FAIR

E3 SEM2 PROJECT REPORT

Submitted by

PRATHYUSHA P

Roll No : R161804

In partial fulfilment of the project requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

RK Valley Campus

Under the Guidance of

Mr. N. Satyanandaram,

Lecturer,Dept.Of CSE,

RGUKT, RK Valley.

IBM

BONAFIDE CERTIFICATE

This is to certify that the project report titled "FAIR" submitted by PRATHYUSHA P bearing registration number R161804, respectively is a record of bonafide project work carried out under my supervision.

(Head of the Department)	(Project Guide)
Ms.Challa Ratnakumari	Mr.N Satyanandaram
Asst.professor in CSE	Lecturer in CSE
Department of CSE	Department of CSE
RGUKT, R K VALLEY	RGUKT, R K VALLEY

IBM **SkillsBuild** for Job Seekers

Completion Certificate

This certificate is presented to

p prathyusha

for the completion of

Build Your First Web Pages With HTML and CSS | OpenClassrooms

(URL-EB0350DFBA28)

As indicated by this learner

Completion date: 02 Dec 2021 (GMT)

Learning hours: 10 hrs

IBM **SkillsBuild** for Job Seekers

Completion Certificate

This certificate is presented to

p prathyusha

for the completion of

Learn CSS Tutorial | w3schools

(URL-718973C05E69)

As indicated by this learner

Completion date: 02 Dec 2021 (GMT)

Learning hours: 9 hrs

IBM **SkillsBuild** for Job Seekers

Completion Certificate

This certificate is presented to

p prathyusha

for the completion of

Learn JavaScript | Codecademy

(URL-4FF4AA6A2C77)

As indicated by this learner

Completion date: 02 Dec 2021 (GMT)

Learning hours: 30 hrs

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people, who made it possible and whose constant guidance and encouragement crowns all efforts with success.

Firstly, I would humbly thank Mr. N Satyanandaram, Lecturer, CSE, for being the channel and guide for me to have availed such a rewarding opportunity.

I would like to express my deep sense of gratitude and whole-hearted thanks to IBM SkillBuild for giving me the privilege of learning under the guidance and for the valuable tutorials,during the course and turning the plans into concrete outputs.

ABSTRACT

The Fair acts as an platform for the farmer to sell his crops in online. The customer can directly get the fresh food items directly from the farmer itself. This is an android application that allows a farmer to register his detai.ls through a sms/phone call/login. After registering the farmer will update the crop details like the crop type and the no.of hectares. The user/customer will also registers in the same way.Once the crop is ready to be sold out, the farmer updates that,then his number will beavailable/visible to the registered customers near to him. This process will be done by accessing the location. The rate will be fixed by the farmer in the phone, then the customer can go and get the items. The customer can review if the farmer is going far behind the market price. If the order cannot be taken by the customer then he can take our delivery boy's help. For thefacility of farmer there will also be a language switch option.And we also add a scanner which scans the passport image of the user.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 About the Android

World is contracting with the growth of mobile phone technology. As the number of users is increasing day by day, facilities are also increasing. Starting with simple regular handsets which were used just for making phone calls, mobiles have changed our lives and have become part of it. Now they are not used just for making calls but they have innumerable uses and can be used as a Camera , Music player, Tablet PC, T.V. , Web browser etc . And with the new technologies, new software and operating systems are required.

What is Android?

Operating Systems have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android.

One of the most widely used mobile OS these days is ANDROID. Android does a software bunch comprise not only operating system but also middleware and key applications.

Android Inc was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later Android Inc. was acquired by Google in 2005. After original release there have been number of updates in the original version of Android.

Fig 1.1: Android versions

Features & Specifications

Android is a powerful Operating System supporting a large number of applications in Smart Phones. These applications make life more comfortable and advanced for the users. Hardwares that support Android are mainly based on ARM architecture platform. Some of the

current features and specifications of android are:

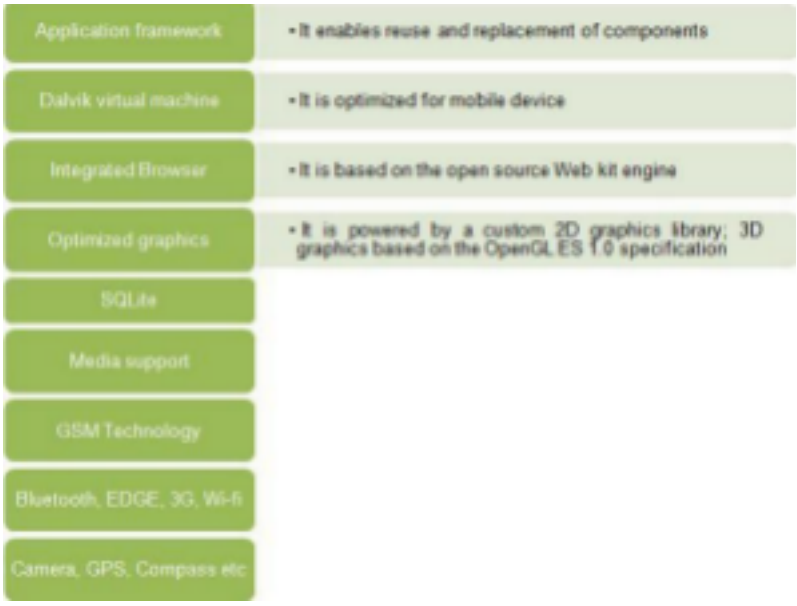


Fig 1.2: Features of Android OS

Android comes with an Android market which is an online software store. It was developed by Google. It allows Android users to select, and download applications developed by third party developers and use them. There are around 2.0 lack+ games, application and widgets available on the market for users.

Android applications are written in java programming language. Android is available as open source for developers to develop applications which can be further used for selling in android market. There are around 200000 applications developed for android with over 3 billion+ downloads. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. For software development, Android provides Android SDK (Software development kit). Read more about open source software.

Applications

These are the basics of Android applications:

- Android applications are composed of one or more application components (activities, services, content providers, and broadcast receivers)
- Each component performs a different role in the overall application behavior, and each one can be activated individually (even by other applications)
- The manifest file must declare all components in the application and should also declare all application requirements, such as the minimum version of Android required and any hardware configurations required.

- Non-code application resources (images, strings, layout files, etc.) should include alternatives for different device configurations (such as different strings for different languages)

Google, for software development and application development, had launched two competitions ADC1 and ADC2 for the most innovative applications for Android. It offered prizes of USD 10 million combined in ADC1 and 2. ADC1 was launched in January 2008 and ADC 2 was launched in May 2009. These competitions helped Google a lot in making Android better, more user friendly, advanced and interactive.

1.2 About the Project

The basic idea of selling crops after harversting them to a customer is through a dealer.If we are able to linkup the farmers and customers that are in a reachable distance or if we provide a delivery boy for this purpose both the farmer and customer will get benefited through this application. To do this process the farmer and customer has to register their details in the application and the farmer has to register the crop.And after the harvesting time the crop and farmer will be displayed to the nearest customers.

1.3 Objective

It is proposed for the purpose of simplifying the work of a farmer and the health of a consumer. We wish to do this by a mobile application. The customer can get the fresh vegetables/products at a click of a button. And the farmer can sell their crops a click of button.After harvesting the crop the farmer will upload the product status through which the availabaility will be provided to the nearer location customers.

1.4Purpose

The main purpose of this project ensure the android based application that facilitates the customers and farmers without consulting the dealers in between.

2.System Requirements

2.1 Hardware Requirements

Processor : intel i5

HDD : 400 GB

RAM : 4 GB

Mouse : Optical Mouse

2.2 Software Requirements

Operating System : ubuntu 20.04

Coding Language : JAVA

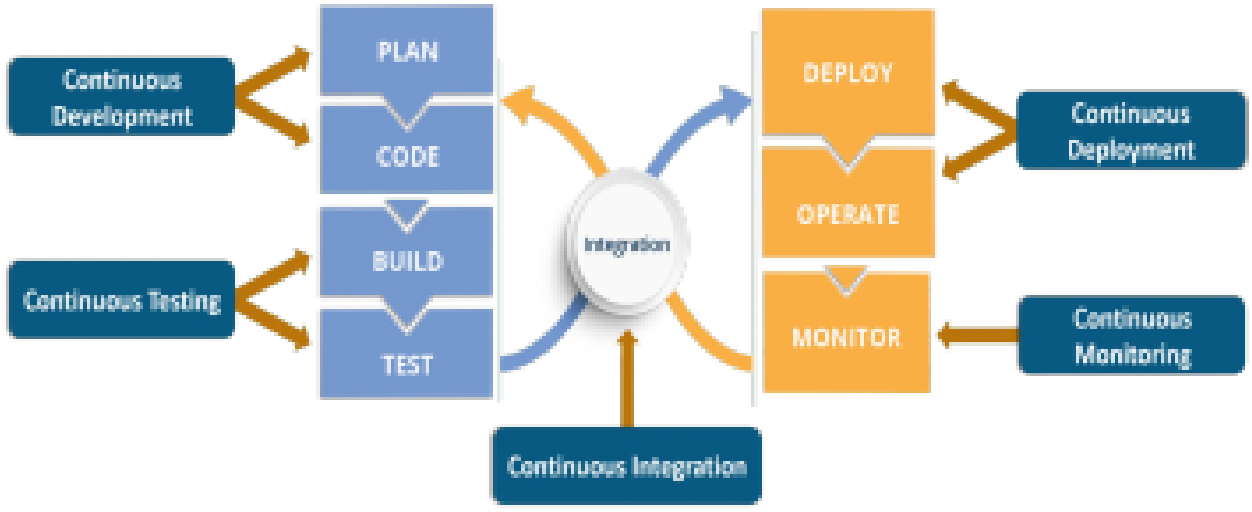
Front End : Android UI

IDE Tools : Android SDK

3. LITERATURE SURVEY

Devops is a software Strategy that is a combination of practices and tools designed to increase an organization's ability to deliver applications and services faster than the traditional software processes.

LifeCycle of Devops:



Devops Stages:

1.Version Control

Also known as Source Code Mangement has two types:

i.)Centralized Version Control

Uses the central server to store all the files and enables team collabration.Works in a single Respository to which users can directly acces the control server.

-Always need to be connected to Internet.

-Once the central server crashes then everthying is going to be lost.

ii.)Distributed Version Control

Every Distributor has a local repository that contain all the files and metadata of the repository.

-All actions except push and pull are very fast and also they don't need any network.

-Commiting new data source can be done without manipulating the data on main repository.

-Once everything is ready then we can push the code at once.

-Even if the data in the main repository is lost it can be recovered from the local repositories.

Git is the most famous tool in the market that is used for Source code Management.

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

pull[Remote repository---> Local Repository]

push[Local Respository--->Remote repository]

Some git codes:

mkdir-for making directory

cd xxx-for moving to the created repository

git init-intializing it as an empty repository

touch x.java-creating a file

git commit

git remote add origin "link of the git repository"

2.Continuos Integration

Continuous intergration is basically a development practice in which the developers are required to commit changes.

-The source code is shared in a repository several times a day.

-Every commit is built,this allows the teams to detect the problems early.

-**Jenkins** is a continuous Integration tool which is widely used now a days.

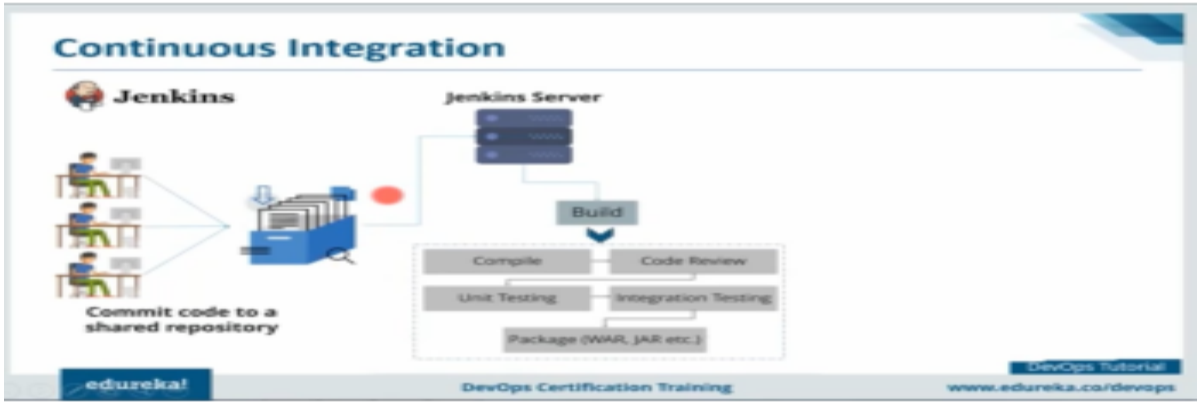
- Jenkins have a continuos process over 2500+ plugins.

