CAPSTONE PROJECT REPORT

(Project Term January-April, 2018)

"SMART FARM"

Submitted by

Name: Ujjwal Singh Registration Number: 11413193

Name: Tandra Sandeep Reddy Registration Number: 11402341

Name: P.Pranita Kumari Registration Number: 11406811

Project Group Number – CSERGC0022

Course Code – CSE 445

Under the Guidance of

Aditya Bakshi

Asst. Professor (CSE)

School of Computer Science and Engineering

INDIA'S LARGEST UNIVERSITY

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education, Transforming India

TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering

Program : 1202D::B.Tech -M.Tech (Dual Degree) - CSE

COURSE CODE : CSE445 **REGULAR/BACKLOG :** Regular **GROUP NUMBER :** CSERGC0022

Supervisor Name : Aditya Bakshi **UID :** 17433 **Designation :** Assistant Professor

Qualification : _____ **Research Experience :** _____

SR.NO.	NAME OF STUDENT	REGISTRATION NO	BATCH	SECTION	CONTACT NUMBER
1	Ujjwal Singh	11413193	2014	K1401	7415468160
2	P. Pranita Kumari	11406811	2014	K1402	07209766680
3	Tandra Sandeep Reddy	11402341	2014	K1423	9872393911

SPECIALIZATION AREA : Networking and Security **Supervisor Signature:** _____

PROPOSED TOPIC : Smart Farm

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	**Parameter**	**Rating (out of 10)**
1	Project Novelty: Potential of the project to create new knowledge	8.00
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.83
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.83
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	8.33
5	Social Applicability: Project work intends to solve a practical problem.	8.00
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	8.00

PAC Committee Members		
PAC Member 1 Name: Prateek Agrawal	UID: 13714	Recommended (Y/N): Yes
PAC Member 2 Name: Deepak Prashar	UID: 13897	Recommended (Y/N): Yes
PAC Member 3 Name: Pushpendra Kumar Pateriya	UID: 14623	Recommended (Y/N): Yes
PAC Member 4 Name: Balraj Singh	UID: 13075	Recommended (Y/N): Yes
PAC Member 5 Name: Anupinder Singh	UID: 19385	Recommended (Y/N): Yes
DAA Nominee Name: Kuldeep Kumar Kushwaha	UID: 17118	Recommended (Y/N): Yes

Final Topic Approved by PAC: **Smart Farm**

Overall Remarks: Approved

PAC CHAIRPERSON Name: 11024::Amandeep Nagpal **Approval Date:** 15 Mar 2018

4/27/2018 1:33:25 PM

DECLARATION

We hereby declare that the project work entitled ("Smart Farm") is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University Phagwara, under the guidance of Aditya Bakshi, during January to April 2018. All the information furnished in this Capstone Project report is based on our own intensive work and is genuine.

Project Group Number: CSERGC0022

Name: Ujjwal Singh	Signature of Student1
Registration Number: 11413193	Date:
Name: Tandra Sandeep Reddy	Signature of Student1
Registration Number: 11402341	Date:
Name: P.Pranita Kumari	Signature of Student1
Registration Number: 11406811	Date:

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this capstone project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of this work has ever been submitted for any other degree at any other University. The capstone project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

Mentor: Mr. Aditya Bakshi

Designation: Asst. Professor

Signature:

School of Computer Science and Engineering

Lovely Professional University, Phagwara, Punjab

Date:

ACKNOWLEDGEMENT

We are using this opportunity to express our gratitude to everyone who supported us throughout the course of this B.Tech project. We are thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. We are sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to this project.

We express our warm thanks to Mr. Aditya Bakshi for his support and guidance at Lovely Professional University.

Thank you,

Ujjwal Singh

Tandra Sandeep Reddy

P.Pranita Kumari

TABLE OF CONTENTS

1. INTRODUCTION

We know that farming contributes 17.32% to the Indian GDP. Farming has a huge potential in India. PM Narendra Modi wants to double the income of farmers, make them tech savvy and boost agricultural research and education. In this smart world we need a smart way to do the farming. We need to harness the power of technology for the effective use of agricultural resources and hence increase the agricultural production. So we came up with idea of Smart Farm which has three modules. First module is related to Smart Irrigation in which we use Raspberry Pi with Arduino and moisture sensor to sense the moisture. In this whenever the moisture is low the farmer gets the message and automatically motor pump switches on and the real-time moisture values are displayed in the android application and the farmer can switch off the motor by the toggle of button. Second module deals with the protection of crops by the help of face detection. In this module we use simple python scripts and Opencv.If farmer comes the camera recognizes him and the door opens and on the other hand if an unknown person comes then a Beep sound is made to ward him off. Third module deals with Bird detection by the help of Canny edge detection and contours. In this module also we use python scripts and Opencv.If this whenever a bird comes within the sphere of influence of camera a Beep sound is made to scare it.

2. PROFILE OF PROBLEM

2.1 PROJECT TITLE

Smart Farm

2.2 OBJECTIVE

Our main objective is to help the farmers

2.3 PROBLEM

Problem1: Inefficient irrigation system. In India we know that farmers use motor pump to irrigate the field but due to inefficient irrigation system a lot of water is wasted and the irrigation becomes a cumbersome and lengthy process.

Problem2: After harvesting period, farmers store their crops. While theft or damage of same is also a problem faced by farmers as some person may try to steal the crops stored in the store room or in the fields.

Problem3: Damage to crops by birds. Birds like Rock Pigeon, House Sparrow, and Heron etc. damage the crops very badly due to which farmers face huge loss in their earning.

2.4 SOLUTION

Solution1: Smart Irrigation by the use of Raspberry Pi, Arduino and moisture sensor. Moisture sensor senses the moisture and if it is low sends a message to farmer and automatically the water pump switches on. While the motor is running moisture values are sent by the help of Firebase in Android application which is there in Farmer's mobile phone and after some time Farmer can switch off the motor by toggling the button in application.

Solution2: Crop theft can be controlled by authorized entry of Farmer in the field or store room. By the help of face detection the camera recognizes farmer and on the other hand wards off thief by making a loud noise.

Solution3: Damage to the crops can be reduced by detecting the birds with help of Canny edge detection and contour method. In this whenever a bird comes within the sphere of influence of camera then a loud noise is generated to scare the bird.

2.5 SIGNIFICANCE

The idea of Smart Farming is very significant considering the fact that the farmer need not go to farm to check the irrigation as he/she can control it by application. Farmer need not worry about the safety of crop as by the help of Face detection and Bird detection the crop is safe.

3. EXISTING SYSTEM

3.1 INTRODUCTION

Currently if we look at IOT projects related to irrigation we have systems with sensors which can detect the moisture and the motor can be switched on and off accordingly.

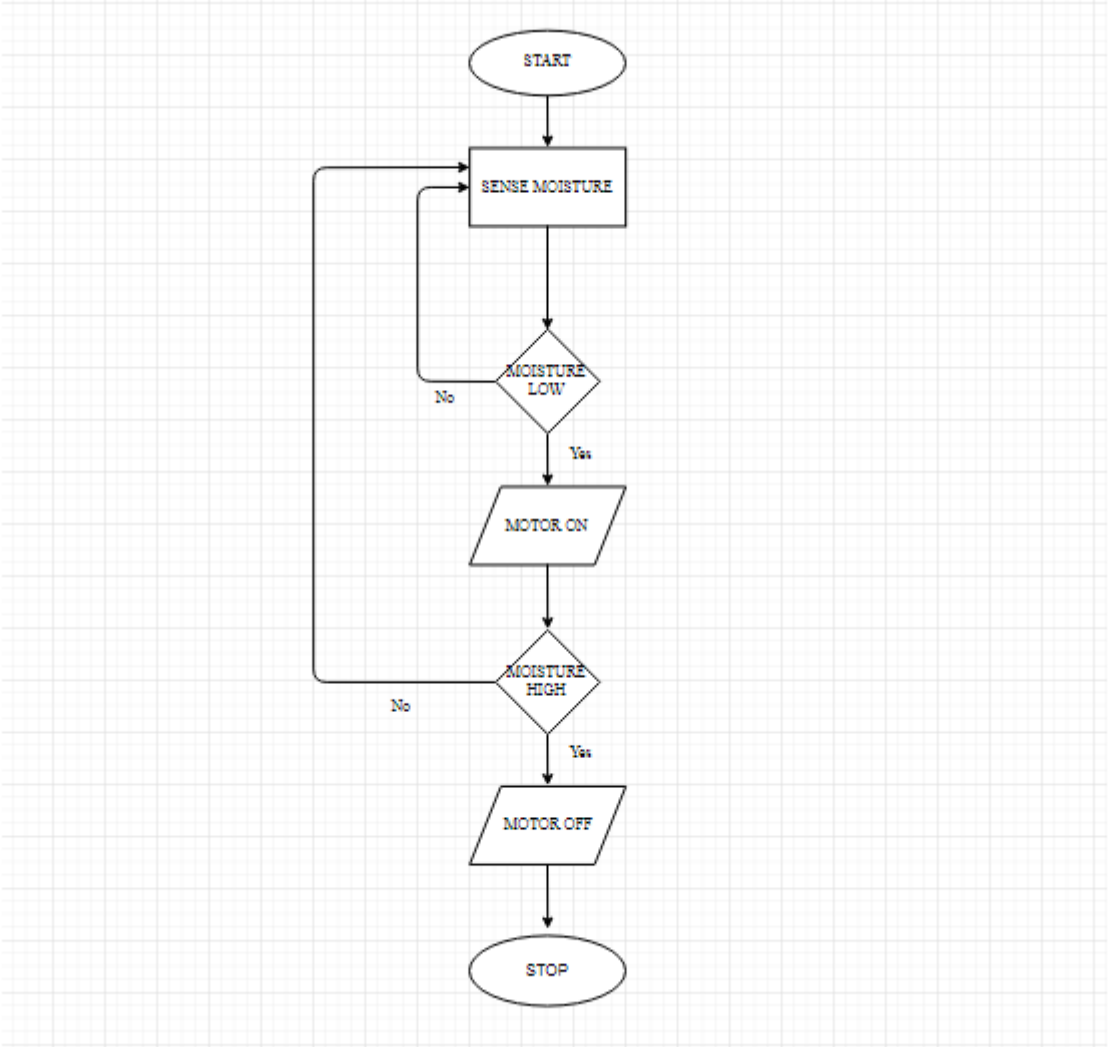
3.2 EXISTING HARDWARE

Currently IOT projects dealing with irrigation use Raspberry Pi, Arduino, Jumper wires, Breadboard, Motor pump, Relay and Moisture sensor. Some projects use even cloud platforms. The language involved in the development of these projects is mainly python.

3.3 WHAT'S NEW IN THE SYSTEM TO BE DEVELOPED

In our IOT project of Smart Farm we have divided our project into three modules. First module deals with Smart Irrigation mechanism by sensing the moisture content of the soil. If the moisture content is low a message is sent to the farm owner about the low moisture content in the field and at the same time sensor value with date and time is sent to the firebase from which android application fetches the sensor values in real-time and the farmer switches on the motor pump. The android application also provides guide for some of the popular crops in India. As the moisture content increases in the soil the moisture value fall from 1023 to 300-500 and as the moisture values are displayed in the android application the farmer on seeing that the soil has enough moisture switches off the motor by the click of button in android application. Second module deals with the safety of the crops either in the store room or in the field by using face detection. In this the farm owner's face is first trained and then after the training the camera detects the farmer by displaying the name of the farmer and if an unknown person is there in the field or store room the camera displays "None" and creates a loud Beep sound to scare the person. Third module deals with Bird detection by using python and OpenCV. In this we use Canny edge detection and contours to detect an object in front of the camera and this can be used in Bird detection also. Whenever a moving object comes in front of camera of some specific area a rectangle is displayed and a Beep sound is generated. This method can be used to detect birds and then by creating sound the birds can be warded off.

3.4 FLOW CHART FOR CURRENT SYSYEM



4. REQUIREMENT GATHERING AND ANALYSIS

4.1 INTRODUCTION

Requirement gathering and analysis is very important in order to develop a quality product.

4.2 DESCRIPTION

We have to first gather the hardware components and then the software components essential to make the product. We develop this project keeping in mind that it is prototype and hence it can further be developed in future with some other modifications.

4.3 REQUIREMENT GATHERING

4.3.1 HARDWARE REQUIREMENTS

- **Raspberry Pi**
- **Relay**
- **Motor Pump**
- **Arduino**
- **Jumper Wires**
- **Breadboard**
- **Moisture Sensor**

4.3.2 SOFTWARE REQUIREMENTS

- **Android Studio 3.0.1**
- **Python 2.7**
- **OpenCV 2.4.11**
- **Raspbian OS**

4.4 REQUIREMENT ANALYSIS

4.4.1 RASPBERRY PI

Raspberry Pi is a single board computer which uses Raspbian OS as its operating system. It is used in the development of IOT based projects. It has python as its primary language. It has 1GB SRRAM as RAM. It has Quad Cortex A53 @1.2GHz.

