**SecureGate**

**An IoT security project for granting controlled access to infrastructure using multi factor authentication.**

|  |  |  |
| --- | --- | --- |
| **DOCUMENT INFORMATION** | | |
| **Student Name:** |  | **Kieron Garvey** |
| **Student#:** |  | **96358157** |
| **eMail:** |  | **96358157@mail.wit.ie** |
| **Supervisor:** |  | **Frank Walsh** |
| **Module:** |  | **Placement & Project** |
| **Course:** |  | **Higher Diploma in Computer Science** |
| **Version:** |  | **1.0** |
| **Effective Date:** |  | **23rd January 2025** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Declaration of Authenticity** | | | |
| I declare that the work which follows is my own, and that any quotations from any sources (e.g. books, journals, the internet) are clearly identified as such by the use of ‘single quotation marks’, for shorter excerpt and identified italics for longer quotations. All quotations and paraphrases are accompanied by (date, author) in the text and a fuller citation is the bibliography. I have not submitted the work represented in this report in any other course of study leading to an academic award. | | | |
| Student: | Kieron Garvey\_\_\_\_\_\_ | Date: | 23/01/2025\_\_\_\_\_\_ |

|  |
| --- |
| **Acknowledgements** |
| **TBD** |

|  |
| --- |
| **Preface** |
| **TBD** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| **Revision #** | Description | **Author** | **Effective Date** |
| 1.0 | Initial Draft Submission | Kieron Garvey | 23/01/2025 |

**Table of Contents**

[1 Introduction 5](#_Toc188530082)

[1.1 Objective 5](#_Toc188530083)

[1.2 Purpose 5](#_Toc188530084)

[1.3 AMS (Original) Proposal 5](#_Toc188530085)

[1.4 Acronyms and Abbreviations 6](#_Toc188530086)

[2 Research and Analysis 6](#_Toc188530087)

[2.1 Context 6](#_Toc188530088)

[2.2 Existing Solutions 6](#_Toc188530089)

[2.3 Tools 6](#_Toc188530090)

[2.3.1 Proxmox 6](#_Toc188530091)

[2.3.2 Portainer 7](#_Toc188530092)

[2.3.3 Docker 7](#_Toc188530093)

[2.3.4 TinyTuya 7](#_Toc188530094)

[2.3.5 Supabase 7](#_Toc188530095)

[2.3.6 MQTT (Message Queuing Telemetry Transport) 7](#_Toc188530096)

[2.3.7 OpenCV 8](#_Toc188530097)

[2.3.8 Face-Recognition 8](#_Toc188530098)

[2.3.9 Fast API 8](#_Toc188530099)

[2.4 Technologies 8](#_Toc188530100)

[2.4.1 Python 8](#_Toc188530101)

[2.4.2 React 9](#_Toc188530102)

[2.4.3 React Native 9](#_Toc188530103)

[2.4.4 Flutter 9](#_Toc188530104)

[2.5 Equipment 9](#_Toc188530105)

[3 Design & Modeling 11](#_Toc188530106)

[3.1 User Personas 11](#_Toc188530107)

[3.1.1 Admin Persona 11](#_Toc188530108)

[3.1.2 Chemist Persona 12](#_Toc188530109)

[3.1.3 Lab Technician Persona 12](#_Toc188530110)

[3.1.4 Cleaner Persona 12](#_Toc188530111)

[3.2 User Case Diagram 12](#_Toc188530112)

[3.3 User Interface 12](#_Toc188530113)

[3.4 Physical Model Design 12](#_Toc188530114)

[3.4.1 Electrical Drawing 13](#_Toc188530115)

[3.4.2 Front View Design 14](#_Toc188530116)

[3.4.3 Side View Design 14](#_Toc188530117)

[4 Attachments 16](#_Toc188530118)

[4.1 Attachment#1:- Electrical Drawing 17](#_Toc188530119)

[4.2 Attachment#2:- 18](#_Toc188530120)

[4.3 Attachment#3:- 18](#_Toc188530121)

# Introduction

## Objective

Automated Management Systems (AMS) was a project I created for the Computer Systems & Networks Module of the Higher Diploma in Computer Science course.

My objective is to refactor this project removing several of its dependencies, rename it too SecureGate and making it more business friendly.

## Purpose

## AMS (Original) Proposal

AMS is an innovative IoT project focused on enhancing security and privacy in a controlled company environment. AMS addressed the challenges of granting controlled access to a room and the devices within the room without compromising sensitive information.

AMS integrates facial recognition and remote access technologies to regulate entry, ensuring only authorized personnel enter the controlled environment. It employs sophisticated security monitoring to restrict access to specific devices, preventing unauthorized usage within the environment.

The system optimizes energy consumption by selectively powering on devices based on user permissions. When multiple individuals are present, AMS requires facial recognition or QR code authentication for device activation, ensuring access control.

