# Secure Door Lock

Software Design Specification

## Team Members


| | |
|---|---|
| Luke Bucher | lbucher2017@my.fit.edu |
| Christopher Kiefer | ckiefer2019@my.fit.edu |
| James Pabisz | jpabisz2020@my.fit.edu |
| Warren Smith | wsmith2019@my.fit.edu |


## Faculty Advisor


Dr. Marius Silaghi
Graduate Professor at Florida Institute of Technology
msilaghi@fit.edu


## Client


Dr. Marius Silaghi
Graduate Professor at Florida Institute of Technology
msilaghi@fit.edu

Revision History

| Version | Name | Reason For Changes | Date |
|---|---|---|---|
| *1.0* | *Luke Bucher* | *Initial Creation* | *10/01/2022* |
| 1.1 | Luke Bucher | Revision for Presentation | 10/05/2022 |

# 1. Introduction

## 1.1 Purpose

This design document will document the various subsystems of the Secure Door Lock Project.

## 1.2 System Overview

Our goal in this project is to create a secure, easy to use, and inexpensive product that has the security features that users expect from a door lock while integrating modern connectivity features that users can access through the mobile application. The secure door lock and the accompanying mobile application will allow the user to check whether their doors are locked or unlocked, lock and unlock the door, unlock the door with their or trusted individuals faces, and receive a notification with a picture of who is at their door.

## 1.3 Design Map

Within this document the full system for the Secure door lock system will be presented with its intended configuration. Systems will be broken down by their sub system: Mobile Application, Server, and Firmware. While additional breakdowns can be performed within these applications.

All itemized requirements that are found within the accompanying SRS are covered within this document to ensure validity and their relationships with each other. Additionally Integration within the subsystems is highlighted within any Diagrams.

# 2. Design Considerations

## 2.1 Assumptions

Designs within this document assume a single device, server, application follow through. While there's room for vertical scaling of devices and users requirements are designed for one device and one user currently.

## 2.2 Constraints

Applications must be performant due to the limited resources available within the Raspberry pi. Ensuring functionality is not hampered by limited resources is critical.

Similarly due to the requirements of a facial recognition profile we must ensure that a performant system is chosen that focuses on accuracy and maintains a lightweight footprint within the server system.

## 2.3 System Environment

The mobile application will be created within a React JS system that will be focused on running properly on the android operating system.

The server instances will be containerized within a lightweight Linux container for resource friendly API response. As well the container will be hosted within an AWS instance and compute power will be provided by this instance.

The firmware of the IOT device will be on a lightweight Ubuntu-Core operating system which will be operating within a Raspberry Pi 3 module.

# 3. Architecture

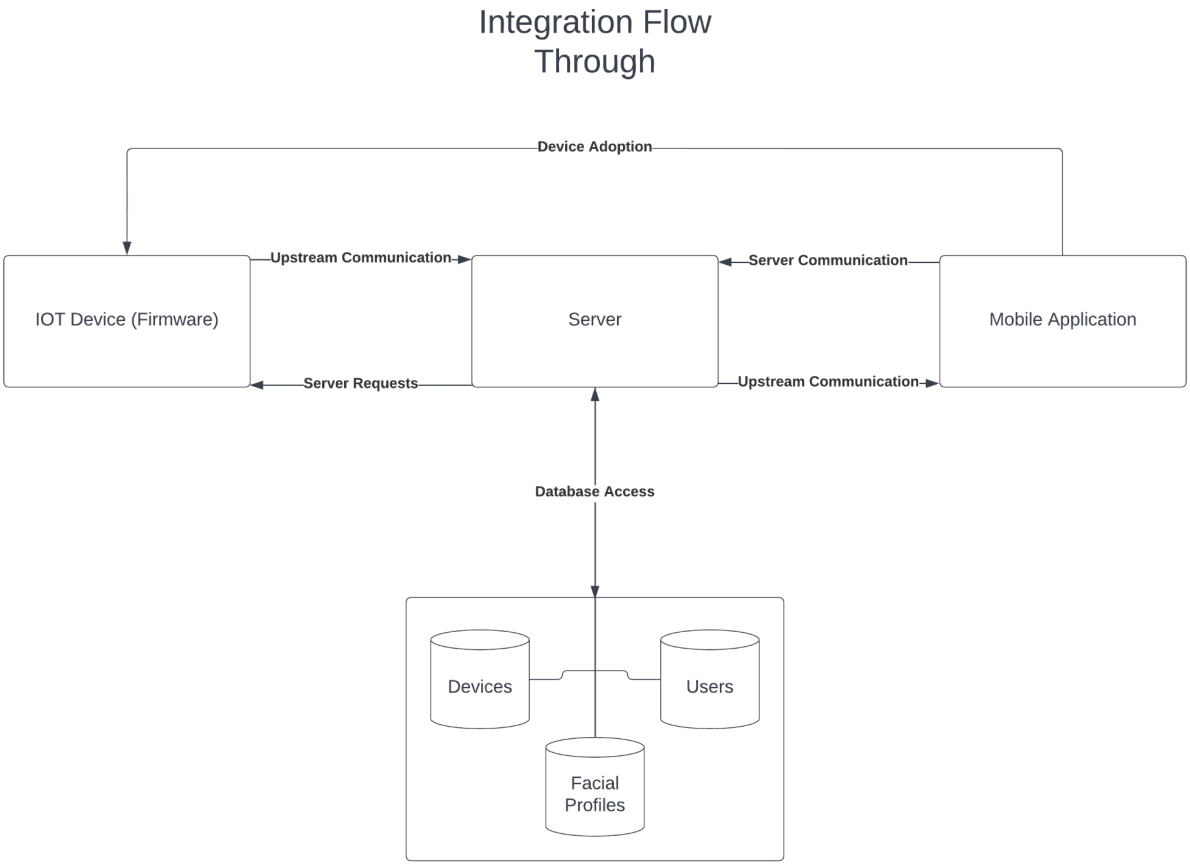
The architecture provides the top level design view of a system and provides a basis for more detailed design work

