Abdillah Bin Kashem Bhuiyan Lecturer

Department of CIS
Daffodil International University

Submitted by:



(Abu Al Ibna Hossain)

ID: 183-16-392
Department of CIS
Daffodil International University

ACKNOWLEDGEMENT

First, i express our heartiest thanks and gratefulness to almighty God for His divine blessing makes me possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Abdullah Bin Kashem Bhuiyan Lecturer**, Department of CIS Daffodil International University, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of “**Grammo**” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express our heartiest gratitude to **Professor Mr. Md Sarwar Hossain Mollah** Head, Department of CIS, for his kind help to finish our project and also to other faculty member and the staff of CIS department of Daffodil International University.

I would like to thank my entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Bangladesh is an agricultural country. Her economy mainly depends on agriculture. In village most of the family depends on agriculture. After freedom farming area was Bangladesh's super monetary main thrust. That time the contribution of our economy is around 60 percent. But in Bangladesh the situation of the farmers is too bad. Because they do not get enough for their products. On the other hand, in market place the price of their products is high. The reason of this inequality is third party deal. Simply this is called syndicate. In Bangladesh the number of syndicates is too high. For this they do not get enough value who deserve this. That's why the situation of farmers in our country is worse than the other's countries farmers. For this reason, I develop a mobile app named grammo. Grammo is a mobile app where the farmers create account and can create their account and can sell their products directly to the real customer by using Grammo app. In our country maximum farmer and garden owner are illiterate. Sometimes they cannot find out the actual problem of his plant why they cannot get their aimed crop or can find out the problem but don't know how to solve this. For this reason, the grammo app has a feature. If any farmer or garden owner contact with us and ask for help, The grammo team will help to them. The grammo team will also sell the gardening plants to the customer in a reasonable price. Thus, we can help to the poor people and can help our economy. I think it will be very effective for our society.[1]



TABLE OF CONTENTS

CONTENTS	PAGE NO
Board of examiners	ii
Declaration	iv
Acknowledgments	iv
Abstract	vi
Table of Contents	vii-xi
List of Figures	xi-xii
List of Tables	xiii

CHAPTERS	PAGE NO
CHAPTER 1: INTRODUCTION	1
1.1 Introduction	1
CHAPTER 2: Initial Study	(2-7)
2.1 Project proposal	2
2.2 Background of the project	5
2.3 Problem area	6
	7
2.4 Possible solution	

CHAPTER 3: Literature Review	(8-12)
3.1 Discussion on problem domain based on published articles	8
3.2 Discussion on problem solution based on published articles	9
3.3 Comparison leading solutions	10-12
CHAPTER 4: Methodology	(13-18)
4.1 What to use	13
4.2 Why to use	13-14
4.3 Sections of methodology	14-15
4.4 Implementation plans	16-18
CHAPTER 5: Planning	(18-22)
5.1 Project plan	18-19
5.2 Test plan	19-22
CHAPTER 6: Feasibility testing	(23-26)
6.1 Different types of feasibility	23-24
6.2 Budget and schedule risk	25
6.3 DSDM (Dynamic System Development Method)	26
CHAPTER 7: Foundation	(27-30)
7.1 Problem area identification	27
7.2 Rich picture	27
7.3 Requirement	28

7.4 Technology to be implemented	29-30
7.5 Recommendation and justification	30
CHAPTER 8: Exploration	(31-35)
8.1 Old use case diagram	31
8.2 Activity diagram	32
8.3 Full system use case diagram	33
8.4 Full system activity diagram	34
8.5 Prototype of the new system	35
CHAPTER 9: Engineering	(36-48)
9.1 New systems module is given below	36
9.2 Use case details information	38
9.3 Class Diagram	40
9.4 ERD	41
9.5 Sequence Diagram	42
9.6 Component diagram	43
9.7 Development diagram	44
9.8 system interface and design	45-48

CHAPTER 10: Development	(49-60)
10.1 Coding sample	49-59
10.2 Possible problem break down	60
CHAPTER 11: Engineering	(61-84)
11.1 Acceptance testing	61-62
11.2 Test case	63-64
11.3 Unit testing	65-67
11.4 Module testing	68-70
11.5 Integration testing	71-72
11.6 Acceptance testing	73-78
11.7 Usability testing	79-84
CHAPTER 12: Implementation	85
12.1 Training	85
12.2 Big Bang (no pilot, parallel implementation scheme)	85
CHAPTER 13: Engineering	(86-87)
13.1 Critical Appraisal and evolution	86
13.2 Object that could be met	87

CHAPTER 14: Lesson learned	(87-88)
14.1 Pre-project -Review -Closing	87
14.2 What I have learned	88
CHAPTER 15: Conclusion	(88-89)
15.1 Summary of the project	88
15.2 Goal of the project	88
15.3 Success of the project	89
15.4 Value of the project	89
15.5 My experience	89

LIST OF FIGURES

Figure 3.3.1 user panel	12
Figure 4.4.1 DSDM	16
Figure 5.5.1 Gantt Chart	19
Figure 5.2.1 Client acknowledgement test plan	20
Figure 7.2.1 Rich picture	27
Figure 8.1 Old Use case	31

Figure 8.2 Old activity diagram	32
Figure 8.3 New use case	33
Figure 8.4 Full activity diagram	34
Figure 8.5 Prototype of new system	35
Figure 9.1.1 New system module	36
Figure 9.3 Class diagram	40
Figure 9.3 Class diagram	41
Figure 9.5 Sequence diagram	42
Figure 9.6 Component diagram	43
Figure 9.7 Development diagram	44
Figure 9.8.1 Registration page	45
Figure 9.8.2 Verify page	46
Figure 9.8.3 Home page	47
Figure 11.1.1 Profile update	61
Figure 11.1.2 Purchase figure	63
Figure 9.7 Development diagram	44
Figure 9.8.1 Registration page	45

LIST OF TABLES

TABLE	PAGE NO
Table 5.1: Time boxing	16
Table 6.2: Cost benefit analysis	26
Table 9.1.2 Use case table	37-39
Table 11.2.1 Test case	63-64

CHAPTER 1

INTRODUCTION

The system I developed:

Grammo is a mobile app where the farmers create account and can create their account and can sell their products directly to the real customer by using Grammo app. Sometimes farmer/garden owner cannot find out the actual problem of his plant why they cannot get their aimed crop or can find out the problem but don't know how to solve this. For this reason, the grammo app has a feature. If any farmer or garden owner contact with us and ask for help, The grammo team will help to them. The grammo team will also sell the gardening plants to the customer in a reasonable price.

Motivation behind developing this app is decreasing syndicate problem in our society and ensure that the farmer can get their actual value of his product. And make our economy stronger.

The results user review and field tests on the prototype under a range of conditions are then presented and analyzed. Finally, limitations of the project are discussed, and recommendations made for future work.



Chapter:2

Initial Study

2.1 Project proposal

I want to develop a mobile application named Grammo. Which will be buy and sell seasonal fruits and vegetables. Also sell nursery products and delivery. It also takes care about garden and farmer's plant. It will be finished utilizing mobile app upheld dialects like XML, Kotlin, Firebase database, Firebase database & Firebase phone number authentication, Firebase storage. XML For front end and Kotlin for backend sensible activities.

. Requirements

- Create account
- Admin panel create account
- Login
- Gardening services
- Farm services
- Message
- Nursery product
- Buy
- Sell

Admin:

- Login
- Check pending request
- Give approval
- Add category for user
- Check payment history
- Can update category
- Can block user
- Can add user
- Log out

User:

- Register
- Login
- Search
- Apply for garden owner
- Check information
- Update information
- Make payment
- Purchase
- Save invoice
- Feedback
- Share & Rate us
- Sign out

2.2 Background of the project

Bangladesh is an agricultural country. Her economy mainly depends on agriculture. For our economy the main thing role is the farmer. After independence the farmer took an important role for our country. But now a day the situation of the farmers is too bad. The farmers cannot get the actual value of their crops but in market the farmers products are too costly. The only thing of this situation is third party deal. Shortly we can call it syndicate. And sometimes the farmers face many of the problems during farming. Sometimes they are not able to solve their problem. For this reason, I want to develop this mobile app. It can be decreasing the syndicate and can solve any problems of the farmer about farming. It can also increase the number of tree plantation and can change our climate.

That's why I want to develop this application.



2.3 Problem area

Our economy mainly depends on agriculture sector. The hero of this agriculture sector is the farmer. But the farmer cannot get the proper value of his crop only for third party deal. Because they cannot sell their product directly to the customer. During farming the farmer sometime face many problems and the cannot able to solve their problem. On the other hand, the number of the plants in our country are too low than the safe zone.



Problem pf farming

2.4 Possible solution

This app can be reducing the number of syndicates of our country and can be solve any kind of firming issues in our country which will be helpful for our economic issues. It also helps to increase the number of tree plantation. By increasing the number of trees plantation, it will be taking a great impact for climate change which will be very helpful for us.



The effect of Grammo app

Chapter 3

Literature Review

3.1 Discussion on problem domain based on published articles

In Bangladesh there are some these types of e-commerce site. But by using these apps the farmer cannot sell their products directly. These apps are also one kind of third-party deal. They develop this system only for their own business. The name of these applications is fashol.com, fashol community, shwapno, aarong and so many e-commerce sites. The main work of these applications is to collect the products from the farmer and sell this in the online platform. But they do not have extra feature like directly sell product in online platform, any types of help option of the farmer etc.



Similar apps like grammo

3.2 Discussion on problem solution based on published articles

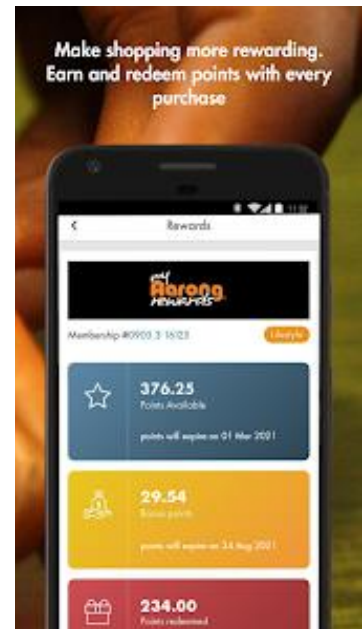
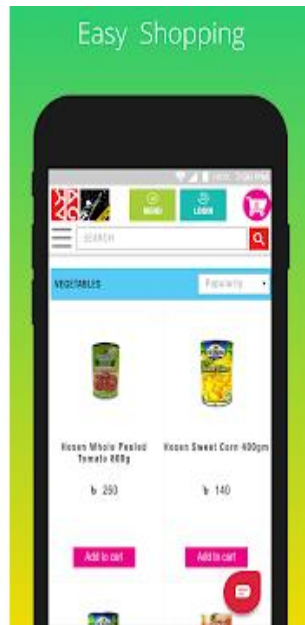
Bangladesh is an agricultural country. Her economy mainly depends on agriculture. But in our country the situation of the farmer's is too bad. That's why I want to develop the system named grammo. This apps can be solving the problems of the farmer. And the customer can also buy fresh product directly from the customer. By This apps both the buyer and seller will be benefited. If any farmer face any problem and he do not able to solve the problem then they can contact with the grammo team. Then grammo team will help to solve the problem. It can also helps to plantation trees more and more.



Similar apps like grammo

3.3 Comparison leading solutions

The farmer can sell their product directly and can take his payment by bkaash and nagad. The farmer can take help from the firming issues. Can be solve the syndicate issues in Bangladesh. And finally, the customer can plant trees more and more.



Problems of similar apps

Best features:

- Farmer can take his payment by online transaction
- The farmer can solve their firming issues very easily
- The syndicate problems can be solved
- Helps to plant trees more

Limitations:

- Can store all the data as long as the user want
- All firming issues can be solved through one app
- Can record all the information renters

Recommended approach:

After survey all the system I personally felt that every apartment management system should have these feature:

Admin panel

- Show all user information
- Approve user
- Add new category
- Survey record

User panel

- Add user information
- Apply for garden owner
- Pay bill
- Choose category
- Check Product availability

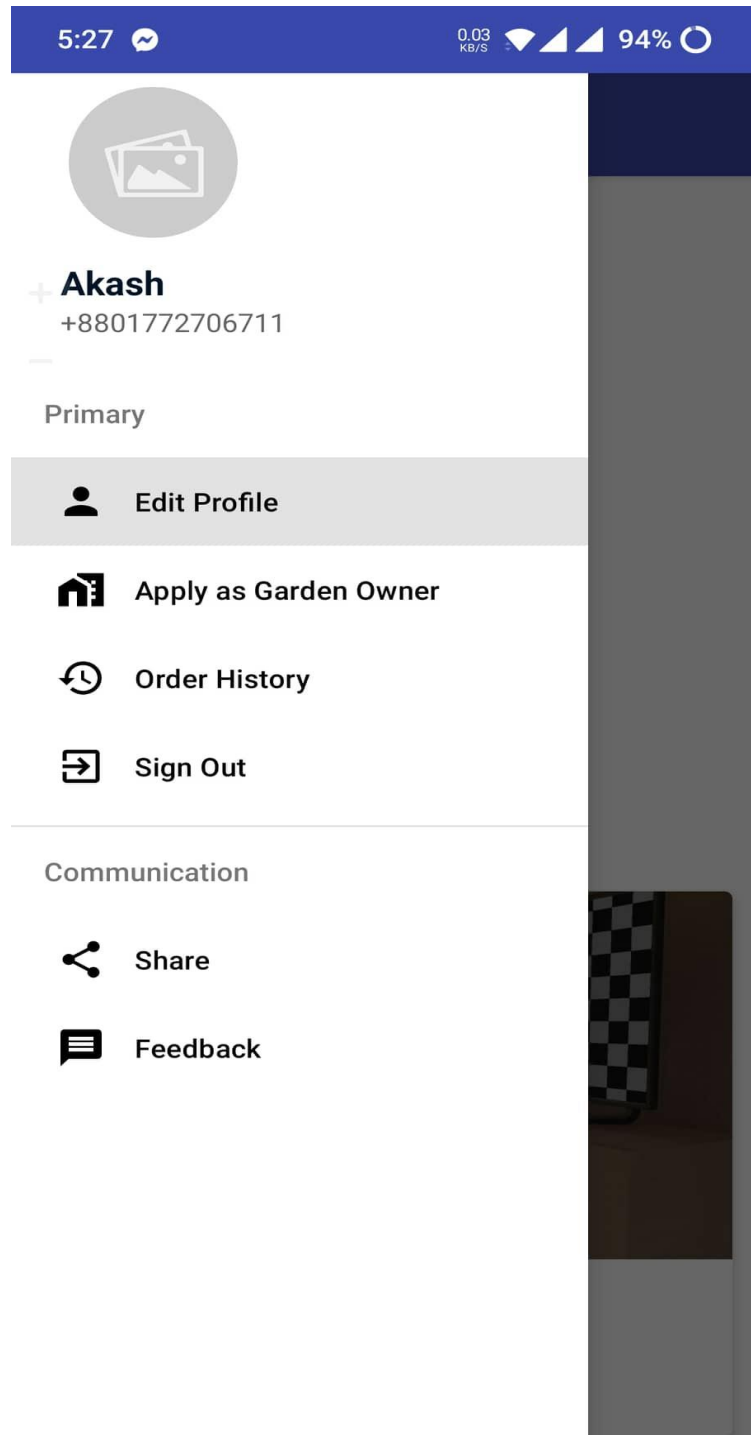


Figure 3.3.1 user panel

Chapter 4

Methodology

4.1 What to use

Methodology is significant piece of any flip of occasions. Setting legitimate and remarkable Methodology is a crucial piece of improvement. The framework must be conceivable with advancement zone. Flexible is a celebrated and convincing framework. In composed there's a framework names Dynamic Systems Development Method (DSDM). This is a way which offers a shape to make and adapt to a structure improvement. It arrived from the variety of Social scientist Guideline. DSDM is an iterative code technique among which each unmarried accentuation proceeds with the 80% boundless that somewhat adequate artworks is believed for every expansion should motivate improvement to the accompanying increase. The extra element is generally completed later as soon as numerous commercial enterprise conditions are stated or adjustments are stated and obliged.

4.2 Why to use

DSDM highlights an extra substantial awareness than maximum choice agile methodologies

in that it manages tasks instead of simply the flip of activities and conveyance of an item

(Frequently programming). The project placing desires attention at the extra substantial

enterprise want and every one components of the best reaction that develops to satisfy

that require. DSDM highlights an extended report of fruitful agile venture conveyance via

and via sorts of business enterprise conditions, and has checked to be definitely versatile,

running efficiently in small sincere organizations, enormous, convoluted

institutions and in a lot directed conditions. It furthermore has been established to be in addition, feasible for each IT and non-IT tasks, for instance enterprise alteration tasks.

I will specific some focuses for which this approach might be nice for this venture

- The enterprise is sure to sense obligation for association because it advances and, in particular, because its adjustments into stay use.
- The threat of constructing a few unacceptable associations is incredibly decreased
- Deployment is sure to move easily, because of the co-hobby of all gatherings involved at some stage in improvement.

These are a few focuses which made DSDM an ideal approach for this venture.

4.3 Sections of methodology

DSDM consistently centers on certain chiefs which conveys an ideal item to the client.

Focus on the business need

This continuously maintains and spotlights at the commercial enterprise needs. This will take a look at the requirements and stipulations of any commercial enterprise first at that factor start building the mission making use of the ones requirements.

Deliver on the schedule

This DSDM gadget constantly ensures that the task is conveyed on time consistently. This makes the cycle in accentuations and the sum total of what accentuation has its time boxing. So my gathering immaculate boxes, transport on time is set.

Collaborate

DSDM generally bunches up with the client. Constantly keeps in contact in each accentuation. In the event that the clients are peppy, degree moves. So planned endeavor is consistently ensured.

Never bargain quality

Quality is the urgent component to keep on satisfying accomplishment. So DSDM guarantees high-acceptable in each cycle. On the off risk that high-agreeable isn't ensured, it's far dispatched lower back and stuck to head ahead.

Develop iteratively

Emphasis is an essential aspect of agile system. It encourages the framework to be built consummately from establishment. Every emphasis finishes totally then paintings actions to the subsequent cycle.

Communicate consistently and plainly

In every cycle correspondence with consumer is accessible. If important consumer correspondence occurs constantly and assured top notch consistently.

4.4 Implementation plans

To execute the DSDM in the undertaking improvement, I need to follow a few stages which will permit me actualize DSDM impeccably.

