## CROP PROGRESSION

***Project submitted in partial fulfillment of the requirements for the award of the degree of***

# BACHELOR OF TECHNOLOGY

### IN

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

**B KEERTHANA (18C91A0511)**

**B MANISH (18C91A0512)**

**K ANTONY (18C91A0545)**

**M MOHITH (18C91A0556)**

**Under the Esteemed Guidance of**

**Dr. B.NARSIMHA**

PROFESSOR & HEAD OF THE DEPARTMENT



### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE**

**(COLLEGE OF ENGINEERING)**

***(Approved by AICTE New Delhi, Permanently Affiliated to JNTU Hyderabad, Accredited by NAAC with ‘A’ Grade)***

**Bogaram (V), Keesara (M), Medchal District -501 301.**

**2021-2022**

**HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE**

#### (COLLEGE OF ENGINEERING)

***(Approved by AICTE New Delhi, Permanently Affiliated to JNTU Hyderabad, Accredited by NAAC with ‘A’ Grade)***

**Bogaram (V), Keesara (M), Medchal Dist-501301.**

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



**CERTIFICATE**

This is to certify that the mini project entitled “CROP PROGRESSION” is being submitted by KEERTHANA(18C91A0511),B.MANISH(18C91A0512),K.ANTONY(18C91A0545),M.MOHIT

(18C91A0556), in Partial fulfillment of the academic requirements for the award of the degree of Bachelor of Technology in “COMPUTER SCIENCE AND ENGINEERING” HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE, JNTU Hyderabad during the year 2021- 2022.

**INTERNAL GUIDE HEAD OF THE DEPARTMENT**

DR. NARSIMHA BANOTHU DR .B.NARSIMHA M.Tech, Ph.D.

Professor & HoD Professor & HoD

Dept. of Computer Science & Engineering Dept. of Computer Science & Engineering

**EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, who’s constant guidance and encouragement crowns all effort with success.

I take this opportunity to express my profound gratitude and deep regards to My Guide **Dr. B. Narsimha , Professor & HoD**, Dept. of Computer Science & Engineering, Holy Mary Institute of Technology & Science for his / her exemplary guidance, monitoring and constant encouragement throughout the project work.

My special thanks to **Dr. B. Narsimha, Head of the Department**, Dept. of Computer Science & Engineering, Holy Mary Institute of Technology & Science who has given an immense support throughout the course of the project.

I also thank to **Dr. P. Bhaskara Reddy,** the **honorable Director** of my college Holy Mary Institute of Technology & Science for providing me the opportunity to carry out this work.

At the outset, I express my deep sense of gratitude to the beloved **Chairman A. Siddartha Reddy** of **Holy Mary Institute of Technology & Science**, for giving me the opportunity to complete my course of work.

I am obliged to **staff members** of Holy Mary Institute of Technology & Science for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Last but not the least I thank **ALMIGHTY** and My **Parents**, and **Friends** for their constant encouragement without which this assignment would not be possible.

**B KEERTHANA (18C91A0511)**

**B MANISH (18C91A0512)**

**K ANTONY (18C91A0545)**

**M MOHITH (18C91A0556)**

**DECLARATION**

This is to certify that the work reported in the present project titled **“CROP PROGRESSION”** is a record of work done by me in the Department of Computer Science & Engineering, Holy Mary Institute of Technology and Science.

No part of the thesis is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred in the text the reported are based on the project work done entirely by me not copied from any other source.

**B KEERTHANA (18C91A0511)**

**B MANISH (18C91A0512)**

**K ANTONY (18C91A0545)**

**M MOHITH (18C91A0556)**

**CONTENTS**

ABSTRACT

Chapter Name of the Chapter Page No.

### INTRODUCTION

Problem Statement…………………………………………………… 2

Objectives……………………………………………………………. 2

Motivation…………………………………………………………… 2

Existing System……………………………………………………… 3

Proposed System…………………………………………………….. 3

### 2 LITERATURE SURVEY

Existing System……………………………………………………... 4

Proposal System…………………………………………………….. 5

Applications………………………………………………………… 5

Summary……………………………………………………………. 6

# 3 SOFTWARE REQUIREMENTS

# SPECIFICATIONS

Software Requirements……………………………………………..7

Hardware Requirements………………………………….…………7

### SYSTEM DESIGN

System Architecture…………………………………………….8-11

**5** **IMPLEMENTATION**

Environmental Setup…………………………………………… 12

Module Description……………………………………………. 13

Software Description………………………………………........14

Sample Code…………………………………………………….15

### SYSTEM TESTING

Tests…………………………………………………………….45

### 7 RESULTSCREEN SHOTS…………………………..46-47

### 8 CONCLUSION……………………………………….48

### 9 BIBLIOGRAPHY…………………………………... 50

LIST OF TABLES I

LIST OF FIGURES II

LIST OF SCREEN SHOTS III

**LIST OF FIGURES:**

Fig.4.1: flow chart of crop progression**……………………………………………………………**8

Fig.4.2: Use case diagram of crop progression………………………………………………10

Fig.4.3: Sequence Diagram…………………………………………………………………..11

**LIST OF SCREEN SHOTS**

Fig. 6.1: Home Page Of Crop Progression………………………………………………………….57

Fig. 6.2: Roof Gardening……………………………………………………………………58

Fig. 6.3: Land Cultivation…………………………………………………………………..58

Fig. 6.4: Chat Bot…………………………………………………………………………...59

# ABSTRACT

The ability to accurately estimate crop planting date and planting progression has major implication in crop management. Most of people have less knowledge of crop progression our project objective is to provide the basic information about the crop progression and to help our users in growing crop healthily and economically.

