IoT Based Electricity Conservation System

Final Year Project

2020-2024

A project submitted in partial fulfillment of the degree of

BS in Computer Science



Submitted to

Dr. Abdul Hameed

Department of Computer Science

Faculty of Computing & Artificial Intelligence (FCAI)

Air University, Islamabad

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type (Nature of project) | | | [✔] Development [ ] R&D | | |
| Area of specialization | | | [✔] WebApp [ ] Mobile App  [✔] AI based [✔] Embedded System | | |
| FYP ID | | | F-23-IoT-01 | | |
| **Project Group Members** | | | | | |
| Sr.# | Reg. # | Student Name | | Email ID | \*Signature |
| (i) | 200828 | Muhammad Junaid Nazir | | 200828@students.au.edu.pk |  |
| (ii) | 200924 | Ahmad Imran | | 200924@students.au.edu.pk |  |
| (iii) | 201260 | Azfar Tariq | | 201260@students.au.edu.pk |  |

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

# Plagiarism Certificate

This is to certify that, I **Muhammad Junaid Nazir**, S/D of **Nazir Ahmad**, group leader of FYP under registration no **200828** at **Computer Sciences Department, Air University**. I declare that my FYP report is checked by my supervisor.

Date: May 29th, 2024 Name of Group Leader: Muhammad Junaid Nazir Signature:\_\_\_\_\_\_\_\_

Name of Supervisor: Dr. Abdul Hameed Co-Supervisor: Mr. Hassan Zaib

Designation: Assistant Professor Designation: Lecturer

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HoD: Dr. Mehdi Hassan

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**IoT Based Electricity Conservation System**

**Change Record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author(s)** | **Version** | **Date** | **Notes** | **Supervisor’s Signature** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**APPROVAL**

|  |  |  |
| --- | --- | --- |
| **PROJECT SUPERVISOR** | | |
| Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |  | | |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| **PROJECT MANAGER** | | | |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |

|  |  |
| --- | --- |
| **CHAIR DEPARTMENT** | |
|  |  |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Dedication

*This work is dedicated to our parents, whose unwavering support and encouragement have been a constant source of inspiration throughout our academic journey. Their belief in our potential has been instrumental in achieving this milestone. Additionally, we dedicate this project to our mentors and teachers who have guided us with their knowledge and wisdom. Lastly, we extend our dedication to our friends and colleagues whose camaraderie and support have made this journey memorable and rewarding.*

# Acknowledgements

We would like to express our deepest gratitude to our supervisor, Dr. Abdul Hameed, for his invaluable guidance, continuous support, and patience throughout the development of this project. His insightful feedback and encouragement have greatly contributed to the successful completion of this work. Our sincere thanks also go to our co-supervisor, Mr. Hassan Zaib, for his expertise and assistance with the image processing aspects of our project, which played a crucial role in enhancing the quality of our work. We are also thankful to the Department of Computer Science at Air University Islamabad for providing the necessary resources and a conducive environment for research and development. We extend our sincere thanks to each other as project group members, Muhammad Junaid Nazir, Ahmad Imran, and Azfar Tariq, for the hard work, dedication, and collaboration. Without our commitment and teamwork, this project would not have been possible. Finally, we acknowledge all those who have directly or indirectly supported and contributed to this project. Thank you for your encouragement and support.

# Executive Summary

*Electricity wastage is a significant global issue, particularly pressing in Pakistan due to its ongoing energy crisis. The inefficient use of electrical equipment exacerbates this problem, with students and faculty often leaving devices such as computers, monitors, and peripherals on when not in use. This practice results in substantial energy wastage, contributing to high electricity consumption and increased financial burdens for educational institutions. The proposed project, "IoT-Based Electricity Conservation System," addresses this issue by implementing a smart, automated solution to manage power consumption across various settings.*

*The system employs a high-resolution camera, an IoT Box equipped with a Raspberry Pi, an IoT Platform, a smart board, and a desktop agent. The camera captures live video feeds, which the Raspberry Pi processes using advanced image recognition algorithms to detect human presence. When the system determines that a space is unoccupied, it automatically turns off the electrical devices connected to the smart board and shuts down the PCs. This automated approach ensures that energy is conserved whenever the space is unoccupied, eliminating the need for manual intervention and significantly reducing electricity wastage.*

*The benefits of implementing the IoT-Based Electricity Conservation System are multifaceted. It directly contributes to reducing energy consumption and lowering electricity bills. Additionally, it supports Pakistan's broader efforts to address its energy crisis and aligns with global sustainability goals by curbing carbon emissions and promoting environmental preservation. The project also fosters a culture of energy awareness and responsible consumption among students, faculty, and staff, encouraging a more sustainable approach to energy usage.*

# Table of Contents

[Plagiarism Certificate 3](#_Toc167959040)

