Middle East Technical University Northern Cyprus Campus
Computer Engineering Program

CNG491 Computer Engineering Design I

Alpsen Zeybek

Abdullah Küçük

Sabawun Afzal Khattak

Onur Can Güven

# Student Attendance System using Facial Recognition

Supervised by

Meryem Erbilek & Enver Ever

First Iteration

# 1. Introduction

## Three questions

How to detect a face from a photograph or a video feed?

What are the factors to consider when recognizing a face?

How to take attendance according to the recognized faces?

## 1.1. Motivation

The motivation of this project is to make the lives of students and professors easier by not making them waste any time or resources to record the attendance.

## 1.2. Aim & Objectives

We aim to implement automated attendance system using face recognition.

1.2.1 - Recognize the faces from a live video feed

1.2.2 - Process the class and the section information using time and date.

1.2.3 - Recognize and identify the face of the students to take attendance.

## 1.3. Methodology

1.3.1 - Implementing KNN (K-Nearest Neighbors Algorithm) and the public data sets that we found online suitable for this project via OpenCV Library using Python language.

1.3.2 - Testing the program using the test data of the public data set.

1.3.3 - Implementing a database including student photos, ID, Name and Surname.

1.3.4 - Testing the program on students to check if the program works correctly.

# 2. Requirements

## 2.1. Stakeholders

2.1.1 - Students

2.1.2 - Professors

2.1.3 - IT Administrators

## 2.2. Functional System Requirements

2.2.1 - Takes the attendance and records in an excel file.

2.2.2 - Supports recognizing students face with glasses and accessories.

2.2.3 - Supports printing the generated excel file.

2.2.4 - Tracks the time to find the correct lecture and the students registered to that section.

## 2.3. Non-functional Requirements

2.3.1 - Camera angle and quality should be sufficient.

2.3.2 - Recording the attendance shouldn’t take more than 10 seconds.

2.3.3 - It should store at least 5 different photos of each student.

## 2.4. Domain Requirements – if any.

2.4.1 - The live video feed will not be recorded, and cameras shut down after class hours.

## 2.5. Assumptions and Justifications

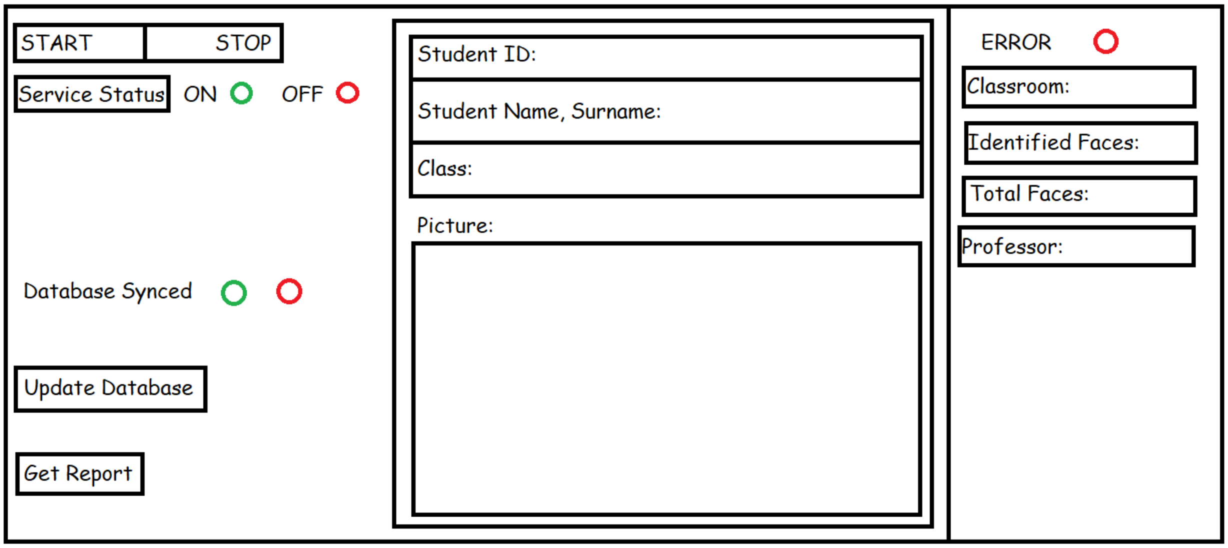
2.5.1 - Assuming the dedicated class room is always used during the designated timeslots.

