TEST REPORT

for

FACIAL ATTENDANCE SYSTEM

**Version 1.0 approved**

**Prepared by Team Alpha**

**Team Alpha**

**20th September 2019**

**Table of Contents**

**Table of Contents [ii](#_gjdgxs)**

**Revision History [ii](#_30j0zll)**

1. **Introduction [1](#_1fob9te)**
   1. Purpose [1](#_3znysh7)
   2. Document Conventions [1](#_2et92p0)
   3. Intended Audience and Reading Suggestions [1](#_tyjcwt)
   4. Product Scope 1
2. **Overall Description [2](#_4d34og8)**
   1. Product Perspective [2](#_2s8eyo1)
   2. Product Functions [2](#_17dp8vu)
   3. User Classes and Characteristics [3](#_3rdcrjn)
   4. Operating Environment [4](#_26in1rg)
   5. Design and Implementation Constraints [4](#_lnxbz9)
   6. User Documentation [4](#_35nkun2)
   7. Assumptions and Dependencies [4](#_1ksv4uv)
3. **External Interface Requirements [4](#_44sinio)**
   1. User Interfaces [4](#_2jxsxqh)
   2. Hardware Interfaces [4](#_z337ya)
   3. Software Interfaces [5](#_3j2qqm3)
   4. Communications Interfaces [5](#_1y810tw)
4. **System Features [5](#_4i7ojhp)**
   1. System Feature 5
5. **Other Nonfunctional Requirements [6](#_3whwml4)**
   1. Performance Requirements [6](#_2bn6wsx)
   2. Safety Requirements [6](#_qsh70q)
   3. Security Requirements 6
   4. Business Rules [6](#_49x2ik5)
6. **Other Requirements [6](#_2p2csry)**

**Appendix A: Glossary [7](#_147n2zr)**

**Appendix B: Analysis Models [7](#_3o7alnk)**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Team Alpha:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Registration Number** | **Roll No.** | **Sem & Branch** | **Section** |
| Sampurna | 170911174 | 41 | IT (V) | B |
| Shambhavi Krishna | 170911240 | 55 | IT (V) | B |
| Abhinav Agarwal | 170911238 | 54 | IT (V) | B |
| Prashant | 170911230 | 52 | IT (V) | B |

# Introduction

## Purpose

*This SRS is to provide documentation for the team project created by team Alpha.*

*This project involves the creation of a user friendly application which may ease the pen and paper attendance system currently being employed in our college. This is the primary release of our software and still is in beta testing phase. Users are requested to provide any insights or possible bugs that they might encounter in the duration they use the software. Above all we intend to provide a user friendly application for the users to utilise.*

## Document Conventions

*We intend to use standard documentation conventions while specifying the scope of this SRS documentation and will also be adhering to generic coding conventions and practices while creating the required product. For example we have used FAS as a convention for “Facial Attendance System” throughout the software specification.*

## Intended Audience and Reading Suggestions

*The SRS can enable fellow students and faculty to get rid of the manual and unreliable task of taking attendance every class. For the Students they don't need to rely on their teachers mood for their day to day attendance and is of no problem if they missed their roll call by any chance. The teachers can save their valuable time that they may waste on taking attendance in every class of the semester.sss*

## Product Scope

*This software focuses on facial recognition using a given dataset which is used as a training set , using the training set the algorithm acts as a classifier which can segregate faces based on the input provided by the camera or given input device. It will then classify the given image and then mark the attendance for the classified face.*

# Overall Description

## Product Perspective

*FAS is a highly in demand feature of software. Many companies for eg- Apple and any major smartphone brand strive for better accuracy in their facial detecting system as it is one of the features that has completely replaced the fingerprint login system of smartphones. Many companies have several patents pertaining to this particular feature where they have found out various methods to improve accuracy in detection.*

## Product Functions

*The software outside the core product provide the user with functionalities like:*

*1. Storing their name and details in the database.*

*2. Making their attendance in real time with timestamp.*

*3. Recognising the faces with certain accuracy.*

*In the core product, the user is provided with some basic functions like:*

*1. Enrolling a student to the database.*

*2. Removing a student from the database.*

*3. If the current user is Admin, they can add or remove students from any database.*

## User Classes and Characteristics

*The Classes that we might find essential for this project are listed as follows(in descending priority):*

1. *Recognition System: This consists of the main python scripts that correctly classifies the images and maps them to their respective tuples in the DB and the camera which captures the image to be processed and cleans it of any background noise or unnecessary features.*
2. *Website: All the features are accessible to the user via an interface of a website that is being implemented with the help of Javascript, HTML and CSS.*