-Jenkins integrates all the stages of the devops lifecycle and gives a nice User Interface,which gives 360 degree view of the software life cycle.

-Jenkins Server pulls the code and builds it,when any developer commits a change in the code.

-BUILD contains stages like compiling-->code reviewing-->unittesting--Integrationtesting--->Package(war,jar).Then the package is sent to test servers.

-The advantage with this is that if a build fails then we can immediately know which commit has caused the error.The same happens with testing also.



3.Continuos Delivery

Continuous delivery lets developers automate testing beyond just unit tests so they can verify application updates across multiple dimensions before deploying to customers. These tests may include UI testing, load testing, integration testing, API reliability testing, etc.

4.Continuos Deployment

In continuous deployment the updated versions are released to the customers without the human interventions.

-Though it reduces the time to the developers this is not considered as best practice.

Configuration management: It is a process for maintaining computer systems, servers, and software in a desired, consistent state. It's a way to make sure that a system performs as it's expected to as changes are made over time.

-Helps to solve the dependency issues.

-we always have the previous version stored,which makes it easier to revert back to the previous version whenever any failure occurs because of a software update.

-Tools used Chef,Ansible,puppet,Saltstack. Among these puppet is broadly used)

PUPPET FUNCTIONALITIES:

-puppet is a configuration management tool that is used for deploying configuring and managing severs.

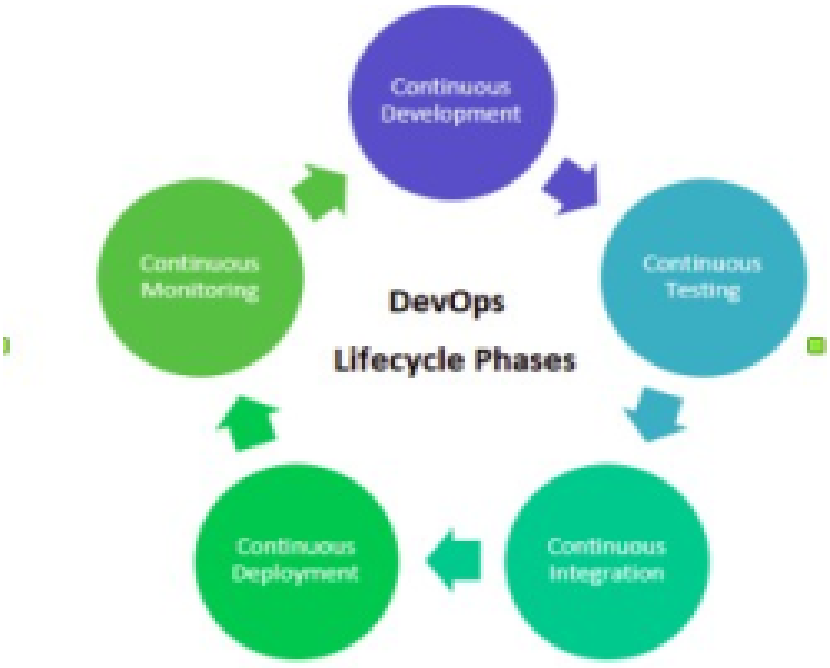
-It follows the master slvae architecture .

-In this we can define distinct configurations for each and every host and it will continuously check and confirm that the configurations didn't alter.

-If the configurations are altered puppet will revert them back to the required configurations.

-It can also help in dynamic scaling up and scaling down machines.

Life cycle of Devops and Tools Used for them:



Continuous Development tools:

Git, Subversion, Jira

Git: Git is afree open source distributed version control system lesigned to handle everything from small to very large projects with speed and efficiency.

Subversion:Subversion is a free/open source version control system (VCS). That is, **Subversion** manages files and directories, and the changes made to them, over time. This allows you to recover older versions of your data, or examine the history of how your data changed.

Jira: Jira Software is an agile project management **tool** that supports any agile methodology, be it scrum, kanban, or your own unique flavor.

Continuos Testing tools:

Selenium,Junit,gradle

Selenium:Selenium is basically **used** to automate the testing across various web browsers. It supports various browsers like Chrome, Mozilla, Firefox, Safari, and IE, and you can very easily automate browser testing across these browsers using **Selenium** WebDriver.

Junit:JUnit is a Java unit testing framework that's one of the best test methods for regression testing. An open-source framework, it is **used to** write and run repeatable automated tests. As with anything else, the **JUnit** testing framework has evolved over time.

Gradle:Gradle is a build automation tool known for its flexibility to build software. A build automation tool is **used** to automate the creation of applications. The building process includes compiling, linking, and packaging the code. The process becomes more consistent with the help of build automation tools.

Continuos Integration tools:

Jenkins,

can be **used to** automate the release management for a **software** application, creating a continuous delivery pipeline.

Hudson: It is a discontinued continuous Integration (CI) tool written in java, which runs in a servlet container such as Apache tomcat or the GlassFish application server.

Continuous Deployment tools:

Dokcer,puppet

chef:Chef is **used to** streamline the task of configuring and maintaining a company's servers, and can integrate with cloud-based platforms such as Amazon EC2, Google Cloud Platform, Oracle Cloud, OpenStack, IBM Cloud, Microsoft Azure, and Rackspace to automatically provision and configure new machines.

Ansible:Ansible is an open-source automation tool, or platform, **used for** IT tasks such as configuration management, application deployment, intraservice orchestration, and provisioning.

Saltstack:SaltStack, also known as **Salt**, is a configuration management and orchestration tool. It **uses** a central repository to provision new servers and other IT infrastructure, to make changes to existing ones, and to install software in IT environments, including physical and virtual servers, as well as the cloud.

Continuous Monitoring tools:

Nagios:Nagios is an open source monitoring system for computer systems. ... **Nagios** software runs periodic checks on critical parameters of application, network and server resources.

splunk:splunk is a software platform widely used for monitoring, searching, analyzing and visualizing the machine-generated data in real time.

Containerization:-

-Light weight alternatives to virtual machines.

-It will use the host's operating system.

-Contains the binary and library dependencies that are required to run an application dependecies.

Tools provided by containerization:

Docker,Google kubernetes Engine,Azure kubernetes service,AWS ECR.

DOCKER:

- A docker file can create docker image.
- With the help of a docker image we can build as many containers as possible.
- Docker image is nothing but a template for our file.
- Docker image can be uploaded into docker hub(git).
- Docker doesn't need any preallocated memory/RAM where a virtual machine needs one.

Why is devops more popular than ever?

There are three possible theories to explain the rise of DevOps over the past decade.

- The first is to argue that Debois simply happened to have some very good ideas, and he promoted them well.
- The second possible explanation for why DevOps became popular is it is simply an evolved form of agile. But DevOps is broader, and has a much

persuasive—is DevOps was born out of a particular moment in the history of computing and IT. That moment, which started in the mid-2000s, was marked by several key features that helped give rise to DevOps, including:

- The widespread adoption, starting circa 2007, of cloud computing.
- The success of the open source software movement around 2010 in making open source the go-to licensing strategy for many popular software projects and in making collaboration around code easier.
- The rise beginning around 2010 of microservices and containers, which made applications much more complex to design, deploy and manage.
- The explosive popularity of mobile computing starting with the launch of the iPhone in 2007. The mobile revolution meant organizations increasingly had to deliver software for multiple target environments (PCs, Android phones and iOS, for example) at the same time, necessitating branched delivery pipelines.
- The demand for ever-increasing speed in software delivery as a way to gain an edge in software markets which, by the late 2000s, had grown more competitive than ever.

When we put these changes together, we end up with a need to manage much more complex, fast-moving software delivery pipelines. Powered by cloud infrastructure and microservices architectures, the software market starting around 2010 required more responsive, better coordinated management by IT teams. DevOps provided an answer by increasing efficiency, decreasing response times and making software delivery faster.

several specific developments in the world of IT between the mid- and late-2000s. Debois may certainly deserve credit for helping to spread the idea, but it would not have spread so quickly if it had not emerged at that particular time.

Advantages of Devops:

- Greater professional development opportunities
- Faster delivery of features
- More stable operating environment
- More time to innovate

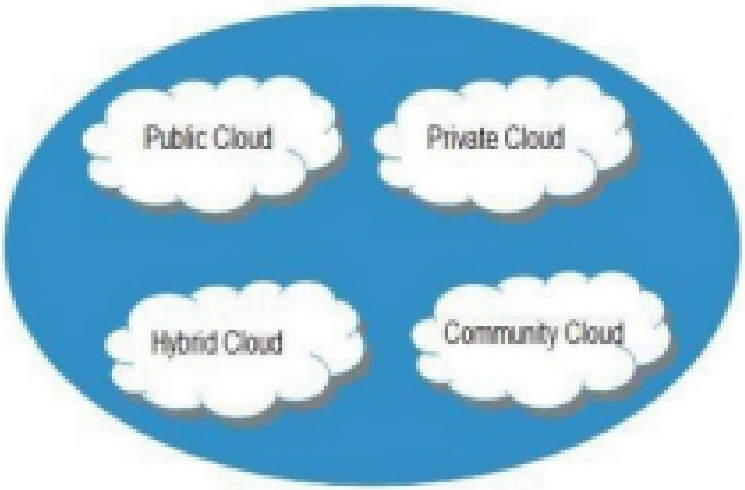
Also, for the companies to be able to respond to failures of their products, it's essential that they approach DevOps practice to keep their systems and software up all the time.

Cloud Computing:

Cloud computing refers to manipulating configuring,anmd accessing the applications online.It offers online data storage,infrastructure and apllication.

Cloud computing is both a combination of software and hardware based computing resources delivered as a network service.

Deployment models define the type of access to the cloud i.e., how the cloud is located?Cloud can have any of the four types of access:



Public Cloud: The public cloud allows systems and services to be easily accesible to the general public,Public cloud may be less secure becuase of its openness

Private Cloud: The private cloud allows systems and services to be accessible within an organization.It offers increased security because of its private nature.

Hybrid Cloud: The Hybrid cloud is mixture of public and private cloud.However,the critical activities are performed using privare cloud while the non-critical activities are performed using public cloud.

Community Cloud: The community cloud allows systems and services to be accessible by group of organizations.

2.Service Models: These are the reference models on which the cloud computing is based.These can be categorized into

is the delivery of technology infrastructure a
scalable service.

provides access to fundamental resources su
machines,virtual mchines,virtual storage etc.

id,terremark,at&t,blizzard.

n as a service(Paas)

provides the runtime environment for
ns,development & deployment tools,etc.

provides all of the facilities required to supp
life cycle of building and delivering web app
ces entirely from the internet.

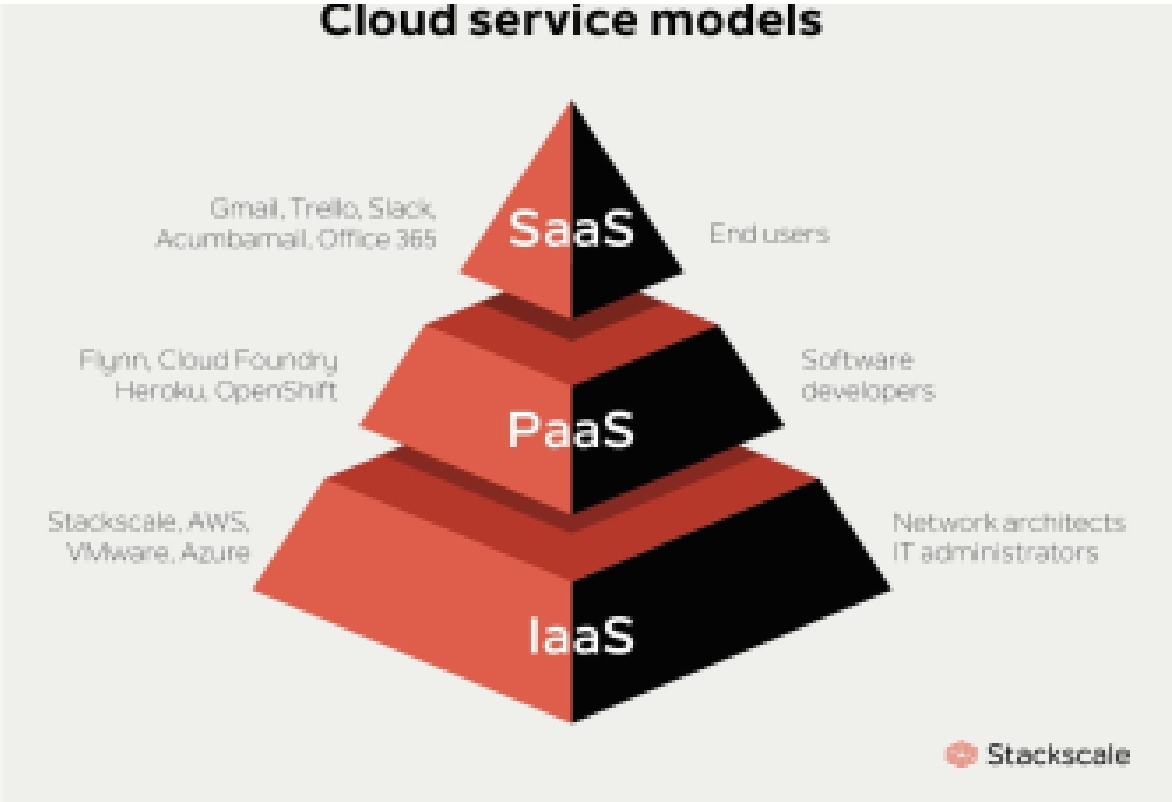
Azure,Salesforce.com

e as a Service

model allows to use software applications as
end users.

software delivery methodology that provides
multi-tenant access to software and its functi
as a web-based service.

force.com,Google,netsuite.