2.5.2 - Assuming the students’ faces are facing the camera while the attendance recording is ongoing.

2.5.3 - Assuming the program always has enough storage for storing students’ photos.

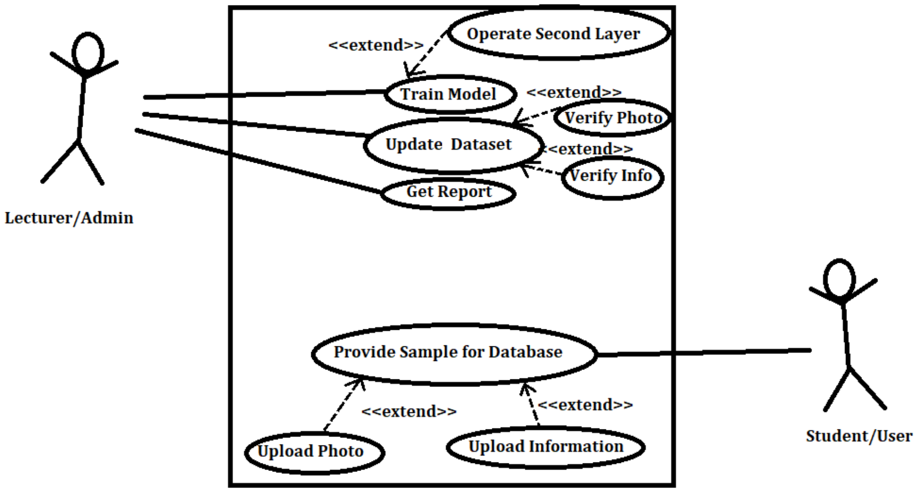
## 3. Graphical User Interface

Since our project is not necessarily need a complicated GUI, we are just providing a simple example.

