SOFTWARE ENGINEERING – 1

CSE 5324 (FALL - 2020)

AGRO FARM

TEAM 2 - AGRO CHARGERS

Mujahed Khaled Sheikh
Akshay Lingamaneni
Sai Sumanth Boda
Rashmitha Gaddam
Cheruku Laxmi Shalini

TABLE OF CONTENTS

S.NO	CONTENT	Page Number
1.	PROJECT DESCRIPTION	3-4
2.	REQUIREMENTS	5-6
3.	High-Level Use Cases	7
4.	USE CASE DIAGRAMS	8-9
5.	USE CASE TRACEABILITY MATRIX	10
6.	Domain Model	11
7.	Expanded use cases	12-15
8.	Expanded Use case UI Screenshots	16-24
9.	Sequence diagram	25-29
10.	Design class diagram	30
11.	Android Screenshots	31-35
12.	Increment Matrix	36
13.	TASK ASSIGNED LIST	37-38
14	Video Link	**39**

PROJECT DESCRIPTION

1. Agriculture has remained the backbone of the Indian Economy, though it is the main source of livelihood for most of the Indian population
2. little has been done to revive the sector. From production challenges to financing inefficiencies, Indian agriculture is plagued by several issues.
3. Some of the challenges faced by our farmers are Inadequate farm equipment,
4. lack of access to fair prices, Farmers are the sole risk bearers of all these challenges that arise in the farming cycle
5. The middlemen usually come into the picture in Big Farms which we are focusing on, Agro Farm app is the approach to address one of the above issues of distribution challenges faced by farmers due to the intervention of the middlemen
6. Farmers are victims to information asymmetries which means that middlemen control access to markets and thereby farmers get low prices as they have no choice but to see their good to the middlemen
7. Our goal is to assist farmers and provide a platform to make the right choices by analyzing demand, pricing while also providing tools to train themselves via one-click lectures
8. The primary goal of this app is to provide the information, communication which is the main backlog in Indian farming, to the farmers about the price range they need to sellto gain max profit
9. To have greater visibility for the consumers, we are not focusing on the delivery part which includes the purchase, delivery
10. At this stage of the app, we want to provide information, communication which can help them directly to the right retailers
11. This is for big farms(meaning we have large fields that produce, and the farmers sell those many times through middlemen)
12. Amount plans to mention it as a daily market price range, from which a farmer can select
13. This app would directly connect the farmers with consumers, by contacting them directly through a call, as we will be having a calling option, once they click the name
14. The farmers will be able to upload the images, keeping a verified badge, App requires the internet access, for this stage, we are not considering the location.
15. The app will contain features like –
16. <u>Login/Registration:</u> The opening screen would consist of two tabs Farmers and customers can create an account in this app by filling the registration form and login to the app through the Login Page using their credentials.
17. <u>Farmers Information:</u> On clicking the farmer's section tab the farmer will be able to enter his details like name, phone number, quantity.
18. <u>Veggies Upload:</u> Once the farmers click the farmer's tab, in the Veggies upload section the Farmers can upload details in a form with the information of the available veggies, their description, and pricing.

19. Customer Purchases: Customers can view the products available from various farmers, select the required one, and can make a call to the respective farmer.
20. Training: In this section, we curate videos to educate farmers about the latest scientific advances and technology.
21. News Section: The news section can help farmers and customers in being up to date with information about agriculture.
22. Review Section: The users can add the review about their experience with the product so that the system becomes more transparent and helpful for other users after they have communicated with the farmer, this will give an idea of whether to deal with that particular farmer or not
23. Farmers Forum: The farmers may have questions related to production or any issues they might need advice on now, this forum can help the farmers in coordinating to help each other.
24. About us Section: The about us section is to build to give the information about the app and to give the users what is expected from the app

RESOURCES: Databases, Internet

TEAM EXPERIENCE WITH ANDROID:

MUJAHED: I am proficient in java and its core concepts, knowledge about the object-oriented programming and design frameworks in android, and developed apps working in a team using android, experience in the integration of API's in android, working with the backend They are two apps that were developed in my under graduation

LAXMI SHALINI: Previously I have built basic android applications like chat messenger during my undergrad. I have a good command of java, and I have worked on java applications and build a web app for caterer management system, Also I am good with Object-oriented design concepts which help to understand the use of different APIs and its implementation strategies.

SAI SUMANTH: My Knowledge about Android Studio and Building Android based Applications is from my Undergraduate Project where we made a Humanoid Robot. That is when I came across Android Studio and build my First Android App to Control our Arduino Based Humanoid Robot from the App using A Bluetooth Module. I have Ample command over the programming languages such as Java, C++ which I believe are Useful while working on Android.

RASHMITHA: I have 2 years of work experience in JAVA. I have a good understanding of core JAVA, object-oriented programming concepts, and MVC structure. I am naïve to Android Development but understand the software process that goes into building an app.

AKSHAY: I worked as a JAVA Developer for 2 years. I am fond of building Front-End applications but never built a mobile app. This experience would be challenging for 2 reasons. I would get to learn mobile application development and build the app using the right methodologies.

REQUIREMENTS

Req ID	Requirement	
R1	Application Shall initialize as follows	
R1.1	Application Shall have an Opening screen displaying Farmers, Customer Tab	16
R1.2	Application shall provide signup facility for user with email id, password and role	16
R1.3	Application shall show a message of 'created new farmer/customer' when the signup is completed.	Derived
R1.4	Application shall provide login facility for both farmer and customer with username and password.	16
R1.5	Application shall redirect to the Customers tab after the login is successful by the customer	Derived
R1.6	Application shall redirect to the Farmers tab after the login is successful by the farmer.	Derived
R2	Application shall allow the user to redirect to the farmers section page after clicking the farmers tab	17
R2.1	Application shall allow the farmers to enter his personal information on clicking the 'Farmers Information' tab	17
R2.1.1	Application shall show the input to enter the name, phone, quantity	17
R2.1.2	Application shall show the error message "Enter Name", if not entered	Derived
R2.1.3	Application shall show the error message "Enter Phone", if not entered.	Derived
R2.1.4	Application shall show the error message "Enter Quantity", if not entered	Derived
R2.2	Application shall show the Veggies upload section which will be achieved once the tab of Veggies Upload is clicked	18
R2.2.1	Application shall show the input to enter the Veggies name, description, pricing	18
R2.2.2	Application shall show the error message "Enter Veggies Name", if not entered	Derived
R2.2.3	Application shall show the error message "Enter description", if not entered	Derived

R2.2.4	Application shall show the error message “Enter Price”, if not entered	Derived
R2.3	Application shall show the Training Section which will be achieved once the tab of Training Section is clicked	20
R2.4	Application shall show the Farmers Forum Section which will be achieved once the tab of Farmers Forum tab is clicked	23
R2.4.1	Application shall improve communication and increase collaboration among farmers in the Farmer Forum section.	23
R3	Application shall show the Customer Purchases section which will be achieved once the tab of Customer Purchases is clicked	19
R3.1	Application shall allow customers to view the details of farmers and their uploaded veggies information.	19
R3.2	Application shall allow customers to give their feedback/review about the farmers products.	22
R4	Application shall show the News Section which will be achieved once the tab of News Section is clicked	21
R4.1	Application shall display the market pricing, news articles related to farming in the News Section	21
R5	Application shall show a button named ‘About Us’ at the bottom of the page	24
R5.1	Application shall show steps on how to use the App and its features	24

High-Level Use Cases

UC 1: Register for Customer

TUCBW the customer clicks on the register button on the login/signup page.

- TUCEW the customer gets the success message as ‘created successfully’ andthe ‘Home Page’ is displayed.

UC 2: Login for Customer

- TUCBW the customer clicks on the login button on the login/signup page of customers’ tab.
- TUCEW the customer gets signed in and the ‘Home Page’ screen is displayed.

UC 3: Login for Farmer

- TUCBW the Farmer clicks on the login button on the login/signup page of Farmers tab.
- TUCEW the Farmer gets signed in and the ‘Home Page’ screen is displayed.

UC 4: View News Section

- TUCBW the farmer/customer clicking on the ‘View News’ section button on the Home Page.
- TUCEW the famer/customer can view news feed.

UC 5: View Training Section

- TUCBW the Farmer clicks on the ‘Training’ button on the homepage.
- TUCEW the Farmer can view latest scientific advances in technology.

UC 6: View Customer Purchase Section

- TUCBW the Customer clicks on the ‘View Customers Purchase’ button in the customer’s Tab
- TUCEW the Customer can view the details of farmers, veggies information

UC 7: Veggies Upload

- TUCBW the Farmer clicks on the ‘Veggies Upload’ button on the Farmer’s Tab.
- TUCEW the Farmer can directly upload veggies information.

UC 8: View Review Section

- TUCBW the Customer clicks on the ‘Review Section’ button.
- TUCEW the Customer can provide feedback.

UC 9: View Farmers Forum

- TUCBW the farmer clicks on the ‘Farmers Forum’ button.
- TUCEW the farmers can raise the questions.

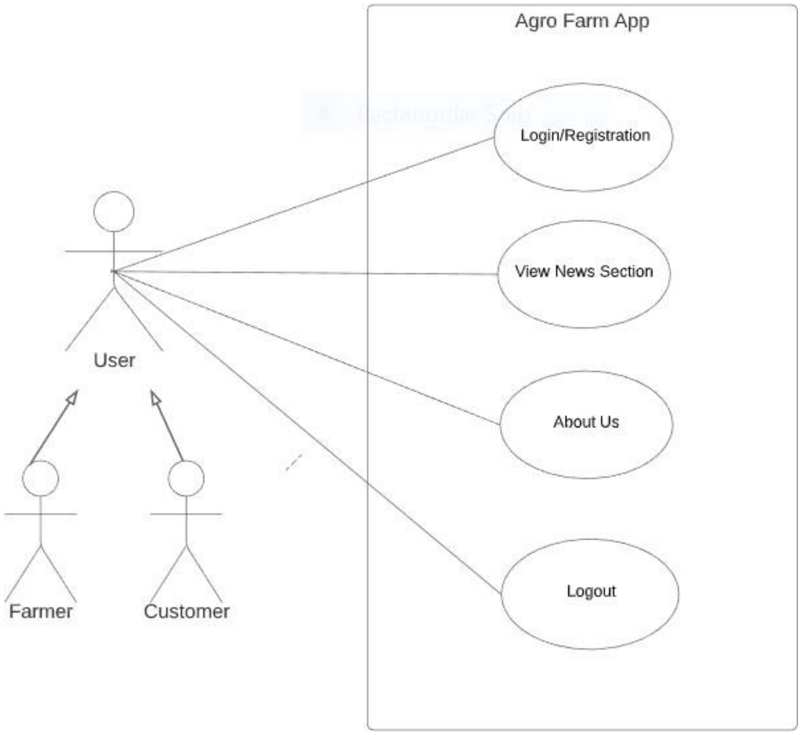
UC 10: View Farmers Information

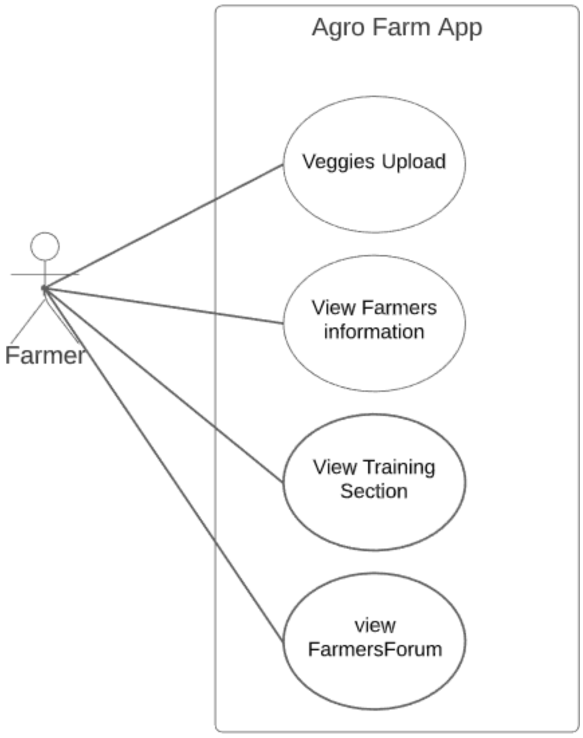
- TUCBW the Farmer clicks on the ‘Framers Information’ button.
- TUCEW the farmer can enter personal details.

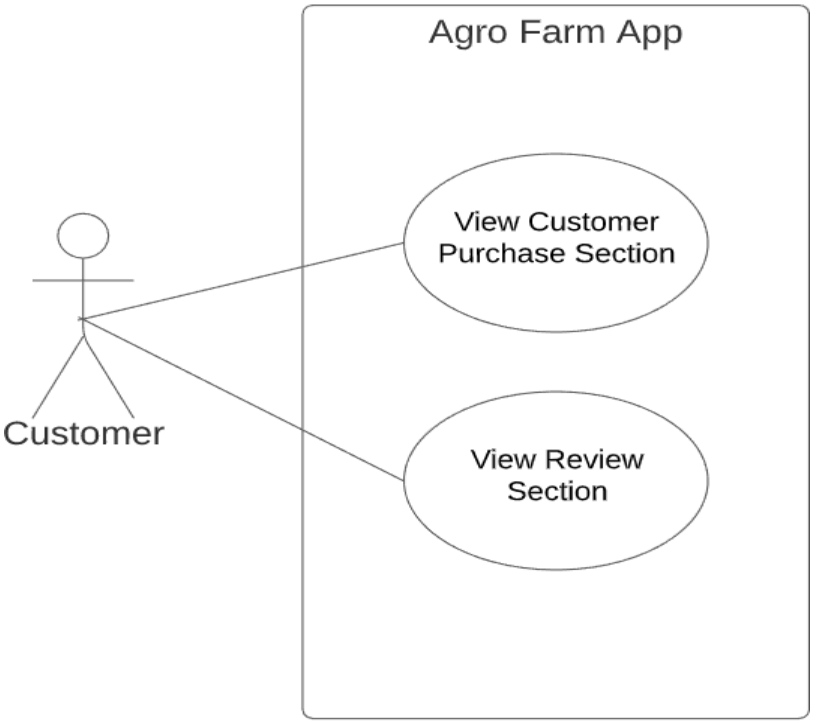
UC 11: Show About Us

- TUCBW the Farmer, Customer when clicks on ‘About Us’ button.
- TUCEW the Farmer, Customer can know about the app.