In our project we provide information about the stages of crops, the method of cultivation and it also includes chat bot and personal assistant to help the users with their concern.

# INTRODUCTION

Have you ever wondered where our agricultural crops come from? And what were they like thousands of years ago, or hundreds of years ago? Our food crops today are in fact very different from the original wild plants from which they were derived. About 10,000 years BC, people harvested their food from the natural biological diversity that surrounded them, and eventually domesticated crops and animals. During the process of domestication, people began to select better plant materials for propagation and animals for breeding, initially unwittingly, but ultimately with the intention of developing improved food crops and livestock. Over thousands of years farmers selected for desirable traits in crops, and thus improved the plants for agricultural purposes. Desirable traits included crop varieties (also known as cultivars, from "cultivated varieties") with shortened growing seasons, increased resistance to diseases and pests, larger seeds and fruits, nutritional content, shelf life, and better adaptation to diverse ecological conditions under which crops were grown. Over the centuries, agricultural technology developed a broad spectrum of options for food, feed, and fiber production. In many ways, technology reduces the amount of time we dedicate to basic activities like food production, and makes our lives easier and more enjoyable. Everyone is familiar with how transportation has changed over time to be more efficient and safer (Figure 1). Agriculture has also undergone tremendous changes, many of which have made food and fiber production more efficient and safer (Figure 1). For example in 1870, the total population of the USA was 38,558,371 and 53% of this population was involved in farming; in 2000, the total population was 275,000,000 and only 1.8% of the population was involved in farming. There are negative aspects to having so few members of society involved in agriculture, but this serves to illustrate how technological developments have reduced the need for basic farm labor. Many different tools are available for increasing and improving agricultural production. These tools include methods to develop new varieties such as classical breeding and biotechnology. Traditional agricultural approaches are experiencing some resurgence today, with renewed interest in organic agriculture; an approach that does not embrace the use of genetically engineered crops. The role that genetic engineering stands to play in sustainable agricultural development is an interesting topic for the future. As with the development of any new technology there are concerns about associated risks, and agricultural biotechnology is no exception. All crops developed using genetic engineering are subjected to extensive safety testing before being released for commercial use. Risk assessments are conducted for these new varieties, and only those that are safe for human use are released. Some concerns arise through people not fully understanding the reporting of risk. Many consider any level of risk unacceptable. Some prefer the application of the precautionary principle when releasing new technology, but this is not a realistic interpretation of what risk assessments tell us (See information presented by Land Grant Universities of the USA). Extensive risk assessment and safety testing of crops developed through the use of genetic engineering has shown that there are no varieties in use that pose risks to consumers. This is not to say that new varieties should not be carefully examined for safety; each case should be considered on its unique merits. In this app we have chatbot for the help assistance we can know lot of things from web through this chatbot and it is most helpful source for the people and we also have virtual assistance for the people who are uneducated not for them but also it can react to our day crop habitation

## PROBLEM STATEMENT

## 

As We All Know That There Is Rapid Increase In The Requirement Of Food And Demand For Agriculture Is Increasing.

Day By Day. Also Most Of The People Are Showing Interest Towards Agriculture And Are Trying To Learn Farming. But There Is Lot Of Information Available So People Get Confused With This Information. So, Our Team Decided To Create An Application Which Provides Detailed Information Of Crops Which Can Be Grown In Roofs And Land. So That People Select Their Way Of Farming And The Production Of Food Increases.

## OBJECTIVE

The main objective of this application is to develop a system with improved facilities and provide convenience to the user about crop cultivation. To educate the people who have less knowledge about farming and help them to grow the crops in healthy and economical way and educate them in different techniques of farming based on their interest. And to develop a user-friendly app which provides complete information about crop cultivation at one place. This project is intended to help the user with information about stages of crop.

## MOTIVATION

Demand of farming increases in coming years, most of the people will show interest towards farming /roof gardening this application will provide the necessary information about land cultivation and roof gardening.

Crop progression has been developed to provide user friendly UI and UX which is easy to use and understand.

## EXISTING SYSTEM

There are lot of problems and limitations associated with existing system like UI is not at all user friendly and requires lot of effort and hard work to gain knowledge about crop cultivation and there is lot of information available in web about crop cultivation so people get confused and tired with that information and won’t be able to get interest on the agriculture and crop cultivation as searching in web is time consuming. Some sites either have the information related to land cultivation and roof gardening but not both at one place. So, if the information is easily available to the user in an organized manner then it will be convenient to the user to understand and gain knowledge about crop cultivation.

## PROPOSED SYSTEM

Crop Progression is an android based application which is developed using kotlin ,java for logic development and XML for user interface. The proposed system will solve all the problems associated with the existing system. This application will provide complete info about crop cultivation and user know both about land cultivation and roof gardening at one place in an organized manner with user friendly interface and it also includes features like chat bot and personal assistant which will help the user while using the app and both these features are very much useful and provides much more convenient than searching in the web about crop cultivation . And also helps the user to grow the crops in healthy and economical way and educate them in different techniques of farming based on their interest.

