

**ANDROID BASED PLATFORM FOR CONNECTING FARMERS AND
CONSUMERS**

by

FAZLEY RABBI

ID: 162-15-8085

MD. HUMAYUN KABIR

ID: 162-15-8008

MD. NAJMUL HUDA LITON

ID: 162-15-8090

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering.

Supervised

by

Moushumi Zaman Bonny

Senior Lecturer

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

MAY 2019

APPROVAL

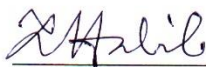
This Project titled “Farm Food”, submitted by Fazley Rabbi, ID No: 162-15-8085, Md. Humayun Kabir, ID No: 162-15-8008, Md. Najmul Huda Liton ID No: 162-15-8090, to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on May 04, 2019.

BOARD OF EXAMINERS



Dr. Syed Akhter Hossain
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



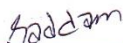
Md. Tarek Habib
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Moushumi Zaman Bonny
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Md. Saddam Hossain
Assistant Professor
Department of Computer Science and Engineering
United International University

External Examiner

DECLARATION

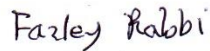
We hereby declare that, this project has been done by us under the supervision of **Moushumi Zaman Bonny, Senior Lecturer, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

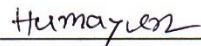


Moushumi Zaman Bonny
Senior Lecturer
Department of CSE
Daffodil International University

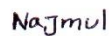
Submitted by:



Fazley Rabbi
ID: -162-15-8085
Department of CSE
Daffodil International University



Md Humayun Kabir
ID: -162-15-8008
Department of CSE
Daffodil International University



Md Najmul Huda Liton
ID: -162-15-8090
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project successfully.

We really grateful and wish our profound our indebtedness to **Moushumi Zaman Bonny, Senior Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “Android Application” to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Dr. Syed Akhter Hossain, Professor and Head, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

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

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Mobile applications are a massive way to connect a user with a phone. Enhancement of mobile network gives a fabulous demanding of mobile applications. Nowadays every person using at least one mobile phone for business purpose or communication purpose. The main aim of this project is to build an android application that helps the farmer and consumer to maintain communication with each other. Another aim is it can be useful for an extinct middle man and it will help to maintain the same food price hole of the country. In this platform, any farmer whole of the country can add his foods and they can run his own business from home. Farmer will get the actual price of the product what they expected after cultivation. Also, the consumers will be benefited they will buy pure foods at a reasonable price. In this way, the middle man will totally extinct from our country. Farmers and consumers will build up a strong relationship. The whole of the country products price will remain the same all of the season. This platform is a win-win platform for farmers, consumers and also for our country. The best part of the platform is farmers will think he is not only a farmer he is also a businessman.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
CHAPTER	
CHAPTER 1: INTRODUCTION	1-3
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Expected Outcome	2
1.5 Report Layout	3
CHAPTER 2: BACKGROUND	4-5
2.1 Introduction	4
2.2 Related Works	4
2.3 Comparative Studies	4
2.4 Scope of the Problem	5
2.5 Challenges	5
CHAPTER 3: REQUIREMENT SPECIFICATION	6-10
3.1 Business Process Modeling	6

3.2 Requirement Collection and Analysis	7
3.3 Use Case Modeling and Description	7
3.4 Logical Data Model	9
3.5 Design Requirements	10
CHAPTER 4: DESIGN SPECIFICATION	11-23
4.1 Front-end Design	11
4.2 Back-end Design	19
4.3 Interaction Design and UX	23
4.4 Implementation Requirements	23
CHAPTER 5: IMPLEMENTATION AND TESTING	24-29
5.1 Implementation of Database	24
5.2 Implementation of Front-end Design	25
5.3 Implementation of Interactions	26
5.4 Testing Implementation	26
5.5 Test Results and Reports	27
CHAPTER 6: CONCLUSION AND FUTURE SCOPE	30
6.1 Discussion and Conclusion	30
6.2 Scope for Further Developments	30
REFERENCES	31-32
APPENDICES	33-44

List of figures	Page
Figure 3.1: A general Block diagram of business process model	6
Figure 3.2: Admin Use Case Diagram	8
Figure 3.3: User use case diagram	8
Figure 3.4: Logical Data Model Diagram	9
Figure 4.1: Figure 4.1: (a) Login and (b) Registration Screen	11
Figure 4.2: (a) Home and (b) Navigation Screen	12
Figure 4.3: (a) Product Entry Button and (b) Product Entry Screen	13
Figure 4.4: (a) Product Package Entry and (b) Product Description Screen	14
Figure 4.5: Figure 4.5: Product Order and Order Confirmation Message	15
Figure 4.6: (a) Order and (b) Interests View Screen	16
Figure 4.7: (a) Saved Items View, Delete option and (b) Contact Us Screen	17
Figure 4.8: (a) Product Like, Saving Option and (b) App Share Option	18
Figure 4.9: Firebase Home	19
Figure 4.10: Firebase Database User Authentication	19
Figure 4.11: User Information in Database	20
Figure 4.12: Product Description in Database	20
Figure 4.13: Product Package Description in Database	21
Figure 4.14: Product Order Description in Database	21
Figure 4.15: Saved Product Description in Database	22
Figure 4.16: Contact Us Information in Database	22

List of tables	Page
Table 5.1: Test Case for login	28
Table 5.2 Test Case for Test Home page	28
Table 5.3 Test Case for Test order confirm	29

CHAPTER 1

Introduction

1.1 Introduction

Every time we buy foods from any shop we think about the food price and purity. We want to buy pure foods at a reasonable price but the problem is we have to buy foods with extra cost because of the middle man. The middle man also preserves the food with the harmful chemicals that are dangerous for human health. That's why Farm Food

This Android-based platform for connecting consumers and farmers. Consumers will buy foods from farmers directly and this will reduce the food cost also it will ensure pure food. Farmers will also get paid the actual price of the product. So this platform is for farmers and consumer.

1.2 Motivation

The foundation of F2C platform is the connection of trust between farmers and consumers. Farm direct marketing enables the producer to take responsibility for and rewards the delivery quality agro-food products directly to the consumer through a variety of delivery methods for examples farmers' directly deliver his product [1].

1.3 Objectives

The main focus of the project is to connect farmers with consumers. The goal is to ensure that farmers get the higher remuneration for the Crop production and meeting the satisfaction level of the consumers through direct sale of the agricultural commodity by the farmers to the consumer. Direct marketing of agricultural produce helps in complete elimination of middle men and commission agents who charge high level of commission

fee from the agriculturists/farmers coming to the market yards for selling their produce and then artificially inflate the retail prices [2]. In a manner of speaking, the farmer using this method grows a “product” more than a crop.

1.4 Expected Outcome

- Price will be same whole of the country.
- Making a direct connection Farmers to Consumers.
- Direct marketing of agricultural products helps the middle men and commission agents to be completely eliminated.
- Since small amounts of agricultural products can be sold, small producers can actively participate [3].
- The farmer is setting the market price or more price control. Good products and services can have attractive prices and can therefore be profitable for small farms.
- Generally payment is instant.
- Farmers also receive direct feedback on products and service from clients. Through this input, the farmer can improve his / her business and increase the profitability of the farm.
- Personal pleasure and satisfaction. Every farmer knows how satisfying it is to produce a good crop, but direct farmers have a special feeling because they provide fresh, healthy food directly. They have the opportunity to share the food with people who appreciate its value and let them know how much they appreciate their efforts.
- Work with your family at house.
- Farmers can run a personal business enterprise.

1.5 Report Layout

We have organized this report as follows:

In the chapter 1, motivation, purpose and expected outcome of this project is already described. We briefly discuss about background of our project. In the second section. We are also describing related works, comparative studies, scope of the problem and challenges. After the chapter 2, we discuss the Requirement Specification of the project. In this section we defined the business process modeling, requirement collection and analysis, use case modeling and description, logical data model and design requirements. We specified the front-end design and back-end design on the chapter number 4. After that chapter 5, we implementation the whole project and testing every part of the project. The next chapter is 6, we discuss about the conclusion and scope for the future developments. At last we declared all of the reference to relate our report information.

CHAPTER 2

Background

2.1 Introduction

In recent days, most of us spend loss of time on internet. We are more comfort online shopping or online activity. That's why we thought that to develop something new that can helpful for farmer and consumer both and it could make their connectivity between farmer and consumer very easier. It could be help easier decision for buying food direct from farmer. When we start to design the platform, we always keep some face in our knowledge that the design should be only for who will use the app. It supports farmer can sell all types food from the app. Consumers can easily select foods check the price of the food and they can confirm the order. The type of application is not present in time. That's why we inspired and can develop this application.

2.2 Related Works

This is very difficult to connect farmers with consumers. It could be said there is no suitable platform for connect them. In the present time android smart phone user increasing day by day [4]. Most of the smart phone screen 5 to 7 inch that's why web page is not properly displayed in this smart phone screen. There for we believe that "Android Based Farmer to Consumer" Application help people for buying foods without extra cost.

2.3 Comparative studies

In the early stages of the development of a science branch, the comparative technique is often used [5]. It can help the researcher climb the initial level of exploratory research studies to a higher level of the general theoretical model. Most of the people are now buying foods through chaldal.com this is one of the popular online shop for buying

grocery foods. But this company supplying foods by their own. So the food price is relatively same as market price. They didn't ensure the purity of the foods, so we develop our platform to give consumers pure and cost effective foods also ensure the farmers get paid actual price.

2.4 Scope of the problem

Farm food is a food selling and buying platform through "Android Base Application". Its main programming language for front-end will be java and for back-end will be Jason. We have focused to simplify upload products and buying product easily our target is to minimize manual data entry and ensure data accuracy also the security during the data process. We have created options for the consumers they can see the products and the package prices also the orders. Farmers will also be able to view there uploaded products and the orders.