AmazonWebServices:

Aws is a suite of services which offers cloud-computing services/platforms ataffordable rates.Therefore making its customer base strong from small-scal to big enterprises.

AWS is the world's most comprehensive and broadly adopted cloud platform,pffering over 175 fully featured services from data centers globally.

AWS has millions of customers including the fastest-growing startups,largest enterprises,and leading government agencies are using AWS to lower costs,become more agile,and innovate faster.

Computation services in AWS

Elastic Compute Cloud(EC2):These are just the virtual machines in the cloud on which we have the OS level

compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon **EC2's** simple web service interface allows you to obtain and configure capacity with minimal friction.

LightSail: It automatically deploys and manages compute,storage and networking capabilities required to run your apllications.

Elastic Container Service(ECS): It is a highly scalable container service to allows you to run Docker containers in the cloud.

Elastic Container Service for Kubernetes(EKS): Allows user to use Kubernetes on AWS without installing and managing your own kubernetes control plane.It is a relatively new service.

Storage Services

S3(Simple Storage Service)-It can be used to store files,but not for installation of games and os.

EFS(Elastic File System)-Provides file storage for use with your EC2 instances.Uses NFSv4 protocol.

Storage Gateway-A virtual machine that you install on your on-premise servers.Your on-premise data can be backed up to AWS providing more durability.

Database Services:

Relational Database Service(RDS): Allows the user to run relational databselikeMySQL,MariaDB,PostgreSQL,Oracle or SQL Server.These databases are fully managed by AWS like installing antivirus and patches.

DynamoDB-It is a highly scalable,high performance NoSQL databse.It provides single digir millisecond latency at any scale.

Elasticache-It is a way of caching data inside the cloud.It can be used to take load off of your database by caching most frequent queries.

Neptune-It has been launched recently,It is a fast,reliable and scalable graph databse service.

RedShift-It is AWS's data warehousing solution that can be used to run complex OLAP queries.

Networking and Content delivery

Virtual Private Cloud(VPC)-It is simply a data center in the cloud in which you deploy all your resources.It allows you to better isolate your resources and secure them.

CloudFront-It is AWS's Content Delivery Network that consists of Edge locations that cache resources.

Route53-It is AWS's highly available DomainNameSystem service.Through which we can register domain names.

Direct Connect-Using it you can connect your data center to an availability zone using a highspeed dedicated line.

API Gateway-Allows you to create,store and manage APIs at scale.

Developer Tools

CodeStar-It is a cloud-based servie for creating,managing,and working with software development projects on AWS.you can quickly develop,build,and deploy applications on AWS with an AWS codestar project.

CodeCommit-It is AWS's version control service that allows you to store your code and other assets privately in the cloud.

CodeBuild-It automates the process of building your code.

Cloud9-It is an IDE(Integrated Development Environment) for writing,running,and debugging code in the cloud.

Security and Identity:

Identity and Access Management(IAM)-Allows you to manage users,assign policies,create groups to manage multiple users.

Inspector-It is an agent that you install on our virtual machines,which then reports any security vulnerabilities.

Certificate Manager-It gives free SSL certificates for your domains that are managed by Route53.

Directory Service-A way of using your company's account to log in to AWS.

Mobile Services

MobileHub-Allows you to add,configure and design features for mobile apps,It is a console for mobile app development

Cognito-Allows the users to signup usign social identity providers.

Device Farm-Enables you to improve quality of apps by quickly testing on hundreds of mobile devices.

Devops in AWS(cloud):

We use the resources of a cloud to perform the devops lifecycle.

Why Devops on cloud?

-Brings products at a faster rate to the market.

-Reduction of maintaince of servers.

-Increased Security

-Increased Scalability

-At each stage we choose the actions that are performed on our code,such as how and where the code should build test and deploy.

-when the pipeline is active every code change that we do automatically undergoes the actions that we defined.

-The code pipeline allows to setup manual approvals at each stage of the pipeline,so that we can review and approve/reject the code before it progress to next stage of the pipeline.

-The codepipeline also automatically stops our pipeline when there is failure.

For eg:-Failure of unit testing.

-Pay for use.

AWS Components for Devops:

1.AWS Code Commit:A fully managed source central service that hosts secure and highly scalable.

It works like git,but it doesn't need any operating system.

How it works?

We create a repository in AWS code commit service via the console and later using git from the

2.CodePipeline

3.CodeBuild: Fully managed build se
which compiles your source code runs unit
artifacts that are ready to deploy.

-Provides a pre-packaged build enviro
popular programming languages and buildi

-The build will perform the specific tas
t would be uploaded to the S3 Bucket.

se the console to get the detailed build infor

4.CodeDeploy: Code deploy is a servi
s the application deployment and updates a
ws ec2 of any size it automates your code de
tance handles the complexity of updating th

It also avoids rolls back automatically
d apart from that it also integrates with thi
aws to make our job easier.

5.CodeStar(optional): Codestar is b
ased development service that provides tool
to quickly develop build and deploy applica
asically a very templatized format.

4 .FEASIBILITY REPORT

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

4.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

5. SYSTEM ANALYSIS

System Analysis is first stage according to System Development Life Cycle model. This System Analysis is a process that starts with the analyst.

Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. One aspect of analysis is defining the boundaries of the system and determining whether or not a candidate system should consider other related systems. During analysis, data are collected on the available files, decision points, and transactions handled by the present system.

Logical system models and tools that are used in analysis. Training, experience, and common sense are required for collection of the information needed to do the analysis.

5.1 Existing system

In the existing system, Business Model revolves around **Online Grocery Delivery Services** that deliver grocery goods found in home essentials, convenience stores, and food supplies to its users.

Disadvantages

- Farmers will loss
- Not fresh
- Not healthy

5.2Proposed system

The proposed system is a simple android app which enables the users(farmers) to sell their

crops at a button click to the users(consumers) nearer to their location.

Advantages

1. It reduces the intervention of dealers.
2. Health and economical efficient.
3. Time efficient.

5.3 Feasibility study

All projects are feasible given unlimited resources and infinite time. But the development of software is plagued by the scarcity of resources and difficult delivery rates. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time.

Three key considerations are involved in the feasibility analysis.

Economic Feasibility:

This procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise, further justification or alterations in proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

Technical Feasibility:

Technical feasibility centers on the existing computer system (hardware, software, etc.,) and to what extent it can support the proposed addition. If the budget is a serious constraint, then the project is judged not feasible.

Operational Feasibility:

People are inherently resistant to change, and computers have been known to facilitate change. It is understandable that the introduction of a candidate system requires special effort to educate, sell, and train the staff on new ways of conducting business.

5.4 FEASIBILITY STUDY IN THIS PROJECT

1. Technical feasibility:

The system is self-explanatory and does not need any extra sophisticated training. As the system has been built by concentrating on the Graphical User Interface Concepts, the application can also be handled very easily with a novice User. The overall time that is required to train the users upon the system is less than half an hour.

The System has been added with features of menu-driven and button interaction methods, which makes the user the master as he starts working through the environment. The net time the customer should concentrate is on the installation time.

2. Financial Feasibility:

i) <u>Time Based</u>: Contrast to the manual system management can generate any report just by single click. In manual system it is too difficult to maintain historical data which become easier in this system. Time consumed to add new records or to view the reports is very less compared to manual system. So this project is feasible in this point of view.

ii) <u>Cost Based</u>: No special investment need to manage the tool. No specific training is required for employees to use the tool. Investment requires only once at the time of installation. The software used in this project is freeware so the cost of developing the tool is minimal and hence the overall cost. **6.**

SYSTEM DESIGN

6.1INTRODUCTION

Software vogue sits at the technical kernel of the pc code engineering methodology and is applied despite the event paradigm and house of application. vogue is that the gap among the event section for any designed product or system.

The designer's goal is to supply a model or illustration of associate entity which will later be built. Beginning, once system demand ar like and analysed, system vogue is that the first of the three technical activities -design, code and take a glance at that is required to form and verify code.

The importances are declared with one word "Quality". vogue is that the place where quality is fostered in code development. vogue provides North yankee country with representations of code which will assess for quality.

Design is that the alone manner that we are going to accurately translate a customer's browse into a finished product or system. code vogue could be a foundation for all the pc code engineering steps that follow. whereas not a strong vogue we've got an inclination to risk building associate unstable system – one which will be difficult to envision, one whose quality cannot be assessed until the last stage.

During vogue, progressive refinement of knowledge structure, program structure, and

procedural details unit developed reviewed and documented. System vogue are viewed from either technical or project management perspective.

From the technical purpose of browse, vogue is comprised of four activities – field vogue, system vogue, interface vogue and procedural vogue.

6.2 WORK FLOW DIAGRAM

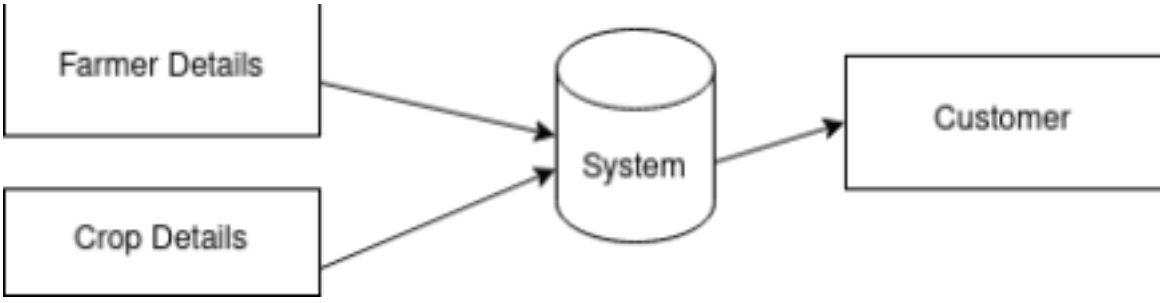


Fig. 6.1: Work Flow Diagram

6.3 UML BASICS

The Unified Modelling Language™ (UML®) can be a customary visual modelling language purported to be used for modelling business and similar processes, analysis, design, and implementation of software-based systems

UML can be a typical language for business analysts, software system package architects and developers accustomed describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software system package systems.

UML could also be applied to varied application domains (e.g., banking, finance, internet, aerospace, healthcare, etc.) it's going to be used with all major object and half software system package development methods and for various implementation platforms (e.g., J2EE, .NET).

UML can be a customary modelling language, not a software system package development methodology. UML 1.4.2 Specification explained that process:

- provides steerage on the order of a team's activities,
- specifies what artifacts ought to be developed,
- directs the tasks of individual developers and therefore the team as an entire, and offers criteria for observation and activity a project's merchandise and activities.

UML is designedly methodology freelance and can be applied inside the context of assorted processes. Still, it's best suited to be used case driven, unvarying and progressive

development processes. Associate in Nursing example of such methodology is Rational Unified methodology (RUP).

UML is not complete and it isn't totally visual. Given some UML diagram, we are going to not ensure to understand drawn [*fr1] or behavior of the system from the diagram alone. Some information are often designedly omitted from the diagram, some information pictured on the diagram might need altogether completely different interpretations, and a number of concepts of UML have no graphical notation within the least, so there is no due to depict those on diagrams.