# 2. LITERATURE SURVEY

This application is mainly developed to provide complete information about roof gardening and land cultivation and covers all the information related to both at one place in a well-organized manner. Our food crops today are in fact very different from the original wild plants from which they were derived. About 10,000 years BC, people harvested their food from the natural biological diversity that surrounded them, and eventually domesticated crops and animals. During the process of domestication, people began to select better plant materials for propagation and animals for breeding, initially unwittingly, but ultimately with the intention of developing improved food crops and livestock.Over thousands of years farmers selected for desirable traits in crops, and thus improved the plants for agricultural purposes.

## EXISTING SYSYTEM

In the past, it was difficult for farmers and others to correlate production techniques and crop yields with land variability. This limited their ability to develop the most effective soil or plant treatment .There are lot of problems and limitations associated with existing system like UI is not at all user friendly and requires lot of effort and hard work to gain knowledge about crop cultivation and there is lot of information available in web about crop cultivation so people get confused and tired with that information and won’t be able to get interest on the agriculture and crop cultivation as searching in web is time consuming. Some sites either have the information related to land cultivation and roof gardening but not both at one place. So, if the information is easily available to the user in an organized manner then it will be convenient to the user to understand and gain knowledge about crop cultivation.

## PROPOSED SYSTEM

Crop Progression is an android based application which is developed using kotlin, java for logic development and XML for user interface. The proposed system will solve all the problems associated with the existing system. This application will provide complete info about crop cultivation and user can know both about land cultivation and roof gardening at one place in an organized manner with user friendly interface and it also includes features like chat bot and personal assistant which will help the user while using the app and both these features are very much useful and provides much more convenient than searching in the web about crop cultivation. And also helps the user to grow the crops in healthy and economical way and educate them in different techniques of farming based on their interest.

## APPLICATIONS

There are many applications associated with this app which will increase the ability to explore about crop cultivation .This app is not only target conventional but could also be new levers to uplift other growing in agriculture like family farming and also enhance high transparent farming with advanced technologies.

The user of this app will be able to take assistance and help from chat bot and personal assistant which will overall increase the user experience and provides lot of convenience .And key component of this crop cultivation and agriculture is the use of information technology .If this is done in real time which increases efficiency of the whole process of crop cultivation. One more thing that has been made possible with the advancements of IT is the convenience and ease to access all the information at one place and also in an organized manner

## SUMMARY

So, this crop progression app is an android based application which is developed using kotlin, java, and XML. This project is to educate people who have a passion for agriculture and crop development, this crop progression targets people who have a passion for agriculture and people who want to know the new methods of the plantation, by using this project we can grow plants on top of our roof, this project consists of two virtual bots one is personal assistant and chat bot, these will help to develop our skills in agriculture and helps in day-to-day activity. It also helps the user to grow the crops in healthy and economical way and educate them in different techniques of farming based on their interest and which will lead to healthy and balanced life of user if we cultivate healthy and organic crops.

# 3. SOFTWARE & HARDWARE REQUIREMENTS:

## SOFTWARE REQUIREMENTS:

* Front-end : XML
* Back-end : Kotlin, Java
* OS : Windows 10
* IDE : Android Studio

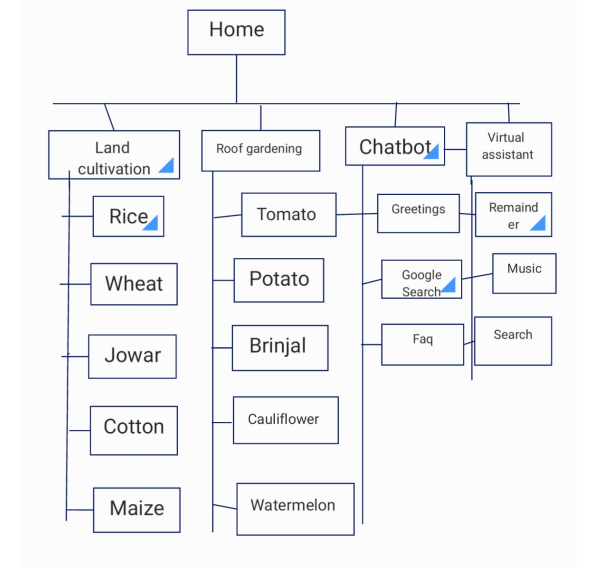
## HARDWARE REQUIREMENTS:

* HDD : 128Gb
* RAM : minimum 4Gb is required
* CPU : i3 or higher version
* INPUT DEVICES : Keyboard, Mouse

# 4. SYSTEM DESIGN

After we finish analyzing our project, we start design our software using different ways. Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.  Systems design could be seen as the application of systems theory to product development. System design is the first design stage in which the basic approach to solving the problem is selected. During system design, the overall structure and style are decided. The system architecture is the overall organization of the system into components called subsystems.

## FLOW CHART :

******

**Fig.4.1 flow chart of crop progression**

The architecture provides the context in which more detailed decisions are made in later design stages. By making high level decisions that apply to the entire system, the system designer partitions the problem into subsystems so that further work can be done by several designers working independently on different subsystems.

