BONAFIDE CERTIFICATE

Certified that this mini project work titled “SMART ATTENDANCE” is the bonafide work of **Premaswati V, Priyanka S, Priyashaw M, Sonali Kannan, Sowbarnikkaa S V** who carried out the project work during the even semester of the academic year 2017-2018 under the professional core course 15Z402 – Software Engineering.

SIGNATURE

Dr. S. Suriya

Associate Professor,
Department of CSE, PSG
College of Technology,
Coimbatore.

TABLE OF CONTENTS

SMART ATTENDANCE

A MINI PROJECT REPORT

Submitted towards the professional elective course

15Z402 Software Engineering

Premaswati V (16Z335)
Priyanka S (16z336)
Priyashaw M (16z337)
Sonali Kannan (16z347)
Sowbarnikkaa S V (16z349)

BACHELOR OF ENGINEERING

Branch: Computer Science and Engineering

April 2018

Department of Computer Science and Engineering PSG College of Technology, Coimbatore - 641 004.

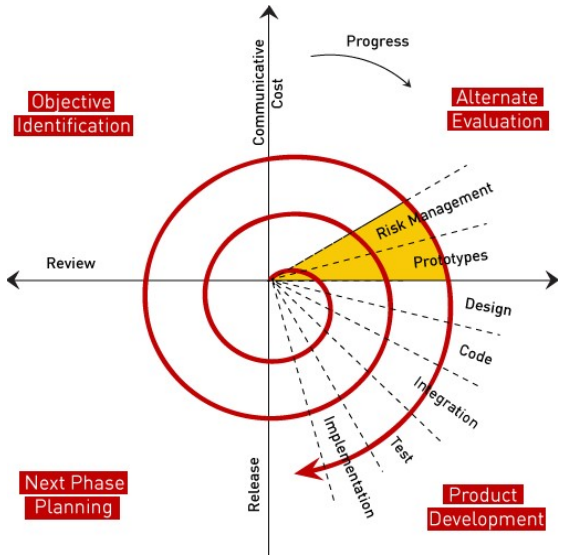
1. PROBLEM STATEMENT:

To develop a software that replaces the conventional attendance system which is time consuming, ambiguous and requires human effort. The software developed should be accurate and efficient in use and maintenance unlike the pen and paper method.

2. LIFE CYCLE MODEL :

SPIRAL MODEL

Spiral Model is also called as Risk-Driven Process model, which combines the idea of Waterfall and Incremental model but provides more emphasis on risk analysis.



Why spiral model?

- Emphasis more on risk analysis, hence avoidance of Risk is assured
- Using this model the software can be developed early and also the earlier working prototype is done - sooner users can point out the flaws
- The development process works in a systematic way
- Major edits are expected during the development cycle
- The development process is precisely documented yet scalable to the changes
- The scalability allows to make changes and add new functionality even at the relatively late stages

RISK MANAGEMENT APPROACHES

TYPES OF RISK :

1. Generic risks

- These risks are a potential threat to each software project.

2. Product-specific risks

- These risks are recognised by those with a clear understanding of the technology, the people and the environment which is specific to the software that is to be built.
- A method for recognising risks is to create item checklist.

S.NO	CATEGORIES	YES/NO
1	Product Size	YES
2	Process Definition	NO
3	Development environment	NO
4	Staff size and experience	YES

CATEGORIES OF RISK :

S.NO	CATEGORIES	YES/NO
1	Project Risk	YES
2	Technical Risk	NO
3	Business Risk	YES
4	Known Risks	YES
5	Predictable Risks	YES
6	Unpredictable Risks	NO

FACE DETECTION AND RECOGNITION ALGORITHMS:

S.NO	NAME	DESCRIPTION
1	Principal Component Analysis (PCA)	Given an s-dimensional vector representation of each face in a training set of images, PCA tends to find a t-dimensional subspace whose basis vectors correspond to the maximum variance direction in the original image space. This new subspace is normally lower dimensional (t<<s). If the image elements are considered as random variables, the PCA basis vectors are defined as eigenvectors of the scatter matrix.
2	Independent Component Analysis (ICA)	ICA minimizes both second-order and higher-order dependencies in the input data and attempts to find the basis along which the data (when projected onto them) are - statistically independent
3	Linear Discriminant Analysis (LDA)	LDA finds the vectors in the underlying space that best discriminate among classes. For all samples of all classes the between-class scatter matrix SB and the within-class scatter matrix SW are defined. The goal is to maximize SB while minimizing SW, in other words, maximize the ratio det|SB|/ det|SW| . This ratio is maximized when the column vectors of the projection matrix are the eigenvectors of (SW^-1 × SB).
4	Evolutionary Pursuit (EP)	An eigenspace-based adaptive approach that searches for the best set of projection axes in order to maximize a fitness function, measuring at the same time the classification accuracy and generalization ability of the system. Because the dimension of the solution space of this problem is too big, it is solved using a specific kind of genetic algorithm called EP.
5	Elastic Bunch Graph Matching (EBGM).	All human faces share a similar topological structure. Faces are represented as graphs, with nodes positioned at fiducial points. (exes, nose...) and edges labeled with 2-D distance vectors. Each node contains a set of 40 complex Gabor wavelet coefficients at different scales and orientations (phase, amplitude). They are called "jets". Recognition is based on labeled graphs. A labeled graph is a set of nodes connected by edges, nodes are labeled with jets, edges are labeled with distances.

