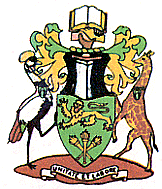
**UNIVERSITY OF NAIROBI**



**School of Computing and Informatics**

**Milestone 2: Project Documentation**

**Faceby:**

**A Raspberry Pi driven door lock system using Facial Recognition**

By

**Wambugu Isaac Waweru**

**P15/2548/2010**

**Supervisor: Mr. Lawrence Muchemi**

## DECLARATION

I hereby declare that this project is my original work, and has to the best of my knowledge not been submitted to any other institution of higher learning

Student: Isaac Waweru Reg. No: P15/2548/2010

Signature: …………………………………. Date: ……………………………………..

I declare that this project has been done under my supervision as the University supervisor of this project.

Supervisor: Mr. Lawrence Muchemi

Signature: …………………………………. Date: ……………………………………..

## ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to my project supervisor Mr. Muchemi for his commitment, encouragement and support during the development of the project. The project would not have been the same without your continuous guidance and support.

Secondly, I would like to thank my fellow class mates for their encouragement, support and constructive criticism. They provided assistance when I was stuck or in a dilemma and for that I thank you.

Finally, I would like to appreciate my parents and sisters for their support both financially and mentally during the project construction period. I would not have made it without your help.

## ABSTRACT

Face recognition is a field that is still under numerous researches from dedicated technology enthusiast. The reason being that it is not very accurate compared to other biometric methods such as finger print recognition and iris recognition. The reason for facial recognition is the great user experience it brings on the table. Users respond more positively to facial recognition better than any other biometric authentication. Raspberry pi is a new kid in the world of electronics. It mimics a Linux operating system and has a GPIO unit component that helps programs interact with the hardware components. It is therefore possible to communicate with an external hardware component by sending binary digits to the hardware through the GPIO (General Purpose Input Output) pins.

Faceby is a system that uses facial recognition to provide authentication system. The software module encompasses facial recognition algorithms including the eigenfaces algorithm for face training and recognition. The hardware module uses a relay circuit that helps to separate both circuits; one circuit that is attached to the raspberry pi, and the other that is attached to the door motor. Once the identity of the user has been confirmed, the software grants him/her access enter the room.

Table of Contents

[DECLARATION 2](#_Toc378896202)

[ACKNOWLEDGEMENT 3](#_Toc378896203)

[ABSTRACT 4](#_Toc378896204)

[INTRODUCTION 7](#_Toc378896205)

[Background Information 7](#_Toc378896206)

[Problem Statement 7](#_Toc378896207)

[Project Justification 8](#_Toc378896208)

[Project Objectives 8](#_Toc378896209)

[LITERATURE REVIEW 9](#_Toc378896210)

[Introduction 9](#_Toc378896211)

[Facial Detection 9](#_Toc378896212)

[Face Preprocessing 10](#_Toc378896213)

[Machine Learning – Training 11](#_Toc378896214)

[Face Recognition 12](#_Toc378896215)

[Raspberry Pi 12](#_Toc378896216)

[Algorithms 13](#_Toc378896217)

[METHODOLOGY 14](#_Toc378896218)

[Introduction 14](#_Toc378896219)

[System Development 14](#_Toc378896220)

[Testing and Evaluation 15](#_Toc378896221)

[Resources 16](#_Toc378896222)

[Project Activity Plan 17](#_Toc378896223)

[Project Budget 18](#_Toc378896224)

[SYSTEM ANALYSIS AND DESIGN 19](#_Toc378896225)

[Feasibility Study 19](#_Toc378896226)

[Investigation of current system 19](#_Toc378896227)

[Logical design 20](#_Toc378896228)

[Physical design 20](#_Toc378896229)

[SYSTEM DESIGN 22](#_Toc378896230)

[Introduction 22](#_Toc378896231)

[System Modeling 22](#_Toc378896232)

[System Architecture 25](#_Toc378896233)

# INTRODUCTION

## Background Information

Security has become one of the most essential needs at home and in the work places of late. The need to protect ones belonging and assets from intruders has become a paramount need in the current age. Vandalism and theft in the work places and at homes are crimes at the increase in the police watch list. Placing padlocks and password enabled key lock systems have not been of great use since as technology increases so do the hacking community try to find loopholes in the system.