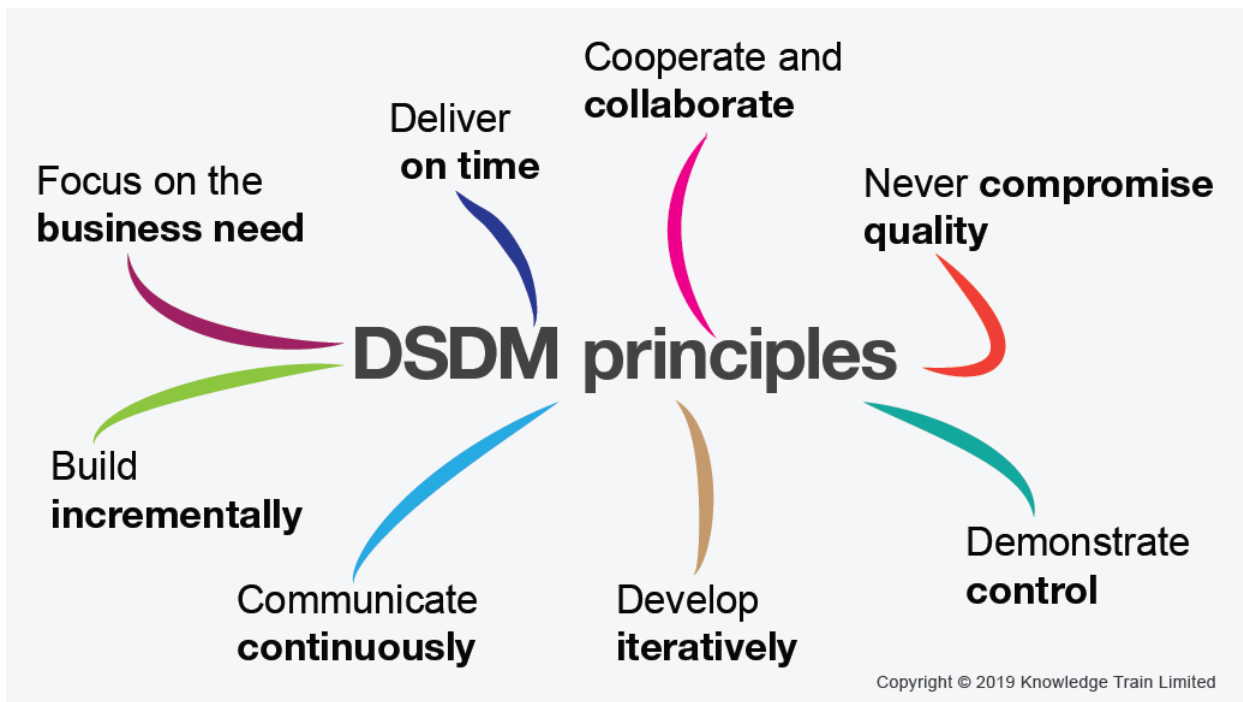


Figure 4.4.1 DSDM

Feasibility Study

Its units up the essential commercial enterprise wishes and boundaries along with the making use of to be deliberate at that factor surveys whether or not or now no longer the making use of is probably a viable contender for the DSDM system

Business Study

Its units up the use and statistics desires in order to allow the making use of to supply business esteem; furthermore, it's the essential utility fashion and distinguishes the practicality desires for the making use of.

Functional mode iteration

It creates a meeting of reformist fashions that display comfort for the customer. The intention at some stage in this unvarying cycle is to build up any wishes through inspiring enter from customers as they exercise the worldview.

Design and build iteration

It gets back to designs purposeful at a phase in steady rendition accentuation to shape positive that anyone has been conscious at a smart stage in a way while heading to change it to supply functional business venture regard for surrender clients. Occasionally, obliging variant cycle and style and create accentuation show up at a comparable time.

Implementation

It puts the latest code increase (an “operationalized” model) into the operational environmental factors. It is acceptable to realize that:

- The expansion may not be 100% complete
- Changes are in like manner referenced in light of the fact that the expansion is presently being set up.

In both the cases, DSDM improvement work proceeds by getting back to the accommodating model cycle action.

CHAPTER 5

Planning

5.1 Project plan

Project organizing suggests all that you are never assisting with increasing you're attempted for achievement. It's the technique you bear to determine the means expected to diagram your undertaking destinations, explain the extent of what must be done and build up the errand rundown to attempt to do it. Undertaking arranging programming framework is generally wont to encourage create intensive task plans

Work Breakdown Structure with timeboxing:

Task	Starting Date	Ending date	Time periods
Starting Project	1/04/2022	10/04/2022	10 days
Project Proposal and Submission	20/04/2022	25/04/2022	5 days

Table 5.1

Gantt Chart:

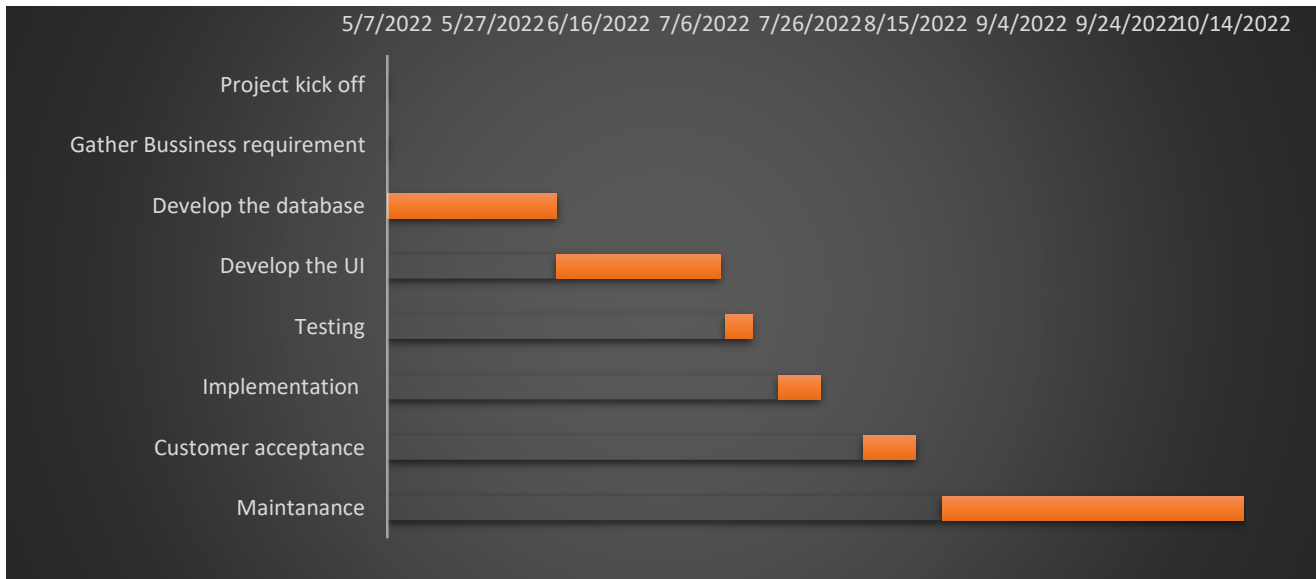


Figure 5.5.1 Gantt Chart

5.2 Test plan

The system should be endeavored in sure techniques so the structure develops to be amazing and contains no bugs. The tests will assist with making the structure more prominent ideal for credible use. Without them programming can likewise furthermore contain a lot of bugs that permits you to cause fundamental issues, all matters considered. The necessary tests are given underneath.

- Unit testing
- Module testing
- Integration testing

Experiment

Try is a lot of conditions or factors under which an analyzer will choose if a system under test satisfies essentials or works successfully.

Client acknowledgement test plan

have plan to make client acknowledgment testing of this framework. I have given my framework to my friend Mojes Tudu Sagor and Sany Abdullah. They install it on his device and check all the functions. Then they finally ensure it that the system work fully. And there was no bug. So, I can take their inputs and gain proficiency with the client acknowledgment of the framework. This will help me construct future advancements too.

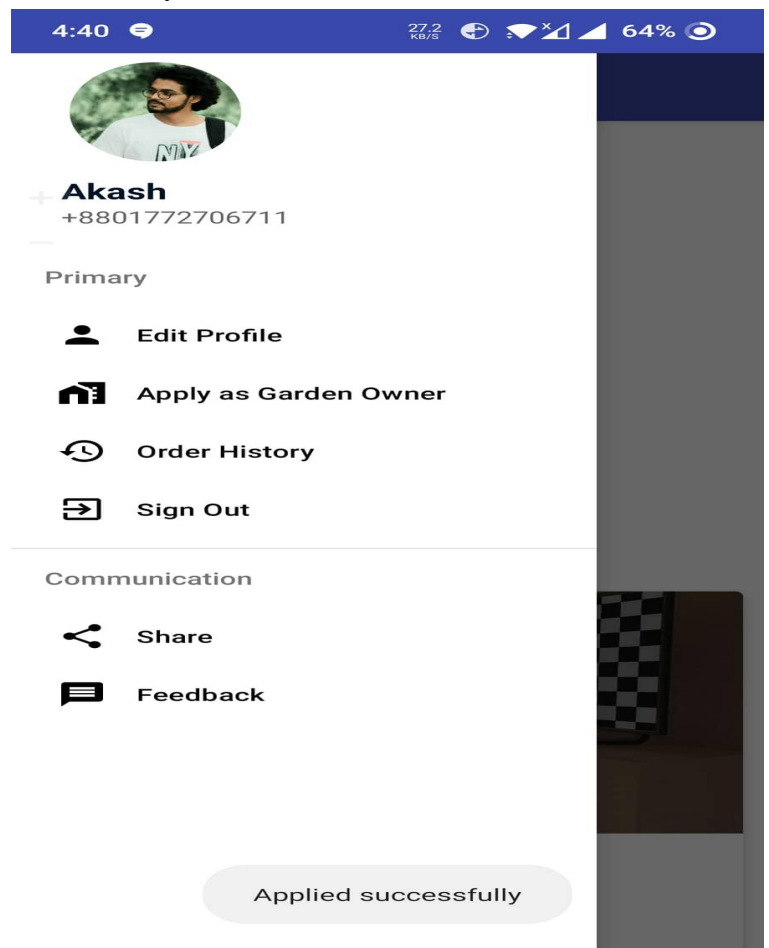


Figure 5.2.1 Client acknowledgement test plan

System Proposal

Project Name	Grammo
Creation date	1/04/2022
Project charter status	
Project sponsor signature	
Proposed project start date	20/04/2022
And end date	10/16/2022
Project Description	Farmer can sell their product and can take help from us about farming.
Project Purpose	Online whole sell of firming product by farmer and take firming help
Project goals	Goals: <ol style="list-style-type: none">1. Sell firming product2. Sell gardening product3. Stop syndicate4. Bright future for farmer5. More tree plantation6. Climate change

Project scope

Features

- Have individual user id and password.
- Take payment from the customer.
- This application support all of the android version mobile phone.
- Payment history will be stored permanently.
- Payment history can be check by user.
- A user can apply for garden owner or as a farmer.

Users

The target market for our product is app-based system.

Proposed Technology

XML and Kotlin

Stakeholder

My self

Competitive advantage

This is the first app and never had anyone makes this kind of app.

Chapter 6

Feasibility testing

6.1 Different types of feasibility

Technical Feasibility

This application is made under the necessities what purchaser needs, so accordingly this application has no specialized blunders and issues which can annoy the shopper. The endeavor should be developed with the ultimate objective that the fundamental limits also, execution is achieved. The endeavor is made inside latest development. Through the development might get obsolete after some time period, due to the way that is everything except challenging to execute the structure to new development. The construction has been made utilizing KOTLIN the endeavor is truly possible for headway. To run the design, the framework should be introduced or set up. By then printer should be connected with the framework for printing reports and receipt. Space and working with need to purchase to encounter the framework.



Kotlin icon

Operational Feasibility

There is a decision has been given in this application named as "how to utilize" will help the client with getting to know this application and to use it. The functional opportunity is the one that will be utilized agreeably after it has been made. Furthermore, the united jail controlling and seeing construction is helpful challenge to be understand and practice the framework with the connection. Additionally, it is everything aside from hard to work after understood and we express this task is functionally plausible.

Financial Feasibility

People can download it from app store and google play store mind out any charge so this application is absolutely free. The design is permitting to resolve several exercises and it changes over the manual framework into mechanize so as indicated by the continuous framework there are immense expenses. Thusly, the proposed structure is helpful than the continuous framework, it has near nothing cost to be executed. Besides, there are assets are as of now accessible, it offers a sprinkle of the framework is monetarily doable for progress.

Managerial Feasibility

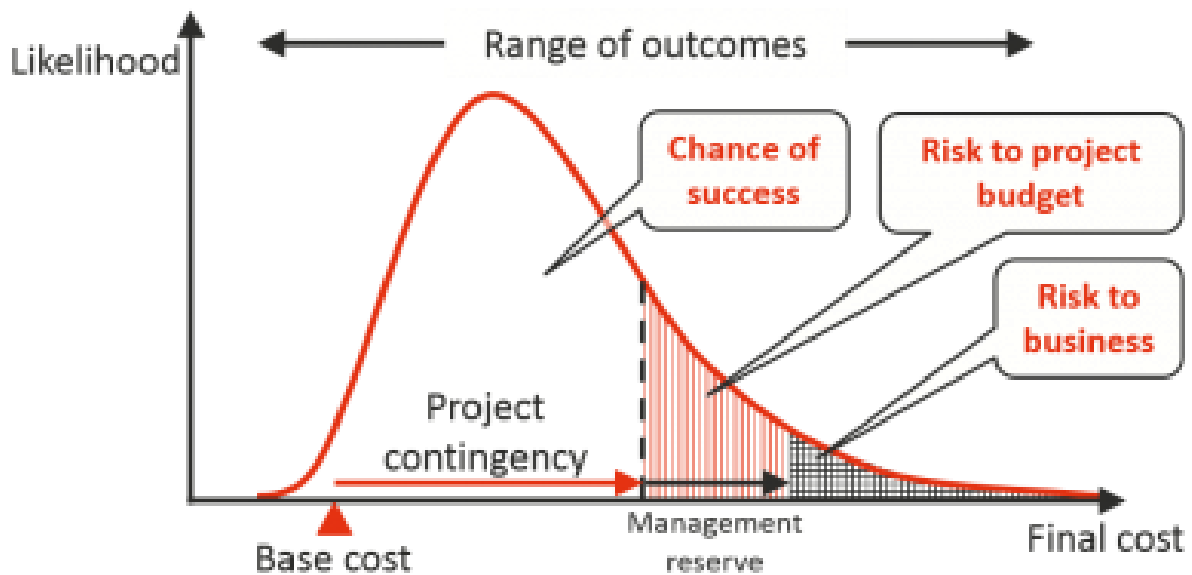
This app will help the consumer to understand the app for first time using. This feature is included specially to help the user to get the app.

Political Feasibility

This app will help to decrease syndicate and helps to climate change.

6.2 Budget and schedule risk

Whenever they've accumulated every one of the information, they can on fundamental subtleties, assessed the risks, and are as sure about your measuring as they can be, they can continue to additionally foster the association monetary arrangement with help practices for all of the perils recognized. For instance, if a detail has a high-risk score, they can ward against likely fallout. You can in like manner have substitute strategies set up a long ways in front of a bet event occurring. You may to be sure, really close, considering the bet assessment that the monetary arrangement ought to be accustomed to address the bet reward compromise you've uncovered. This is particularly critical while looking at subtleties considering low conviction assumptions. By and large, reviewing and keep the risks in arranging will set every master included straight. Months not excessively far off, no one should be set in a position of scrambling to legitimize their assessing after it gone south.



Budget schedule list

Hardware	1 st year	2 nd year	3 rd year
Software	300	800	1000
Server	500	500	3000
Internet	800	700	500
Training	200	900	200
Development	800	1000	400
Total	2600	3900	5100

Table 6.2

6.3 DSDM (Dynamic System Development Method)

DSDM is merchant free, covers the whole lifecycle of an undertaking and gives best work on bearing to on-time, in-spending transport of attempts, with showed versatility to address tasks, considering everything, and for any business area. So DSDM is exceptionally fundamental for this venture.

Chapter 7

Foundation

7.1 Problem area identification

- Interview: It is one of the most widely recognized elicitation procedures. Interviews include seeking clarification on pressing issues, paying attention to the responses, and asking follow-up inquiries. Meetings should be possible one-on-one, however they should likewise be possible in a little social environment in the event that you're mindful so as to get every one of the viewpoints out. [2]
- Observation: The elicitation method perception is a powerful method for understanding how a client takes care of their business by evaluating their workplace. This procedure can be utilized to grasp necessities and give setting to the prerequisite. [3]
- For Survey/Questionnaire, a bunch of inquiries is given to partners to evaluate their contemplations. Subsequent to gathering the reactions from partners, information is dissected to recognize the area of interest of partners. Questions ought to be founded on high need gambles. It would be ideal for questions to be immediate and unambiguous. When the study is prepared, advise the members and remind them to take an interest. [4]

7.2 Rich Picture:

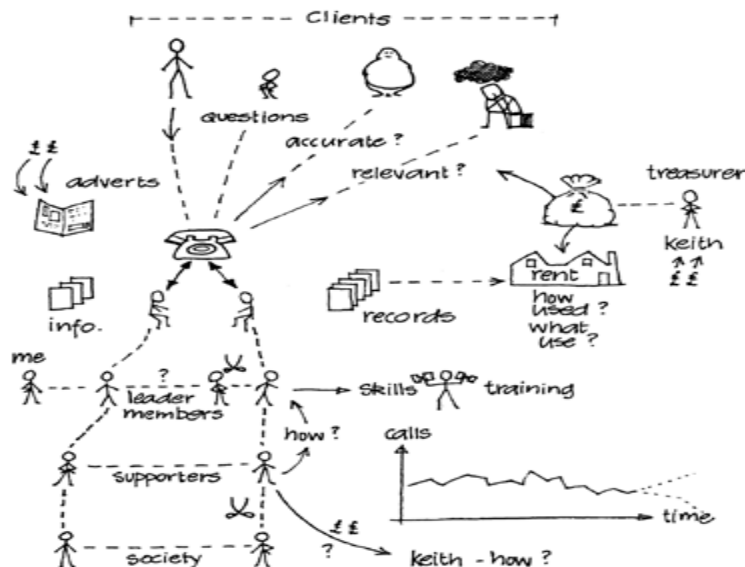


Figure 7.2.1 Rich picture

7.3 Requirements

Functional requirements (admin)

- Create account
- Login to the system
- Approve pending registration
- Add category
- Check history

Functional requirements (user)

- Register account
- Login system
- User ID
- Select category
- Apply for garden owner
- Select category
- Select payment gateway
- Ok to proceed
- Save transaction history

Non-Functional Requirements

- Login system
- Automatically data store
- Print invoice
- User friendly system
- The site ought to have gotten secret phrase validation framework
- The site ought to be completely responsive for any sort of gadgets

7.4 Technology to be implemented

Implementation of front-end design:

As far as android application advancement, the program's front end is really comprised of various screens or exercises. The plan and it are boundlessly unique to work of these exercises. The UIs were made utilizing the Android Studio IDE and the XML plan language. Extensible Markup Language, or XML, is the norm and official plan language for creating Android applications. I needed to utilize Adobe Photoshop and Artist to make the application's logo, symbols, and foundation in different conditions. In situations, I additionally utilized the Android Studio IDE.