The system designer must make the following decisions:

* Organize the system into subsystems.
* Identify the concurrency inherent in the problem.
* Allocate subsystems to processors and tasks.
* Choose an approach for management of data stores.
* Handle access to global resources.
* Choose the implementation of control in software.

### USE CASE DIAGRAM :

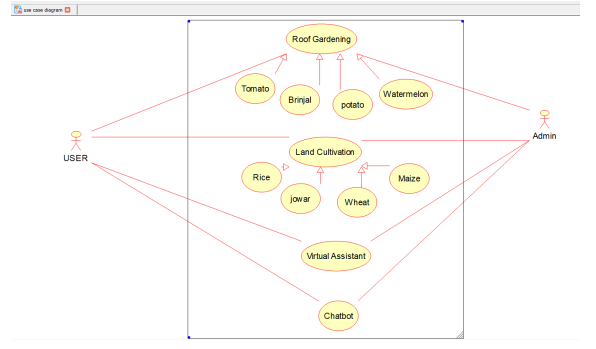
******

Fig. 4.2 : Use case diagram of crop progression

### SEQUENCE DIAGRAM:

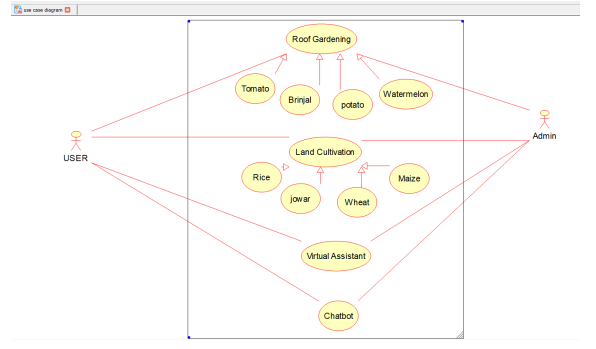
******

Fig.4.3: Sequence Diagram

# 5.IMPLEMENTATION

## ENVIRONMENTAL SETUP:

### Android Environment Setup:

* Install the Java Development Kit. Download and install the Java Development Kit (JDK).
* Download the Android SDK.
* Enable USB debugging on your device.
* Configure the Android SDK path in Unity.
* Download and set up the Android NDK.

### Java Environmental Setup:

Assuming you have installed Java in *c:\Program Files\java\jdk* directory –

* Right-click on 'My Computer' and select 'Properties'.
* Click the 'Environment variables' button under the 'Advanced' tab.
* Now, alter the 'Path' variable so that it also contains the path to the Java executable. Example, if the path is currently set to 'C:\WINDOWS\SYSTEM32', then change your path to read 'C:\WINDOWS\SYSTEM32;c:\Program Files\java\jdk\bin'.

## MODULE DESCRIPTION

Well-structured designs improve the maintainability of a system. A structured system is one that is developed from the top down and modular, that is, broken down into manageable components. In this project we modularized the system so that they have minimal effect on each other.

This application is designed into four independent modules which take care of different tasks efficiently.

* Chat Bot
* Voice Assistant
* Roof Gardening
* Land Cultivation

## Chat Bot:

A chat bot is a software application used to conduct an on-line chat conversation via text or text-to-speech, in lieu of providing direct contact with a live human agent. A chat bot is a type of software that can automate conversations and interact with people through messaging platforms. Designed to convincingly simulate the way a human would behave as a conversational partner, chatbot systems typically require continuous tuning and testing, and many in production remain unable to adequately converse or pass the industry standard Turing test. The term "Chatterbot" was originally coined by Michael Mauldin (creator of the first Verdot) in 1994 to describe these conversational programs.

## Voice Assistant:

Voice Assistant is an artificial intelligence–powered virtual assistant developed by Google that is primarily available on mobile and smart home devices. Unlike the company's previous virtual assistant, Google Now, the Google Assistant can engage in two-way conversations.

## Roof Gardening:

The roof gardening button will show the information about vegetables and others when user clicks on it and it will redirect to new activity which again has some buttons and when user clicks on the specific button it will show specific information.

## Land Cultivation:

The land cultivation button will show the information about vegetables and others when user clicks on it and it will redirect to new activity which again has some buttons and when user clicks on the specific button it will show specific information.

## SOFTWARE DESCRIPTION

Crop Progression is an android based application which is developed using kotlin, java for logic development and XML for user interface. The proposed system will solve all the problems associated with the existing system. This application will provide complete info about crop cultivation and user can know both about land cultivation and roof gardening at one place in an organized manner with user friendly interface and it also includes features like chat bot and personal assistant .