Therefore the trick is to be able to authorize and authenticate a person entering a building in real time and getting notifications of unknown people trying to gain access to the building or office. Facial Recognition is one of the ways to ways to authenticate a person. It helps to identify the identity of the person requesting to be granted access.

Once the face recognition authentication system is in place it can be used for a vast of projects including home automation, hardware manipulation and user software access. It can be incorporated in many projects. However it requires machine learning to collect the faces and train them to identify and recognize individuals.

Numerous works on face recognition has been done with great credit going to openCV, a popular open-source project that contains numerous algorithms on Computer Vision. The open-source community has also greatly improved various functions in the system thus resulting in a better and more user friendly system.

## Problem Statement

Security mechanisms and controls have been of paramount importance in country at this time. This has resulted in the government ordering specific security measures to be adhered to by the business community. The supermarkets and malls have employed guards to ensure the places of business are safe. However how safe are our homes?

The homes and workplace we live and work in are very insecure with a padlock or alarm system being the security systems at our homes. The padlocks can easily be picked and the alarm system can be bypassed by a hacker with a password. Our homes are therefore no longer save and thus there is need be to develop a security system that incorporate both identification and authentication mechanisms as a security measure.

## Project Justification

Facial recognition is an authentication mechanism that is hard enough to bypass. Therefore home security systems developed with biometric authentication are much safer than password authentication systems.

Since the current security systems are powered by electricity, it is easy to manipulate the power access using programs and GPIO pin inputs. Therefore it is possible to customize the system to automate any home appliance powered by electricity.

## Project Objectives

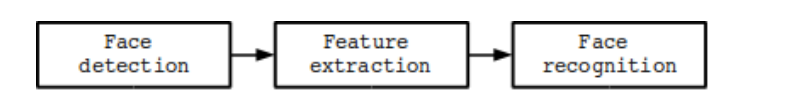
* System Development objectives
  + To demonstrate hardware and software interfacing
  + To enable users control electric appliances locally as well as remotely.
  + To demonstrate facial recognition authentication model in enhancing system security.
* Research objectives
  + To understand the working of facial recognition system.
  + To understand how home automation system works and how they can be customized.
* Project objectives
  + Perform requirement gathering through research
  + Perform system design for both hardware and software interface.
  + Perform system implementation.
  + Perform system testing and execution.
  + Perform system maintenance.

# LITERATURE REVIEW

## Introduction

Facial recognition involves detecting a face from a camera viewpoint, extracting facial features and comparing it with a database of stored faces. It recognizes the faces by measuring various face features including distance between the right and left eye, width of the nose, depth of the eye sockets, the shape of the cheekbones and the length of the jaw line. For an image to be captured, the face must be looking almost directly at the camera with little variance of light or facial expression from the image in the database. The light variation and poses causes inaccuracy in recognition thus causing facial recognition become a less popular authentication mechanism.

There are several stages involved in the facial recognition algorithm. These are face detection, feature extraction and face recognition.



Face detection poses various challenges. They include pose variation, Feature occlusion, Facial expression and Imaging conditions. For a video input, the process involves face detection, face location and face tracking.

Facial Detection

Face detection is the process of locating a face region in an image of a camera feed. This process does not care who the person is as long as it is a human face. Face detection is quite reliable in real-world conditions and applications. It has been incorporated in numerous camera applications to help users focus better when taking photographs. Viola and Jones brought a great change in the world of face detection when they invented the Haar-based cascade classifier for object detection. It is both fast and accurate at face detection.The Haar-based detector operates on the principle that in frontal faces the region with the eyes should be darker than the forehead and the cheeks and so on. It performs about 20 comparison stages before deciding if it is a face or not.

The other face detector algorithm is the LBP-based detectors. It is faster than Haars and has less licensing issues. The LBP-based uses the same idea as the Haars but uses histograms of pixel intensity comparisons such as edges, corners and flat regions. Both detectors can be trained using thousands of faces and tens of thousands of none-faces objects. It takes hours or weeks to train the cascades classifiers. However OpenCV comes with pre-trained Haars and LBP detectors. It makes it possible to detect frontal faces, side faces, eyes and noses by choosing the right cascade classifier. For the purpose of the project, I decided to use haarcascade frontal default and right and left eye cascades. The following table shows the various cascades classifiers.