Kotlin icon



XML icon

Implementation of database

I have used four different types of databases in this project to increase the efficiency and the speed.

Share preferences: Shared preference is androids interior information stockpiling framework. It is utilized to store limited quantity of information that are expected during the runtime of the application. Little information, for example, login status, username, secret phrase and a few other numeric qualities are put away in this capacity. Information put away in shared preference are simply open to the actual application; client doesn't have direct admittance to any information put away in shared inclination.

Room database: This is disconnected information base administration framework for android to get to telephone memory or SD cards. Information put away in Room data set can be gotten to from the application as well as different applications introduced in the gadget. I utilized Space to store and access information for disconnected use.

Firestore authentication: Firestore is a devoted to android data set administration and capacity framework created and given by Google. As far as putting away and overseeing huge number of clients online Firestore is a lot quicker and simple to carry out than ordinary another validation framework. Our application utilizes Firestore to check clients.

7.5 Recommendation and justification

I used a responsive and user to utilize. I've included a friendly UI to make it easier for an icon and a button so that the user may quickly grasp how our software works. My program has been designed in such a way that it is simple to use. Here I use Kotlin for back-end development, For front end I use Xml, And for the database I use firestore database and for authentication I use firestore mobile number authentication.

Chapter 8

Exploration

8.1 Old use case diagram:

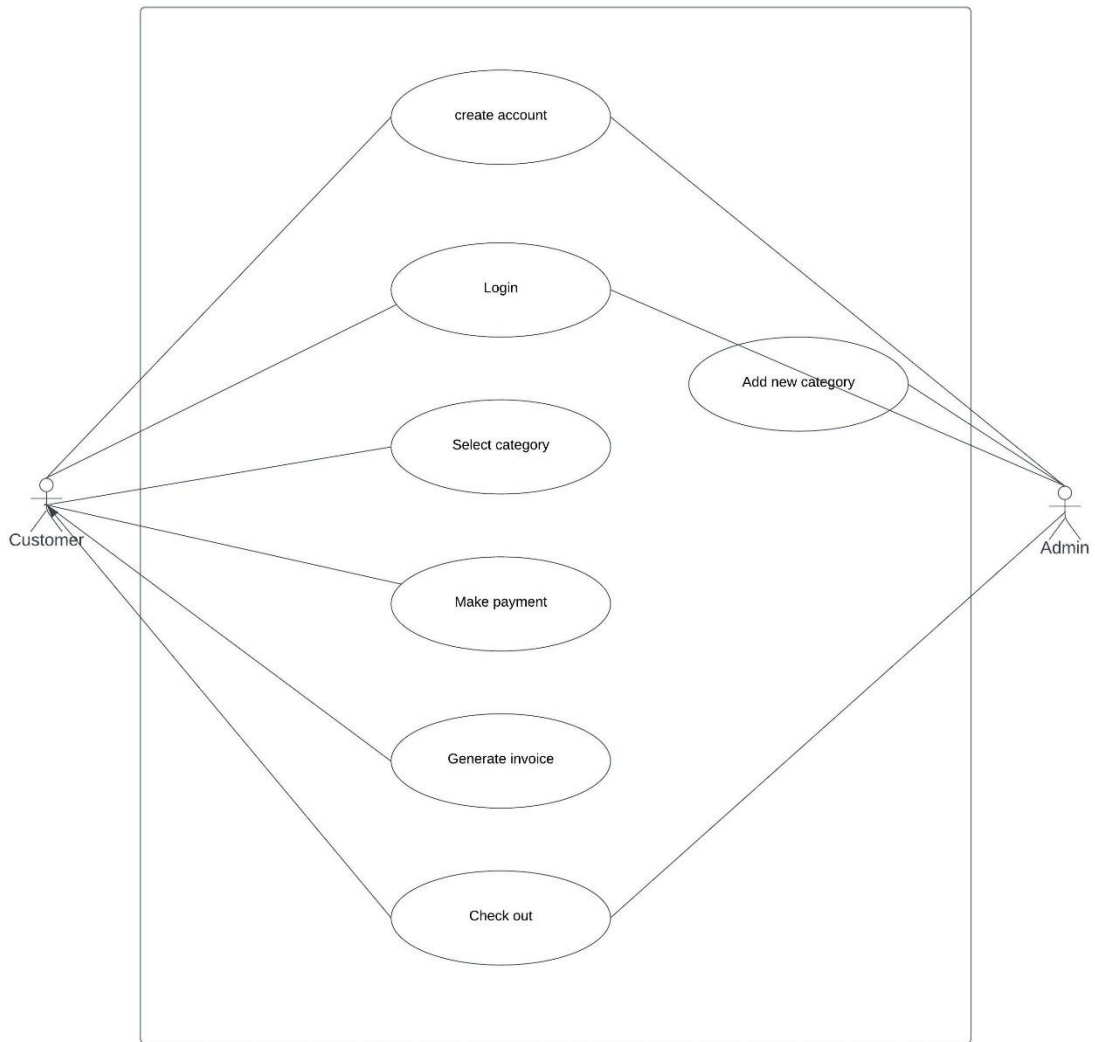


Figure 8.1 Old Use case

8.2 Activity Diagram:

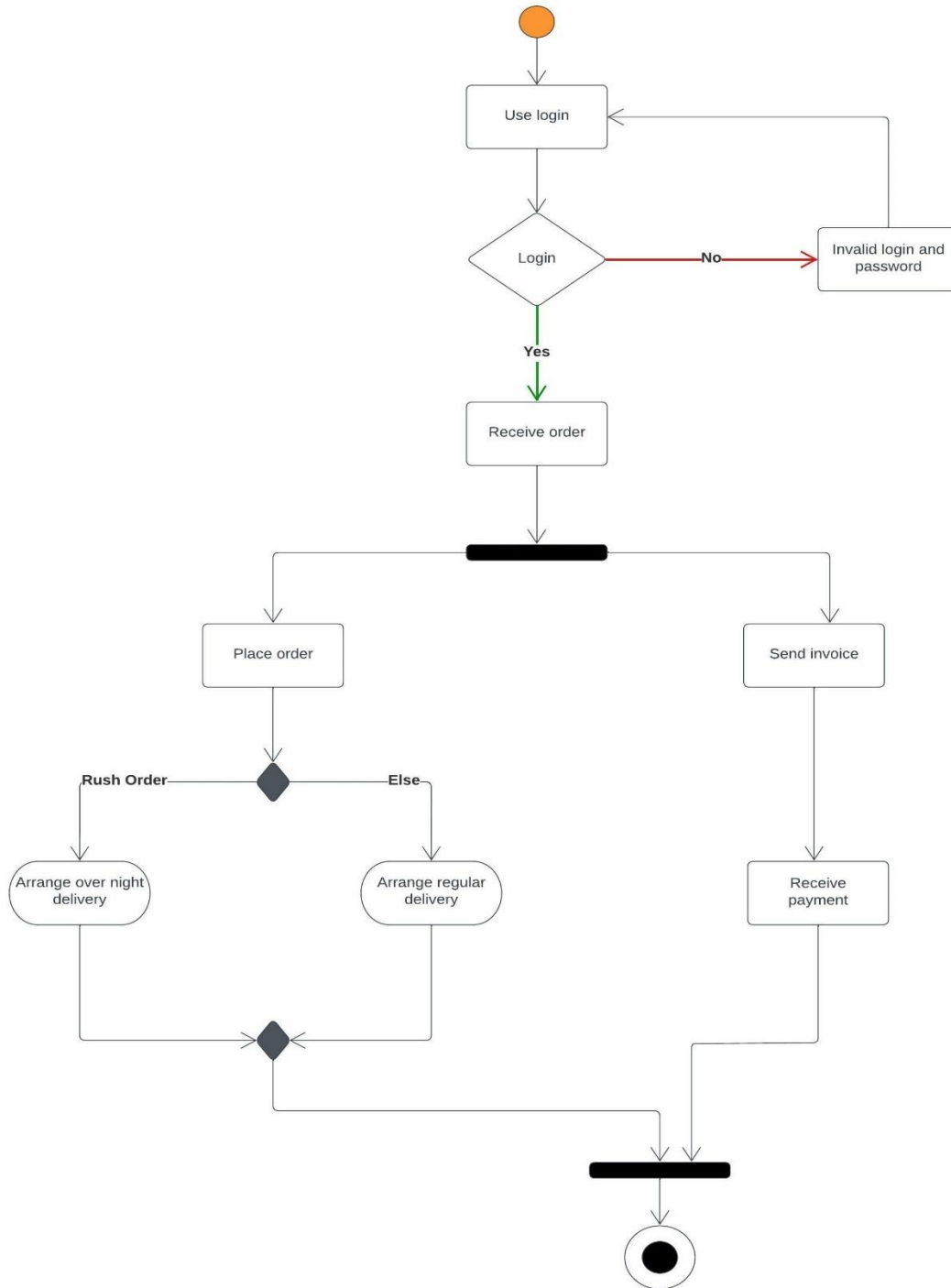


Figure 8.2 Old activity diagram

8.3 Full system use case diagram:

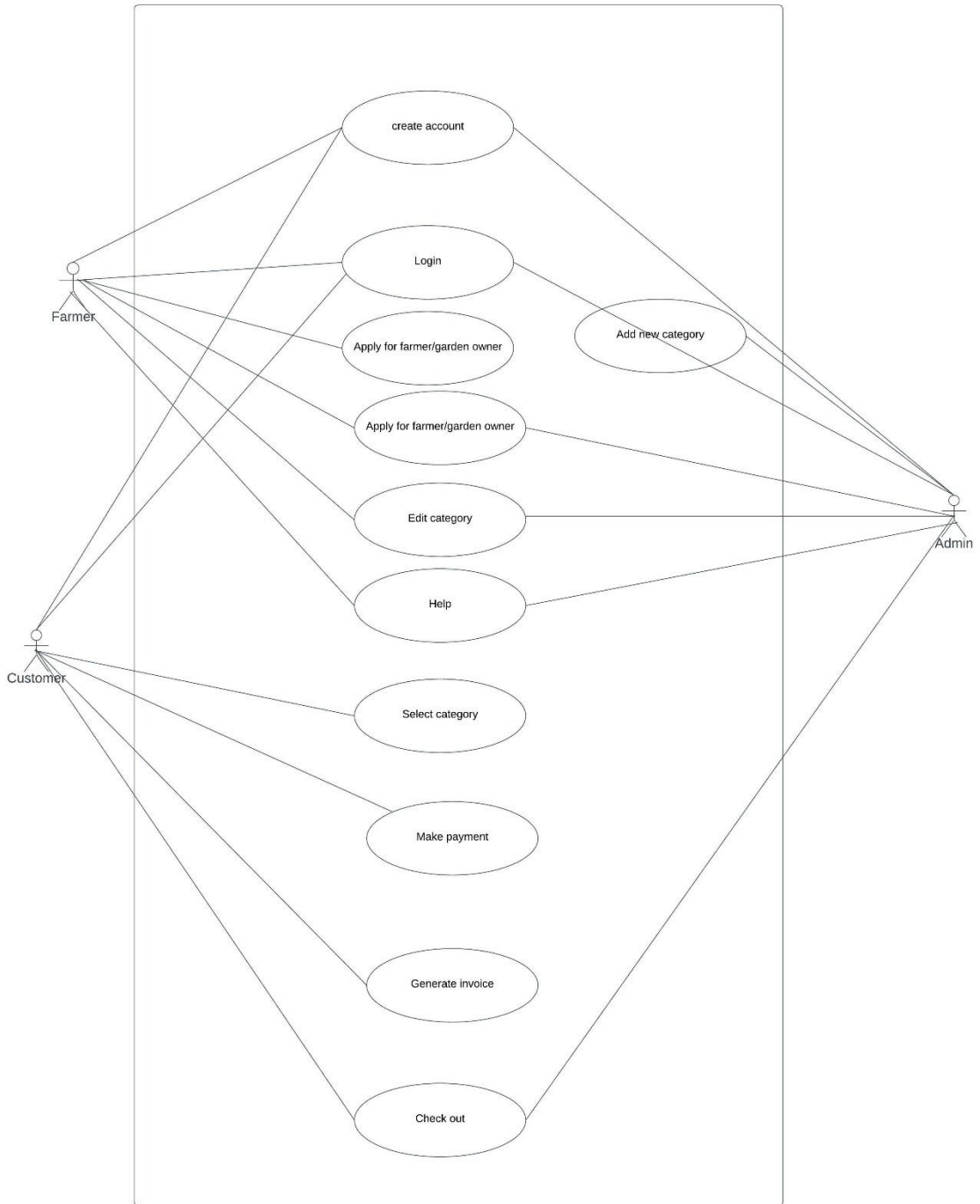


Figure 8.3 New use case

8.4 Full system activity diagram:

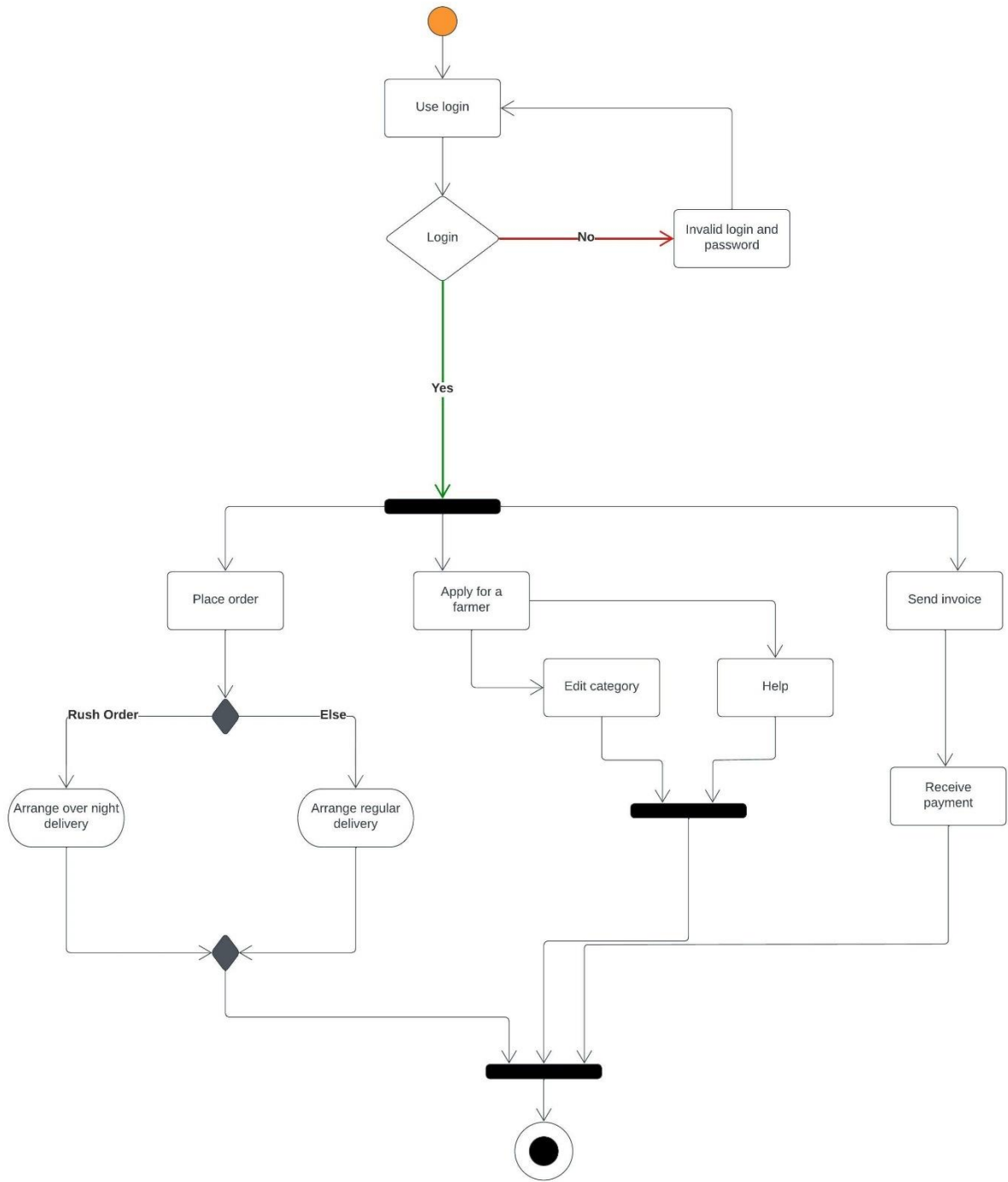


Figure 8.4 Full activity diagram

8.5 Prototype of the new system

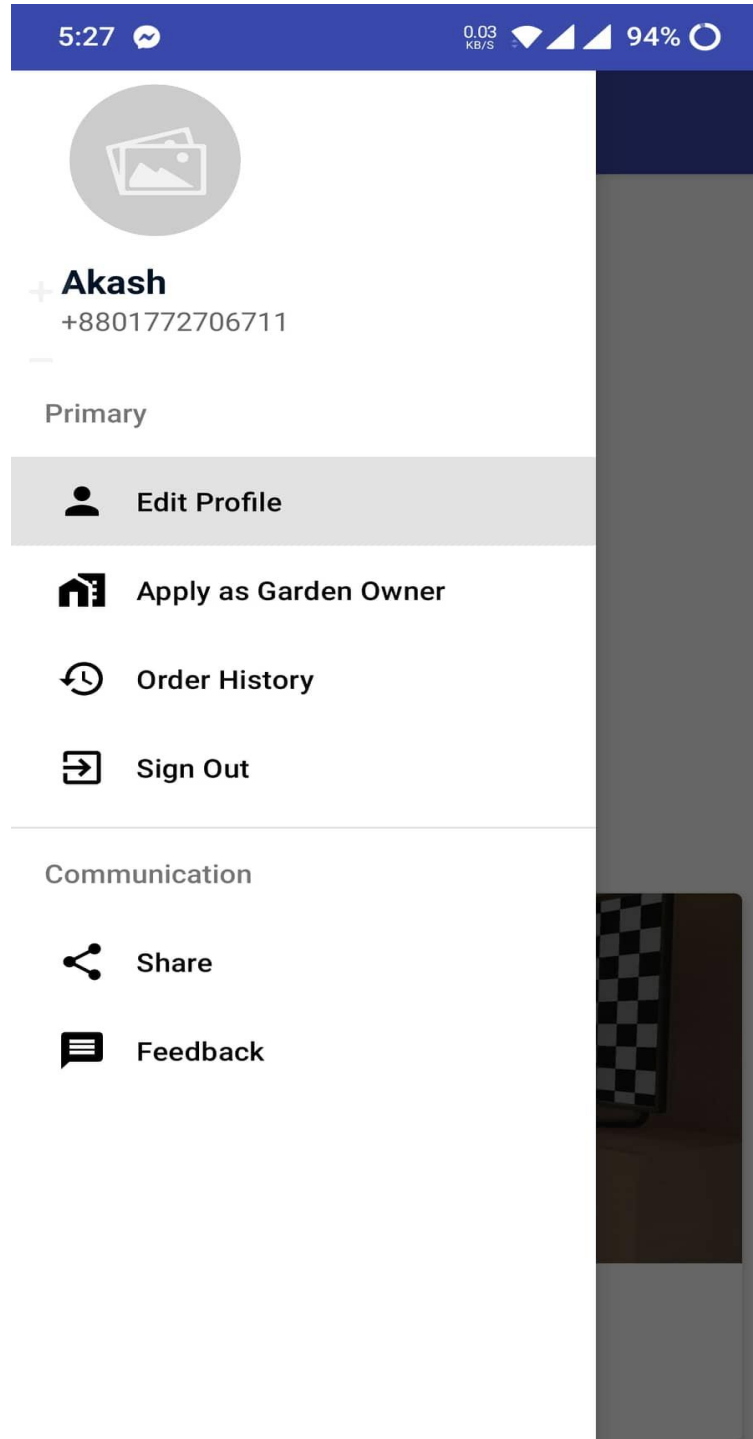


Figure 8.5 Prototype of new system

Chapter 9

Engineering

9.1 New systems module is given below:

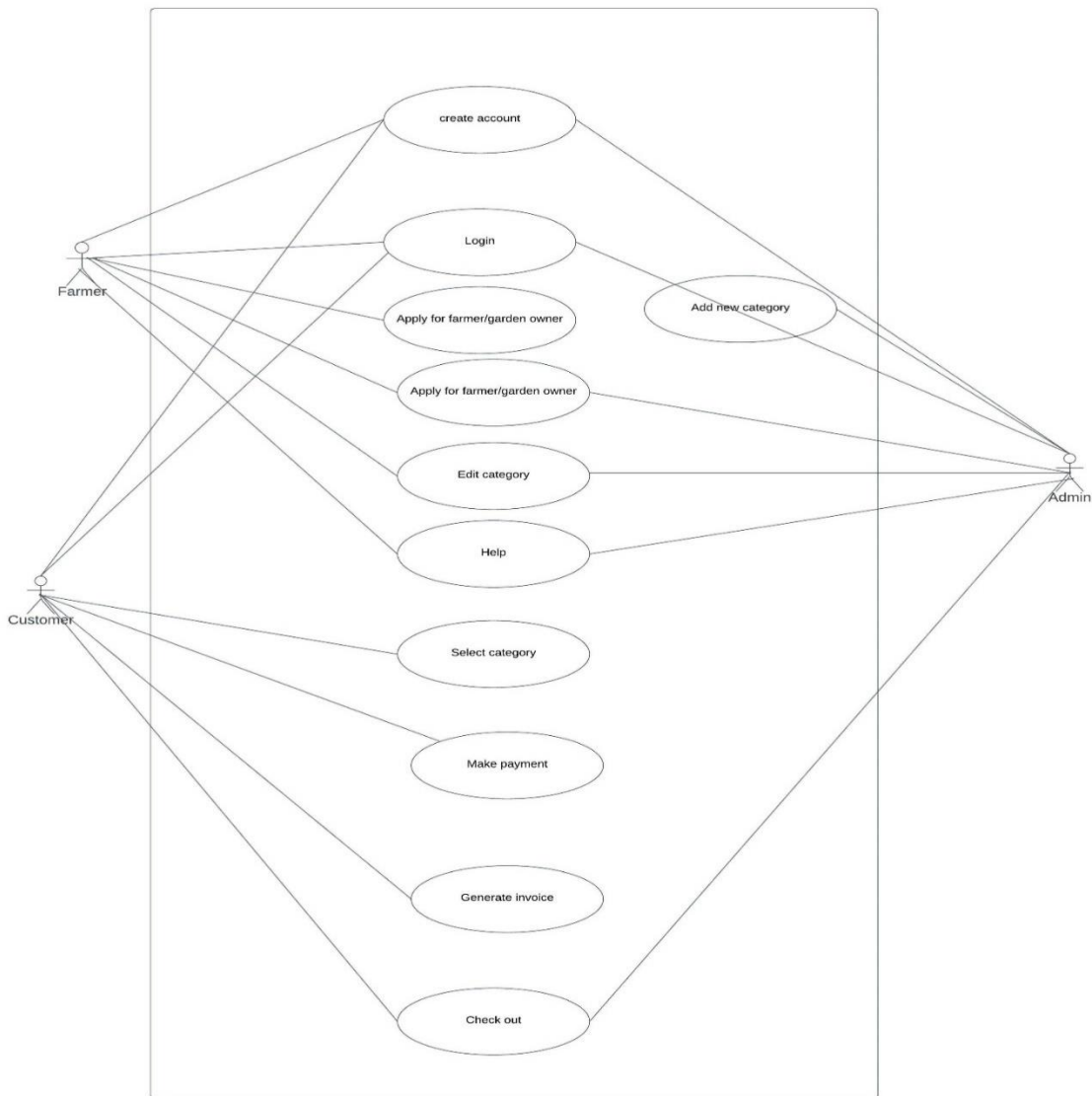


Figure 9.1.1 New system module

Use Case Name: Grammo	ID: 01	Importance	Level:
		High	
Primary Actor: User	Use Case Type: Detail, Essential		
Stakeholders and Interests:			
User – Can be apply for garden owner and edit category.			
Brief Description: This Use case describes how user use the app to help the farmers or garden owner.			
Trigger:			
Type: External			
Relationships:			
Association:	User		
Include:			
Extend:			
Generalization:	sell crops, solve any kind of firming problems		
Normal Flow of Events:			
1. The User can apply their crops and take help from the app.			
Sub flows:			
1. Print invoices.			
2. Check payment history.			

Alternate/Exceptional Flows:

- Sell crops.
- Record firming issues.

9.2 Use case details information:

Admin	<ul style="list-style-type: none">• Log in• Check pending request• Give approval• Add category• Check firming history• Can block user• Log out
User	<ul style="list-style-type: none">• Register• Log in• Search• Add category• Check information• Edit or update information• Delete information• Check product cost

	<ul style="list-style-type: none"> • Pay bill
	<ul style="list-style-type: none"> • Save invoices • logout
system	<ul style="list-style-type: none"> • Save the data • Save invoices

Table 9.1.2 Use case table

9.3 Class Diagram:

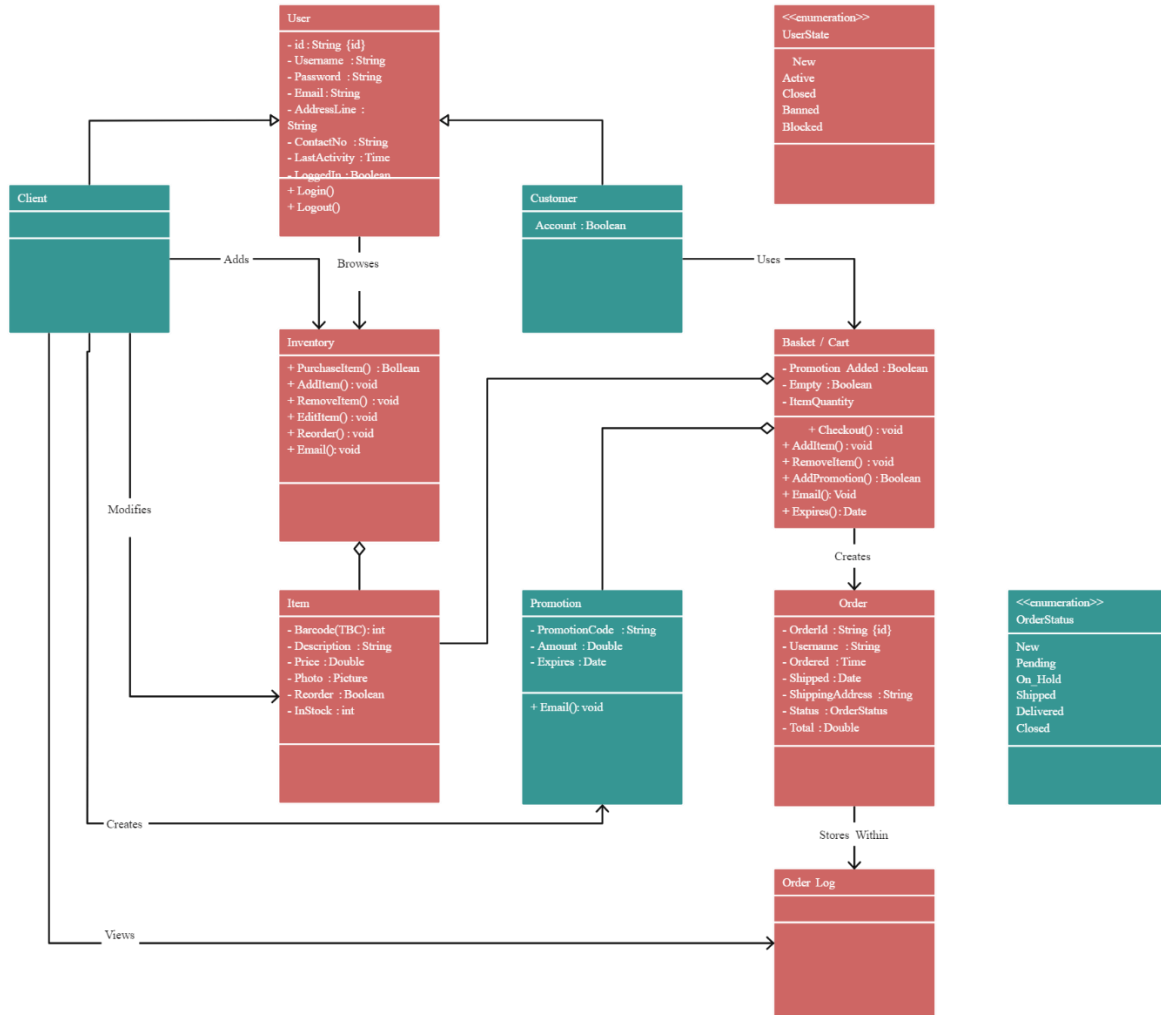


Figure 9.3 Class diagram

9.4 ERD

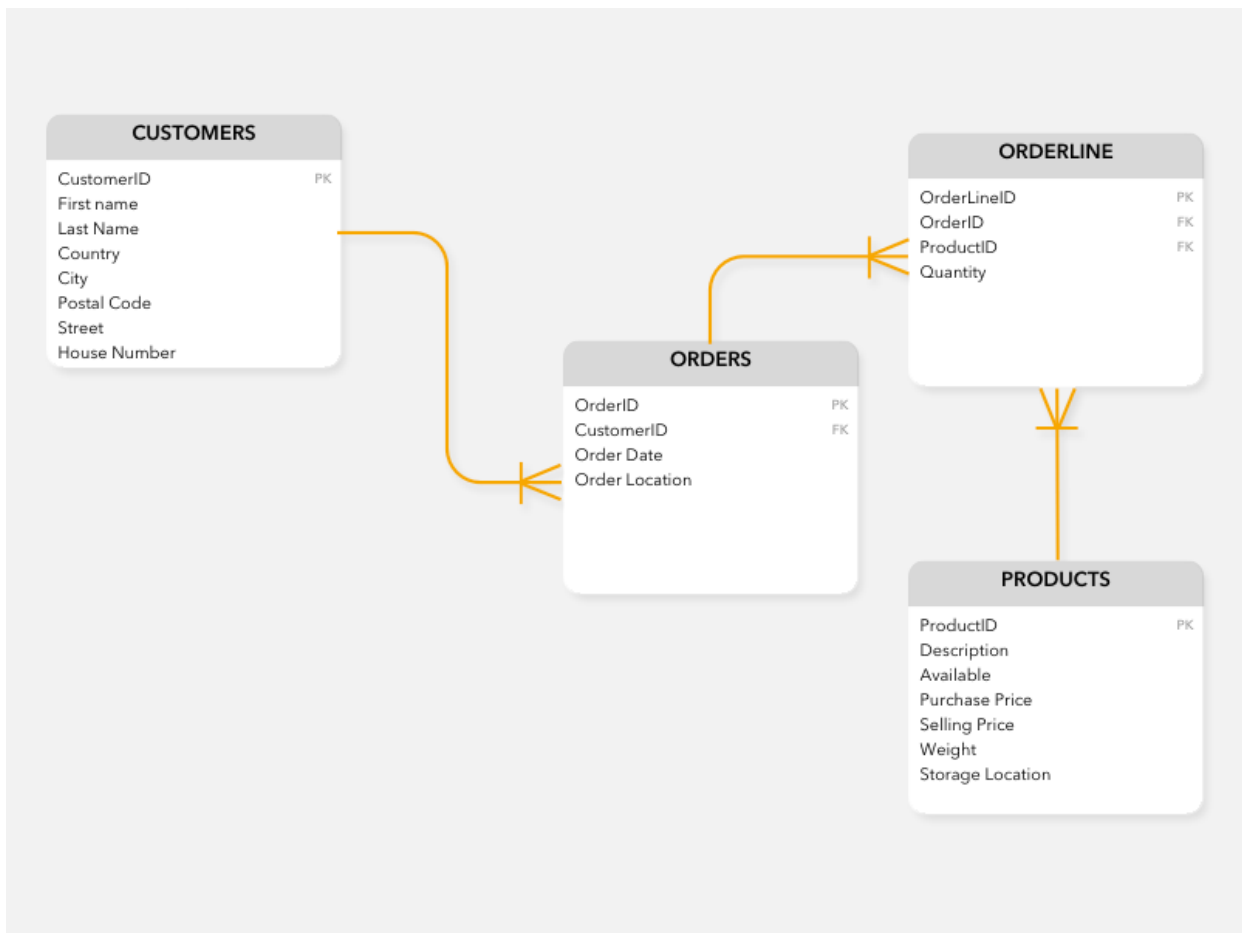


Figure 9.4 ERD

9.5 Sequence Diagram:

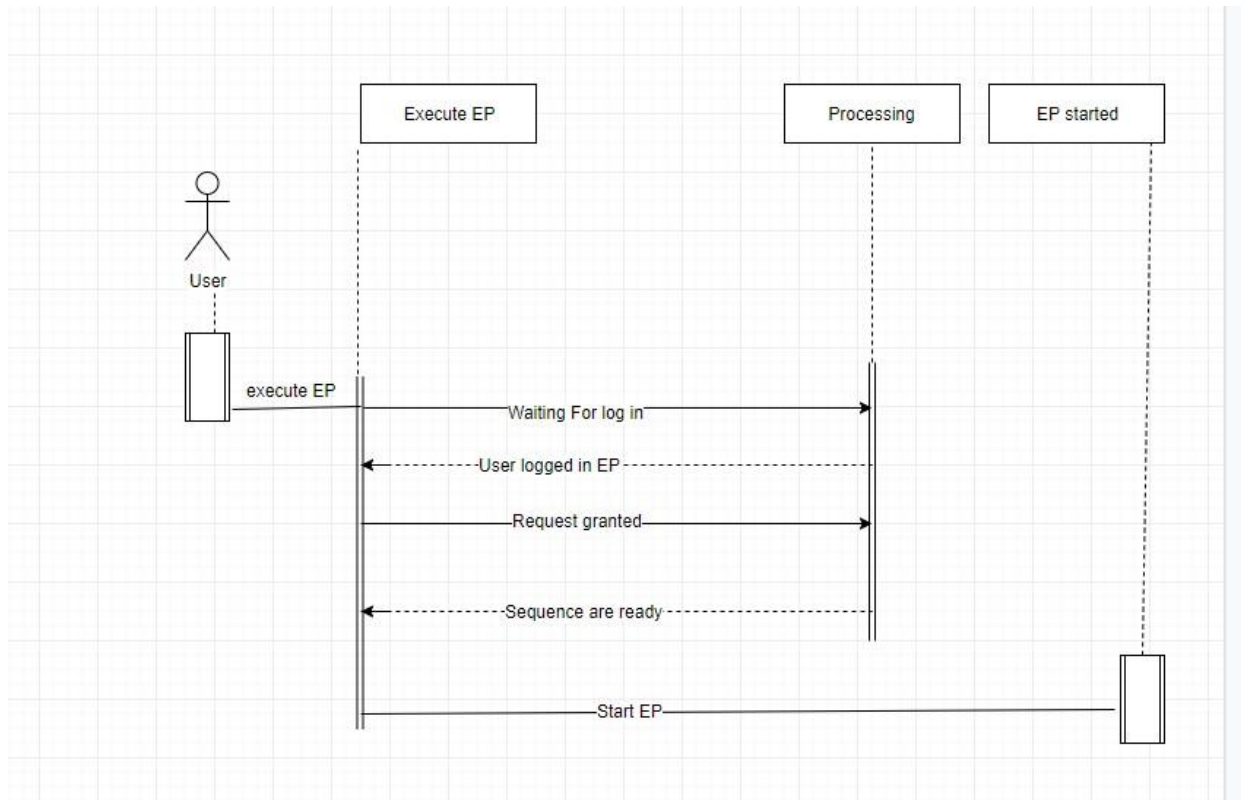


Figure 9.5 Sequence diagram

9.6 Component Diagram:

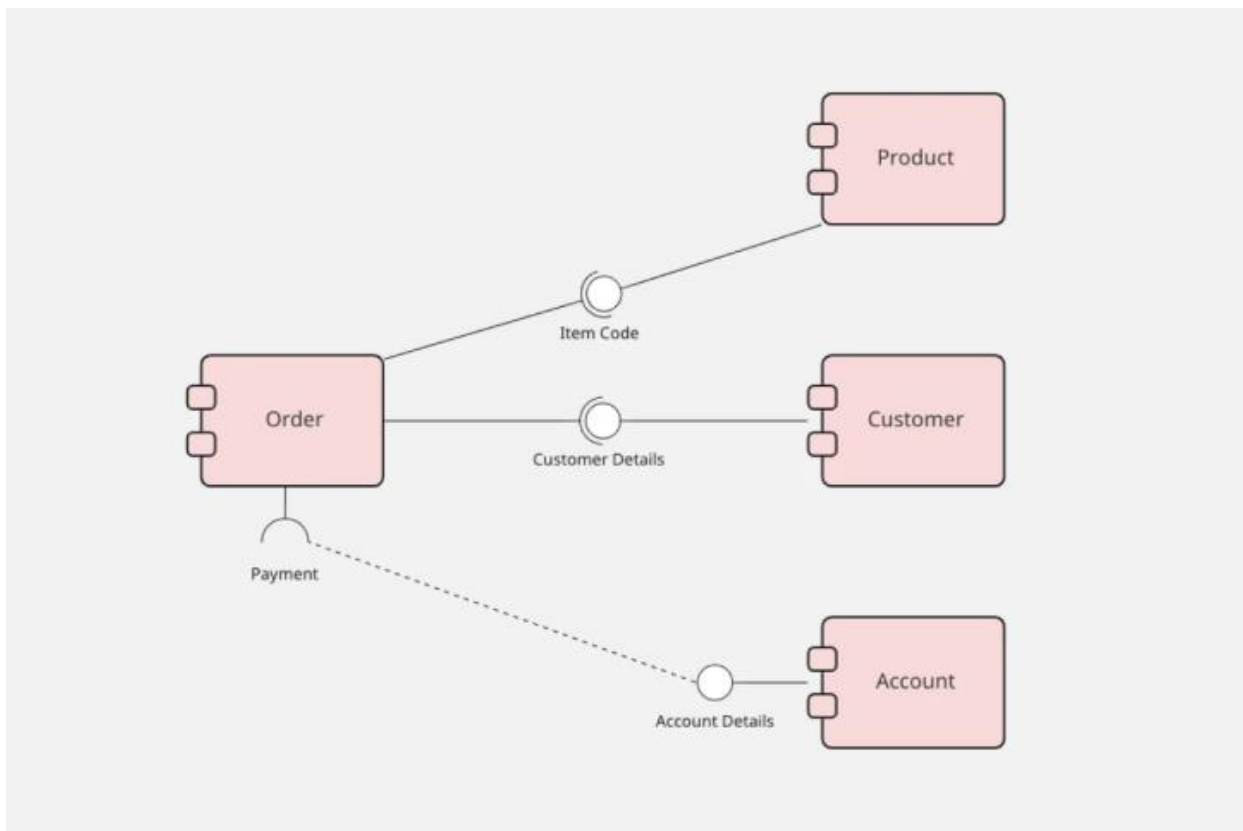


Figure 9.6 Component diagram

9.7 Development diagram:

Development diagram for this system is below here:

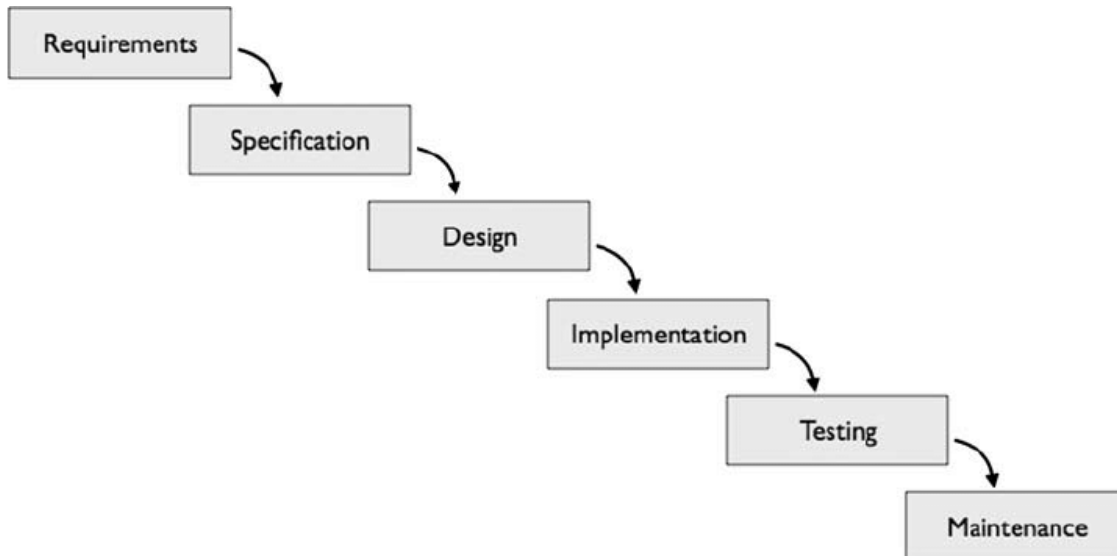
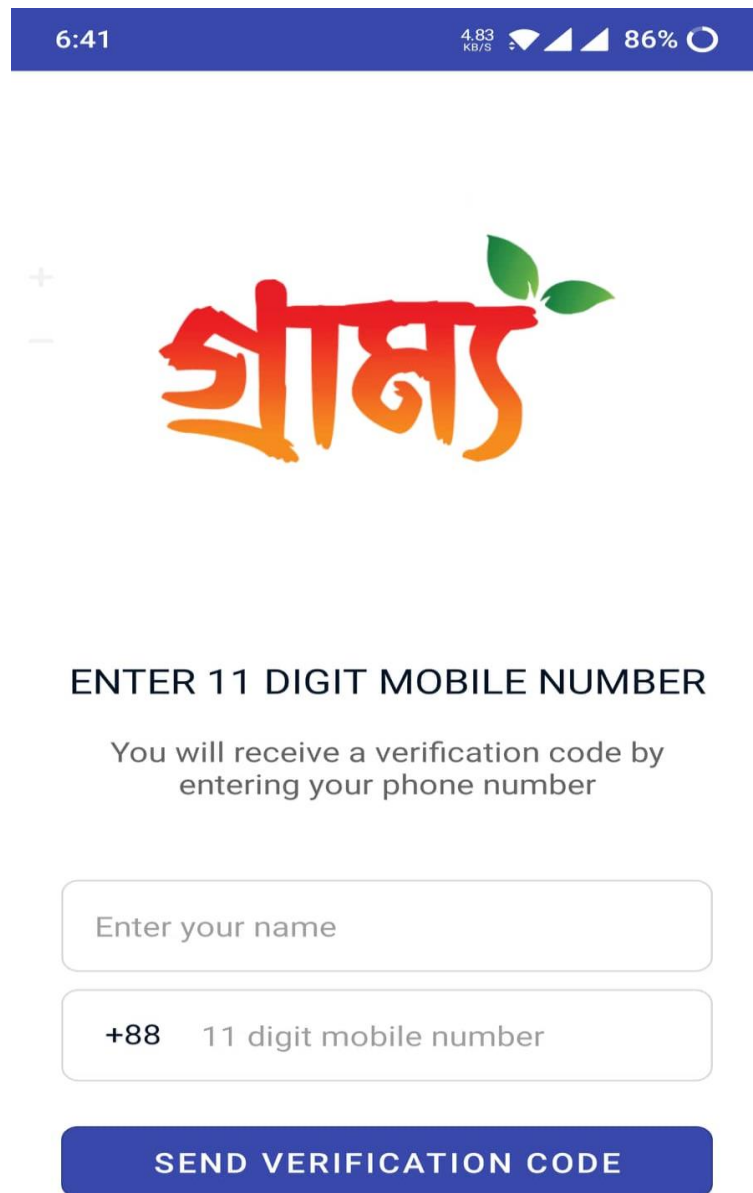


Figure 9.7 Development diagram

9.8 System interface and design:



6:41 4.83 KB/S 86%

शिक्षा

ENTER 11 DIGIT MOBILE NUMBER

You will receive a verification code by entering your phone number

Enter your name

+88 11 digit mobile number

SEND VERIFICATION CODE

Figure 9.8.1 Registration page

6:41

0.76 KB/S    86% 



ENTER VERIFICATION CODE

Enter the 6 digit verification code sent to your phone number

+8801772706711

RESEND CODE

52 SECONDS

VERIFY

Figure 9.8.2 Verify page

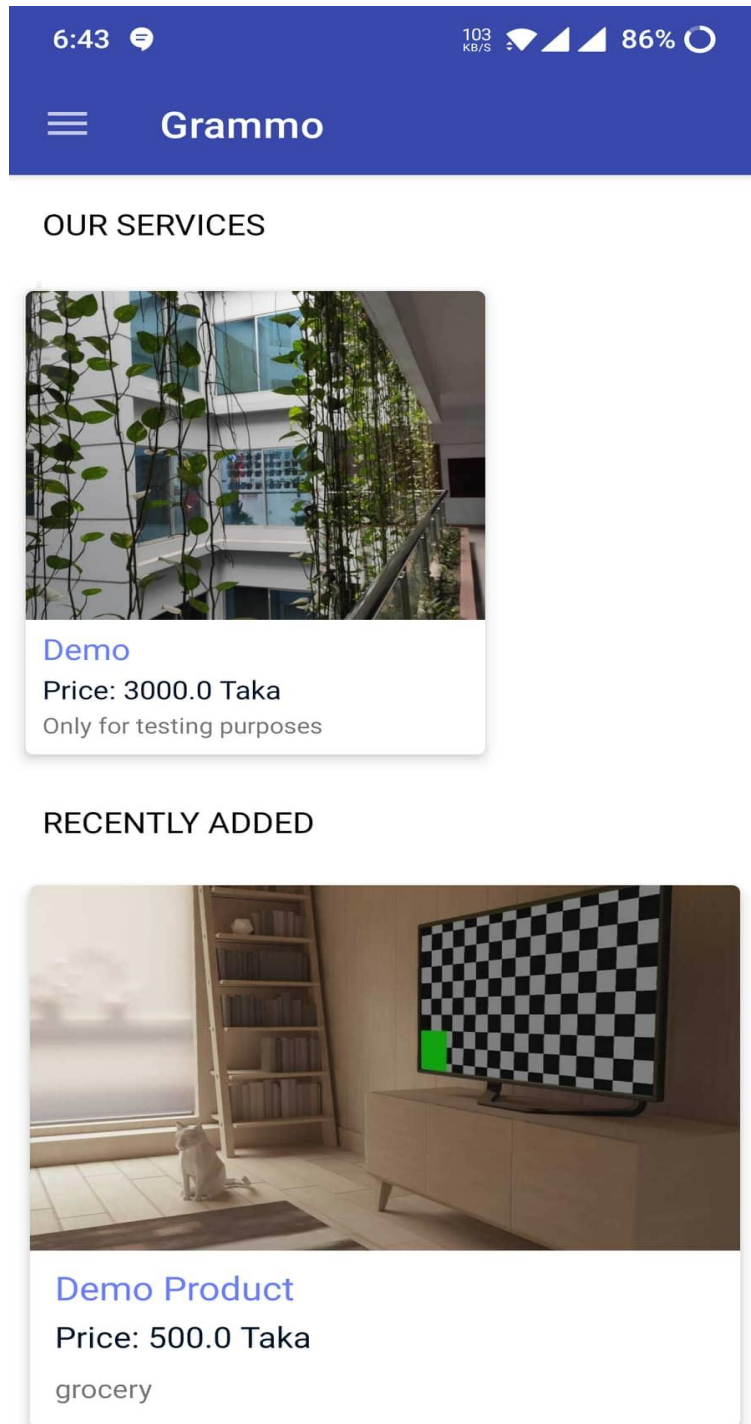
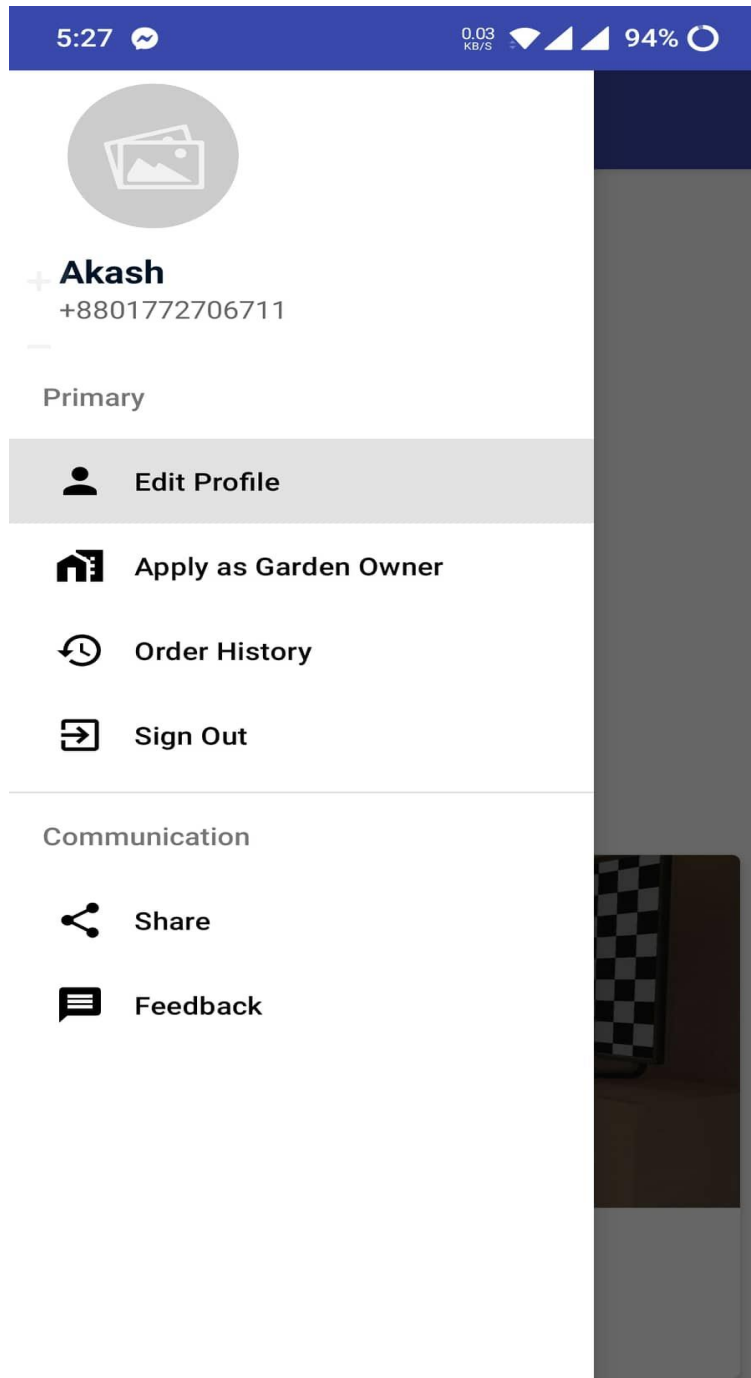


Figure 9.8.3 Home page



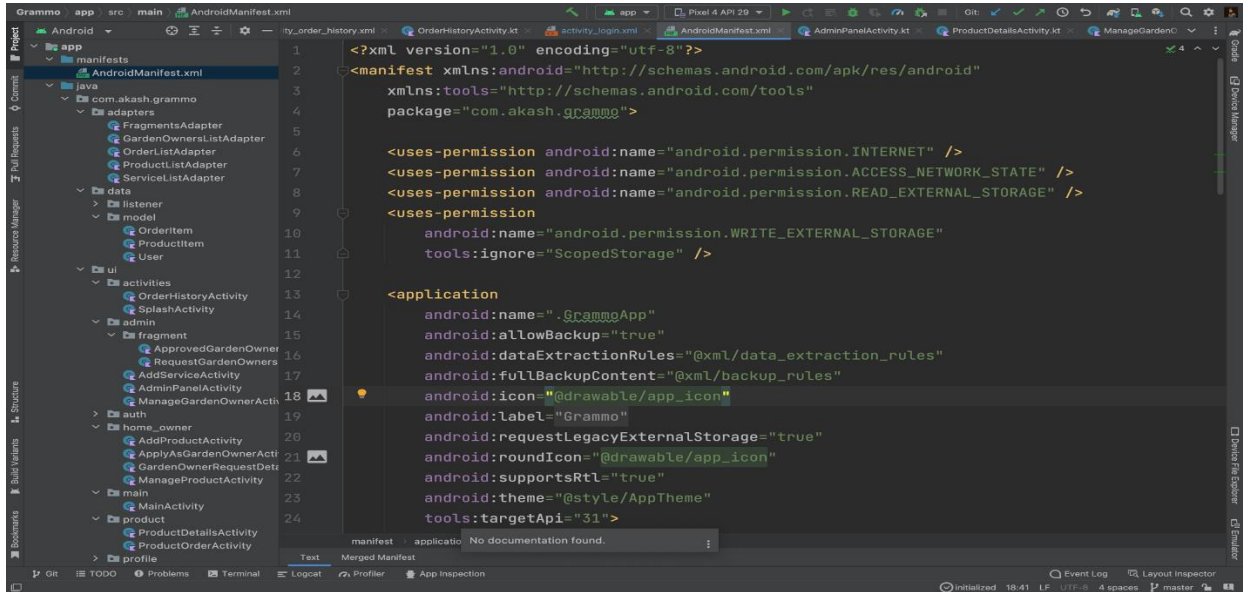
User interface

Chapter 10

Development

10.1 Coding sample

Front end



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.akash.grammo">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE"
        tools:ignore="ScopedStorage" />

    <application
        android:name=".GrammoApp"
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/app_icon"
        android:label="Grammo"
        android:requestLegacyExternalStorage="true"
        android:roundIcon="@drawable/app_icon"
        android:supportRtl="true"
        android:theme="@style/AppTheme"
        tools:targetApi="31">
```

Main function

```

package com.akash.grammo.ui.main

import ...

class MainActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener {

    private lateinit var binding: ActivityMainBinding

    private lateinit var mAuth: FirebaseAuth
    private lateinit var userReference: DatabaseReference
    private lateinit var productReference: DatabaseReference
    private lateinit var serviceReference: DatabaseReference

    private lateinit var tvName: TextView
    private lateinit var tvPhone: TextView
    private lateinit var profileImage: CircleImageView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        //init
        mAuth = FirebaseAuth.getInstance()
        userReference = FirebaseDatabase.getInstance().getReference(path: "Users").child(uid)
    }
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