USE CASE DIAGRAMS:




Agro Farm App
Farmer
Veggies Upload
View Farmers information
View Training Section
view FarmersForum

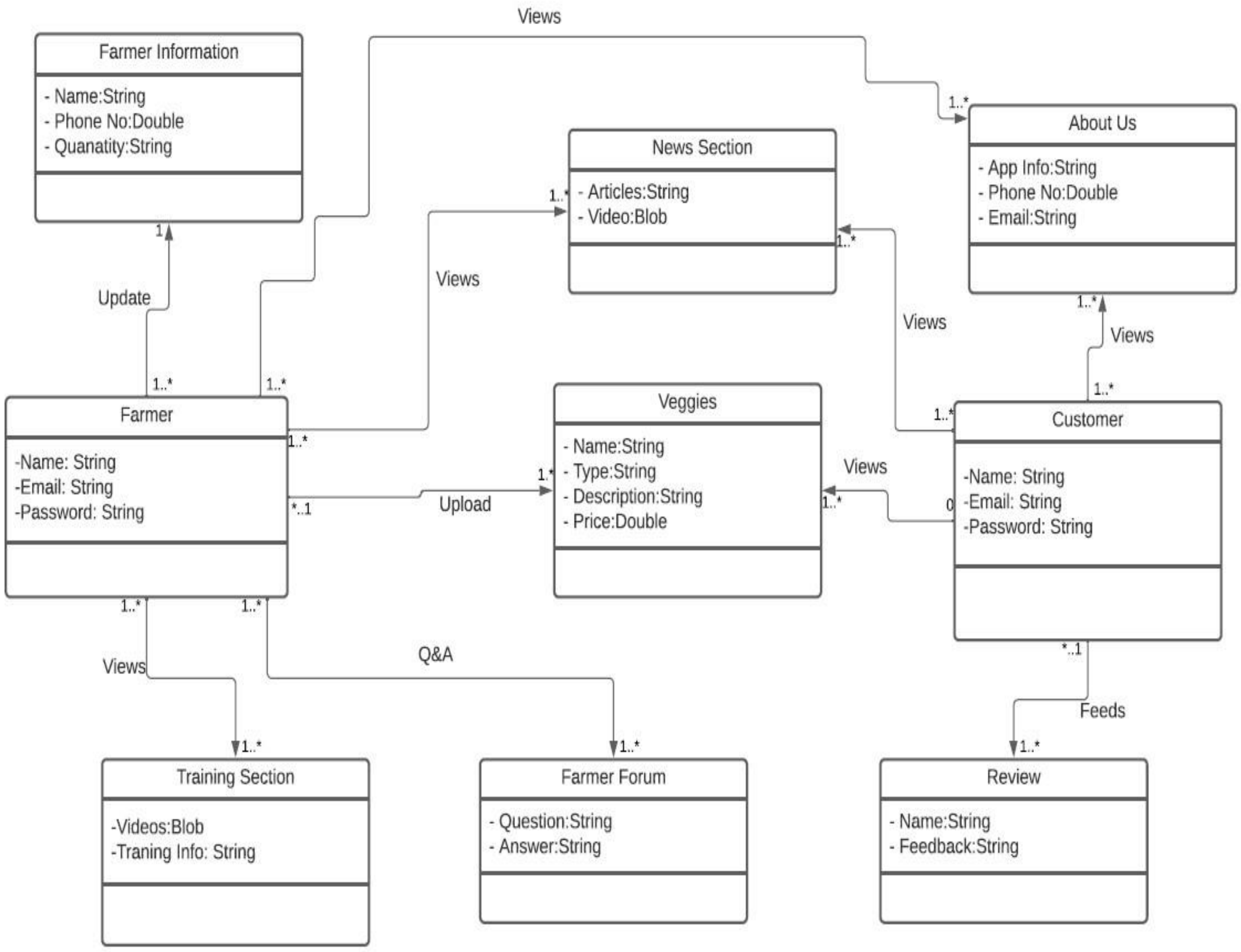

Agro Farm App
Customer
View Customer Purchase Section
View Review Section

USE CASE TRACEABILITY MATRIX

	Priority Weight	UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7	UC 8	UC 9	U1 0	U1 1
R1	1	X										
R1.1	1	X										
R1.2	1	X										
R1.3	1	X										
R1.4	1	X										
R1.5	1		X									
R1.6	1			X								
R2	4										X	
R2.1	2										X	
R2.1.1	2										X	
R2.1.2	2										X	
R2.1.3	2										X	
R2.1.4	2										X	
R2.2	3							X				
R2.2.1	3							X				
R2.2.2	3							X				
R2.2.3	3							X				
R2.2.4	3							X				
R2.3	3					X						
R2.4	2									X		
R2.4.1	2									X		
R3	3						X					
R3.1	2						X					
R3.2	3								X			
R4	4				X							
R4.1	4				X							
R5	5											X
R5.1	5											X
	Score	5	1	1	8	3	5	15	3	4	14	10

Note: Highest Priority = 1

DOMAIN MODEL:



Expanded Use Cases:

UC1: Register For Farmer/Customer	
Precondition: This use case assumes that the customer and farmer has an authenticated Email-Id	
Actor: Farmer/Customer	System: AgroFarm
	(0) System displays the login/register page
(1) TUCBW Farmer/Customer clicks on the register button on the page	(2) System displays the form fields, name, email_id, password, role and sign up button
(3) Farmer/Customer fills out the form details in the displayed page by specifying the respected info by clicking on the submit button	(4) * System save the submitted data and updates in the database
(5) TUCEW Farmer/Customer gets the success message and redirects to the respected Login Page.	
Postcondition: Farmer/Customer successfully register in the app	
* Indicates Non-Trivial Steps	

UC2: Login for farmer	
Precondition: This use case assumes that the farmer has a valid email_id and password	
Actor: Farmer	System: ArgoFarm
	(0) System displays the login/register page (check weather login/register or just login)
(1) TUCBW Farmer clicks on the Login button on the page	(2) System displays the form fields, email_id, password
(3) Farmer fills out the form details in the displayed page by specifying the respected info by clicking on the submit button	(4) * System save the submitted data and updates in the database
(5) TUCEW Farmer gets the success message and redirects to the landing page of famer	
Postcondition: Farmer successfully logs in to the app	
* Indicates Non-Trivial Steps	

UC3: Login for Customer	
Precondition: This use case assumes that the Customer has a valid email_id and password	
Actor: Customer	System: ArgoFarm
	(0) System displays the login/register page(check weather login/register or just login)
(1) TUCBW Customer clicks on the Login button on the page	(2) System displays the form fields, email_id, password
(3) Customer fills out the form details in the displayed page by specifying the respected info by clicking on the submit button	(4) * System save the submitted data and updates in the database
(5) TUCEW Customer gets the success message and redirects to the landing page of famer	
Postcondition: Customer successfully logs in to the app	
* Indicates Non-Trivial Steps	

UC4: View News Section	
Precondition: This use case assumes that the farmer/Customer is logged in the system	
Actor: Farmer/Customer	System: ArgoFarm
	(0) System displays the home page of Farmer/Customer (Refer Figure 1)
(1) TUCBW Farmer/Customer clicks on the View News Section button on the home page	(2) *System processes the request and then displays the news feed and latest articles related to farming
(3) TUCEW Farmer/Customer can view the news feed.(Refer Figure 2)	
Postcondition: Farmer/Customer can view the news feed.	
* Indicates Non-Trivial Steps	

UC5: View Training Section	
Precondition: This use case assumes that the farmer is logged in the system	
Actor: Farmer	System: ArgoFarm
	(0) System displays the home page of Farmer
(1) TUCBW Farmer training button on the home page	(2) * System processes the request and then displays available trainings to the farmer
(3) TUCEW Farmer being able to view the latest scientific advances in technology, useful farming techniques and able to access these trainings(Refer Figure 3)	
Postcondition: Farmer will be able to view the training details available in the training section	
* Indicates Non-Trivial Steps	

UC 6: View Customer Purchase Section	
Precondition: This Use Case Assumes that the Customer logs into the System	
Actor: Customer	**System:** Agro Farm
	(0) System Displays the Homepage of Customer
(1) **TUCBW** The Customer clicks on the "View Customer Purchase" button in the Customer Tab	(2) *System processes the requirements and Displays the Veggies and the Farmer Info
(3) **TUCEW** The Customer can view the Details of the Farmer, Veggies Information. (Refer Figure 4)	
Postcondition: Farmer will be able to view the training details available in the training section	
* Indicates Non-Trivial Steps	

UC 7: Veggies Upload	
Precondition: This Use Case Assumes that the Farmer logs into the System	
Actor: Farmer	**System:** Agro Farm
	(0) System Displays the Homepage of Farmer.
(1) (1) **TUCBW** The Farmer clicks on the "Veggies Upload" button on the Farmer's Tab	(2) * System processes the requests and Displays the Blank Fields of Available Veggies, Name, Description, Pricing.
(3) **TUCEW** The Farmer Uploads Veggies Information.	
Postcondition: The Farmer will be Uploading the Veggies Info.	
* Indicates Non-Trivial Steps	

UC 8: View Review Section	
Precondition: This Use Case Assumes that the Customer logs into the System	
Actor: Customer	**System:** Agro Farm
	(0) System Displays the Homepage of Customer
(1) **TUCBW** The Customer clicks on the "Review Section" button on the Customer's Page	(2) * System processes the requests and Displays a Section to give their Feed Back.
(3) **TUCEW** The Customer can Provide the Feedback, Experience in Buying the Product which will help in Transparency.	
Postcondition: The Customer will be able to give Feedback	
* Indicates Non-Trivial Steps	

<table>
<tr><th colspan="2">UC 9: View Farmers Forum</th></tr>
<tr><td colspan="2"><b>Precondition:</b> This Use Case Assumes that the Farmer logs into the System</td></tr>
<tr><td><b>Actor:</b> Farmer</td><td><b>System:</b> Agro Farm</td></tr>
<tr><td></td><td>(0) System Displays the Homepage of Farmer.</td></tr>
<tr><td>(1) <b>TUCBW</b> The Farmer clicks on the “Farmer Forum” button on the Farmer’s page.</td><td>(2) * System processes the requests and Provides a Section to Q & A.</td></tr>
<tr><td>(3) <b>TUCEW</b> The Farmers can Raise the Questions, get and Advice related to their Farming Issues.</td><td></td></tr>
<tr><td colspan="2"><b>Postcondition:</b> The Farmer will be able to post their queries and get Answers</td></tr>
<tr><td colspan="2">* Indicates Non-Trivial Steps</td></tr>
</table>

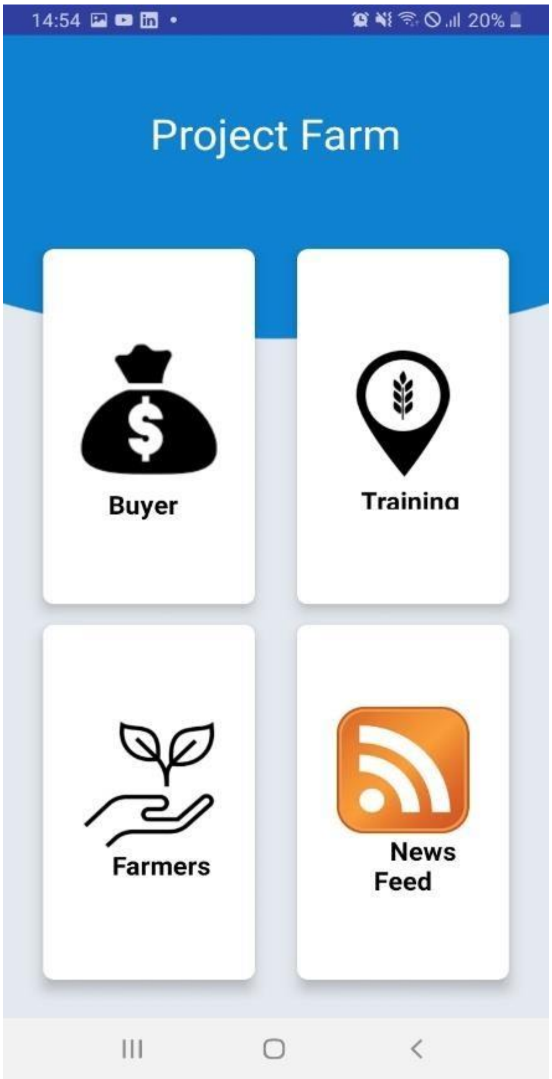
<table>
<tr><th colspan="2">UC 10: View Farmers Information</th></tr>
<tr><td colspan="2"><b>Precondition:</b> This Use Case Assumes that the Farmer logs into the System</td></tr>
<tr><td><b>Actor:</b> Farmer</td><td><b>System:</b> Agro Farm</td></tr>
<tr><td></td><td>(0) System Displays the Homepage of Farmer.</td></tr>
<tr><td>(1) <b>TUCBW</b> The Farmer clicks on the “Farmers Information” button on the Farmer’s Tab.</td><td>(2) * System processes the requests and Displays the Blank Fields of Enter Name, Phone Number, Quantity.</td></tr>
<tr><td>(3) <b>TUCEW</b> The Farmer can enter his Personal Details in the Respective Fields.</td><td>(4) System validates the Information and Updates it.</td></tr>
<tr><td colspan="2"><b>Postcondition:</b> The Farmer will be able to enter their Personal Details</td></tr>
<tr><td colspan="2">* Indicates Non-Trivial Steps</td></tr>
</table>

<table>
<tr><th colspan="2">UC 11: Show About Us</th></tr>
<tr><td colspan="2"><b>Precondition:</b> This Use Case Assumes that the Farmer/Customer logs into the System</td></tr>
<tr><td><b>Actor:</b> Farmer/Customer</td><td><b>System:</b> Agro Farm</td></tr>
<tr><td></td><td>(0) System Displays the Homepage.</td></tr>
<tr><td>(1) <b>TUCBW</b> The Farmer/Customer clicks on the “About Us” button.</td><td>(2) * System processes the requests and Displays the Info Related to the App.</td></tr>
<tr><td>(3) <b>TUCEW</b> The Farmer/Customer can View the Info Related to the App.</td><td></td></tr>
<tr><td colspan="2"><b>Postcondition:</b> The Farmer/Customer can Learn about the App</td></tr>
<tr><td colspan="2">* Indicates Non-Trivial Steps</td></tr>
</table>