Its storage is microSD card. It has a GPIO header which has 40 pins. It video and audio output are HDMI/Composite and HDMI/Headphone respectively. In this sensors can also be attached as well as camera. The Raspberry Pi processes and analyses data using python.

4.4.2 RELAY

A relay is an electrically operated switch. These are used to switch on/off a motor pump.

4.4.3 MOTOR PUMP

Motor pumps are used in irrigation as they pump water from one place to another.

4.4.4 ARDUINO

Arduino is an open-source platform for building electronics projects. Arduino consists of both a physical programmable circuit board (microcontroller) and a piece of software, or IDE that runs on computer, used to write and upload computer code to the physical board.

4.4.5 JUMPER WIRES

Jumper wires are used for the connections.

4.4.6 BREADBOARD

Breadboard is a constructing base for prototyping of electronics. It is used to test the circuit designs.

4.4.7 LAPTOP

It is used here to run the python scripts and to run the Raspberry Pi.

4.4.8 MOISTURE SENSOR

A soil moisture sensor can read the amount of moisture present in the soil surrounding it. It's a low tech sensor. This is a must have tool for a connected and smart field. This sensor uses the two probes to pass current through the soil, and then it reads that resistance to get the moisture level. More water makes the soil conduct electricity more easily (less resistance), while dry soil conducts electricity poorly (more resistance).Here we are using moisture sensor YL-69.

4.4.9 ANDROID STUDIO 3.0.1

Android studio 3.0.1 is used to develop android applications as it provides the environment for the development of applications.

4.4.10 PYTHON 2.7 and OPENCV 2.4.11

Python 2.7 with OpenCV 2.4.11 are used to run python scripts in the project.

4.4.11 RASPBIAN OS

Raspbian is a free OS based on Debian optimized for the Raspberry Pi hardware. It comes with over 35000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.

5. DESIGN

Development of the project Smart Farm requires:

1. Flow Chart

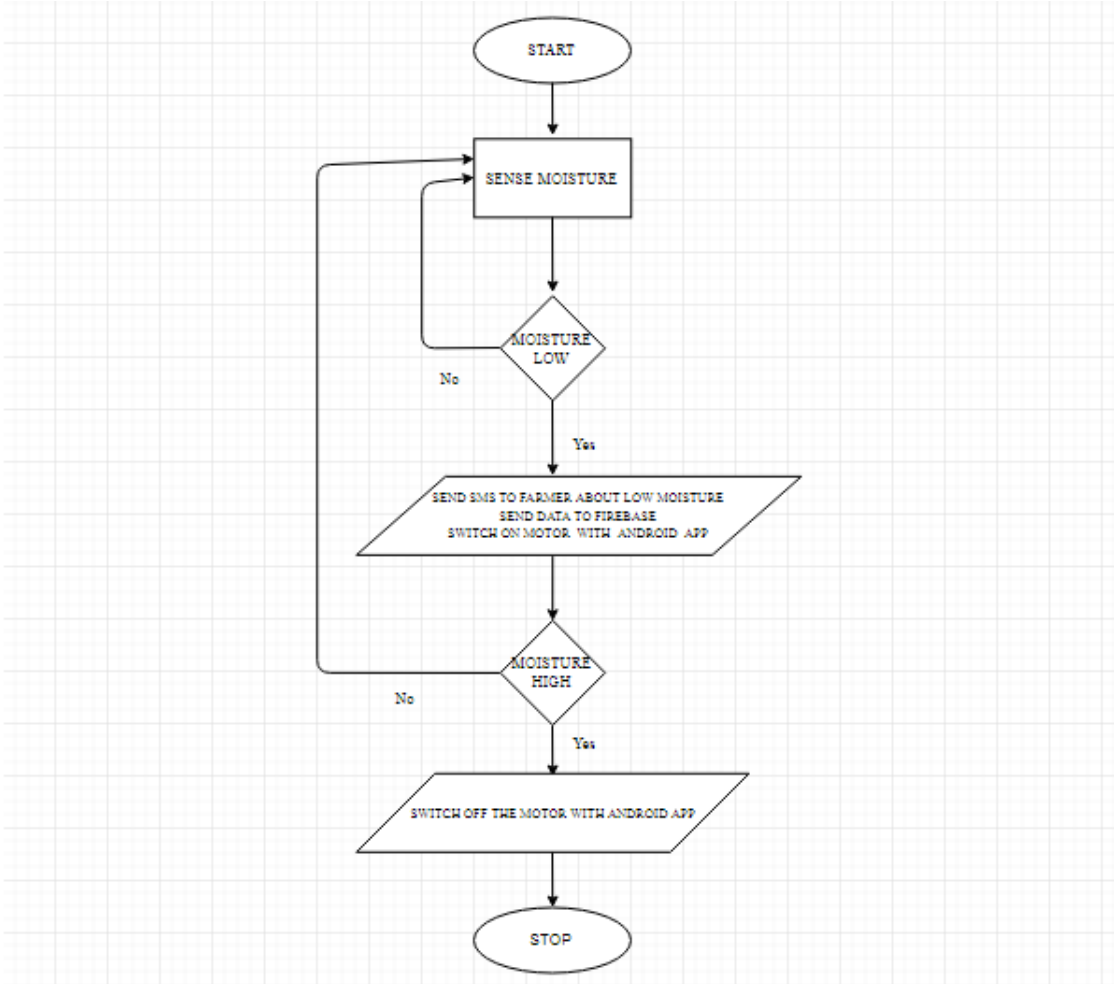
2. Pseudocode

5.1 FLOW CHART

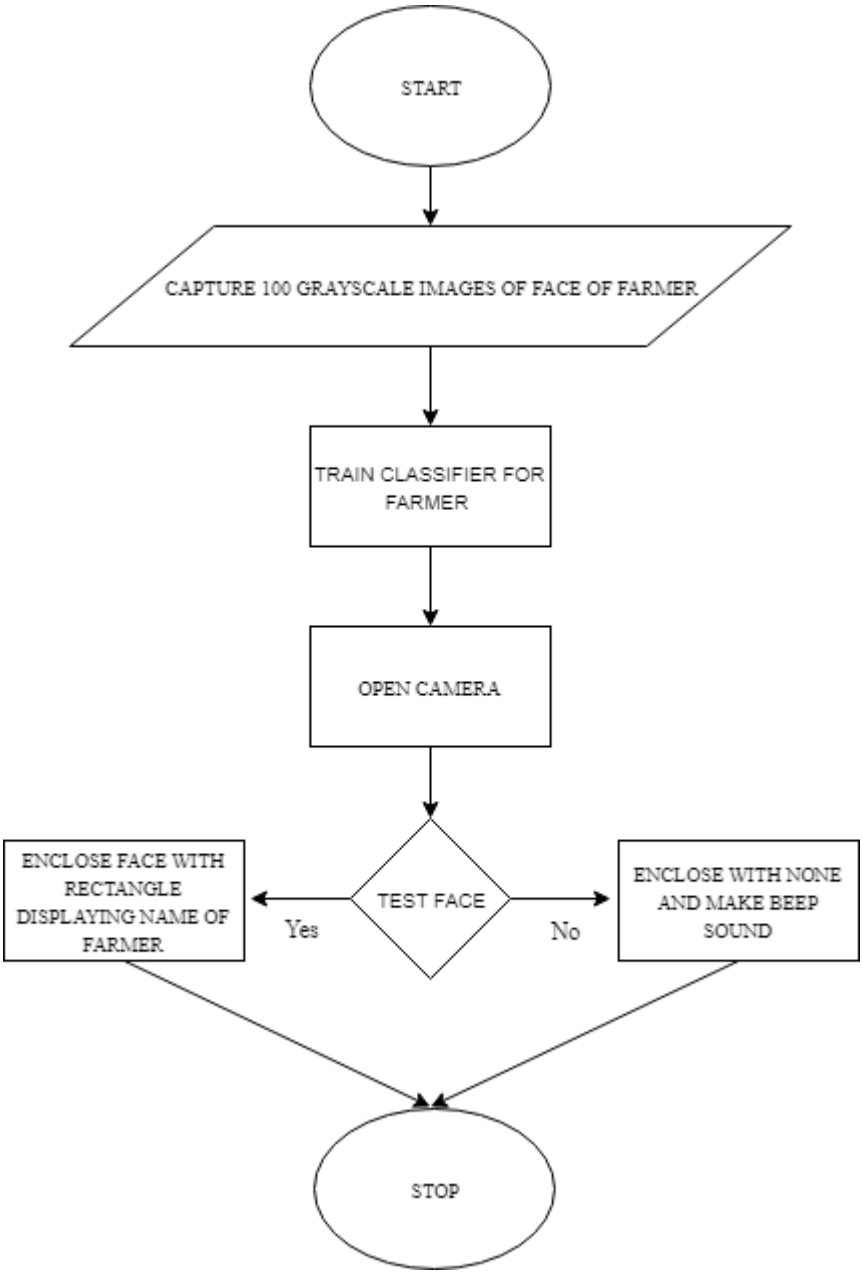
5.1.1 OVERVIEW OF PROJECT

Overview provides information about the outline of the project. The project contains three modules.

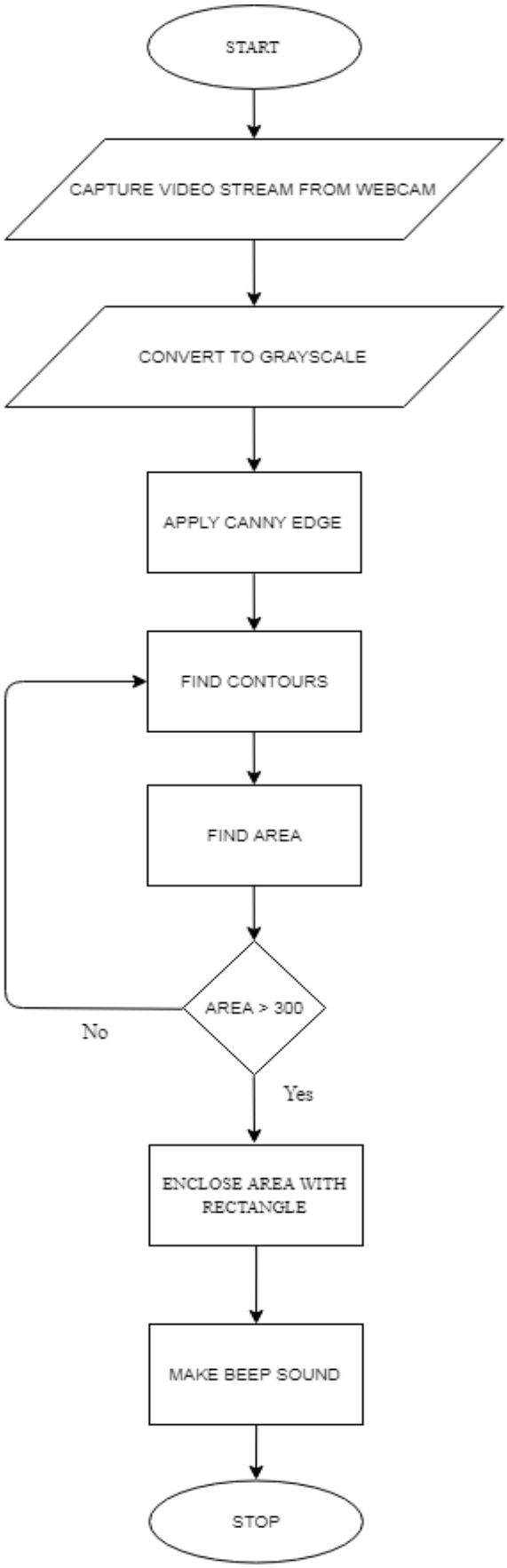
5.1.2 SMART IRRIGATION



5.1.3 FACE DETECTION



5.1.4 BIRD DETECTION



5.2 PSEUDOCODE

5.2.1 PSEUDOCODE FOR SMART IRRIGATION

Start

Sense the moisture by using moisture sensor

If moisture is low:

Send the sms to farmer that the moisture is low

Send the moisture value with date and time into the firebase

Farmer switches on the motor pump with android application

Else:

Sense moisture

Keep on sending moisture values, date and time to the firebase

After some time

If moisture is high:

Farmer switches off the motor pump with android application

Else:

Sense moisture

Stop

5.2.2 PSEUDOCODE FOR FACE DETECTION

Capture video stream from Webcam

While true:

Read frame from video stream of Webcam

Convert to grayscale

Show the frame

From frame capture Region of Interest for face of the person

Take 100 grayscale images for training

Train the face classifier by the 100 grayscale images

After training

Open the webcam for face detection

If the person for which classifier was trained is the same person:

Display the name of Person enclosing his face with rectangle of red color

Else:

Display “None” for unknown Person face

And make Beep sound for unknown person detection

5.2.3 PSEUDOCODE FOR BIRD DETECTION

Capture video stream from Webcam

Apply BackgroundSubtraction

While True:

Read frame from video stream of Webcam

Apply GaussianBlur

Convert to grayscale

Apply BackgroundSubtraction to new frame

Use Canny edge detection

Find contours

For each contour in contours:

Find area

If area>300:

Enclose the area with rectangle of specific color

And make a Beep sound

Show initial frame

Show final frame

If key pressed is “Q” then break

Release video stream

Destroy all windows

6. TESTING

6.1. DEFINITION

Testing is the process of executing a system with the intent of finding an error. It is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction. The quality is defined as justification of the requirements. Defect is nothing but deviation from the requirements and bug. Testing can demonstrate the presence of bugs, but not their absence. Debugging and Testing is not the same thing. Testing is a systematic attempt to break a program. Debugging is the art or method of uncovering why the script /program did not execute properly.