## Operating Environment

*The provided software supports platform independency and can support any device capable of running on any web browser that has support for a webcam and is able to run particular python scripts with database connectivity to MongoDB.*

## Design and Implementation Constraints

*Devices running a version of their OS below the versions specified might have trouble or not be able to run the specified software:-*

*Windows: Windows 95 or higher*

*MAC OS: OS 8.1 or higher*

*Linux: Ubuntu Linux or Any Linux running Debian*

*The Device running the software must have the minimum hardware requirements:*

*RAM: 512 MB or higher*

*GPU: Intel HD 430 or equivalent with 256MB or higher VRAM*

*Directx: Not required*

*Camera: Any functioning camera with a resolution > 4MP*

*Storage: 24MB or higher*

## User Documentation

*No specific documentation is required for the product as it is not proprietary and can be treated as an open source project with an MIT license. The website is quite straightforward and can be learned to operate without any prior knowledge as it contains all the instructions for operating the software.*

## Assumptions and Dependencies

*All the components used in developing this particular software are original but they have been referenced from various online websites that have been listed in the references. One of the major contributors to this project is the repositories uploaded by various people on Github. All the classes required for the project have been developed and created by the members of this project.*

# External Interface Requirements

## User Interfaces

*The user interface is quite minimalistic and user friendly. The main menu for the software has options to add or remove student records from the database and a window that displays what the camera is capturing at the current time. The capture button when pressed would then display the picture taken by the camera and send it to the algorithm for further classification. The UI is pretty simple and easy to grasp, whereas there is not much focus on the UX as the website does not contain any subpages which can be optimised for user experience.*

## Hardware Interfaces

*Since the software is a FAS, the only hardware interface required is a working webcam that can take images in grayscale and color.*

## Software Interfaces

*As the user opens the website, they can clearly notice the options present on the index page there are options to register a new student in the database and to capture a new image for which the camera is activated and is displaying whatever information it is capturing. On pressing the capture button the camera sends the taken snapshot to the algorithm for further classification and processing.  
 Also any crash dumps and memory dumps that might be created due to faulty execution or severe bugs will be handled by the software and the storage facility that is being used by the component device.*

## Communications Interfaces

*The software communicates to the user through the website to the user whereas for the backend we have implemented python scripts along with the usage of MongoDB which helps us store and classify our data. This can be useful as the changes made are highlighted everywhere in the DB and are in general real time.*

# System Features

*Some of the features that we plan of implementing on this software are as follows:-*

## Saving Backups Offline

4.1.1 **Description and Priority**

*This feature of saving backups of user data is quite high on the list of priorities as if the website loses connectivity to the database due to some technical error the application can store the data on the system drive rather than waiting for the problem to be solved.*

4.1.2 **Functional Requirements**

*There are not many functional requirements but some of the details are necessary for the program to store the save files.*

REQ-1: Enough memory in the storage device to save the save files.

REQ-2: The processing power of the CPU must be high enough for the program to execute tasks simultaneously.

# Other Nonfunctional Requirements

## Performance Requirements

*Need the system requirements as stated above in the Design And Implementation tab. Other than that*

## Safety Requirements

*Crash and Memory dumps in case of a system crash and unexpected errors the software might result in occupying more space than the specified amount as it is needed to prevent the errors that might be caused by the same errors in the future or to notify the devs whether or not he same recurring problem can be fixed by them. Although the extra memory requirements will be specified along with the updates provided.*

## Security Requirements

*Plagiarism and selling counterfeit copies of the product which are not issues by the devs and have not been authorised by the marketing committee might result in serious consequences. The seller and the selle of these copies will face serious charges if they play a role in this unfair business of counterfeit.*

*They might even have to pay enormous amounts of money if caught doing the same. The benefit of the doubt might be given to the customer but there will always be a detailed enquire about these matters.*

*We do support devs who are willing to pay a small amount of fee in order to get hold of the source code of out product to develop similar or better product for educational or research purposes but our policies restrict them from copying full classes of out core code.*

## Business Rules

*Under fair use policy we do encourage other people to use our source code for a nominal fee and we might also support them in their venture if their idea appeals to us in any way, financial or PR. For further details contact the developers and ask for the detailed rules of our business.*

*We do sell copies of our product to customers who are limited to two per customer and for other companies of devs we charge a small fee.*

# Other Requirements

*All the requirements for this software have been stated prior to this section of the SRS. All the necessary documentation is also provided in the prior sections.*

**Appendix A: Glossary**

*The following have been used to provide an ease in reading this SRS:-*

*DB: Database*

*FAS: Facial Attendance System*

*Req: Requirements*

*RAM : Random Access Memory*

*CPU: Central Processing Unit*

**Appendix B: Analysis Models**