S.NO	NAME	DESCRIPTION
6	Kernal methods	The face manifold in subspace need not be linear. Kernel methods are a generalization of linear methods. Direct non-linear manifold schemes are explored to learn this non-linear manifold.
7	Trace Transform	A generalization of the Radon transform, is a new tool for image processing which can be used for recognizing objects under transformations, e.g. rotation, translation and scaling. To produce the Trace transform one computes a functional along tracing lines of an image. Different Trace transforms can be produced from an image using different trace functionals.
8	Active Appearance Model (AAM)	An integrated statistical model which combines a model of shape variation with a model of the appearance variations in a shape-normalized frame. An AAM contains a statistical model of the shape and gray-level appearance of the object of interest which can generalize to almost any valid example. Matching to an image involves finding model parameters which minimize the difference between the image and a synthesized model example projected into the image.
9	3-D Morphable Model	Human face is a surface lying in the 3-D space intrinsically. Therefore the 3-D model should be better for representing faces, especially to handle facial variations, such as pose, illumination etc. Blantz et al. proposed a method based on a 3-D morphable face model that encodes shape and texture in terms of model parameters, and algorithm that recovers these parameters from a single image of a face
10	3-D Face Recognition	The main novelty of this approach is the ability to compare surfaces independent of natural deformations resulting from facial expressions. First, the range image and the texture of the face are acquired. Next, the range image is preprocessed by removing certain parts such as hair, which can complicate the recognition process. Finally, a canonical form of the facial surface is computed. Such a representation is insensitive to head orientations and facial expressions, thus significantly simplifying the recognition procedure. The recognition itself is performed on the canonical surfaces.

FACE ATTRIBUTES:

- **Age:** an estimated "visual age" number in years. It is how old a person looks like rather than the actual biological age.
- **Gender:** male or female.
- **Smile:** smile intensity, a number between [0,1].
- **FacialHair:** return lengths in three facial hair areas: moustache, beard and sideburns. The length is a number between [0,1]. 0 for no facial hair in this area, 1 for long or very thick facial hairs in this area.
- **HeadPose:** 3-D roll/yaw/pitch angles for face direction. Note, Pitch value is a reserved field and will always return 0.
- **Glasses:** glasses type. Values include 'NoGlasses', 'ReadingGlasses', 'Sunglasses', 'SwimmingGoggles'.
- **Emotion:** emotion intensity, including neutral, anger, contempt, disgust, fear, happiness, sadness and surprise.
- **Hair**: group of hair values indicating whether the hair is visible, bald, and hair color if hair is visible.
- **Makeup**: whether eye, lip areas are made-up or not.
- **Accessories**: accessories around face, including 'headwear', 'glasses' and 'mask'. Empty array means no accessories detected. Note this is after a face is detected. Large mask could result in no face to be detected.
- **Blur**: face is blurry or not. Level returns 'Low', 'Medium' or 'High'. Value returns a number between [0,1], the larger the blurrier.
- **Exposure**: face exposure level. Level returns 'GoodExposure', 'OverExposure' or 'UnderExposure'.
- **Noise**: noise level of face pixels. Level returns 'Low', 'Medium' and 'High'. Value returns a number between [0,1], the larger the noisier

3. SRS

1. PREFACE

This SMART ATTENDANCE SYSTEM is the first version developed.

2. INTRODUCTION

SCOPE - Use of the face detection and recognition system in lieu of the traditional methods will provide a fast and effective method of capturing student attendance to some extent of accuracy.

PURPOSE - Use of the face detection and recognition system .A fast and effective method of capturing student attendance accurately.Secure, stable and robust storage of the system records.

3. USER REQUIREMENT DOCUMENT:

This system must be able to mark attendance using face recognition techniques. A variety of non-functional characteristics were likely considered which includes the following,

- The application should be portable and possible to users .
- Since the application will be displaying attendance for the users, the response time for a user or a viewer should be not be greater than 3-4 seconds for a respectable internet connection speed.
- Error handling should be implemented and the application should be able to handle all run time errors.
- The application should be flexible for future enhancements, for example, the addition of a few more research analysis questions.

4. SYSTEM ARCHITECTURE

The python platform(software requirement) connects with the laptop webcam(hardware requirement) for facial inputs.
The python platform along with cognitive faces helps to detect, identify and match the faces with various platforms.

5. SYSTEM REQUIREMENT SPECIFICATION:

- **Adding new students:**

If there are any lateral students, we could easily add their details in the database provided a key user (faculties or any person assigned for this purpose) to the system.

- **Report students:**

When the students have enrolled in the class, they are now able to check on their current attendance situation through the application. In the system, they will be shown a page that gives them the whole attendance status in the semester.