The main objectives of testing are:

1. Testing is the process of executing a program with the intent of finding errors.

2. A good test case is one that has the high probability of finding the undiscovered errors.

3. A successful test is one that uncovers all the undiscovered errors.

Testing Methodologies:

- **Black box (Functional) Testing**: is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application. Usually Test Engineers are involved in the black box testing.
- **White box (Structural) Testing:** is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually The Developers are involved in white box testing.
- **Grey box Testing**: is the process in which the combination of black box and white box tonics' are used.

6.2 TESTING TYPES IN PROJECT

1. Functionality Testing

2. Usability Testing

3. Interface Testing

4. Compatibility Testing

5. Performance Testing

6. Security Testing

6.2.1 FUNCTIONAL TESTING

Under this section we test all the major components of the project.

- The moisture sensor is reading the moisture level in soil.
- Reading are in range of 0-1023.
- Raspberry Pi is running with is OS properly.
- Arduino connection to Raspberry Pi and to soil moisture sensor indicated with a green LED.
- All the readings of the sensor can be read from real-time database storage in Firebase Cloud.
- Notifications send to farmer through mail or message on mobile phone.
- Readings can be collected on App – Smart Farm.
- Motor Pump is working as the program is detecting low moisture by sensors.
- For the detection of birds above the field and producing loud noise to scare away the birds program is running.
- Detecting and identification of the farmer under camera as well generating alarming sound to scare away the unknown person on detection.

6.2.2 STRUCTURAL TESTING

Structural test design techniques include**:**

Control flow testing

Whether the flow of control of the code is in order i.e. Level wise

Data flow testing

Does the data flow between two blocks or within a block occurs? Is it flowing as needed or any bugs are present?

Statement coverage testing

Has each statement in the program been executed?

Branch flow testing

Tests whether each branch of each control structure has been executed or not. For example in if statement whether both the true and false branches been executed or not.

Condition coverage testing

Has each Boolean sub-expression evaluated both to true and false or not?

Path testing

Has every path through a program been executed at least once or not?

6.3 USABILITY TESTING

Testing For Navigation: Navigation means how user browses the android app and explore the item in app like pages, buttons, menu bar, images, etc. Main menu should be provided in each page for easy navigation.

Content Testing: Content should be easy, simple and understandable. Check for spelling in the content. Check for colors which annoys the user. Have to follow the standard rules of web content development. Errors in fonts, colors, themes etc., should be verified. Content should be meaningful.

6.4 INTERFACE TESTING

The main interface in app is the Firebase Cloud connectivity which in turn is connected with Arduino and soil moisture sensor. Check all the connection among these interfaces are connected well and working properly. Errors are handled properly. If database interface gives any error then server application should show the error messages.

6.5 COMPATIBILTY TESTING

Compatibility testing is very important aspect of any project. It contains:

- App compatibility
- Operating system compatibility
- Raspberry Pi compatibility
- Cloud real-time database compatibility
- Sensing operation

6.6 PERFORMANCE TESTING

- The Android App loading and fetching all reading collected on the sensor.
- The sensor reading the data between the ranges 0-1023.
- The Water pump is switching on according the program.
- All the Notifications are being send to Phone through mail or message.
- Detection of unknown person and generating a sound to alert.
- Detection of the bird and generating a distressed call to scare away the birds.
- Sound of particular frequency to be generated.

6.7 SECURITY TESTING

- Test by pasting internal URL directly into browser of firebase cloud address bar without login.
- Try invalid inputs in input fields by giving wrong range of values of the sensor and check whether they are validating correctly or not.
- Detection of the farmer only in person detection.

6.8 LEVELS OF TESTING

- **Unit testing:** Testing the modules or a specific section of code individually.
- **Integration testing:** After individually testing modules they are integrated and again go through integration testing
- **Regression testing:** Focuses on finding defects after a major code change has occurred.
- **Smoke testing:** This consists of minimal attempts to operate the software, designed to determine whether there are any basic problems present in the system that will prevent it from working at all.
- **Performance Testing**: Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements.
- **Usability Testing:** This testing tests the GUI whether it is user friendly or not. Whether it is easy to understand or not.
- **A/B testing:** It is basically like: run a test, change one thing, run the test again, compare the results. This is more useful with more small-scale situations, but very useful in fine-tuning any program.
- **System testing:** Testing the complete system as whole.
- **Installation testing:** An installation test assures that the system is installed correctly and working at actual customer's hardware.
- **Alpha testing**: It's an actual operational testing by potential users or an independent test team known by the developers. Alpha testing is often internal acceptance testing, before the software goes to beta testing.
- **Beta testing:** The software is released to groups of people (mostly general users) so that further testing can ensure the product has few faults or bugs.

7. IMPLEMENTATION

Implementation is the final and important phase. The most critical stage is in achieving a successful new system and in giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it found to working according to the specification.

Implementation is the stage in the project where the theoretical design is turned into the working system and is giving confidence to the new system for the users i.e. will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of method to achieve the changeover, an evaluation of change over methods. A part from planning, major task of preparing the implementation is to educate the targeted users. The more complex system is designed, the more complex will be the system implementation and publicity effort required.

There are three types of implementation:

- Implementation of a computer system to replace a manual system.
- Implementation of a new computer system to replace an existing system.
- Implementation of a modified application to replace an existing one, using the same computer.

7.1 IMPLEMENTATION OF THE PROJECT

The implementation type of this project falls under the first type of different categories of implementation discussed in the above introduction of this section i.e. implementation of a computer system to replace a manual system.

After the completing the designing, development and in-house & alpha-beta testing phases of the application; the implementation phase will be carried out in the following major steps:

7.1.1 PURCHASING THE COMPONENTS

Purchase of the components Raspberry Pi to run all the programs written in python.

Arduino and soil moisture sensor YL-69 to sense all the moisture content in the field and convert it into digital values. Water Pump and wires to connect the circuit.

7.2 POST-IMPLEMENTATION AND WEBSITE MAINTENANCE

Post Implementation: Monitoring all the storage area in firebase cloud and continuous mail or notification service under limit. Deploying more than one Sensor and connection of each sensor to code.

Website Maintenance:

Maintenance is the enigma of the project development. It holds the IOT industry captive, tying up programming resources. Analysts and programmers spend more time maintaining programs than they writing them. Though maintenance is not considered a part of software development, but it's an activity that's extremely important in the life of website service.

1. Corrective Maintenance:

After the Implementation, correcting the residual errors if any. If such errors are discovered, the source of it should be detected and removed. This phenomenon falls under corrective maintenance.

2. Perfective Maintenance:

Sometimes changes have to be done according to the user requirements. This type of changes to the website is called perfective maintenance.

3. Adaptive Maintenance:

Website often must be upgraded and enhanced to include more features and provide more services. This also requires modification of website.

8. PROJECT LEGACY

8.1 CURRENT STATUS OF THE PROJECT

Our project SmartFarm is working properly. It is an online project helping farmers to automate the field and monitor the irrigation system in the field.

8.2 REMAINING AREAS OF CONCERN

Censorship:

The most important area of concern is to prohibit or restrict of uploading incorrect and illegal contents over the application. The concern of identifying and learning which contents aren't appropriate is the major challenge for us. For now, to identify and restrict such kind of publishing activities, we're taking the first hand users' reports and then taking action. But in near future we, as the administrator of the app, want and will try to identify and restrict such kind of contents from publishing beforehand.

Data Manageability:

Another important area of our concern is to manage the generated data. Being just in the initial stage, the management of data is very much an easy task but with every passing day, with every new user being added and with ever increasing generation of data the complexity and manageability of the database becomes a complicated task. Therefore, to avoid such kind of situation, we've to start looking for better manageability and scalability options.

Data Security:

Any administrator can never forget about the privacy and protection of the identities of his/her users'. Similarly, we totally care and in concern with our users data security whether his/her confidential data or general activity data. We treat all the data same and try to keep respective level of security for the respective type of data. Even after implementing many security measures to protect the user's credentials, there always remains a security concern over such issues and all we can do is to keep making our application software more secure with every updates.

Future Scope of Improvements:

So far what we have developed and deployed is just a basic structural project which is capable of handling the various farmer issues which a farmer using the system and app is expected to solve its problem. It is a prototype to work for demonstration purpose presently. Using configuring system and deployment of the whole project into field can help farmers to increase their productivity.

8.3 TECHNICAL AND MANAGERIAL LESSONS LEARNT

This project has benefited us in many ways .The first benefit is that we have got organizational exposure and it has provided us with an opportunity to know the environment, the practices and the system. It has helped us to sharpen our knowledge and skills, develop better appreciation of practical problems of app development and to apply the concepts and technique to developmental problems. This experience is going to help each of us immensely in further learning of advanced concepts in IOT project development and to plan our career in the light of practical experiences. We have examples to relate and it will facilitate better and easier learning for us.

Technical lessons learnt

- Installation and usage of Raspberry Pi
- Connection of sensor and Arduino
- OpenCV and Python integration
- Configuration of Firebase Cloud
- Android application development