For example, linguistics of multiplicity of actors and multiplicity of use cases on use case diagrams is not made public precisely among the UML specification and can mean either coincident or successive usage of use cases.

Name of academic degree abstract classifier is shown in italics whereas final classifier has no specific graphical notation, so there is no because of ensure whether or not or not classifier is final or not from the diagram.

There square measure a pair of broad caetgories of diagrams thus square measure all over again divided into sub-categories:

- ✓ Structural Diagrams
- ✓ Behavioral Diagrams

Structural Diagrams:

The structural diagrams represent the static facet of the system. These static aspects represent those components of a diagram that forms the most structure and so stable. These static components are represents by categories, interfaces, objects, parts and nodes. The four structural diagrams are:

- • Class diagram
- • Object diagram
- • Component diagram
- • Deployment diagram

Behavioral Diagrams:

Any system will have 2 aspects, static and dynamic. therefore a model is taken into account as complete once each the aspects area unit lined totally.

Behavioral diagrams essentially capture the dynamic side of a system. Dynamic side are often any delineated because the changing/moving components of a system.

UML has the subsequent 6 forms of activity diagrams:

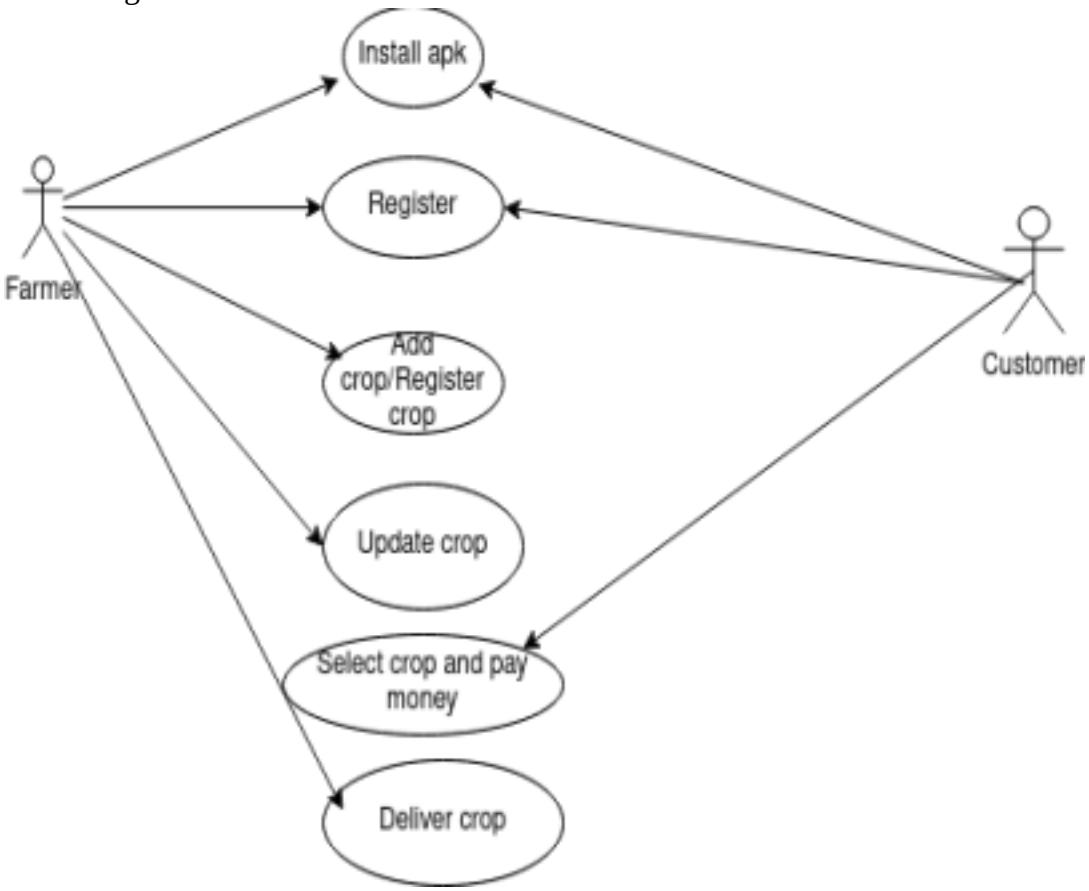
- ✓ Use case diagram
- ✓ Sequence diagram
- ✓ Collaboration diagram
- ✓ State chart diagram
- ✓ Activity diagram

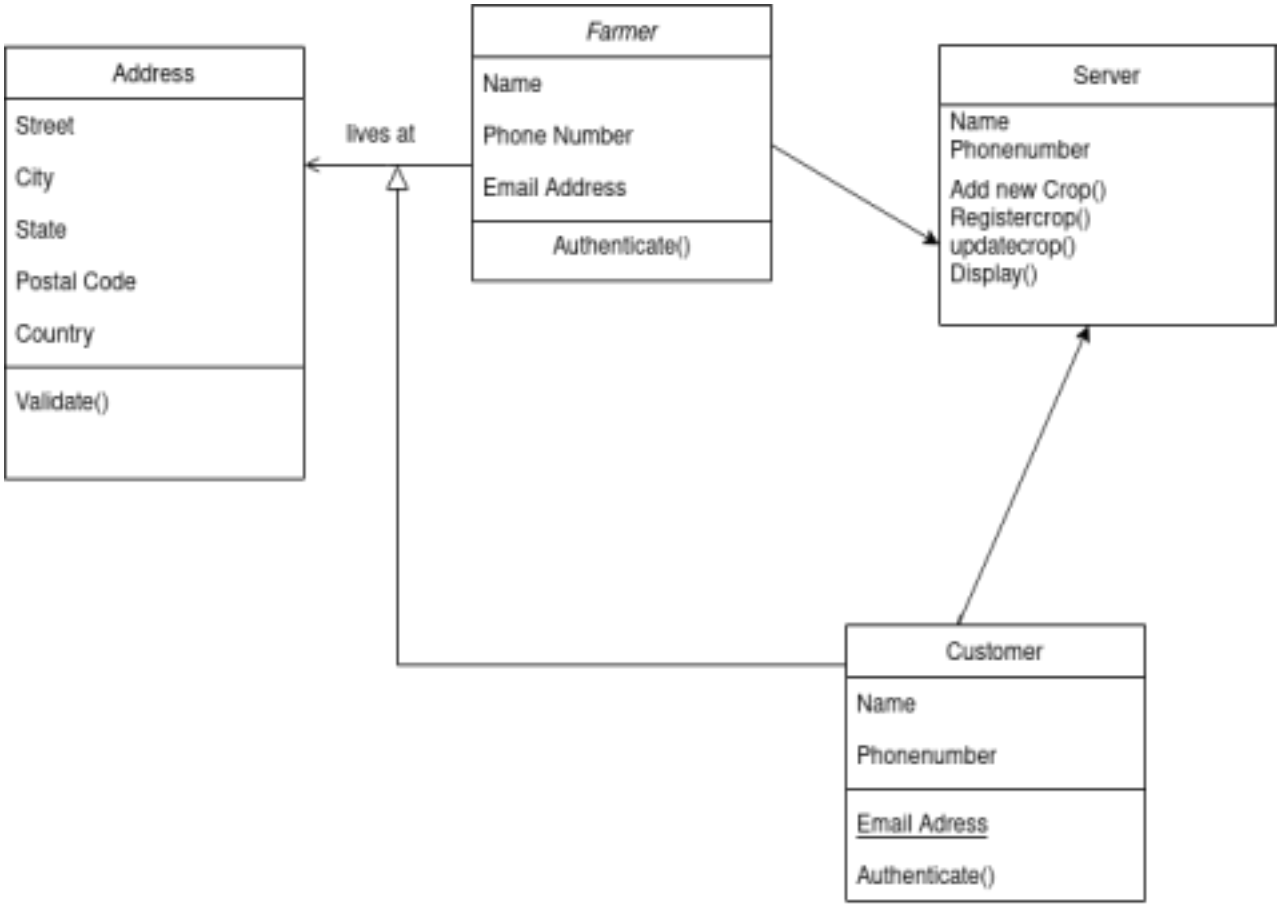
6.3.1 Usecase Diagram

Use case diagrams are a group of use cases, actors and their relationships. They represent the employment case read of a system.

So use case diagram is employed to explain the relationships among the functionalities and their internal/external controllers. These controllers are called actors.

Class Diagrames



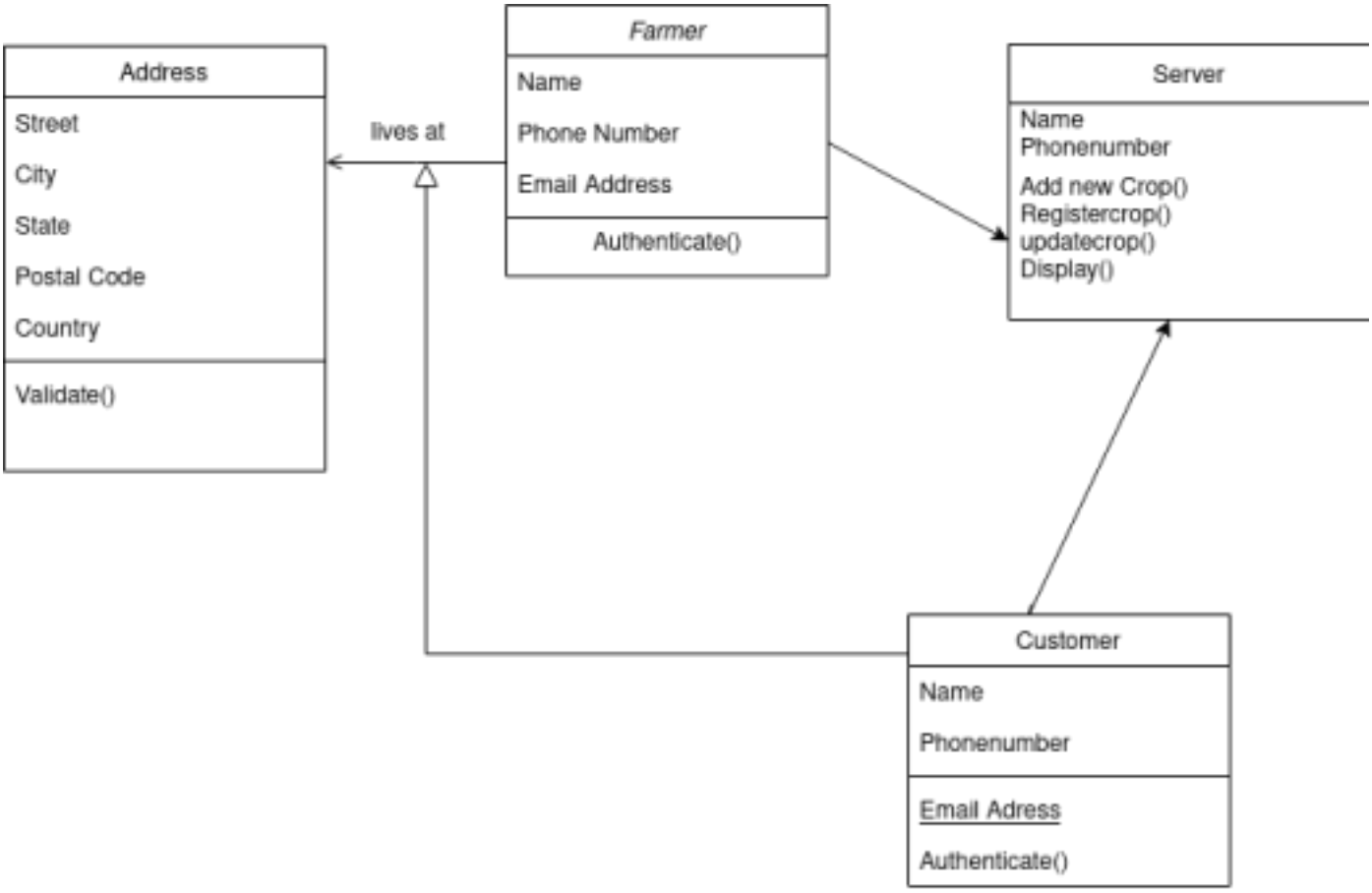


Class diagrams area unit the foremost common diagrams employed in UML. category diagram consists of categories, interfaces, associations and collaboration. Category diagrams primarily represent the thing directed read of a system that is static in nature. Active category is employed in a very category diagram to represent the concurrency of the system. Class diagram represents the thing orientation of a system. Therefore it's usually used for development purpose. This can be the foremost wide used diagram at the time of system construction.