[Dedication 6](#_Toc167959041)

[Acknowledgements 7](#_Toc167959042)

[Executive Summary 8](#_Toc167959043)

[Table of Contents 9](#_Toc167959044)

[List of Figures 13](#_Toc167959045)

[List of Tables 14](#_Toc167959046)

[List of Abbreviations 15](#_Toc167959047)

[Chapter 1 1](#_Toc167959048)

[Introduction & Background 1](#_Toc167959049)

[1.1. Background 2](#_Toc167959050)

[1.2. Motivations and Challenges 2](#_Toc167959051)

[1.3. Goals and Objectives 3](#_Toc167959052)

[1.4. Literature Review/Existing Solutions 3](#_Toc167959053)

[1.5. Gap Analysis 3](#_Toc167959054)

[1.6. Proposed Solution 4](#_Toc167959055)

[1.7. Project Plan 4](#_Toc167959056)

[1.7.1. Work Breakdown Structure 4](#_Toc167959057)

[1.7.2. Roles & Responsibility Matrix 5](#_Toc167959058)

[1.7.3. Gantt Chart 6](#_Toc167959059)

[1.8. Report Outline 6](#_Toc167959060)

[Chapter 2 7](#_Toc167959061)

[Software Requirement Specifications 7](#_Toc167959062)

[1.1. Introduction 8](#_Toc167959063)

[1.1.1. Purpose 8](#_Toc167959064)

[1.1.2. Document Conventions 8](#_Toc167959065)

[Bold font could emphasize keywords or important terms. Highlighted text might signify critical requirements or areas needing special attention. 8](#_Toc167959066)

[1.1.3. Intended Audience and Reading Suggestions 8](#_Toc167959067)

[1.1.4. Product Scope 9](#_Toc167959068)

[1.2. Overall Description 9](#_Toc167959069)

[1.2.1. Product Perspective 9](#_Toc167959070)

[1.2.2. User Classes and Characteristics 10](#_Toc167959071)

[1.2.3. Operating Environment 10](#_Toc167959072)

[1.2.4. Design and Implementation Constraints 11](#_Toc167959073)

[1.2.5. Assumptions and Dependencies 13](#_Toc167959074)

[1.3. External Interface Requirements 13](#_Toc167959075)

[1.3.1. User Interfaces 13](#_Toc167959076)

[1.3.2. Hardware Interfaces 14](#_Toc167959077)

[1.3.3. Software Interfaces 15](#_Toc167959078)

[1.3.4. Communications Interfaces 17](#_Toc167959079)

[1.4. System Features 17](#_Toc167959080)

[1.4.1. User Authentication 18](#_Toc167959081)

[1.4.1.1. Description and Priority 18](#_Toc167959082)

[1.4.1.2. Stimulus/Response Sequences 18](#_Toc167959083)

[1.4.1.3. Functional Requirements 18](#_Toc167959084)

[1.4.2. Area and Camera Management 18](#_Toc167959085)

[1.4.2.1. Description and Priority 18](#_Toc167959086)

[1.4.2.2. Stimulus/Response Sequences 18](#_Toc167959087)

[1.4.2.3. Functional Requirements 19](#_Toc167959088)

[1.4.3. Image Processing 20](#_Toc167959089)

[1.4.3.1. Description and Priority 20](#_Toc167959090)

[1.4.3.2. Stimulus/Response Sequences 20](#_Toc167959091)

[1.4.3.3. Functional Requirements 21](#_Toc167959092)

[1.4.4. Smart Board Control 21](#_Toc167959093)

[1.4.4.1. Description and Priority 21](#_Toc167959094)

[1.4.4.2. Stimulus/Response Sequences 21](#_Toc167959095)

[1.4.4.3. Functional Requirements 22](#_Toc167959096)

[1.4.5. IoT Platform 22](#_Toc167959097)

[1.4.5.1. Description and Priority 22](#_Toc167959098)

[1.4.5.2. Stimulus/Response Sequences 22](#_Toc167959099)

[1.4.5.3. Functional Requirements 22](#_Toc167959100)

[1.4.6. IoT Platform 23](#_Toc167959101)

[1.4.6.1. Description and Priority 23](#_Toc167959102)

[1.4.6.2. Stimulus/Response Sequences 23](#_Toc167959103)

[1.4.6.3. Functional Requirements 23](#_Toc167959104)

[1.4.7. Desktop Agent 24](#_Toc167959105)

[1.4.7.1. Description and Priority 24](#_Toc167959106)

[1.4.7.2. Stimulus/Response Sequences 24](#_Toc167959107)

[1.4.7.3. Functional Requirements 24](#_Toc167959108)

[1.5. Nonfunctional Requirements 24](#_Toc167959109)

[1.5.1. Performance Requirements 24](#_Toc167959110)

