**Software Requirements Specification**

**For**

**Face Recognition System (FRS)**

Version 0.2

Prepared by:

* Sara Hassan Ali Hassan
* Nada Hassan Mohamed
* Noha Ahmed Amer
* Marwan Abdel-Rahman
* Yasmeen Nabil Mohamed

## **Table of Contents**

1. Preface 5

1.1 Document Purpose 5

1.2 Target Users 5

1.3 Revision History 5

2. Introduction 5

2.1 Purpose 5

2.2 Scope 5

2.3 Flow Chart of FRS: 5

2.4 Work Breakdown Structure: 5

2.5 Overview 5

3. Glossary 5

3.1 Acronyms, definitions, and abbreviations. 5

4. System Users 5

4.1 System stakeholders 5

4.2 Users objectives 5

5. User Requirements definitions 5

5.1 System Function 5

5.2 Constraints 6

6. System Architecture 6

7. System Functional requirements 7

7.1 Detects user’s face 7

7.2 Authenticate users 7

7.3 Add user’s face 7

7.4 Delete user’s face 7

7.5 Open the door automatically 7

8. Interface requirements 9

8.1 User interfaces: 9

9. Non-functional requirements 12

9.1 Efficiency: 12

9.2 Security: 12

10. System Models and Diagrams 12

11. System Evolution 12

12. Time Plan 12

12.2 Gantt chart 12

# Preface

## 1.1 Document Purpose

The purpose of this document is to provide a detailed and complete specification of Face Recognition System (FRS)

The document will provide an overview of the system in the first section; then each part will be explained in detail in the second and third sections.

## 1.2 Target Users

This document is composed based on the requirements gathered from online searching for what the society miss or has a daily life issue but machine learning and the new technology has the solution for it .

This document is intended to be offered for who want to improve his security system in anything (Automatic Opening Home Doors, Employee Attendance, etc.)

## 1.3 Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Description | Date |
| 0.1 | Sara | Initial | 28-4-2019 |
| 0.2 | Noha | Update structure | 1-5-2019 |

# Introduction

## 2.1 Purpose

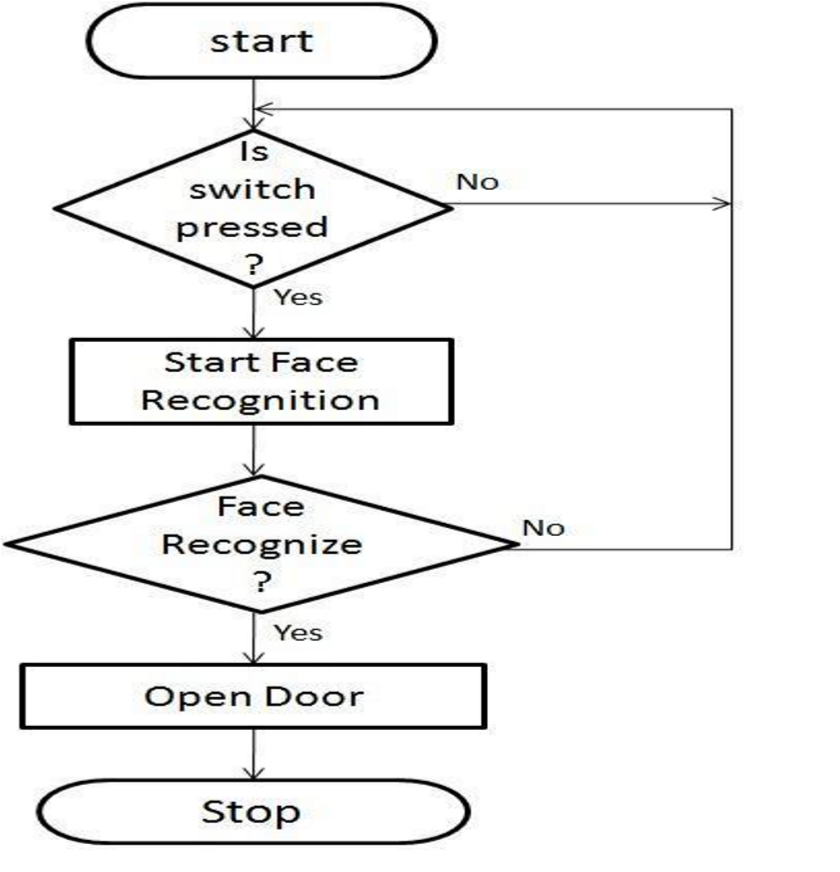
FRS aims to uniquely identify and verify a person by comparing and analyzing patterns based on the person's facial contours, this aim is beneficial in many applications as social media can tag people automatically and security systems as smart opening doors which will be our main concern in this document, to help achieve the following:

* Keep records of all persons that has authentication
* The System must be able to search for faces in images as an input and search for a matching face in folder and then show the result.
* Opening the door automatic

## 2.2 Scope

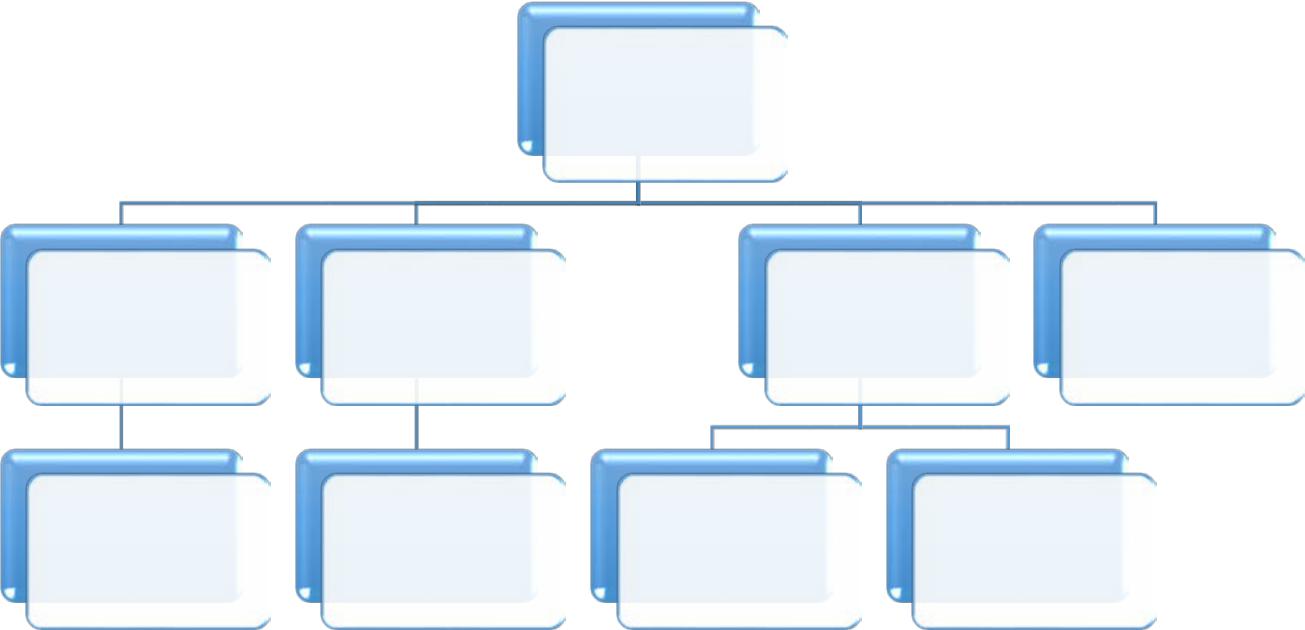
FRS is a software for storing people images and then for any further door entry the system extract and detect the person image and then give permission or reject the person depending on the images stored from the system. It basically increases the security

## 2.3 Flow Chart of FRS:



**Is person in front of camera**?

2.4 Work Breakdown Structure:



Face

Recognition

System

|  |  |  |  |
| --- | --- | --- | --- |
| Insert Image | Data Analysis | Validate | Give final |
| Image | result |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| User enters | DBMS | Check Image | Make |
| his Photo | from DB | comparisons |
|  |

## 2.5 Overview

This document is organized as follows: first, an Overview description of the Face recognition System (FRS) and its high-level functions are presented (section 2.1 and 2.2). Section 2.3 states types of users who can use FRS. Then a list of general constraints that should be followed, assumptions and dependencies are presented in sections 2.4 and 2.5. Section 2.6 shows future work that should be done.

Section 3 in the document provides a detailed description of the system functions and requirements.