Expanded Use case UI Prototypes:

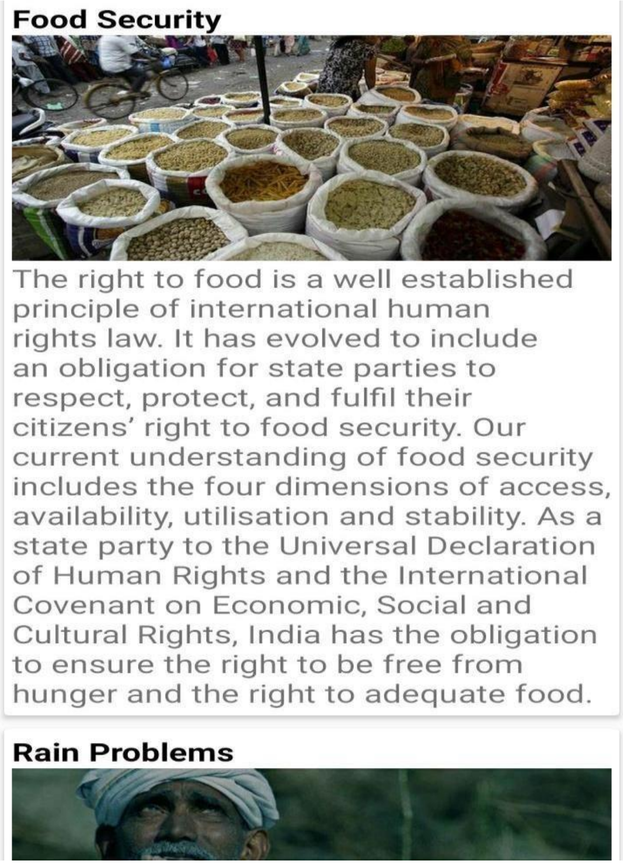
HOME PAGE:

Figure 1:



NEWS FEED:

Figure 2:



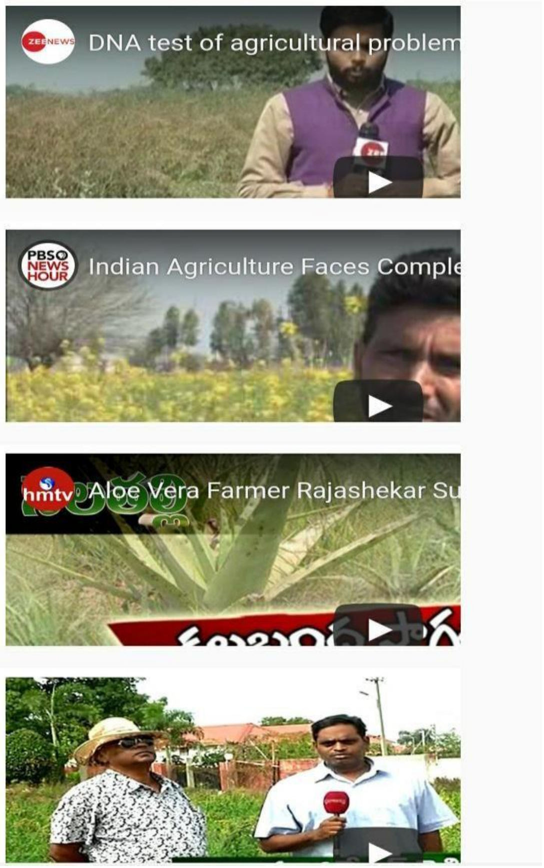
Food Security

The right to food is a well established principle of international human rights law. It has evolved to include an obligation for state parties to respect, protect, and fulfil their citizens' right to food security. Our current understanding of food security includes the four dimensions of access, availability, utilisation and stability. As a state party to the Universal Declaration of Human Rights and the International Covenant on Economic, Social and Cultural Rights, India has the obligation to ensure the right to be free from hunger and the right to adequate food.

Rain Problems

TRAINING SECTION:

Figure 3:



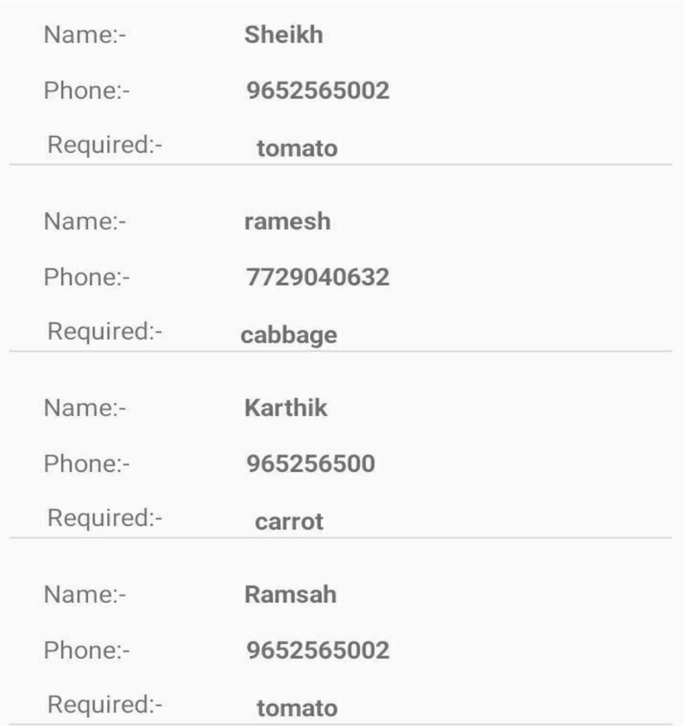
DNA test of agricultural problem

Indian Agriculture Faces Comple

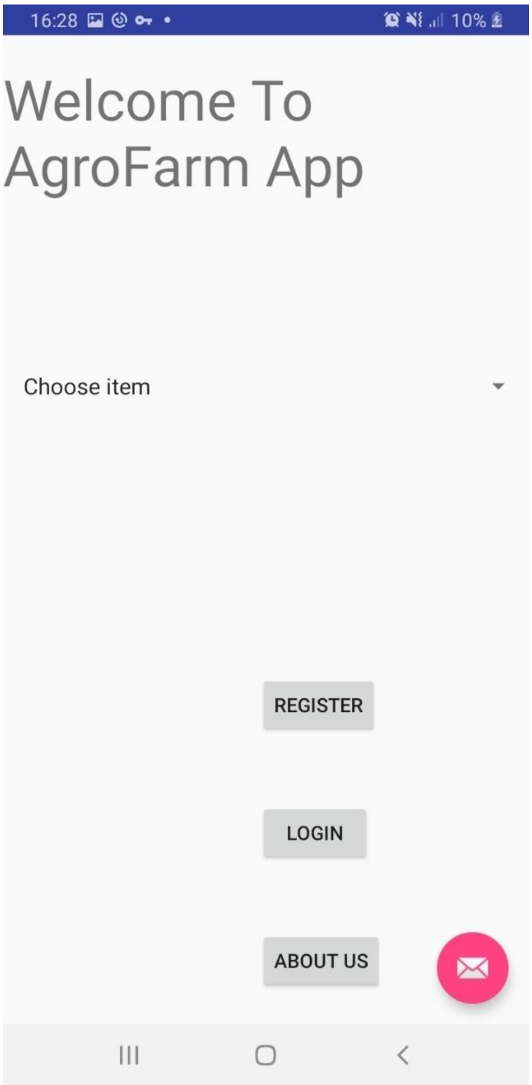
Aloe Vera Farmer Rajashekar Su

BUYERS SECTION:

Figure 4:



Name:-	Sheikh
Phone:-	9652565002
Required:-	tomato
Name:-	ramesh
Phone:-	7729040632
Required:-	cabbage
Name:-	Karthik
Phone:-	965256500
Required:-	carrot
Name:-	Ramsah
Phone:-	9652565002
Required:-	tomato

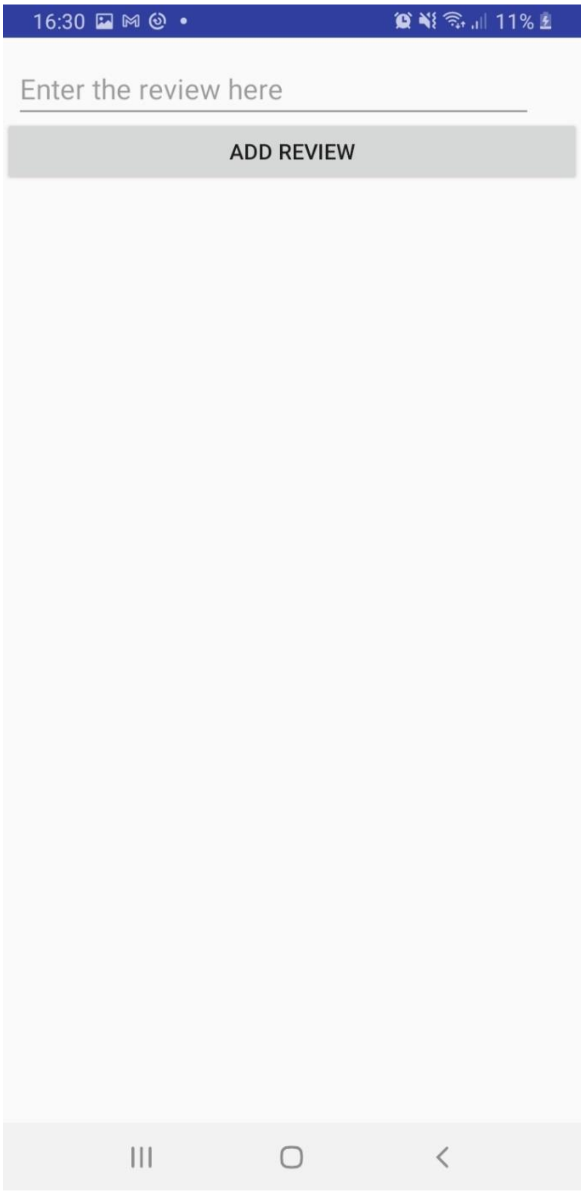


16:28 10%

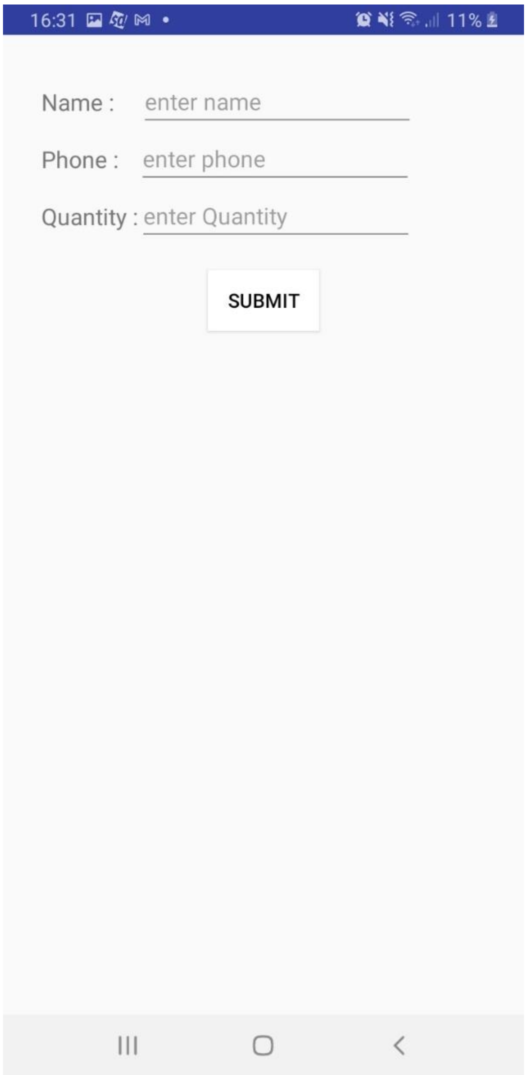
Welcome To AgroFarm App

Choose item

Review Page:



FARMERS INFO PAGE:



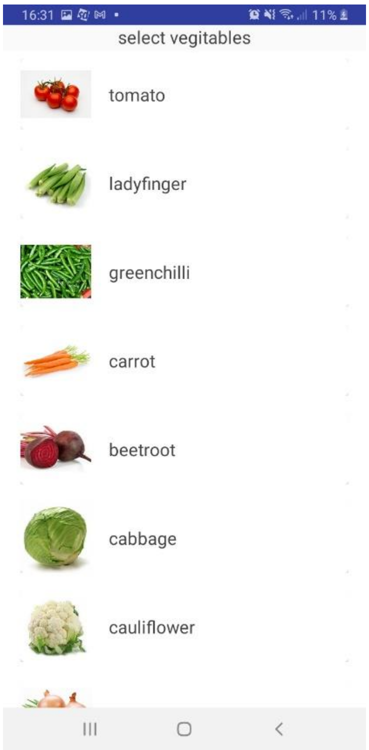
16:31 11%

Name : enter name

Phone : enter phone

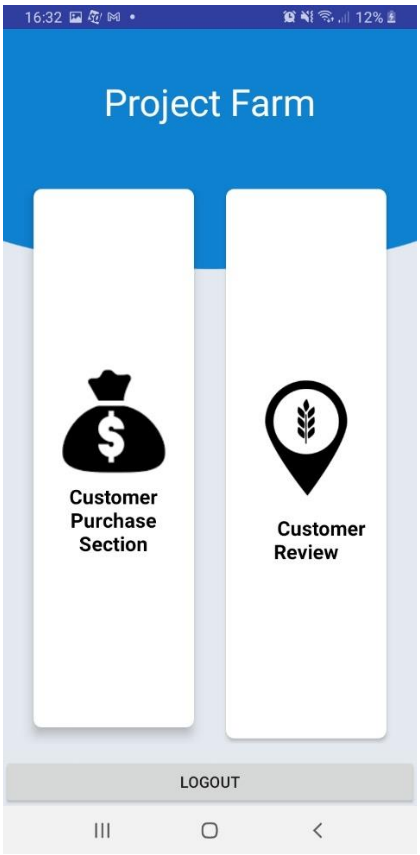
Quantity : enter Quantity

Selecting Veggies:

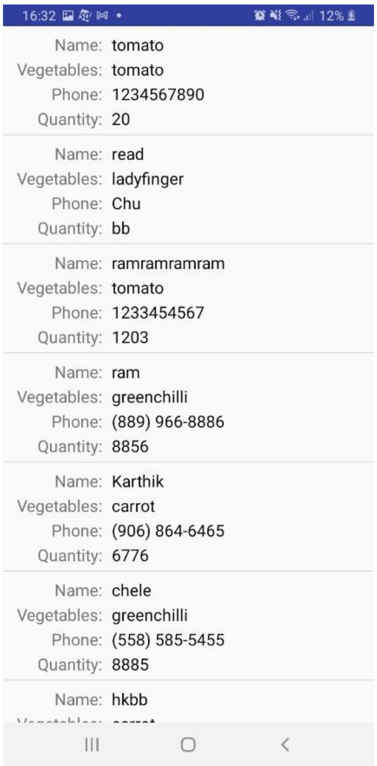

16:31 11%

select vegitables

Customer Home Page:


16:32 12%

Farmers &Veggies Information:



Name: tomato
Vegetables: tomato
Phone: 1234567890
Quantity: 20

Name: read
Vegetables: ladyfinger
Phone: Chu
Quantity: bb

Name: ramramramram
Vegetables: tomato
Phone: 1233454567
Quantity: 1203

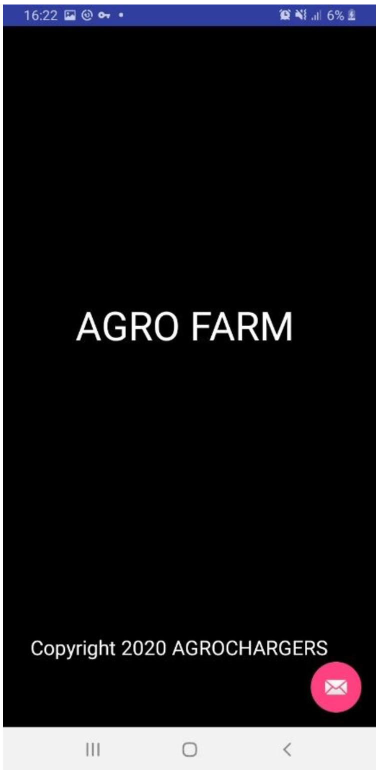
Name: ram
Vegetables: greenchilli
Phone: (889) 966-8886
Quantity: 8856

Name: Karthik
Vegetables: carrot
Phone: (906) 864-6465
Quantity: 6776

Name: chele
Vegetables: greenchilli
Phone: (558) 585-5455
Quantity: 8885

Name: hkbb

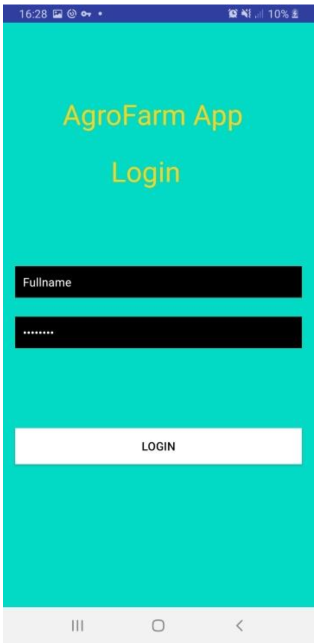
Splash Screen:



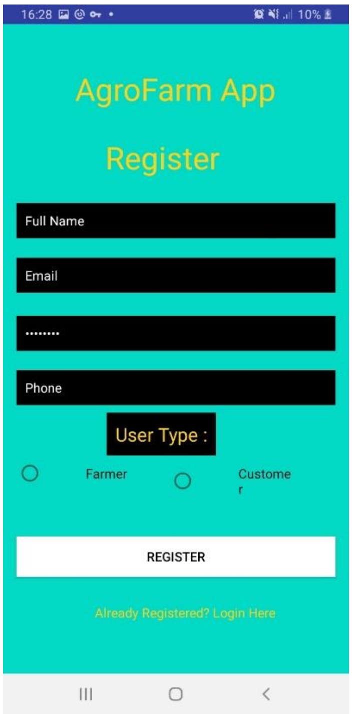
AGRO FARM

Copyright 2020 AGROCHARGERS

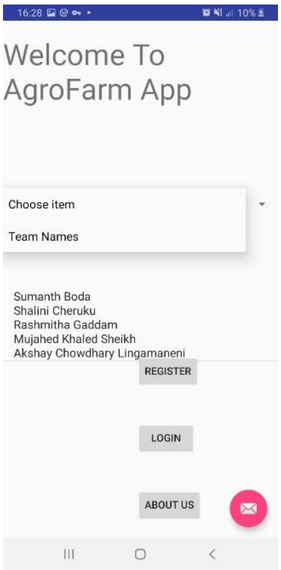
Login Page:



Register Page:



Displaying Team Names:



16:28

Welcome To AgroFarm App

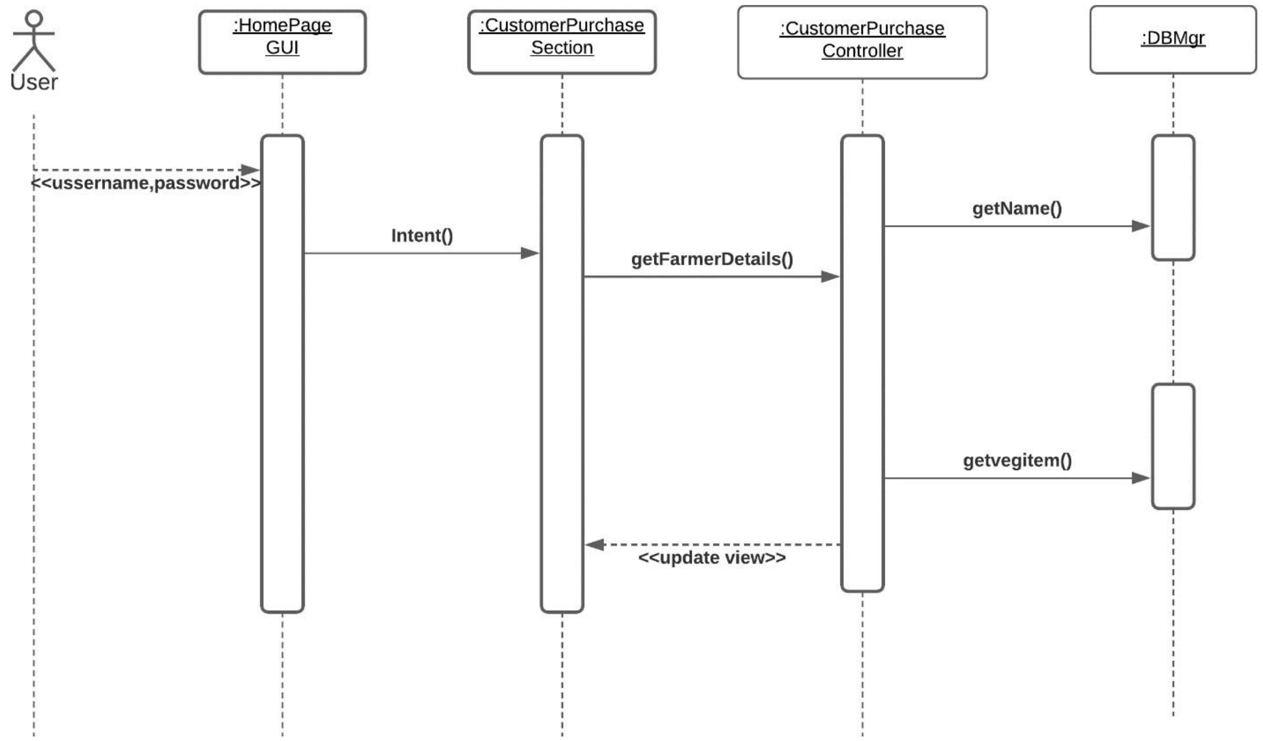
Choose item

Team Names

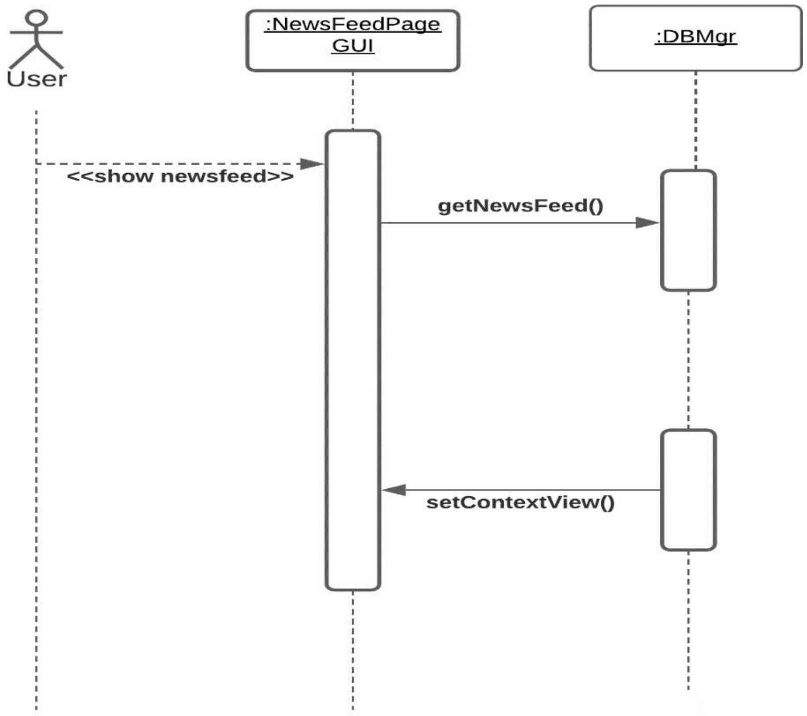
Sumanth Boda
Shalini Cheruku
Rashmitha Gaddam
Mujahed Khaled Sheikh
Akshay Chowdhary Lingamaneni

Sequence diagrams:

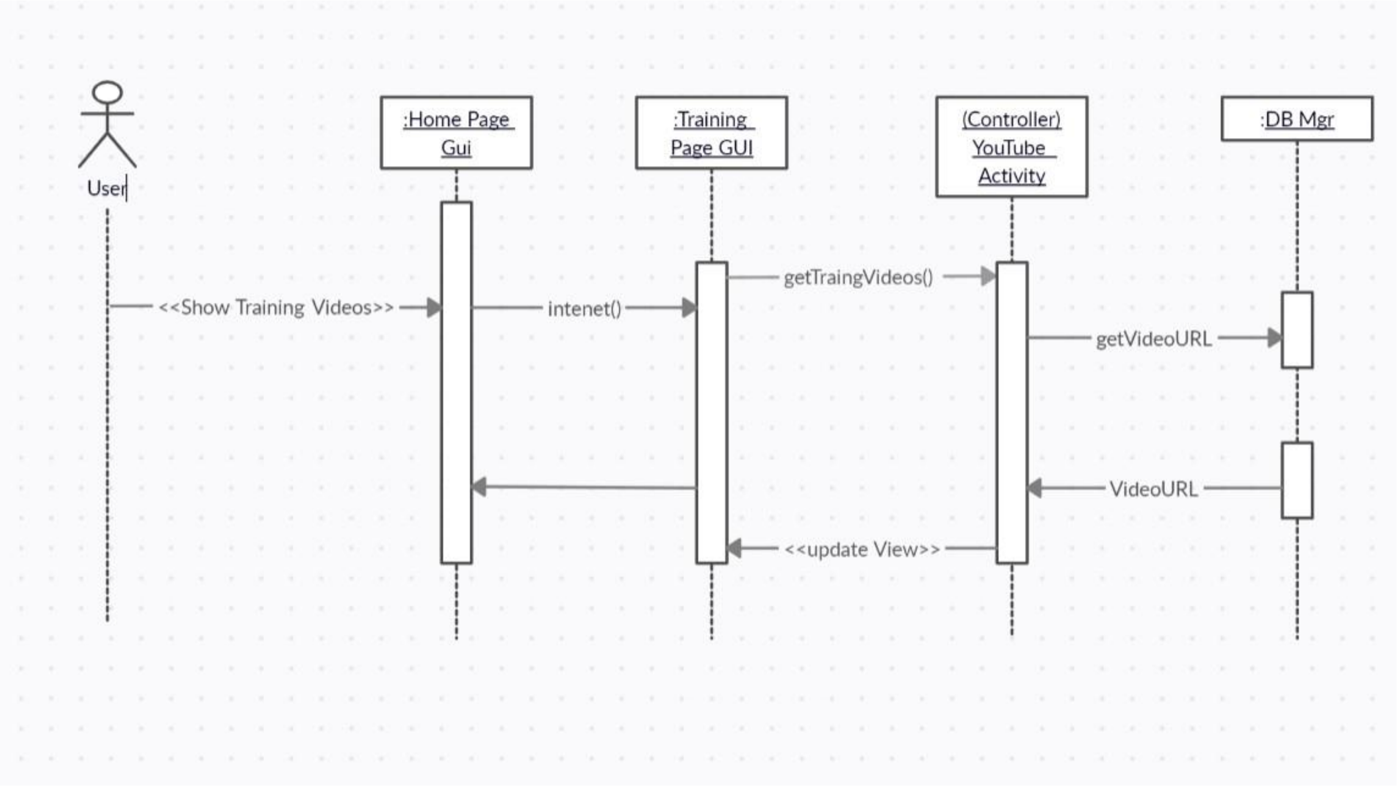
1. View Customer Purchase Section



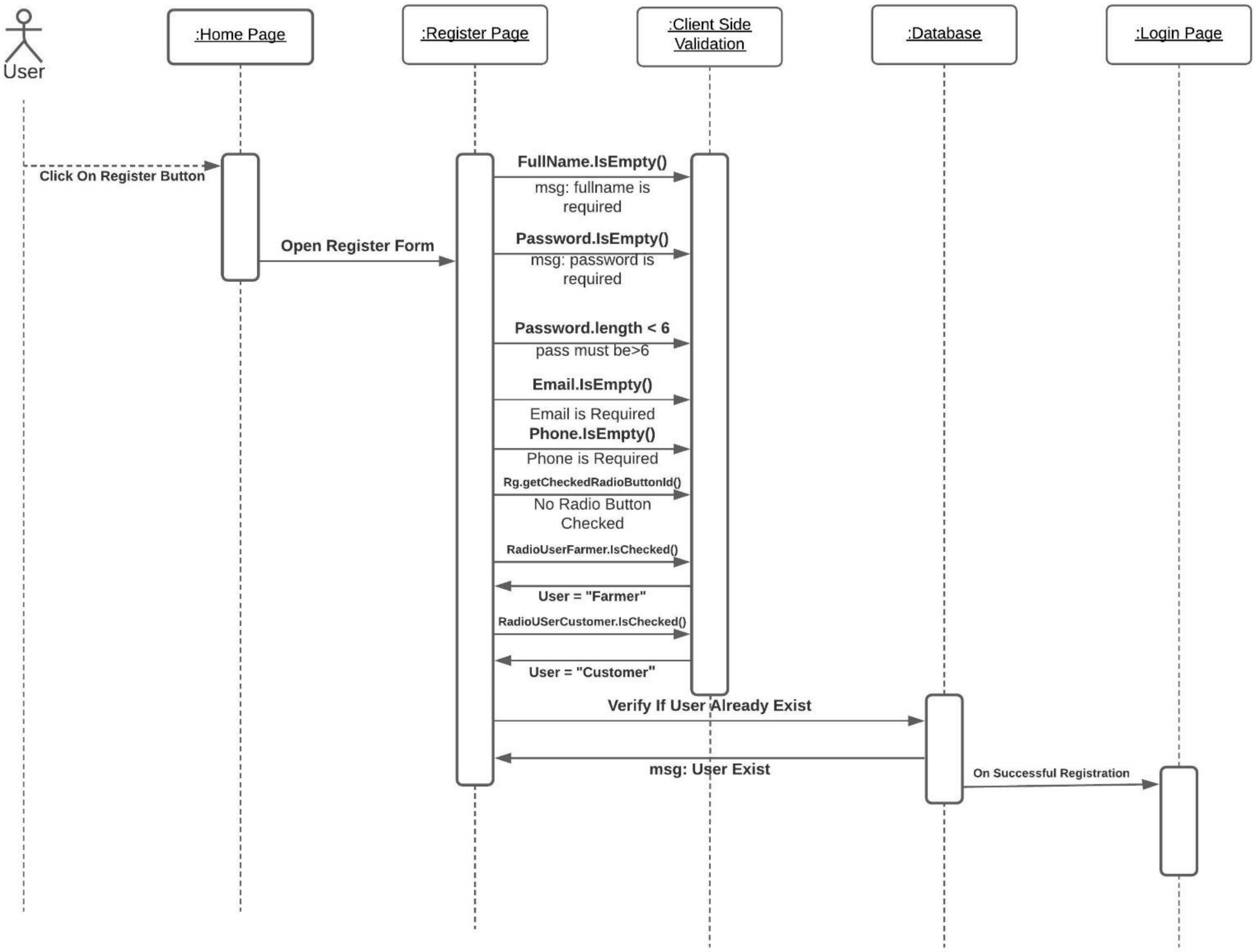
2. View News Section



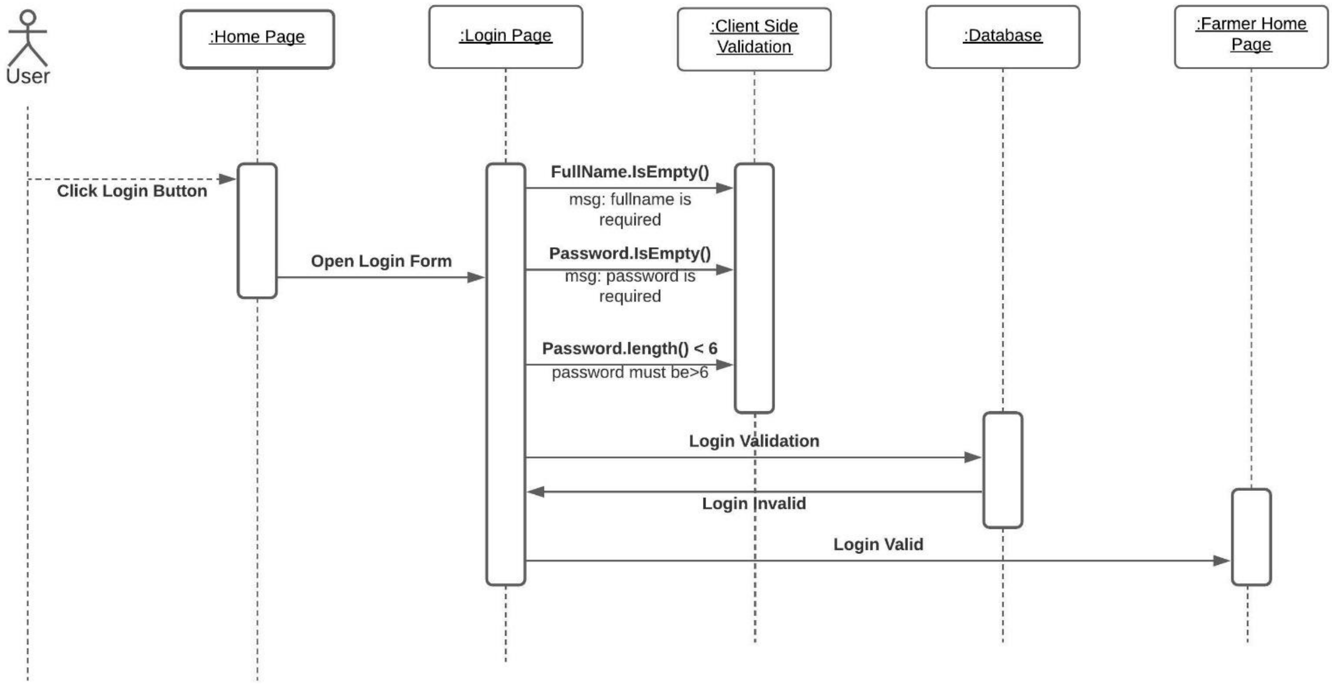
3. View Training Section



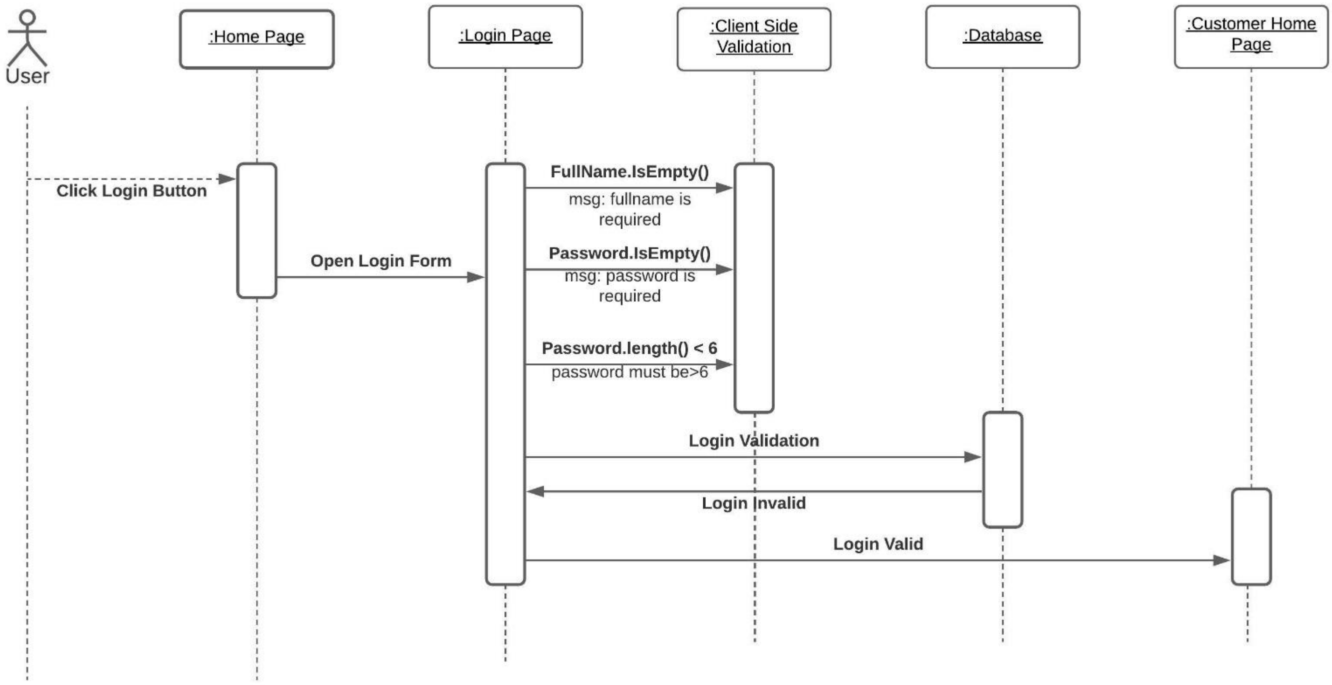
4. Register Page



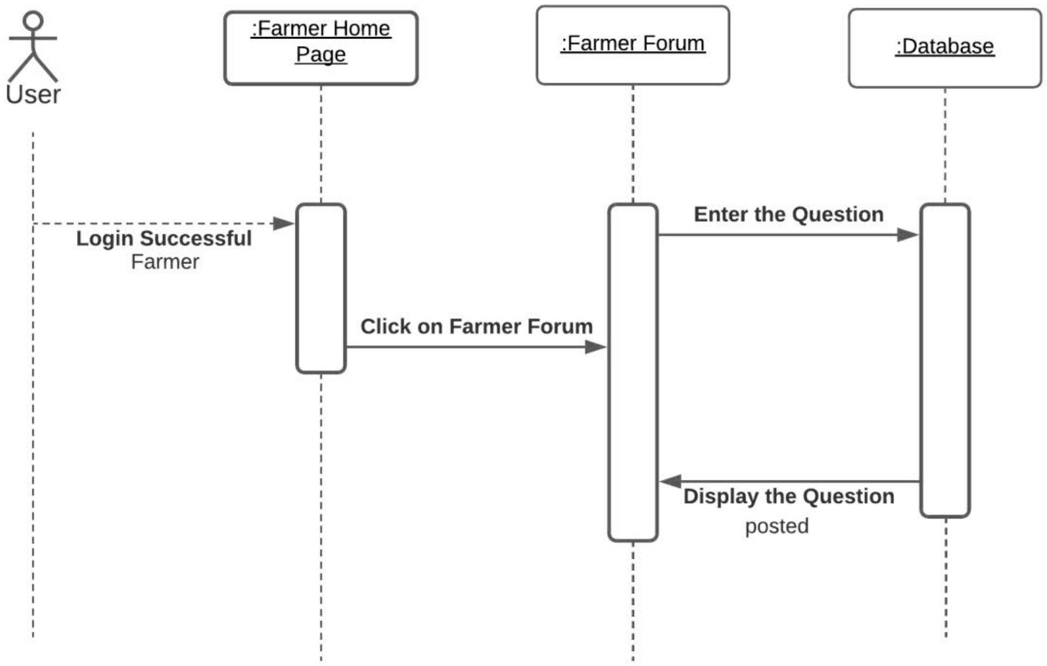
5. Farmer Login



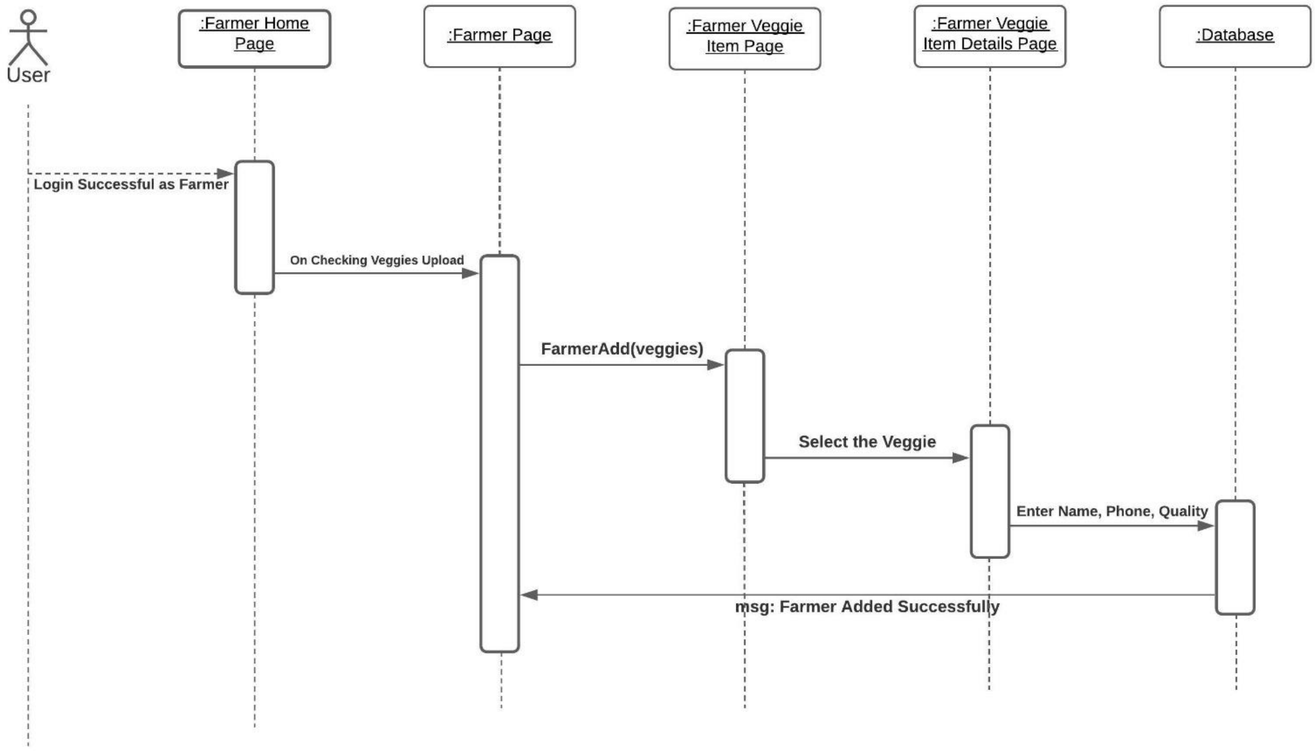
6. Customer Login



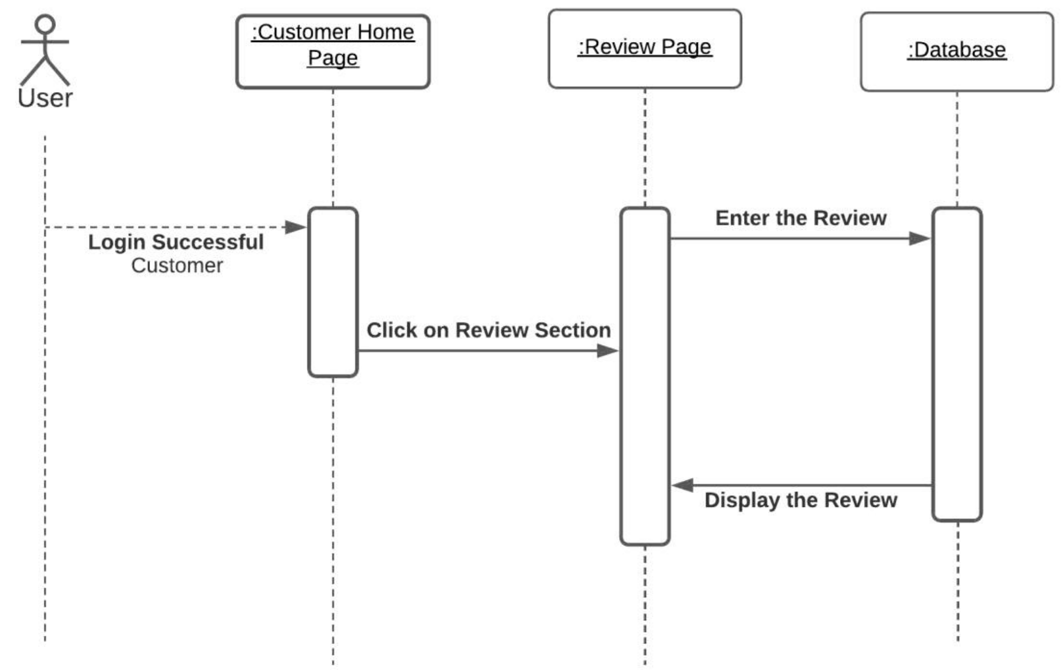
7. Farmer forum page



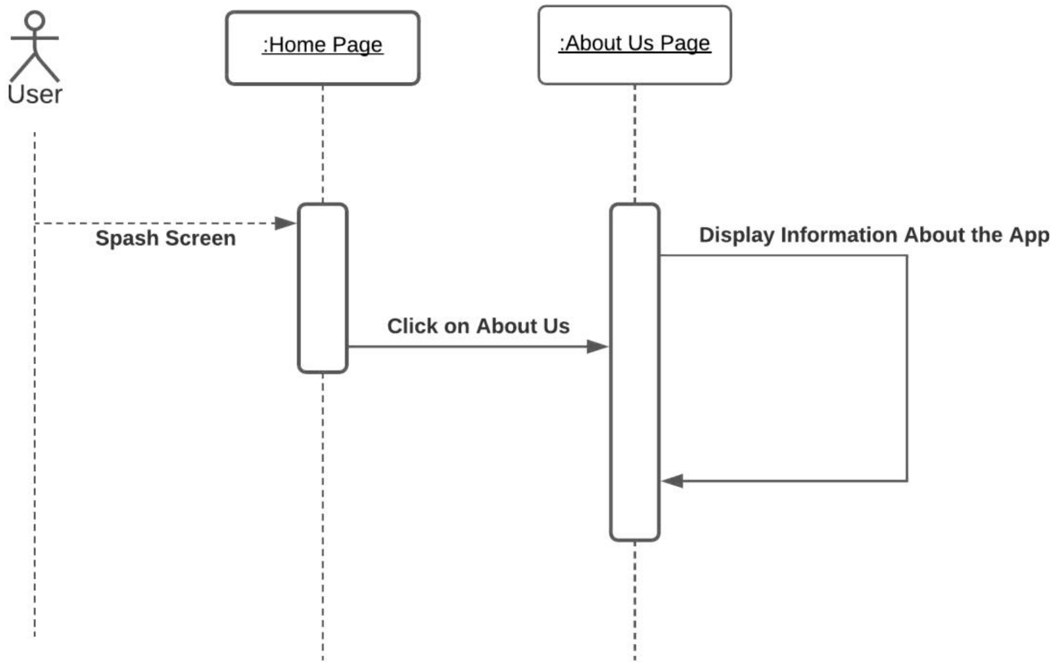
8. Veggies upload Page

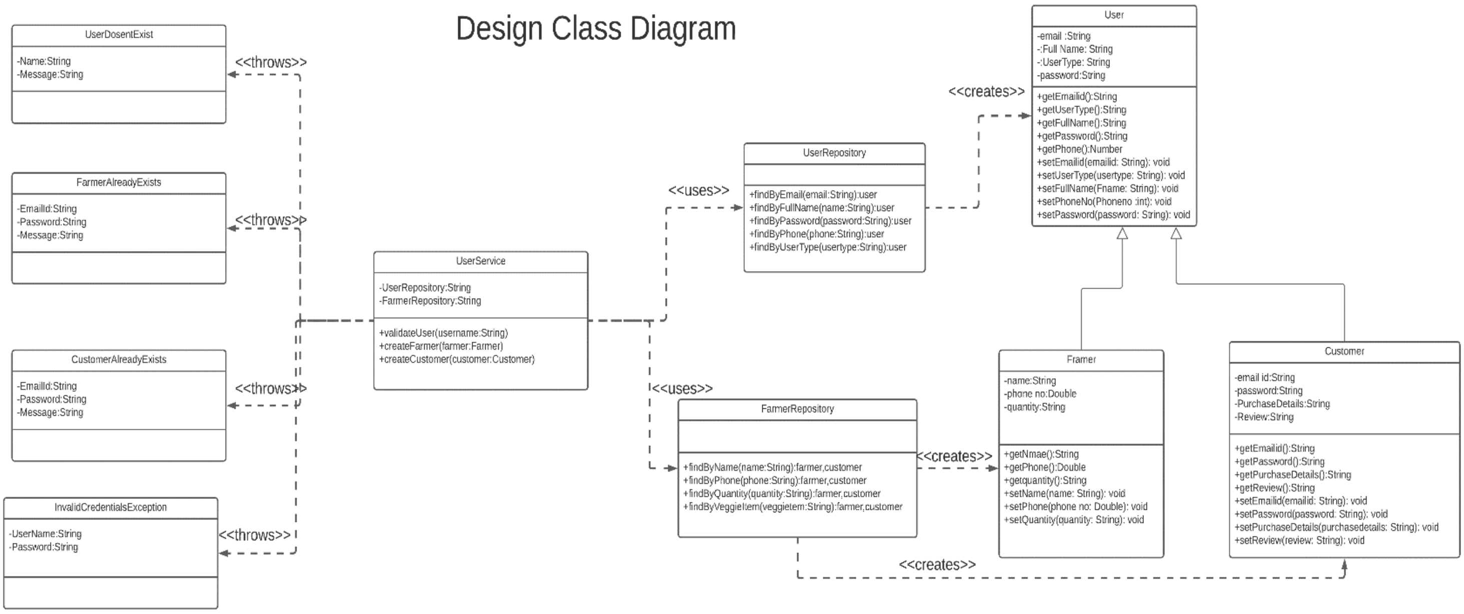

User
:Farmer Home Page
:Farmer Page
:Farmer Veggie Item Page
:Farmer Veggie Item Details Page
:Database
Login Successful as Farmer
On Checking Veggies Upload
FarmerAdd(veggies)
Select the Veggie
Enter Name, Phone, Quality
msg: Farmer Added Successfully

9. Review Page



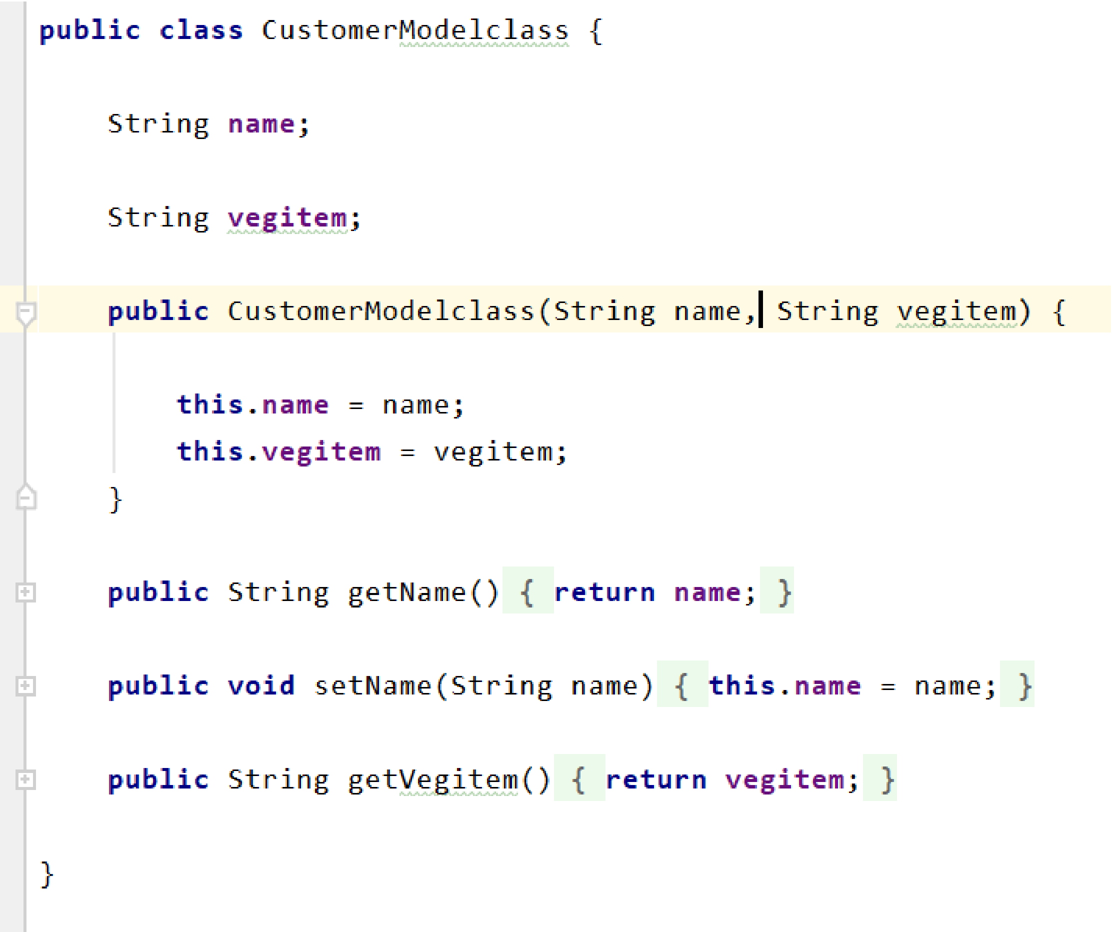
10. About Us




Design Class Diagram
UserDosentExist
-Name:String
-Message:String
<<throws>>
FarmerAlreadyExists
-EmailId:String
-Password:String
-Message:String
<<throws>>
CustomerAlreadyExists
-EmailId:String
-Password:String
-Message:String
<<throws>>
InvalidCredentialsException
-UserName:String
-Password:String
<<throws>>
UserService
-UserRepository:String
-FarmerRepository:String
+validateUser(username:String)
+createFarmer(farmer:Farmer)
+createCustomer(customer:Customer)
<<uses>>
UserRepository