Fig. 6.3: Class Diagram

7. System Development and Environment 7.1

IMPLEMENTATION



7.1.1 INTRODUCTION

The basic idea of selling crops after harversting them to a customer is through a dealer.If we are able to linkup the farmers and customers that are in a reachable distance or if

we provide a delivery boy for this purpose both the farmer and customer will get benefited through this application. To do this process the farmer and customer has to register their details in the application and the farmer has to register the crop.And after the harvesting time the crop and farmer will be displayed to the nearest customers.

7.1.2 MODULES:

This Android project has the 2 modules

1. Farmer
2. Customer

MODULES DESCRIPTION

USER

This module provides the Android UI for the users to register a new crop and add updations of the particular crop they have registered for.

MAIL SERVER:

This module will allow the user to choose the nearest farmer they have, and make deals.

7.1.3 FUNCTIONALAND NON-FUNCTIONAL REQUIREMENTS

Function Requirements

The functional requirements of the system are very important role in the design of any kind of system the following are functional requirements.

- Allows user to select the farmer/customer role
- Allows user to enter email & Password
- Allows users to phone number
- Allows farmerrs to register for different kind of crops
- Allows farmers to register to multiple crops.
- Allows mail server to verify the users mail authentication.

Non Functional Requirements:

In this system the Non-functional Requirements are as follows

- ❖ 24x7 Availability
- ❖ Durability
- ❖ Improving the performance of the system using Better Component.

7.2. SOFTWARE ENVIRONMENT

7.2.1 INTRODUCTION

Android provides a rich application framework that allows you to build innovative apps and games for mobile devices in a Java language environment. The documents listed in the left navigation provide details about how to build apps using Android's various APIs. If you're new to Android development, it's important that you understand the following fundamental concepts about the Android app framework:

Apps provide multiple entry points

Android apps are built as a combination of distinct components that can be invoked individually. For instance, an individual activity provides a single screen for a user interface, and a service independently performs work in the background.

From one component you can start another component using an intent. You can even start a component in a different app, such as an activity in a maps app to show an address. This model provides multiple entry points for a single app and allows any app to behave as a user's "default" for an action that other apps may invoke.

7.2.2 STEP BY STEP PROCEDURE

7.2.2.1 Install Android SDK and Emulator (AVDs)

Android is a large and fast-growing segment of the mobile phone market. With potentially 350,000 users activating a new Android phone every day and multiple Android App Stores popping up (Google's Android Market was recently revamped, and now other markets are coming online, like Amazon's Android App Store), now is a great time to jump into Android development.

This walk-through will get you started installing the Android Software Development Kit (Android SDK), installing and configuring the Eclipse IDE for Android development, and choosing and installing Android Virtual Devices (AVDs) to emulate the Android environment right on your local computer. After following these steps, you will be ready to create your first Android application!

1. Download the Android Software Development Kit (SDK)

http://developer.android.com/sdk/index.html

The very first step is to download the Android Software Development Kit (SDK) that will let you emulate Android on your local computer. It is not too large (only ~30MB, compared to the monolithic XCode/iPhone SDK, which is almost 7.2GB!). From the Android SDK Download Page, make sure to choose the version that is correct for your operating system.

After your Android SDK download is complete, unzip and move the new folder to a permanent location (*not* your downloads directory). I use a folder in my home directory (~/Android/android-sdk-mac_x86/) but you can move it anywhere you would like. There is no wrong location. Wherever you choose will hereby known as $ANDROID for future reference.

2. Install Android SDK Components

http://developer.android.com/sdk/adding-components.html

Android is packaged in such a way that the base Android SDK (downloaded in Step #1) is distinct and separate from each API version of the Android SDK. This means that for each version we want to support (from Step #6), we need to download a separate Android SDK for that version. This can be very annoying when installing (notice how many steps we have done by now), but in the long-run is a very beneficial design for us Android Developers.

As new API versions are added and old API versions are phased out, we can install/uninstall the APIs as compnents, rather than a single huge download like XCode is for iPhone (7.2GB! I just can't get over that! Who has a 7.2GB download for a minor version change?!)

We need to download the Software Development Kits (SDKs) for the Android versions that we want to support.

To do this, we can use the Eclipse IDE + Android ADT that we installed in Step #3. From within Eclipse:

- Click on "Window" then "Android SDK and AVD Manager"
- In "Available packages", select the platforms you want to support. You can either choose all, or pick-and-choose what you want to develop for. For example, 2.1, 2.2, and 2.3.3 are all I care about, so I am using API 7, 8, and 10. In the "Android Repository" package, I checked the boxes next to:
    - Android SDK Platform-tools, revision 3
    - SDK Platform Android 2.3.3, API 10, revision 1
    - SDK Platform Android 2.2, API 8, revision 1
    - SDK Platform Android 2.1, API 7, revision 1
    - Samples for SDK API 10, revision 1
    - Samples for SDK API 8, revision 1
    - Samples for SDK API 7, revision 1
    - Android Compatibility package, revision 1
- In the "Third party Add-ons", decide what you are interested in. If you are going to be using Google Maps (or anything Google beyond Android), you want to install their APIs. If you want their licensing / billing packages, get those too. I checked the boxes next to:
    - Google APIs by Google Inc., Android API 10, revision 1
    - Google APIs by Google Inc., Android API 8, revision 1
    - Google APIs by Google Inc., Android API 9, revision 1
    - Google Market Licensing package, revision 1
    - Google Market Billing package, revision 1
- Choose "Install Selected", then the "Accept All" radio button, then "Install". This may take a while. If it seems like your download has paused, check for any confirmation

dialogs that you need to click “Accept” to. These can sometimes be hiding in the background.

8. Create Your Android Virtual Devices (AVDs)

http://developer.android.com/guide/practices/screens_support.html#testing

Last but not least, we need to create Android Virtual Devices (AVDs) that will be our Android Emulators for running and testing our Android applications on our local computer. In the same “Android SDK and AVD Manager” from Step #7, choose “Virtual Devices” on the left and create “New…” ones. I like to create AVDs to represent different Android versions that I want to test, as well as different hardware specs and screen densities my users are likely to be using.

The main idea is to test different versions of the Android API, as well as different screen resolutions and densities. I tend to pair older versions of Android (most likely running on older hardware) with lower screen densities, and new versions of Android (most likely running on newer hardware) with better screen resolutions.

7.2.3 ANDROID LIBRARIES

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access. A summary of some key core Android libraries available to the Android developer is as follows −

android.app − Provides access to the application model and is the cornerstone of all Android applications.

android.content − Facilitates content access, publishing and messaging between applications and application components.

android.database − Used to access data published by content providers and includes SQLite database management classes.

android.opengl − A Java interface to the OpenGL ES 3D graphics rendering API.

android.os − Provides applications with access to standard operating system services including messages, system services and inter-process communication.

android.text − Used to render and manipulate text on a device display.

android.view − The fundamental building blocks of application user interfaces.

android.widget − A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.

android.webkit − A set of classes intended to allow web-browsing capabilities to be built into applications.

Having covered the Java-based core libraries in the Android runtime, it is now time to turn our attention to the C/C++ based libraries contained in this layer of the Android software stack.

Android Runtime

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

The Android framework includes the following key services −

Activity Manager − Controls all aspects of the application lifecycle and activity stack.

Content Providers − Allows applications to publish and share data with other applications.

Resource Manager − Provides access to non-code embedded resources such as strings, color settings and user interface layouts.

Notifications Manager − Allows applications to display alerts and notifications to the user.

View System − An extensible set of views used to create application user interfaces.

Applications

You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Games etc.

Application components are the essential building blocks of an Android application. These components are loosely coupled by the application manifest file AndroidManifest.xml that describes each component of the application and how they interact.

There are following four main components that can be used within an Android application:

Sr.No	Components & Description
1	**Activities** They dictate the UI and handle the user interaction to the smart phone screen.
2	**Services** They handle background processing associated with an application.
3	**Broadcast Receivers** They handle communication between Android OS and applications.
4	**Content Providers** They handle data and database management issues.

Table 7.2.1: Components of Android

Activities

An activity represents a single screen with a user interface,in-short Activity performs actions on the screen. For example, an email application might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. If an application has more than one activity, then one of them should be marked as the activity

that is presented when the application is launched. An activity is implemented as a subclass of Activity class as follows −

public class MainActivity extends Activity {

}

Services

A service is a component that runs in the background to perform long-running operations. For example, a service might play music in the background while the user is in a different application, or it might fetch data over the network without blocking user interaction with an activity. A service is implemented as a subclass of Service class as follows −

```
public class MyService extends Service {

}
```

Broadcast Receivers

Broadcast Receivers simply respond to broadcast messages from other applications or from the system. For example, applications can also initiate broadcasts to let other applications know that some data has been downloaded to the device and is available for them to use, so this is broadcast receiver who will intercept this communication and will initiate appropriate action. A broadcast receiver is implemented as a subclass of BroadcastReceiver class and each message is broadcaster as an Intent object.

```
public class MyReceiver extends BroadcastReceiver {

public void onReceive(context,intent){}

}
```

Content Providers

A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the ContentResolver class. The data may be stored in the file system, the database or somewhere else entirely.

A content provider is implemented as a subclass of ContentProvider class and must implement a standard set of APIs that enable other applications to perform transactions.

```
public class MyContentProvider extends ContentProvider {
    public void onCreate(){}

}
```

We will go through these tags in detail while covering application components in individual chapters.

Additional Components

There are additional components which will be used in the construction of above mentioned entities, their logic, and wiring between them. These components are –

S.No	Components & Description
1	**Fragments** Represents a portion of user interface in an Activity.
2	**Views** UI elements that are drawn on-screen including buttons, lists forms etc.
3	**Layouts** View hierarchies that control screen format and appearance of the views.
4	**Intents** Messages wiring components together.
5	**Resources** External elements, such as strings, constants and drawable pictures.
6	**Manifest** Configuration file for the application.

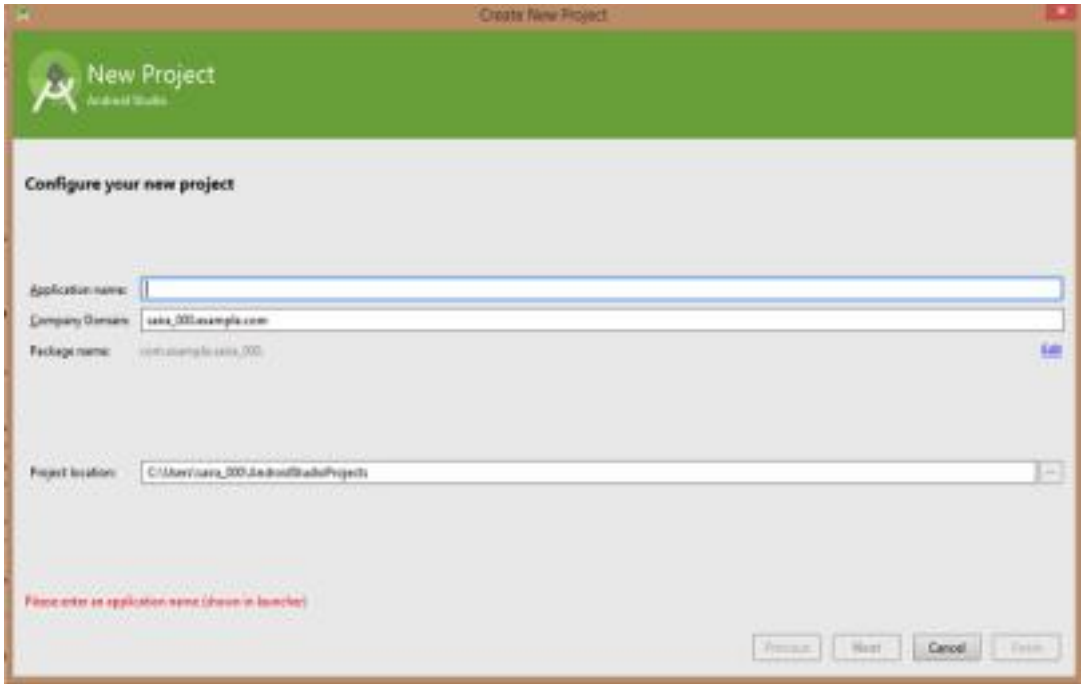
Table 7.2.2: Additional Components of Android