Managerial lessons learnt

• Ensuring quality and integrity of data

• Planning of Duration and schedule of the project

• Strategic planning to avoid miscommunication among the team members

• Participative Leadership

• Allocating resources

• Coordination

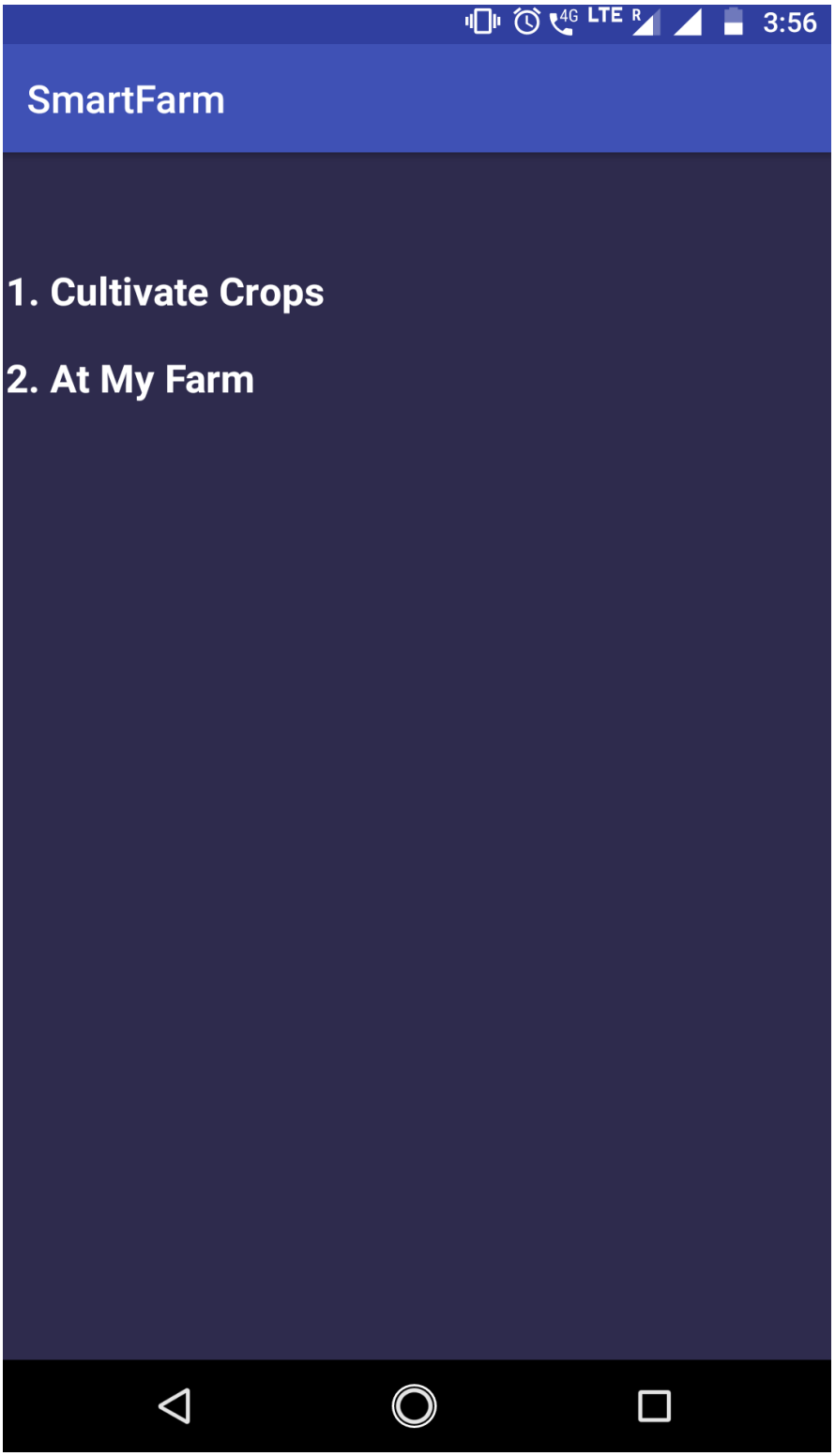
• Risk Analysis and prevention

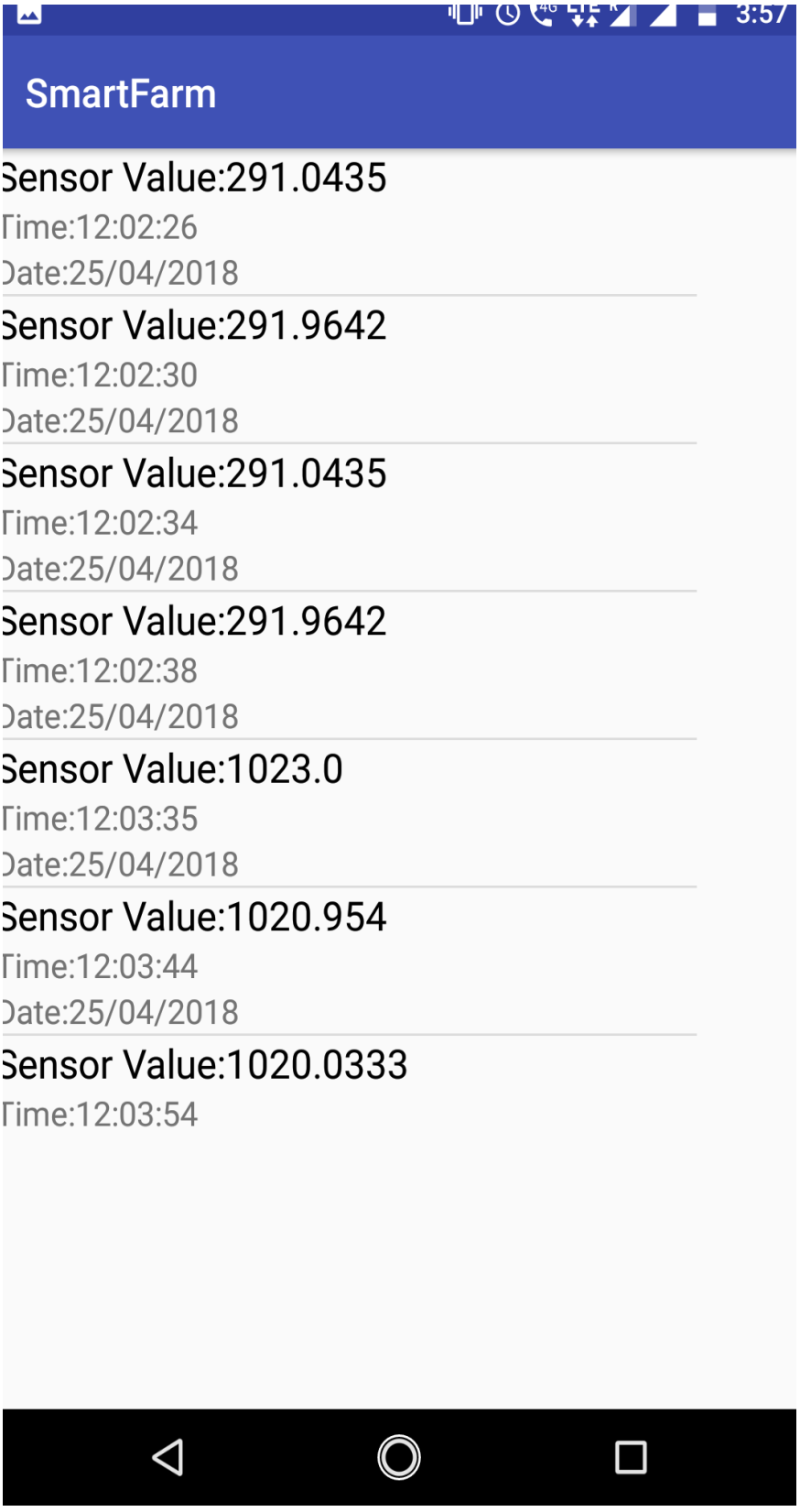
- Integrating individual work to make it collaborative work
- Defining smaller goals to achieve a bigger common goal
- Organizational Exposure and an insight to the business

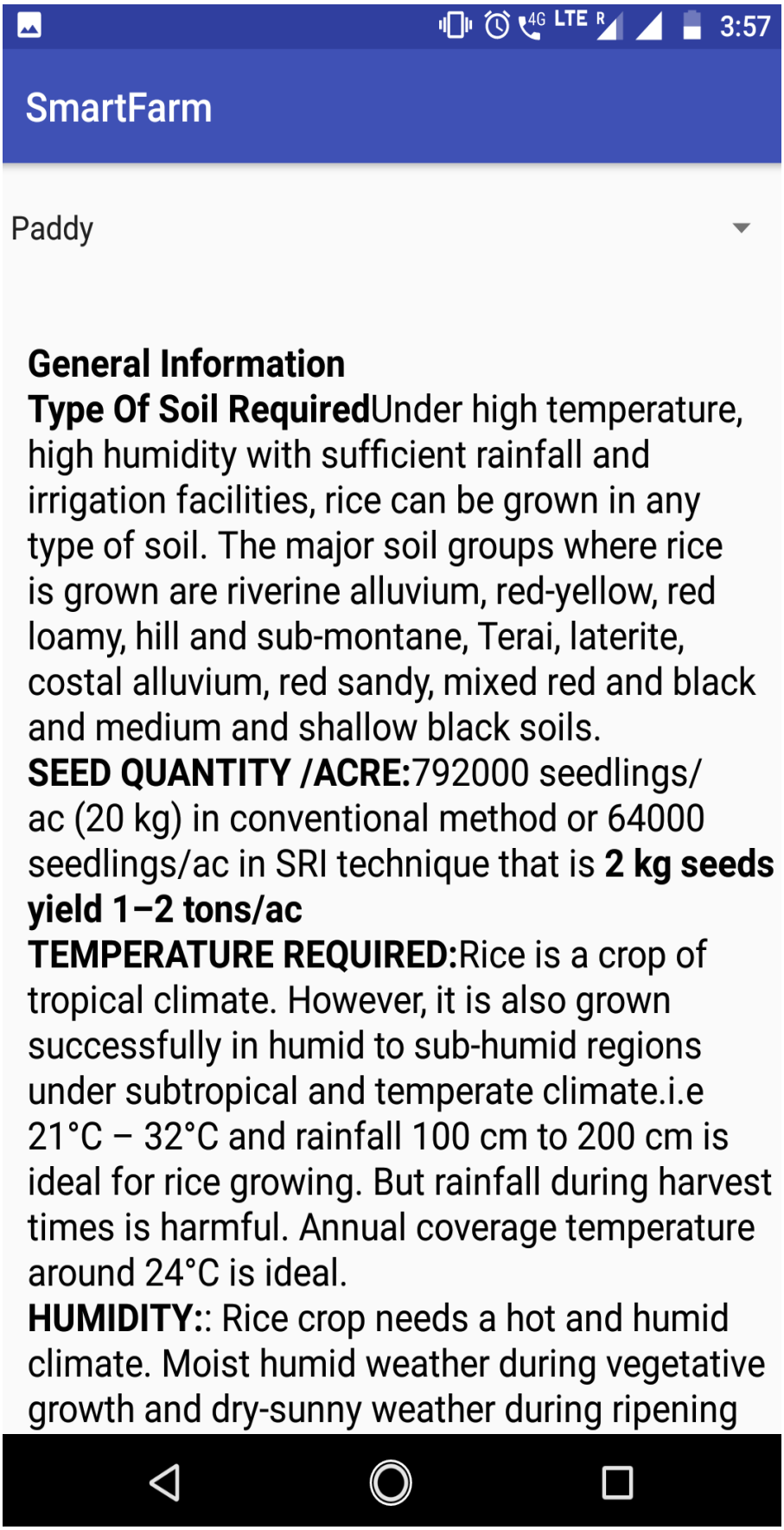
9. USER MANUAL

A user manual is a technical communication document intended to give assistance to people using a particular system. User manual of our project SmartFarm is divided into three modules.