2.5 Challenges

There is a common thing to perform an activity is challenge There is no work without challenge similarly, in our project we also face some challenges. The first challenge is make this platform usable for the farmers they can put their products easily and maintain their business. We know in our country farmers are not very educated so we try to give easier functions that can help farmers to operate the app. Second challenge is connecting consumers to the farmers

CHAPTER 3

Requirement specification

3.1 Business process Modeling

Modeling business processes map regular business processes and find ways to improve them. It is part of the business process management practice. Process modeling software provides an analysis of the process in an organization and contrasts it with processes to make it more efficient [6].

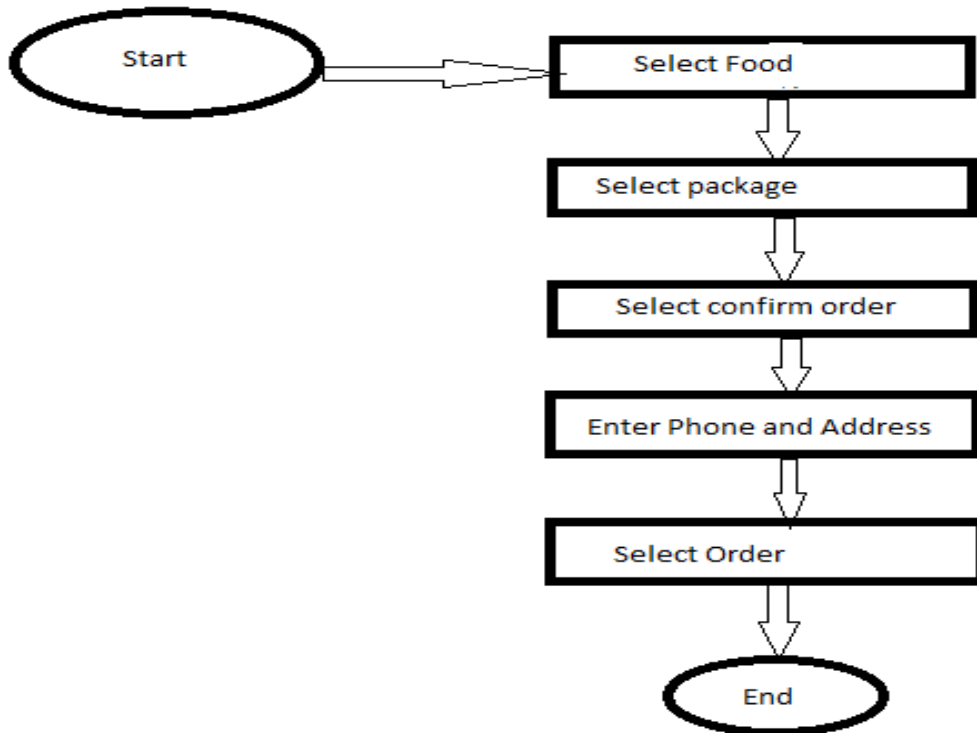


Figure 3.1: A general Block diagram of business process model

3.2 Requirement collection and Analysis

The software requirement in the system is functional and non-functional. Function means providing users with a specific service. For example, when the customer selects "view balance" they must be able to look at their latest account balance in the context of the banking application. Software requirement may also be non-functional, a requirement for performance. A non-functional, for example. The requirement is to show users every page of the system within 5 seconds [7].

Therefore, there are two types of software requirement, functional and non-functional.

3.3 Uses-Case modeling & Description

A Unified Modeling Language (UML) use case diagram is a type of interactive diagram defined by and created from a use case analysis. Its resolution is to present a graphical overview of the functionality of a system in terms of actors, their objectives and any interdependencies between those uses [8]. A use case diagram's main purpose is to show for which actor what system functions are performed.

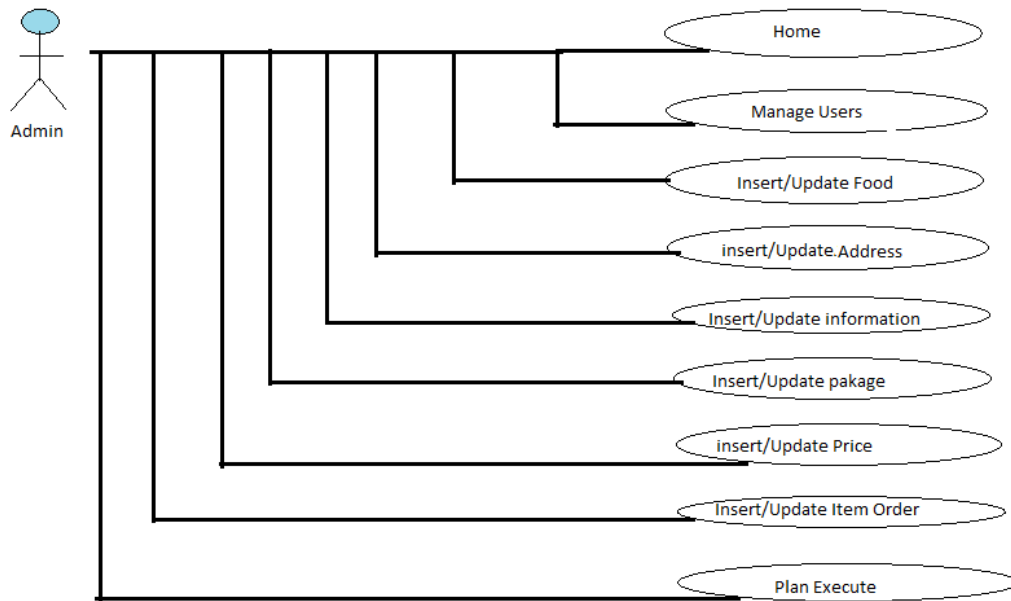


Figure 3.2: Admin Use Case Diagram

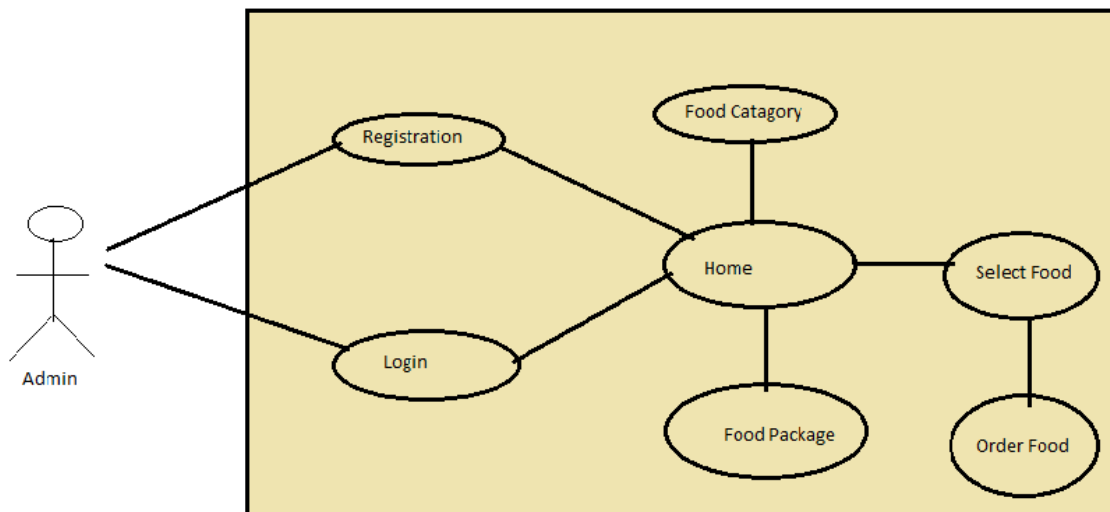


Figure 3.3: User use case diagram

3.4 Logical Data Model

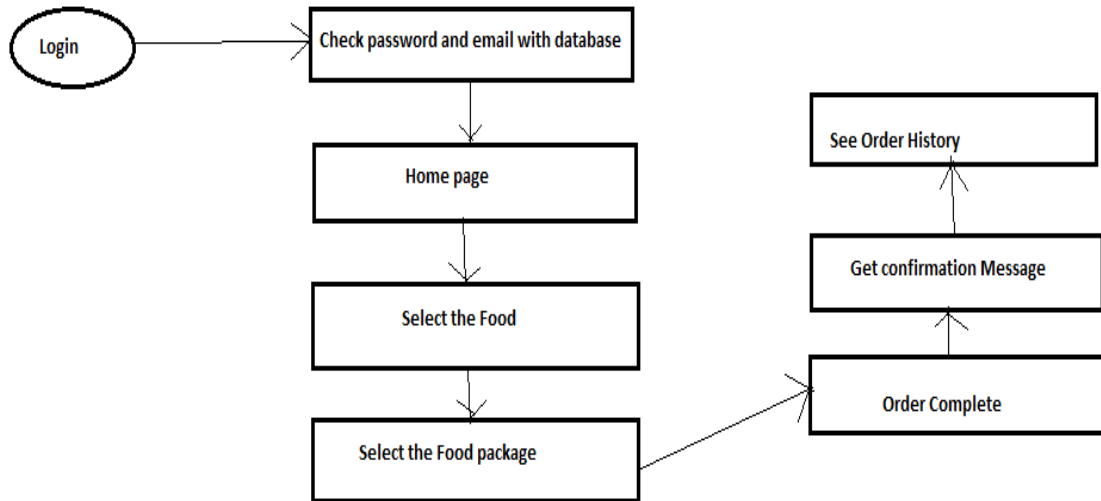


Figure 3.4: Logical Data Model Diagram

3.5 Design Requirements

In this system we work in android platform. In this system farmer can easily sell his product and consumer can easily buy product. Android is known to be an open source mobile operating system based on the Linux kernel [9]. It is mainly designed for mobile touch screens with smartphones and tablet computers. Digital cameras, smart watch, smart television and other electronics appliances also implemented the technology. We used xml layout for design purpose. An XML - based layout in android is a file that describes the various widgets to be used in the UI and the relationships between those widgets and their containers. Android deals with the layout files as resources [10]. XML - based layouts are very helpful when you know the components of the UI.