## SAMPLE CODE

### ANDROID MANIFEST.XML:

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.crop">

<uses-permission android:name="com.android.alarm.permission.SET\_ALARM" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.GET\_ACCOUNTS" />

<uses-permission android:name="android.permission.READ\_SMS" />

<uses-permission android:name="android.permission.READ\_CALL\_LOG" />

<uses-permission android:name="android.permission.CALL\_PHONE" />

<uses-permission android:name="android.permission.READ\_CONTACTS" />

<uses-permission android:name="android.permission.SEND\_SMS" />

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity

android:name=".ui.Needs"

android:label="Registration"

android:screenOrientation="portrait" />

<activity

android:name=".ui.va"

android:screenOrientation="portrait" />

<activity

android:name=".ui.ComposeMail"

android:label="Mail"

android:screenOrientation="portrait" />

<activity

android:name=".ui.PhoneModule"

android:screenOrientation="portrait" />

<activity

android:name=".ui.ReminderModule"

android:screenOrientation="portrait" />

<receiver android:name=".ui.AlarmReciver" />

<activity

android:name=".ui.myClass"

android:screenOrientation="portrait" />

<meta-data

android:name="preloaded\_fonts"

android:resource="@array/preloaded\_fonts" />

<activity

android:name=".ui.NoteModule"

android:screenOrientation="portrait" />

<activity

android:name=".ui.NoteModule2"

android:label="@string/title\_activity\_main2"

android:screenOrientation="portrait" />

<activity

android:name=".ui.Music"

android:screenOrientation="portrait" />

<activity android:name=".ui.MainActivty14" />

<activity android:name=".ui.MainActivity13" />

<activity android:name=".ui.MainActivity12" />

<activity android:name=".ui.MainActivity11" />

<activity android:name=".ui.MainActivity10" />

<activity android:name=".ui.MainActivity9" />

<activity android:name=".ui.MainActivity5" />

<activity android:name=".ui.MainActivity8" />

<activity android:name=".ui.MainActivity7" />

<activity android:name=".ui.MainActivity6" />

<activity android:name=".ui.MainActivity4" />

<activity android:name=".ui.MainActivity3" />

<activity android:name=".ui.chatbot"/>

<activity android:name=".ui.home"/>

<activity android:name=".ui.MainActivity" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

### Home.java

package com.example.crop.ui;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.ImageButton;

import androidx.appcompat.app.AppCompatActivity;

import com.example.crop.R;

public class home extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_home);

ImageButton roofgardening = (ImageButton)

findViewById(R.id.imageButton2);

ImageButton landcultivation = (ImageButton)

findViewById(R.id.imageButton4);

ImageButton chatbot = (ImageButton) findViewById(R.id.imageButton);

ImageButton va = (ImageButton) findViewById(R.id.imageButton1);

roofgardening.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i = new Intent(home.this,MainActivity3.class);

startActivity(i);

}

});

landcultivation.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i = new Intent(home.this,MainActivity4.class);

startActivity(i);

}

});

chatbot.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i = new Intent(home.this,chatbot.class);

startActivity(i);

}

});

va.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i = new Intent(home.this,va.class);

startActivity(i);

}

INDEX**.**JAVA

package com.example.crop.ui;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import com.example.crop.R;

public class MainActivity3 extends AppCompatActivity {

Button b1,b2,b3,b4,b5;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main3);

b1 = findViewById(R.id.button);

b2 = findViewById(R.id.button2);

b3 = findViewById(R.id.button3);

b4 = findViewById(R.id.button4);

b5 = findViewById(R.id.button5);

b1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i = new Intent(MainActivity3.this,MainActivity6.class);

startActivity(i);

}

});

b2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent in = new Intent(MainActivity3.this,MainActivity7.class);

startActivity(in);

}

});

b3.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent in1 = new Intent(MainActivity3.this,MainActivity8.class);

startActivity(in1);

}

});

b4.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent in2 = new Intent(MainActivity3.this,MainActivity9.class);

startActivity(in2);

}

});

b5.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent in3 = new Intent(MainActivity3.this,MainActivity5.class);

startActivity(in3);

}

});

}

}

### BOTRESPONSE.KT

package com.example.crop.utils

import com.example.crop.utils.constant.OPEN\_GOOGLE

import com.example.crop.utils.constant.OPEN\_SEARCH

import java.sql.Date

import java.sql.Timestamp

import java.text.SimpleDateFormat

object BotResponse {

fun basicResponses(\_message: String): String {

val random = (0..2).random()

val message =\_message.toLowerCase()

return when {

//Flips a coin

message.contains("flip") && message.contains("coin") -> {

val r = (0..1).random()

val result = if (r == 0) "heads" else "tails"

"I flipped a coin and it landed on $result"

}

//Math calculations

message.contains("solve") -> {

val equation: String? = message.substringAfterLast("solve")

return try {

val answer = SolveMath.solveMath(equation ?: "0")

answer.toString()

} catch (e: Exception) {

"Sorry, I can't solve that."

}

}

//Hello

message.contains("hello") -> {

when (random) {

0 -> "Hello there!"

1 -> "Sup"

2 -> "Buongiorno!"

else -> "error" }

}

//How are you?

message.contains("how are you") -> {

when (random) {

0 -> "I'm doing fine, thanks!"

1 -> "I'm hungry..."

2 -> "Pretty good! How about you?"

else -> "error"

}

}

//What time is it?

message.contains("time") && message.contains("")-> {

val timeStamp = Timestamp(System.currentTimeMillis())

val sdf = SimpleDateFormat("dd/MM/yyyy HH:mm")

val date = sdf.format(Date(timeStamp.time))

date.toString()

}

//Open Google

message.contains("open") && message.contains("google")-> {

OPEN\_GOOGLE

}

//Search on the internet

message.contains("search")-> {

OPEN\_SEARCH

}

//When the programme doesn't understand...

else -> {

when (random) {

0 -> "I don't understand..."