|  |  |
| --- | --- |
| **Type of cascade classifier** | **XML filename** |
| Face detector (default) | haarcascade\_frontalface\_default.xml |
| Face detector (fast Haar) | haarcascade\_frontalface\_alt2.xml |
| Face detector (fast LBP) | lbpcascade\_frontalface.xml |
| Profile (side-looking) face detector | haarcascade\_profileface.xml |
| Eye detector (separate for left and right) | haarcascade\_lefteye\_2splits.xml |
| Mouth detector | haarcascade\_mcs\_mouth.xml |
| Nose detector | haarcascade\_mcs\_nose.xml |
| Whole person detector | haarcascade\_fullbody.xml |

Detection process

After opening the camera, the image undergoes the following stages before face detection is complete:

* Grayscale color conversion – face detection only occurs on grayscale images and thus all images from the camera must first be converted to grayscale.
* Shrinking image – face detection speed depends on the size of input image frame. So the images are resized to smaller images for faster recognition.
* Histogram equalization – face recognition is not reliable in low-light condition thus there is need to improve the contrast and brightness.

## Face Preprocessing

Face recognition is vulnerable to lighting conditions, face orientation, expressions and poses. It is important to reduce such differences as much as possible. To do this, we need more complex functions such as facial feature detection and extraction like eyes, nose, mouth and eyebrows. For my project I decided to use eye detection as it is very useful for frontal images. At times face detector may recognize something else and claim it is a face. Thus it is important to use a second validation point, in this case the eyes. It is rare to fool the face detector and the eye detector at the same time.

After the face and both eyes are detected then the following processes are carried out:

* Geometric transformation and cropping – it comprises of scaling, rotating and translating the images so eyes are aligned. The background, forehead, chin and ears are removed.
* Histogram equalization for left and right sides – This process standardizes the brightness and contrast on both left and right eye independently.
* Smoothing – this process reduces image noise using bilateral filter.
* Elliptical mask – removes some remaining hair and background from the face image.

## Machine Learning – Training

This process involves collecting faces and training them. After training has been carried out you could then save the generated knowledge to a file or memory and use it later to recognize which person is seen in front of the camera. It is important that you provide a good training set that covers the types of variations you expect to occur in your testing set. For instance it is important to incorporate images of those people looking upwards and sideways too if the algorithm is to detect real-life conditions.

One way to obtain a good training set that will cover many different real -world conditions is for each person to rotate their head from looking left, to up, to right, to down then looking directly straight. Then the person tilts their head sideways and then up and down, while al so changing their facial expression, such as alternating between smiling, looking angry, and having a neutral face. If each person follows a routine such as this while collecting faces, then there is a much better chance of recognizing everyone in the real -world conditions. For even better results, it should be performed again with one or two more locations or directions, such as by turning the camera around by 180 degrees and walking in the opposite direction of the camera then repeating the whole routine, so that the training set would include many different lighting conditions.

The system is trained by machine-learning algorithms such as Eigenfaces, Artificial Neural Networks, Fisherfaces and many more. In openCV there are various implementations of such algorithms including Eigenfaces, Fisherfaces and Local Binary Pattern Histogram. Other implementations have been extended by openCV team and Philipp Wagner's “libfacerec” contribution. Both the Eigenfaces and Fisherfaces algorithms first calculates the average face that is a mathematical average of all the trained images, so they can subtract the average image from facial image to have better face recognition results.

## Face Recognition

It is a process of checking which of the collected people are the most similar to the face on the camera. OpenCV has a FaceRecognizer class used to identify the person in a photo simply by calling the FaceRecognizer::predict() function. We calculate the similarity of two faces by reconstructing the input face using a function getSimilarity(). Where a value is less than 0.3, it implies that the two images are very similar. For Eigenfaces there is one eigenvector for each face, so reconstruction tends to work well and therefore we can typically use a threshold of 0.5, but Fisherfaces has just one eigenvector for each person, so reconstruction will not work as well and thus set threshold as 0.7.

## Raspberry Pi

A raspberry pi is a credit-card-sized linux box developed by the Raspberry pi Foundation. It is meant to improve teaching of electronic and computer science but as usual computer enthusiast found a better purpose for it. The pi is a single-board computer where all peripherals such as keyboards and mouse have to be purchased separately. The pi currently operates on various distros of the linux operating system though it is possible to configure the android operating system in the pi since android is linux driven. It supports many programming languages.