[3.1.1. Safety Requirements 25](#_Toc167959111)

[3.1.2. Security Requirements 25](#_Toc167959112)

[3.1.3. Usability Requirements 25](#_Toc167959113)

[3.1.4. Reliability Requirements 26](#_Toc167959114)

[3.1.5. Maintainability/Supportability Requirements 26](#_Toc167959115)

[3.1.6. Portability Requirements 26](#_Toc167959116)

[3.1.7. Efficiency Requirements 27](#_Toc167959117)

[3.2. Domain Requirements 27](#_Toc167959118)

[3.2.1. Database Requirements 27](#_Toc167959119)

[3.2.2. Legal Requirements 27](#_Toc167959120)

[Chapter 3 28](#_Toc167959121)

[Use Case Analysis 28](#_Toc167959122)

[3.1. Use Case Model 30](#_Toc167959123)

[3.2. Configure Region of Interest 31](#_Toc167959124)

[3.2.1 Configure Region of Interest 31](#_Toc167959125)

[3.3. Monitor History and Statistics 31](#_Toc167959126)

[3.3.1 Monitor History and Statistics 31](#_Toc167959127)

[3.4. Smart Board Control 32](#_Toc167959128)

[3.4.1 Smart Board Control 32](#_Toc167959129)

[3.5. Detection in ROI 33](#_Toc167959130)

[3.4.1 Detection in ROI 33](#_Toc167959131)

[3.6. PCs Shutdown 33](#_Toc167959132)

[3.4.1 PC Shutdown 33](#_Toc167959133)

[Chapter 4 35](#_Toc167959134)

[System Design 35](#_Toc167959135)

[4.1. Architecture Diagram 36](#_Toc167959136)

[4.2. Entity Relationship Diagram with data dictionary 37](#_Toc167959137)

[4.3. Class Diagram 39](#_Toc167959138)

[4.4. Sequence / Collaboration Diagram 40](#_Toc167959139)

[4.5. Activity Diagram 41](#_Toc167959140)

[4.6. State Transition Diagram 42](#_Toc167959141)

[4.7. Data Flow diagram 43](#_Toc167959142)

[Chapter 5 45](#_Toc167959143)

[Implementation 45](#_Toc167959144)

[5.1. Important Flow Control/Pseudo codes 46](#_Toc167959145)

[5.1.1. Website: 46](#_Toc167959146)

[5.1.2. Raspberry Pi: 46](#_Toc167959147)

[5.1.3. IoT Server: 47](#_Toc167959148)

[5.1.4. Smart Board: 48](#_Toc167959149)

[5.2. Components, Libraries, Web Services and stubs 48](#_Toc167959150)

[5.3. Deployment Environment 49](#_Toc167959151)

[5.4. Tools and Techniques 49](#_Toc167959152)

[5.5. Best Practices / Coding Standards 50](#_Toc167959153)

[5.6. Version Control 50](#_Toc167959154)

[Chapter 6 51](#_Toc167959155)

[Business Plan 51](#_Toc167959156)

[6.2 Market Analysis & Strategy 52](#_Toc167959157)

[6.3 Competitive Analysis 53](#_Toc167959158)

[6.4 Products/Services Description 53](#_Toc167959159)

[6.5 SWOT Analysis 54](#_Toc167959160)

[Chapter 7 56](#_Toc167959161)

[Testing & Evaluation 56](#_Toc167959162)

[7.1 Use Case Testing 57](#_Toc167959163)

[7.2 Equivalence Partitioning 60](#_Toc167959164)

[7.3 Boundary Value Testing 61](#_Toc167959165)

[7.4 Unit Testing 62](#_Toc167959166)

[7.5 Integration Testing 63](#_Toc167959167)

[Chapter 8 65](#_Toc167959168)

[Conclusion & Future Enhancements 65](#_Toc167959169)

[8.1 Achievements and Improvements 66](#_Toc167959170)

[8.2 Critical Review 66](#_Toc167959171)

[8.3 Lessons Learnt 67](#_Toc167959172)

[8.4 Future Enhancements/Recommendations 67](#_Toc167959173)

[Appendices 68](#_Toc167959174)

[Appendix A: Information / Promotional Material 69](#_Toc167959175)

# List of Figures

[Figure 1: Use Case for IoT Based Electricity Conservation System 30](#_Toc167959176)

[Figure 2: Architecture Diagram 36](#_Toc167959177)

[Figure 3: ER-Diagram 39](#_Toc167959178)

[Figure 4: Sequence Diagram 40](#_Toc167959179)

[Figure 5: Activity Diagram 41](#_Toc167959180)

[Figure 6: State Transition Diagram 42](#_Toc167959181)

[Figure 7: DFD 0 43](#_Toc167959182)

[Figure 8: DFD 1 44](#_Toc167959183)

