Facial Recognition Access Control System
Functional Specification
by

Hoda Ahmed;
Student ID: C00214991

April 17, 2020

Abstract

Companies and organizations often overlook the importance of physical security to equipment rooms due to the increase of cybercrime and the emphasis that cybersecurity industries put on cybercrime. This report discusses the importance of physical security to equipment rooms and proposes a solution to fix this flaw: The Facial Recognition Access Control System. Facial recognition is the future, and it has already started being implemented in most places for security, criminal identification, law enforcements and even social media where users can upload photos of other people and tag them.

Contents

1 Introduction

1.1 Document Purpose

This Functional Specification Document is a document that provides detailed information on how the facial recognition access control system will function and the requested behavior. This document provides a clear idea of the actors that will be interacting with the system and in what way they are expected to interact with the system, what they should expect from the system and what they can cannot do while interacting with the system.

1.2 Project Scope

This document focuses in all scopes of the project, taking into considerations every aspect of it. Ranging from the simple desktop application on the Raspberry Pi, to the frontend and backend of the web application, to the operation of the lock and how it will be implemented to carry out its function in locking or unlocking the door.

2 Project Overview

2.1 System Functions

The facial recognition access control system is a system that uses multi-factor authentication, including biometric authentication, to grant users access to equipment rooms. The system will be used to control unauthorized access to equipment rooms in corporate buildings, through a Raspberry Pi computer. This system will use Amazon AWS Rekognition API to process users' faces and determine whether they should be granted access to the rooms or not.
The system will also keep a log of the users that have been granted access to the rooms and the date/time at which the room was accessed. Denied access to users will also be logged. Administrators will be able to view these logs and view user information and other related information through the web application that will be created. The web application may also be used by normal users to update their information and view their own access logs.

As explained thoroughly in the Research Document for the Facial Recognition Access Control System, there are a few solutions that already exist in the market that offer the same service offered by this project. However, the unique aspect about this FRACS project is that it offers access control, taking into consideration both security and privacy. The other solutions that exist in the wild integrate facial recognition into their full monitoring suite, storing captured images and videos, and not fully considering the privacy of the system users. This is not the case for FRACS, as the only information stored on them is a face ID, and the image captured is stored only for the duration of the processing and is permanently deleted after. And this is a great advantage to us.

2.2 System Actors

There is a total of 6 actors in this system. Depending on the addition of features in the future, the number of actors may increase. The actors that will be communicating with the system are as follows:

- **Administrator:** The administrator is responsible for the management of the facial recognition access control system. They are responsible for managing the users that should be allowed access equipment rooms by registering them to the system, removing them, or black-listing them temporarily. They also have the authority to view logs of room access through the web application.
- **User:** The user is the person who is trying to gain access to the equipment rooms. They will interact with the system by entering their username and password, and then getting their face scanned by the camera, which is connected the Rekognition API. If the credentials and the face pattern match, the door is unlocked, and they are granted access.
- **Rekognition API:** The API is considered as an external system which will be responsible for verifying that the user trying to gain access is authorized to do so.
- **Twilio SMS API:** The API is also considered as an external system which will be responsible for sending an SMS code to the admin to verify themselves when attempting to register a

user into the system.

- **Raspberry Pi (& camera):** The Raspberry Pi will be used for all the user interaction with the system, as well as the controlling of the solenoid lock. The camera will assist the Rekognition API in determining whether the user is authorized to access the room or not by capturing the user's face and sending the image to the API so that it can verify the person.
- **Solenoid Lock:** The solenoid lock will be attached to and controlled by the Raspberry Pi to lock/unlock the room.

2.3 System Components

The facial recognition access control system consists of two sets of components: Hardware components and software components:

2.3.1 Hardware Components

- **Raspberry Pi** [1]**:** The Raspberry Pi is where a desktop application will be designed for users to enter their credentials to verify themselves. It will also be responsible for providing power to the physical lock that will be attached to it.
  The Raspberry Pi has a GPIO board that's outline below, with the purpose of each pin in the described in the image.