Perhaps the most important part of the $25computer is the General Purpose Input Output(GPIO) pins. The GPIO is a generic pin on a chip whose behavior can be programmed (controlled) through software. The pi has 26 pins with GPIO voltage levels of 3.3 V and 5 V tolerant.



## Algorithms

There are a number of algorithms that will be used in the development of the facial recognition module. They include:

* Eigenfaces
* K-Nearest Neighbor
* Histogram equalizer

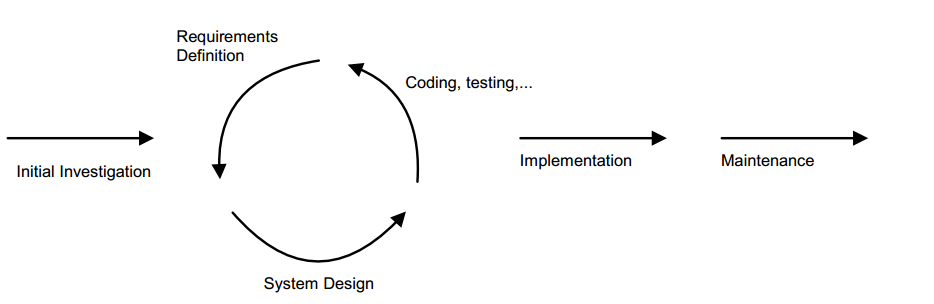
# METHODOLOGY

Introduction

The methodology details the rules and procedures followed during the development of the system. For this project, there was need for a methodology that allowed for numerous iterations since the project required various functions to be tried and tested before being implemented. The details of the methodology are discussed below in details.

System Development

The methodology of choice for this system is the Rapid application Development methodology. The reason behind the methodology is due to the risky attribute of this project. Various iterations have to be made before coming up with the final product. RAD attempts to reduce the inherent project risk by breaking the project into smaller segments and providing more ease-of-change during development process. It also aims at producing high quality systems quickly primarily through the use of iterative prototyping, active user involvement and computerized development tools. RAD also emphasizes on fulfilling a business need. Iteratively produces production software as opposed to a throwaway prototype. Standard systems analysis and design techniques can be fitted into this framework. For every module from the broken down project, prototyping methodology will be used.



Prototyping methodology

Advantages

1. The operational version of an application is available much earlier than with waterfall Incremental, or Spiral frameworks.
2. Concentrates on essential system elements from user viewpoint.
3. Provides the ability to rapidly change system design as demanded by users.
4. Produces a tighter fit between user requirement and system specification.
5. Generally produces a dramatic savings in time, money, and human effort.

Disadvantages

1. Danger of misalignment of developed system with the business due to missing information.
2. Project may end up with more requirements than needed (gold-planting).
3. Potential for violation of programming standards related to inconsistent naming conventions and inconsistent documentation.
4. Since some modules will be completed much earlier than others, well-defined interfaces are required.

Testing and Evaluation

3.3.1 Enrolment

This is where I will capture a sample of images. These images will undergo through a process of preprocessing and training until an average face is produced. Then the eigenvalues will represent the differences between the images in the database from the average face. The image set will be referenced against an identifier such as a numerical pin.

Entry.PIN🡨 PIN

Entry.Train 🡨 Capture(image)

3.3.2 Storage of Images

The images will be stored in an image database or a central repository. They will be stored with the images alongside their labels. Also the trained dataset will be stored so as to be called upon system startup.

3.3.3 Verification

This process will require a user to claim an identity by entering a PIN and the verifying it’s him/her. The results are either it is a match or not. The percentage of similarity will be indicated. Once a face is recognized then it is possible to grant access to the electronic door by manipulating the GPIO pins.

3.3.4 Performance Measures

False accepts – this is the likelihood that an imposter may be falsely accepted by the system.

False rejects – This is the likelihood that a valid or genuine user is rejected by the system.

Equal error rates - Somewhere between the two extremes is the equal error point where the two curves cross and which may represent a more realistic measure of performance.

Enrolment and verification times – The actual enrolment times will depend upon a number of variables inherent in the enrolment process.

Biometric Uniqueness –the degree of individuality or similarity within a user base will naturally affect performance to some degree.