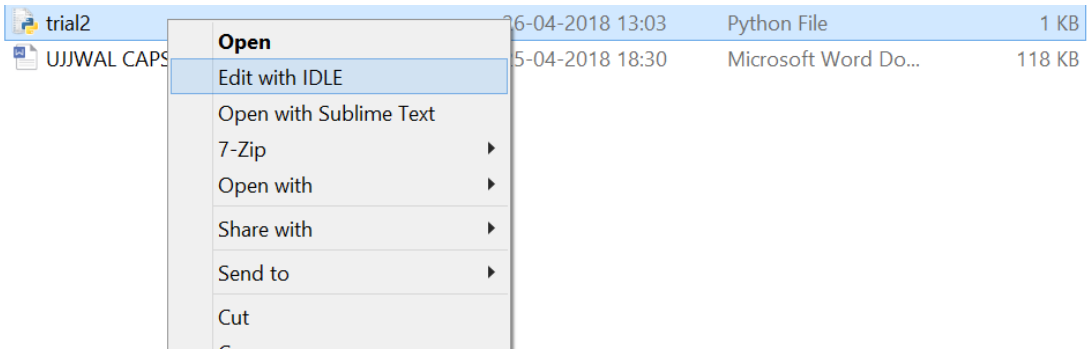
9.1 MODULE1

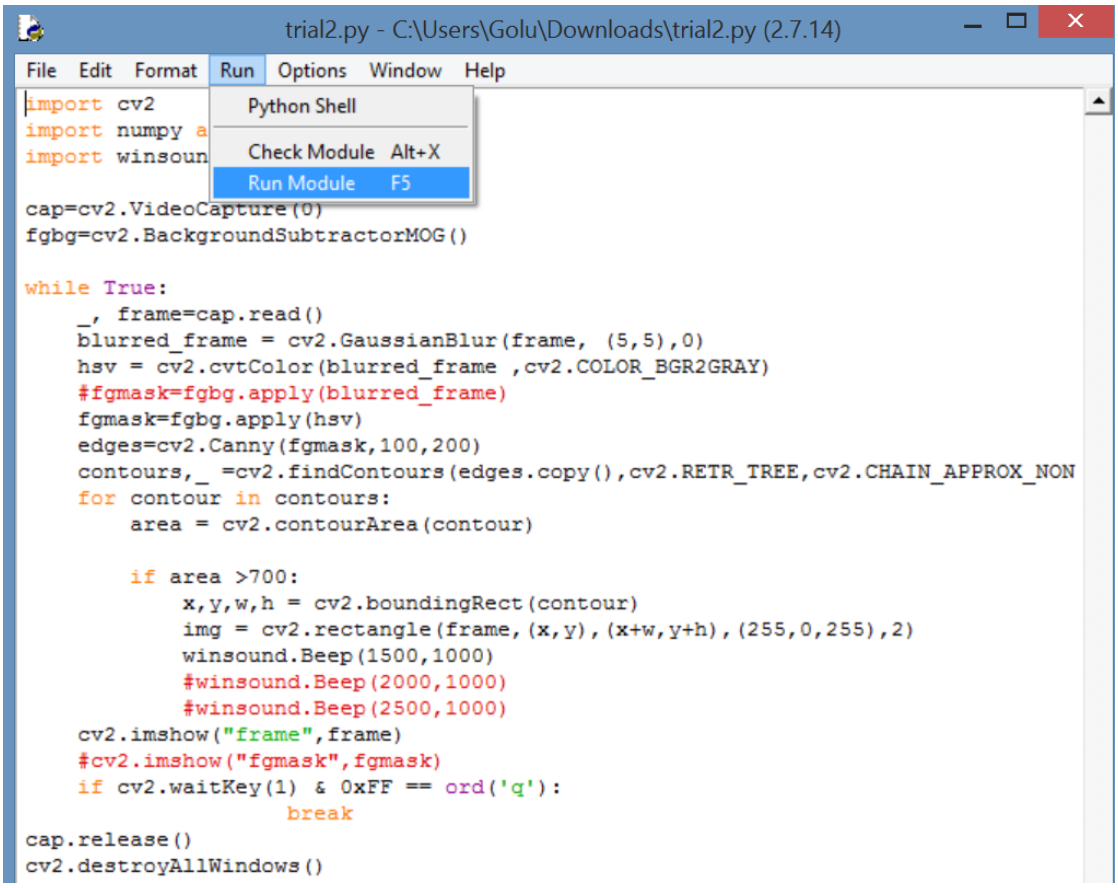

3:56
SmartFarm
1. Cultivate Crops
2. At My Farm


3:57
SmartFarm
Sensor Value:291.0435
Time:12:02:26
Date:25/04/2018
Sensor Value:291.9642
Time:12:02:30
Date:25/04/2018
Sensor Value:291.0435
Time:12:02:34
Date:25/04/2018
Sensor Value:291.9642
Time:12:02:38
Date:25/04/2018
Sensor Value:1023.0
Time:12:03:35
Date:25/04/2018
Sensor Value:1020.954
Time:12:03:44
Date:25/04/2018
Sensor Value:1020.0333
Time:12:03:54


3:57
SmartFarm
Paddy
General Information
Type Of Soil RequiredUnder high temperature, high humidity with sufficient rainfall and irrigation facilities, rice can be grown in any type of soil. The major soil groups where rice is grown are riverine alluvium, red-yellow, red loamy, hill and sub-montane, Terai, laterite, costal alluvium, red sandy, mixed red and black and medium and shallow black soils.
SEED QUANTITY /ACRE:792000 seedlings/ ac (20 kg) in conventional method or 64000 seedlings/ac in SRI technique that is 2 kg seeds yield 1–2 tons/ac
TEMPERATURE REQUIRED:Rice is a crop of tropical climate. However, it is also grown successfully in humid to sub-humid regions under subtropical and temperate climate.i.e 21°C – 32°C and rainfall 100 cm to 200 cm is ideal for rice growing. But rainfall during harvest times is harmful. Annual coverage temperature around 24°C is ideal.
HUMIDITY:: Rice crop needs a hot and humid climate. Moist humid weather during vegetative growth and dry-sunny weather during ripening

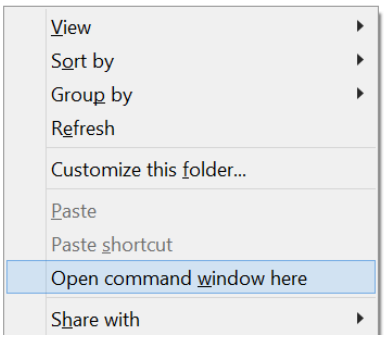
9.2 MODULE2

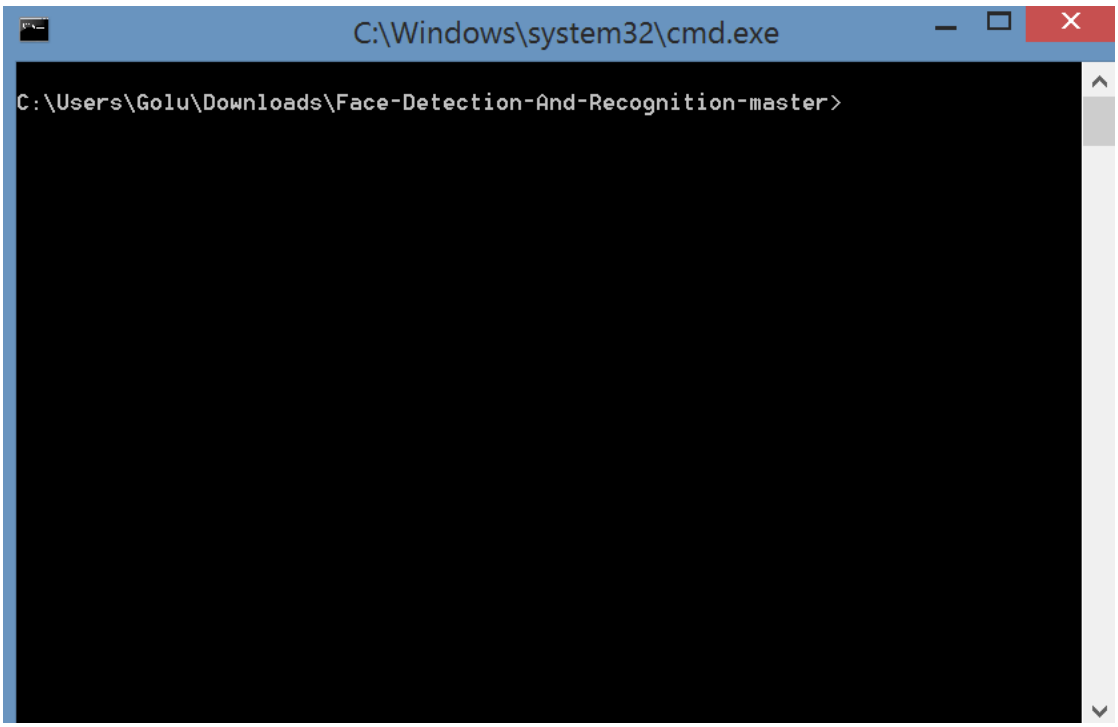

trial2
26-04-2018 13:03
Python File
1 KB
UJJWAL CAPS
5-04-2018 18:30
Microsoft Word Do...
118 KB
Open
Edit with IDLE
Open with Sublime Text
7-Zip
Open with
Share with
Send to
Cut

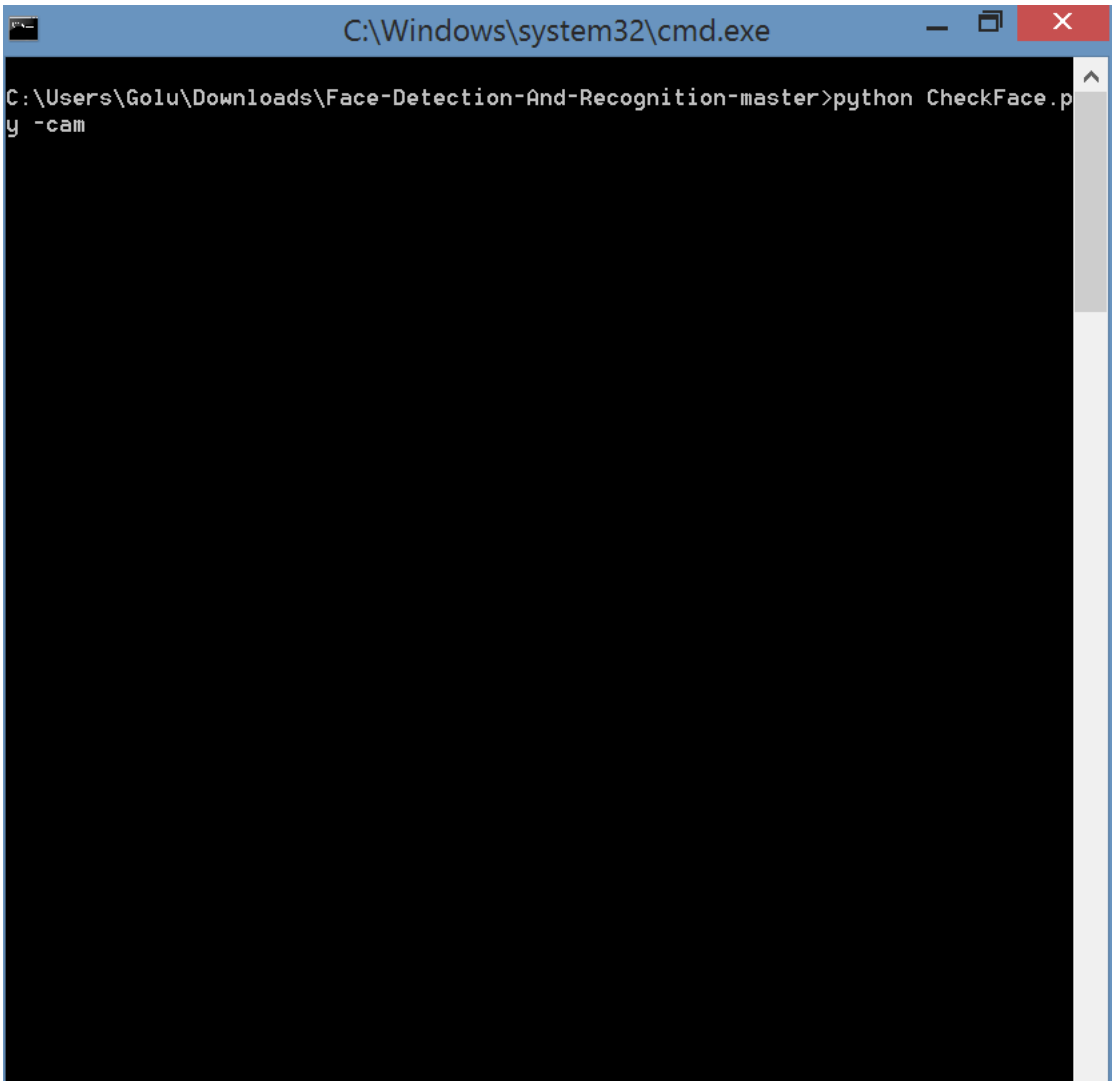

trial2.py - C:\Users\Golu\Downloads\trial2.py (2.7.14)
File Edit Format Run Options Window Help
Python Shell
Check Module Alt+X
Run Module F5
import cv2
import numpy a
import winsoun
cap=cv2.VideoCapture(0)
fgbg=cv2.BackgroundSubtractorMOG()
while True:
_, frame=cap.read()
blurred_frame = cv2.GaussianBlur(frame, (5,5),0)
hsv = cv2.cvtColor(blurred_frame ,cv2.COLOR_BGR2GRAY)
#fgmask=fgbg.apply(blurred_frame)
fgmask=fgbg.apply(hsv)
edges=cv2.Canny(fgmask,100,200)
contours,_ =cv2.findContours(edges.copy(),cv2.RETR_TREE,cv2.CHAIN_APPROX_NON
for contour in contours:
area = cv2.contourArea(contour)
if area >700:
x,y,w,h = cv2.boundingRect(contour)
img = cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,255),2)
winsound.Beep(1500,1000)
#winsound.Beep(2000,1000)
#winsound.Beep(2500,1000)
cv2.imshow("frame",frame)
#cv2.imshow("fgmask",fgmask)
if cv2.waitKey(1) & 0xFF == ord('q'):
break
cap.release()
cv2.destroyAllWindows()