Provide or reference a detailed description and diagrams of the architecture..

## 3.1 Overview

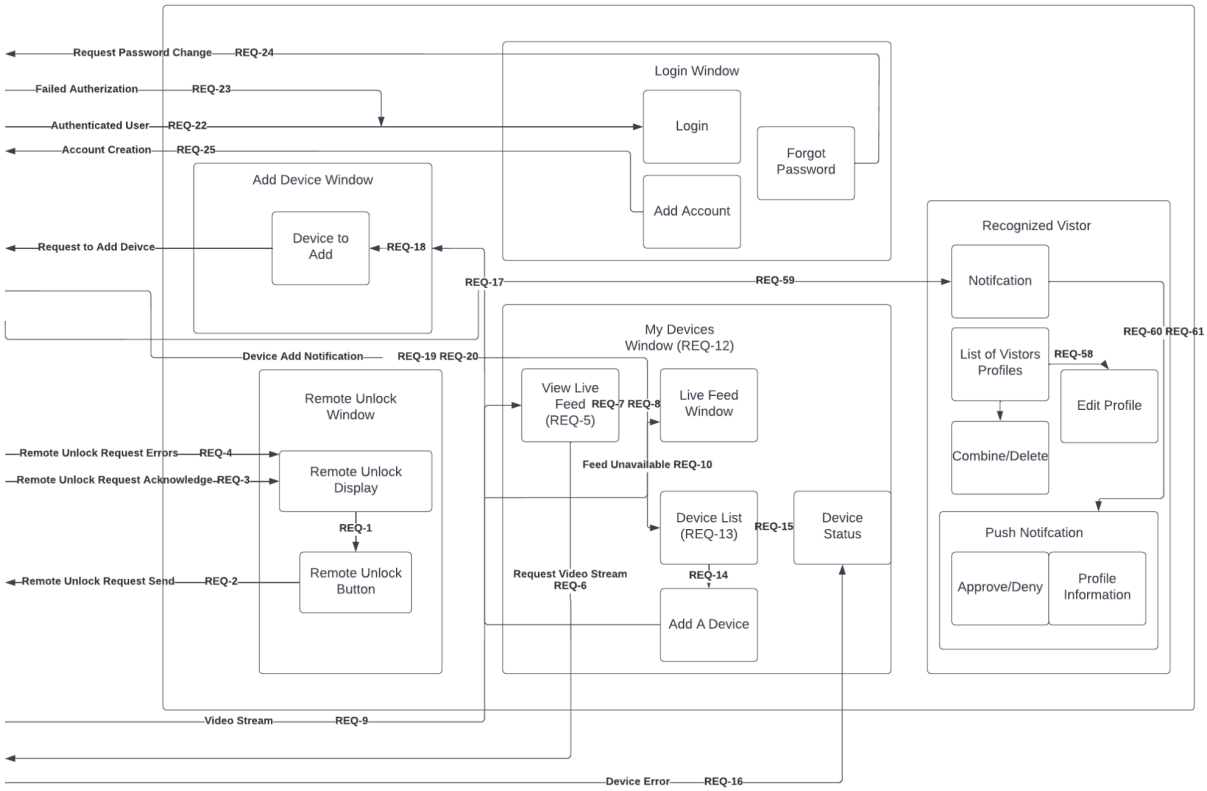
This section provides a high level overview of the structural and functional decomposition of the system. Focus on how and why the system was decomposed in a particular way rather than on details of the particular components. Include information on the major responsibilities and roles