# List of Tables

[Table 1: Configuration of ROI 31](#_Toc167959184)

[Table 2: User Login Function: Use Case 1 31](#_Toc167959185)

[Table 3: Monitor History and Statistics 31](#_Toc167959186)

[Table 4: Monitor History and Statistics Use case 1 32](#_Toc167959187)

[Table 5: Smart Board Control 32](#_Toc167959188)

[Table 6: Smart Board Control Use case 1 32](#_Toc167959189)

[Table 7: Detection in ROI 33](#_Toc167959190)

[Table 8: Detection in ROI Use case 1 33](#_Toc167959191)

[Table 9: PC Shutdown 33](#_Toc167959192)

[Table 10: PC Shutdown Use case 1 34](#_Toc167959193)

[Table 11: PC Shutdown use case 2 34](#_Toc167959194)

[Table 12: Area Data 37](#_Toc167959195)

[Table 13: Camera Data 37](#_Toc167959196)

[Table 14: Bounded Rectangle Data 37](#_Toc167959197)

[Table 15: Admin Data 38](#_Toc167959198)

[Table 16: Board Status Data 38](#_Toc167959199)

[Table 17: Energy Consumption Data 38](#_Toc167959200)

# List of Abbreviations

1.1 UML Unified Model Notation

1.2 SRE Software Requirement Engineering

2.1 SDR Software Defined Radios

2.2 ROI Region of Interest

2.3 IoT Internet of Things

2.4 HTTP Hyper Text Transfer Protocol

2.5 HTTPS Hyper Text Transfer Protocol Secure

2.6 PC Personal Computer

2.7 API Application Programming Interface

2.8 RTSP Real-Time Streaming Protocol

3.1 SQL Structured Query Language

3.2 SSMS SQL Server Management Studio

3.3 NPM Node Package Manager

4.1 GUI Graphical User Interface

4.2 DFD Data Flow Diagram

4.3 ER Entity Relationship

# 

# Chapter 1

# Introduction & Background

**Chapter 1: Introduction**

This document specifies the software requirements for the "IoT Based Electricity Conservation System." This system aims to address the prevalent issue of electricity wastage in buildings by implementing a real-time energy management solution. The scope of this document covers the first release of the system, which includes the integration of IoT-enabled devices, image processing for occupancy detection, and the automated control of electrical devices to optimize energy usage.

## Background

Electricity wastage is a significant concern on a global scale, with building being a major contributor. Unfortunately, these devices are often left running unattended, leading to significant energy consumption and increased operational costs for institutions.

Exacerbating the problem in Pakistan is the ongoing energy crisis. The country grapples with a significant electricity shortfall, placing a heavy burden on the already stressed power grid. Universities, with their frequently neglected equipment, contribute considerably to this energy waste. This situation necessitates a solution to address inefficient energy utilization in these critical research spaces. Addressing this issue, the "IoT Based Electricity Conservation System" aims to implement a real-time energy management solution. This system will leverage IoT-enabled devices and image processing techniques to automatically detect occupancy and manage electrical devices. By doing so, it ensures that energy is utilized efficiently, promoting sustainability and reducing electricity costs.

## Motivations and Challenges

The motivation for developing the "IoT Based Electricity Conservation System" arises from the need to curb electricity wastage, lower operational expenses, and contribute to environmental sustainability. Commercial areas, in particular, can significantly benefit from such a system due to their high energy consumption resulting from the continuous use of various electrical devices. However, several challenges must be addressed to implement this system effectively. These include ensuring accurate real-time occupancy detection, achieving seamless integration of IoT devices, and maintaining reliable and secure communication between hardware and software components. Overcoming these challenges is essential to achieve the desired energy savings and operational efficiency.

## Goals and Objectives

The primary goal of the "IoT Based Electricity Conservation System" is to minimize electricity wastage in commercial areas by automating the control of electrical devices based on real-time occupancy data. The project's objectives include developing an IoT platform for real-time occupancy detection using cameras and image processing algorithms, integrating smart switches to control electrical devices, creating a user-friendly interface for system configuration, and developing a desktop agent to control the PCs and monitoring. Successfully achieving these objectives will result in significant energy savings, reduced operational costs, and enhanced control over buildings.

## Literature Review/Existing Solutions

Efforts to address electricity wastage have yielded a range of solutions, from basic manual practices to more sophisticated automated systems. Traditional methods rely on user behavior, encouraging individuals to manually switch off equipment when not in use. However, this approach is prone to human error and inconsistency.

