|  |
| --- |
| Open Source Security |
| Security System |
|  |

|  |
| --- |
|  |

Contents

[Introduction 2](#_Toc152064062)

[Background Research 2](#_Toc152064063)

[What is a distributed system? 2](#_Toc152064064)

[Mechanisms of a Distributed System 2](#_Toc152064065)

[Zookeeper 2](#_Toc152064066)

[Mesh 2](#_Toc152064067)

[Dynamo 2](#_Toc152064068)

[Communication Methods 2](#_Toc152064069)

[I2C 2](#_Toc152064070)

[Wi-Fi 2](#_Toc152064071)

[UART 2](#_Toc152064072)

[GPIO 2](#_Toc152064073)

[Project Proposal 3](#_Toc152064074)

[Control Node – Keypad 3](#_Toc152064075)

[Function Overview 3](#_Toc152064076)

[Unlocking 3](#_Toc152064077)

[Locking 5](#_Toc152064078)

[Alarm Test 5](#_Toc152064079)

[Support 6](#_Toc152064080)

[Bugs 6](#_Toc152064081)

[Testing 7](#_Toc152064082)

[Limitations 7](#_Toc152064083)

[Application 7](#_Toc152064084)

[Dashboard 8](#_Toc152064085)

[New Nodes 8](#_Toc152064086)

[Communication System 9](#_Toc152064087)

[Jewelry Box Node 9](#_Toc152064088)

[Safe Node 9](#_Toc152064089)

[Bug Testing and Critical Reflections during integrations 9](#_Toc152064090)

# Introduction

Block diagrams

# Background Research

## What is a distributed system?

A distributed system “is one which components located at networked computers communicate and coordinates their actions only by passing messages” \*, this suggests the idea that distinct computing systems will need to communicate with one another in order to perceive what actions they need to do, and thus creating what is known as a distributed system.

<nodes>

<messages>

## Mechanisms of a Distributed System

### Zookeeper

### Mesh

### Dynamo

## Communication Methods

To be able to send information from each node within a distributed system to one another, there has to be standardized methods of communication, for the purposes of our project, this comes in the form of I2C, Wi-Fi, UART and GPIO pins. Each of the communication methods used are discussed below and the reasonings to why they have been used.

### I2C

Inter-Interconnected Communication

### Wi-Fi

### UART

### GPIO

# Project Proposal

# Control Node – Keypad

## Function Overview

A close-up of a computer

Description automatically generatedNode 1 is one of the main interfaces between the end user and the system. The main function of this node is to provide a terminal that the user can input a code to disarm the system, so it doesn’t go off in the day when you’re just walking around the area. This node has a 16 key keypad and a display screen for inputs and visual prompts, these prompts are the unlocking function to disarm, a locking function when unlocked to rearm the system, an alarm sound test to make sure the systems alarm is working, a network function to test if it is connected to the other nodes, and a help function that shows a website and a phone number to contact for help.

### Unlocking

A computer screen with wires and a keypad

Description automatically generatedWhen the system is wanted to be unlocked the user can go to the terminal as seen in the photo below. The system will show text asking for a password to be inputted, this password can be a mix of numbers 0-9 and letters A-D, for the testing the password was ‘123A’ but the password can be any number of characters long.

A small electronic device with a blue screen

Description automatically generatedA small electronic device with a blue screen

Description automatically generatedA close up of a circuit board

Description automatically generatedIf the password is inputted correctly then a welcome screen is shown followed by the menu for further actions, shown below. These actions allow for further control and testing of the system.

A finger on a finger pressing a blue screen

Description automatically generatedA finger pressing a keypad on a small rectangular device

Description automatically generatedA finger touching a small rectangular device

Description automatically generatedOn the other hand, if the password is inputted incorrectly, the system will tell you this and tell you then there are 2 more attempts allowed. If the password is correct on any of these other attempts, the system carries on as normal. But if the password is inputted in wrong 3 times in a row, the system will go into warning mode. As seen in the image below, this means there is a countdown that gives the chance for one more attempt of the password, but if the password is wrong again, or the timer runs out, then the alarm sounds, a code is sent to the cloud to alert it something is wrong, and the keypad system locks up.