the system (or portions) must play.

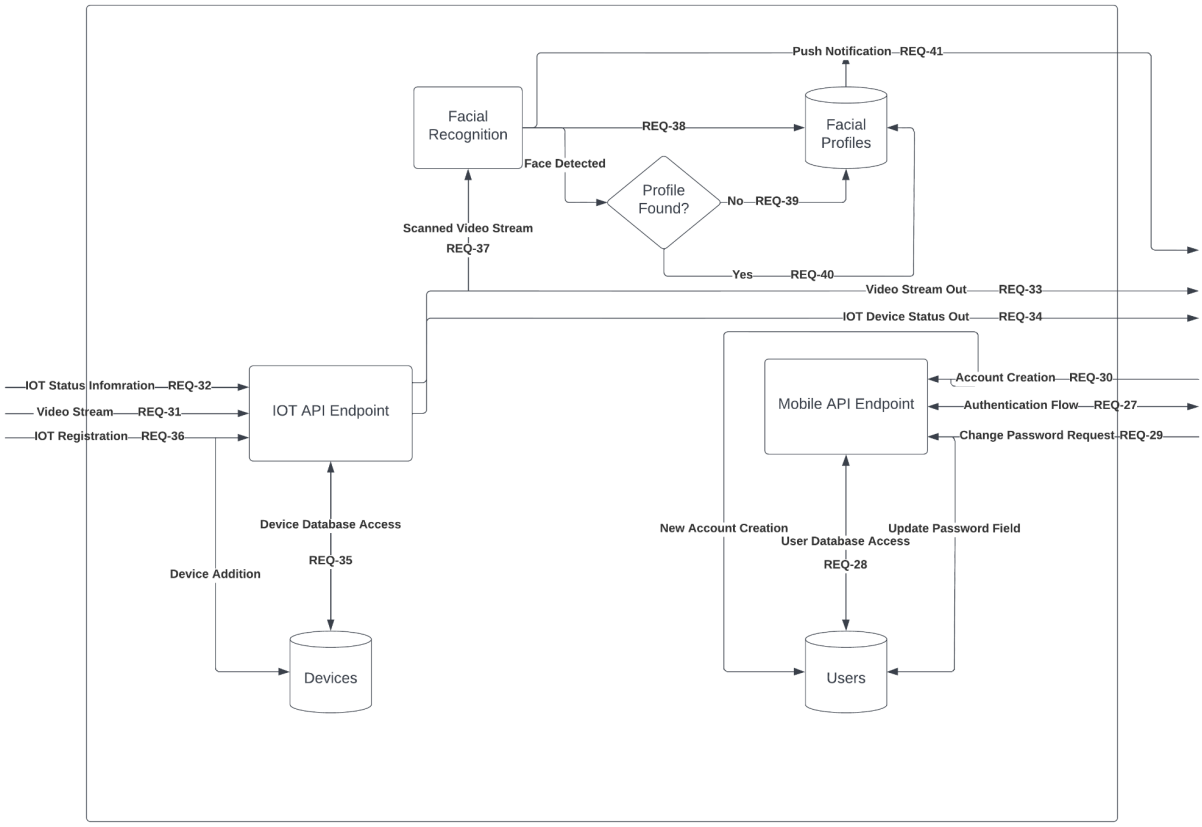


# **3.2** Sub System Diagrams

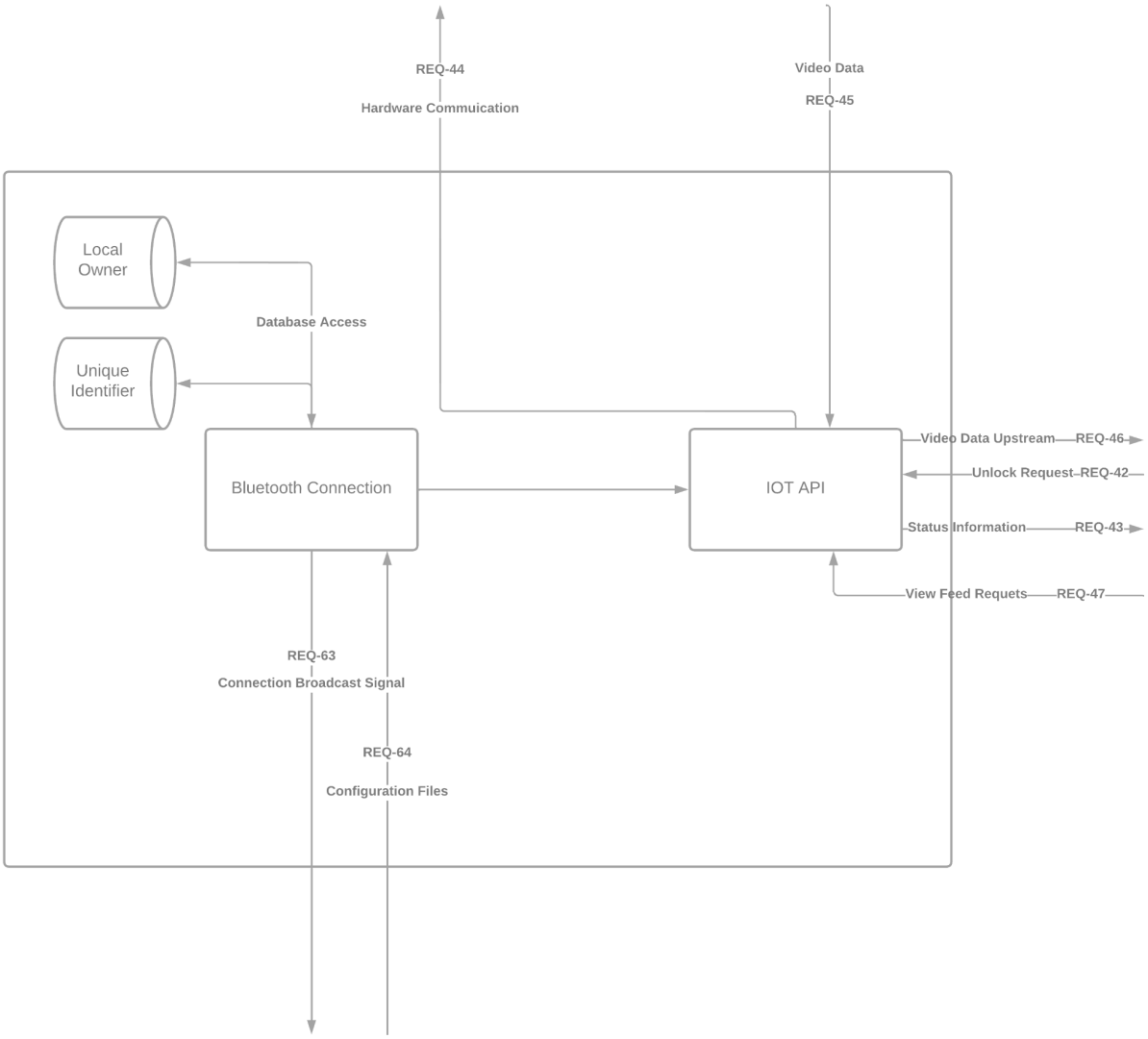
## 3.2.1 Mobile Application Process Diagram



## 3.2.2 Server Process Diagram



## 3.2.3 Firmware Process Diagram



## 3.3 Modules

### 3.3.1 Mobile Application

Remote Unlock

| | |
|---|---|
| Input | N/A |
| Output | Put Request |
| Description | The mobile application communicates with the endpoint and sends a request to change the hardware status within the IOT device |

Device View

| | |
|---|---|
| Input | Image Data |
| Output | Get Request |
| Description | The mobile application communicates with the endpoint and sends requests to view the live camera feed. In response the Endpoint begins to send a video stream of the selected device. |