Technological advancements have led to the development of automated systems utilizing sensors and timers. These systems often employ basic occupancy sensors that detect motion within a space and automatically control lighting and HVAC systems based on occupancy status. While effective in some scenarios, these existing solutions may lack the granularity and automation required for comprehensive energy management in complex buildings.

|  |  |  |  |
| --- | --- | --- | --- |
| **Solution Tyle** | **Description** | **Advantages** | **Disadvantages** |
| Manual Practices | Relies on individuals to manually switch off equipment when not in use. | Simple and low-cost implementation. | Prone to human error and inconsistency; dependent on user behavior. |
| Automated Sensors | Utilizes basic occupancy sensors to detect motion within a space and control lighting and HVAC systems. | Reduces dependency on human intervention; improves consistency. | Limited granularity; may not account for all variables in complex environments. |
| Timer-Based Systems | Uses timers to switch off equipment after a set period of inactivity. | Reduces wastage by automating power-off schedules. | Fixed schedules may not align with actual usage patterns. |
| Advanced Sensors | Employs more sophisticated sensors and algorithms to manage power based on detailed occupancy data. | Higher granularity and precision in energy management. | Often more expensive and complex to implement. |

## Gap Analysis

Despite the availability of various energy management solutions, gaps remain in terms of accuracy, reliability, and user-friendliness. Many existing systems rely on basic motion sensors, which can lead to false positives or negatives, resulting in inefficient energy management. Additionally, integrating these systems with existing infrastructure can be challenging, and they often lack advanced features like real-time monitoring and control via a user-friendly interface

## Proposed Solution

The "IoT Based Electricity Conservation System" offers a comprehensive solution to tackle electricity wastage in commercial areas. This system employs IoT cameras to capture real-time occupancy data, which is processed using image processing algorithms to accurately determine occupancy status. Based on this data, smart switches control electrical devices, turning them off when no occupancy is detected and back on when the space is occupied. In addition to that, the desktop agent automatically shouts down the PCs to avoid HDD damage and data loss. The system includes a web-based interface for easy configuration, monitoring, and control, ensuring users can manage the system efficiently. Additionally, robust security measures are incorporated to protect data and ensure only authorized users have access.

## Project Plan

The project plan for the "IoT Based Electricity Conservation System" outlines the necessary steps and structure to achieve the project's goals.

## Work Breakdown Structure

The project utilized a Work Breakdown Structure to decompose the development process into manageable tasks, ensuring a well-defined roadmap for project execution. The Work Breakdown Structure encompassed several key phases:

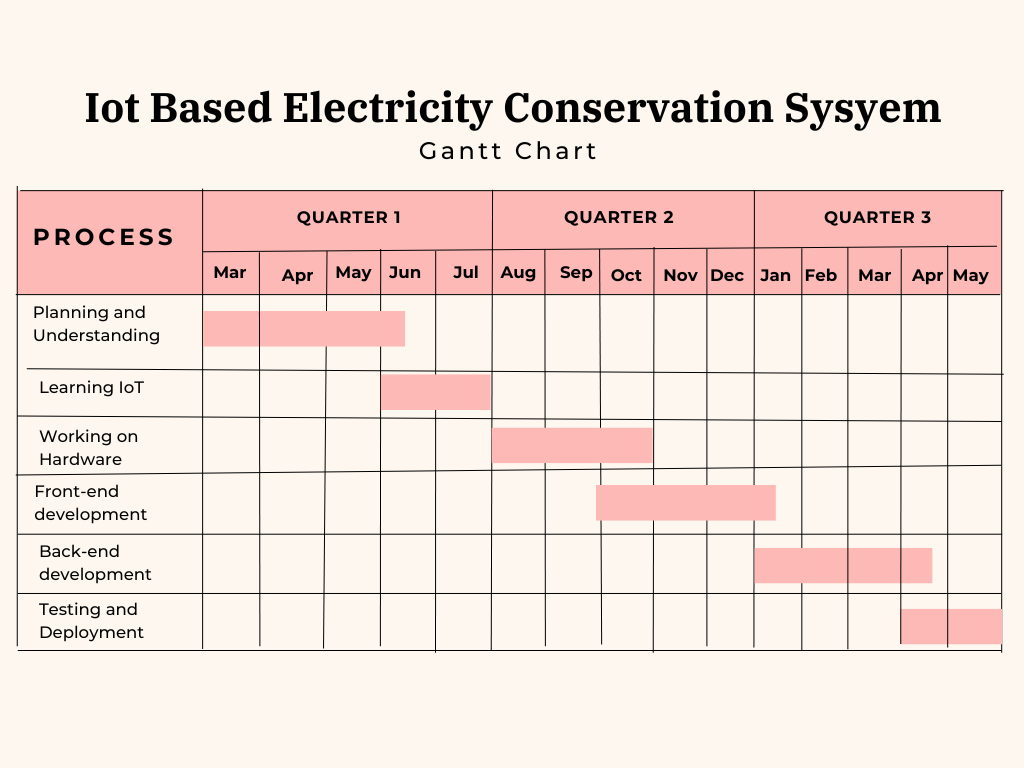
* **Research and Development:** An initial phase involved in-depth research on existing energy conservation solutions for buildings. This research identified relevant technologies and best practices to inform the development process.
* **Software Development:** This phase focused on creating two software components. First, a web application was developed to empower administrators with functionalities like configuring camera settings, and defining specific regions of interest (ROIs) within the an area for occupancy detection. Second, an IoT platform was developed to act as the central server, handling data processing and communication with all connected devices within the system.
* **Hardware Integration:** This phase concentrated on integrating the hardware components of the system. High-resolution cameras were configured to capture clear images within an area. The core functionality involved implementing digital image processing algorithms on a Raspberry Pi, which constructed an IoT Box. These algorithms were specifically designed to detect human presence only, minimizing false positives triggered by movement from non-human sources.
* **Smart Board Development:** This phase focused on the development of the smart board, a crucial component responsible for controlling power based on occupancy information. The smart board was programmed with logic to receive occupancy data from the server and utilize it to turn power on or off within the building.
* **System Testing:** Following the development of individual components, a comprehensive testing phase was conducted. This phase involved three key stages. First, each component – the IoT platform, IoT box (Raspberry Pi), and smart board – was tested independently to ensure proper functionality. Second, a local network was established to facilitate seamless integration of all components. Finally, rigorous testing and debugging of the entire integrated system was conducted to identify and rectify any potential issues before deployment.