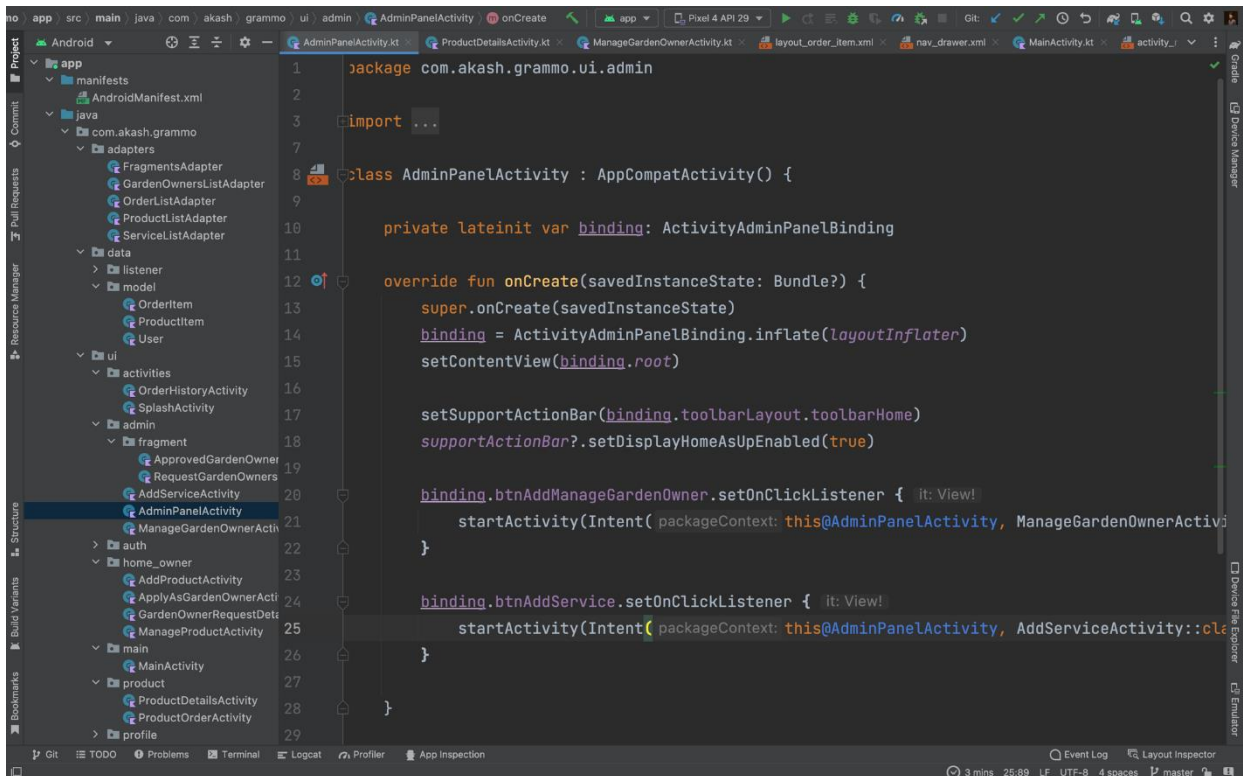
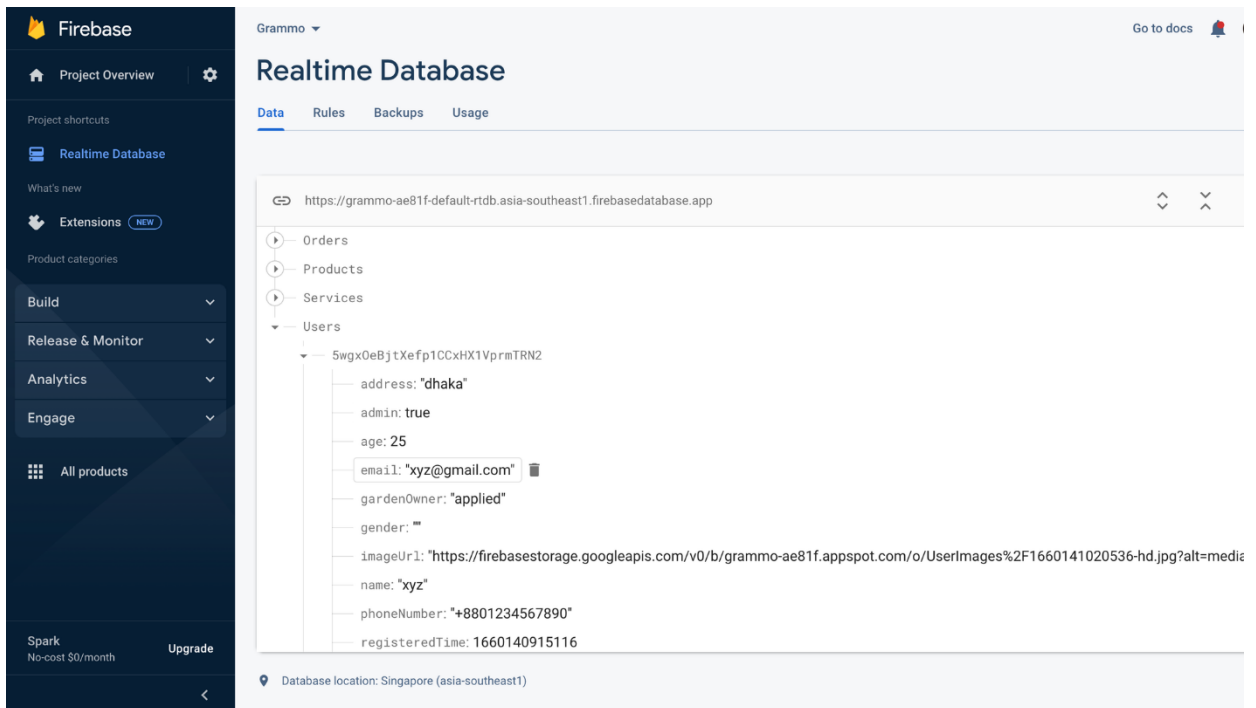
        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar_home"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="@color/colorPrimary"
            app:title="Grammo" />

    </com.google.android.material.appbar.AppBarLayout>

    <include
        android:id="@+id/content"
        layout="@layout/content_home" />

```

Admin panel



Manage garden owner activity


```

package com.akash.grammo.ui.admin

import ...

class ManageGardenOwnerActivity : AppCompatActivity() {

    private lateinit var binding: ActivityManageGardenOwnerBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityManageGardenOwnerBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setSupportActionBar(binding.toolbar)
        supportActionBar?.setDisplayHomeAsUpEnabled(true)

        val adapter = FragmentsAdapter(supportFragmentManager)

        adapter.addFragment(RequestGardenOwnersFragment(), title = "Requests")
        adapter.addFragment(ApprovedGardenOwnersFragment(), title = "Approved")

        binding.viewPager.adapter = adapter
        binding.tabLayout.setupWithViewPager(binding.viewPager)
    }
}

```

Fragments Adapter

```

package com.akash.grammo.adapters

import ...

class FragmentsAdapter(fm: FragmentManager) : FragmentStatePagerAdapter(fm) {
    private val mFragmentManager: FragmentManager = fm
    private val mFragmentManager: MutableList<Fragment> = ArrayList()
    private val mFragmentManager: MutableList<String> = ArrayList()

    override fun getItem(position: Int): Fragment {
        return mFragmentManager[position]
    }

    fun addFragment(fragment: Fragment, title: String) {
        mFragmentManager.add(fragment)
        mFragmentManager.add(title)
    }

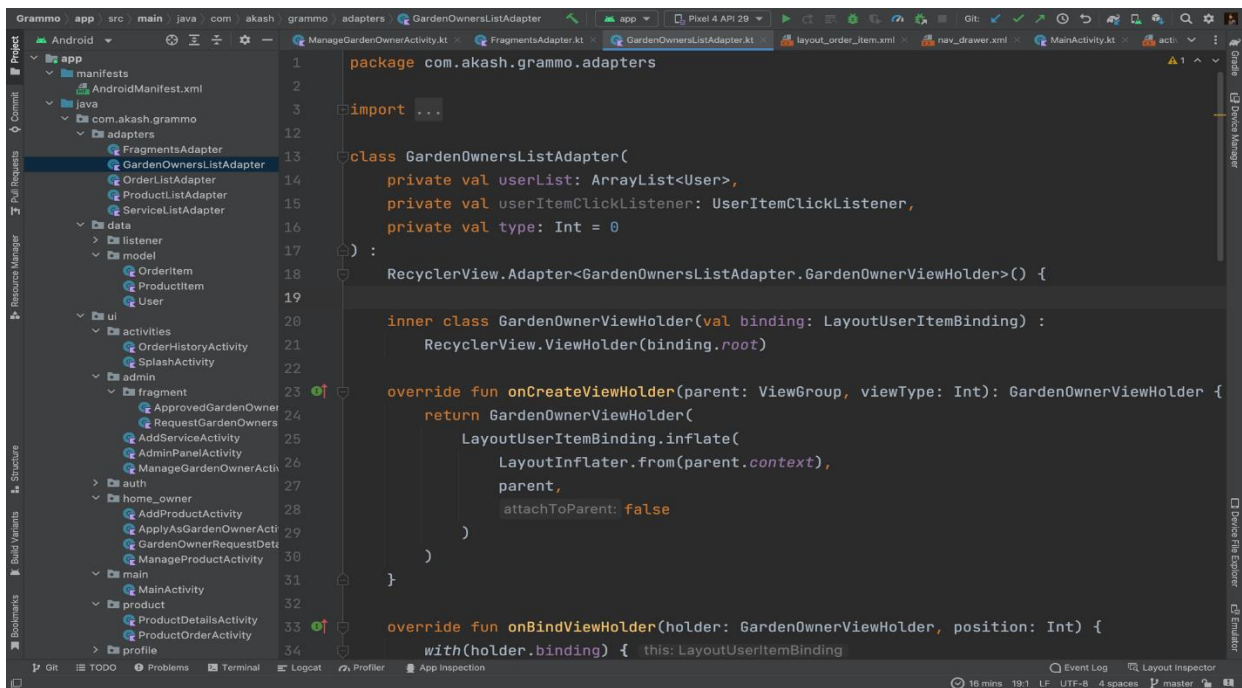
    override fun getPageTitle(position: Int): CharSequence {
        return mFragmentManager[position]
    }

    override fun getCount(): Int {
        return mFragmentManager.size
    }

    fun getFragmentManager(): List<String> {

```

Garden owner list adapter



```
1 package com.akash.grammo.adapters
2
3 import ...
4
5 class GardenOwnersListAdapter(
6     private val userList: ArrayList<User>,
7     private val userItemClickListener: UserItemClickListener,
8     private val type: Int = 0
9 ) :
10     RecyclerView.Adapter<GardenOwnersListAdapter.GardenOwnerViewHolder>() {
11
12     inner class GardenOwnerViewHolder(val binding: LayoutUserItemBinding) :
13         RecyclerView.ViewHolder(binding.root)
14
15     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): GardenOwnerViewHolder {
16         return GardenOwnerViewHolder(
17             LayoutUserItemBinding.inflate(
18                 LayoutInflater.from(parent.context),
19                 parent,
20                 attachToParent: false
21             )
22         )
23     }
24
25     override fun onBindViewHolder(holder: GardenOwnerViewHolder, position: Int) {
26         with(holder.binding) { this: LayoutUserItemBinding
```

Order list adapter

```

1 package com.akash.grammo.adapters
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17 class OrderListAdapter(
18     private val orderList: ArrayList<OrderItem>,
19     private val typeCode: Int = 0
20 ) : RecyclerView.Adapter<OrderListAdapter.OrderViewHolder>() {
21
22     lateinit var progressDialog: CustomProgressDialog
23
24     inner class OrderViewHolder(val binding: LayoutOrderItemBinding) :
25         RecyclerView.ViewHolder(binding.root)
26
27     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): OrderViewHolder {
28         progressDialog = CustomProgressDialog(parent.context)
29         return OrderViewHolder(
30             LayoutOrderItemBinding.inflate(
31                 LayoutInflater.from(parent.context),
32                 parent,
33                 attachToParent: false
34             )
35         )
36     }
37
38     override fun onBindViewHolder(holder: OrderViewHolder, position: Int) {

```

Product list adapter

```

1 package com.akash.grammo.adapters
2
3 import ...
4
5
6
7
8
9
10
11
12 class ProductListAdapter(
13     private val productList: ArrayList<ProductItem>
14 ) : RecyclerView.Adapter<ProductListAdapter.ProductViewHolder>() {
15
16     inner class ProductViewHolder(private val binding: LayoutProductItemBinding) :
17         RecyclerView.ViewHolder(binding.root) {
18         fun setView(productItem: ProductItem) {
19             binding.tvTitle.text = productItem.title
20             val priceText = "Price: \${productItem.price} Taka"
21             binding.tvPrice.text = priceText
22             binding.tvCategory.text = productItem.productType
23             binding.ivProduct.loadImageFromUrl(binding.root.context, productItem.imageUrl)
24         }
25     }
26
27     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ProductViewHolder {
28         return ProductViewHolder(
29             LayoutProductItemBinding.inflate(
30                 LayoutInflater.from(parent.context),
31                 parent,
32                 attachToParent: false
33             )
34         )
35     }

```

Service list adapter

```

package com.akash.grammo.adapters

import ...

class ProductListAdapter(
    private val productList: ArrayList<ProductItem>
) :
    RecyclerView.Adapter<ProductListAdapter.ProductViewHolder>() {

    inner class ProductViewHolder(private val binding: LayoutProductItemBinding) :
        RecyclerView.ViewHolder(binding.root) {
        fun setView(productItem: ProductItem) {
            binding.tvTitle.text = productItem.title
            val priceText = "Price: \${productItem.price} টাকা"
            binding.tvPrice.text = priceText
            binding.tvCategory.text = productItem.productType
            binding.ivProduct.loadImageFromUrl(binding.root.context, productItem.imageUrl)
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ProductViewHolder {
        return ProductViewHolder(
            LayoutProductItemBinding.inflate(
                LayoutInflater.from(parent.context),
                parent,
                attachToParent: false
            )
        )
    }
}

```

Order item

```

package com.akash.grammo.data.model

data class OrderItem(
    var id: String = "",
    val title: String = "",
    val description: String = "",
    val price: String = "",
    val imageUrl: String = "",
    val name: String = "",
    val phoneNumber: String = "",
    val deliveryAddress: String = "",
    val orderedBy: String = "",
    val orderType: String = "",
    val orderPlaceTime: Long = System.currentTimeMillis(),
    val orderStatus: String = "pending",
)

```

Product item

```

1 package com.akash.grammo.data.model
2
3 import java.io.Serializable
4
5 data class ProductItem(
6     val _id: String = "",
7     val title: String = "",
8     val description: String = "",
9     val imageUrl: String = "",
10    val productType: String = "",
11    val price: Float = 0f,
12    val addedBy: String = "",
13    val addedAt: Long = 0L
14) : Serializable

```

Order history activity

```

1 package com.akash.grammo.ui.activities
2
3 import ...
4
5 class OrderHistoryActivity : AppCompatActivity() {
6
7     private lateinit var binding: ActivityOrderHistoryBinding
8
9     private lateinit var orderList: ArrayList<OrderItem>
10    private lateinit var orderReference: DatabaseReference
11
12    override fun onCreate(savedInstanceState: Bundle?) {
13        super.onCreate(savedInstanceState)
14        binding = ActivityOrderHistoryBinding.inflate(layoutInflater)
15        setContentView(binding.root)
16
17        setSupportActionBar(binding.toolbarLayout.toolbarHome)
18        binding.toolbarLayout.toolbarHome.title = "Order History"
19        supportActionBar?.setDisplayHomeAsUpEnabled(true)
20
21        //init
22        orderList = ArrayList()
23        orderReference = FirebaseDatabase.getInstance().getReference(path: "Orders")
24
25        binding.recyclerView.layoutManager = LinearLayoutManager(context: this)
26
27    }
28
29
30
31
32
33
34
35
36

```

Splash activity

```

package com.akash.grammo.ui.activities

import ...

class SplashActivity : AppCompatActivity() {

    private val SPLASH_TIME: Long = 500
    private var mAuth: FirebaseAuth? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_splash)

        mAuth = FirebaseAuth.getInstance()
        val user = FirebaseAuth.getInstance().currentUser

        Handler(Looper.getMainLooper()).postDelayed({
            if (user != null) {
                startActivity(Intent(packageContext: this, MainActivity::class.java))
            } else {
                startActivity(Intent(packageContext: this, LoginActivity::class.java))
            }
            finish()
        }, SPLASH_TIME)
    }
}

```

Approved garden owner

```

package com.akash.grammo.ui.admin.fragment

import ...

class ApprovedGardenOwnersFragment : Fragment(), UserItemClickListener {

    private var _binding: FragmentApprovedGardenOwnersBinding? = null
    private val binding get() = _binding!!

    private lateinit var databaseReference: DatabaseReference
    private lateinit var users: ArrayList<User>

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        _binding = FragmentApprovedGardenOwnersBinding.inflate(inflater, container, attachToPa
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        //init
        databaseReference = FirebaseDatabase.getInstance().getReference(path: "Users")
        users = ArrayList()
    }
}

```

Request garden owner

```

package com.akash.grammo.ui.admin.fragment

import ...

class RequestGardenOwnersFragment : Fragment(), UserItemClickListener {

    private var _binding: FragmentRequestGardenOwnersBinding? = null
    private val binding get() = _binding!!

    private lateinit var databaseReference: DatabaseReference
    private lateinit var users: ArrayList<User>

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        _binding = FragmentRequestGardenOwnersBinding.inflate(inflater, container, attachToParent = false)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        //init
        databaseReference = FirebaseDatabase.getInstance().getReference(path: "Users")
        users = ArrayList()
    }
}

```

Add service activity

```

package com.akash.grammo.ui.admin

import ...

class AddServiceActivity : AppCompatActivity(), AdapterView.OnItemClickListener {

    private lateinit var binding: ActivityAddServiceBinding
    private lateinit var storageReference: StorageReference
    private lateinit var customProgressDialog: CustomProgressDialog

    private lateinit var mReference: DatabaseReference

    private var pickedImageUri: Uri? = null
    private var bitmap: Bitmap? = null
    private var serviceCategory: String = ""
    private var imageUrl: String? = ""

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityAddServiceBinding.inflate(layoutInflater)
        setContentView(binding.root)

        //setup toolbar
        setSupportActionBar(binding.toolbarLayout.toolbarHome)
        binding.toolbarLayout.toolbarHome.title = "Add Product"
    }
}

```

Product details activity

```

package com.akash.grammo.ui.product

import ...

class ProductDetailsActivity : AppCompatActivity() {

    private lateinit var binding: ActivityProductDetailsBinding
    private var issueId: String = ""

    private lateinit var orderReference: DatabaseReference
    private lateinit var customProgressDialog: CustomProgressDialog

    private var productItem = ProductItem()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityProductDetailsBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setSupportActionBar(binding.toolbarLayout.toolbarHome)
        binding.toolbarLayout.toolbarHome.title = "Details"
        supportActionBar?.setDisplayHomeAsUpEnabled(true)

        //init
        orderReference = FirebaseDatabase.getInstance().getReference(path: "Orders")
        customProgressDialog = CustomProgressDialog(context: this)
    }
}

```

Extensions

```

package com.akash.grammo.utils

import ...