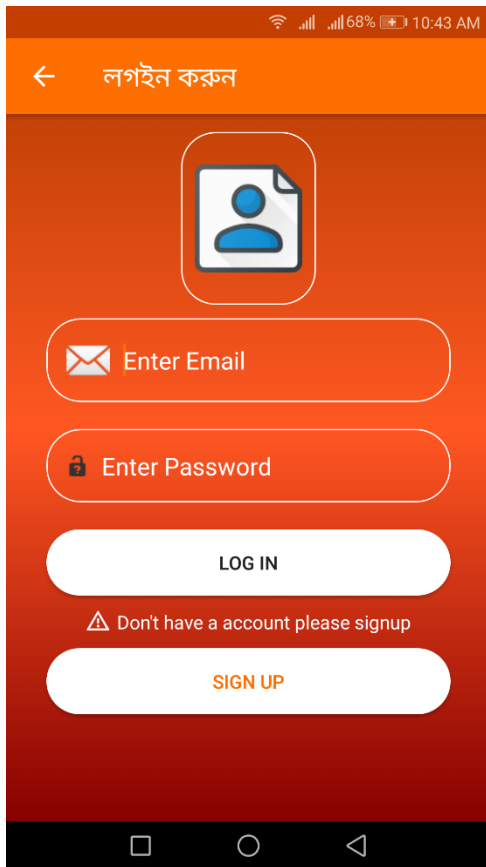
XML based layout has the following benefits

XML is a popular format that is commonly used. Many developers are relaxed with it later. It helps to divide the UI from the logic of code. This gave flexibility to change one without having a significant impact on the other. Production of XML output is cooler than writing direct code, making drag-and-drop UI tools easier to create android interfaces [11].

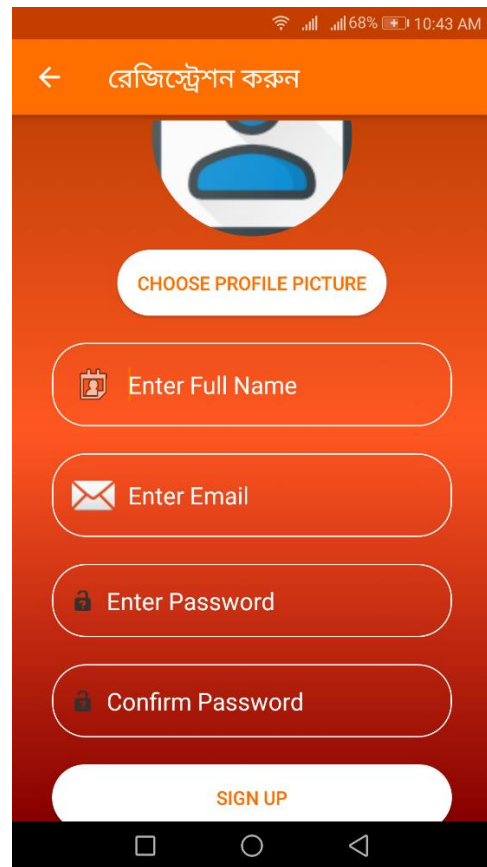
CHAPTER 4

Design Specifications

4.1 Front End Design

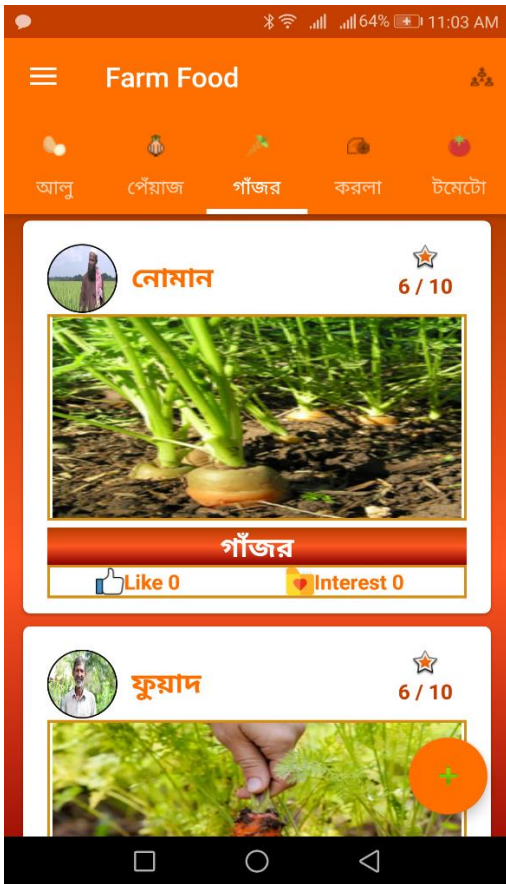


(a)

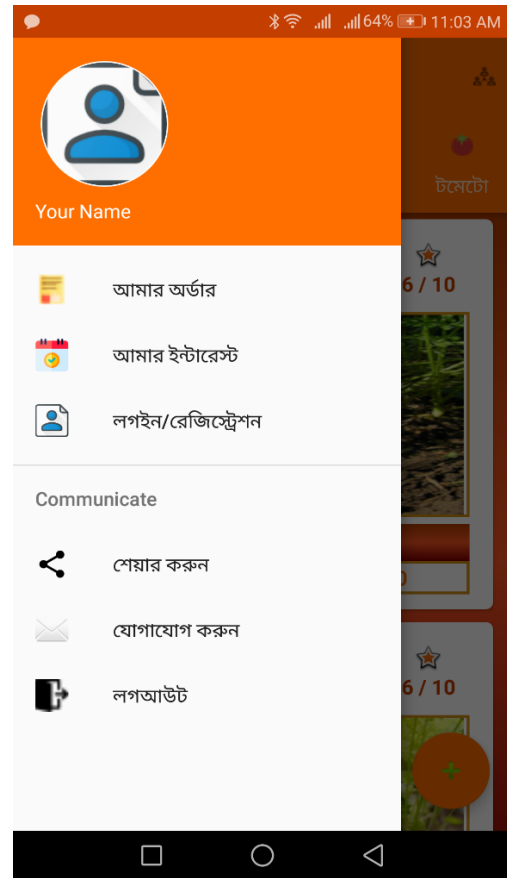


(b)

Figure 4.1: (a) Login and (b) Registration Screen



(a)

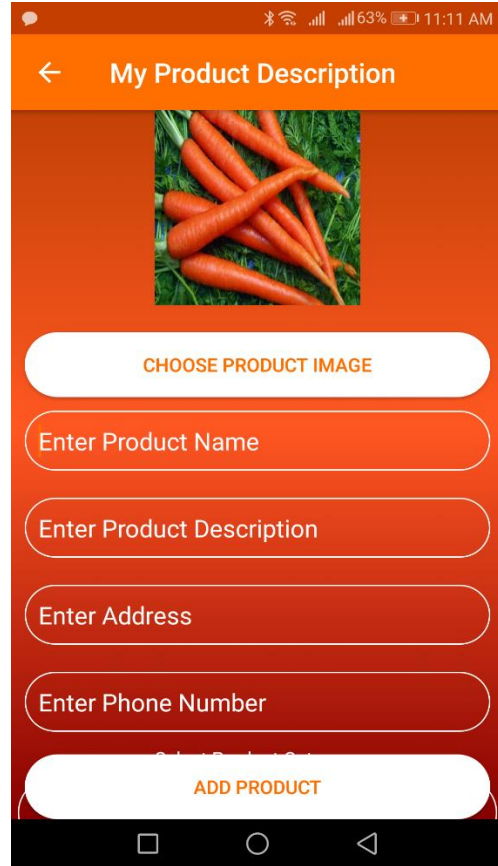


(b)

Figure 4.2: (a) Home and (b) Navigation Screen

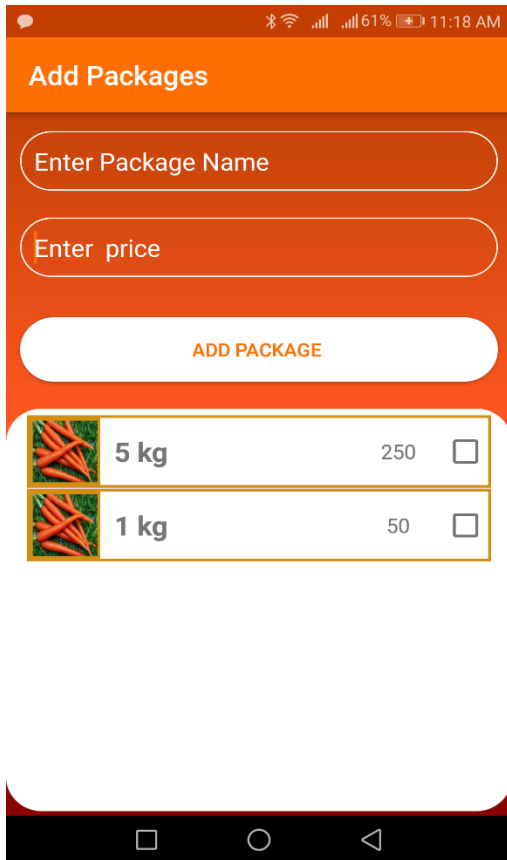


(a)