Finally, section 4 presents some helping information and diagrams that will facilitate the understanding of this document.

**3. Glossary**

3.1 Acronyms, definitions, and abbreviations

* **FRS** : Face Recognition System
* **DB:** Data Base
* **Administrator:** anyone who want to secure his place by using our system
* **Users:** Any people use this software

# 4. System Users

## 4.1 System stakeholders

* System Engineer
  + Responsible for requirements gathering
  + Responsible for development
  + Responsible for deployment and support
* Administrator
  + Add users
  + Delete Users

## 4.2 Users objectives

* System Engineer:
  + Gain Experience in software engineering and development
* Administrator:
  + Ensure safety and security

# 5. User Requirements

* 1. System Functions:

1. Detect user's face
2. Authenticate users
3. Add user’s face
4. Delete user’s face
5. Open the door automatically

## Constraints

* Technologies Limitations
  + No support for Linux

# 6. System Architecture

# Image result for web application architecture

# 7. System Functional requirements

## 7.1 Detects user’s face

When the user stand in front of the camera, the system will detect the user’s face and identify it by

1. capturing an image of the user
2. then extract is features
3. compare the features to ones the system has in its DB
4. finally the system find a matching or not

## 7.2 Authenticate users

If the system finds a matching then it authenticate the user and gives a permission to open the door automatically else the system rejects the user and the door will stay locked

## 7.3 Add user’s face

The Administrator can add a user’s face into the system’s DB

## 7.4 Delete user’s face

The Administrator can delete a user’s face from the system’s DB

## 7.5 Open the door automatically

The door open automatically after receiving a permission to open

# 8. Interface requirements

## 8.1 User interfaces:

# 9. Non-functional requirements

## 9.1 Efficiency:

The system should be responding within two seconds.

## 9.2 Security:

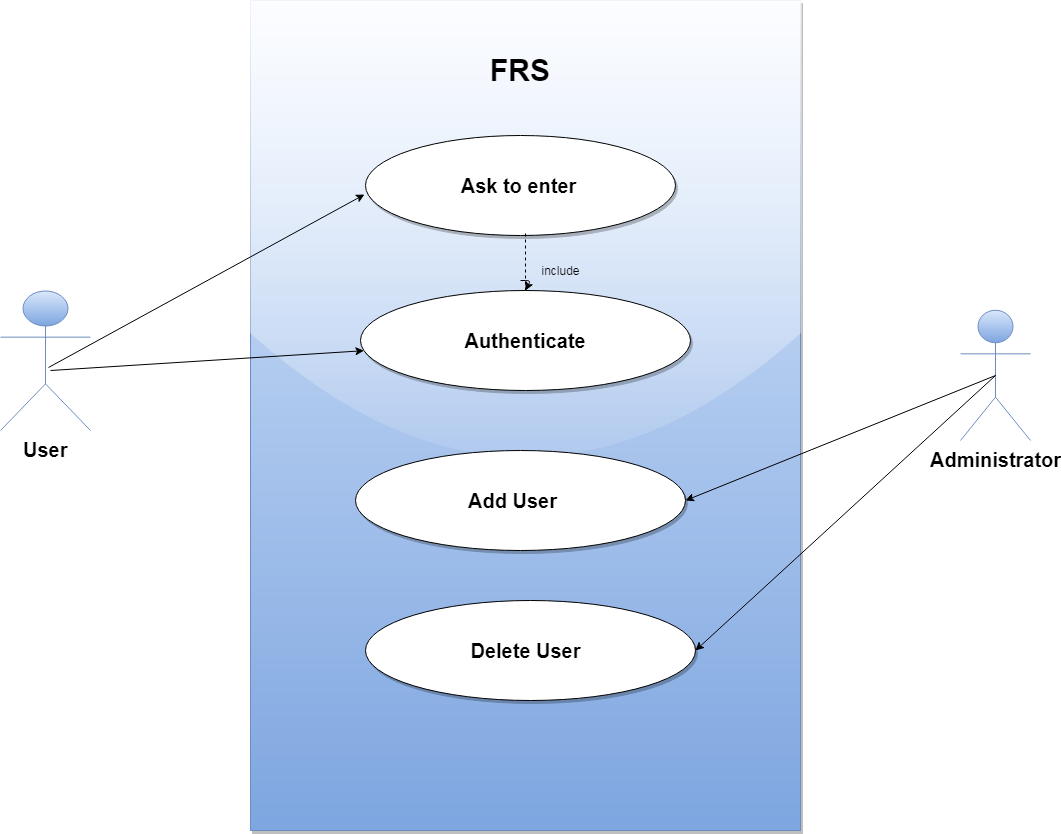
No one can access the system from outside the system’s DB

# 10. System Models and Diagrams

Here you should include the system’s context diagram, use case diagram, class diagram and sequence diagrams for any needed functional requirement.