fun Context.toast(message: String) = Toast.makeText(context: this, message, Toast.LENGTH_LONG)

fun View.snackBar(message: String) = Snackbar.make(view: this, message, Snackbar.LENGTH_SHORT)

fun checkConnection(context: Context): Boolean {
    val connMgr = context.getSystemService(Context.CONNECTIVITY_SERVICE) as ConnectivityManager
    val activeNetworkInfo = connMgr.activeNetworkInfo
    return activeNetworkInfo != null
}

fun View.hideKeyboard() {
    val imm = context.getSystemService(Context.INPUT_METHOD_SERVICE) as InputMethodManager
    imm.hideSoftInputFromWindow(windowToken, flags: 0)
}

fun ViewPager.onPageSelected(onPageSelected: (Int) -> Unit) {
    addOnPageChangeListener(object : ViewPager.OnPageChangeListener {
        override fun onPageScrollStateChanged(state: Int) {
        }
    })

    override fun onPageScrolled(

```


10.2 Possible problem break down

I have focused a few possible fundamental issues and made breakdown with the objective that I don't have to defy them while building my system. The potential issues could be:

- Data repetition can occur.
- Lack of mistake dealing with
- Requirement investigation need
- Lack of testing
- User acknowledgment lacking

To maintain a strategic distance from these potential issues, I have made a few strides like:

- Created information structure maintaining a strategic distance from information repetition.
- Covered all conceivable mistake taking care of
- Created prerequisite investigation and confirmed from genuine client
- Tested utilizing different techniques
- User acknowledgment study done from genuine clients All these potential issues are addressed.

Chapter 11

Testing

11.1 Acceptance testing

Acknowledgment testing is a test, arranged and prompted affirm that specific essentials are met. Affirmation testing is applied in planning to conclude whether a structure satisfies the predefined affirmation measures. Meeting the affirmation measures is the fundamental step that goes before the improvement of new things, organizations, features, etc. In programming improvement, for instance, test circumstances help to perceive botches, execution issues, or affirm quality in a test environment before the headway can start. [5]

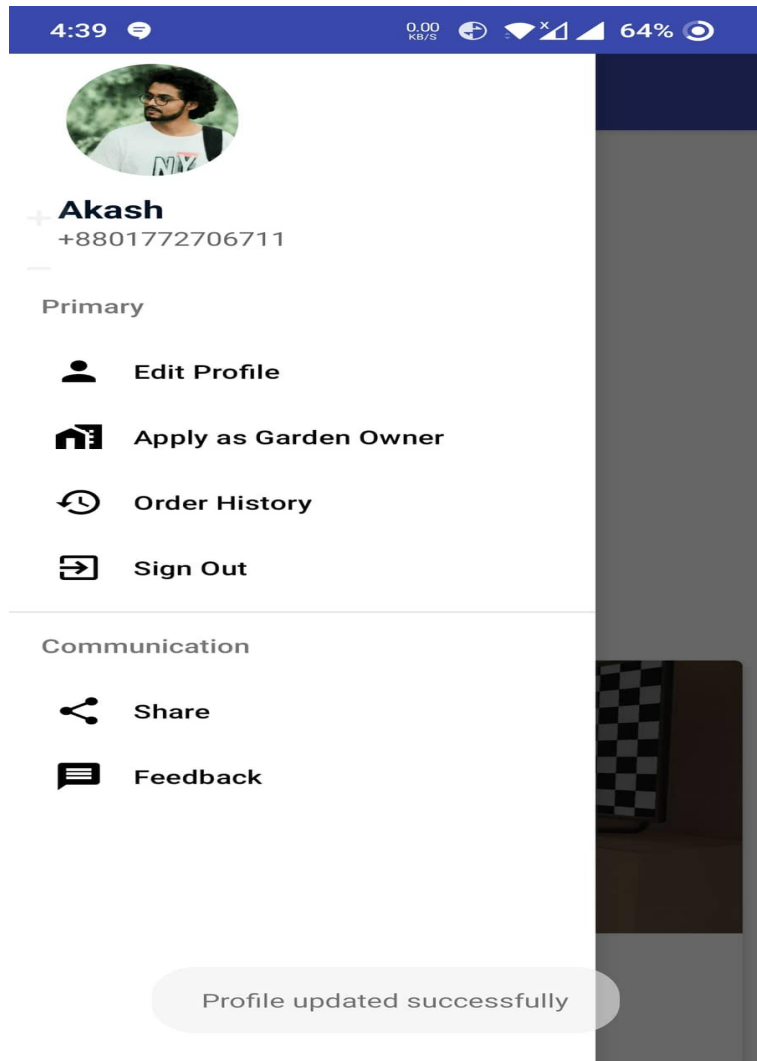


Figure 11.1.1 Profile update

4:36 4.62 KB/S 64%

← Details



Demo
Price: 3000.0
Only for testing purposes

PURCHASE

Figure 11.1.2 Purchase figure

11.2 Test case:


Test Case	Test Input	Expected Outcome	Obtained Outcome	Pass/Fail	Tested On
1.Interface testing	Testing in various android devices and browsers	Perfectly tested in various devices and browsers	App is supported in all devices	Pass	7/10/2022
2. ANR (Application is not responding)	Testing in various devices	Very rare not responding	Problem rate is very rare	Pass	7/10/2022
3. Data Load	Data loading in the application	Data loaded successfully	Loaded successfully	Pass	7/10/2022
4. Re order	Reorder button in the application	Can be reorder easily from the previous order	Re order successfully	Pass	8/10/2022
5.Select category	Selection successfully	List of different categories	List of different categories founded successfully	Pass	9/10/2022
6. Apply for garden owner	Select garden owner panel	For extra feature	Successfully requested for garden owner	Pass	11/10/2022

7. Login	Login by various device	Login Successfully	Login Successfully	Pass	12/10/2022
8. Comment/Query	Comment/query	Comment/query did Successfully	Comment/query did Successfully	Pass	12/10/2022
9. Internet access permission	Connect through application	Connected	Connected	Pass	12/10/2022
10. SDK testing	Checking in 2130 SDK of version	Support in all version	Support in all version	Pass	14/10/2022
11. Database	Database connection	Support in all version	Support in all version	Pass	15/10/2022

Table 11.2.1 Test case

11.3 Unit testing

4:33 1.92 MB/S 65%



ENTER 11 DIGIT MOBILE NUMBER

You will receive a verification code by entering your phone number

Akash

+88 66897657322

SEND VERIFICATION CODE

Invalid Request

Verification test

4:34

1.74 MB/S



65%



ENTER 11 DIGIT MOBILE NUMBER

You will receive a verification code by entering your phone number


Akash

+88 01772706711

The sms verification code used to create the phone auth credential is invalid. Please resend the verification code sms and be sure use the verification code provided by the user.





Verification test

4:39 0.00 KB/S 64%





Akash
+8801772706711

Primary

-  Edit Profile
-  Apply as Garden Owner
-  Order History
-  Sign Out

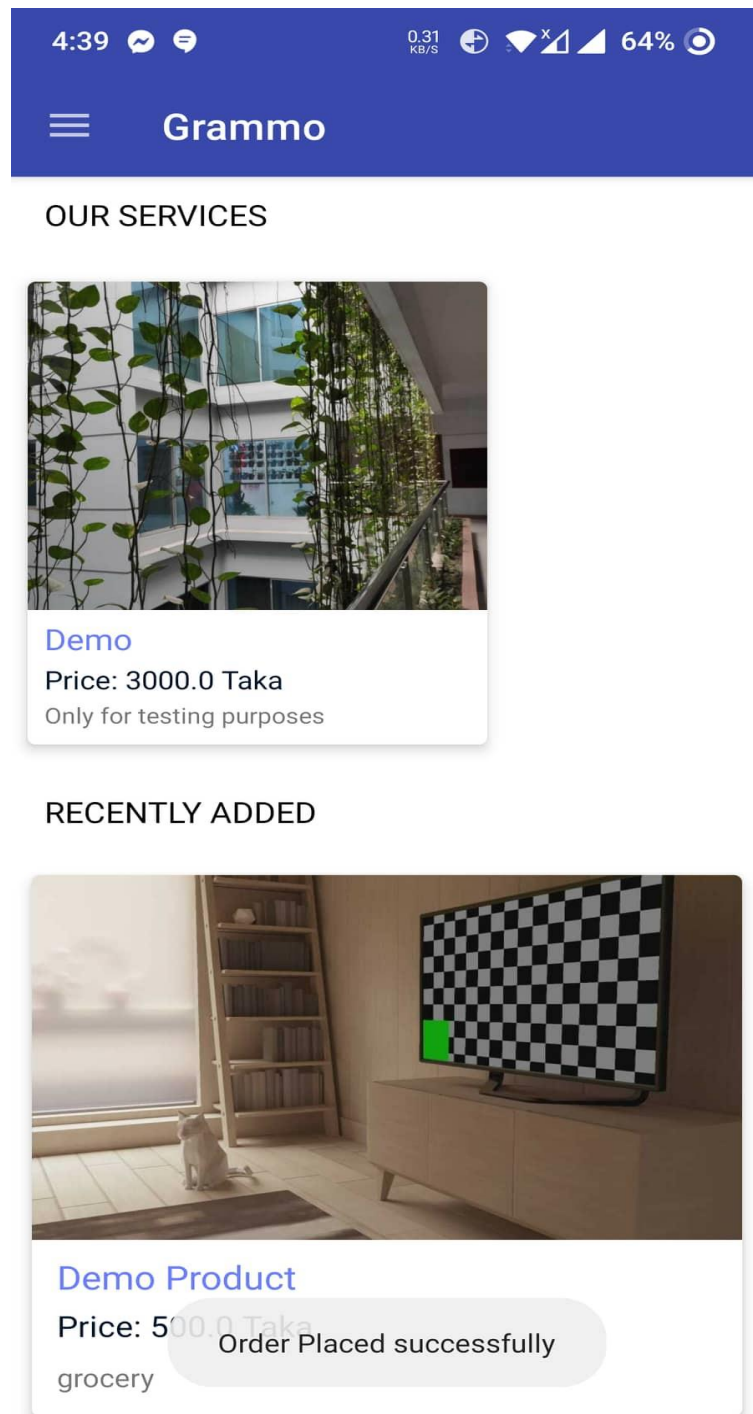
Communication

-  Share
-  Feedback

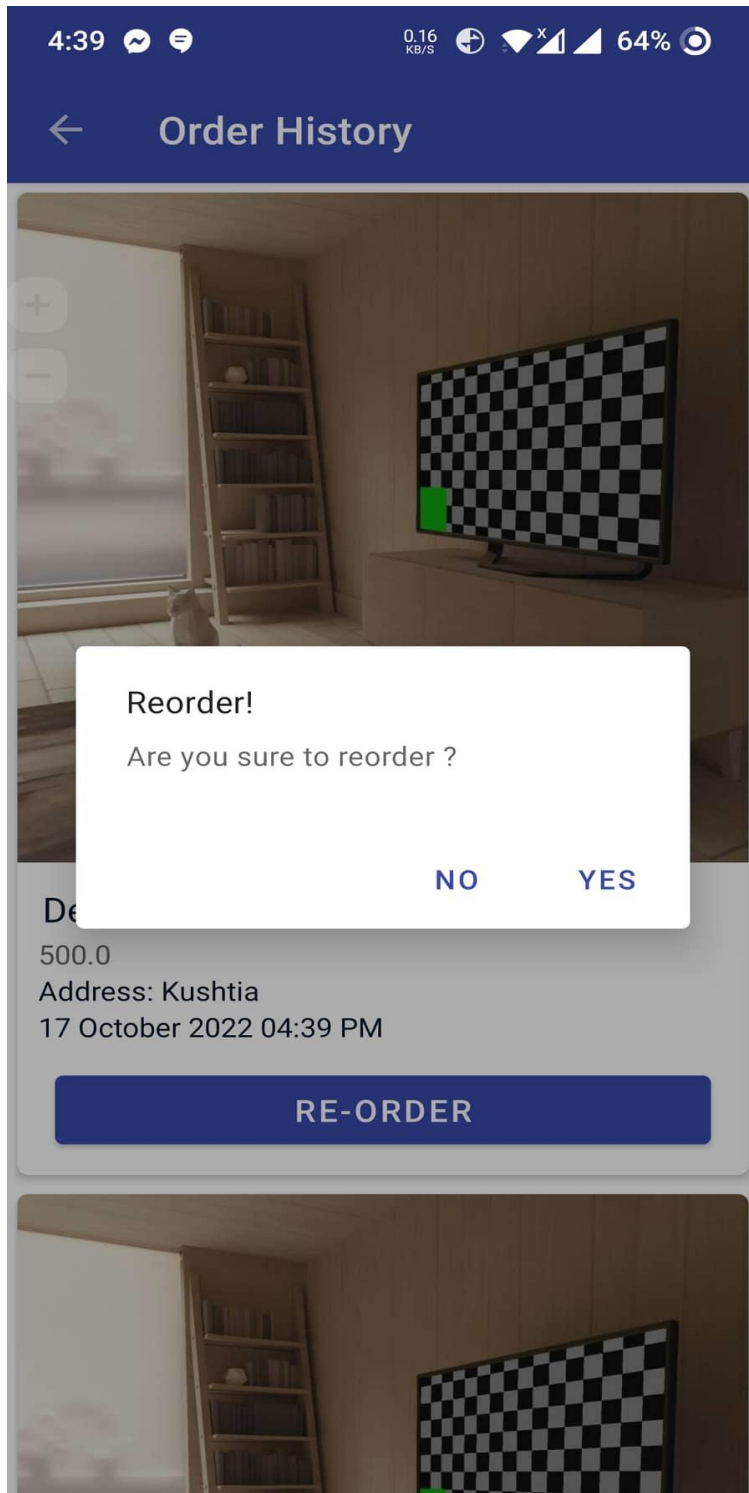
Profile updated successfully

Profile test

11.4 Module testing:



Order place



Re order



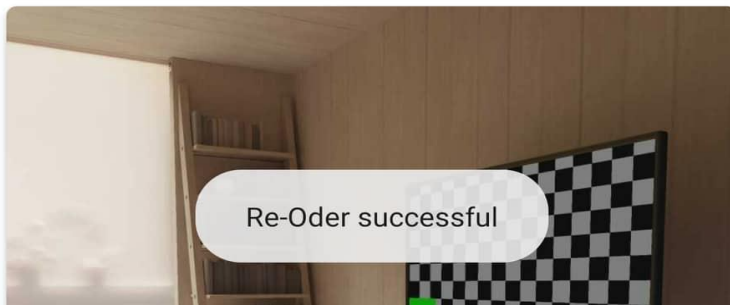
Demo Product

500.0

Address: Kushtia

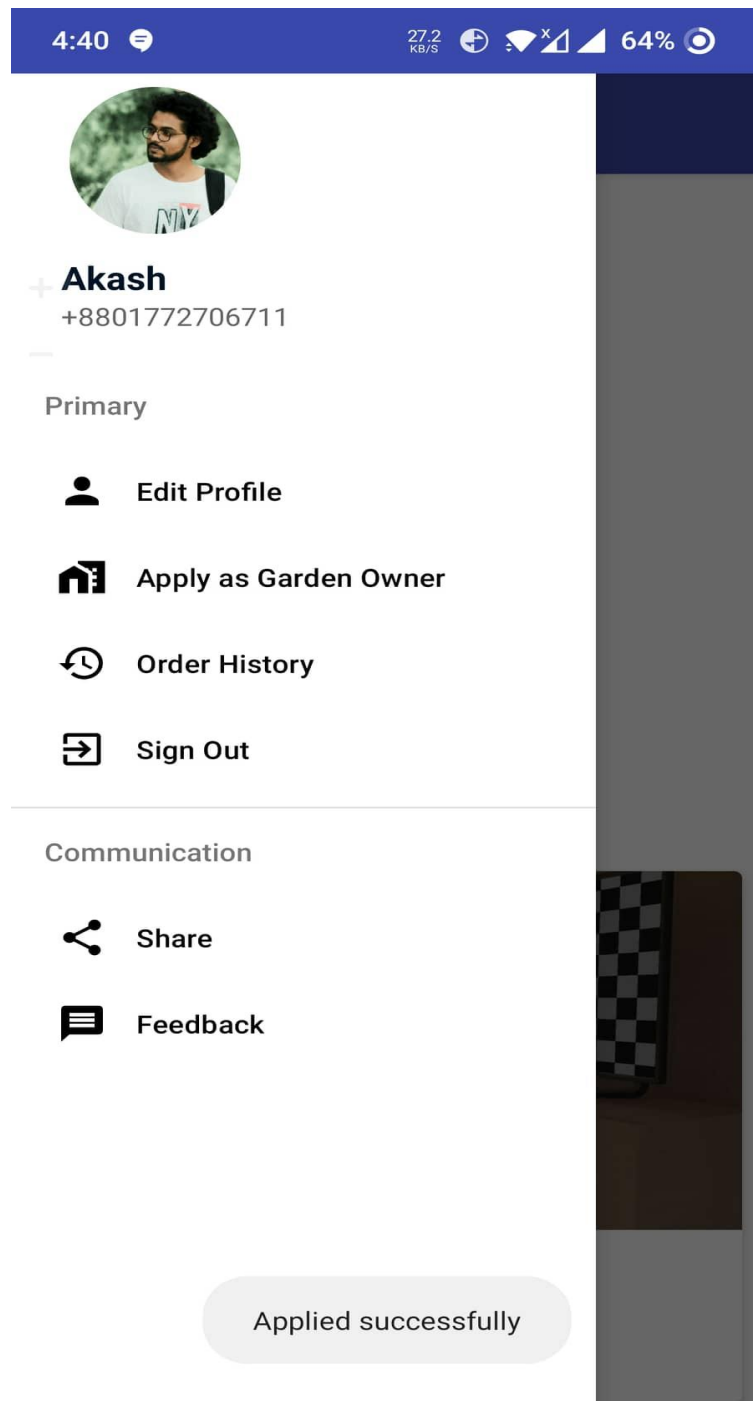
17 October 2022 04:39 PM

RE-ORDER



Re order Successful

11.5 Integration testing



Apply for garden owner

OUR SERVICES



Demo
Price: 3000.0 Taka
Only for testing purposes

RECENTLY ADDED



Demo Product
Price: 500.0 Taka
grocery

Order Placed successfully

Place order

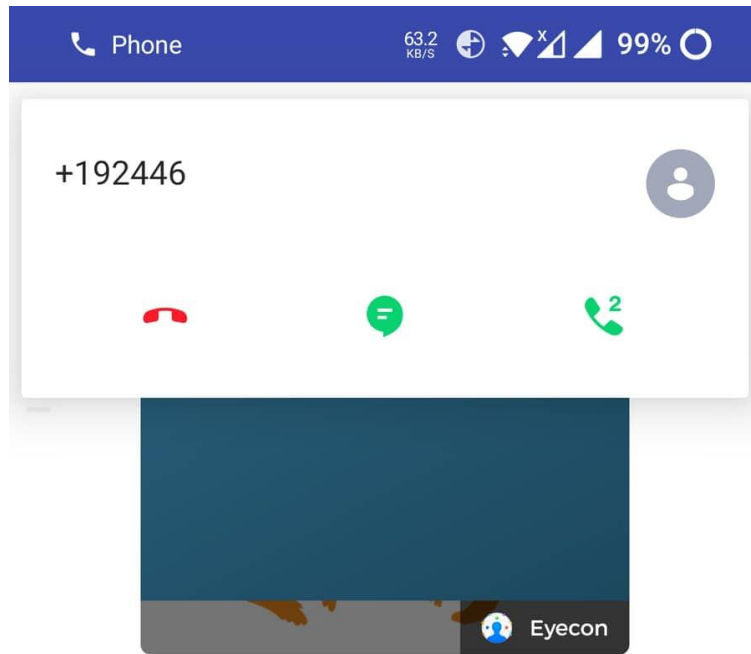
11.6 Acceptance testing:



ENTER 11 DIGIT MOBILE NUMBER

You will receive a verification code by
entering your phone number

SEND VERIFICATION CODE



ENTER VERIFICATION CODE

Enter the 6 digit verification code sent to your phone number

+8801772706711

RESEND CODE

46 SECONDS

VERIFY

6:23

9.29 KB/S   99% 



Grammo

OUR SERVICES



Demo

Price: 3000.0 Taka

Only for testing purposes

RECENTLY ADDED



Demo Product

Price: 500.0 Taka

grocery

6:23

5.08
KB/S




+ **Akash**

+8801772706711

Primary

 **Edit Profile**

 **Apply as Garden Owner**

 **Order History**

 **Sign Out**

Communication

 **Share**

 **Feedback**

6:23

15.9 KB/S    99% 

← Update Profile



Akash

+8801772706711

akash.itentlimited@gmail.com

Kustia

24

SAVE CHANGES

6:23

0.00 KB/S   98% 



+ Akash
+8801772706711

Primary

 **Edit Profile**

 **Apply as Garden Owner**

Already Applied

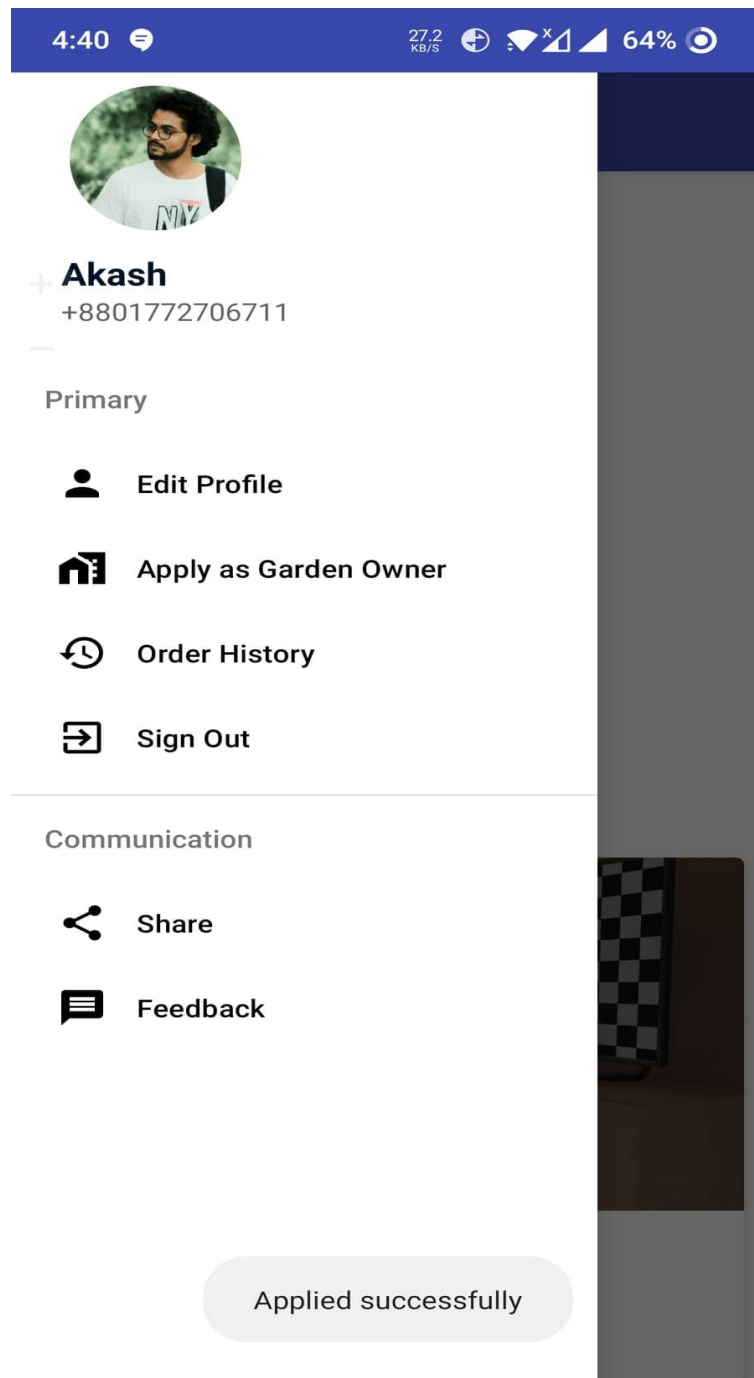
You already applied for garden owner.
Please wait until an admin accept
your request.

OKAY

 **Share**

 **Feedback**





11.7 Usability testing







+ Akash
+8801772706711

Primary

-  Edit Profile
-  Apply as Garden Owner
-  Order History
-  Sign Out

Communication

-  Share
-  Feedback

Profile updated successfully

4:39

0.28 KB/S

64%

Order History



Demo Product

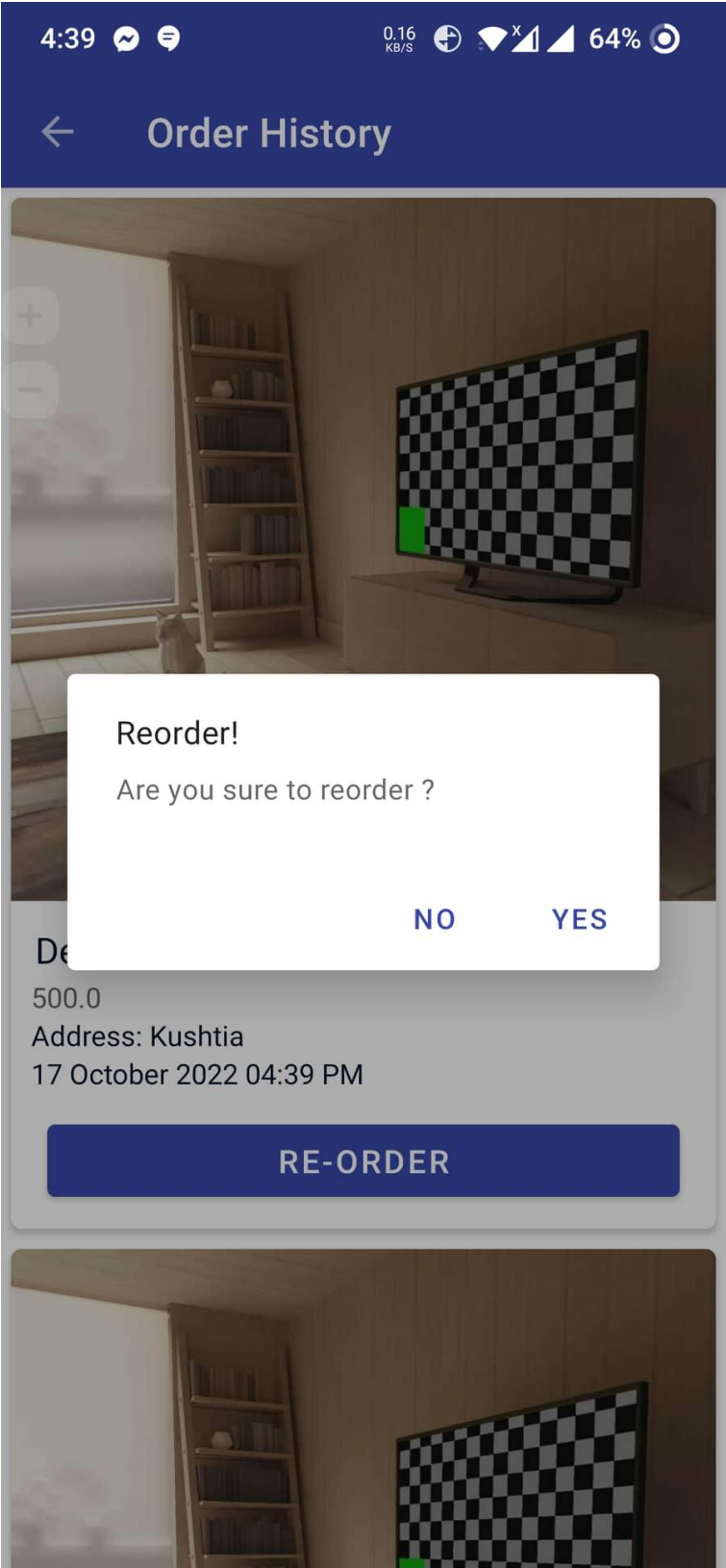
500.0

Address: Kushtia

17 October 2022 04:39 PM

RE-ORDER





OUR SERVICES



Demo
Price: 3000.0 Taka
Only for testing purposes

RECENTLY ADDED

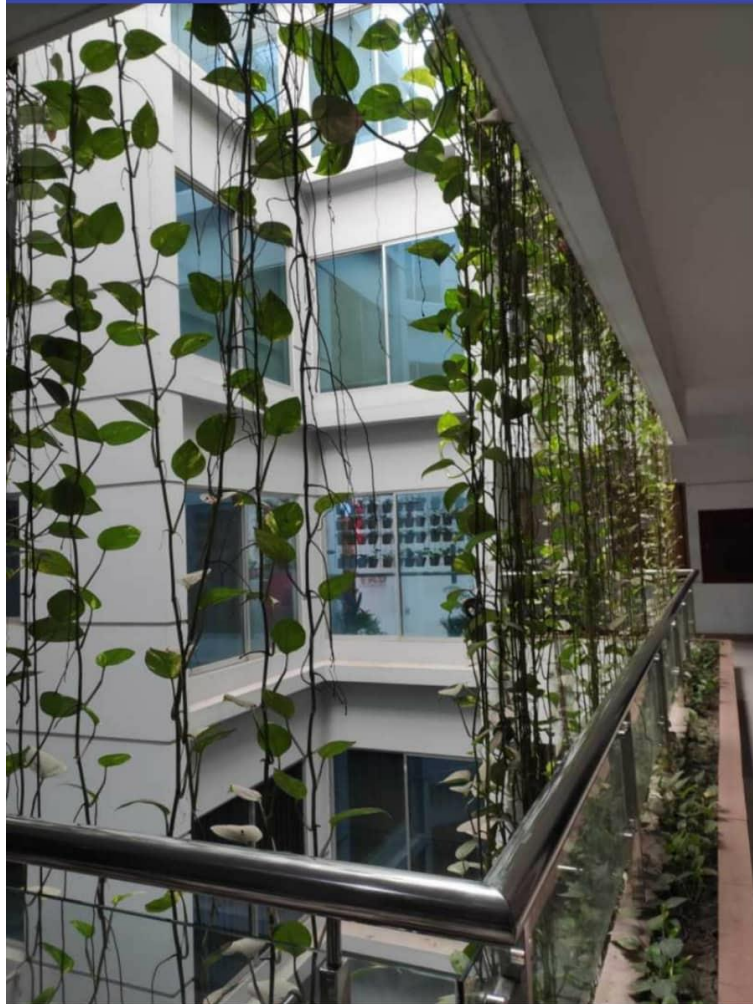


Demo Product
Price: 500.0 Taka
Order Placed successfully
grocery

4:36

4.62 KB/S

← Details



Demo

Price: 3000.0

Only for testing purposes

PURCHASE

Chapter 12

Implementation

12.1 Training

Software design are made to reduce torture of working, modernize most noteworthy things. A respectable structure ought to be not difficult to utilize so clients of that system can without a very remarkable stretch run the system. Nevertheless, whether or not being straightforward, considering nonattendance of data in IT region a couple of clients might get bother using the system. That is the explanation there should be a readiness time of any enormous system.

My framework is absolutely simple to utilize. Regardless, I will similarly run arranging stage in the school where I will sell it.

I truly need to set up a lot of experts of school who can set others up in the school and social individuals. I will show entire framework support and value to them essentially in a demo worker district. After my show I will check whether or not they can do that impeccably. Just in the wake of showing them the arranging stage should be possible in fact.

12.2 Big Bang (no pilot, parallel implementation scheme)

The Big Bang model of programming movement is organized around the likelihood that, starting with nothing, a fast development and extension of code will rapidly arise, subsequently making something completed in an essential second. Huge blast is a framework where a construction can move beyond structure information ordinarily. It seems to be strengthening structures where information doesn't free at any rate the framework gets resuscitated. This part making is a long sureness. I kept away from this part in my framework. Client should enter old framework information truly. Regardless, I have plans to acquire this part inside my framework future.

