
Software Requirements Specification

for

Face Recognition Attendance System

Prepared by

Kata Nithin
Lakkakula Bindu
Kommarajula Praveen

227Z1A0576
227Z1A0591
227Z1A0582

nithinkata29@gmail.com
lakkakulabindu2584@gmail.com
praveenkommarajula8@gmail.com

Instructor: Mr. S. Srikanth Reddy (Assistant Professor)

Contents

CONTENTS	II
1 INTRODUCTION	1
1.1 DOCUMENT PURPOSE	ERROR! BOOKMARK NOT DEFINED.
1.2 PRODUCT SCOPE.....	ERROR! BOOKMARK NOT DEFINED.
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	ERROR! BOOKMARK NOT DEFINED.
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	ERROR! BOOKMARK NOT DEFINED.
1.5 DOCUMENT CONVENTIONS	2
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	ERROR! BOOKMARK NOT DEFINED.
2 OVERALL DESCRIPTION	2
2.1 PRODUCT OVERVIEW	2
2.2 PRODUCT FUNCTIONALITY	ERROR! BOOKMARK NOT DEFINED.
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS.....	ERROR! BOOKMARK NOT DEFINED.
2.4 ASSUMPTIONS AND DEPENDENCIES	ERROR! BOOKMARK NOT DEFINED.
3 SPECIFIC REQUIREMENTS	4
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	ERROR! BOOKMARK NOT DEFINED.
3.2 FUNCTIONAL REQUIREMENTS	ERROR! BOOKMARK NOT DEFINED.
4 OTHER NON-FUNCTIONAL REQUIREMENTS	4
4.1 PERFORMANCE REQUIREMENTS	4
4.2 SAFETY AND SECURITY REQUIREMENTS	5
4.3 SOFTWARE QUALITY ATTRIBUTES	5
5 OTHER REQUIREMENTS	5

1 Introduction

1.1 Document Purpose

This document describes the software requirements for the **Face Recognition Attendance System**. The purpose of this system is to automatically mark attendance based on facial recognition technology. The system will use Python libraries such as OpenCV for real-time face detection, tkinter for the user interface, and Pandas for attendance tracking. The system aims to enhance attendance recording by making it more accurate and efficient through automation.

1.2 Product Scope

The system will use facial recognition to automatically identify individuals and mark their attendance. OpenCV will be used to capture and process real-time video feeds, while pre-trained face recognition models will be employed to match faces. The attendance data will be stored and managed using Pandas. The system will also include a tkinter-based user interface for both administrators and users to view and manage attendance records. Applications include educational institutions, corporate environments, and event management.

1.3 Intended Audience and Document Overview

This document is intended for:

- Developers implementing the system
- Project managers overseeing the development
- End-users (e.g., staff members, students) who will interact with the system

The document provides an overview of the system, functional and non-functional requirements, and technical constraints.

1.4 Definitions, Acronyms, and Abbreviations

- **OpenCV**: Open Source Computer Vision Library
- **Tkinter**: Python library for creating graphical user interfaces
- **Pandas**: Data analysis and manipulation library in Python

- **Face Recognition:** Process of identifying or verifying an individual based on their face
- **Attendance:** Record of presence/absence for individuals in a session

1.5 Document Conventions

- Font: TimesNewRoman, size 12
- Sections follow IEEE SRS formatting
- Diagrams and figures included where necessary

1.6 References and Acknowledgments

- OpenCV documentation
 - Python libraries: tkinter, numpy, pandas, datetime
 - Research papers on face recognition technology
 - Tutorials and guides on facial recognition systems in Python
-

2 Overall Description

2.1 Product Overview

The Face Recognition Attendance System captures video input from a webcam or camera, detects faces, and compares them to a pre-built database of registered faces. The system marks attendance based on matching faces and records the date and time. The system's user interface, built with Tkinter, allows users to view attendance logs, manage enrolled faces, and view real-time attendance status.

2.2 Product Functionality

- Accept input from a webcam or camera
- Pre-process captured images (face detection, alignment, etc.)
- Use facial recognition to match detected faces with registered users
- Log attendance with date and time for each recognized face
- Provide a Tkinter-based interface to view and manage attendance records

2.3 Design and Implementation Constraints

- Must use Python libraries such as OpenCV for face detection and recognition, Tkinter for the UI, NumPy for data handling, and Pandas for attendance records
- Requires a webcam or camera for capturing real-time images
- Needs sufficient memory and CPU/GPU resources for processing face recognition in real-time
- Attendance records must be stored in CSV or Excel format via Pandas

2.4 Assumptions and Dependencies

- Pre-trained face recognition models are available (e.g., from OpenCV or dlib)
- The system will be used in an environment with proper lighting and stable camera input
- Users are registered with their facial data beforehand

3 Specific Requirements

3.1 External Interface Requirements

- **User Interface:** A Tkinter-based dashboard will allow administrators to manage users, view attendance, and configure system settings.
- **Hardware Interfaces:** Requires a webcam or camera to capture images.
- **Software Interfaces:** Integrates with OpenCV for face detection, NumPy for matrix operations, Pandas for storing attendance logs, and DateTime for logging time.

3.2 Functional Requirements

- **F1:** The system shall allow users to register their face images for future recognition.
- **F2:** The system shall detect faces in real-time from the webcam feed.
- **F3:** The system shall match detected faces with registered user data.
- **F4:** The system shall mark attendance by recording the user's name and time of entry.
- **F5:** The system shall allow administrators to view and manage attendance records.
- **F6:** The system shall log attendance data into a CSV or Excel file using Pandas

4 Other Non-functional Requirements

4.1 Performance Requirements

- The system shall process face recognition and mark attendance within 2 seconds per person.
- The system shall store attendance data with a minimum of 95% accuracy in real-time recognition scenarios.

4.2 Safety and Security Requirements

- The system shall ensure that sensitive personal data (faces) are stored securely using encryption techniques.
- Only authenticated users shall be able to register, delete, or update their face data.

4.3 Software Quality Attributes

- **Reliability:** The system shall maintain high accuracy in recognizing faces, even under varying lighting conditions.
- **Usability:** The Tkinter-based user interface shall be intuitive and easy to navigate for administrators.
- **Scalability:** The system shall be able to handle multiple users and sessions without performance degradation

5 Other Requirements

- **Database Storage:** The system will use CSV/Excel for storing attendance logs. Optionally, integration with a database system can be implemented for large-scale deployments.
- **API Integration:** The system may provide API endpoints for integrating with other management systems (e.g., educational software or HR systems) for seamless attendance data exchange.