1 -> "Try asking me something different"

2 -> "Idk"

else -> "error"

}

}

### CHATBOT.KT

package com.example.crop.ui

import android.content.Intent

import android.net.Uri

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import androidx.recyclerview.widget.LinearLayoutManager

import com.example.crop.R

import com.example.crop.data.message

import com.example.crop.utils.constant.RECEIVE\_ID

import com.example.crop.utils.constant.SEND\_ID

import com.example.crop.utils.BotResponse

import com.example.crop.utils.constant.OPEN\_GOOGLE

import com.example.crop.utils.constant.OPEN\_SEARCH

import com.example.crop.utils.Time

import kotlinx.android.synthetic.main.activity\_main.\*

import kotlinx.android.synthetic.main.chatbot.\*

import kotlinx.coroutines.\*

class chatbot : AppCompatActivity() {

private val TAG = "chatbot"

//You can ignore this messageList if you're coming from the tutorial,

// it was used only for my personal debugging

var messagesList = mutableListOf<message>()

private lateinit var adapter: MessagingAdapter

private val botList = listOf("cropro","crop")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.chatbot)

recyclerView()

clickEvents()

val random = (0..3).random()

customBotMessage("Hello! Today you're speaking with ${botList[random]},

how may I help?")

}

private fun clickEvents() {

//Send a message

btn\_send.setOnClickListener {

sendMessage()

}

//Scroll back to correct position when user clicks on text view

et\_message.setOnClickListener {

GlobalScope.launch {

delay(100)

withContext(Dispatchers.Main) {

rv\_messages.scrollToPosition(adapter.itemCount - 1)

}

}

}

}

private fun recyclerView() {

adapter = MessagingAdapter()

rv\_messages.adapter = adapter

rv\_messages.layoutManager = LinearLayoutManager(applicationContext)

}

override fun onStart() {

super.onStart()

//In case there are messages, scroll to bottom when re-opening app

GlobalScope.launch {

delay(100)

withContext(Dispatchers.Main) {

rv\_messages.scrollToPosition(adapter.itemCount - 1)

}

}

}

private fun sendMessage() {

val message = et\_message.text.toString()

val timeStamp = Time.timeStamp()

if (message.isNotEmpty()) {

//Adds it to our local list

messagesList.add(message(message, SEND\_ID, timeStamp))

et\_message.setText("")

adapter.insertMessage(message(message, SEND\_ID, timeStamp))

rv\_messages.scrollToPosition(adapter.itemCount - 1)

botResponse(message)

}

}

private fun botResponse(message: String) {

val timeStamp = Time.timeStamp()

GlobalScope.launch {

//Fake response delay

delay(1000)

withContext(Dispatchers.Main) {

//Gets the response

val response = BotResponse.basicResponses(message)

//Adds it to our local list

messagesList.add(message(response, RECEIVE\_ID, timeStamp))

//Inserts our message into the adapter

adapter.insertMessage(message(response, RECEIVE\_ID, timeStamp))

//Scrolls us to the position of the latest message

rv\_messages.scrollToPosition(adapter.itemCount - 1)

//Starts Google

when (response) {

OPEN\_GOOGLE -> {

val site = Intent(Intent.ACTION\_VIEW)

site.data = Uri.parse("https://www.google.com/")

startActivity(site)

}

OPEN\_SEARCH -> {

val site = Intent(Intent.ACTION\_VIEW)

val searchTerm: String? =

message.substringAfterLast("search")

site.data =

Uri.parse("https://www.google.com/search?&q=$searchTerm")

startActivity(site)

}

}

}

}

}

private fun customBotMessage(message: String) {

GlobalScope.launch {

delay(1000)

withContext(Dispatchers.Main) {

val timeStamp = Time.timeStamp()

messagesList.add(message(message, RECEIVE\_ID, timeStamp))

adapter.insertMessage(message(message, RECEIVE\_ID, timeStamp))

rv\_messages.scrollToPosition(adapter.itemCount - 1)

}

}

}

}

### VIRTUALASSISTANT.JAVA

package com.example.crop.ui;

import android.app.FragmentManager;

import android.app.SearchManager;

import android.content.ActivityNotFoundException;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.AsyncTask;

import android.os.Bundle;

import android.preference.PreferenceManager;

import android.speech.RecognizerIntent;

import android.speech.tts.TextToSpeech;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.ImageButton;

import android.widget.TextView;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.crop.R;

import org.xml.sax.InputSource;

import org.xml.sax.XMLReader;

import java.net.URL;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Locale;

import javax.xml.parsers.SAXParser;

import javax.xml.parsers.SAXParserFactory;

/\*\*

\* Created

\*/

public class va extends AppCompatActivity implements TextToSpeech.OnInitListener

{

TextView showUspeak, dateView;

Button help;

public static String module = "";

ImageButton speak;

String command = "blabla";

boolean check = false;

private final int REQ\_CODE = 100;

private TextToSpeech tts;

String welcome, date;

String city = "jabalpur", country = "India";

final String baseUrl =

"https://query.yahooapis.com/v1/public/yql?q=select%20\*%20from%20weather.forecas

t%20where%20woeid%20in%20(select%20woeid%20from%20geo.places(1)%20where%20text%3

D%22" +

city +

"%2C%20" +

country +

"%22)&format=xml&env=store%3A%2F%2Fdatatables.org%2Falltableswithkeys";

