



SOFTWARE REQUIREMENTS SPECIFICATION

For

Project no. 7

Classroom Visualization App1

Team no. 16

Ansh Sood - 160101012

Kanika Agarwal - 160101038

Shimona Verma - 160101065

Table of Contents

1. Introduction.....	04
1.1 Purpose.....	04
1.2 Scope.....	04
1.3 Document Conventions.....	05
1.4 Intended Audience and Reading Suggestions.....	05
1.5 References	05
2. Overall Description.....	06
2.1 Product Perspective.....	06
2.2 Product Functions.....	06
2.3 User Characteristics.....	06
2.4 Design and Implementation Constraints.....	07
2.5 Assumptions and Dependencies.....	07
3. External Interface Requirements.....	08
3.1 User Interfaces.....	08
3.2 Hardware Interfaces.....	08
3.3 Software Interfaces.....	08
4. Functional Requirements.....	09
4.1 Sign-up.....	09
4.2 Login.....	09

4.3 Student Database Creator.....	09
4.3.1 Add Student Details.....	10
4.3.2 Update Student Details.....	10
4.3.3 Remove Student Details.....	10
4.3.4 State Allocator.....	10
4.4 Frames Extractor.....	11
4.5 Face Detection.....	11
4.6 Face Recognition.....	11
4.6.1 Feature Extraction.....	11
4.6.2 Data Comparing.....	12
4.7 State Mapping.....	12
4.8 Output Generator.....	12
5. Non-Functional Requirements.....	13
5.1 Performance Requirements.....	13
5.2 Functionality Requirements.....	13
5.3 Software Quality Attributes.....	13
5.3.1 Availability.....	13
5.3.2 Security.....	14
5.3.3 Maintainability.....	14
5.3.4 Portability.....	14

6. User Study.....14

7. Appendix.....15

1. INTRODUCTION

This SRS aims at developing a Android App for the general use of teachers. This app recognises student faces and displays a color coded bounded box around their faces in real time. The color of the box is determined by the state value (ranging from 1 to 10) corresponding to each student given in the database.

1.1 Purpose

In this document, we describe the software requirements of a mobile application named “Classroom Visualisation App”. The purpose of this document is to give detailed description of the requirements for the proposed app. It will illustrate complete declaration regarding development of the software, keeping in consideration the system constraints, user interface, and interaction with other external applications. This SRS describes the entire application including the front end graphical user interface and the back end of the product.

1.2 Product Scope

The purpose of the product is to create convenient and easy-to-use android app for users. This app will detect and recognize faces of students from their video being taken from device’s camera and thus displaying color coded boxes around their faces. This app can also be used to take attendance. The system is based on image processing and machine learning. Above all, we hope to provide a comfortable user experience along with best results.

1.3 Document Conventions

Main Section Titles

- Font: Arial
- Face: Bold
- Size: 24

Sub-Section Titles

- Font: Arial
- Face: Bold
- Size: 23

Other Text Explanations

- Font: Arial
- Face: Normal
- Size: 14

1.4 Intended Audience and Reading Suggestions

The intended audience is supposed to be the teachers who will be using this app during their lectures to recognize the students' state determined by the color coded boxes around their faces.

1.5 References

1. IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
2. <https://www.superdatascience.com/opencv-face-recognition/>
3. <http://digital.cs.usu.edu/~xqi/Teaching/REU05/Notes/FaceRecognition.pdf>

2. OVERALL DESCRIPTION

2.1 Product Perspective

This product is a part of a classroom teaching app which uses Android Application to let the professors know about the state of their students during lectures.

2.2 Product Functions

Functions included in the final product will be as follows :-

1. Sign-up
2. Login
3. Student Database Creator
4. Frames Extractor
5. Face Detection
6. Face Recognition
7. State Mapping
8. Output Generator

2.3 User Characteristics

The intended users for the product will have the following characteristics :-

1. Should possess Android device with the specified API level.
2. Able to understand the functioning and operation of the software on a basic level.
3. Should have basic understanding of English Language to operate the application.

2.4 Design and Implementation Constraints

The main constraint we faced is distinguishing between identical twins.

2.5 Assumptions and Dependencies

Assumptions :-

1. OS used on the device :- Android 6.0 or higher / API level 23 or higher.
2. Device must have a provision of a rear camera with minimum of 5.0 MP capacity.
3. Device must have a minimum of 150 MB storage space for storing the application and also some minimal data generated whilst operation.

Dependencies :-

1. Android Camera API with basic camera operation capabilities
2. Must have ARM architecture based hardware design.

It depends on accuracy of OpenCV libraries in facial recognition which is the main building block of result evaluation. Android and OpenCV are open source softwares leading our app to be free of cost. Performance of app will depend on android version and hardware architecture.

3. External Interface Requirements

3.1 User Interfaces

The application will have a user friendly interface for the teachers. It will have signup, login and help options.

After log in, a camera view will be present along with student database creator options on one side. This will include Add Student Details, Update Student Details and Remove Student.

3.2 Hardware Interfaces

The application can be run on any Android device with the following minimum requirement -

1. Rear Camera: 5.0 Mega Pixels (MP).
2. CPU: Dual Core 1.2 GHz Krait .
3. RAM: At Least 200 Megabytes (MB).
4. Free Internal Memory : 150 Megabytes (MB).