### Locking

A small green electronic device with a blue screen

Description automatically generatedOnce unlocked the user has the option to relock the system with one button press. As seen in the image below, the option on the menu shows up as ‘Lock’ press it and the system goes back to asking or a password. This function allowed the user to easily arm the space that its installed in, protecting the building and themselves. The system is built to be more problematic to unlock than lock, for security reasons.

### Alarm Test

A finger on a computer screen

Description automatically generatedIncluded in the menu is an alarm test, much like a smoke alarm, every now and then you’ll want to test if the alarm is working. It sounds the alarm for a few seconds, enough to know it works, then turns off.

### Support

A close up of a computer

Description automatically generatedThe last option on the menu list is simply a help page with the website link on it and a phone number to contact a help line for the system.

## Bugs

For Bugs for integration, see Bug Testing and Critical Reflections section.

For Node 1 there were many bugs that needed to be fixed, as show in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Node | Problem | Description | Solution |
|  | A-B Password Ending | If the password ended in either A or B, the corresponding option would be selected in the proceeding menu. | Clear the key pressed queue after unlocking. |
|  | Debounce | A single press of a key would result in ~30 inputs of the pressed key. | A delay was added after every button press to avoid multiple inputs being received. |
|  | Infinite locking loop | Locking the system would result in an infinite loop if the password ended in A. | Solved when the previous bug-fixes were introduced, as the combination of them resulted in the first menu option (Lock) always being selected. |
| Control Pad (1) | Alarm spam | Pressing enough keys would eventually stop the alarm. | The inputs would act as another attempt on password, when 3 attempts were made, the timer reset. |
|  | Fake timer (display) | During/after the lockdown timer, password attempts could still be seen on the LCD. | Introducing a full lockdown mechanism disabled the keypad, not allowing any further inputs during alarm phase. |
|  | Fake timer (function) | Entering the correct password after the alarm started would unlock the system. | After Lockdown has initiated, the only method to stop is using an external device (cloud). |
|  | Alarm crash | Introducing a second timer during the alarm phase would crash the system. | While not confirmed, it is believed either the testing board only had one timer, or the library only allowed for one timer.  Code was re-developed to only require one timer. |

## Testing

MAKE A TESTING TABLE

Vigorous testing was performed on the node to make sure all functioned as well as expected.

We tested getting the password wrong many times, to get the security alarm to go off.

We tested getting the password wrong, but getting it correct at each step-in order. Once the wrong password section was done, we tested getting the password right, then locking the system and then getting it wrong again to check that the system did work as intended and still alarmed once locked.

We tested the menu system, to make sure each option worked as intended, that the lock option locked and the help option showed the information.

We tested that the C and D keys moved the menu around.

We tested the alarm test.

## Limitations

# Application

This section of the documentation describes the purpose of Application. The Application (“App”), is a fundamental part of the distributed system, providing a crucial way of monitoring and controlling the security system from a remote location. The App is split into three main sections: Dashboard, New Nodes, and Information.

## Dashboard

A screenshot of a computer

Description automatically generated

<Figure Caption>

The above figure demonstrates the overview that is seen on the dashboard. The main component of this page is the table which displays if the node is active, the Node Name, addresses and if the Node Alarm is active. The Update button allows the consumer to effectively update the node list to check if there are any changes to the node, the list will only update automatically when the Node alarm changes from “Neutral” to “Active”.

## New Nodes

A screenshot of a computer

Description automatically generated

<Figure Caption>

This section works by entering the details of the node, and then submitting them, to create the node, deleting them, to delete the node, or Update node, to update the node. How this works is explained in more detail in the communication system section of this documentation.

# Communication System

This section of the project is not easily seen when viewing the security system, it is effectively the middleware of the project, this is because it is found on all of the devices but is not easily seen when looked at.

# Jewelry Box Node

# Safe Node

# Bug Testing and Critical Reflections during integrations