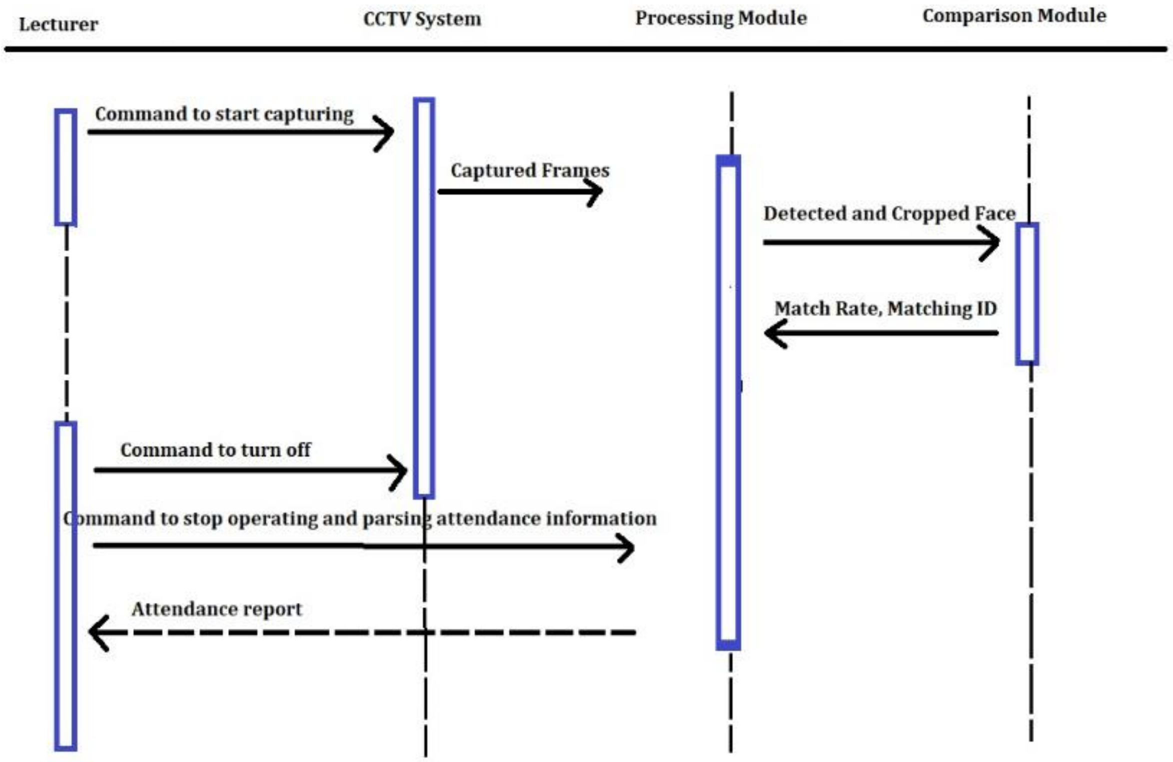


# 4. System Modelling

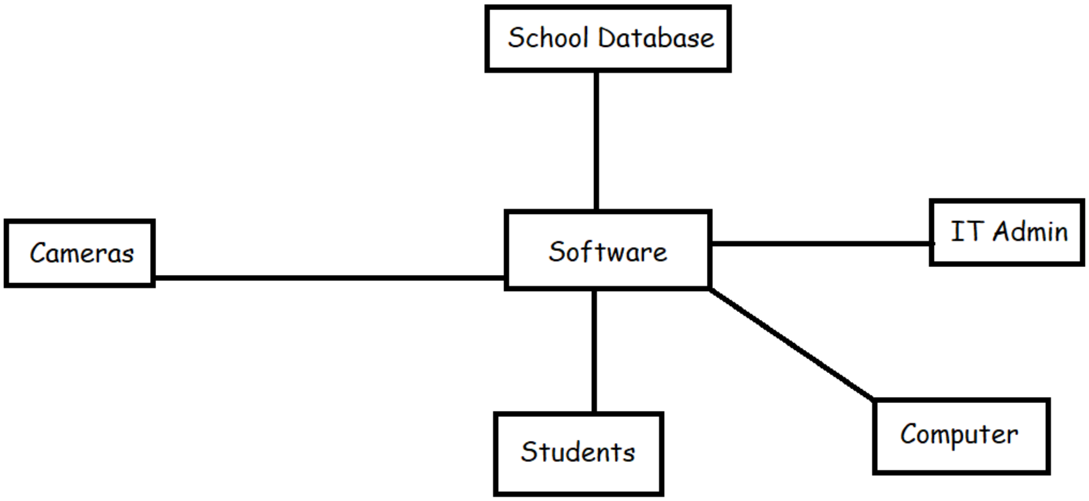
## 4.1. Structured Use Case Diagram



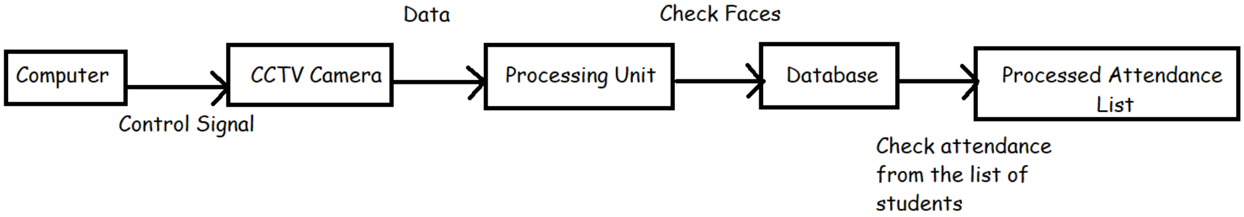
## 4.2. Sequence Diagrams of the Major Use Cases