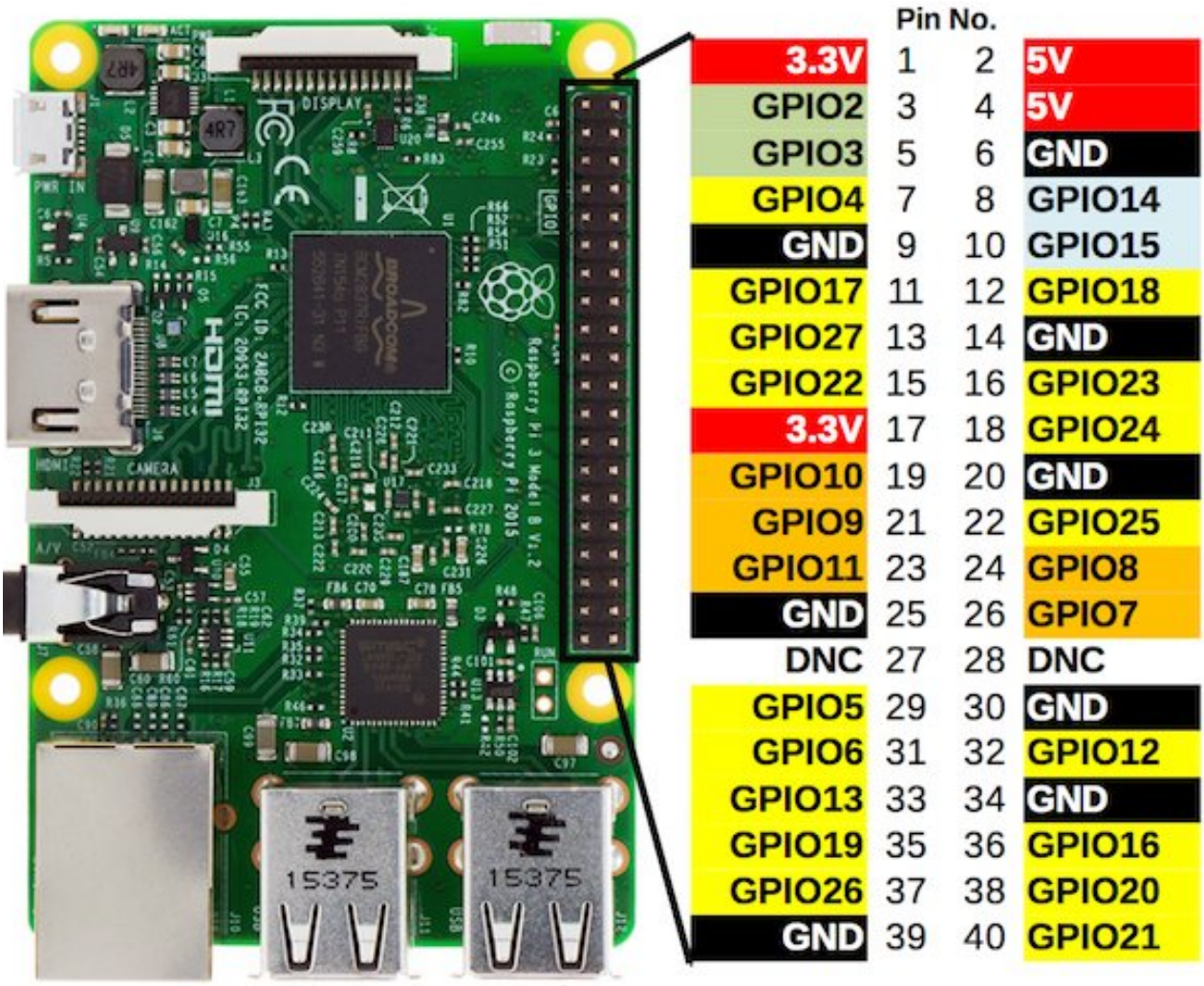

Figure 1: Raspberry Pi 3 GPIO Board

- **Raspberry Pi camera:** The camera will be attached to Raspberry Pi to capture the user's face trying to access the equipment room.
- **LCD touchscreen:** This touchscreen will provide the user interface and will be the interactive point for the user.