Chapter 13

Critical Appraisal and evolution

13.1 Object that could be met

No framework on the planet is completely wrapped up. There is dependably a level of stimulating. I have tried to satisfy each central and required targets so a school can run this design and use it for quite a while. Nonetheless, there are several extensions which could be met. They are:

- UI for school frontend for overall view.
- CMS for the UI.
- Different quantifiable depiction of data.
- Staff pay the board.

These are a couple of expansions which could be met anyway without them the structure can be used fittingly in light of the fact that all huge objectives are fulfilled.

Success rate against the object:

Success rate for the entire framework and all goals are acceptable. All connected information is taken and appeared in a superior manner so client can undoubtedly work. Understudy, parent and educator gateways contains all data with respect to them and all blunders are appropriately taken care of.

How much better have been done

The framework is now assembled utilizing Kotlin, XML which made the framework refreshed and quicker than some other comparable frameworks. Be that as it may, it would be better if I would remember artificial intelligence brainpower for the framework.

How better is the feature of the solution:

I have made a grammo mobile app for the farmer. If I use the system socially in our country and train the farmers then I think the features of my application will be useable to all and the goal of the system will be come out.

13.2 Which feature could not be touched:

I wanted to add some more feature like AI and maintenance dairy these features are not e touched because of covid-19 pandemic situation. Otherwise, I would definitely include those features too.

Chapter 14**Lessons learned****14.1 Pre-project -Review -Closing**

Pre project I find out about this kind of usage of web and study those articles and consider my endeavor. I share my thinking in regards to numerous people of santal neighborhood they increase me to achieving something of this idea. for completing start to end I for the most part get help with molding numerous people. We, as a matter of some importance, go with a decision about the essential point of convergence of this undertaking and make a manual for finish the endeavor. The overview of this kind of purpose of this neighborhood very impressive which is really help me to and give me more energy to finish this application.

14.2 What I have learned

Completing this endeavor, I have used the programming language Kotlin and XML. By encouraging this clear undertaking, I have progressed such countless models like rehashing, portraying capacities, making class and models, making objects, different articles, getting to various things data and individual models objects which is essentially used here. What's more, besides, insightful informational collection thought related with the design. By this model I have collected a good data on Kotlin and XML. These particular things are fundamental which I amassed from this endeavor.

I likewise realize who to work in crucial point in time and how to deal with that sort of circumstance

This work was a genuine project to me.

Chapter 15

Conclusion

15.1 Summary of the project

I fostered a framework named grammo. By this application the rancher and the nursery proprietor can sell their item straightforwardly to the client. Assuming the rancher deals with any issue, they can contact with us. The grammo group will help the clients for firming.

15.2 Goal of the project:

- Decrease syndicate
- Annual economic support
- Climate change

15.3 Success of the project:

I figure this task will be exceptionally useful for our country. It tends to be decline organization and would be extremely compelling for our economy.

15.4 Value of the project:

In our country maximum farmer and garden owner are illiterate. Sometimes they cannot find out the actual problem of his plant why they cannot get their aimed crop or can find out the problem but don't know how to solve this. For this reason, the grammo app has a feature. If any farmer or garden owner contact with us and ask for help, The grammo team will help to them.

15.5 My experience:

Grammo app is very useful. It decreases syndicate. And by this app I can buy freash food very easily.

Appendix

Name of requirement: Sign up (Register)			
Source: User	Sign off Users Must	Priority: User ID: have 01	
Functional requirements:	System will allow users to sign up for an account.		
Non-functional requirements:	The system has been designed as per the requirement so that it does not harm any human user.		
Details:	Target range	Acceptable range	Overview
User will own a User account	As the need of situation	As the need of situation	By sign up user can get an account.

Name of requirement: Login			
Source: User	Sign off Users	Priority: Must have	User ID: 01
Functional requirements:	System will allow users to login on registered account. User name: Akash		
Non-functional requirements:	The system has been designed as per the requirement so that it does not harm any human user.		
Details:	Target range	Acceptable range	Overview
User will own an User account	As the need of situation	As the need of situation	By login user can access his account.

Name of requirement: Logout			
Source: User	Sign off Users	Priority: Must have	User ID: 01
Functional requirements:	System will logout users from the accesses account.		
Non-functional requirements:	Get back to the home page.		
Details:	Target range	Acceptable range	Overview
User will own a user account	As the need of situation	As the need of situation	By logout user will stop the current access of the account.

Appendix 2

I have described the use case described in an earlier chapter of this statement. The remaining parts are described here

See category information's
Brief description: Registered user can see category information/documents.
Actor of this use case: Registered user
Pre-conditions: Must be need to register first in the system. Database stores all information.
Basic flow of events: On the profile page user will see the button as see his info and his product info. User will press and see the info or a message.
Extension of this: Must be added a valid land paper
Post Conditions: Successfully see category.
Special requirements: This is only available on the local server.

See Case status
Brief description: Registered user can category information or edit if available.
Actor of this use case: Registered user
Pre-conditions: Must be need to register first in the system. Database stores all information.
Basic flow of events: On the profile page user will see the button as see his info and his product info. User will press and see the info or a message.
Extension of this: Must be added a valid land paper
Post Conditions: Successfully see the vehicles case information.
Special requirements: This is only available on the local server.

Reference:

[1]<https://www.fao.org/asiapacific/perspectives/agricultural-statistics/global-strategy/results-in-the-region/bangladesh/en/>

[2]https://scholar.google.com/scholar?q=interview+elicitation+technique&hl=en&as_sdt=0&as_vis=1&oi=scholart

[3]https://scholar.google.com/scholar?q=Observation+elicitation+technique&hl=en&as_sdt=0&as_vis=1&oi=scholart

[4]<https://www.softwaretestinghelp.com/requirements-elicitation-techniques/>

[5] <https://kanbanize.com/blog/acceptance-testing/>

183-16-392

ORIGINALITY REPORT

28% SIMILARITY INDEX	22% INTERNET SOURCES	5% PUBLICATIONS	20% STUDENT PAPERS
--------------------------------	--------------------------------	---------------------------	------------------------------

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	13%
2	Submitted to Daffodil International University Student Paper	8%
3	Submitted to University of Greenwich Student Paper	3%
4	Submitted to NCC Education Student Paper	1%
5	Submitted to Curtin University of Technology Student Paper	1%
6	eprints.utar.edu.my Internet Source	<1%
7	Submitted to University of Technology, Sydney Student Paper	<1%
8	ir.library.ui.edu.ng Internet Source	<1%
9	Submitted to Sunway College Student Paper	<1%

10	Submitted to King's Own Institute Student Paper	<1 %
11	Submitted to Melbourne Institute of Technology Student Paper	<1 %
12	Submitted to Middlesex University Student Paper	<1 %
13	Submitted to University of Ulster Student Paper	<1 %
14	Submitted to Rajarambapu Institute of Technology Student Paper	<1 %
15	eprints.utm.my Internet Source	<1 %