(b)

Figure 4.3: (a) Product Entry Button and (b) Product Entry Screen

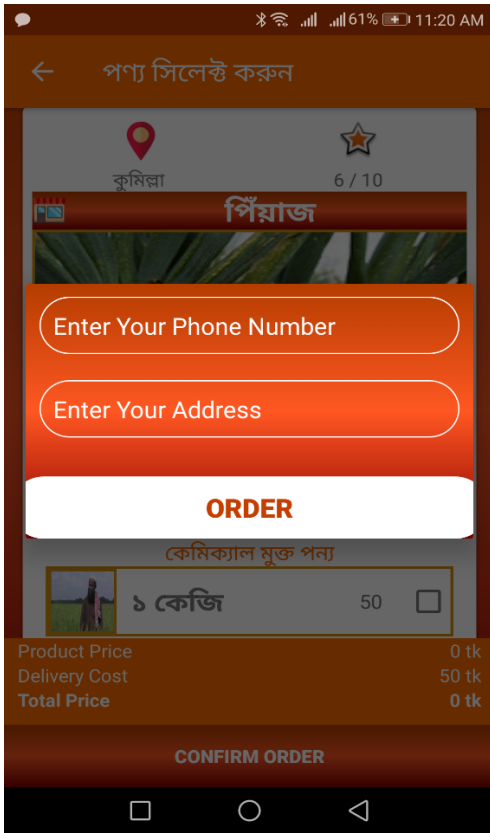


(a)

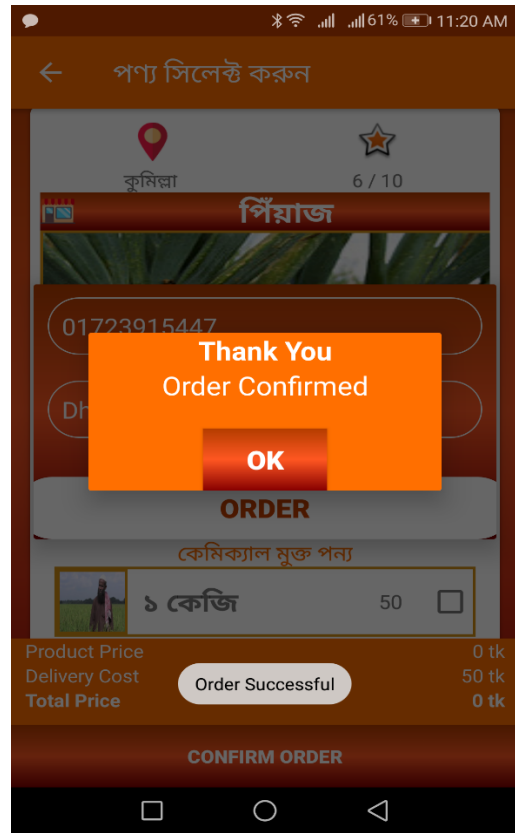


(b)

Figure 4.4: (a) Product Package Entry and (b) Product Description View Screen



(a)



(b)

Figure 4.5: Product Order and Order Confirmation Message

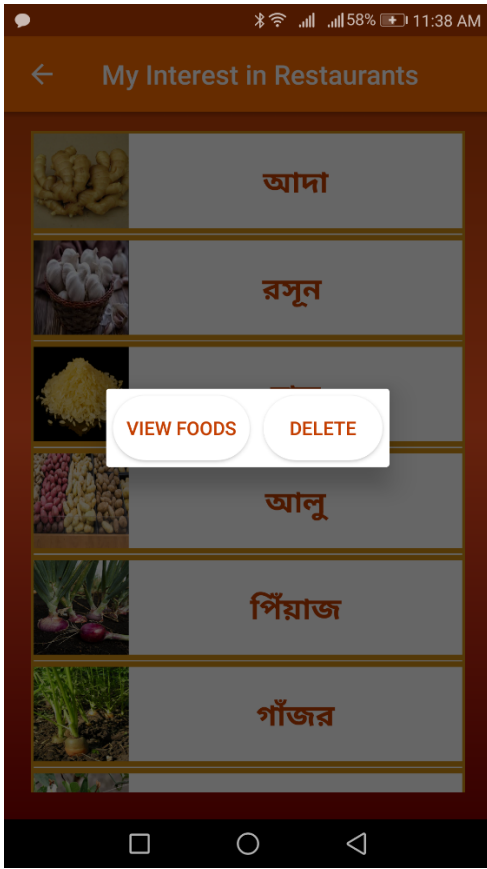


(a)

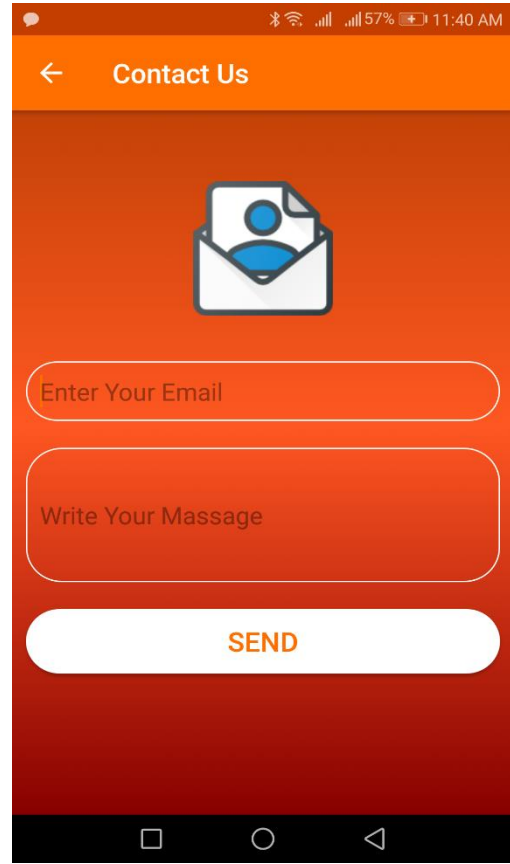


(b)

Figure 4.6: (a) Order and (b) Interests View Screen



(a)

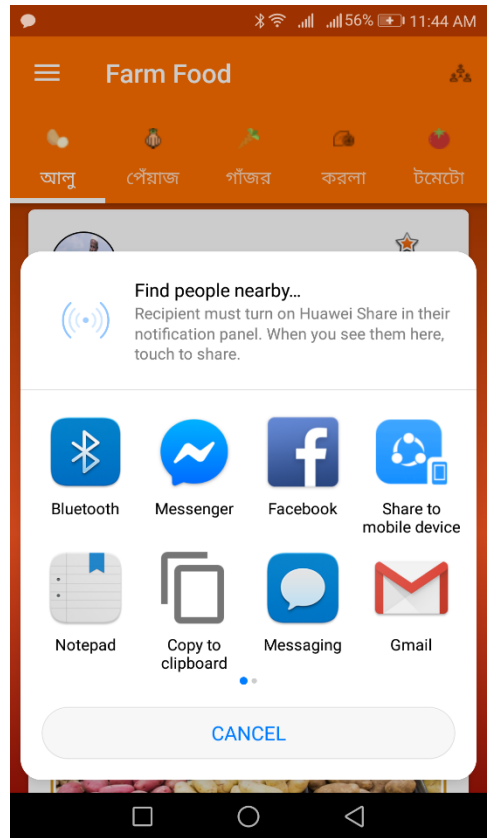


(b)

Figure 4.7: (a) Saved Items View, Delete option and (b) Contact Us Screen



(a)



(b)