- **Report faculties**:

Teachers can also update the attendance in case of any on-duty(OD), exception or medical reason. This can be done by logging in into their account on their page.

6. SYSTEM MODELS:

The conceptual model of the software includes:.

Structure – Modules involve capturing student images ,detecting live faces, matching using pca,spreadsheet.

7.SYSTEM EVOLUTION:

The main objective of software evolution is ensuring the reliability and flexibility of the system.

8.APPENDIX:

It consists of 2 portions namely

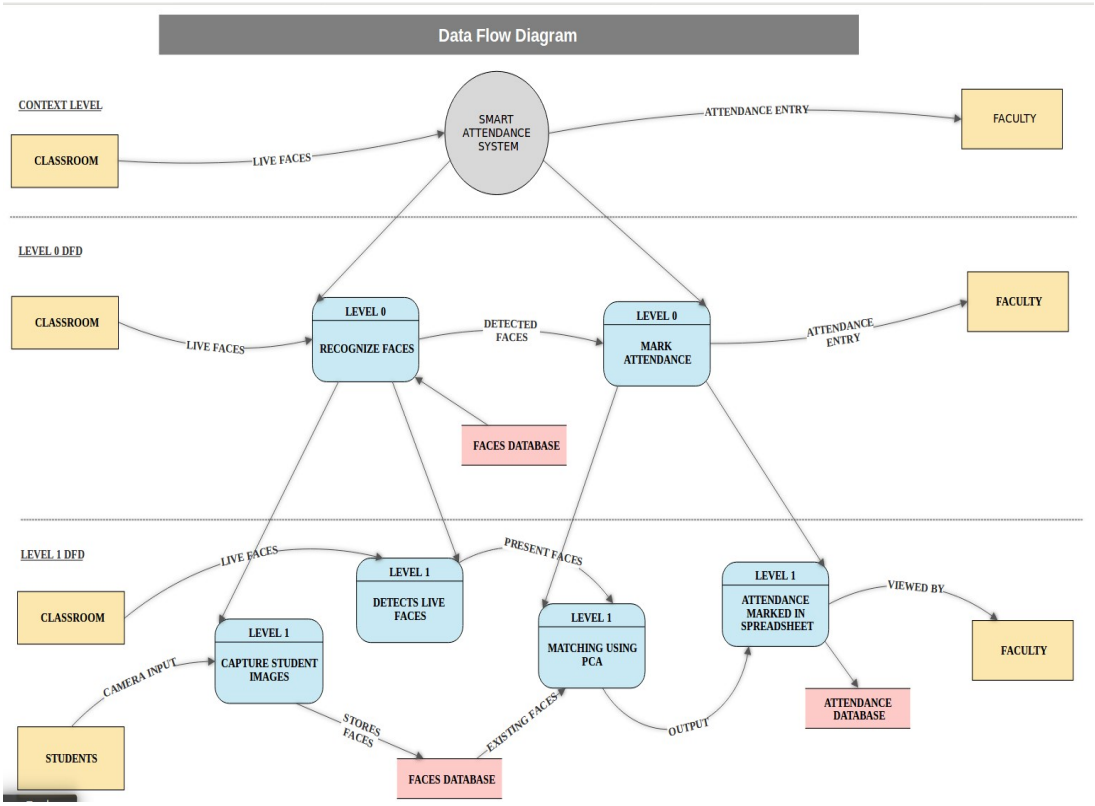
1)Hardware configuration details

- Web Cam is the basic hardware requirement.

2)Database connectivity details

- DB Browser for SQLite is the software used to create,alter and modify the SQLite databases.
- SQLite3 is the package that connects the python code with database.

4. DESIGNING Using DFD



5. MODULES DESCRIPTION

There are totally four modules which completes the functionality of our SMART ATTENDANCE SYSTEM.

5.1 MODULE – 1: CAPTURING STUDENT IMAGES

- INPUTS STUDENT NAME AND ROLL NUMBER
- CAPTURES IMAGES OF THE STUDENT
- CREATES A FILE WITH 20 IMAGES FOR ONE STUDENT

- UPDATES THE DATABASE

	ID	Name	Roll	personID
	Filter	Filter	Filter	Filter
1	35	Swati	16z335	a5cf02bd-5...
2	21	Sanju	16z321	dc4d22b7-a...

FORMAT OF IMAGES USED :

We have stored the images in PNG format and not in the usual JPEG format. Because , JPEG file format is not compatible with the MICROSOFT COGNITIVE SERVICES (Face API) which we have used for generating unique Person ID.