## Roles & Responsibility Matrix

The project leveraged a Roles and Responsibility Matrix to ensure clear delineation of tasks and accountability throughout the development process. This matrix identified the following key stakeholders and their contributions:

* Azfar Tariq and Ahmad Imran were responsible for the development, individual testing, and debugging of the core system components: the IoT platform and the web application. Their efforts ensured the smooth operation and functionality of these critical elements.
* Junaid Nazir's focus was on the development of the IoT box (Raspberry Pi) software. He handled the development, individual testing, debugging, and deployment of the code on this device. Additionally, he took the initiative to develop a desktop agent for personal computers, further expanding the system's functionality.
* The entire development team, with significant help and guidance from the project supervisor, collaborated on the development, testing, and debugging of the smart board. This collaborative approach ensured a comprehensive and well-functioning control unit.
* The responsibility of integrating the IoT platform, IoT box, and smart board through a local network fell upon the entire team. This collaborative effort ensured seamless communication and data flow between the system's components. Following integration, the team conducted comprehensive testing to identify and rectify any potential issues before deployment.

## Gantt Chart



## Report Outline

The final report provides a comprehensive overview of the "IoT Based Electricity Conservation System," detailing each phase of the project from inception to integration. It includes an introduction to the problem of electricity wastage, the motivations and challenges faced, the goals and objectives, a review of existing solutions, gap analysis, the proposed solution, test cases, testing, user interfaces, functional and non-functional requirements. The project plan will be elaborated upon, detailing the work breakdown structure, roles and responsibility matrix, and Gantt chart. The report will conclude with a summary of findings, implications for energy management in commercial areas, and recommendations for future work and improvements.

# Chapter 2

# Software Requirement Specifications

**Chapter 2:** Software Requirement Specifications

## Introduction

## Purpose

The purpose of this document is to specify the requirements for the development of the "IoT Based Electricity Conservation System." This system aims to intelligently manage electrical equipment in buildings through the integration of Internet of Things (IoT) and image processing technologies. By identifying the different types of readers, including developers, users, and documentation writers, this document aims to provide a comprehensive understanding of the system's functionality, constraints, and interfaces.

## Document Conventions

## Bold font could emphasize keywords or important terms. Highlighted text might signify critical requirements or areas needing special attention.

## Intended Audience and Reading Suggestions

This document is intended for various stakeholders involved in the lifecycle of the project. The primary readers include:

**Developers**: To gain a detailed understanding of the system's functional and non-functional requirements, architecture, and technical specifications.

**Users**: To understand the capabilities and limitations of the system, facilitating a better user experience and expectations alignment.

**Documentation Writers**: To aid in the development of user manuals, technical documentation, and training materials.

By addressing the needs of these diverse readers, this document seeks to ensure a comprehensive and accurate interpretation of the "IoT Based Electricity Conservation System" throughout its development lifecycle.

## Product Scope

The "IoT Based Electricity Conservation System" is a proposed solution designed to intelligently manage electrical equipment in buildings. The primary purpose of the system is to enhance energy efficiency by dynamically controlling the power supply to computers based on the occupancy status detected through IoT and image processing technologies.

The proposed solution addresses the need for an automated and intelligent approach to electricity conservation in commercial areas. By leveraging IoT-enabled cameras and sophisticated image processing algorithms, the system aims to reduce energy wastage, optimize resource utilization, and contribute to a sustainable and cost-effective operational environment.