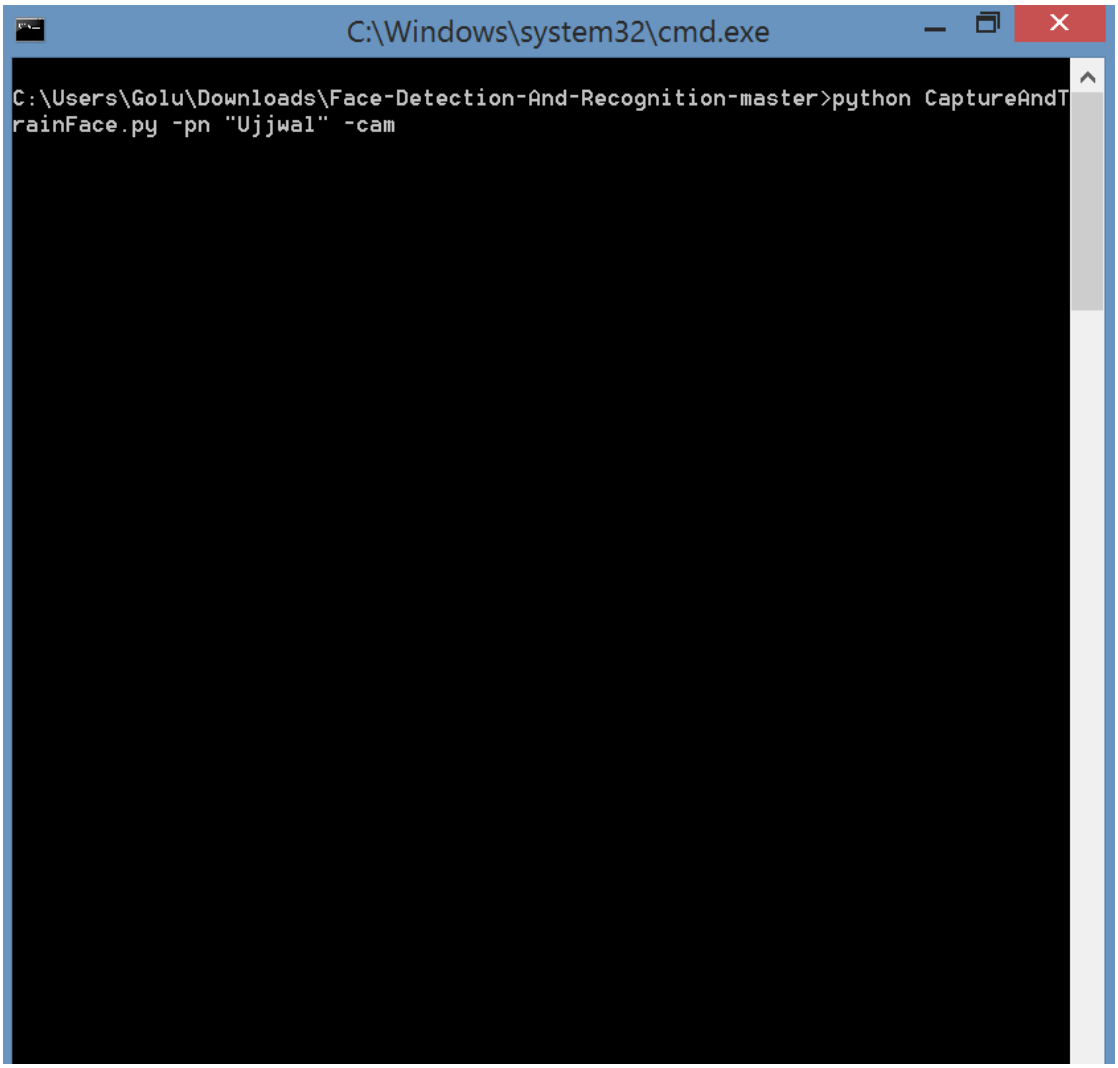
9.3 MODULE3

temp	25-04-2018 11:43	File folder	
train	16-04-2018 10:54	File folder	
.gitignore	01-03-2018 15:47	GITIGNORE File	2 KB
CaptureAndTrainFace	01-03-2018 15:47	Python File	4 KB
CheckFace	25-04-2018 11:41	Python File	4 KB
haarcascade_frontalface_default	01-03-2018 15:47	XML Document	909 KB
LICENSE	01-03-2018 15:47	File	2 KB
README.md	01-03-2018 15:47	MD File	2 KB
SampleImage	01-03-2018 15:47	PNG image	220 KB
SimpleFaceClassifier	01-03-2018 15:47	PNG image	185 KB






C:\Windows\system32\cmd.exe
C:\Users\Golu\Downloads\Face-Detection-And-Recognition-master>python CheckFace.p
y -cam


C:\Windows\system32\cmd.exe
C:\Users\Golu\Downloads\Face-Detection-And-Recognition-master>python CaptureAndTrainFace.py -pn "Ujjwal" -cam

10. SOURCE CODE

10.1 MODULE1

activity_farmer_home_page.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.visionframer.hemanthreddy.smartfarm.FarmerHomePage">


    <ListView
        android:id="@+id/listViewIrrigation"
        android:layout_width="368dp"
        android:layout_height="495dp"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp" />
</android.support.constraint.ConstraintLayout>
```

list_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="TextView"
        android:textSize="20dp"
        android:textColor="#000"
        android:id="@+id/textViewValue" />

    <TextView
        android:id="@+id/textViewTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/textViewValue"
        android:textSize="17dp"
        android:text="TextView" />

    <TextView
        android:id="@+id/textViewDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/textViewTime"
        android:textSize="17dp"
        android:text="TextView" />


</RelativeLayout>
```

detailsofcrops.xml

smartfarmhome.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:background="#2E2B4D"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/knowaboutcrops"
        android:layout_width="100dp"
        android:layout_height="45dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="58dp"
        android:text=" 1. Cultivate Crops"
        android:textColor="#FFF"
        android:textStyle="bold"
        android:textSize="20sp" />

    <TextView
        android:id="@+id/mycrops"
        android:layout_width="100dp"
        android:layout_height="45dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/knowaboutcrops"
        android:textStyle="bold"
        android:text=" 2. At My Farm"
        android:textColor="#fff"
        android:textSize="20sp" />

</RelativeLayout>
```

splash.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#272958">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="160dp"
        android:textStyle="bold|italic"
        android:textSize="45dp"
        android:textColor="#FFF"
        android:text="SmartFarm" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="129dp"
        android:textStyle="italic|bold"
        android:textSize="25dp"
        android:textColor="#FFF"
        android:text="A vision to help farmers..." />
</RelativeLayout>
```

AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.visionframer.hemanthreddy.smartfarm">
    <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=".FarmerHomePage">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.DEFAULT"  />
            </intent-filter>
        </activity>
        <activity android:name=".SmartFarmHome">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.DEFAULT"  />
            </intent-filter>
        </activity>
        <activity android:name=".Splash" android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".DetailsOfCrops">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

DetailsOfCrops.java

```java
public class DetailsOfCrops  extends AppCompatActivity implements AdapterView.OnItemSelectedListener
{
    TextView tv;
    Spinner spinner;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.detailsofcrops);
        spinner = (Spinner)findViewById(R.id.spinnercrops);
        spinner.setOnItemSelectedListener(this);
        List<String> categories = new ArrayList<String>();
        categories.add("Select Crops");
        categories.add("Paddy");
        categories.add("Cotton");
        ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, categories);
        dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
         spinner.setAdapter(dataAdapter);
         tv =(TextView)findViewById(R.id.detailsofcrops);
    }


    @Override
    public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
           String text = spinner.getSelectedItem().toString();


              // Toast.makeText(DetailsOfCrops.this, spinner.getSelectedItem().toString(), Toast.LENGTH_SHORT).show();

               if(text.equals("Paddy"))
               {
                   tv.setText(Html.fromHtml(getString(R.string.htmlpaddy)));
               }
               else if(text.equals("Cotton"))
               {
                   tv.setText(Html.fromHtml(getString(R.string.htmlcotton)));
               }

    }

    @Override
    public void onNothingSelected(Adapterview<?> adapterview) {

    }
```

FarmerHomePage.java

```
public class FarmerHomePage extends AppCompatActivity {
DatabaseReference databaseReference;
ListView listViewIrrigation;
List<Irrigation> irrigationList;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_farmer_home_page);
        databaseReference= FirebaseDatabase.getInstance().getReference("JALANDHAR");
        listViewIrrigation=(ListView)findViewById(R.id.listViewIrrigation);
        irrigationList= new ArrayList<>();
    }

    @Override
    protected void onStart() {
        super.onStart();
        databaseReference.child("moisture").addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                irrigationList.clear();
              for(DataSnapshot irrigationSnapshot:dataSnapshot.getChildren())
              {
                  Irrigation irrigation = irrigationSnapshot.getValue(Irrigation.class);
                  irrigationList.add(irrigation);

              }
                IrrigationList adapter= new IrrigationList(FarmerHomePage.this,irrigationList);
               listViewIrrigation.setAdapter(adapter);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });
    }
}
```

IrrigationList.java

```
public class IrrigationList extends ArrayAdapter<Irrigation> {
    private Activity context;
    private List<Irrigation> irrigationList;
    public IrrigationList(Activity context,List<Irrigation> irrigationList)
    {
        super(context,R.layout.list_layout,irrigationList);
     this.context=context;
     this.irrigationList=irrigationList;
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
        LayoutInflater inflater = context.getLayoutInflater();
        View listViewItem= inflater.inflate(R.layout.list_layout,null,true);
        TextView tvValue=(TextView)listViewItem.findViewById(R.id.textViewValue);
        TextView tvTime=(TextView)listViewItem.findViewById(R.id.textViewTime);
        TextView tvDate=(TextView)listViewItem.findViewById(R.id.textViewDate);
        Irrigation irrigation=irrigationList.get(position);
        tvValue.setText("Sensor Value:"+irrigation.getValue());
        tvTime.setText("Time:"+irrigation.getTime());
        tvDate.setText("Date:"+irrigation.getDate());
        return listViewItem;
    }
}
```

Irrigation.java

```java
public class Irrigation {
    String Value;
    String Date;
    String Time;
    public Irrigation()
    {}
    public Irrigation(String Value,String Date,String Time)
    {
        this.Value=Value;
        this.Date=Date;
        this.Time=Time;
    }

    public String getValue() {
        return Value;
    }



    public String getDate() {
        return Date;
    }



    public String getTime() {
        return Time;
    }


}
```

SmartFarmHome.java

```java
public class SmartFarmHome extends AppCompatActivity implements View.OnClickListener {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.smartfarmhome);
        TextView tv_how_to_cultivate = (TextView)findViewById(R.id.knowaboutcrops);
        TextView tv_know_at_my_farm = (TextView)findViewById(R.id.mycrops);
        tv_how_to_cultivate.setOnClickListener(this);
        tv_know_at_my_farm.setOnClickListener(this);


    }

    @Override
    public void onClick(View view) {
        if(view.getId()==R.id.knowaboutcrops)
        {
            Intent intent = new Intent(SmartFarmHome.this,DetailsOfCrops.class);
            startActivity(intent);
        }
        else if(view.getId()==R.id.mycrops)
        {
            Intent intent = new Intent(SmartFarmHome.this,FarmerHomePage.class);
            startActivity(intent);
        }
    }
}
```

Splash.java

```
public class Splash extends AppCompatActivity{
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.splash);
        Thread timer = new Thread()
        {
            public void run()
            {
              try{
                    sleep(3000);
              }catch(InterruptedException e)
              {
                  e.printStackTrace();
              }finally {
                  Intent to_Login_Page = new Intent(Splash.this,SmartFarmHome.class);
                  startActivity(to_Login_Page);
              }
              }
            };
        timer.start();
        }
        @Override
        protected void onPause()
        {
            super.onPause();
            finish();
        }

    }
```

Smartfarm.py

```
import pyfirmata
import zerosms
import time
import serial
import requests
import json
def zerosmssend():
    #user_name=getpass.getpass("Username:")
    user_name="6280225149"
    #user_password=getpass.getpass("Password:")
    user_password="way2sms123PK"
    msg="ALERT!!! our project is working and sensing low moisture values "
    #send_to=getpass.getpass("Send To:")
    send_to="9872393911"
    zerosms.sms(phno=user_name,passwd=user_password,message=msg,receivernum=send_to)

firebase_url='https://smartfarm-497ed.firebaseio.com/'
ser=serial.Serial('/dev/ttyACM0',9600,timeout=0)
fixed_interval=1
board=pyfirmata.Arduino('/dev/ttyACM0')
analog_pin=board.get_pin('a:0:i')
digital_pin=board.get_pin('d:8:o')
it=pyfirmata.util.Iterator(board)
it.start()
analog_pin.enable_reporting()
while True:
    try:

        reading=analog_pin.read()
        if reading>0.39 :
            digital_pin.write(1)
            zerosmssend()
        else:
            digital_pin.write(0)
        if reading!=None:
            moisture_val=reading*1023
           #moisture_val=map(reading,1023,250,0,100)
            print("Reading=%f moisture_val=%f" %(reading,moisture_val))
            print("%")
            time_hhmmss=time.strftime('%H:%M:%S')
            date_mmddyyyy=time.strftime("%d/%m/%Y")
            moisture_location='JALANDHAR'
            moisture_val=str(moisture_val)
            print(moisture_val+','+time_hhmmss+','+date_mmddyyyy+','+moisture_location)

            data={'Date':date_mmddyyyy,'Time':time_hhmmss,'Value':moisture_val}
            result=requests.post(firebase_url+'/'+moisture_location+'/moisture.json',data=json.dumps(data))

            print('Record Inserted.Result Code='+str(result.status_code) +','+result.text)
            time.sleep(fixed_interval)

    except IOError:
        print('Error!Something went wrong')
    time.sleep(fixed_interval)
```