Create Android Application

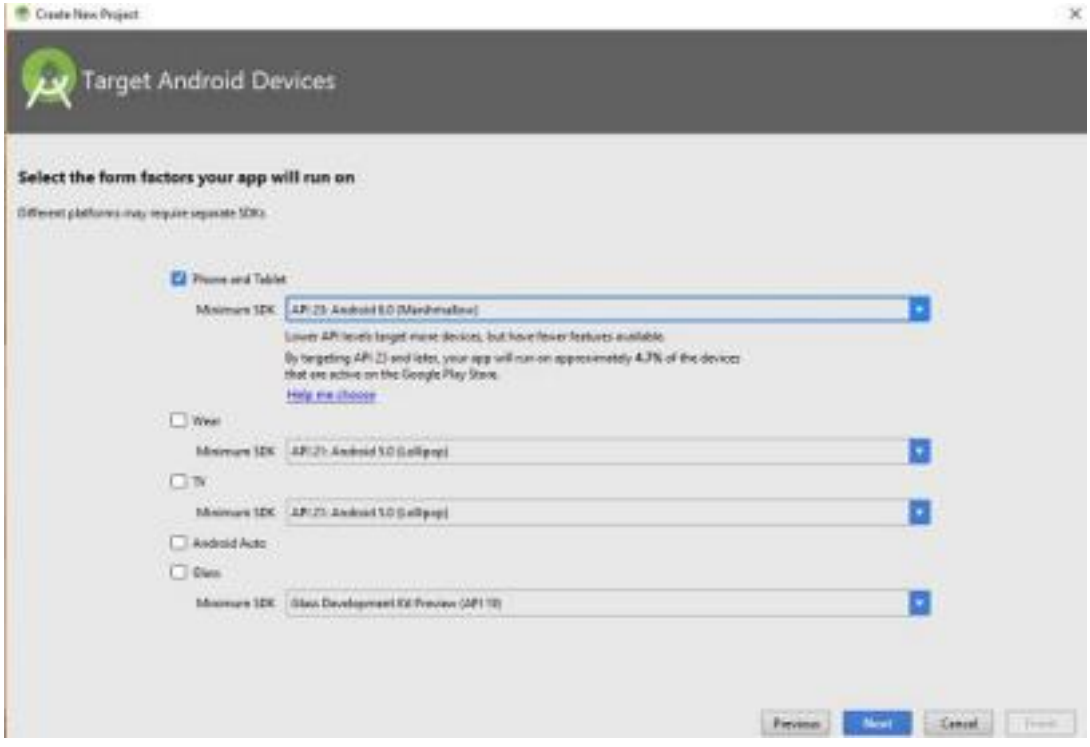
The first step is to create a simple Android Application using Android studio. When you click on Android studio icon, it will show screen as shown below

You can start your application development by calling start a new android studio project. in a new installation frame should ask Application name, package information and location of the

project.−



After entered application name, it going to be called select the form factors your application runs on, here need to specify Minimum SDK, in our tutorial, I have declared as API23: Android 6.0(Mashmallow) −



The next level of installation should contain selecting the activity to mobile, it specifies the default layout for Applications.

At the final stage it going to be open development tool to write the application code.

Anatomy of Android Application

Before you run your app, you should be aware of a few directories and files in the Android project −

Running the Application

Let's try to run our **Hello World!** application we just created. I assume you had created your **AVD** while doing environment set-up. To run the app from Android studio, open one of your project's activity files and click Run icon from the tool bar. Android studio installs the app on your AVD and starts it and if everything is fine with your set-up and application, it will display following Emulator window −

8. TESTING

8.1 TESTING

The purpose of testing is to find errors. Testing is that the method of making an attempt to find each conceivable fault or weakness during a work product. It provides the simplest way to examine the practicality of parts, sub assemblies, assemblies ANd/or a finished product it's the method of workout code with the intent of making certain that the software package meets its needs and user expectations and doesn't fail in an unacceptable manner. There ar varied sorts of check. every check kind addresses a particular testing demand.

TYPES OF TESTS

1. Unit testing

Unit checking involves the look of test cases that validate that the inner program logic is functioning properly, which program inputs turn out valid outputs. All call branches and internal code flow ought to be valid. it's the testing of individual computer code units of the applying .it is done once the completion of a personal unit before integration. this is often a structural testing, that depends on information of its construction and is invasive. Unit checks perform basic tests at part level and test a selected business method, application, and/or system configuration. Unit tests make sure that every distinctive path of a business method performs

accurately to the documented specifications and contains clearly outlined inputs and expected results.

2. Integration testing

Integration tests area unit designed to check integrated software system elements to work out if they really run jointly program. Testing is event driven and is additional involved with the fundamental outcome of screens or fields. Integration tests demonstrate that though the elements were on an individual basis satisfaction, as shown by with success unit testing, the mix of elements is correct and consistent. Integration testing is specifically geared toward exposing the issues that arise from the mix of elements.

Functional test

Functional tests offer systematic demonstrations that functions tested square measure out there as such by the business and technical necessities, system documentation, and user manuals.

Functional testing is cantered on the following items:
Valid Input : identified classes of valid input must be accepted. Invalid

Input : identified classes of invalid input must be rejected. Functions :

identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of useful tests is targeted on needs, key functions, or special check cases. Additionally, systematic coverage concerning establish Business method flows; knowledge fields, predefined processes, and ordered processes should be thought of for testing. Before useful testing is complete, further tests area unit known and therefore the effective worth of current tests is decided.

3. System Test

System testing ensures that the whole integrated computer code meets necessities. It tests a configuration to make sure known and certain results. AN example of system take a look ating is that the configuration directed system integration test. System testing is predicated on method descriptions and flows, accentuation pre-driven method links and integration points.

4. White Box Testing

White Box Testing may be a testing during which during which the computer code tester has

information of the inner workings, structure and language of the computer code, or a minimum of its purpose. it's purpose. it's wont to check areas that can't be reached from a recorder level.

5. Black Box Testing

Black Box Testing is testing the software package with none information of the inner workings, structure or language of the module being tested. recording equipment tests, as most different kinds of tests, should be written from a definitive supply document, like specification or necessities document, like specification or necessities document. it's a take a look rating during which the software package beneath test is treated, as a recording equipment.

You cannot “see” into it. The take a look at provides inputs and responds to outputs while not considering however the software package works.

Unit Testing:

Unit checking is typically conducted as a part of a combined code and unit test section of the computer code lifecycle, though it's not uncommon for secret writing and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is that the progressive integration testing of 2 or additional integrated code elements on one platform to supply failures caused by interface defects.

The task of the mixing check is to ascertain that elements or code applications, e.g. elements in a very code or – one accelerate – code applications at the corporate level – move while not error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing may be a essential part of any project and needs vital participation by the tip user. It additionally ensures that the system meets the useful needs.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.2 TEST PROCEDURE

The following are the Testing Methodologies:

- **Unit Testing.**
- **Integration Testing.**
- **User Acceptance Testing.**
- **Output Testing.**
- **Validation Testing.**

Unit Testing

Unit testing focuses verification effort on the littlest unit of software package style that's the module. Unit testing exercises specific methods during a module's management structure to make sure complete coverage and most error detection. This take a look at focuses on every module separately, guaranteeing that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, every module is tested separately and also the module interfaces square measure verified for the consistency with style specification. All necessary process path square measure tested for the expected results. All error handling methods are tested.

Integration Testing

Integration testing addresses the problems related to the twin problems of verification and program construction. Once the code has been integrated a collection of high order tests area unit conducted.

The main objective during this testing method is to require unit tested modules and builds a program structure that has been set advisedly.

The following are the types of Integration Testing:

1) Top Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are

incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2) Bottom-up Integration

This technique begins the development and testing with the modules at rock bottom level within the program structure. Since the modules area unit integrated from all-time low up, process needed for modules subordinate to a given level is usually on the market and also the want for stubs is eliminated. all-time low up integration strategy is also enforced with the subsequent steps:

- ✓ The low-level modules area unit combined into clusters into clusters that perform a selected software package sub-function.
- ✓ A driver (i.e.) the management program for testing is written to coordinate action input and output.
- ✓ The cluster is tested.
- ✓ Drivers area unit removed and clusters area unit combined moving upward within the program structure

The bottom up approaches tests every module severally then every module is module is integrated with a main module and tested for practicality.

8.3 OTHER TESTING METHODOLOGIES

User Acceptance Testing

User Acceptance of a system is that the key issue for the success of any system. The system into consideration is tested for user acceptance by perpetually keeping in-tuned with the possible system users at the time of developing and creating changes where needed.

The system developed provides a friendly interface that may simply be understood even by an individual UN agency is new the system.

Output Testing

After activity the validation testing, future step is output testing of the projected system, since no system might be helpful if it doesn't manufacture such as|the desired} output within the specified format. Asking the users regarding the format needed by them tests the outputs generated or displayed by the system into consideration. thus the output format is taken into account in a pair of ways that – one is on screen and another in written format.

Validation Checking

Validation checks are performed on the following fields.

Text Field

The text field will contain solely the amount of characters lesser than or capable its size. The text fields are alphamerical in some tables and alphabetic in alternative tables. Incorrect entry continually flashes and error message.

Numeric Field

The numeric field will contain solely numbers from zero to nine. associate entry of any character flashes a blunder messages. The individual modules ar checked for accuracy and what it's to perform. every module is subjected to check run together with sample knowledge. The one by one tested modules ar integrated into one system. Testing involves execution the $64000 knowledge info is employed within the program the existence of any program defect is inferred from the output. The testing ought to be planned thus that each one the wants ar one by one tested.

Preparation of Test Data

Taking numerous types of take a look at knowledge will the on top of testing. Preparation of take a look at knowledge plays a significant role within the system testing. once making ready the take a look at knowledge the system beneath study is take a look ated victimization that test knowledge. whereas take a look ating the system by victimization test knowledge errors ar once more uncovered and corrected by victimization on top of testing steps and corrections are noted for future use.

Using Live Test Data:

Live check information area unit people who are literally extracted from organization files. once a system is partly created, programmers or analysts usually raise users to key in a very set of knowledge from their traditional activities. Then, the systems person uses this information as how to partly check the system. In different instances, programmers or analysts extract a collection of live information from the files and have them entered themselves.

It’s troublesome to get live information in spare amounts to conduct intensive testing. And, though it's realistic information can|which will|that may} show however the system will perform for the everyday process demand, forward that the live information entered area unit in reality typical, such information usually won't check all combos or formats which will enter the system. This bias toward typical values then doesn't give real systems check and in reality

ignores the cases possibly to cause system failure.

Using Artificial Test Data:

Artificial check information area unit created entirely for check functions, since they will be generated to check all mixtures of formats and values. In alternative words, the synthetic information, which may quickly be ready by a knowledge} generating utility within the

information systems department, modify the testing of all login and management methods through the program.

The most effective check programs use artificial check information generated by persons apart from people who wrote the programs. Often, associate degree freelance team of testers formulates a testing arrange, exploitation the systems specifications.

The package “Virtual non-public Network” has glad all demands such as per package requirement specification and was accepted.

USER TRAINING

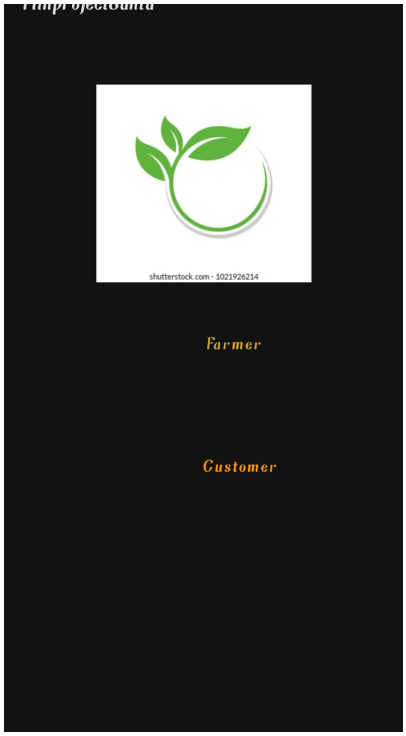
Whenever a brand new system is developed, user coaching is needed to teach them concerning the operating of the system in order that it is place to economical use by those for whom the system has been primarily designed. For this purpose the conventional operating of the project was incontestable to the possible users. Its operating is well graspable and since the expected users are those that have sensible information of computers, the employment of this technique is extremely simple.