## 4.3. Context Model



## 4.4. Architectural Model



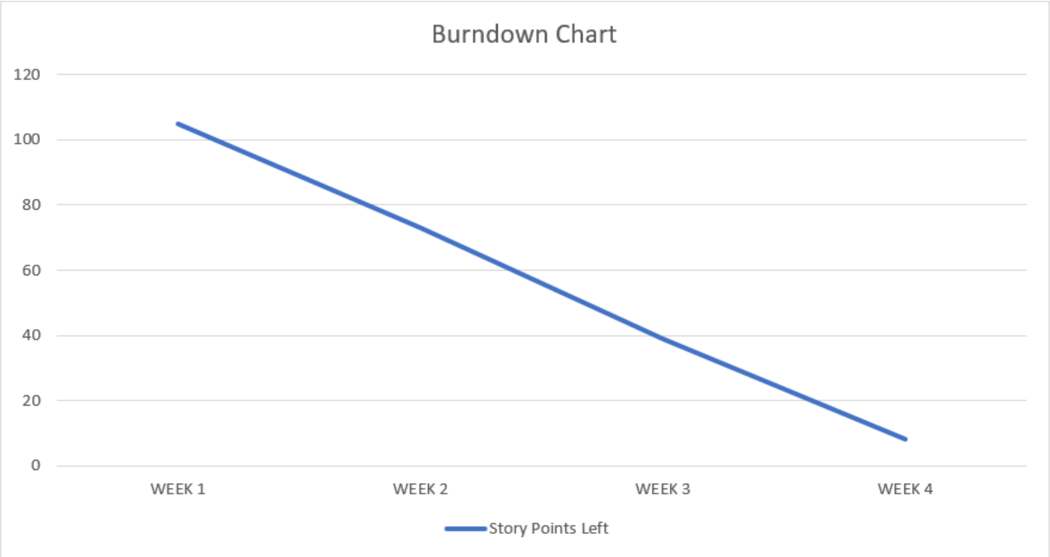
## 4.5. Process Model

# 5. Agile Development with Scrum

## 5.1. Sprint Backlog

| | Member | Estimated Hours | | | |
|---|---|---|---|---|---|
| | | W1 | W2 | W3 | W4 |
| Reading Related Work | All | 2 | 2 | 1 | 0 |
| Image Processing Alg. Research | All | 3 | 3 | 3 | 0 |
| Datasets Research | Abdullah | 3 | 3 | 2 | 0 |
| Face Detection Met. Research | Alpsen | 3 | 3 | 2 | 2 |
| Face Recognition Met. Research | Sabawun | 4 | 4 | 4 | 2 |
| Introduction | Onur | 2 | 2 | 2 | 0 |
| Related Work | Onur | 2 | 2 | 2 | 0 |
| Requirements | Onur | 0 | 2 | 2 | 2 |
| System Modeling | Alpsen | 5 | 5 | 5 | 0 |
| Graphical User Interface | Sabawun | 2 | 2 | 2 | 0 |
| Agile Development with Scrum | Onur | 2 | 2 | 2 | 0 |
| Project Estimation | Abdullah | 4 | 4 | 4 | 2 |

## 5.2. Sprint Burndown Chart



## 5.3. Sprint Review

We started by reading and researching related topics. The purpose of this is to understand the concept of face recognition and the algorithms that is related to it.

In week two, we started by talking about the First Iteration of Report. We started to talk about it before it is released to give us more time to research. Then we made work sharing between us. Everyone started searching and preparing their parts so that we can discuss it after First Iteration of Report is released.

In week four, we started to arrange our work for group discussion. Then we checked our work together and made necessary corrections and additions. After that, we prepared our report and submitted it for the feedback session on week 5.

To conclude, we didn’t face any problems while doing our own researches and preparations. We encountered some schedule crashes, but other than that it was going as planned. We finished our first iteration before the time. However, we are not fully satisfied yet, so waiting feedback session to improve our report.