FUNCTIONS USED :

- DLIB - Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems.
- CV2 - OpenCV (Open Source Computer Vision) is a library of programming functions for realtime computer vision.
- **`Get_frontal_face`** – Detects the faces that is captured by the camera and stores it in detector variable.
- **`dets = detector(img,1)`** - Run the face detector, upsampling the image 1 time to find smaller faces.
- **`cv2.imread`** - The function imread loads an image from the specified file and returns it. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format), the function returns an empty matrix.
- **`cv2.imwrite`** – Saves image to the file

5.2 MODULE-2 : DETECTING LIVE FACES

- CAPTURES CLASSROOM IMAGE.
- OUTPUTS THE NUMBER OF FACES DETECTED IN THE CLASSROOM IMAGE
- CROPS THE DETECTED FACES AND STORES IT IN A SEPARATE FILE CALLED *'Cropped_faces'*
- CREATES A PERSON ID FOR EACH UNIQUE FAC

FUNCTIONS USED :

- **`img = cv2.imread()`** – This function is used to read an image from a file present in the same directory.
- **`dets = detector()`** – Counts the number of faces in the image.
- **`os.makedirs('./Cropped_faces')`** – Creates a file called 'Cropped_faces'.
- **`cv2.imwrite('./Cropped_faces/face' + str(i + 1) + '.png', img[d.top():d.bottom(), d.left():d.right()])`** - This will save the image in PNG format in the specified directory.

5.3 MODULE 3: MATCHING USING PCA

- MATCHES EACH AND EVERY IMAGE PRESENT IN *'Cropped_faces'* WITH THE ALREADY EXISTING STUDENT IMAGES.
- OUTPUTS THE NAME OF THE STUDENT WHOSE FACE MATCHES.

FUNCTIONS USED :

- **`res = CF.face.identify(faceIds, personGroupId)`** – matches the generated faceIds for the classroom picture against the several faceIds present in personGroupId
- **`c.execute("SELECT * FROM Students WHERE personID = ?", (personId,))`** – Checks whether the PersonId is present in the database and outputs the corresponding name.

5.4 MODULE 4: SPREADSHEET

- DISPLAYS THE ATTENDANCE INFORMATION IN THE FORM OF ROWS AND COLUMNS.
- OPENS A SPREADSHEET WITH 3 COLUMNS –
    - ROLL NUMBER
    - NAME
    - DATE
- STORES '1' IN THE DATE COLUMN IF THE STUDENT IS PRESENT ELSE IT LEAVES THE CELL BLANK

FUNCTIONS USED :

- Openpxl is python library for reading and writing Excel 2010 files.
- Openpxl along with sqlite3 is used for reading data from database and storing it in Excel sheets

6. TESTING

Why is testing done?

It is done to check the "**correctness**" of the software just developed.

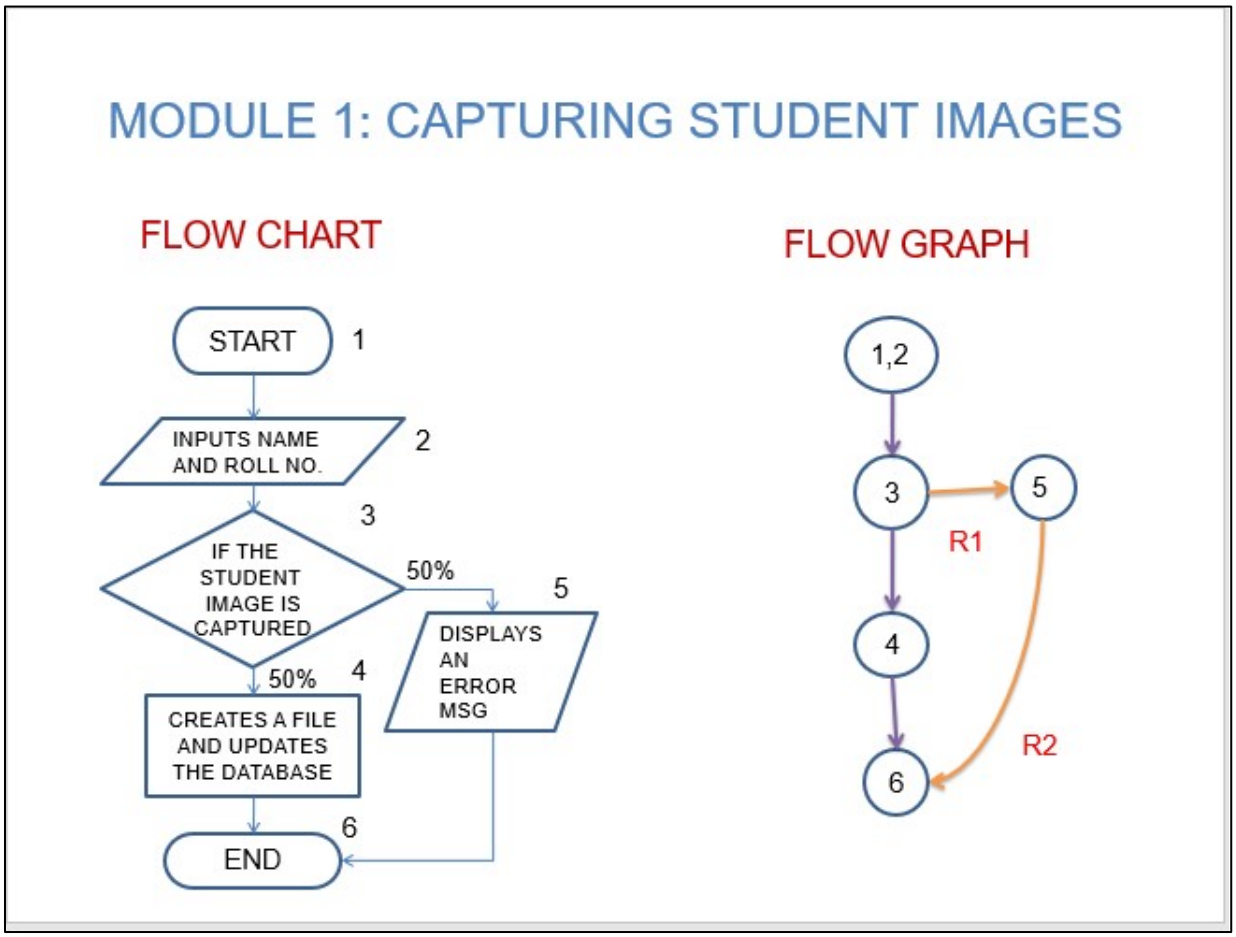
How can testing be done?