[1] https://blog.adafruit.com/2018/06/01/some-thoughts-on-raspberry-pi-gpio-programming-in-c-piday-raspberrypi-raspberry_pi/

- **Solenoid lock:** The solenoid lock will be responsible for locking/unlocking the door of the equipment room. It will be attached to and controlled by the Raspberry Pi, by receiving power from it and unlocking the door when given a signal and when the door is expected to unlock.
- **Relevant cables and connectors:** e.g. for connecting Raspberry Pi to LCD touchscreen, Raspberry Pi to solenoid lock, and solenoid lock to door.

2.3.2 Software Components

- **Rekognition API:** API to be used for all the facial recognition operations e.g. face detection, normalization, feature extraction etc.
- **Twilio SMS API:** API to be used for sending authentication code through SMS to admin for high-level operations like registering new user in system.

3 System Requirements

3.1 Use-Case Diagrams

A Use-Case Diagram (UCD) for this system is shown below in Figure 1. After some analysis of the system development process, the Use-Case Diagram is created. This Use-Case Diagram may be slightly modified later, during the system development.

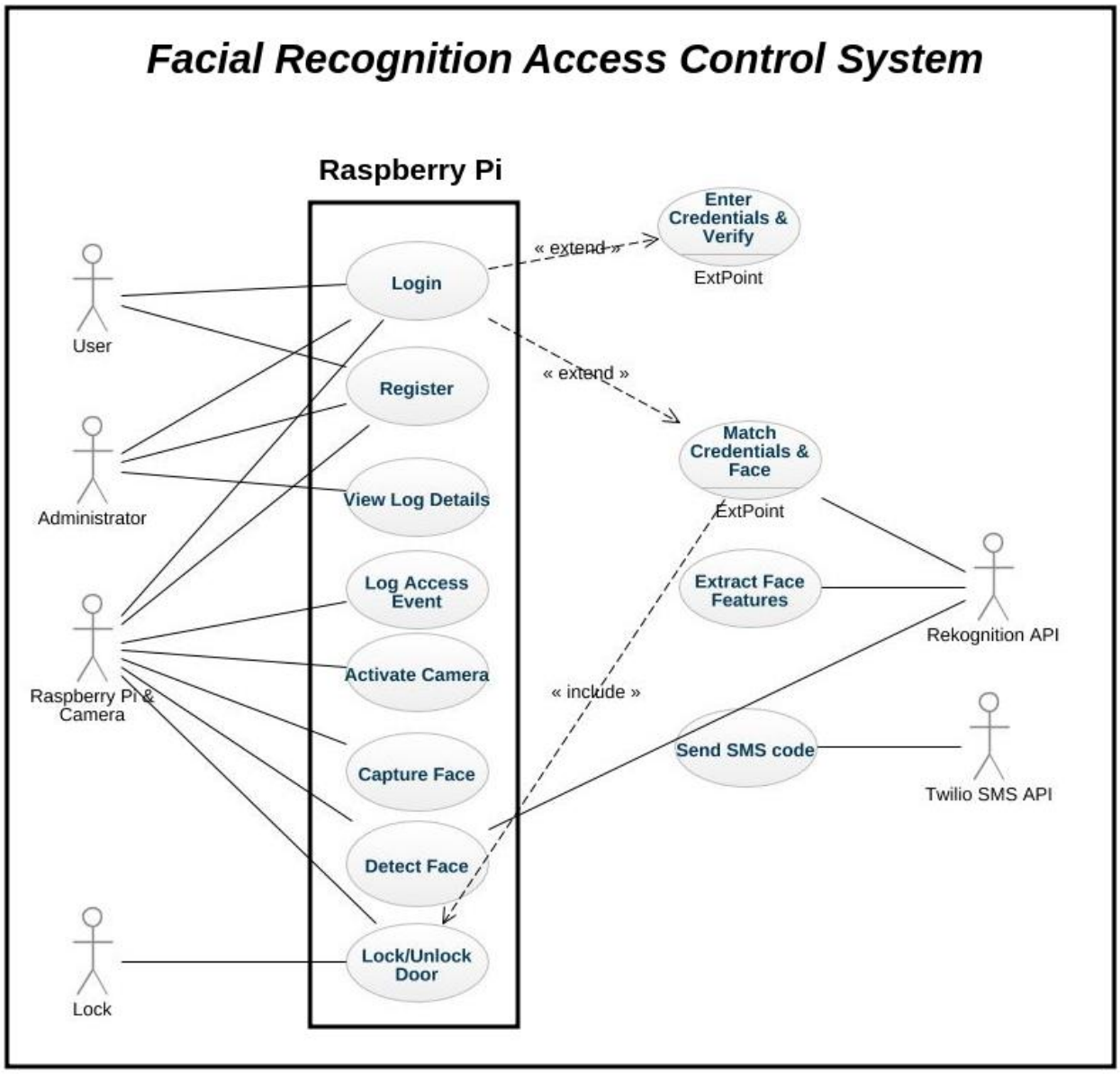

Figure 2: Facial Recognition Access Control System Use-Case Diagram

3.2 Brief Use-Cases

3.2.1 Login

Description: This use case begins when a user wishes to login to the system to access the room.
Actors Involved: Admin, User, Raspberry Pi, Rekognition API.
Steps:

1. User clicks on Login button on desktop app
2. User enters their username/password and clicks login
3. Camera is enabled, countdown of 3 starts and then user's face is captured
4. If credentials match user's face, door is unlocked.
5. Access log recorded to database

3.2.2 Register User

Description: use case begins when the admin wishes to register a user to the system.
Actors Involved: Admin, User, Raspberry Pi, Rekognition API.
Steps:

1. Admin clicks on Register button on desktop app.
2. Admin enters their username/password and clicks login.
3. Camera is enabled, countdown of 3 starts and then user's face is captured.