10.2 MODULE2 FACE DETECTION

Source code for training

```
import cv2
import numpy as np
import os
import pickle
import sys

class CaptureAndTrain(object):

    def __init__(self):
        print('Hi')
        print('Image Training Started.')
        self.faceDetector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

    def captureFromDir(self,path):
        count = 0

        for file in os.listdir(path):
            img = cv2.imread(path+'/'+file)
            img = cv2.resize(img,(500,400))
            img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

            faceCord = self.faceDetector.detectMultiScale(img,1.3,5)

            for x,y,w,h in faceCord:
                roi = img[y:y+h,x:x+w]
                roi = cv2.resize(roi,(200,200))
                count += 1
                cv2.imwrite('temp/'+str(count)+'.jpg', roi)
```

```
def captureFromCam(self,cameraNumber):

    count = 0

    cap = cv2.VideoCapture(cameraNumber)

    while(True):

        _,frame = cap.read()

        frame = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)

        cv2.imshow('Image', frame)

        faceCord = self.faceDetector.detectMultiScale(frame,1.3,5)

        for x,y,w,h in faceCord:
            roi = frame[y:y+h,x:x+w]
            roi = cv2.resize(roi,(200,200))
            count += 1

            cv2.imwrite('temp/'+str(count)+'.jpg', roi)
            cv2.putText(roi,str(count),(20,20),2,0.7,(0,0,255),2)
            cv2.imshow('ROI', roi)


        if (cv2.waitKey(1) & 0xFF == ord('q')) or count >=100:
            break


    cap.release()
    cv2.destroyAllWindows()
```

```
def trainImage(self,personName):
    nameList = []
    imagePath = 'temp/'
    imageList = os.listdir(imagePath)
    trainingImage = []
    labelImage = []

    for i,item in enumerate(imageList):
        img = cv2.imread(imagePath+item)
        img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
        trainingImage.append(np.asarray(img,dtype='uint8'))
        labelImage.append(i)
    model = cv2.createLBPHFaceRecognizer()
    model.train(np.asarray(trainingImage),np.asarray(labelImage))
    count = -1
    try:
        with open('train/FaceData/imagecount.txt','rb') as f:
            count = int(f.read())

        with open('train/FaceData/imagecount.txt','wb') as f:
            f.write(str(count+1))

        with open('train/FaceData/nameList','r') as f:
            nameList = pickle.load(f)
            nameList.append(personName)

        with open('train/FaceData/nameList','wb') as f:
            pickle.dump(nameList,f)
    except:
        with open('train/FaceData/imagecount.txt','wb') as f:
            f.write('0')

        with open('train/FaceData/nameList','wb') as f:
            nameList.append(personName)
            pickle.dump(nameList,f)
    model.save('train/FaceData/'+str(count+1)+'.cv2')
```

```
def __del__(self):
    print('Image Trained Succefully.')
    for item in sorted(os.listdir('temp/')):
            os.unlink('temp/'+item)
    print('Bye')
```

```
if __name__ == '__main__':
    if not os.path.exists('train'):
        os.makedirs('train/FaceData/')
        os.makedirs('temp/')
        print('Directory created')
    try:
        personName = 'Your Name' # Change person name than run
        captrainObj = CaptureAndTrain() # Creating object for training
        if len(sys.argv)<=1:
            raise
        if sys.argv[1] == '-pn':
            if len(sys.argv[2])>0:
                personName = sys.argv[2]
                if sys.argv[3] == '-dir':
                    if os.path.exists(sys.argv[4]):
                        captrainObj.captureFromDir(sys.argv[4])
                        captrainObj.trainImage(personName)
                    else:
                        raise
                elif sys.argv[3]=='-cam':
                    cameraNumber = 0
                    try:
                        cameraNumber = int(sys.argv[4])
                    except:
                        pass
                    captrainObj.captureFromCam(cameraNumber)
                    captrainObj.trainImage(personName)
                else:
                    raise
            else:
                raise
        else:
            raise
    except:
        print('Please pass arguments , while running program.')
        print("-pn 'NameOfPerson' -dir Absclute_Path_To_Directory_Where_Training_Image_Is_Present ")
        print("-pn 'NameOfPerson' -cam Camera_Number_Default_Is_Set_To_0")
        raise
```

Source code for checking face

```
import cv2
import numpy as np
import pickle
import sys
import os
import winsound
import zerosms
def zerosmssend():
    #user_name=getpass.getpass("Username:")
    user_name="6280225149"
    #user_password=getpass.getpass("Password:")
    user_password="way2sms123PK"
    msg="Prajwal Hello"
    #send_to=getpass.getpass("Send To:")
    send_to="6280225149"
    zerosms.sms(phno=user_name,passwd=user_password,message=msg,receivernum=send_to)

face_classifier = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
flag = 0
```

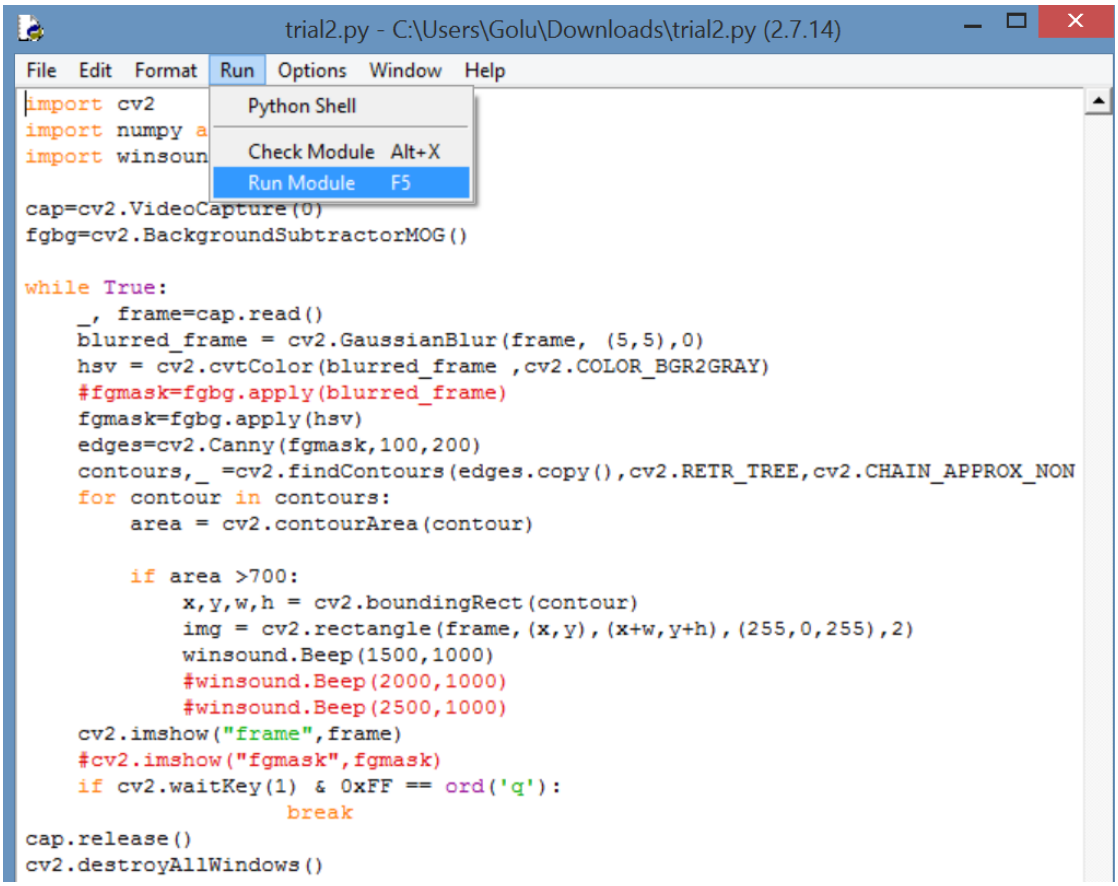
```python
def face_detector(frame):

    # Convert image to grayscale
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray, 1.3, 5)
    if faces is ():
        return None,None

    roi = []
    cord = []

    for (x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,255),2)
        cv2.rectangle(frame, (x,y-30), (x+w,y), (0,255,255), -1)
        cv2.putText(frame, 'None', (x+2,y-10), 4, 0.6, (255,255,255), 1)
        # cv2.circle(frame, (x+130,y), 1, (0,0,255), -1)
        croi = frame[y:y+h, x:x+w]
        l = []
        l.append((x,y))
        l.append((x+w,y+h))
        cord.append(l)
        roi.append(cv2.resize(croi, (200, 200)))
    return roi,cord
```

```python
if __name__ == '__main__':

    totalModel = 0

    with open('train/FaceData/imagecount.txt','r') as f:
        totalModel = int(f.read())+1

    modelNameList = []

    for i in range(totalModel):
        modelNameList.append('train/FaceData/'+str(i)+'.cv2')

    nameList = []

    modelList = []

    pmodelList = []

    for i in range(totalModel):
        modelList.append(cv2.createLBPHFaceRecognizer())

    for i in range(totalModel):
        modelList[i].load(modelNameList[i])

    with open('train/FaceData/nameList','r') as f:
            nameList = pickle.load(f)
```

```
try:

    if sys.argv[1]=='-cam':

        camNumber = 0

        try:
            camNumber =sys.argv[2]
        except:
            pass

        cap = cv2.VideoCapture(camNumber)

        while True:

            ret, frame = cap.read()

            frame = cv2.resize(frame, (500,400))

            faces,positions = face_detector(frame)

            if faces is None:
                cv2.imshow('Face Recognition', frame )
            else:
                try:
                    for i,face in enumerate(faces):
                        face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
                        flag = 0

                        for j,model in enumerate(modelList):
                            results = model.predict(face)

                            if results[1] < 500:
                                confidence = int( 100 * (1 - (results[1])/400) )
                                if confidence>=75:
                                    flag = 1
                                    cv2.rectangle(frame, (positions[i][0][0],positions[i][0][1]-30), (positions[i][1][0],positions[i][0][1]), (0,0,255), -1)
                                    cv2.putText(frame, nameList[j], (positions[i][0][0]+2,positions[i][0][1]-10), 4, 0.6, (255,255,255), 1)
                                    cv2.rectangle(frame, positions[i][0], positions[i][1], (0,0,255), 2)
                                    break
```

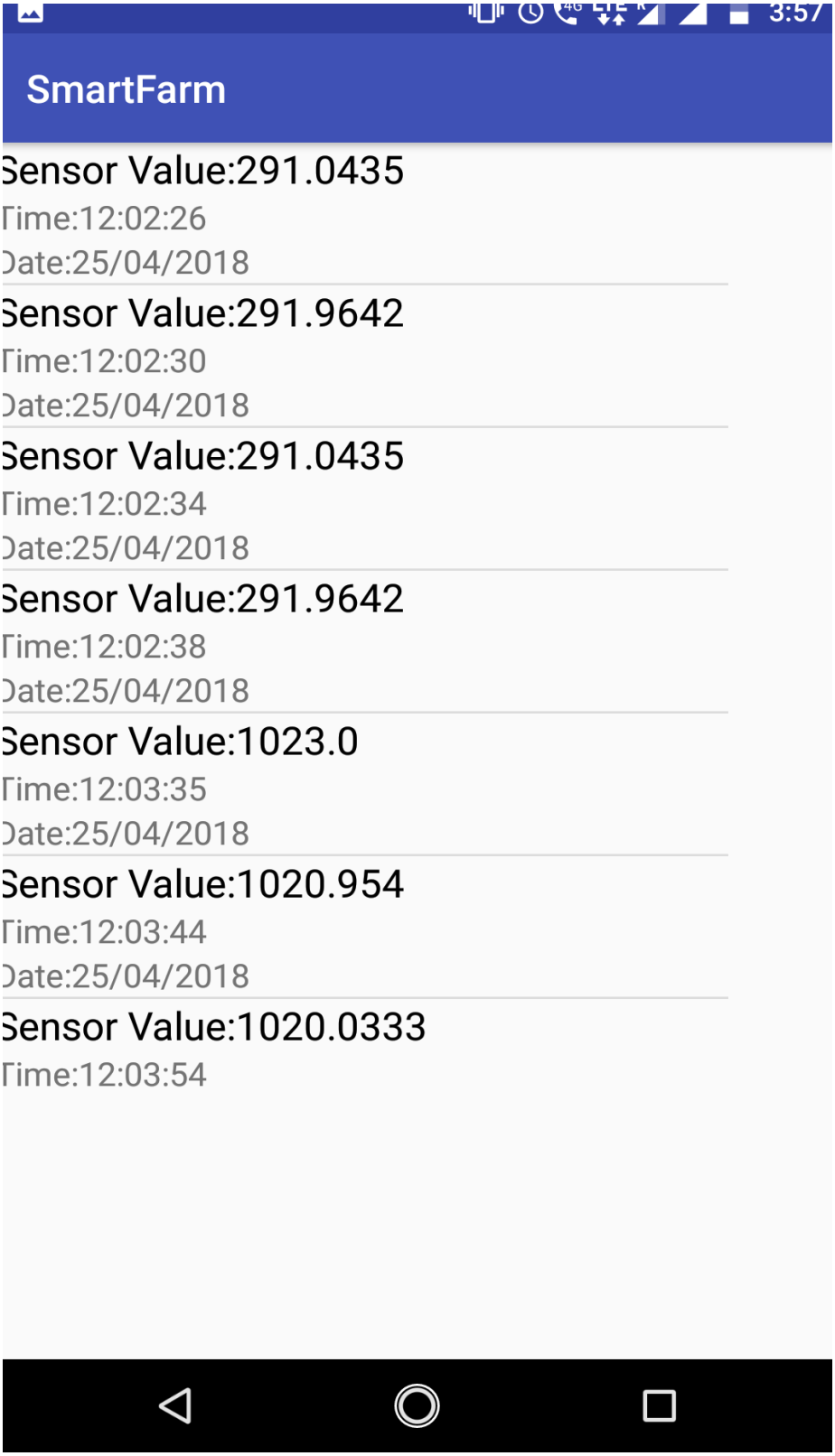
```
                            if results[1] < 500:
                                confidence = int( 100 * (1 - (results[1])/400) )
                                if confidence>=75:
                                    flag = 1
                                    cv2.rectangle(frame, (positions[i][0][0],positions[i][0][1]-30), (positions[i][1][0],positions[i][0][1]), (0,0,255), -1)
                                    cv2.putText(frame, nameList[j], (positions[i][0][0]+2,positions[i][0][1]-10), 4, 0.6, (255,255,255), 1)
                                    cv2.rectangle(frame, positions[i][0], positions[i][1], (0,0,255), 2)
                                    break
                        if flag == 0:
                            winsound.Beep(2500,100)
                            #zerosmssend()

                except:
                    raise

            cv2.imshow('Face Recognition', frame )

            if cv2.waitKey(1) & 0xFF == ord('q'):
                break

        cap.release()
        cv2.destroyAllWindows()

    else:
        raise

except:
    raise
```