Figure 4.8: (a) Product Like, Saving Option and (b) App Share Option

4.2 Back End Design

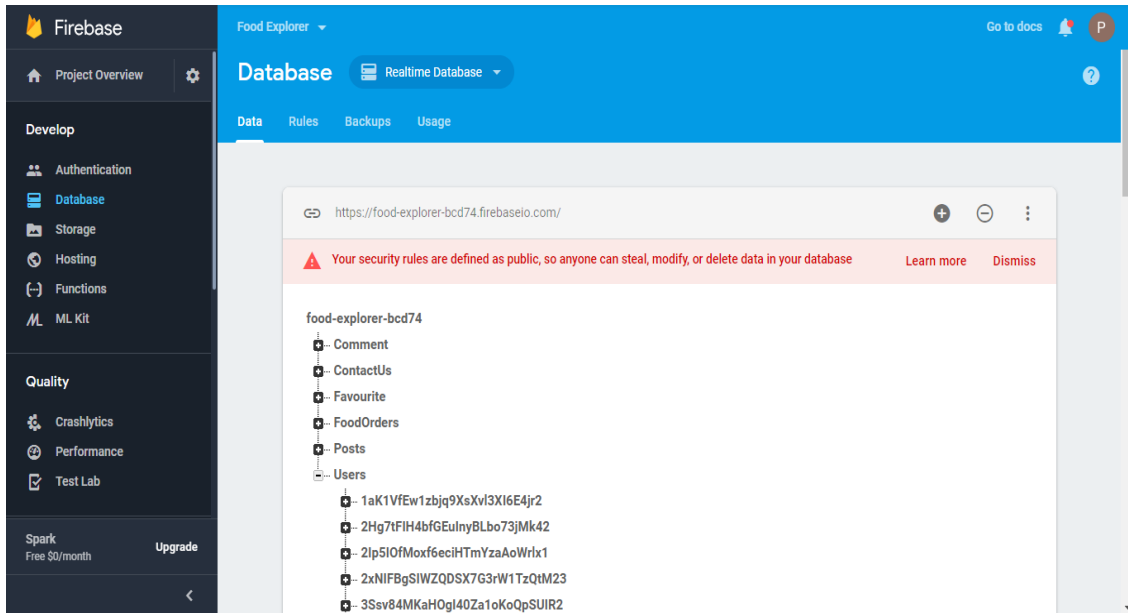


Figure 4.9: Firebase Home

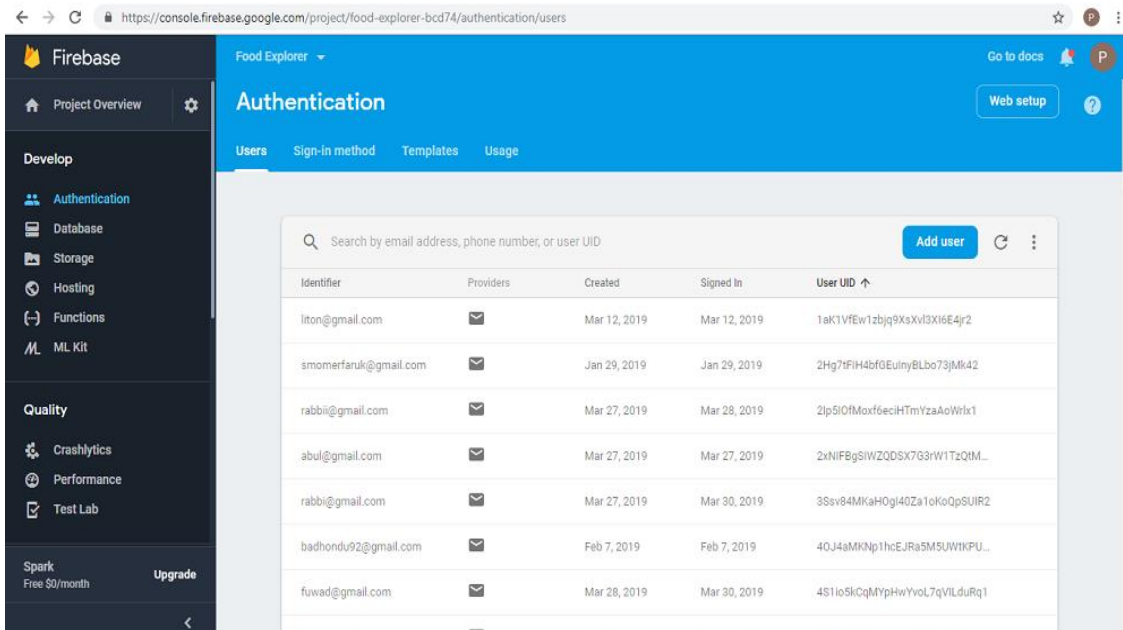


Figure 4.10: Firebase Database User Authentication

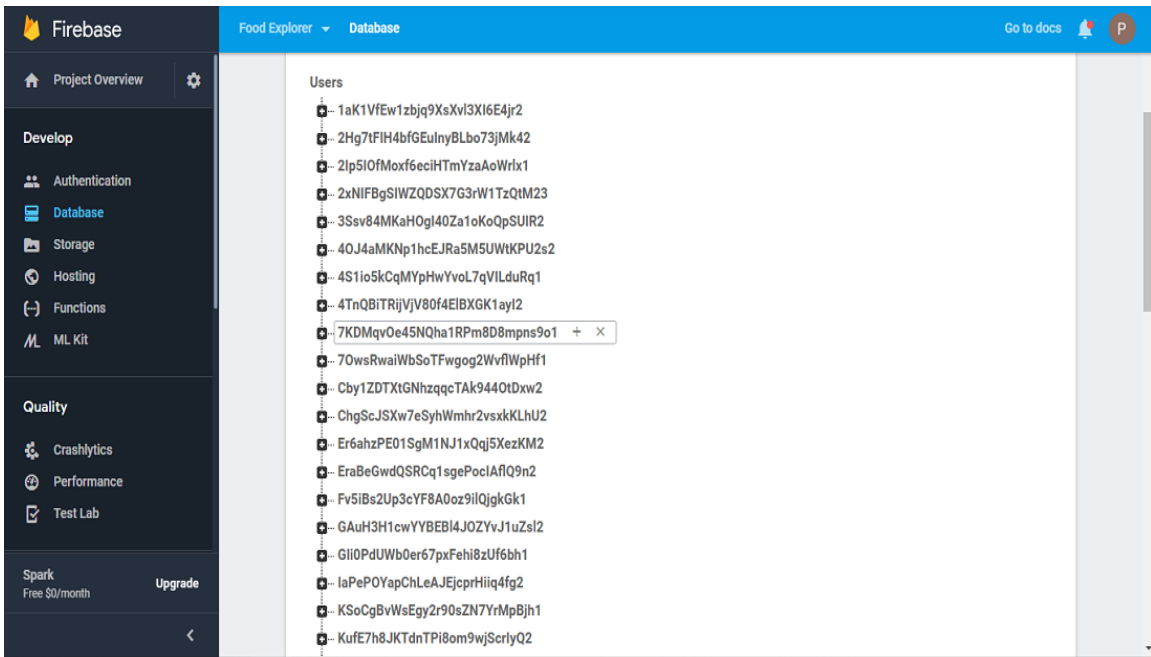


Figure 4.11: User Information in Database

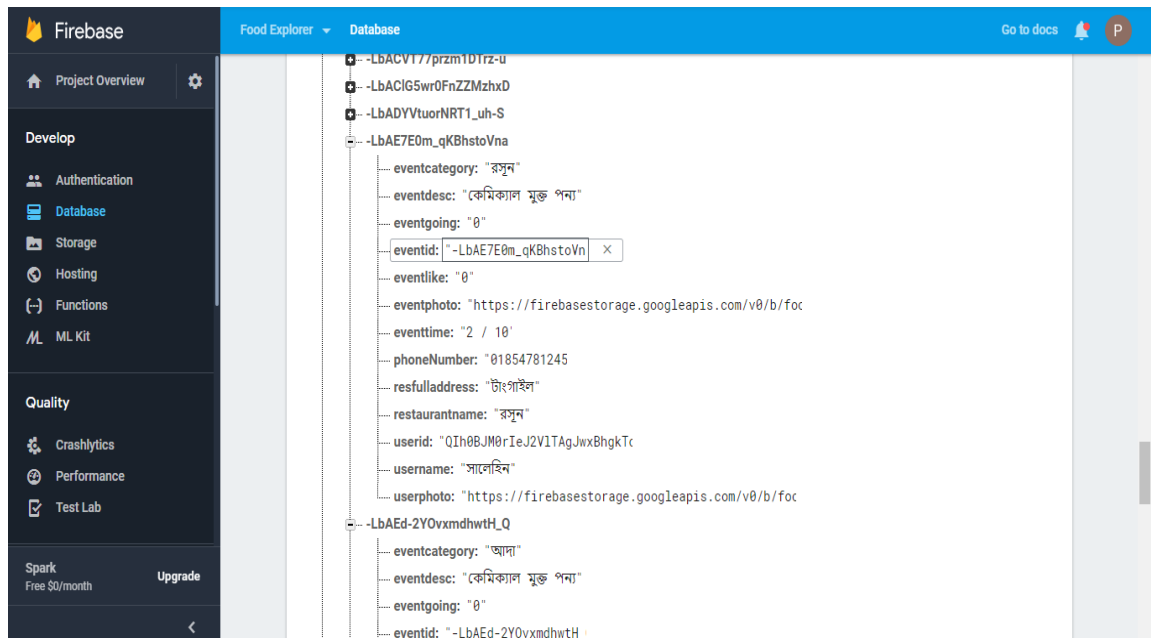


Figure 4.12: Product Description in Database

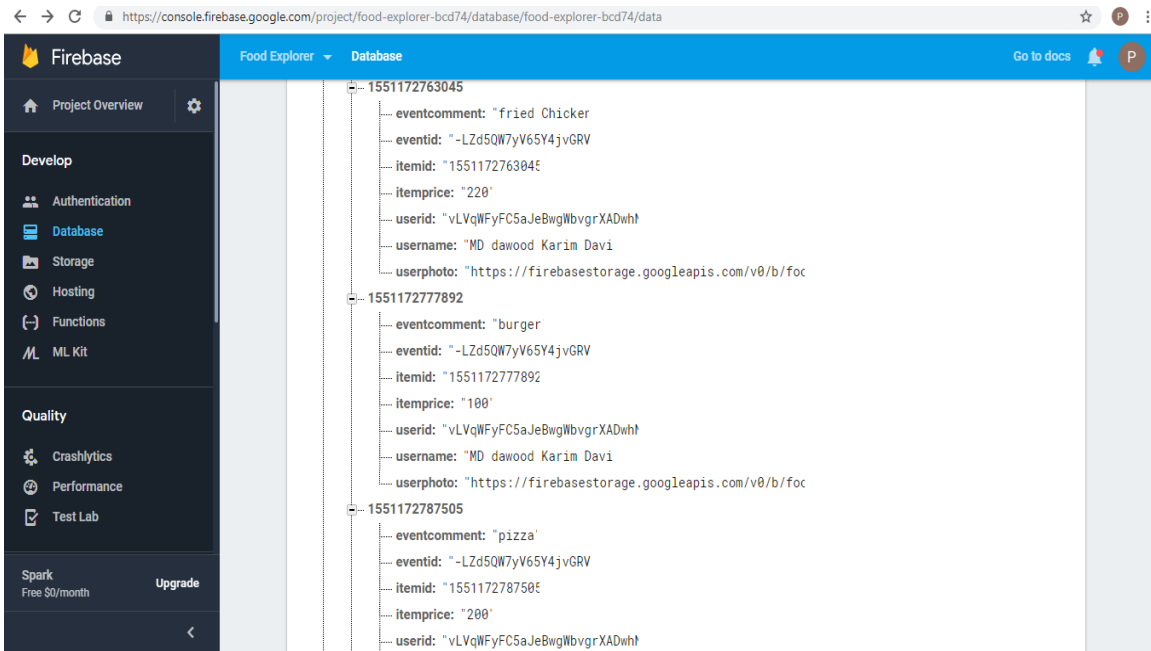


Figure 4.13: Product Package Description in Database

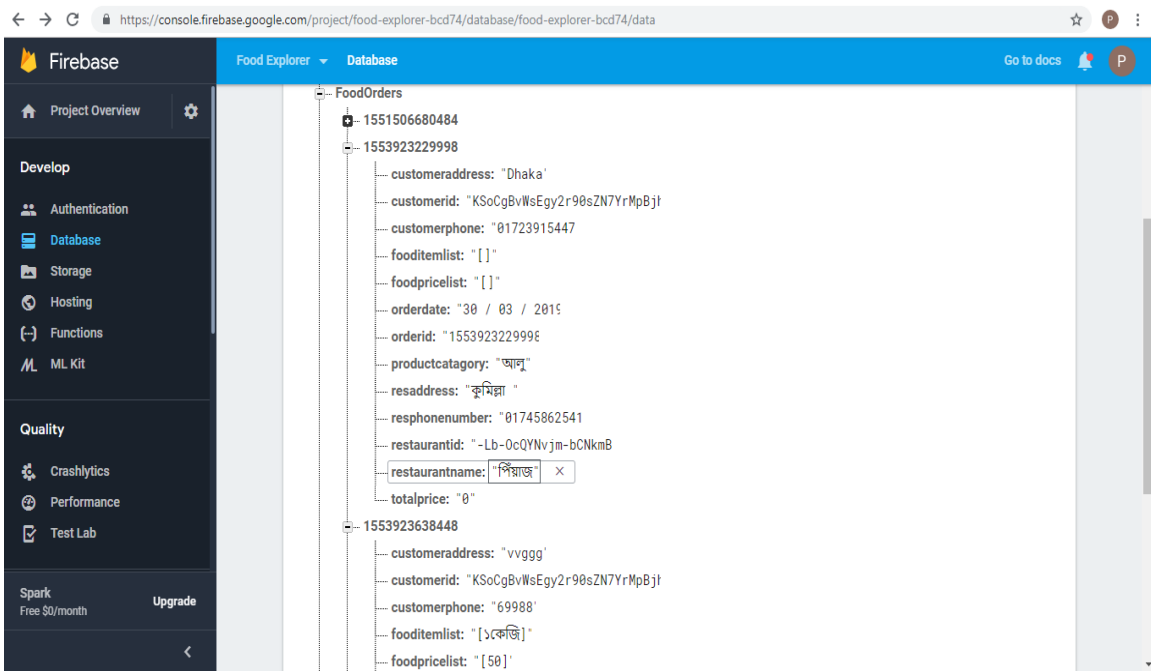


Figure 4.14: Product Order Description in Database

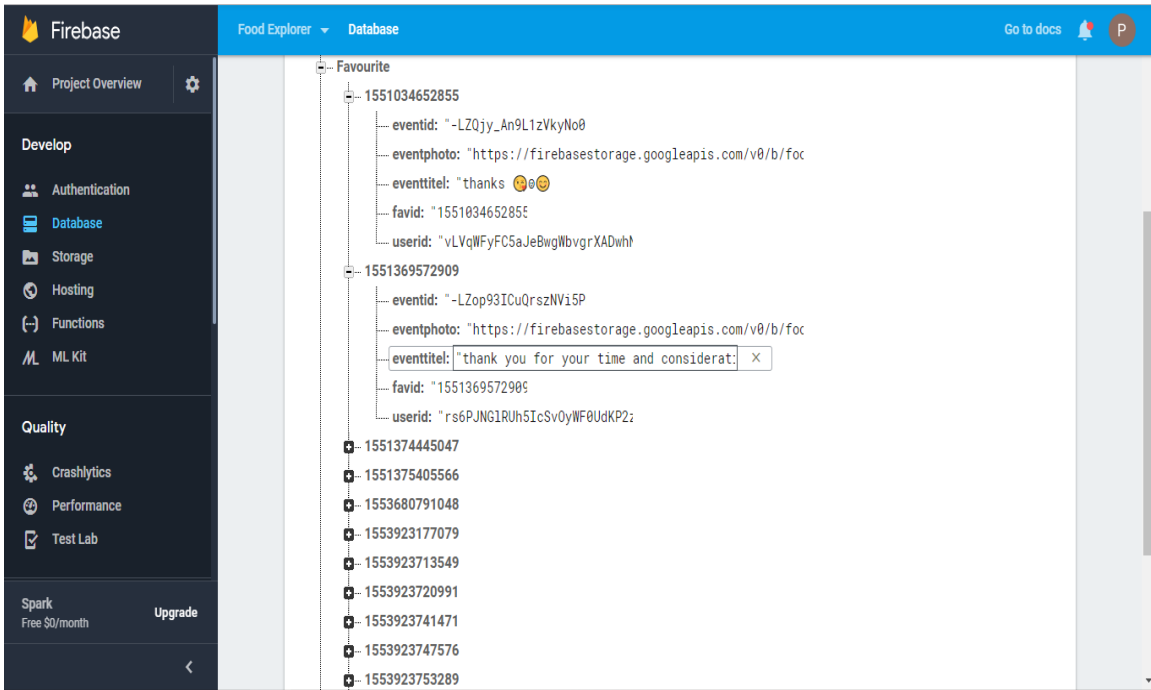


Figure 4.15: Saved Product Description in Database

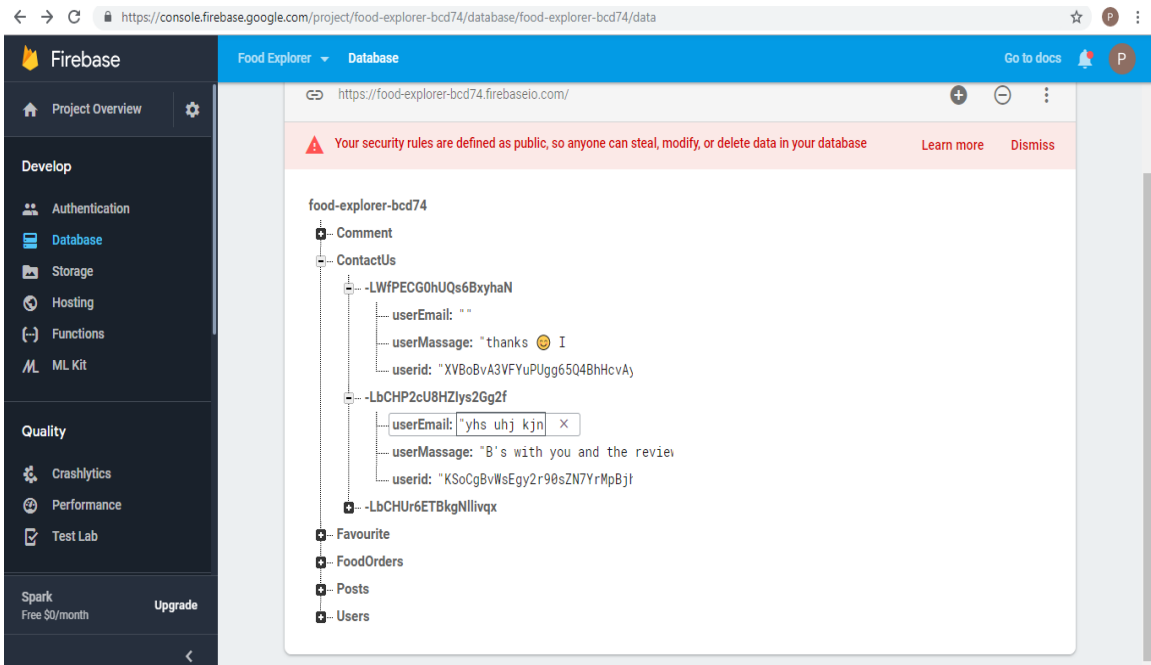


Figure 4.16: Contact Us Information in Database

4.3 Interaction Design and UX

Interaction design is specifically a discipline, while system - user interaction via interface. Actually, an interaction design which can describe the user problem domain, process the problem and find the result, do the action and solve the problem [12]. On the other case, user experience focuses on the overall experience between user and application. In our project there are some feature that can interact the user. For UX we keep our system vary simple and user friendly and better performance to work [13].

4.4 Implementation Requirements

Implementation is a plan's execution, execution and practice. It also works with a standard policy of method, design and idea to do something. Implementation is the action that must follow the preliminary thinking so that it can actually take place [14].

The development of this project uses few technology and tools. These are: mobile phones Android, Java, Json and Android.

4.4.1 Android

Android is a Google - developed mobile operating system that is used by smartphones. Android's Linux Kernel - based operating system. Android is open source that's why different Android based phones often in different type of graphical use interface. Typically, Android phones come with several built - in apps. It also supports programs from third parties. Developers can easily create Android programs with the free Android developer kit [15] [16]. Android programs are written in java and run on a virtual machine in java. And also run through a smartphone.

CHAPTER 5

Implementation and testing

5.1 Implementation and Database

We used firebase from google to implement the database. Database Firebase Real-time is a cloud-hosted database that supports multiple Android, IOS and web platforms. All data is stored in JSON setup and any data changes reflect a sync through all platforms and devices directly through execution. This enables us to easily build more elastic apps in real time with nominal effort.

