Logo

Description automatically generated

David Campion, Michael Flynn, Nathan Field & Nikita Fedans

A 2FA Security System for securing single-entry rooms

DMN sYSTEMS

DMN Alarms

**DMN Alarms**

David Campion, Michael Flynn, Nathan Field & Nikita Fedans

Table of Contents

[Introduction 2](#_Toc116660732)

[Hardware 3](#_Toc116660733)

[Circuit Diagrams 3](#_Toc116660734)

[Hardware Requirements: 4](#_Toc116660735)

[Data, Data Storage and Data Processing 5](#_Toc116660736)

[What Data is gathered 5](#_Toc116660737)

[How it will be stored 5](#_Toc116660738)

[How it will be processed 5](#_Toc116660739)

[Security 5](#_Toc116660740)

[How to secure stored data 5](#_Toc116660741)

[Physical security 5](#_Toc116660742)

[The UI, User and User Testing 5](#_Toc116660743)

[The UI 5](#_Toc116660744)

[The User 6](#_Toc116660745)

[User Testing 6](#_Toc116660746)

[Versioning 6](#_Toc116660747)

# Introduction

Our IoT application is an alarm system for single-entry rooms containing valuable documents of some form. After passing fingerprint and facial recognition, the electric strike door lock is unlocked, and entry is granted. There is a door sensor to detect when the door is open and closed. A Database is used to store fingerprint, facial images, date, and time of access. Each stage of the process is accompanied by a speaker, buzzer, and LCD display to guide the user through the system. A flask website is to be used to facilitate a typical front-end user.

Team Members Roles:

David Campion: Frontend + UX

Michael Flynn: Database + Testing

Nathan Field: Backend + Testing

Nikita Fedans: Backend + Architecture

# Hardware

## Circuit Diagrams

**Circuit:**

Diagram

Description automatically generated

**Fritzing circuit commentary:**

* Raspberry pi 3 used for diagram. (4 was unavailable)
* Assuming LEDs need 2.2 forward voltage, the resistor needed is a 68.75 ohm one.
* The grounds of all the LEDs may be connected, and then plugged into a single ground pin on the pi, freeing up n-1 ground pins.
* Momentary switch button will be used for sending a signal to raspberry pi.
* No physical pull down resistor is needed for the switch since an internal pull down resistor may be assigned to a pin through code.
* LCD display requires a large number of wires and specific part was unavailable in fritzing.
* On the PIR sensor in the fritzing diagram, the VCC, OUT, and GND are inverted compared to the PIR sensors bought from pihut, so exercise caution when wiring the real thing.

**Problems to solve:**

* Research how to restrict the “Field of vision” of the pir sensor to only see a specific area.
* Build homemade door sensor.
* Cross-communication of 2 raspberry pi’s for each side of a doorway.

## Hardware Requirements:

**Acquired:**

1x pir sensor

1x pi camera module (specially made, plugs into special port)

1x buzzer (works with GPIO and 220-ohm resistor)

1x speaker (hdmx, connected to aux, perhaps powered through USB)

**Purchases required:**

1x fingerprint sensor <https://thepihut.com/products/round-all-in-one-capacitive-fingerprint-sensor-d?variant=41540357849283>

1x USB to UART <https://thepihut.com/products/usb-to-uart-module-micro-mini-type-a-or-type-c?variant=41771472584899>

1x 1.3” IPS LCD Display <https://thepihut.com/collections/raspberry-pi-screens/products/1-3-ips-lcd-display-module-for-raspberry-pi-pico-240x240>

1x electric strike lock <https://www.amazon.co.uk/gp/product/B0027VB810?psc=1>

1x door switch (commercial or built)

1x button

# Data, Data Storage and Data Processing

## What Data is gathered

## How it will be stored

## How it will be processed

# Security

How to secure stored dataSecurity is one of the most crucial aspects of this project. As we are storing an individual’s personal information such as their name, fingerprint, and image, we must ensure that this sensitive information is therefore protected and not used in a malicious way.