## Resources

**3.4.1 Hardware Requirement**

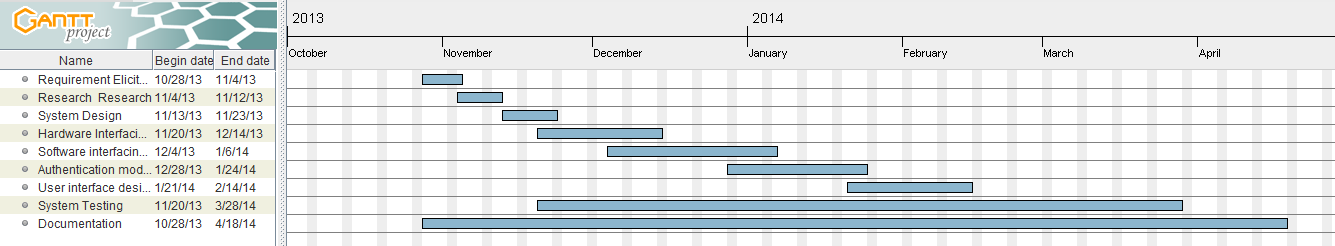
* A computer with at least 1GB ram, 2 GHz, 20 GB Hard disk
* A Raspberry Pi
* A breadboard
* Copper wire / jumper wires
* Camera / webcam
* Relay board
* Resistors, diodes and capacitors
* Smart phone

**3.4.2 Software Requirements**

* Eclipse / Visual C++ express
* Yales photo database
* MySQL – Database
* Linux / Raspbian – Operating system
* XAMPP server – Local server

## Project Activity Plan

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task No.** | **Task Name** | **Planned hours** | **Actual hours** | **Planned start date** | **Actual start date** | **Planned end date** | **Actual End date** | **Deliverables** |
| 1. | Requirement elicitation | 6 hrs |  | 28/10/13 |  | 04/11/13 |  | Formal Requirement Documentation |
| 2. | Research | 8 hrs |  | 04/11/13 |  | 12/11/13 |  | Formal Research Notes |
| 3. | System design | 15 hrs |  | 13/11/13 |  | 23/11/13 |  | System prototypes |
| 4. | Hardware interfacing module | 20 hrs |  | 20/11/13 |  | 14/12/13 |  | Complete digital circuit system |
| 5. | Software interfacing module | 25 hrs |  | 4/12/13 |  | 06/01/14 |  | Software application |
| 6. | Authentication module | 15 hrs |  | 28/12/13 |  | 24/01/14 |  | Facial recognition login application |
| 7. | User interface | 10 hrs |  | 21/01/14 |  | 14/02/14 |  | Graphical user interface |
| 8. | System Testing | 40 hrs |  | 20/11/13 |  | 28/03/14 |  | Working prototype / product |
| 9. | Documentation | 40 hrs |  | 20/11/13 |  | 18/04/14 |  | Formal Documentation report |



## Project Budget

|  |  |  |
| --- | --- | --- |
| No. | Product/ service | Cost (Ksh.) |
| 1. | Raspberry pi | 5000 |
| 2. | Digital Electronic kit | 3000 |
| 3. | Research | 1000 |
| 4. | Relay | 1000 |
|  | **Total** | **10000** |

# SYSTEM ANALYSIS AND DESIGN

## Feasibility Study

The implications, goals and scope of the project are investigated to determine the feasibility and applicability of the project.

* + 1. Technical feasibility

This is to determine if the project is technically possible. The project uses the face recognition algorithms that are proved to be 60% to 70% accurate. This is because unlike other biometric authentications, face recognition is affected by light, face orientation and poses.

In this project, in order to increase the accuracy of the face recognition, several libraries and classes such as histogram equalization, mirroring of photos for training and smoothening the images will be carried out.

* + 1. Financial Feasibility

The project uses algorithms that are open-source. The openCV computer vision program has thousands of algorithms and functions that can be used to make good face recognition applications for less cost. It also has a dedicated community that improves the code every now and then.

The hardware uses a raspberry pi which is a $25 computer and a relay board that is around $10. The project is therefore financially feasible.

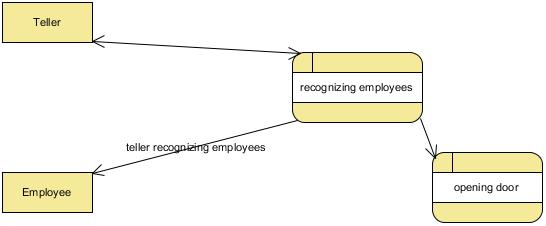
* + 1. Organizational Feasibility

The project can be incorporated in the current organization systems. The project uses existing electric doors. One need to attach the system to a power system and all is well. The system will work with existing circuit system in place since the circuits are separated by a relay circuit.