Firestore real-time database is a fewer database diagram where the data is stored in JSON format. Mostly, a large JSON tree with many nodes is the entire database. So when planning your database, you need to prepare the JSON construction so that the data is available in a calmer manner by avoiding child node nesting [17].

When it comes to offline data, Firebase provides great support. If there is no internet connection, it automatically stores the data offline. All data will be pushed to the real-time database when the device connects to the internet. Despite the fact that app resumes, qualifying disk determination stores the data offline. You can enable disk tenacity by calling one line code below. Here is a complete offline firebase guide.

You can use `setValue()` on the path of the database situation to insert data. The value on the provided path will be created or updated. In order to store the data, the real-time database accepts multiple string data types, long, double, boolean, map < string, object >, list <object>. You can also use custom java objects to store data that is very helpful when storing model class in the database directly.

To read the data, the `valueEventListener()` must be attached to the reference database. This event will be activated whenever the data changes in real time. You can execute the desired actions on new data in `onDataChange()`.

You can use the same `setValue()` method to update data by passing new value. By passing the path to update data, you can also use `updateChildren()` without disturbing other child node data.

Data deletion. You can simply call to the catalog reference the `removeValue()` method. You can also pass null to the `setValue()` method that performs the same removal operation [18].

5.2 Implementation of front-end Design

There are two parts in an Android app: the front end and the back end. The front end is the visual part of the app with which the user interacts, and the back end, containing all the code driving the app.

Front-End :

1. it's all about what a user able to see.
2. It's may be buttons, images, layout, input fields etc.
3. To design these components in android you need XML and Android programming.

Therefore, these programming are called front-end programming [19].

5.3 implementation of Interactions

Process of manipulating the factors that avoid unified change so that only accepted changes are realized

- Process to review, analyze, and approve requests – timely answer minimizes the opportunity of harmful effect on time, cost, or the possibility of a change.
- Involves managing the changes that are agreed.
- Involves preserving integrity of the presentation extent baselines.
- Involves supporting the entire impact fashioned by the change request.
- Process to review, approve/reject suggested preventive, or corrective actions and Involves organizing changes across the complete project [20].

5.4 Testing Implementation

Testing purpose is to identify faults. Testing is an attempt to detect any possible fault or weakness in a work product. It provides a way of checking the functionality of

mechanisms, subassemblies and/or a finished product, it is the exercise software procedure with the aim of ensuring that the software system meets its supplies and prospects and that doses do not fail improperly [21]. There are different test types. An exact test requirement is addressed by each test type.

5.4.1 Type of Testing

Unit Testing

Unit testing focuses on confirmation effort on the software design module, the smallest unit. The unit testing is concerned with the white box and some modules are performed in parallel [22].

Integration Testing

For each module, testing is performed. After testing all the components, the module is united and the final system is tested using the test data to show that the system will work successfully in all its situations. The system test is therefore an approval that everything is correct and an opportunity to demonstrate to the user that the system works. The purpose of addition testing is to verify functional, performance and reliability requirements placed on major design items [23].

System Testing

System testing ensures compliance with the requirements of the entire united software system. To ensure known and predictable results, it tests a configuration. The configuration-oriented system additional test is an example of system testing. System testing is based on process metaphors and flows, emphasizing pre-driven process

connections and points of incorporation. We have tested unit testing. finally, we have come a successfully conclusion [24].

5.5 Test Results and Reports

Unit Testing Reports

Unit testing: Unit testing is a software test level that tests individual software units / components.

Table 5.1: Test Case for login

SI No. of test case:	1
Name of test:	Login test
Sample Input:	Click on login that runs Login Activity.java
Expected output:	Login GUI should display
Actual output:	Login GUI displayed as expected.

Remarks:	Test Successful
----------	-----------------

Table 5.2 Test Case for Test Home page

SI No. of test case:	2
Name of test:	Click home page test case
Sample Input:	Clicking the Login Button That runs MainActivity.java
Expected output:	Home page details are viewed.
Actual output:	Home page details are viewed.
Remarks:	Test case successful.

Table 5.3 Test Case for Test order confirm

SI No. of test case:	3
Name of test:	Check order System test case.
Sample Input:	Clicking the order Button that runs Confirmation toast.
Expected output:	Confirmation toast and notification are showed
Actual output:	Confirmation toast and notification are showed
Remarks:	Test case successful.

CHAPTER 6

Conclusion and future scope

6.1 Discussion and conclusion

Farming is a very hard working job for farmers. Farmers are the most valuable person for a country, they maintain the food supply of a country. E commerce business just new in Bangladesh. But holds a lot of promise extension. It offers huge scope for farmers in our country. Farmfood is an application for farmers and consumers. E commerce is an area that growing rapidly. So this platform will be the most valuable platform for farmers and consumers. This platform will change the current market situation and the product prize of Bangladesh. Farmfood have great potential of farmers life. It will help farmers to be a businessman. It will help to change the Economic growth of Bangladesh.

6.2 Scope for Future Developments

In future we added invoice system that user gets total cost of their product order. Also we want to add online payment system so that user can easily pay their requirement payment via master card, visa card, bkaash, roket or any other online payment system. In future we will add a comment feature that user will comment a product that is good or bad. Farmers will get his payment directly to his bank account.

REFERENCES

- [1] "Agriculture and Forestry" <<<https://www.alberta.ca/ministry-agriculture-forestry.aspx>>> last accessed on 18-02-2019 at 09.00 am.
- [2] "The Role of Middlemen in Agricultural Marketing" available at <<https://www.academia.edu/1688238/The_Role_of_Middlemen_in_Agricultural_Marketing_Myths_and_Reality>> last accessed on 18-02-2019 at 10.12 am.
- [3] "Farm Direct Marketing" available at <<<https://www.linkedin.com/pulse/f2c-farmer-consumer-farm-direct-marketing-mandeep-pujara>>>last accessed on 18-02-2019 at 11.20 am.
- [4] "Smartphone users on the rise" available at <<<https://www.dhakatribune.com/uncategorized/2016/01/23/smartphone-users-on-the-rise>>>last accessed on 19-02-2019 at 3.20 pm.
- [5] "Comparative Study" available at <<<http://www2.uiah.fi/projekti/metodi/172.htm>>> last accessed on 22-02-2019 at 7.21 pm.
- [6] "Business process Modeling" available at <<<https://kissflow.com/bpm/business-process-modeling/>>> last accessed on 23-02-2019 at 8.31 pm.

- [7] “Software Requirements Analysis” << <https://www.guru99.com/learn-software-requirements-analysis-with-case-study.html> >> last accessed on 23-03-2019 at 9.40 pm
- [8] “Use Case Model” << <http://www.gatherspace.com/use-case-model/> >> last accessed on 23-03-2019 at 10.00 pm
- [9] “Android Definition” <<[https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))>> last accessed on 23-03-2019 at 10.33 pm.
- [10] “Basic about android” << <https://developer.android.com>>> last accessed on 23-03-2019 at 11.03 pm.
- [11] “Know Your Layouts in Android” << <https://www.sitepoint.com/know-your-layouts-in-android/>>> last accessed on 23-03-2019 at 11.25 pm.
- [12] “Interaction Design and UX” << <https://www.interaction-design.org/literature/article/what-is-interaction-design> >> last accessed on 23-03-2019 at 12.25 pm.
- [13] “User Experience (UX) Design” << <https://www.interaction-design.org/literature/topics/ux-design>>> last accessed on 26-03-2019 at 10.25 pm.
- [14] “Implementation” << <https://searchcrm.techtarget.com/definition/implementation>>> last accessed on 26-03-2019 at 11.25 pm.
- [15] “Android tutorial” << <https://www.tutorialspoint.com/android/>>> >> last accessed on 26-03-2019 at 01.25 am.
- [16] “Android SDK”<< https://www.webopedia.com/TERM/A/Android_SDK.html>> last accessed on 27-03-2019 at 05.25 am.
- [17] “Basic firebase” <<<https://angularfirebase.com/lessons/the-ultimate-beginners-guide-to-firebase/>>> last accessed on 28-03-2019 at 03.25 pm.
- [18] “Firebase by platform” << <https://firebase.google.com/docs/>>> last accessed on 28-03-2019 at 03.30 pm.
- [19] “front- end Design” << <https://www.coursereport.com/blog/front-end-development-vs-back-end-development-where-to-start> >> last accessed on 28-03-2019 at 06.30 pm.
- [20] “Implementation of Interactions” << <https://www.pmstudy.com/trainingdocs2/ProjectIntegrationManagement.pdf> >> last accessed on 29-03-2019 at 07.30 pm.
- [21] “Implementation Testing” << https://www.tutorialspoint.com/software_testing_dictionary/implementation_testing.htm>> >> last accessed on 30-03-2019 at 08.30 pm.
- [22] “Unit Testing” <<<https://stackoverflow.com/questions/5975843/implementation-testing-and-unit-testing>>> last accessed on 30-03-2019 at 10.30 pm.

[23] “Integration Testing” << <http://softwaretestingfundamentals.com/integration-testing/>>> last accessed on 01-04-2019 at 07.30 pm.

[24] “System Testing” << <http://softwaretestingfundamentals.com/system-testing/>>> last accessed on 01-04-2019 at 07.30 pm.