## 5.4. Sprint Retrospective

Because we were learning and executing the assignment at the same time, several of our tasks took longer than they should have. It would take less time if we know the subjects. In certain cases, we might want to start with an outline. Including the communication issues we had, we estimate a ten -hour loss in overall time.

# 6. Project Estimation

## 6.1 UTFP

**Inputs**

(H) Video feed from camera for face detection & recognition

(H) Schedules of the lectures

**Outputs**

(H) Attendance Information

(H) Student Info

(L) Error Message

(L) Unwanted Student

**Inquiries**

(H) Face recognition operation

(H) Attendance marking operation

**Logical Internal Files**

(L) Student Information {Name, Surname, Student ID, Unique ID}

(H) Student Photos

(H) Trained model of the class

**External Interface File**

(H) .csv file

(H) KNN Algorithm

Inputs Total = 2 x 6 = 12

Outputs Total = 2 x 7 + 2 x 4 = 22

Inquiries Total = 2 x 6 = 12

Logical Internal Files = 2 x 15 + 1 x 7 = 37

External Interface Files = 2 x 10 = 20

**Total UFTP** = 103

## 7.2 VAF

| General System Characteristics | Degree of Influence |
|---|---|
| Data communications | 3 |
| Distributed data Processing | 3 |
| Performance | 2 |
| Heavily used configuration | 5 |
| Transaction rates | 1 |
| On line data entry | 3 |
| End-user efficiency | 4 |
| Online updates | 3 |
| Complex processing | 5 |
| Reusability | 5 |
| Installation ease | 0 |
| Operational ease | 3 |
| Multiple sites | 0 |
| Facilitate change | 4 |

**Total** = TDI = 41

**VAF** = TDI * 0.01 + 0.65 = 1.06

## 6.3 ATFP

**ATFP** = UTFP * VAF = 103 * 1.06 = 109.18

## 6.4 Estimated LOC

**LOC** = ATFP * Language Unit Size = 109.18 x 20 = 2183.6 (Using Python)

## 6.5 COCOMO

Development Mode is Organic, a = 2.4, b= 1.05, c = 0.38

**KDSI** = ATFP * Language Unit Size / 1000 = 2.1836

**MM** = a * (KDSI ^ b) = 5.44

**TDEV** = 2.5 * (5.44) ^ c = 4.76

## 6.6 Jones's First Order Schedule Estimation

For the average,

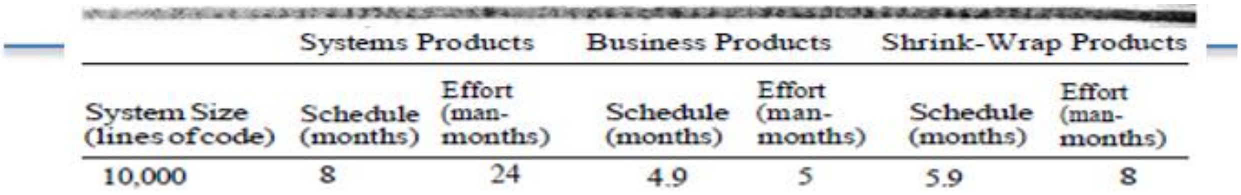
Class Exponent = 0.45 (System Software)

**Estimate Effort (Man-Month)** = ((ATFP) ^ (3*Class Exponent)/27 = 20.90

**Rough Schedule Estimate** = (ATFP)^class exponent = 8.26

**Schedule in months** = 3*man-months^1/3 = 5.28 months

## 6.7 Shortest Possible Schedules



| | Systems Products | | Business Products | | Shrink-Wrap Products | |
|---|---|---|---|---|---|---|
| System Size (lines of code) | Schedule (months) | Effort (man-months) | Schedule (months) | Effort (man-months) | Schedule (months) | Effort (man-months) |
| 10,000 | 8 | 24 | 4.9 | 5 | 5.9 | 8 |

## 6.8 Estimated Team Size

Estimated Team Size = estimated schedule / estimated effort = MM / TDEV = 1.14

Estimated Team Size = 1.14