- white box testing
- black box testing
- testing strategies

6.1 Flow graph Notation – Cyclomatic Complexity Analysis

White box testing

- It is used to test the internal structures or workings of an application or a software.
- Also called clear box testing or glass box testing



CYCLOMATIC COMPLEXITY

V(G) = P (Predicate nodes) + 1

= 1(node 3)+1

= 2

V(G) = E (Edges) - N (Nodes) + 2

= 5 – 5 + 2

= 2

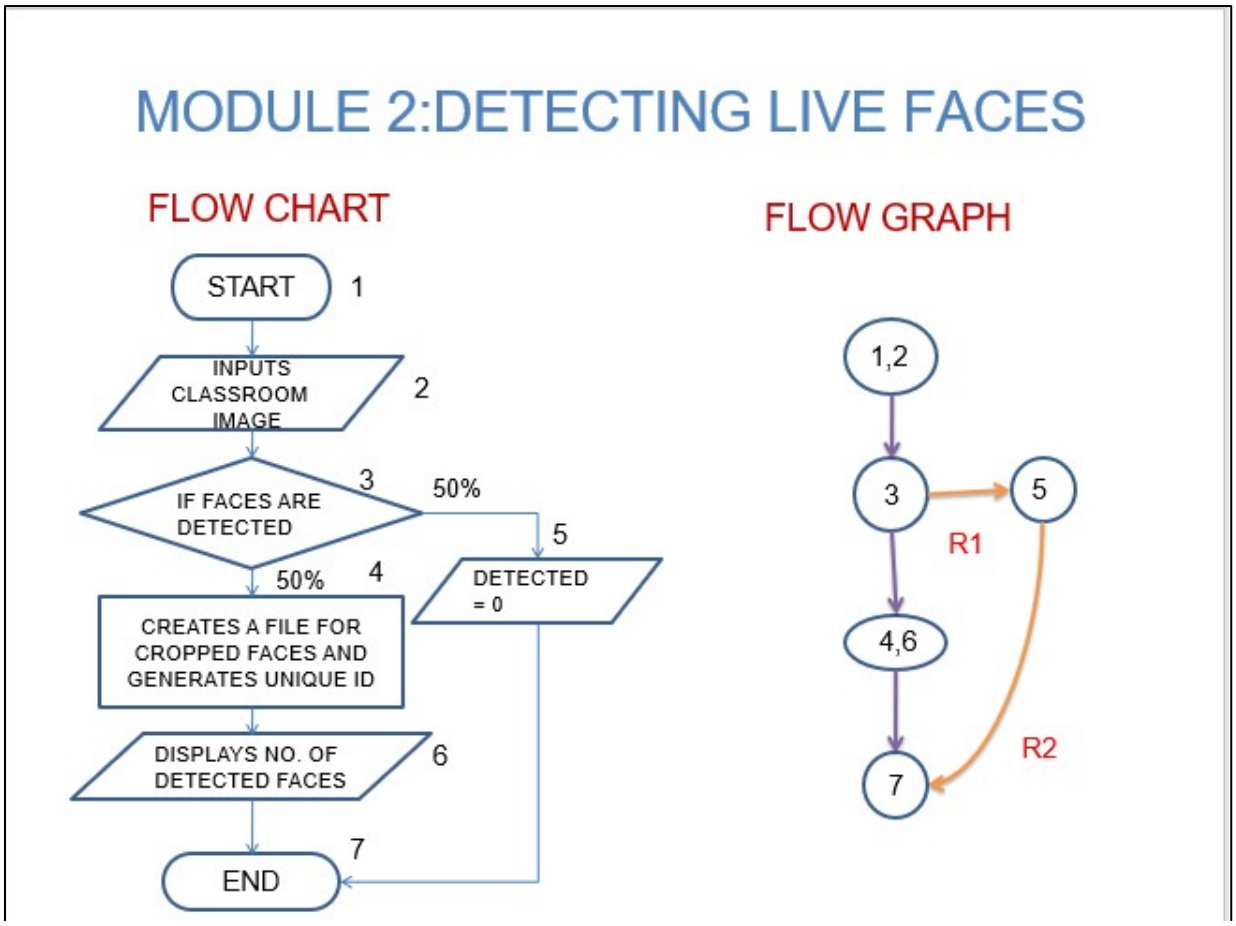
V(G) = R (Total no. of regions)