4. SMS code is sent to Admin's phone.
5. Admin enters code into system.
6. If code is correct, user now enters their information in form on desktop application.
7. User clicks next.
8. Camera is enabled and user gets their face scanned and stored, along with form data.
9. Registration is successful and user can now access room.

3.2.3 Log User Room Access Details

Description: Use case begins when user attempts to login to system and access the room.
Actors Involved: Raspberry Pi.
Steps:

1. User logs into the system using Login use case.
2. Raspberry Pi enters new entry in database with info about access details.

3.2.4 View Users' Room Access Logs

Description: Use case begins when admin wishes to view room access logs through web application.
Actors Involved: Admin
Steps:

1. Admin logs into web application
2. Admin selects the log they wish to view, and can filter or sort data on the web application.

3.2.5 Capture Face

Description: Use case begins when a user has logged in and had their face scanned. This use case also takes place when a user is registering to the system.
Actors Involved: Camera, Rekognition API
Steps:

1. Camera is enabled
2. Face is captured and image is stored in Raspberry Pi.
3. Image of face is examined using Rekognition API.
4. Response is received from Rekognition API.
5. Image is permanently deleted from Raspberry Pi.

3.2.6 Lock/Unlock Door

Description: This use case takes place after the user authenticated themselves and is authorized to access the room.
Actors Involved: Raspberry Pi, Lock
Steps:

1. Raspberry Pi will send signal to lock to trigger it back, leaving the room unlocked for 5 seconds.
2. Room is unlocked and user can access it.
3. Lock becomes untriggered and falls loose, locking the door back.

3.2.7 Remaining Use Cases

The remaining use cases are extended and they are primarily carried out by the Rekognition API. They focus on detecting the faces captured by the camera and comparing them with the faces stored in the database in order to determine whether the user should be granted access or not.