## Investigation of current system

The current system is where electric doors are controlled manually. For instance in the banks, the teller working closest to the electric door is the one who has to recognize the workers and press the button that grant the user access to the banking halls. This manual system can be quite exhausting to the teller or even inefficient if somehow the teller goes on break.

What needs to be reconstructed is the recognition and automatic granting of access to recognized faces. The system should have a backup system in case a genuine face was denied access.



* 1. Business system option

The goal of the system is to automate office entry. The automation extends only to the employees of the organizations or whoever the admin would like into the building. To do this the employees have to have their faces trained to the best of the system

The system can be distributed by having a central face databases and used in other offices. The safety of the database is quite guaranteed since the information is kept in a binary file system. The automation procedure provides various benefits over the cost. The system is able to record when a person enters the building and thus keeps a database of employees’ entry. This has benefit in security issues. The automation also benefits the teller since he/she will be able to concentrate on his/her work.

## Logical design

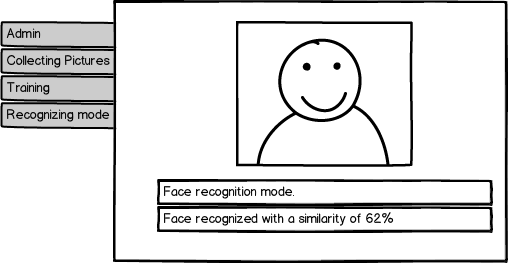
* + 1. Entity relationship diagram

This shows the logic between the database and the system.

## Physical design

* + 1. User Interface Design

This is concerned with how the users of the system input information and how the users receive the processed information from the system. In this case the users input the information from the camera. The video stream captures the images in the form of frames and trains the collected images. The administrator will handle capturing and training during startup.



* + 1. Data Design

This is concern with data representation and data storage. The data collected is in picture form. The face database is in conformity with Yale’s Face database.

./at/s1/1.pgm;0

./at/s1/2.pgm;0

./at/s1/3.pgm;0

…

./at/s2/1.pgm;0

./at/s2/2.pgm;0

./at/s2/3.pgm;0

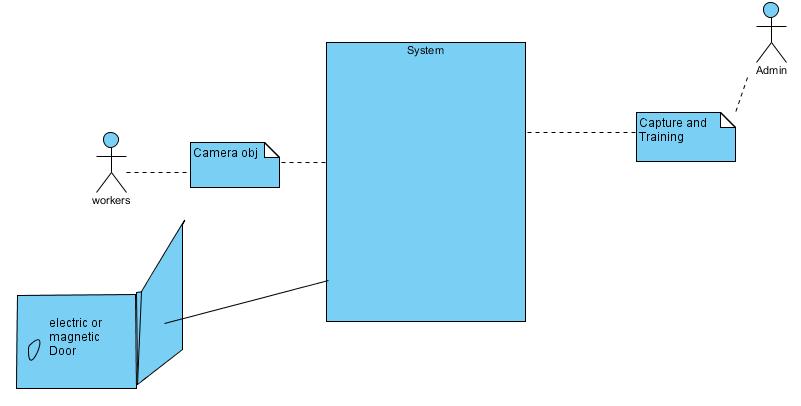
…

./at/s40/1.pgm;0

# SYSTEM DESIGN

## Introduction

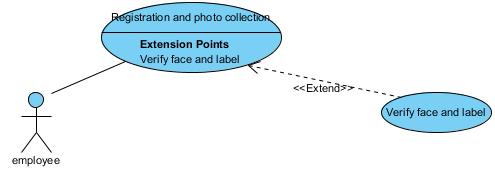
The design process indicates the stages that the project has undergone during the design stage. First is the System Architecture.



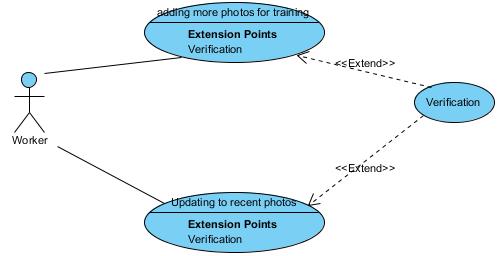
## System Modeling

* + 1. Registration and data collection

Use case diagram for the registration module as shown below. The admin may add users to the system database which learns and is able to recognize images.