= 2 (R1, R2)

INDEPENDENT PATHS

PATH 1: 1-2-3-4-6

PATH 2 : 1-2-3-5-6



CYCLOMATIC COMPLEXITY

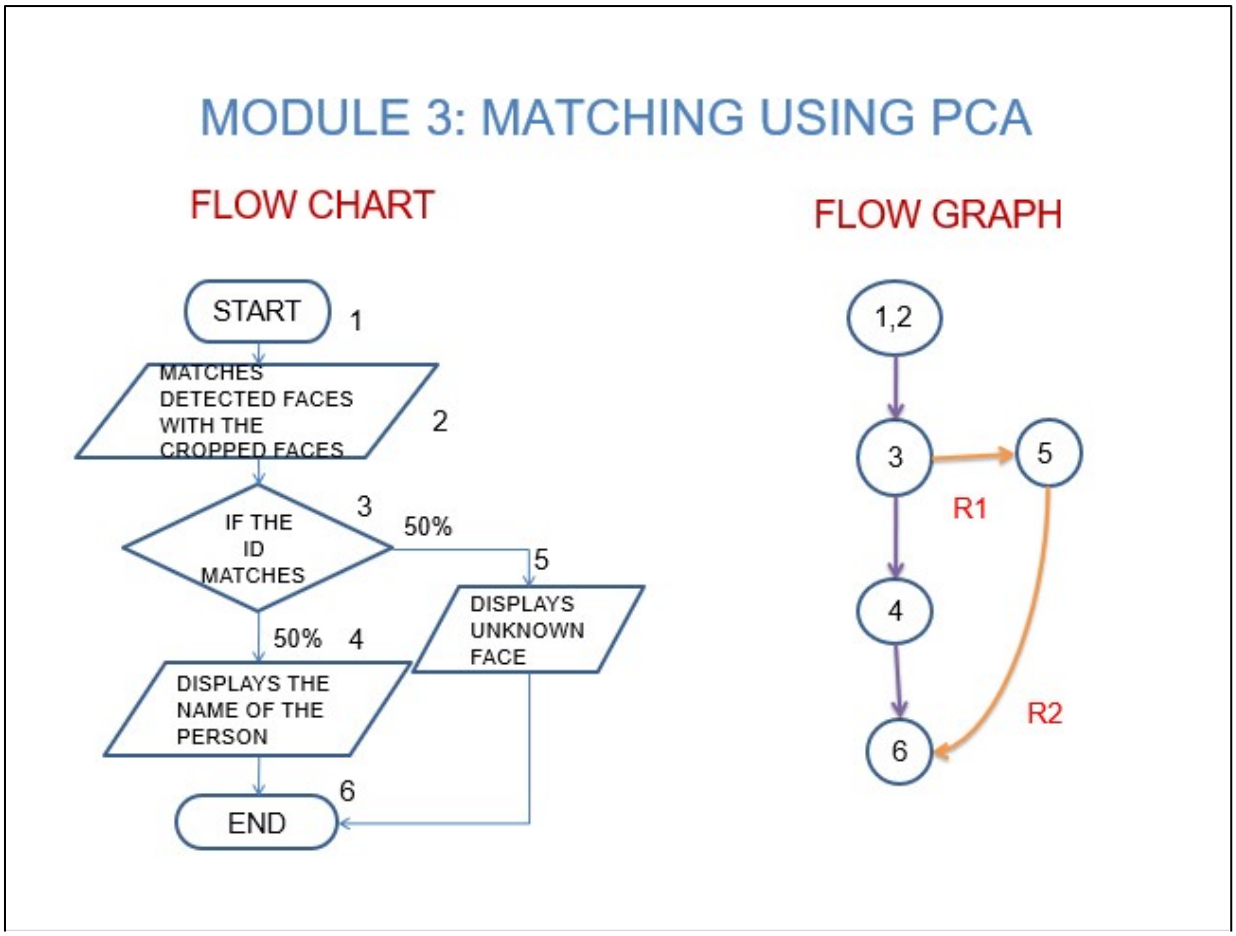
V(G) = P (Predicate nodes) + 1
= 1(node 3)+1
= 2

V(G) = E (Edges) - N (Nodes) + 2
= 5 – 5 + 2
= 2

V(G) = R (Total no. of regions)
= 2 (R1, R2)