Recognized Vistor

| | |
|---|---|
| Input | Push Notification (Image, Name, Approve/Deny system) |
| Output | Put Request |
| Description | The recognized visitor system receives a push notification from the endpoint of a requested visitor. The recognized visitor system provides the end user an approve deny system to allow entry as well as the photo and name of the visitor. |

Login

| | |
|---|---|
| Input | Get Requests (Login Requests) |
| Output | Authentication Token |
| Description | The user will provide login credentials to the system through input fields. From there if the credentials are accepted an authentication token will be given to the system and the user |

| | |
|---|---|
| | will be logged into their respective account. |

### 3.3.2 Server

IOT Endpoint

| | |
|---|---|
| Input | Post/Get Requests From the IOT Device |
| Output | The IOT endpoint manages traffic from the IOT to either the Mobile application or to the facial recognition system. |
| Description | The IOT endpoint is meant to direct relevant data where it needs to be delivered. The endpoint can send data to the IOT device that needs to be sent. As well relevant sensor and image data can be sent to the mobile application |

Mobile Application Endpoint

| | |
|---|---|
| Input | Post/Get Requests from the Mobile Application |
| Output | The mobile application manages traffic from the mobile application to either update database information or mange the IOT device |
| Description | The mobile application endpoint acts as a director for all traffic that stems from the mobile application. Requests are distributed to their relevant subsystem or database request. |

Facial Recognition

| | |
|---|---|
| Input | Image data |
| Output | Selected facial recognition profile determined from the image data. This is outputted to the registered account |
| Description | The facial recognition scans image data and attempts to locate faces and match to profile data that is saved within the cloud instance. |

| | |
|---|---|
| | Once data is either matched or not found a profile is generated that is then sent to an owner account. |

### 3.3.3 Firmware

Bluetooth Connection:

| | |
|---|---|
| Input | Incoming JSON file from bluetooth |
| Output | Complete the registration process by writing the UUID of the owner to the local storage as well as complete registration within the IOT endpoint |
| Description | The bluetooth connection will accept a connection from the mobile application and alongside provide a proper network to connect to. The authenticated user of the mobile application that started the process will be assigned ownership and the IOT endpoint will be updated with the relevant information. |

IOT API:

| | |
|---|---|
| Input | Post/ Get requests |
| Output | Output video and hardware status information to the IOT Endpoint. As well process post requests by dynamically changing the poll rate of the video capture. |
| Description | The primary IOT API is designed to accept requests from the Upstream connection to modify or report hardware level behavior. All APIs use a restful state structure so listen for POST and GET requests from another upstream connection |

# 4. Database Schema

## 4.1 Tables, Fields and Relationships

### 4.1.1 Databases

A singular database called SDLdata will be created to house all data related to this system

### 4.1.2 New Tables

Three additional tables will need to be created: Users, Devices,Facial recognition profiles.

The user table will contain relevant User data needed for their profile and for authentication.

The Device table will contain device listings as well as their owner account

The Facial Recognition profiles will house the owner account, facial recognition data, and any profile information such as a name and an image.

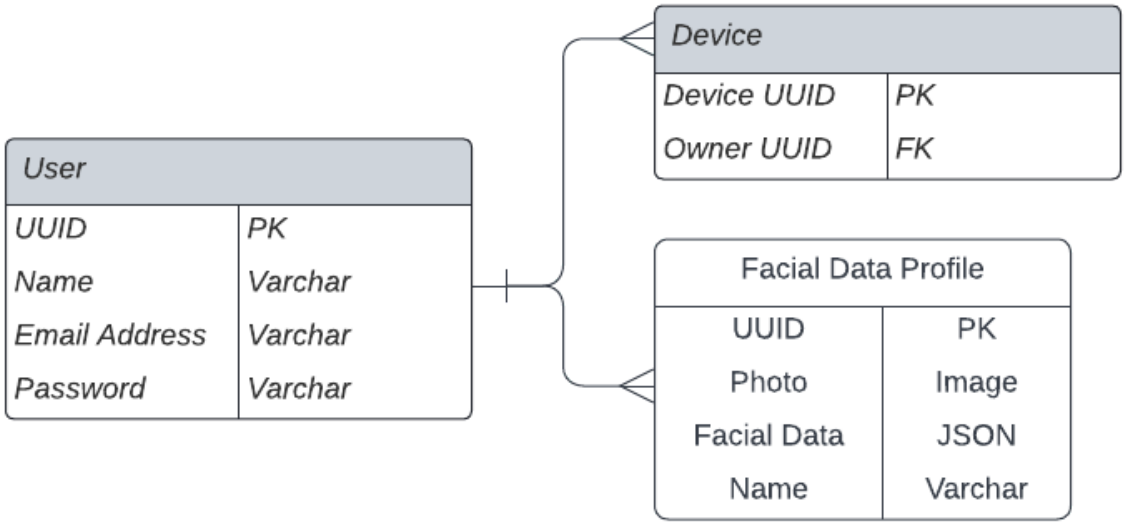
### 4.1.3 New Fields(s)