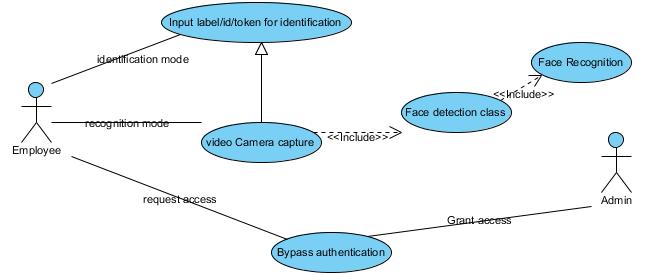


* + 1. Updating the system



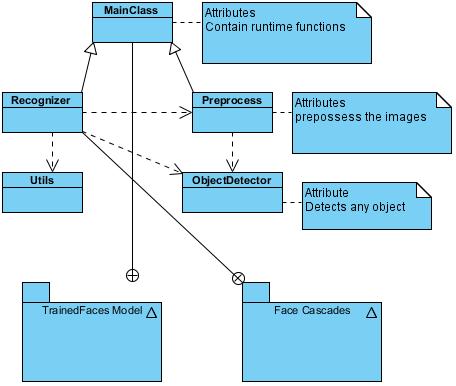
* + 1. Recognizing stage

This involves the actual face recognition.



* + 1. Class Diagram

Class diagram shows the relationship between various classes in the project. This includes their dependencies and the models involved.

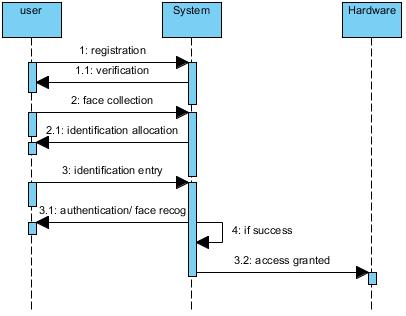


* + 1. Sequence Diagram

This shows the sequence of operations or activities in the system.

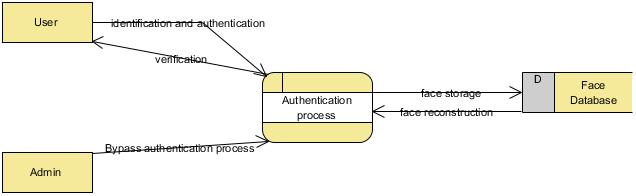
The sequence using the system includes:

1. User is registered into the system.
2. The system collects numerous photos of the user
3. The user is allocated an identification i.e. id/token
4. The user enters the identification
5. The system authenticate the user (face recognition)
6. The system grants access using the hardware.



## System Architecture

Shows the overall system architecture of the project



IMPLEMENTATION AND TESTING

6.1 Introduction

6.2 Implementation Overview

6.3 Testing Plan

6.4 Testing Strategy

6.5 Testing Methods

6.6 Testing Environment

CONCLUSION

7.1 Summary

7.2 Constraints and Challenges

The implementation period was quite challenging as I had to experiment with numerous face recognition algorithms and APIs.

Android FaceRecognizer class – this class could only detect the faces of the person onscreen and extract various features and store them inside an array. The features extracted were not enough to fully recognize a person.

Faceplusplus – This Company provided an API that one could use to detect faces and create datasets of people he wanted to recognize. The only problem with this API is that it had to get information online and thus proved inefficient for my system that required a local recognition solution.

Fisherfaces \_ this algorithm has been since 1997 and is inbuilt in openCV. However it requires at least two people in the database for it to start training. This therefore proved inefficient for a user who would like only one person to be recognized.

Libfacerec – this is an openCV extension created by Philipp Wagner. It provides additional features to the once openCV uses including setting the threshold on the faces being recognized.

7.3 Suggestions for further Works

APPENDICES

APPENDIX A: References

Kaehler, G. B. (septbember 2008). *Learning OpenCV.* Sebastopol, CA: O’Reilly Media, Inc.

Laganière, R. (2011). *OpenCV 2 Computer Vision Application Programming Cookbook.* Birmingham, UK: Packt Publishing Ltd.

APPENDIX B: User Manual

APPENDIX C: Sample Code