+findByEmail(email:String):user
+findByFullName(name:String):user
+findByPassword(password:String):user
+findByPhone(phone:String):user
+findByUserType(usertype:String):user
<<creates>>
User
-email :String
-:Full Name: String
-:UserType: String
-password:String
+getEmailid():String
+getUserType():String
+getFullName():String
+getPassword():String
+getPhone():Number
+setEmailid(emailid: String): void
+setUserType(usertype: String): void
+setFullName(Fname: String): void
+setPhoneNo(Phoneno :int): void
+setPassword(password: String): void
<<uses>>
FarmerRepository
+findByName(name:String):farmer,customer
+findByPhone(phone:String):farmer,customer
+findByQuantity(quantity:String):farmer,customer
+findByVeggieItem(veggietem:String):farmer,customer
<<creates>>
Framer
-name:String
-phone no:Double
-quantity:String
+getNmae():String
+getPhone():Double
+getquantity():String
+setName(name: String): void
+setPhone(phone no: Double): void
+setQuantity(quantity: String): void
Customer
-email id:String
-password:String
-PurchaseDetails:String
-Review:String
+getEmailid():String
+getPassword():String
+getPurchaseDetails():String
+getReview():String
+setEmailid(emailid: String): void
+setPassword(password: String): void
+setPurchaseDetails(purchasedetails: String): void
+setReview(review: String): void
<<creates>>

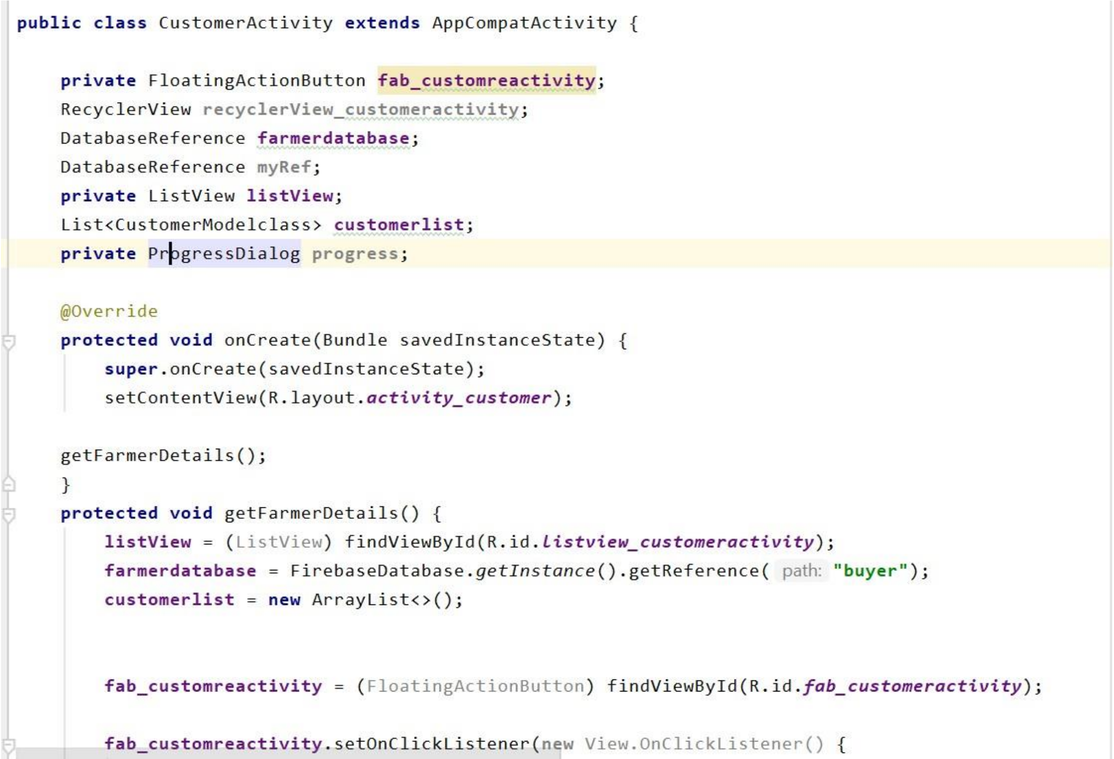
Android Screenshots:

Buyer Model:



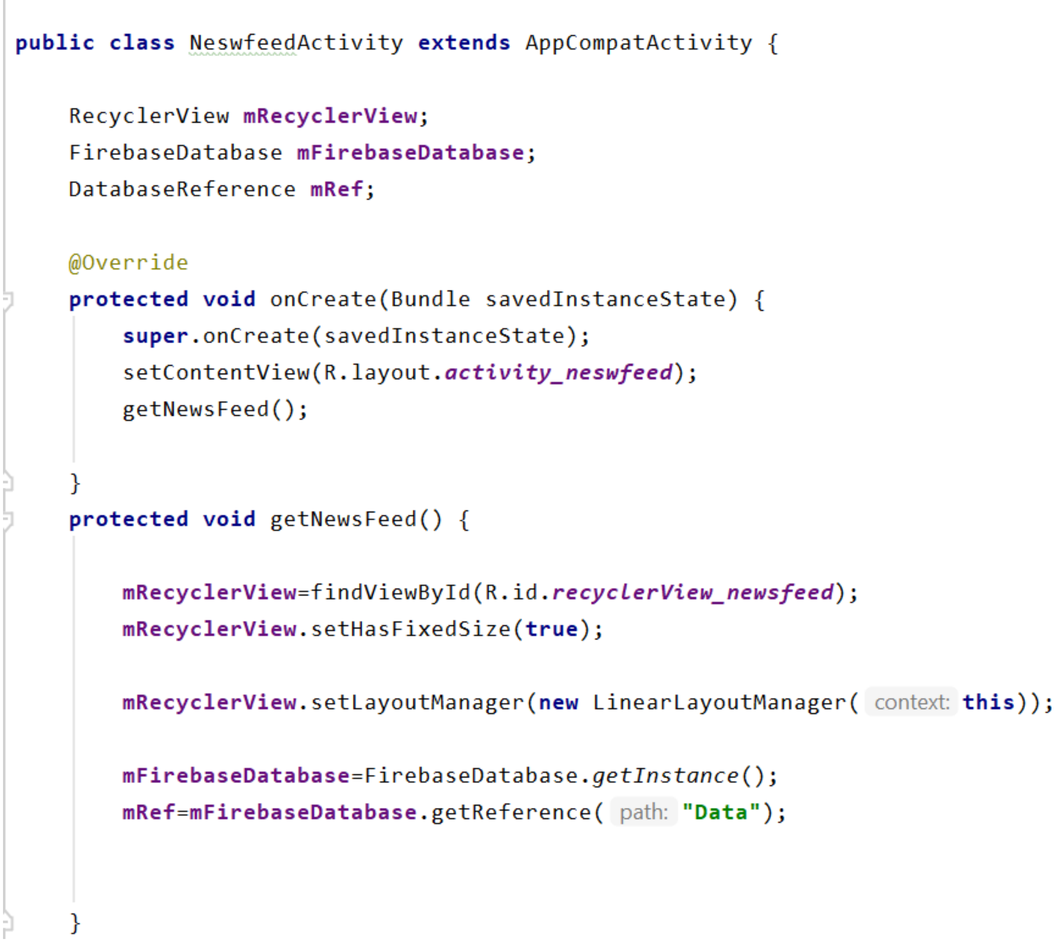
```
public class CustomerModelclass {

    String name;

    String vegitem;

    public CustomerModelclass(String name, String vegitem) {

        this.name = name;
        this.vegitem = vegitem;
    }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public String getVegitem() { return vegitem; }

}
```

Buyer Controller :



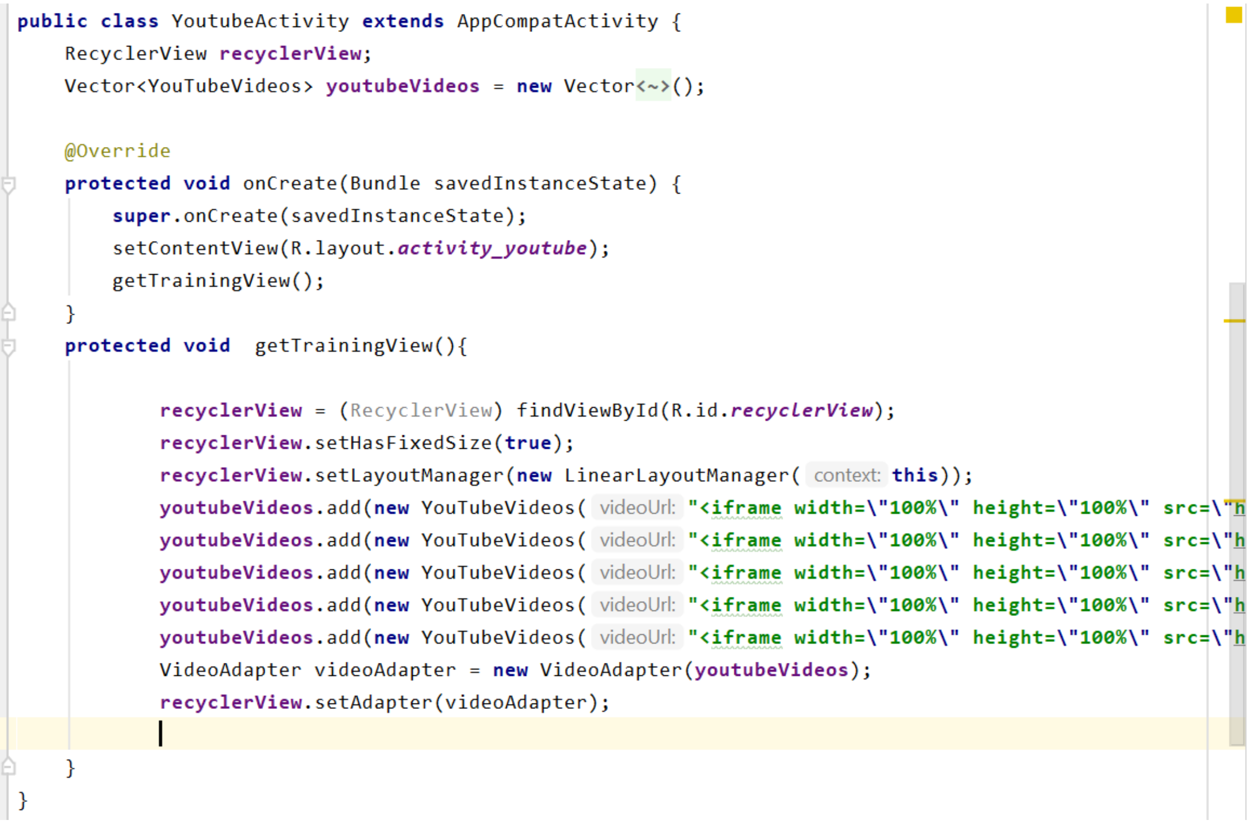
```
public class CustomerActivity extends AppCompatActivity {

    private FloatingActionButton fab_customreactivity;
    RecyclerView recyclerView_customeractivity;
    DatabaseReference farmerdatabase;
    DatabaseReference myRef;
    private ListView listView;
    List<CustomerModelclass> customerlist;
    private ProgressDialog progress;

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

    getFarmerDetails();
    }
    protected void getFarmerDetails() {
        listView = (ListView) findViewById(R.id.listview_customeractivity);
        farmerdatabase = FirebaseDatabase.getInstance().getReference( path: "buyer");
        customerlist = new ArrayList<>();


        fab_customreactivity = (FloatingActionButton) findViewById(R.id.fab_customeractivity);

        fab_customreactivity.setOnClickListener(new View.OnClickListener() {
```

News_Feed:



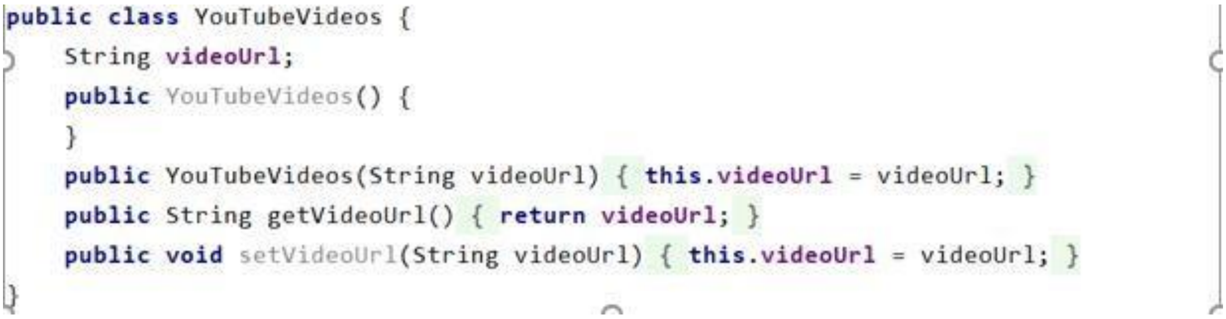
```
public class NeswfeedActivity extends AppCompatActivity {

    RecyclerView mRecyclerView;
    FirebaseDatabase mFirebaseDatabase;
    DatabaseReference mRef;

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

    }
    protected void getNewsFeed() {

        mRecyclerView=findViewById(R.id.recyclerView_newsfeed);
        mRecyclerView.setHasFixedSize(true);

        mRecyclerView.setLayoutManager(new LinearLayoutManager( context: this));

        mFirebaseDatabase=FirebaseDatabase.getInstance();
        mRef=mFirebaseDatabase.getReference( path: "Data");


    }
```

Training_Controller:



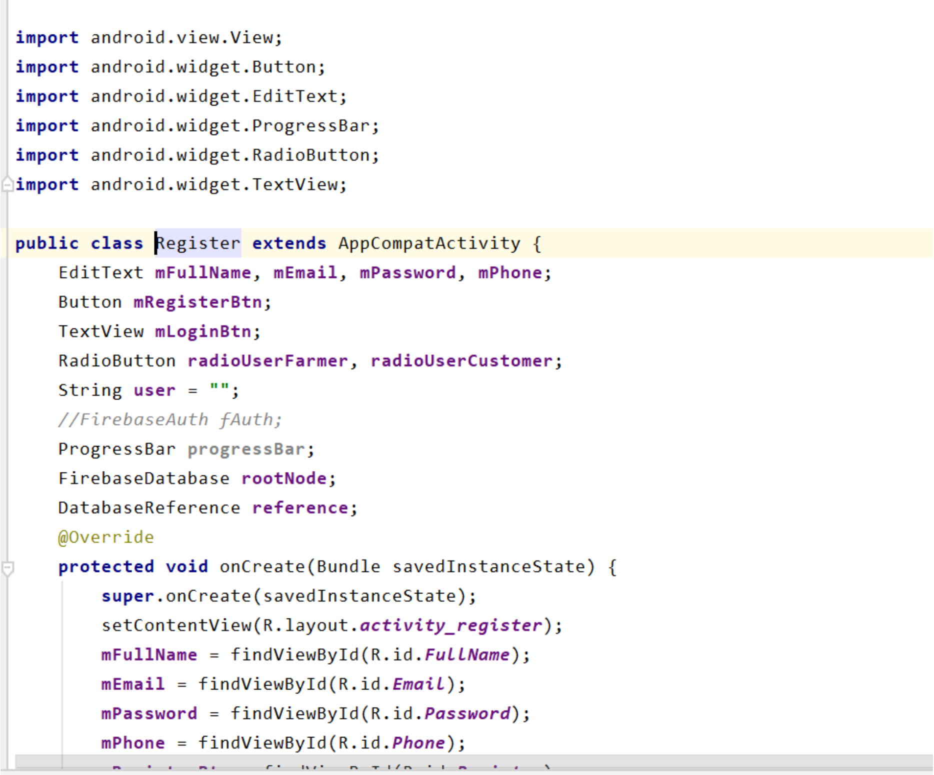
```
public class YoutubeActivity extends AppCompatActivity {
    RecyclerView recyclerView;
    Vector<YouTubeVideos> youtubeVideos = new Vector<~>();

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

        recyclerView = (RecyclerView) findViewById(R.id.recyclerView);
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new LinearLayoutManager( context: this));
        youtubeVideos.add(new YouTubeVideos( videoUrl: "<iframe width=\"100%\" height=\"100%\" src=\"h
        youtubeVideos.add(new YouTubeVideos( videoUrl: "<iframe width=\"100%\" height=\"100%\" src=\"h
        youtubeVideos.add(new YouTubeVideos( videoUrl: "<iframe width=\"100%\" height=\"100%\" src=\"h
        youtubeVideos.add(new YouTubeVideos( videoUrl: "<iframe width=\"100%\" height=\"100%\" src=\"h
        youtubeVideos.add(new YouTubeVideos( videoUrl: "<iframe width=\"100%\" height=\"100%\" src=\"h
        VideoAdapter videoAdapter = new VideoAdapter(youtubeVideos);
        recyclerView.setAdapter(videoAdapter);

    }
}
```

Training_Model:



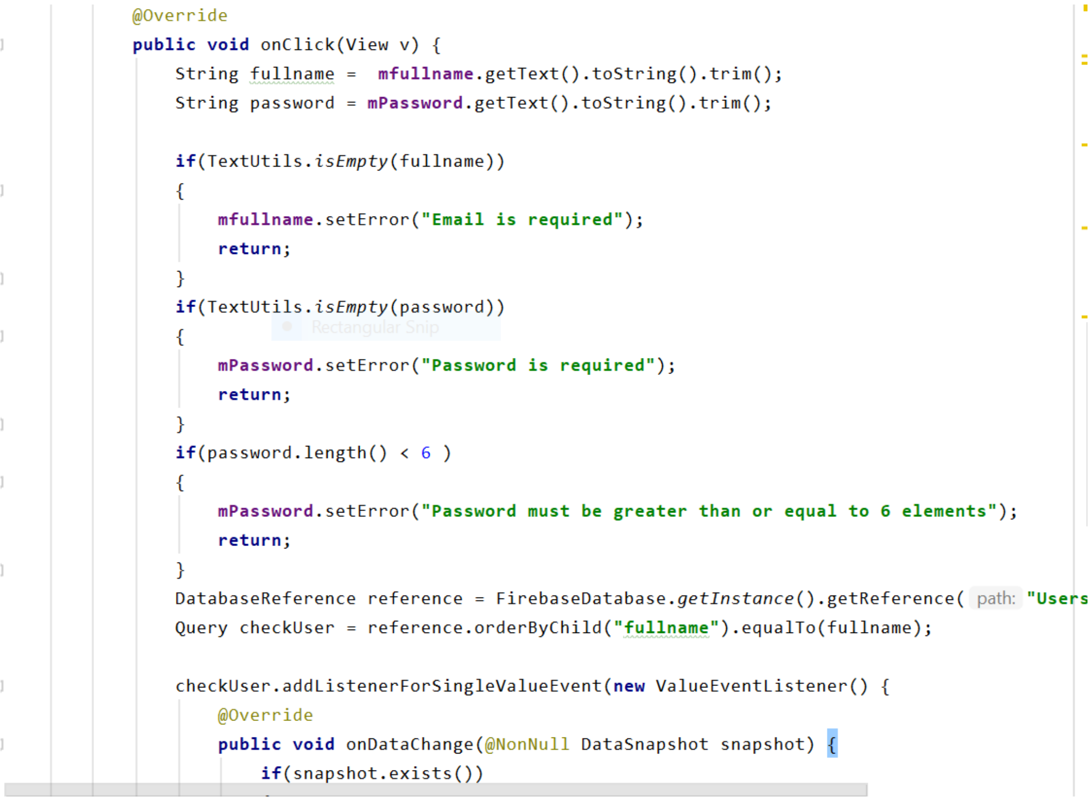
```
public class YouTubeVideos {
    String videoUrl;
    public YouTubeVideos() {
    }
    public YouTubeVideos(String videoUrl) { this.videoUrl = videoUrl; }
    public String getVideoUrl() { return videoUrl; }
    public void setVideoUrl(String videoUrl) { this.videoUrl = videoUrl; }
}
```

Register Page



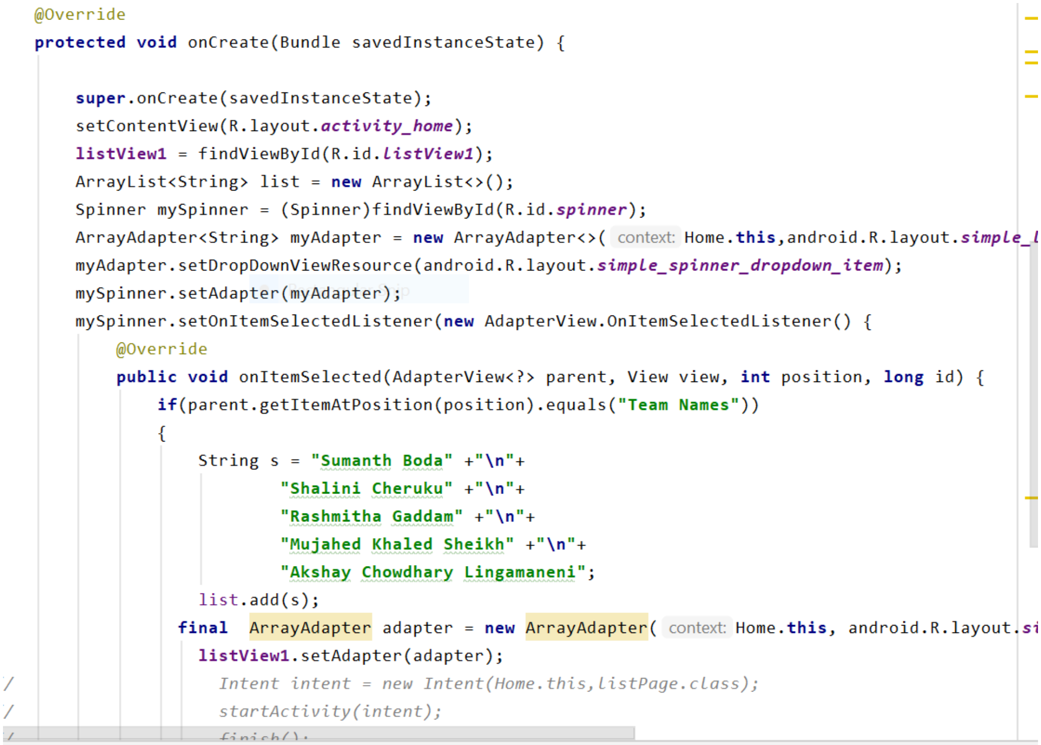
```
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.RadioButton;
import android.widget.TextView;

public class Register extends AppCompatActivity {
    EditText mFullName, mEmail, mPassword, mPhone;
    Button mRegisterBtn;
    TextView mLoginBtn;
    RadioButton radioUserFarmer, radioUserCustomer;
    String user = "";
    //FirebaseAuth fAuth;
    ProgressBar progressBar;
    FirebaseDatabase rootNode;
    DatabaseReference reference;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        mFullName = findViewById(R.id.FullName);
        mEmail = findViewById(R.id.Email);
        mPassword = findViewById(R.id.Password);
        mPhone = findViewById(R.id.Phone);
        mRegisterBtn = findViewById(R.id.Register);
```

Login :



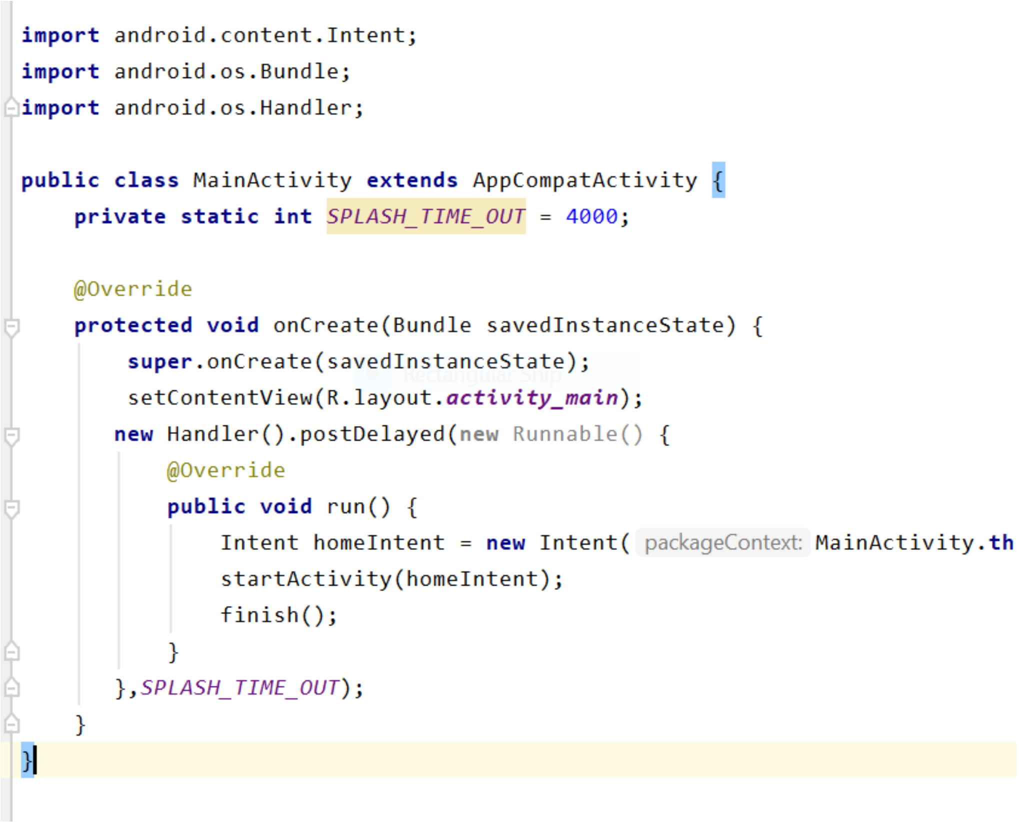
```
@Override
public void onClick(View v) {
    String fullname =  mfullname.getText().toString().trim();
    String password = mPassword.getText().toString().trim();

    if(TextUtils.isEmpty(fullname))
    {
        mfullname.setError("Email is required");
        return;
    }
    if(TextUtils.isEmpty(password))
    {
        mPassword.setError("Password is required");
        return;
    }
    if(password.length() < 6 )
    {
        mPassword.setError("Password must be greater than or equal to 6 elements");
        return;
    }
    DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "Users
    Query checkUser = reference.orderByChild("fullname").equalTo(fullname);

    checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if(snapshot.exists())
```

Home Page:



```
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        listView1 = findViewById(R.id.listView1);
        ArrayList<String> list = new ArrayList<>();
        Spinner mySpinner = (Spinner)findViewById(R.id.spinner);
        ArrayAdapter<String> myAdapter = new ArrayAdapter<>( context: Home.this,android.R.layout.simple_l
        myAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        mySpinner.setAdapter(myAdapter);
        mySpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
                if(parent.getItemAtPosition(position).equals("Team Names"))
                {
                    String s = "Sumanth Boda" +"\n"+
                            "Shalini Cheruku" +"\n"+
                            "Rashmitha Gaddam" +"\n"+
                            "Mujahed Khaled Sheikh" +"\n"+
                            "Akshay Chowdhary Lingamaneni";
                    list.add(s);
                  final  ArrayAdapter adapter = new ArrayAdapter( context: Home.this, android.R.layout.si
                    listView1.setAdapter(adapter);
/                     Intent intent = new Intent(Home.this,listPage.class);
/                     startActivity(intent);
/                     finish();
```

Main A ctivity:



```
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

public class MainActivity extends AppCompatActivity {
    private static int SPLASH_TIME_OUT = 4000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent homeIntent = new Intent( packageContext: MainActivity.th
                startActivity(homeIntent);
                finish();
            }
        },SPLASH_TIME_OUT);
    }
}
```

INCREMENT MATRIX

Use Case	Priority	Effort(person-weeks)	Depends on	Iteration 1 (10/2/2020)	Iteration 2 (11/6/2020)	Iteration 3 (12/4/2020)
UC 1	5	1	None	-		5
UC 2	1	2	UC1	-		2
UC 3	1	2	UC1	-		2
UC 4	8	3	UC2, UC3	-	3	
UC 5	3	2	UC2, UC3	-	2	
UC 6	5	2	UC2	-	2	
UC 7	15	2	U9	-		2
UC 8	3	3	None	-		3
UC 9	4	3	None	-		3
UC 10	14	2	UC7	-		2
UC 11	10	3	None	-		3
Total Effort		25			**7**	**18**

Task Assigned List

Task	Iteration	Done By
Project Description	1.1	Mujahed, Akshay, Shalini, Sumanth, Rashmitha.
Requirements	1.1	Mujahed, Shalini, Rashmitha
High-Level Use Cases UC1	1.2	Shalini, Akshay, Sumanth
High-Level Use Cases UC2	1.2	Shalini, Akshay, Sumanth
High-Level Use Cases UC3	1.2	Shalini, Akshay, Sumanth
High-Level Use Cases UC4	1.2	Shalini, Akshay, Sumanth
High-Level Use Cases UC5	1.2	Shalini, Akshay, Sumanth
High-Level Use Cases UC6	1.2	Rashmitha, Shalini
High-Level Use Cases UC7	1.2	Rashmith, Shalini
High-Level Use Cases UC8	1.2	Rashmitha, Shalini
High-Level Use Cases UC9	1.2	Rashmitha, Sumanth
High-Level Use Cases UC10	1.2	Rashmitha, Sumanth
Use Case Diagram	1.2	Shalini
Use Case Tracebality Matrix	1.2	Rashmitha, Akshay
Domain Diagram	1.2	Shalini, Sumanth
Increment Matrix	1.2`	Rashmitha, Shalini
Project Description	1.3	Mujahed, Akshay, Shalini, Sumanth, Rashmitha.
Requirement	1.3	Mujahed, Shalini, Rashmitha
High-Level Use Cases UC1	1.3	Shalini, Akshay, Sumanth
High-Level Use Cases UC2	1.3	Shalini, Akshay, Sumanth
High-Level Use Cases UC3	1.3	Shalini, Akshay, Sumanth
High-Level Use Cases UC4	1.3	Shalini, Akshay, Sumanth
High-Level Use Cases UC5	1.3	Shalini, Akshay, Sumanth
High-Level Use Cases UC6	1.3	Rashmitha, Shalini, Mujahed
High-Level Use Cases UC7	1.3	Rashmitha, Shalini, Mujahed
High-Level Use Cases UC8	1.3	Rashmitha, Shalini, Mujahed

Use Case Diagram	1.3	Rashmitha, Sumanth
Use Case Tracebality Matrix	1.3	Rashmitha, Sumanth
Domain Diagram	1.3	Shalini
Increment Matrix	1.3	Rashmitha, Akshay
Expanded Use Cases	2	Rashmitha,Sumanth
Sequence Diagrams	2	Akshay ,mujahed
Design Class Diagram	2	Rashmitha,shalini
UC4, UC5	2	Mujahed,Sumanth,akshay
UC6	2	Shalini, Rashmitha
UC1, UC2, UC3	3	Akshay,Sumanth
UC7, UC8, UC9	3	Mujahed,Rashmitha
UC10, UC11	3	Mujahed,Shalini

Video Link:

https://www.youtube.com/watch?v=85RapYqS0zg