## Overall Description

## Product Perspective

This document outlines the System Requirements Specification (SRS) for a newly developed, self-contained IoT-Based Electricity Conservation System designed specifically for universities. The system tackles electricity wastage by leveraging several key components working in concert.

A high-resolution camera captures the building space, sending live video feeds as images to an IoT platform that acts as both server and website. This web interface empowers administrators to manage areas, configure camera settings, and define crucial Regions of Interest (ROIs) within the lab for occupancy detection. These ROIs are user-defined areas where the system focuses its image processing efforts.

An IoT box, typically a Raspberry Pi, receives these image feeds and ROI coordinates from the platform. It then implements digital image processing algorithms specifically on the defined ROIs to detect human presence and minimize false positives triggered by non-human movement. The processed results, indicating human presence or absence, are then sent back to the IoT platform.

Based on this occupancy information, the IoT platform instructs a smart board to control the power supply to building equipment and electrical appliances. If human presence is detected, the power remains on. Conversely, if no presence is detected for a set duration, the power is switched off.

A Desktop Agent is installed on individual PCs within the lab. This agent offers both manual and automatic operation modes. In manual mode, the PC remains powered on regardless of occupancy detection within the ROI. In automatic mode, the PC powers off based on the occupancy information received from the IoT platform.

Through this interplay of components, the system delivers a comprehensive solution for automatic energy management in commercial areas, promoting efficient electricity usage and cost savings.

## User Classes and Characteristics

There are no classes in implementation.

## Operating Environment

**OE-1: Hardware Requirements**

The system requires the following hardware components:

* **A high-resolution camera** with a minimum resolution of 720p for capturing clear images for occupancy detection.
* **IoT Box which consists of a** device like a Raspberry Pi 3 Model B or later to handle image processing.
* **Smart Boards for** electrical outlets.

**OE-2: Software Environment**

* Raspberry Pi OS (latest stable release)
* **Image Processing Library like Yolo 5 and OpenCV**
* Web server operating systems, including Linux (Ubuntu 18.04 LTS), Apache (version 2.4), and Nginx (version 1.16)

**OE-3**: The System shall be accessible from standard web browsers, including but not limited to:

* Google Chrome (latest version)
* Mozilla Firefox (latest version)
* Apple Safari (latest version)
* Microsoft Edge (latest version)
* Opera (latest version)

**OE-4:** A stable local network connection within the building is essential to facilitate communication between all system components.

**OE-5**: The System shall operate effectively in buildings with varying lighting conditions and configurations.

**OE-6**: The System shall provide documentation and support for organizations integrating the solution, including educational institutions, technology providers, and hosting partners.

## Design and Implementation Constraints

**CON-1**: The image processing algorithms for occupancy detection shall be implemented using Python and OpenCV:

**Rationale**: Python offers a rich set of libraries and is widely used for computer vision applications. OpenCV is a robust and well-established library specifically designed for image processing, making it suitable for the project's requirements.

**CON-2:** The Desktop Agent is developed using a programming language compatible with various operating systems.

**Rationale:** Python is a widely used and versatile programming language well-suited for developing user interfaces. It offers a rich set of libraries for creating desktop applications with a clean and intuitive interface for PC users.

**CON-3**: The web-based user interface shall be developed using React:

**Rationale:** React is industry-standard technology that ensure cross-browser compatibility and responsiveness. It provides a versatile and widely supported foundation for creating interactive and dynamic user interfaces.

**CON-4:** Node.js and Express.js will be used to develop the backend server-side logic of the IoT platform.

**Rationale:** This combination leverages JavaScript (Node.js) for efficient real-time development and simplifies web application creation with Express.js. It provides a robust foundation for core functionalities like data processing, communication, and user management.

**CON-5:** The System shall use the latest stable release of the Raspberry Pi OS for Raspberry Pi devices:

**Rationale:** Ensuring compatibility with the latest stable release of the Raspberry Pi OS guarantees access to the most recent features, optimizations, and security updates, enhancing the overall performance and reliability of the system.

**CON-6**: The Smart Board must be compatible with the communication protocols supported by the Raspberry Pi:

**Rationale:** To enable seamless communication between the IoT Platform, IoT Box, the Desktop Agent, and the Smart Board, compatibility with supported local network is necessary. This constraint ensures efficient integration and control of the power supply.

**CON-7**: The web server hosting the user interface shall use HTTPS for secure communication:

**Rationale:** Implementing HTTPS ensures the secure transmission of data between the user's device and the server, protecting sensitive information and preventing potential security vulnerabilities.