List any new tables that will be needed, for each one including table name, table description, and related tables.

| Table Name | Field Name | Data Type | Allow Nulls | Field Description |
|---|---|---|---|---|
| User | UUID | Varchar(36) | No | This field is generated on account creation. Used as the primary key for account ownership as well as foreign keys for determining ownership |
| User | Name | Varchar(255) | No | This field is generated on account creation. Used for login actions. |
| User | Email Address | Varchar(255) | No | Used for account registration. Created during the account creation process |
| User | Password | Varchar(255) | No | Fallback until a proper token system is processed. Created during the account creation process. |
| Device | Device UUID | Varchar(36) | No | Primary key for device registration. Stored locally on the physical device |
| Device | Owner UUID | Varchar(36) | Yes | Ownership key designated after a device has been registered to an account. Can be empty when a |

| | | | | |
|---|---|---|---|---|
| | | | | device is discoverable but not registered. |
| Facial Data Profile | UUID | Varchar(36) | No | Primary key of the data profile. Generated when new data is discovered from the facial recognition system. |
| Facial Data Profile | Photo | Image | Yes | Identifying image of the facial profile. Generated on addition to the database. |
| Facial Data Profile | Facial Data | JSON | No | Identifying data generated during the recognition process. |
| Facial Data Profile | Name | Varchar(255) | No | Identifying name used for display. Auto generated when a new entry is created. |

## 4.2 Entity Relationship Diagram




# 5. User Interface Design

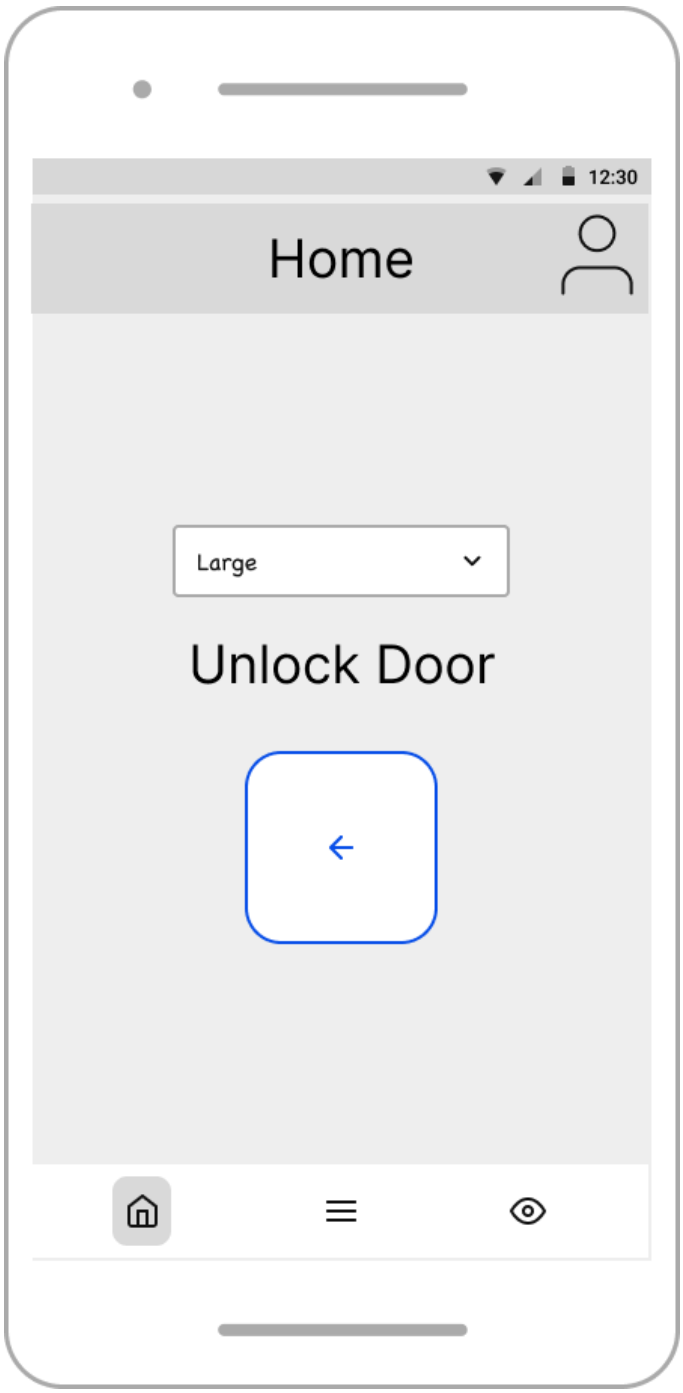
This section provides user interface design descriptions that directly support construction of user interface screens.

