System Name: IoT Simplicity

# Business Requirements

## Problem Statement

Homeowners who buy IoT devices need a centralized system that can control these IoT devices, such as lighting, temperature, and security, in their home. This system should allow for multiple users to interact with it, be easy for users to interact with, be continuously available, and be able to utilize all functionality of the IoT devices.

## Essential Functionality

* Connect to new IoT devices
* Connect to the Wi-Fi
* Connect IoT devices to the Wi-Fi
* Interact with IoT devices

## Target Users

Anyone who buys or wants to use IoT devices

## Business Goals

* Ease of use: Users will want the system to be intuitive to set up and interact with
* Reliability: Users will want the system to be reliably able to connect to the Wi-Fi and the IoT devices.

# Non-Functional Requirements

|  |  |
| --- | --- |
| Scalability | The system should allow up to 100 users to interact with the system and up to 100 IoT devices to be connected. |
| Response Time | The response time should be less than 500 milliseconds. |
| Throughput | The throughput should always be at least 80% of the Wi-Fi router’s capability. |
| Authentication | Each user should have a username and password with which they log into the system. |
| Authorization | There should be a primary owner that can control the level of controls others in the household have. |
| Data Encryption | All data should be encrypted and only information relevant to errors should be send outside the system. |
| Documentation | All troubleshooting strategies should be thoroughly documented. |
| Test Strategies | All methods should be tested several times, such as adding new users, adding new devices, and adding IoT devices to the Wi-Fi. |
| Availability | The app should be available 99% of the time |

# Diagrams

## UML Use Case Diagram

A diagram of a diagram

Description automatically generated

## UML Domain Model

A diagram of a computer system

Description automatically generated

## UML Class Diagram

A diagram of a computer system

Description automatically generated

## UML Sequence Diagrams

### connectSystemToWiFi

A diagram of a device

Description automatically generated

### connectIoTDeviceToWiFi

A diagram of a device

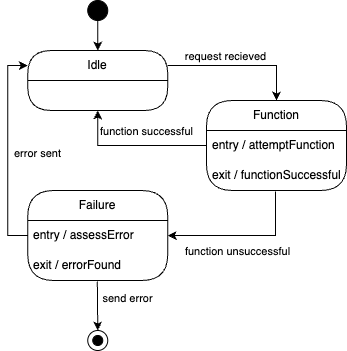
Description automatically generated

### interactWithIoTDevice

A diagram of a device

Description automatically generated

## UML State Diagram: IoT Device



## UML Activity Diagram (Swimlane Diagram)

A diagram of a software flowchart

Description automatically generated

## UML Component Diagram

A diagram of a computer system

Description automatically generated

## Cloud Deployment Diagram

A diagram of a software application

Description automatically generated

## Skeleton Classes and Tables Definition

**Name: User**

User’s Attributes

* ID #: Unique ID created by the system for each user
* Username: Unique String created by user
* Password: Unique protected password created by user

User’s Methods:

* openApp(ID): Uses the user’s ID to open the app
* login(ID, username, string): A user can login to the app by using their username and password. This either grabs an existing ID or creates a new ID for the user.

**Name: Device using App**

Device using App’s Attributes

* Serial Number: The serial number of the device
* MAC Address: The MAC Address of the device
* List of IoT Devices: A list of IoT devices connected to the app

Device using App’s Methods:

displayIoTDevices(listOfIoTDevices): Shows available IoT Devices

**Name: Wi-Fi Router**

Wi-Fi Router’s Attributes

* Wi-Fi Address: Wi-Fi address of the Wi-Fi router
* Name: Name of the Wi-Fi router created by the provider and able to be adjusted by the user
* Provider: Provider of the Wi-Fi router
* Security Type: Security type for the password on the Wi-Fi Router
* Password: Protected password of the Wi-Fi router

Wi-Fi Router’s Methods:

* acceptWiFiLogin(password: password): Uses the password given to it to login a user

**Name: IoT Device**

IoT Device’s Attributes

* ID #: Unique ID created by the system for each IoT device
* Name: Name of the device created by the manufacturer and able to be adjusted by the user
* Manufacturer: Name of the company who created the IoT device
* Device Type: Type of IoT Device (e.g. Light, Thermostat, etc.)
* Capabilities: List of functions that the device can do

IoT Device’s Methods:

* doFunction(ID, capability): A user selects a capability of a device that they want the device to do.

## Design Patterns

GRASP

* Low Coupling: Every object has low coupling, since there is low dependency, and everything can easily be reused. The IoT devices can be modified by multiple users because of this.
* Controller: The app on the device controls all functions of the IoT devices.

SOLID

* Open Closed Principle: The IoT Device class is an open class, as it’s available for extension.

GOF

* Mediator: The Wi-Fi router acts as a mediator between the device with the app and the IoT Device. This allows for the IoT devices to not be tied to a single device and be modifiable by multiple users.

Microservices

* API Gateway: An API Gateway is used to manage the different possible functionalities of the IoT devices.