APPENDICES

Appendix A :

Here some back-end code for login page

```
private Button loginbtn,registerbtn;  
private EditText loginput_txt, pass_input_txt;  
private FirebaseAuth mAuth;  
private ProgressDialog loadingbar;  
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_login_);  
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
    setSupportActionBar(toolbar);  
    getSupportActionBar().setTitle("লগইন করুন");  
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
    getSupportActionBar().setDisplayShowHomeEnabled(true);  
  
    loginbtn = (Button) findViewById(R.id.loginbtn);  
    registerbtn = (Button) findViewById(R.id.registerbtn);  
    loginput_txt =(EditText) findViewById(R.id.login_emailinput);  
    pass_input_txt=(EditText) findViewById(R.id.login_passinput);  
    mAuth = FirebaseAuth.getInstance();  
    loadingbar = new ProgressDialog(this);
```

```

loginbtn.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        loginAllowuser();

    }

});

registerbtn.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        Intent intent = new Intent(Login_Activity.this, RegisterActivity.class);

        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK
Intent.FLAG_ACTIVITY_CLEAR_TASK);

        startActivity(intent);

        finish();

    }

});

}

@Override

protected void onStart() {

    super.onStart();

    FirebaseUser currentuser = mAuth.getCurrentUser();

    if (currentuser!=null)

    {

        Toast.makeText(Login_Activity.this,"You are Logged
In",Toast.LENGTH_LONG).show();

        sendusertoMainActivity();

```

```

    }
}

private void loginAllowuser() {

    String useemail = loginput_txt.getText().toString();

    String userpass = pass_input_txt.getText().toString();

    if (TextUtils.isEmpty(useemail))

    {

        Toast.makeText(Login_Activity.this,"Please
        Email",Toast.LENGTH_LONG).show();

        Enter

        Your

    }

    else if (TextUtils.isEmpty(userpass))

    {

        Toast.makeText(Login_Activity.this,"Please
        Password",Toast.LENGTH_LONG).show();

        Enter

        Your

    }

    else {

        loadingbar.setTitle("Login Account");

        loadingbar.setMessage("Please wait login your account");

        loadingbar.show();

        loadingbar.setCanceledOnTouchOutside(true);

        mAuth.signInWithEmailAndPassword(useemail,userpass)

        .addOnCompleteListener(new OnCompleteListener<AuthResult>() {

            @Override

            public void onComplete(@NonNull Task<AuthResult> task) {

                if (task.isSuccessful())

                {

```

```

        loadingbar.dismiss();

        Toast.makeText(Login_Activity.this,"Successful
",Toast.LENGTH_LONG).show();

        sendusertoMainActivity();

    }else {

        loadingbar.dismiss();

        String exceptioMessage = task.getException().getMessage();

        Toast.makeText(Login_Activity.this,"Error
exceptioMessage,Toast.LENGTH_LONG).show();
    }

}

});

}

}

}

```

Back-end code for registration form

```
public class RegisterActivity extends AppCompatActivity {  
private Button registerbtn,chooseimage;  
private EditText emailinputtxt,passinputtxt,confirmpassinputtxt,username;  
private FirebaseAuth mAuth;  
private ProgressDialog loadingbar;  
private CircleImageView userimageView;  
private StorageReference storageRef;  
private DatabaseReference userRef;  
private static final int PICK_IMAGE_REQUEST=1;  
private Uri mImageUri, imgeAfterCropUri;  
Uri downloadUri;  
String currentUserId;  
private StorageTask uploadTask;  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
super.onCreate(savedInstanceState);  
setContentview(R.layout.activity_register);  
Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
setSupportActionBar(toolbar);  
getSupportActionBar().setTitle("রেজিস্ট্রেশন করুন");
```

```

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setDisplayShowHomeEnabled(true);
registerbtn =(Button) findViewById(R.id.regibtn);
emailinputtxt =(EditText) findViewById(R.id.emailinput);
passinputtxt =(EditText) findViewById(R.id.passinput);
confirmpassinputtxt =(EditText) findViewById(R.id.confirmpassinput);
userimageView =(CircleImageView) findViewById(R.id.userImage);
chooseimage =(Button) findViewById(R.id.choseimageBtn);
username =(EditText) findViewById(R.id.nameinput);
loadingbar = new ProgressDialog(this);
storageRef = FirebaseStorage.getInstance().getReference().child("UserImage");
 mAuth=FirebaseAuth.getInstance();
userRef = FirebaseDatabase.getInstance().getReference().child("Users");
registerbtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        createnewAccount();
    }
});
chooseimage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        filechooserImage();
    }
});

```



```

}

private void filechooserImage() {

Intent intent = new Intent();

intent.setType("image/*");

intent.setAction(Intent.ACTION_GET_CONTENT);

startActivityForResult(intent,PICK_IMAGE_REQUEST);

}

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode==PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data
!=null && data.getData() != null)

{

mImageUri = data.getData();

Picasso.with(this)

.load(mImageUri)

.fit()

.placeholder(R.drawable.user)

.into(userimageView);

}

}

private void putimagetostorage()

{

final StorageReference filepath = storageRef.child(System.currentTimeMillis() + ".jpg");

uploadTask = filepath.putFile(mImageUri);

```

```

        uploadTask.continueWithTask(new Continuation<UploadTask.TaskSnapshot,
Task<Uri>>() {

            @Override

            public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws
Exception {

                if (!task.isSuccessful()) {

                    throw task.getException();

                }

                // Continue with the task to get the download URL

                return filepath.getDownloadUrl();

            }

        }).addOnCompleteListener(new OnCompleteListener<Uri>() {

            @Override

            public void onComplete(@NonNull Task<Uri> task) {

                if (task.isSuccessful()) {

                    downloadUri = task.getResult();

                    saveAccountInfoinDatabase();

                }

                else {

                    // Handle failures

                    // ...

                }

            }

        });

    }

    @Override

```

```

protected void onStart() {
    super.onStart();
    FirebaseAuth currentUser = mAuth.getCurrentUser();
    if (currentUser!=null)
    {
        sendusertoMainActivity();
    }
}

private void createNewAccount() {
    String userEmail = emailinputtxt.getText().toString();
    String userpass = passinputtxt.getText().toString();
    String usercnfirmpass=confirmpassinputtxt.getText().toString();
    if (mImageUri==null)
    {
        Toast.makeText(this,"Please Enter Your Profile
Image",Toast.LENGTH_LONG).show();
    }
    else if (TextUtils.isEmpty(useremail))
    {
        Toast.makeText(this,"Please Enter Your Email",Toast.LENGTH_LONG).show();
    }else if (TextUtils.isEmpty(userpass))
    {
        Toast.makeText(this,"Please Enter Your Password",Toast.LENGTH_LONG).show();
    } else if (TextUtils.isEmpty(usercnfirmpass))
    {

```

```

        Toast.makeText(this, "Please Enter Your Password
Again", Toast.LENGTH_LONG).show();
    } else if (!userpass.equals(usercnfirmpass))
    {
        Toast.makeText(this, "Please retype password", Toast.LENGTH_LONG).show();
    }
    else {
        loadingbar.setTitle("Creating Account");
        loadingbar.setMessage("Please wait your account is creating");
        loadingbar.show();
        loadingbar.setCanceledOnTouchOutside(true);
        mAuth.createUserWithEmailAndPassword(useremail, userpass)
            .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful())
                    {
                        Toast.makeText(RegisterActivity.this, "Congratulation" +
                            " Successfully Registered", Toast.LENGTH_LONG).show();
                        putimagetostorage();
                    }
                    else {
                        loadingbar.dismiss();
                        String erroremassage = task.getException().getMessage();
                        Toast.makeText(RegisterActivity.this, "Error "+erroremassage,

```

```

        Toast.LENGTH_LONG).show();
    }
}

});
}
}

private void saveAccountInfoinDatabase() {
String userName = username.getText().toString();

if (TextUtils.isEmpty(userName))
{
    Toast.makeText(RegisterActivity.this, "Please Enter Your
Name", Toast.LENGTH_LONG).show();
}

else {

    loadingbar.setTitle("Profile Setup");

    loadingbar.setMessage("Please Wait Saving Your Profile ... ");

    loadingbar.show();

    loadingbar.setCanceledOnTouchOutside(true);

    HashMap userMap = new HashMap();

    userMap.put("username", userName);

    userMap.put("profileimageurl", downloadUri.toString());

    currentUserId = mAuth.getCurrentUser().getUid();

    userRef.child(currentUserId).updateChildren(userMap).addOnCompleteListener(new
OnCompleteListener() {

        @Override

        public void onComplete(@NonNull Task task) {

```

```

if (task.isSuccessful())
    {
        loadingbar.dismiss();
        sendusertoMainActivity();
        Toast.makeText(RegisterActivity.this,"Profile
Successful",Toast.LENGTH_LONG).show();
    }
else {
        loadingbar.dismiss();

        String exceptioMessage = task.getException().getMessage();

        Toast.makeText(RegisterActivity.this,"Error
exceptioMessage,Toast.LENGTH_LONG).show();
    }
}
});
}

private void sendusertoMainActivity() {
    Intent intent = new Intent(this,MainActivity.class);

    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK
Intent.FLAG_ACTIVITY_CLEAR_TASK);

    startActivity(intent);
finish();
}

```

Farm

by
Humayun Kabir

Submission date: 03- Apr- 2019 05:56PM (UT C+0600)

Submission ID: 1105095409

File name: farmfood_Report.docx (4.8M)

Word count : 3137

Character count : 16134

Farm

ORIGINALITY REPORT

1 %	%	1 %	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

- 1** Jarle Hansen, Tor-Morten Gronli, Gheorghita Ghinea. "Cloud to Device Push Messaging on Android: A Case Study", 2012 26th International Conference on Advanced Information Networking and Applications Workshops, 2012
Publication <1 %
- 2** Jane L. Pearson, Yeates Conwell. "Suicide in Late Life: Challenges and Opportunities for Research", International Psychogeriatrics, 2005
Publication <1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off