## 5.1 Application Controls

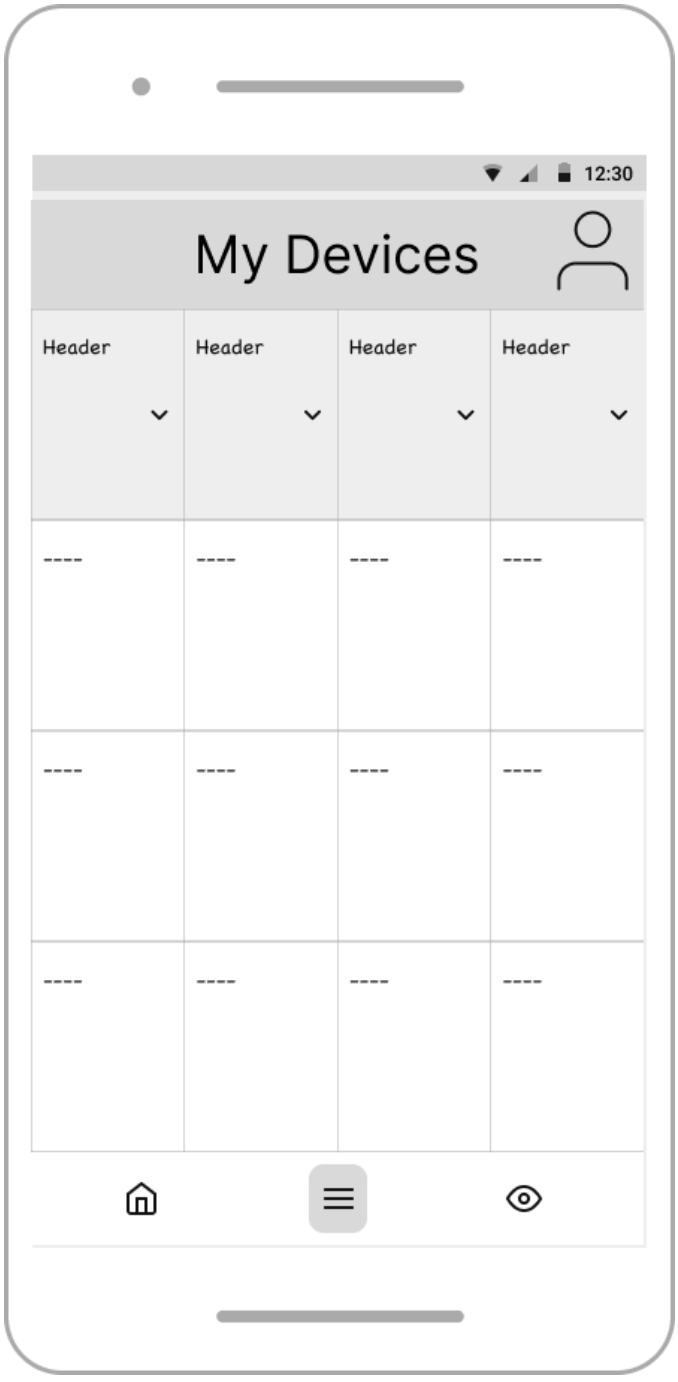
A common navigation bar will be present at all times at the bottom of the screen. As well A profile button will be accessible at the top of the screen for users to view their own profile. The bottom navigation buttons will include a main menu, device listing, and facial recognition profiles

## **5.2** User Interface

5.2.1 Home Screen



5.2.2 My Devices



5.2.3 Visitors Screen