String weatherText;

FragmentManager f;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.va);

f = getFragmentManager();

SharedPreferences preferences =

PreferenceManager.getDefaultSharedPreferences(getBaseContext());

welcome = "Hi " + preferences.getString(Needs.NAME, " ") + " what can i

do for u today ? ";

//Grabbing References

showUspeak = (TextView) findViewById(R.id.textViewShow);

help = (Button) findViewById(R.id.buttonHelp);

speak = (ImageButton) findViewById(R.id.imageButtonSpeak);

tts = new TextToSpeech(this, this);

DateFormat df = new SimpleDateFormat("EEE, d MMM yyyy, h:mm a");

date = df.format(Calendar.getInstance().getTime());

dateView = (TextView) findViewById(R.id.textView7Date);

dateView.setText(date);

new MyTask().execute();

//Welcome

showUspeak.setText(welcome);

tts.speak(welcome, TextToSpeech.QUEUE\_FLUSH, null);

speak.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

//Prompt speech input

promptSpeechInput();

check = true;

}

});

help.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

launchModule(Commands.helpModule);

}

});

}

private void launchModule(String commandTolaunch) {

switch (commandTolaunch) {

case Commands.callModule:

Toast.makeText(getBaseContext(), "Call Module",

Toast.LENGTH\_SHORT).show();

Intent intentc = new Intent(va.this, PhoneModule.class);

startActivity(intentc);

break;

case Commands.emergencyModule:

Toast.makeText(getBaseContext(), "Emergency Module",

Toast.LENGTH\_SHORT).show();

Intent intente = new Intent(va.this, PhoneModule.class);

intente.putExtra(Commands.EMERGENCY, true);

startActivity(intente);

break;

case Commands.musicModule:

Toast.makeText(getBaseContext(), "Music Module",

Toast.LENGTH\_SHORT).show();

Intent intentmu = new Intent(va.this, Music.class);

startActivity(intentmu);

break;

case Commands.DATE:

display\_frag d = new display\_frag();

Bundle bundle = new Bundle();

bundle.putString(Commands.DATE, date);

d.setArguments(bundle);

d.show(getFragmentManager(), "sss");

break;

case Commands.TIME:

display\_frag d2 = new display\_frag();

Bundle bundle2 = new Bundle();

bundle2.putString(Commands.DATE, date);

d2.setArguments(bundle2);

d2.show(getFragmentManager(), "sss");

break;

case Commands.remmodule:

Toast.makeText(getBaseContext(), "Reminder Module",

Toast.LENGTH\_SHORT).show();

Intent intentr = new Intent(va.this, ReminderModule.class);

startActivity(intentr);

break;

case Commands.helpModule:

module = "main";

Toast.makeText(getBaseContext(), "Help Module",

Toast.LENGTH\_SHORT).show();

HelpFrag frag = new HelpFrag();

frag.show(f, null);

break;

case Commands.noteModule:

Toast.makeText(getBaseContext(), "Note Module",

Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(va.this, NoteModule.class);

startActivity(intent);

break;

case Commands.weather:

display\_frag d1 = new display\_frag();

Bundle bundle1 = new Bundle();

bundle1.putString(Commands.DATE, weatherText);

d1.setArguments(bundle1);

d1.show(getFragmentManager(), "sss");

break;

default:

try {

Intent intents = new Intent(Intent.ACTION\_WEB\_SEARCH);

intents.putExtra(SearchManager.QUERY, commandTolaunch);

startActivity(intents);

} catch (Exception e) {

}

break;

}

}

/\*\*

\* Showing google speech input dialog

\*/

private void promptSpeechInput() {

Intent intent = new Intent(RecognizerIntent.ACTION\_RECOGNIZE\_SPEECH);

intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE\_MODEL,

RecognizerIntent.LANGUAGE\_MODEL\_FREE\_FORM);

intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE, Locale.getDefault());

intent.putExtra(RecognizerIntent.EXTRA\_PROMPT,

getString(R.string.speech\_prompt));

try {

startActivityForResult(intent, REQ\_CODE);

} catch (ActivityNotFoundException a) {

Toast.makeText(getApplicationContext(),

getString(R.string.speech\_not\_supported),

Toast.LENGTH\_SHORT).show();

}

}

class MyTask extends AsyncTask<Void, Void, Void> {

myXMLWorker doing;

@Override

protected Void doInBackground(Void... params) {

try {

URL web = new URL(baseUrl);

SAXParserFactory saxParserFactory =

SAXParserFactory.newInstance();

SAXParser sp = saxParserFactory.newSAXParser();

XMLReader xmlReader = sp.getXMLReader();

doing = new myXMLWorker();

xmlReader.setContentHandler(doing);

xmlReader.parse(new InputSource(web.openStream()));

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

String command = Commands.TEMP;

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

switch (command) {

case Commands.TEMP:

weatherText = doing.getTemp();

break;

}

}

}

/\*\*

\* Receiving speech input

\*/

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data)