INDEPENDENT PATHS

PATH 1: 1-2-3-4-6-7
PATH 2 : 1-2-3-5-7



CYCLOMATIC COMPLEXITY

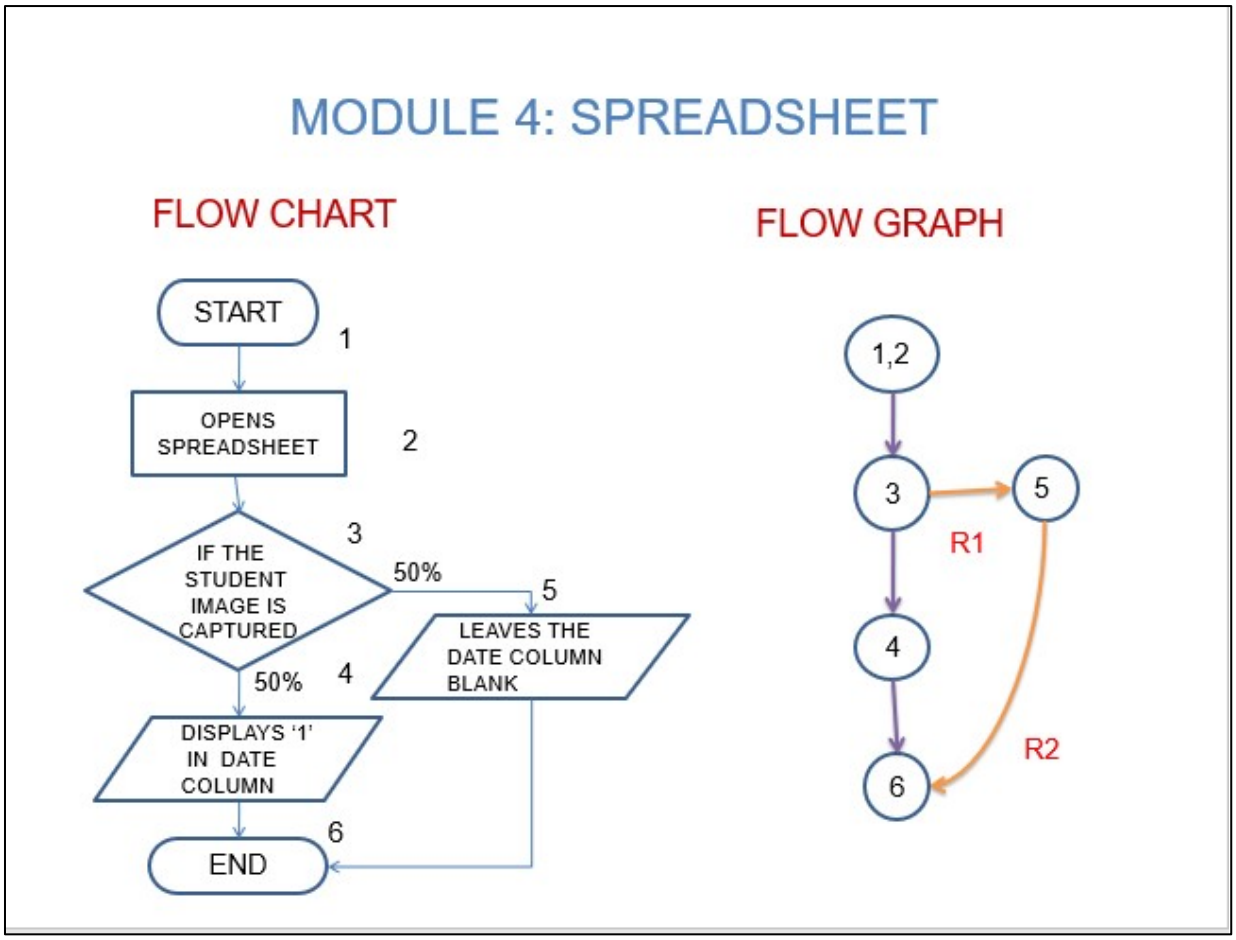
V(G) = P (Predicate nodes) + 1
= 1(node 3)+1
= 2

V(G) = E (Edges) - N (Nodes) + 2
= 5 – 5 + 2
= 2

V(G) = R (Total no. of regions)
= 2 (R1, R2)

INDEPENDENT PATHS

PATH 1: 1-2-3-4-6
PATH 2 : 1-2-3-5-6



CYCLOMATIC COMPLEXITY

V(G) = P (Predicate nodes) + 1
= 1(node 3)+1
= 2

V(G) = E (Edges) - N (Nodes) + 2
= 5 – 5 + 2
= 2

V(G) = R (Total no. of regions)
= 2 (R1, R2)