3.3 Software Interfaces

The application requires Android 6.0 (Marshmallow) or above Android version to operate. The application can run without an internet connection.

4. Functional Requirements

This section will describe in detail all the features in the previous section. The visualization app takes a real time camera image view as an input, and returns colour coded bounded box around the faces.

4.1 Sign-up

- **Input**: User Information.
- **Output**: Home page.
- **Short Description**: This function will register a new user to the application.

4.2 Login

- **Input**: User Credentials.
- **Output**: Home Panel.
- **Short Description**: This function will take user credentials as input and after verification, direct the user to home screen. This will give authorization to edit the student database.

4.3 Student Database Creator

- **Input**: Student details.
- **Output**: Pop up message confirming the action performed.
- **Short Description**: This function creates the database of the students. It also can update or delete a student's details.

4.3.1 Add Student Details

- **Input**: Student Details.
- **Output**: Pop up message with updated database.
- **Short Description**: This function saves student details along with his images (which acts as the training set for face recognition) to the application's database.

4.3.2 Update Student Details

- **Input**: Student name.
- **Output**: Pop up message with updated database.
- **Short Description**: This function updates the details of student whose name is given as input.

4.3.3 Remove Student

- **Input**: Student name.
- **Output**: Pop up message with updated database.
- **Short Description**: This function deletes all the data from the database related to the input student name.

4.3.4 State Allocator

- **Input**: Student Name
- **Output**: Random integer ranging between 1 - 10.
- **Short Description**: This function generates and maps a random integer between 1-10 to the student whose name is given as input. This integer corresponds to the state of the student. The state value of each student changes after every 10 minutes.

4.4 Frames Extractor

- **Input**: real time camera image view.
- **Output**: Image frames.
- **Short Description**: In this function, the real time image view will be taken from the rear camera. The function would then extract image frames from the real time camera image view.

4.5 Face Detection

- **Input**: Image frames.
- **Output**: Location of detected faces.
- **Short Description**: This function will locate all the faces present in the image preview frame and return its coordinates.

4.6 Face recognition

- **Input**: Location of detected faces.
- **Output**: Names of all the detected faces.
- **Short Description**: This function will recognise the names of all the identified faces from the database using the unique features of the detected faces.

4.6.1 Feature Extraction

- **Input**: Location of detected faces.
- **Output**: Features extracted from detected faces.
- **Short Description**: Extract unique characteristics of students' face that are used to differentiate amongst them, like eyes, mouth, nose, etc.

4.6.2 Data Comparing

- **Input**: Features extracted from detected faces.
- **Output**: Names of all the detected faces.
- **Short Description**: Despite variations in light or expression, it will recognize those features out of all the features that are unique to the faces. Then it will compare these features with the given database to determine the name of the students in the image frame.

4.7 State Mapping

- **Input**: Names of all the detected faces.
- **Output**: States corresponding to each recognised face.
- **Short Description**: This function will look up the database with respect to the names provided as input and give the corresponding state values.

4.8 Output Generator

- **Input**: States corresponding to each recognised face.
- **Output**: Color coded boxes around recognised faces.
- **Short Description**: This function displays color coded boxes around all the recognised faces in real time image preview .

5. Non-Functional Requirements

5.1 Performance Requirements

1. The product shall take initial load time depending on OpenCV libraries to load.
2. The performance of the product shall depend on the quality of the video captured .
3. The performance of the product shall also depend on the hardware components of the user's device .
4. Video bitrate maybe high and OpenCV might take high CPU usage and hence app would run smoothly on high end phones.

5.2 Functionality Requirements

1. The user shall capture video in proper lighting conditions.
2. The face of the user should be in proper orientation and only limited no. of faces should be there in focus.
3. The product shall not be pushed to background service in user's device. It should remain in foreground to predict correct results.

5.3 Software Quality Attributes

5.3.1 Availability

The application will be available for use round the clock. The system shall allow users to restart the application in case the application hangs or crashes.

5.3.2 Security

The application will use a login system for authentication and thus will be highly secure. It will prevent any type of unauthorized access to private content.

5.3.3 Maintainability

The system will be updatable from software patches available through the Google Play Store. Any discrepancies will be addressable by any developer as the coding will be done according to the coding standards of IEEE.

5.3.4 Portability

The application will be easily transferable to any Android device satisfying the minimum software dependency requirements as specified in this SRS Document. The application can be installed on an Android using the same method as any other Android App via the Android App Manager.

6. User Study

A survey was conducted among our friends where we asked the following questions to them and their responses were as follows-

Q.1: Would you like to use this app for taking attendance instead of signature based attendance?

Result: Yes: 69%

No: 31%

Q.2 :How frequent would you like to use this application to determine attentiveness of students in class?

Result: 2min-5min :0 %

5min-10min: 11%

10min-:20min: 64%

20min-30min: 25%

Q.3 : Instead of generating random states would you like this app to properly predict mental states of students?

Result: Yes: 62%

No: 38%

(Survey done on 21 people)

7.Appendix: Glossary

1. SRS - System Requirements Specification
2. OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision.
3. Android (stylized as android) is a mobile operating system developed by Google, based on the Linux kernel and

designed primarily for touchscreen mobile devices such as smartphones and tablets.

4. API : Application Programming Interface

- API level : A measure of the version of Android device being used.
- API levels 23 ~ Android 6.0 (Marshmallow)