MAINTAINENCE

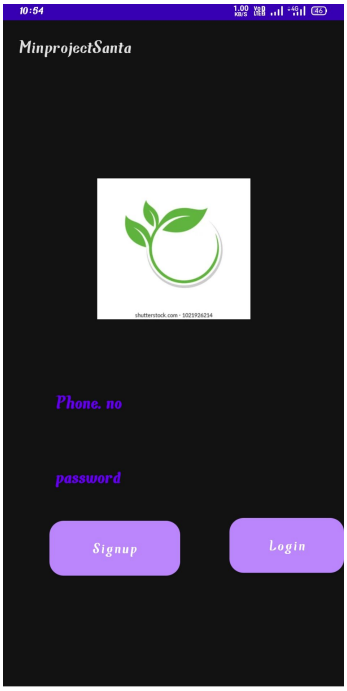
This covers a good vary of activities as well as correcting code and style errors. To scale back the requirement for maintenance within the long-term, we've got additional accurately outlined the user’s needs throughout the method of system development.

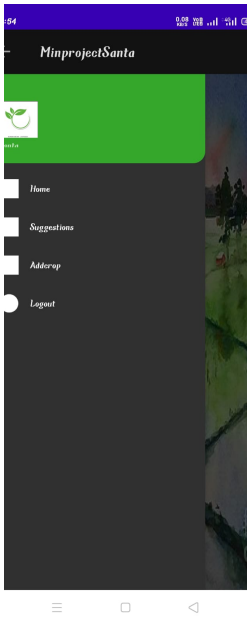
Betting on the necessities, this technique has been developed to satisfy the wants to the biggest attainable extent. With development in technology, it's going to be attainable to feature more options supported the necessities in future. The secret writing and coming up with is straightforward and simple to grasp which is able to create maintenance easier.

9. SCREEN SHOTS


shutterstock.com · 1021926214
Farmer
Customer


MinprojectSanta
Name
email
phone no.
password
confirm password
Address
SUBMIT


10:54
MinprojectSanta
Phone. no
password
Signup
Login


MinprojectSanta
Home
Suggestions
Addcrop
Logout


MinprojectSanta

Product name.....

Product description.

Product price.....

SUBMIT

Register Crop

10. SAMPLE CODE

MainActivity.java

```
package com.example.minprojectsanta;
import androidx.appcompat.app.AppCompatActivity;
Import android.content.Intent;
Import android.os.Bundle;
Import android.view.View;
Import android.widget.Button;
public class MainActivity extends AppCompatActivity {
private Button main_farm,main_cust;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
main_farm=(Button)findViewById(R.id.mainfarm);
main_cust=(Button)findViewById(R.id.maincust);}
main_farm.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent in=new Intent(MainActivity.this,farm_login.class);
startActivity(in);
}
});
main_cust.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
```

```
Intent in=new Intent(MainActivity.this,cust_login.class);
startActivity(in);
}
});
}
```

farmerhome.java

```
package com.example.minprojectsanta;
import androidx.annotation.NonNull;
Import androidx.annotation.RequiresApi;
Import androidx.appcompat.app.ActionBarDrawerToggle;
Import androidx.appcompat.app.AppCompatActivity;
Import androidx.appcompat.widget.Toolbar;
Import androidx.core.view.GravityCompat;
Import androidx.drawerlayout.widget.DrawerLayout;
import android.content.Intent;
Import android.os.Build;
Import android.os.Bundle;
Import android.view.MenuItem;
import com.google.android.material.navigation.NavigationView;
public class farmer_home extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {
DrawerLayout drawerLayout;
NavigationView navigationView;
ActionBarDrawerToggle toggle;
// Toolbar toolbar;
androidx.appcompat.widget.Toolbar toolbar;
// private Object Toolbar;
@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_farmer_home);
```

```
drawerLayout = (DrawerLayout) findViewById(R.id.drawerlayout);
navigationView = (NavigationView) findViewById(R.id.navitem);
// toolbar=(Toolbar)findViewById(R.id.toolbar);
//setSupportActionBar(toolbar);
toggle = new ActionBarDrawerToggle(this, drawerLayout,
R.string.navigation_drawer_open,
R.string.navigation_drawer_close);drawerLayout.addDrawerListener(toggle);
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
// navigationView.bringToFront();
toggle.syncState();
navigationView.setNavigationItemSelectedListener(this);
}
// navigationView.setNavigationItemSelectedListener(this);
@Override
/* public void onBackPressed() {
if(drawerLayout.isDrawerOpen(GravityCompat.START)){
drawerLayout.closeDrawer(GravityCompat.START);
}
else{
super.onBackPressed();}
}*/
public boolean onOptionsItemSelected(@NonNull MenuItem menuItem) {
if (toggle.onOptionsItemSelected(menuItem)) {
return super.onOptionsItemSelected(menuItem);
}
return true;
}
}
// @Override
public boolean onNavigationItemSelected(MenuItem menuItem){
switch (menuItem.getItemId()) {
case R.id.Home:
```

```
break;
case R.id.Suggestions:
Intent intent = new Intent(farmer_home.this, Farmer_suggestions.class);
startActivity(intent);
break;
case R.id.AddCrop:
Intent inten = new Intent(farmer_home.this,
Addnewcrop.class); startActivity(inten);
break;
case R.id.logout:
Intent in = new Intent(farmer_home.this, farm_login.class);
startActivity(in);
break;
}
drawerLayout.closeDrawer(GravityCompat.START);
return true;
}
```

Farmerlogin.java

```
package com.example.minprojectsanta;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;import android.widget.Button;
public class farm_login extends AppCompatActivity {
private Button flogin,fsignup;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_farm_login);
flogin=(Button)findViewById(R.id.farmerlogin);
fsignup=(Button)findViewById(R.id.farmersignup);
flogin.setOnClickListener(new View.OnClickListener() {
```

```
@Override
public void onClick(View view) {
Intent fi=new Intent(farm_login.this,farmer_home.class);
startActivity(fi);
}
});
}
}
```

customerreg.java

```
package com.example.minprojectsanta;
import androidx.appcompat.app.AppCompatActivity;
Import android.content.Intent;
Import android.os.Bundle;
Import android.view.View;
Import android.widget.Button;
Import android.widget.EditText;
import com.google.firebase.auth.FirebaseAuth; import
com.google.firebase.database.DatabaseReference; import
com.google.firebase.database.FirebaseDatabase; public
class cust_reg extends AppCompatActivity { private
Button custregsubmit;
private EditText name,phone,pwd,cpwd,email,adress;
private FirebaseAuth firebaseAuth;
private FirebaseAuth.AuthStateListener authStateListener;
private FirebaseDatabase rootnode;
private DatabaseReference reference;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_cust_reg);
custregsubmit=(Button)findViewById(R.id.cust_reg_submit)
; name=(EditText)findViewById(R.id.cust_reg_name);
```

```
phone=(EditText)findViewById(R.id.cust_reg_phone);
pwd=(EditText)findViewById(R.id.cust_reg_pwd);
cpwd=(EditText)findViewById(R.id.cust_reg_cpwd);
email=(EditText)findViewById(R.id.cust_reg_email);
adress=(EditText)findViewById(R.id.cust_reg_address);
custregsubmit.setOnClickListener(new View.OnClickListener()
{ @Overridepublic void onClick(View view) {
String Name=name.getText().toString();
String Phone=phone.getText().toString();
String Pwd=pwd.getText().toString();
String Cpwd=cpwd.getText().toString();
String Email=email.getText().toString();
String Adress=adress.getText().toString();
rootnode=FirebaseDatabase.getInstance();
reference=rootnode.getReference("Customer_details"); Userhelper
userhelper=new Userhelper(Name,Pwd,Phone,Email,Adress);
reference.child(Phone).setValue(userhelper);
Intent i=new Intent(cust_reg.this,cust_login.class);
startActivity(i);
}
});
}
}
```

customerlogin.java

```
package com.example.minprojectsanta;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
Import android.nfc.Tag;
Import android.os.Bundle;
Import android.util.Log;
Import android.view.View;
```

```
Import android.widget.Button;
Import android.widget.EditText;
Import android.widget.Toast;
import com.google.firebase.database.DataSnapshot;
Import com.google.firebase.database.DatabaseError;
Import com.google.firebase.database.DatabaseReference;
Import com.google.firebase.database.FirebaseDatabase;
Import com.google.firebase.database.Query;
Import com.google.firebase.database.ValueEventListener;
import static android.content.ContentValues.TAG;
public class cust_login extends AppCompatActivity {
private Button custSignup,custLogin;
private EditText loginphone,loginpswd;
private FirebaseDatabase firebaseDatabase;
private DatabaseReference ref,mdatabase;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_cust_login);
custSignup=(Button)findViewById(R.id.custsignup);
custLogin=(Button)findViewById(R.id.custlogin);
loginphone=(EditText)findViewById(R.id.login_phone);loginpswd=(EditText)findViewById(R
.id.login_pswd);
firebaseDatabase=FirebaseDatabase.getInstance();
mdatabase=firebaseDatabase.getReference("Customer_details");
custSignup.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent intent=new Intent(cust_login.this,cust_reg.class);
startActivity(intent);
}
});
```

```
String phnum=loginphone.getText().toString();
String pwd=loginpswd.getText().toString();
custLogin.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
mdatabase.addValueEventListener(new ValueEventListener() {
@Override
public void onDataChange(@NonNull DataSnapshot snapshot) {
if(snapshot.child("Customer_details").child(phnum).exists()) {
Userhelper us =
snapshot.child("Customer_details").child(phnum).getValue(Userhelper.class)
; if (us.getPhone().equals(phnum)) {
if (us.getPwd().equals(pwd)) {
Toast.makeText(cust_login.this, "Not Found",
Toast.LENGTH_SHORT); Intent inten = new Intent(cust_login.this,
cust_homepage.class); startActivity(inten);
} else {
loginpswd.setText("Wrong password");
}
} else {
Toast.makeText(cust_login.this, "Not Found",
Toast.LENGTH_SHORT); }
}
}
@Override
public void onCancelled(@NonNull DatabaseError error) {
Toast.makeText(cust_login.this,"Not Found",Toast.LENGTH_SHORT);
}
});
/*
mdatabase.addListenerForSingleValueEvent(new ValueEventListener() {
@Override
```

```
public void onDataChange(@NonNull DataSnapshot snapshot) {
if(snapshot.child("Customer_details").child(phnum).exists()){
Userhelper
us=snapshot.child("Customer_details").child(phnum).getValue(Userhelper.class);
if(us.getPhone().equals(phnum)){
if(us.getPwd().equals(pwd)){
Toast.makeText(cust_login.this,"Not Found",Toast.LENGTH_SHORT);
Intent inten=new Intent(cust_login.this,cust_homepage.class);
startActivity(inten);
}
else{
loginpswd.setText("Wrong password");
}
}else{
Toast.makeText(cust_login.this,"Not Found",Toast.LENGTH_SHORT);
}
}
}
@Override
public void onCancelled(@NonNull DatabaseError error) {
Toast.makeText(cust_login.this,"Not Found",Toast.LENGTH_SHORT);
}
});
//Query checkuser=(Query)
FirebaseDatabase.getInstance().getReference("Customer_details").orderByChild("phone").equa
lTo(
phnum);
/* Query checkuser=mdatabase.child("Customer_details").orderByChild("Phone");
checkuser.addValueEventListener(new ValueEventListener() { @Override
public void onDataChange(@NonNull DataSnapshot snapshot) {
if(snapshot.exists()){
loginphone.setError(null);
```

```
loginphone.setEnabled(false);
String
Password=snapshot.child(phnum).child("pwd").getValue(String.class);
if(Password.equals(pwd)){
loginpswd.setError(null);
loginpswd.setEnabled(false);
Intent inten=new Intent(cust_login.this,cust_homepage.class);
startActivity(inten);
}
else{
Toast.makeText(cust_login.this,"Not Found",Toast.LENGTH_SHORT);
}
}
else{
Toast.makeText(cust_login.this,"Not found",Toast.LENGTH_SHORT);
}
}
@Override
public void onCancelled(@NonNull DatabaseError error) {
Toast.makeText(cust_login.this,"match not found",Toast.LENGTH_SHORT);
}
});
/* checkuser.addListenerForSingleValueEvent(new ValueEventListener() {
@Override
public void onDataChange(@NonNull DataSnapshot snapshot) {
if(snapshot.exists()){
loginphone.setError(null);
loginphone.setEnabled(false);
String
Password=snapshot.child(phnum).child("pwd").getValue(String.class);
if(Password.equals(pwd)){
loginpswd.setError(null);
```

```
loginpswd.setEnabled(false);
Intent intent=new Intent(cust_login.this,cust_homepage.class);
startActivity(intent);
}
else{
Toast.makeText(cust_login.this,"dEtails not matche",Toast.LENGTH_SHORT);}
}
else{
Toast.makeText(cust_login.this,"dEtails not matche",Toast.LENGTH_SHORT);
}
}
}
});
@Override
public void onCancelled(@NonNull DatabaseError error) {
Toast.makeText(cust_login.this,"dEtails not matche",Toast.LENGTH_SHORT);
}
});*/
// Intent intent=new Intent(cust_login.this,cust_homepage.class);
}
}
```