Additionally, AMS implements secure login protocols for devices, requiring facial recognition and/or QR code authentication, thereby enhancing security measures within the controlled environment. By providing detailed access control and authentication mechanisms, AMS ensures heightened security and restricted access while optimizing resource utilization.

|  |  |
| --- | --- |
| **Original Project Documents** | |
| **DESCRIPTION** | **URL** |
| AMS Proposal | <https://github.com/ki321g/AMS/blob/main/Documentation/AssignmentProposal.pdf> |
| GitHUB Repository | <https://github.com/ki321g/AMS> |
| Video Walk Through | <https://youtu.be/SJLO5kULhWg> |

## Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Name** | **Description** |
| AMS | Automated Management Systems |
|  |  |
|  |  |

# Research and Analysis

planning

## Context

planning

## Existing Solutions

planning

## Tools

I am planning on using some of the following tools this may change as I research the project a bit more.

### *Proxmox*

Proxmox delivers powerful, enterprise-grade solutions with full access to all functionality for everyone - highly reliable and secure. The software-defined and open platforms are easy to deploy and manage. Proxmox Virtual Environment is a complete open-source platform for enterprise virtualization. With the built-in web interface you can easily manage VMs and containers, software-defined storage and networking, high-availability clustering, and multiple out-of-the-box tools using a single solution.

**URL:** <https://www.proxmox.com/en/>

### *Portainer*

Portainer is a universal container management platform. It's multi-cluster and multi-device support which means you can manage environments of any type, anywhere (Docker and Kubernetes, running on dev laptops, in your DC, in the cloud, or at the edge), and we don't require you to run any specific Kubernetes distro.

**URL:** <https://www.portainer.io/>

### *Docker*

Docker is an open-source platform designed to simplify the development, deployment, and management of applications. It uses containerization, a technology that packages an application and its dependencies into a portable and isolated environment known as a container.

### *TinyTuya*

This python module controls and reads state of Tuya compatible WiFi Smart Devices (Plugs, Switches, Lights, Window Covers, etc.) using the local area network (LAN) or the cloud (TuyaCloud API).

**URL:** <https://github.com/jasonacox/tinytuya>

### *Supabase*

Supabase is an open source Firebase alternative.

Start your project with a Postgres database, Authentication, instant APIs, Edge Functions, Realtime subscriptions, Storage, and Vector embeddings.

**URL:** <https://supabase.com/>

### *MQTT (Message Queuing Telemetry Transport)*

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for efficient communication between devices in IoT and other applications. It enables secure, reliable data transfer with minimal bandwidth usage, ideal for interconnected systems and remote device management.

**URL:** <https://mqtt.org/>

### *OpenCV*

OpenCV (Open Source Computer Vision Library) is a powerful open-source software library designed for real-time computer vision and image processing applications.

**URL:** <https://opencv.org/>

### *Face-Recognition*

Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library.

Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

**URL:** <https://github.com/ageitgey/face_recognition>

### *Fast API*

FastAPI is a modern, high-performance web framework for building APIs with Python. It is designed to deliver robust, efficient, and user-friendly APIs

**URL:** <https://fastapi.tiangolo.com/>

## Technologies

Again, I am planning on using the following technologies, but this may change as I research the project a bit more.

### *Python*

Python is a high-level, versatile programming language known for its simplicity and readability. It's used across various domains such as web development, data analysis, artificial intelligence, scientific computing, and more. Python's clean syntax and extensive libraries make it beginner-friendly while offering powerful capabilities for building applications, automating tasks, handling data, and implementing algorithms efficiently.

### *React*

React is a popular JavaScript library for building user interfaces, primarily for web applications. Developed and maintained by Facebook, React enables developers to create reusable UI components that efficiently update and render when data changes.

### *React Native*

React Native is a framework derived from React that focuses on building mobile applications for iOS and Android using JavaScript. Instead of rendering to the web's DOM, React Native components translate to native widgets, delivering a performance and feel akin to apps developed with native technologies.

### *Flutter*

Flutter is an open-source UI toolkit created by Google for crafting natively compiled applications for mobile, web, and desktop from a single codebase. Unlike React Native, Flutter uses Dart, a programming language also developed by Google. It boasts its own rendering engine, Skia, allowing for complete control over every pixel rendered on the screen. This ensures highly customizable, visually rich UIs that look and feel consistent across platforms.

## Equipment

The system will incorporate the following devices

|  |  |
| --- | --- |
| **Name** | **Image** |
| Raspberry Pi 4 & 5 |  |
| Lrtzcbi Touchscreen Portable Monitor 14 Inch |  |
| pHAT Stack Kit for Raspberry Pi | PIMORONI PIM322 |
| Raspberry Pi Camera Module 3 Wide NOIR | RASPBERRY-PI SC0875 |
| USB Accelerator Edge TPU Coporocessor | CORAL G950-01456-01 |
| ELEGOO Upgraded Electronics Fun Kit |  |
| Mini Portable USB Night Light |  |
| Woox R4785 smart plug |  |
| Rekavin USB Plug |  |

# Design & Modeling

planning

## User Personas

planning

### *Admin Persona*

planning

### *Chemist Persona*

planning

### *Lab Technician Persona*

planning

### *Cleaner Persona*

planning

## User Case Diagram

planning

## User Interface

planning

## Physical Model Design

planning

### *Electrical Drawing*

A diagram of a machine

Description automatically generated

### *Front View Design*

A close-up of a door

Description automatically generated

### *Side View Design*

A diagram of a door structure

Description automatically generated

# Attachments

## Attachment#1:- Electrical Drawing

A diagram of a machine

Description automatically generated

## Attachment#2:-

## Attachment#3:-