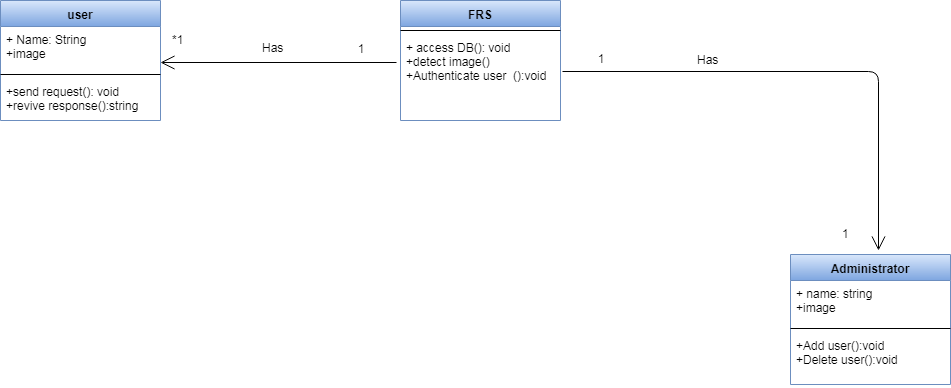
Tools: [https://www.draw.io](https://www.draw.io/)

* Use case:

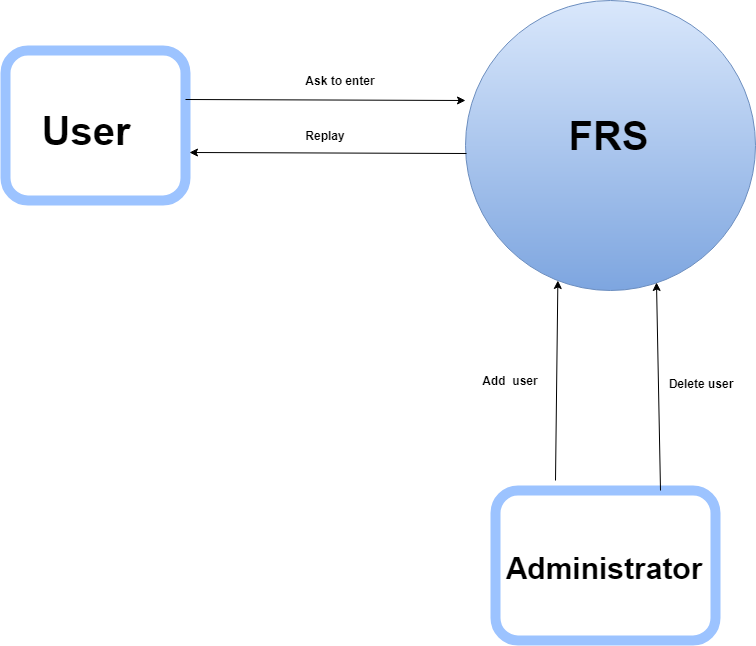


***Scenario:-***

* + **Use case: Ask to enter**
* **Participating Actors:** User
* **Entry Conditions:** when user want to enter to home.
* **Exit Conditions**: when request is reject or accept
* **Flow of events:**
* When the user want to enter the home .He is stand in front of the camera, the system will detect the user’s face.
* **Quality Requirements**: (Performance)
* User receives a response from the system in less than 2 seconds.
* **Include:** Authentication.
* Class Diagram:



* Context Diagram:



# Sequence Diagram:

# C:\Users\7\Downloads\diagram\7.png

# 11. System Evolution

* The system should be able to work on different operating systems.
* It should work properly on devices with low specifications. Only the server on which it’s installed will be powerful.

**12. Time Plan**

## 12.1 Gantt chart