{

super.onActivityResult(requestCode, resultCode, data);

switch (requestCode) {

case REQ\_CODE: {

if (resultCode == RESULT\_OK && null != data) {

ArrayList<String> result = data

.getStringArrayListExtra(RecognizerIntent.EXTRA\_RESULTS);

showUspeak.setText(result.get(0));

//Speak out

speakOut();

}

break;

}

}

}

//Speak Out

private void speakOut() {

String text = showUspeak.getText().toString();

tts.speak(text, TextToSpeech.QUEUE\_FLUSH, null);

command = text;

//Launch Module

if (check) {

launchModule(command);

}

}

@Override

public void onInit(int status) {

if (status == TextToSpeech.SUCCESS) {

int result = tts.setLanguage(Locale.getDefault());

if (result == TextToSpeech.LANG\_MISSING\_DATA

|| result == TextToSpeech.LANG\_NOT\_SUPPORTED) {

Log.e("TTS", "This Language is not supported");

} else {

speak.setEnabled(true);

speakOut();

}

} else {

Log.e("TTS", "Initilization Failed!");

}

}

@Override

public void onDestroy() {

// Shuts Down TTS

if (tts != null) {

tts.stop();

tts.shutdown();

}

super.onDestroy();

}}

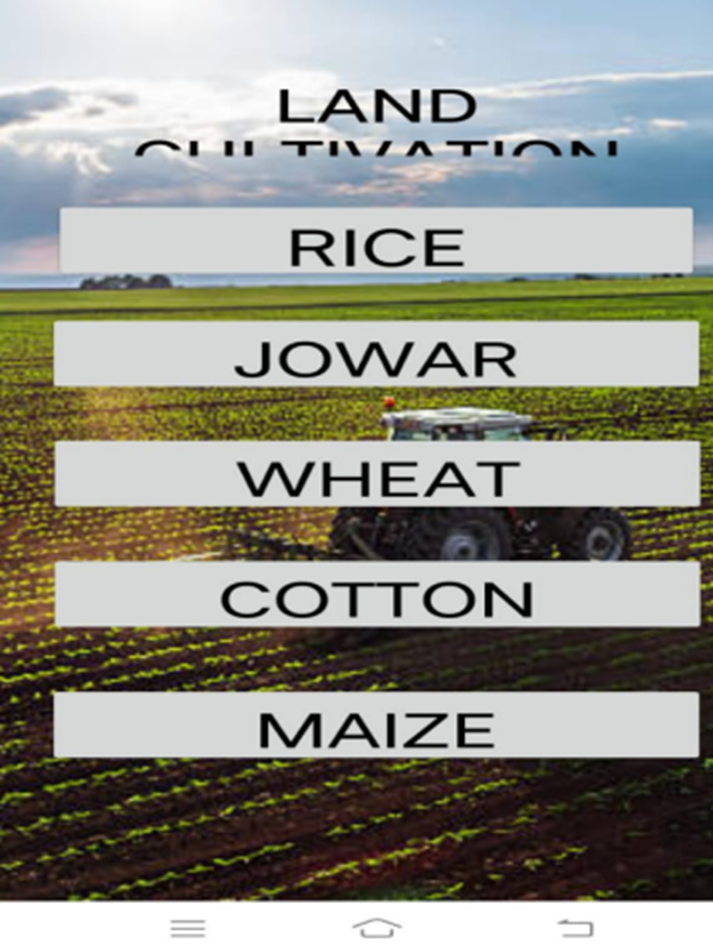
# 6. SYSTEM TESTING

Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design and coding. Testing represents an interesting anomaly for the software. During earlier definition and development phases, it was attempted to build software from an abstract concept to a tangible implementation. No system is error free because it is so till the next error crops up during any phase of the development or usage of the product. A sincere effort however needs to be put to bring out a product that is satisfactory.

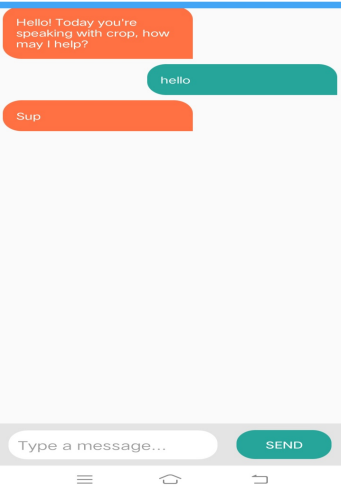
******

***Fig. 6.1:*** Home Page Of Crop Progression.

# 7. RESULTSCREEN SHOTS

***  
 fig.7.1:*** The above figure is displayed after ***fig.7.***2: The above figure is displayed after clicking on roof gardening. clicking on land cultivation

.

******

***Fig. 7.3:*** Chat Bot

# 8. CONCLUSION:

As the demand for food would increase over coming years, and people have the habit to grow the crops in order to make refreshment from their daily routine, from this, they can learn the the process of growing the crops in both land and roof gardening they can also learn the new techniques evolved in the agriculture sector the people can learn by virtual assistant as well as the chat bot in coming years we can provide more features and crops regarding land and roof gardening.

# 9. BIBLOGRAPHY:

## ***E-References:***

<https://www.indiaagronet.com/>

com.google.android.material:material:1.3.0

<https://www.jetbrains.com/opensource/kotlin/>

<https://www.agriculture.com/crops/progress-maps>

https://vaadin.com/blog/building-a-chatbot-in-java