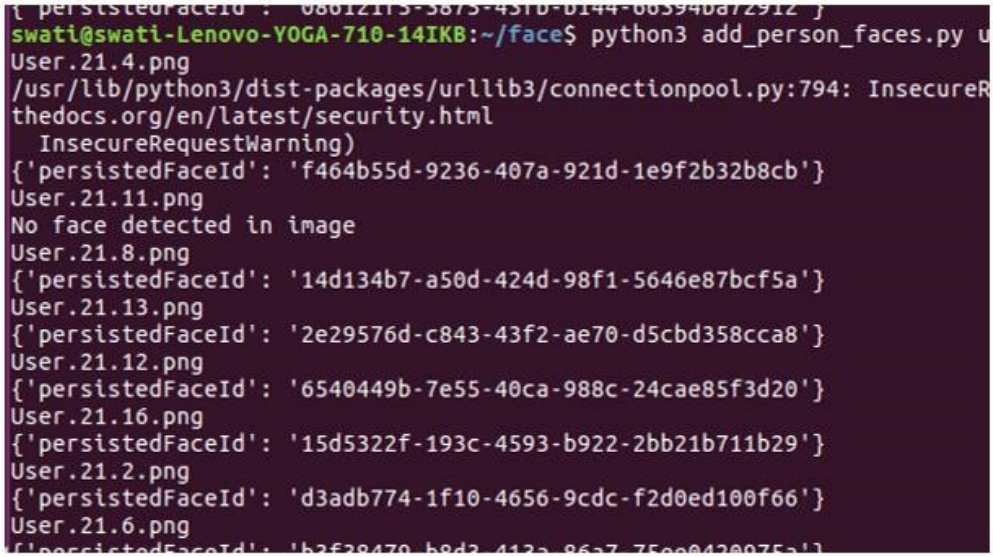
INDEPENDENT PATHS

PATH 1: 1-2-3-4-6
PATH 2 : 1-2-3-5-6

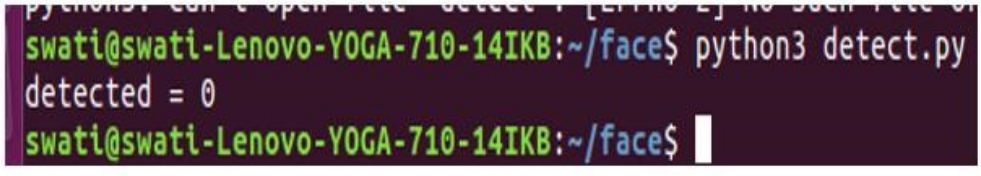
Black box testing:

- This type of testing is done when the internal structure or design or implementation of the software is not known to the tester.
- Also known as behavioral testing.

MODULE 1: CAPTURING STUDENT IMAGES



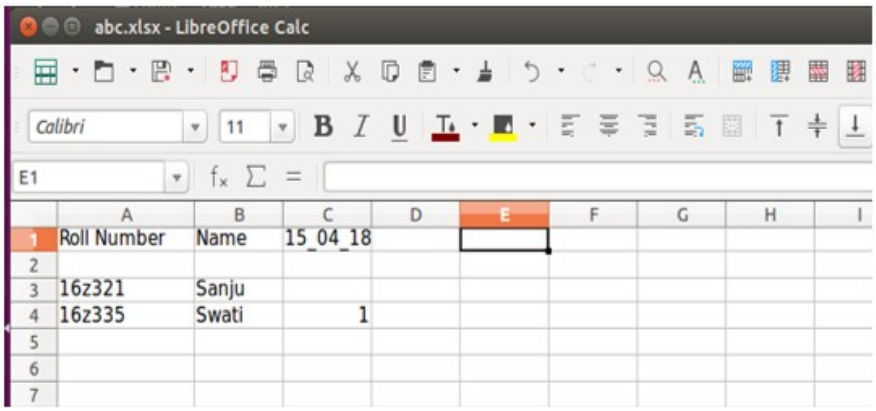
MODULE 2:DETECTING LIVE FACES



MODULE 3: MATCHING USING PCA

```
swati@swati-Lenovo-YOGA-710-14IKB:~/face$ python3 identify2.py
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:794: Ins
thedocs.org/en/latest/security.html
  InsecureRequestWarning)
face1.png
[{'faceId': '8656526a-5af5-4632-901c-94db46583228', 'candidates':
Swati recognized
face2.png
[{'faceId': '52ef71e3-8ee7-48b8-bbe5-9d2f7b3233fd', 'candidates':
Unknown
swati@swati-Lenovo-YOGA-710-14IKB:~/face$
```

MODULE 4: SPREADSHEET



Testing strategies

Unit testing:

Module interface – The interface properly flows in and out of the program.

Local data structures – The data structures used such as array, dictionary, lists, etc,.. helps to maintain the data integrity in all steps of execution . The images are stored in files and the final attendance is updated in the excel sheets.

Boundary conditions – Since the array size is 100 the module operates at 0 and 99 and not above or below that.

Independent paths – All the statements are executed at least once so that the invalid conditions are also checked.

Error handling paths – Whenever the an error is encountered the execution stops and comes out of the module.

Integration testing:

We used bottom up approach here.

System Testing :

1. Recovery Tests – No recovery when it is failed except the files in database.
2. Stress Test – when more amount of stress is given the system will fail-soft
3. Security Test – It provides security to some extent
4. Performance – Low performance

7. Conclusion:

Thus, this smart attendance system provides an efficient and effective way for the students to view the attendance and teachers to update the attendance.