3.3 External Libraries/Tools/Systems

Under this heading, external tools that will be used in this system are discussed. There is only 1 external system that will collaborate with this system; however, this might change in the future.

- **Amazon AWS Rekognition API:** this API will be used to carry out the facial recognition part of the system. Every time the camera is launched, an API call will take place to carry out facial recognition. It will also hold a list the IDs that correspond with the faces of authorized users that can access the equipment room.

- **Twilio SMS API:** this API will be used to send an SMS that contains a code to the mobile phone of an admin who attempts to register a user to the system. This is an additional layer of security for ensuring better security.

4 Supplementary Specification

To define the supplementary specifications for this system the FURPS+ model will be used. FURPS+ is an acronym that stands for Functionality, Usability, Reliability, Performance, Supportability and the plus is used to specify constraints around the design, interface and implementation among a few other things. This section of the document will first give a brief overview of each part of the FURPS+ model followed by the specifics for this project.

4.1 Functionality

For the purpose of this system, the following functionality will be required:

- The system should capture a user's face after they enter their username & password to authenticate themselves.
- The system will allow authorized users gain access to equipment rooms, which only certain individuals should gain access to.
- The system should log every time a user accesses the room and should include the details of the date and time the room was accessed at.
- The system should allow admins to view logs and edit users that can gain access to these equipment rooms.

4.2 Usability

This metric should describe how easy it is for users to use the system.

- The desktop application on the Raspberry Pi should be user-friendly and clear.
- There should be a login page, with an onscreen keyboard with big, visible keys so that they don't enter wrong characters.
- The web application for the admins should be interactive and user-friendly and should show the logs clearly and succinctly

4.3 Reliability

This is a metric that shows the possibility of the system failure.

- The system must be connected to the Rekognition API all the time, or at least every time the desktop app on the Raspberry Pi is launched.
- The admin should be able to access the web application at any time to access & view the room access logs.

4.4 Performance

The performance metrics describes how the system behaves when users interact with it in various scenarios.

- The Rekognition API should be fast enough, so that when a face appears in front of the camera, the user should be either granted or denied access within 4 seconds or less.
- If access is granted, the Raspberry Pi should act very fast to unlock the door immediately.

4.5 Supportability

Supportability asks the following questions: "Is it testable, extensible, serviceable, installable, and configurable? Can it be monitored?"

- The web application should be accessible from all browsers and devices (e.g. Mobile phones, Tablets, etc.)
- The system should be easily expanded upon in the future to add more features. For example, an additional biometric authentication method.

4.6 + (in FURPS+)

The + in FURPS+ covers any additional needs that are not met in the previous headings:

- **Security**
  - Web application & all its data must be delivered over HTTPS
  - User credentials stored in database must all be encrypted (passwords must be hashed and salted using a suitable, secure hashing algorithm)
  - The camera will be activated after every attempt of entering credentials, no matter whether the credentials exists in the database or not. If the credentials are incorrect, a generic message is displayed (e.g. credentials & face don't match). This is to discourage an intruder from carrying out a brute force or a credentials harvesting attack.
  - After the processing of every image of a user stored on the Raspberry Pi, every bit of the said image will be over-written randomly (or with 0s), and then deleted permanently. This is to prevent the image from being viewable even if it was retrieved.

5 Testing

To test out this system, the Raspberry Pi and the LCD touchscreen will be placed outside one of the frequent classrooms that are accessed by my class. The admin roles will be carried out by me and one other student and three other students will act as authorized users to access the room.
Two other students will be registered to the system with the help of one of the admins and then attempt to access the room, which should be unlocked with no issues. An authorized user will be removed and will try to access the room, but will not be granted access.
The admins will log into the website to view the logs and be able to filter the results. A non-admin user will attempt to log in to the website; this should not be a successful login. Web attacks will also take place, like cross-site scripting (XSS), SQL injection and others, to ensure that the web application is not susceptible to a web attack.

List of Figures