cropregistration.java

```
package com.example.minprojectsanta;
import androidx.appcompat.app.AppCompatActivity;
Import android.content.Intent;
Import android.os.Bundle;
Import android.view.View;
Import android.widget.ImageView;
public class cropregistration extends AppCompatActivity {
private ImageView grain,millet,pappu;
private ImageView onions,mirch,spices;
private ImageView fruit,veg,leafveg;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_cropregistration)
; grain=(ImageView)findViewById(R.id.grains);
millet=(ImageView)findViewById(R.id.millets);
pappu=(ImageView)findViewById(R.id.pappulu);
onions=(ImageView)findViewById(R.id.Onions);
mirch=(ImageView)findViewById(R.id.mirchi);
spices=(ImageView)findViewById(R.id.biriyani)
; fruit=(ImageView)findViewById(R.id.fruits);
veg=(ImageView)findViewById(R.id.veggies);
leafveg=(ImageView)findViewById(R.id.leafyveggies);
grain.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent intent=new Intent(cropregistration.this,Addnewcrop.class);
intent.putExtra("Category","grains");
startActivity(intent);
}
});
millet.setOnClickListener(new View.OnClickListener() {
@Overridepublic void onClick(View view) {
Intent intent=new Intent(cropregistration.this,Addnewcrop.class);
intent.putExtra("Category","millets");
startActivity(intent);
}
});
pappu.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent intent=new Intent(cropregistration.this,Addnewcrop.class);
```

```
intent.putExtra("Category","pappu");
startActivity(intent);
}
});
onions.setOnClickListener(new View.OnClickListener()
{ @Override
public void onClick(View view) {
Intent intent=new Intent(cropregistration.this,Addnewcrop.class);
intent.putExtra("Category","grains");
startActivity(intent);
}
});
mirch.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent intent=new Intent(cropregistration.this,Addnewcrop.class);
intent.putExtra("Category","mirch");
startActivity(intent);
}
});
spices.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent intent=new Intent(cropregistration.this,Addnewcrop.class);
intent.putExtra("Category","spices");
startActivity(intent);
}
});
fruit.setOnClickListener(new View.OnClickListener()
{ @Override
public void onClick(View view) {
Intent intent=new Intent(cropregistration.this,Addnewcrop.class);
```

```
intent.putExtra("Category","fruit");
startActivity(intent);
}
});
veg.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent intent=new Intent(cropregistration.this,Addnewcrop.class);
intent.putExtra("Category","veggies");
startActivity(intent);
}
});
leafveg.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent intent=new Intent(cropregistration.this,Addnewcrop.class);}
});
}
intent.putExtra("Category","leafveg");
startActivity(intent);
}
```

Addnewcrop.java

```
package com.example.minprojectsanta;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import android.app.ProgressDialog;
Import android.content.Intent;
Import android.net.Uri;
Import android.os.Bundle;
Import android.text.TextUtils;
Import android.view.View;
```

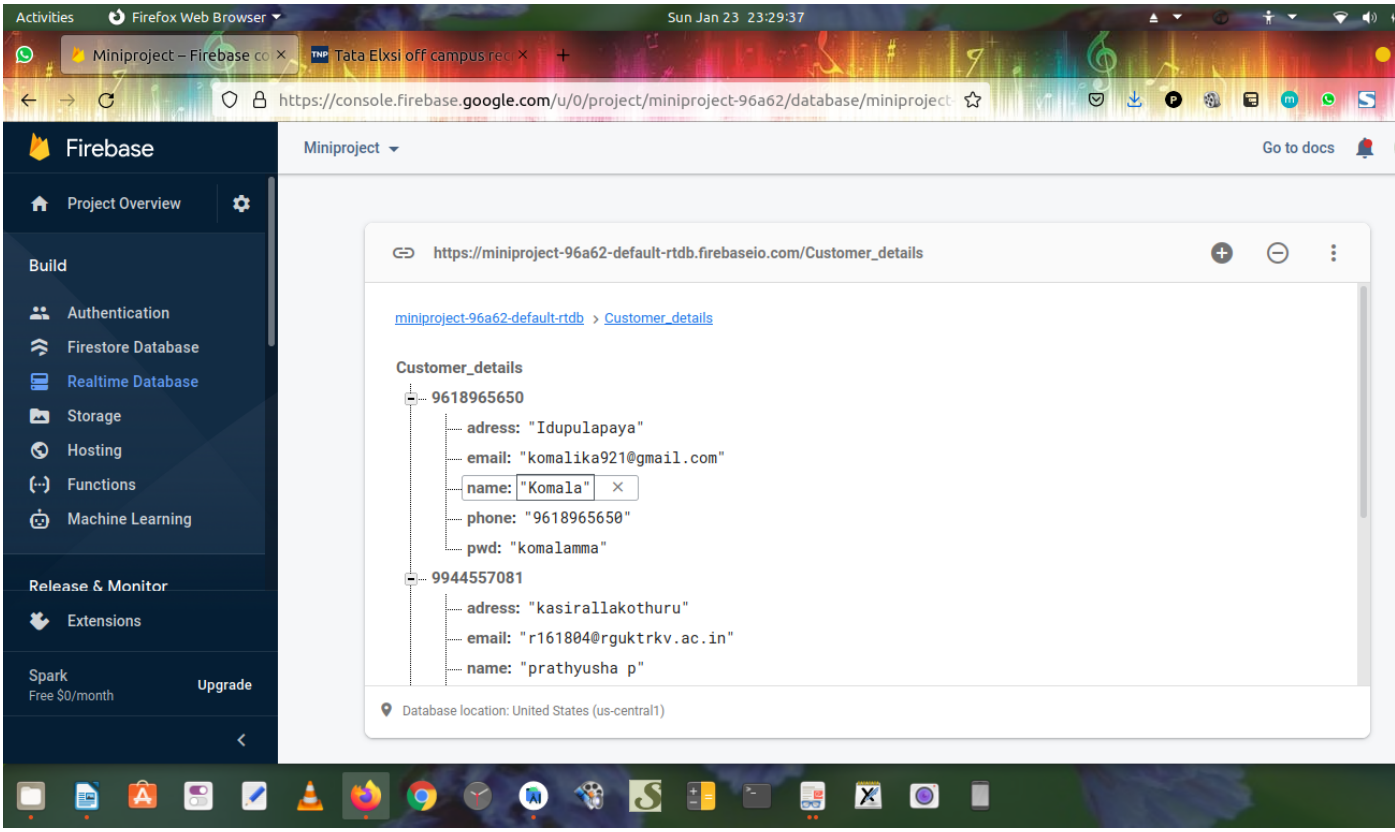
```
Import android.widget.Button;
Import android.widget.EditText;
Import android.widget.ImageView;
Import android.widget.ProgressBar;
Import android.widget.Toast;
import com.google.android.gms.tasks.Continuation;
Import com.google.android.gms.tasks.OnCompleteListener;
Import com.google.android.gms.tasks.OnFailureListener;
Import com.google.android.gms.tasks.OnSuccessListener;
Import com.google.android.gms.tasks.Task;
Import com.google.firebase.database.DatabaseReference;
Import com.google.firebase.database.FirebaseDatabase;
Import com.google.firebase.storage.FirebaseStorage;
Import com.google.firebase.storage.StorageReference;
Import com.google.firebase.storage.UploadTask;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.HashMap;
public class Addnewcrop extends AppCompatActivity {
private String categoryname, description, pname, price, savedate, savetime, prk;
private EditText proname, prodes, proprice;
private ImageView pic;
private Button prodsubmit;
private static final int GALLERYPICK = 1;
private Uri imageuri;
private String downloadimageurl;
private StorageReference imagestore;
private DatabaseReference productref;
private ProgressDialog loadingbar;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_addnewcrop);}
categoryname = getIntent().getExtras().get("Category").toString();
prodsubmit =(Button)findViewById(R.id.addnewcropsubmit);
proname = (EditText) findViewById(R.id.productname);
prodes = (EditText) findViewById(R.id.productdescription);
proprice = (EditText) findViewById(R.id.productprice);
loadingbar=new ProgressDialog(this);
imagestore = FirebaseStorage.getInstance().getReference().child("Promage");
productref=FirebaseDatabase.getInstance().getReference().child("Products");
pic.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
opengallery();
}
});
prodsubmit.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
validateProductData();
}
});
private void opengallery() {
Intent galleryIntent = new Intent();
galleryIntent.setAction(Intent.ACTION_GET_CONTENT);
galleryIntent.setType("image/*");
startActivityForResult(galleryIntent, GALLERYPICK);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
super.onActivityResult(requestCode, resultCode, data);
if (requestCode == GALLERYPICK && resultCode == RESULT_OK && data != null) {
imageuri = data.getData();
```

```
pic.setImageURI(imageuri);
}
}
private void validateProductData() {
description = prodes.getText().toString();
price = proprice.getText().toString();
pname = proname.getText().toString();
if (imageuri == null) {
Toast.makeText(Addnewcrop.this, "Add image", Toast.LENGTH_SHORT);
} else if (TextUtils.isEmpty(description)) {
Toast.makeText(Addnewcrop.this, "Add description", Toast.LENGTH_SHORT);
} else if (TextUtils.isEmpty(price)) {
Toast.makeText(Addnewcrop.this, "Add Price", Toast.LENGTH_SHORT);
} else if (TextUtils.isEmpty(pname)) {
Toast.makeText(Addnewcrop.this, "Add product name",
Toast.LENGTH_SHORT); } else {
storeImage();
}
}private void storeImage() {
loadingbar.setTitle("ImageSaving");
loadingbar.setMessage("Please wait while saving the image");
loadingbar.setCanceledOnTouchOutside(false);
loadingbar.show();
Calendar calendar = Calendar.getInstance();
SimpleDateFormat currentdate = new SimpleDateFormat("MM
dd,yyyy"); savedate = currentdate.format(calendar.getTime());
SimpleDateFormat currenttime = new SimpleDateFormat("HH:mm:ss
a"); savetime = currenttime.format(calendar.getTime());
prk = savedate + savetime;
StorageReference filepath = imagestore.child(imageuri.getLastPathSegment() +
prk); final UploadTask uploadTask = filepath.putFile(imageuri);
uploadTask.addOnFailureListener(new OnFailureListener() {
```

```
@Override
public void onFailure(@NonNull Exception e) {
String msg = e.toString();
Toast.makeText(Addnewcrop.this, "Exception occurred",
Toast.LENGTH_SHORT).show(); loadingbar.dismiss();
}
}).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
@Override
public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
Toast.makeText(Addnewcrop.this, "Success.....",
Toast.LENGTH_SHORT).show(); Task<Uri> uriTask =
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();
}
downloadimageurl = filepath.getDownloadUrl().toString();
return filepath.getDownloadUrl();
}
}).addOnCompleteListener(new OnCompleteListener<Uri>() {
@Override
public void onComplete(@NonNull Task<Uri> task) {
if (task.isSuccessful()) {
downloadimageurl=task.getResult().toString();
Toast.makeText(Addnewcrop.this, "Successfull",
Toast.LENGTH_SHORT).show(); saveproductinfotodatabse();
}
}
});
```

```
}
});
}
private void saveproductinfotodatabse() {
HashMap<String,Object> productmap=new HashMap<>();
productmap.put("pid",prk);
productmap.put("date",savedate);
productmap.put("time",savetime);
productmap.put("description",description);productmap.put("image",downloadimageurl);
productmap.put("category",categoryname);
productmap.put("price",price);
productmap.put("name",pname);
productref.child(prk).updateChildren(productmap).addOnCompleteListener(new
OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void> task) {
if(task.isSuccessful()){
}
}
}
});
Intent intent=new Intent(Addnewcrop.this,cropregistration.class);
startActivity(intent);
loadingbar.dismiss();
Toast.makeText(Addnewcrop.this,"added",Toast.LENGTH_SHORT).show();
}else{
loadingbar.dismiss();
String message=task.getException().toString();
Toast.makeText(Addnewcrop.this,"message",Toast.LENGTH_SHORT).show();
}
```



Firebasedatabase is used for database purpose

11. CONCLUSION AND FUTURE ENHANCEMENT

11.1 CONCLUSION

This android app fulfil the requirement of the farmer to get a good profit and the customer to get a healthy food at a reasonable price.

11.2 FUTURE ENHANCEMENT

This Android App fullfils the requirement of the farmer to get a good profit and the customer to get a healthy food at a reasonable price.