10.3 MODULE3 BIRD DETECTION

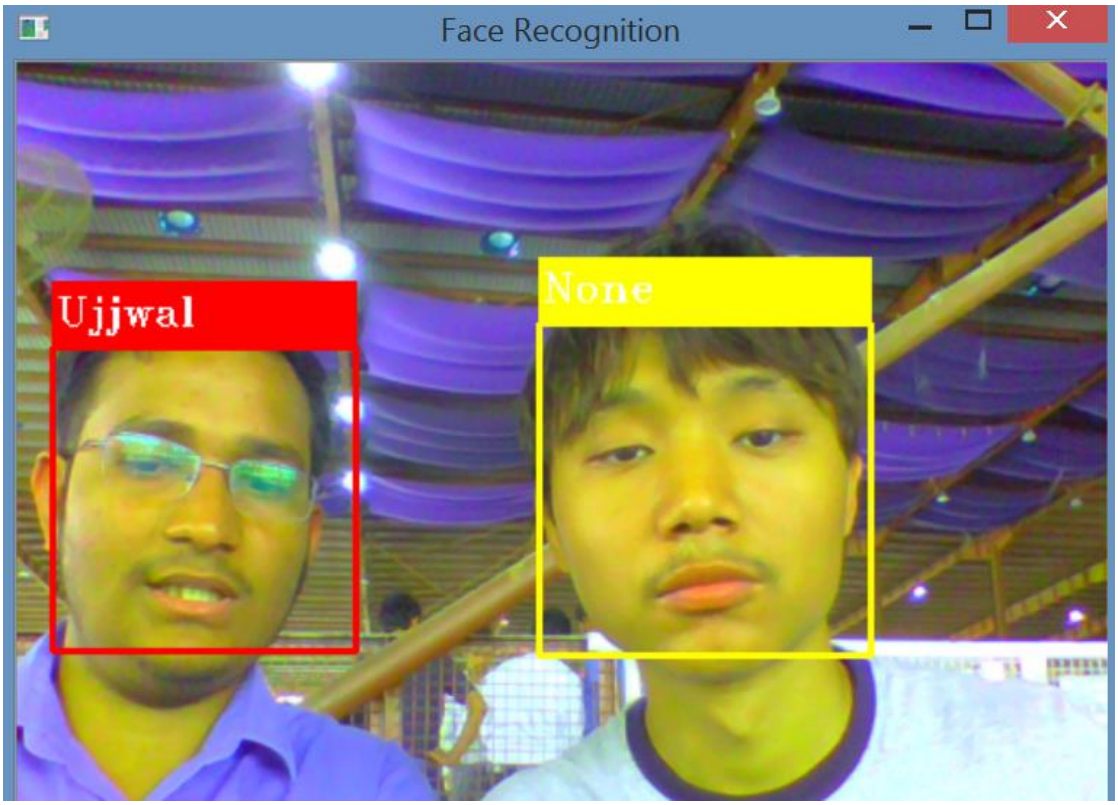


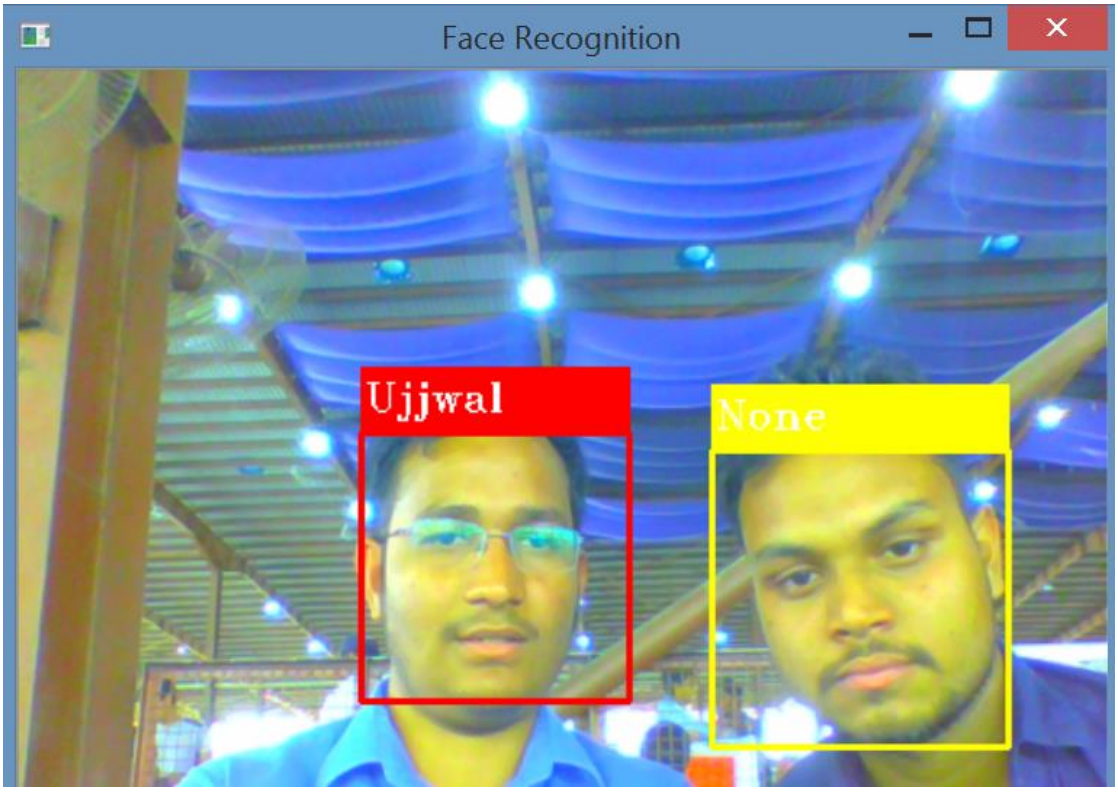
11. PROJECT SNAPSHOTS

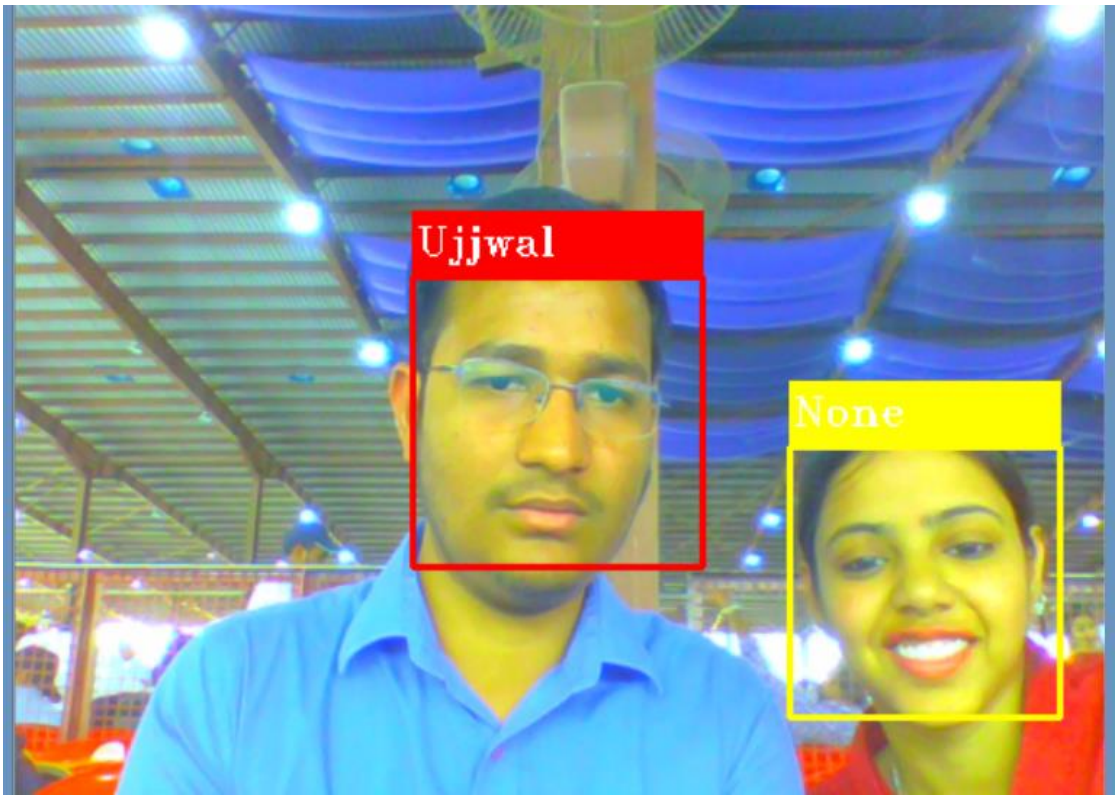
11.1 MODULE1 SMART IRRIGATION



11.2 MODULE2 FACE DETECTION







11.3 MODULE3 BIRD DETECTION

trial2.py - C:\Users\Golu\Downloads\trial2.py (2.7.14)

File Edit Format Run Options Window Help

Python Shell
Check Module Alt+X
Run Module F5

```
import cv2
import numpy a
import winsoun

cap=cv2.VideoCapture(0)
fgbg=cv2.BackgroundSubtractorMOG()

while True:
    _, frame=cap.read()
    blurred_frame = cv2.GaussianBlur(frame, (5,5),0)
    hsv = cv2.cvtColor(blurred_frame ,cv2.COLOR_BGR2GRAY)
    #fgmask=fgbg.apply(blurred_frame)
    fgmask=fgbg.apply(hsv)
    edges=cv2.Canny(fgmask,100,200)
    contours,_ =cv2.findContours(edges.copy(),cv2.RETR_TREE,cv2.CHAIN_APPROX_NON
    for contour in contours:
        area = cv2.contourArea(contour)

        if area >700:
            x,y,w,h = cv2.boundingRect(contour)
            img = cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,255),2)
            winsound.Beep(1500,1000)
            #winsound.Beep(2000,1000)
            #winsound.Beep(2500,1000)
    cv2.imshow("frame",frame)
    #cv2.imshow("fgmask",fgmask)
    if cv2.waitKey(1) & 0xFF == ord('q'):
                    break
cap.release()
cv2.destroyAllWindows()
```

12. BIBLIOGRAPHY

https://www.python.org/

https://opencv.org/

https://developer.android.com

https://www.pyimagesearch.com/

https://www.raspberrypi.org/

https://stackoverflow.com/