### Securing the system

To ensure the employees personal information is protected we based our security measurements on the STRIDE threat model. The first threat we looked at was spoofing. One of the ways we try to combat spoofing is the user of our multi factor authentication system, this system prevents the user from gaining access to the door if they do not have the authorised fingerprint and facial recognition.

Tampering is another threat that could happen on both front and end, It is crucial to our system that there is different levels of access for each employee, i.e an admin should be able to change code on the front end or backend, and a backend developer should not be able to add, edit or delete records from the database.

Repudiation was another issue that we were concerned with, if an unauthorised user got into the room and took sensitive documents’ they could deny taking them and we would have no proof to verify that they took it. We decided the best course of action would be to keep a log of which users accessed the room and at what time, this would be stored in a table in our database and will have a relationship with the employee table (employee to access one to many).

Next we looked at how our system and database could prevent information from being leaked, i.e. how to stop people from finding our database name and password. Based off our lecturers guidance we decided to use env variables to prevent unauthorised access, this means that were we to send our code or push it to a site like GitHub then the aforementioned names and passwords will not be visible and there no information leak. We also hashed passwords for extra protection in the event of our database being breached.

Denial of Service attacks were a big worry for us, we therefore did extensive research on the ways in which to prevent these attacks from affecting us, with firewalls being the answer that appeared the most (VPNs and content filtering were other options in preventing these attacks.

In a recent article by Verizon, they claim that 80% of all data breaches happen due to compromised passwords. (https://www.verizon.com/business/resources/reports/dbir/) , this worrying statistic got us thinking about how we can come up with more ways to stop users with access to the database and the admin section from having weak passwords, with the simplest solution being to make the user’s password to contain a special, uppercase, lowercase and numerical character. We also looked a using multi-level authentication such as having a message being sent to your phone with a secure code like how companies such as Google and Amazon do it, however as our project is only small in size and we aren’t expecting attacks in this area, we decided that having a secure password will do for now.

### Privacy

Regarding privacy there was several laws and legislations that we had to consider. As we were using an individual’s personal information, we had to abide by GDPR and right to be forgotten law.

Firstly, will be looking at the GPDR and how we need to adhere to it’s 7 principles. Due to our usage of an employee’s personal information, it is crucial that we only use this information in a lawful, fair, and transparent manner. In essence we must be open and honest with our employees in how we are using their data as well as not use in inappropriate way such as sending it to third party companies (we are assuming they won’t consent to this)

Another way we must adhere to the GDPR is to only collect data of the employee that is necessary and relevant to what we will be using it for. So, for example we will only need the data such as name, fingerprint, and image and therefore we should be collecting information such as married status or something similar.

A final principle that I think we must adhere to in the GDPR is storage limitation. This law states that data should only be kept for as long as necessary. So if an employee leaves the company in which the system is being implemented then they must be removed from the database as they information is no longer Is required.

This principle ties in with the second legislation we will be looking which is the Right to be forgotten Act. In essence this act allows a person to have their personal information removed from a companies database or from the internet. It is important for us as a company to consider this act as if one of the employees leave our company and ask to be removed from the system we must do so.

## Physical security

System mounted to wall.

Locking mechanism.

# The UI, User and User Testing

## The UI

Flask website

LCD display



## The User

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**A picture containing graphical user interface

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

## User Interaction Flowcharts

1. Entering from the outside, available here: <https://www.canva.com/design/DAFPBEU1cgY/_33aS6_us3ViciJL2B3JKQ/edit?utm_content=DAFPBEU1cgY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>
2. Exiting from the inside, available here: <https://www.canva.com/design/DAFPB6XopUE/RulFr5eydPou51SDKPa_SQ/edit?utm_content=DAFPB6XopUE&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

## Diagram Description automatically generated

## Diagram Description automatically generated

## User Testing

# Versioning

Version 1: Document Creation – 07/10/2022

Version 2: Introduction, Table of Contents and Hardware requirements – 08/10/2022

Version 3: Updated team roles, hardware requirements, STRIDE, User personas and circuit diagram – 14/10/2022