## Assumptions and Dependencies

The successful implementation of this system relies on several key assumptions and dependencies. Firstly, we assume a stable network connection within the building to facilitate communication between system components (cameras, IoT Box, IoT platform, smart boards). Secondly, we depend on the availability of a compatible power distribution system within the area to enable the smart boards to effectively control power supply to various appliances. These assumptions and dependencies will be validated during system deployment and addressed if any discrepancies arise.

## External Interface Requirements

## 

## User Interfaces

1. **Visual Identity**

• Color Scheme: Utilize a visually calming color scheme to enhance the user experience.

• Icons and Images: Implement clear and universally understandable icons and images for intuitive navigation.

1. **Layout and Responsiveness**

• Responsive Design: Ensure that the user interface is responsive to different screen sizes and resolutions.

• Screen Layout: Design and organized layout for optimal user engagement, with focus on usability.

1. **Navigation**

• Standard Navigation Elements: Define consistent navigation elements, such as buttons and links, across all screens.

1. **User Assistance**

• Error Messages: Implement a clear and concise error messages to guide users in case of issues.

• Notification: Provide informative notifications for system events and updates.

## Hardware Interfaces

* **Camera**
* **Device Type:** High Resolution cameras capable of capturing images.
* **Data Interactions:** Captures images for occupancy detection in the building.
* **Control Interactions:** No direct control interactions; operates independently and transmits image data to the IoT Platform for processing.
* **Communication Protocols:** Utilizes standard network protocols (e.g., HTTP/HTTPS) for data transmission to the Raspberry Pi.
* **IoT Box**
* **Device Type:** Raspberry Pi
* **Data Interactions:** Receives image data from camera and Region of Interest from IoT Platform for occupancy detection. Transmits occupancy status to IoT Platform.
* **Control Interactions:** Controls image detection and manages data transmission to IoT Platform.
* **Communication Protocols:** Uses standard communication protocols (e.g., HTTP/HTTPS) for interfacing with IoT cameras and transmitting data to smart electrical outlets.
* **Smart Board**
* **Device Type:** Smart electrical outlets with controllable on/off functionality.
* **Data Interactions:** Receives occupancy status from the IoT Platform.
* **Control Interactions:** Controls the on/off state based on occupancy status received from the IoT Platform.
* **Communication Protocols:** Utilizes communication protocols compatible with the smart electrical outlets, such as Wi-Fi or Bluetooth.

## Software Interfaces

**SI-1.1: NodeJS & ExpressJS**

* The IoT Based Electricity Conservation System shall communicate occupancy data from the IoT Box, make decisions based on the occupancy data, and communicate with the Smart Board on the IoT Platform which is developed in NodeJS & Expressjs server through a programmatic interface.

**SI-1.2: SQL Server**

* The Express backend server shall store and retrieve data from the SQL Server database using SQL queries facilitated by SSMS (SQL Server Management Studio).

**SI-1.3: React JS**

* The React JS frontend shall interact with the IoT Platform through RESTful APIs for real-time data updates and user interactions.

**SI-1.4: Python**

* The Python-based image detection program running on IoT Box shall process captured images and transmit occupancy status to the IoT Platform through a programmatic interface.

**SI-1.5: Packages and Libraries**

* NPM packages shall be used for managing dependencies in both the React JS frontend and NodeJS & ExpressJS.
* OpenCV shall be integrated for efficient development of image detection algorithms on the Raspberry Pi.

## Communications Interfaces

**CI-1.1: Camera-to-IoT Box Communication**

The camera sends live feed to the IoT Box using RTSP communication protocols.

**CI-1.2: IoT Platform-to-IoT Box Communication**

The IoT Platform sends the Region of Interest to the IoT Box using HTTP/HTTPS communication protocols

**CI-1.3: IoT Box-to-IoT Platform Communication**

The system transmits occupancy status updates from the IoT Box to the IoT Platform using HTTP/HTTPS communication protocols.

**CI-1.4: Web Server-to-IoT Platform Communication**

The React JS frontend shall interact with the IoT Platform using RESTful APIs, utilizing HTTP/HTTPS for real-time data updates and user interactions.

**CI-1.5: IoT Platform-to-Smart Board Communication**

The IoT Platform communicates with Smart Board using HTTP/HTTPS communication protocols.

**CI-1.6: IoT Platform-to-Desktop Agent Communication**

The IoT Platform communicates with Desktop Agent using HTTP/HTTPS communication protocols

## System Features

"System Features," will outline the key functionalities and capabilities offered by the IoT-based electricity conservation system. This section will serve as a roadmap, detailing what the system can do to achieve efficient energy management in buildings.

## User Authentication

## Description and Priority

This feature is of high priority and ensures secure access to the system for authorized administrators. Users can log in to the web application's admin mode using valid credentials, restricting access to sensitive functionalities.

## Stimulus/Response Sequences

* User opens the system login page.
* System displays username and password input fields.
* User enters a valid username and password.
* User submits the login credentials.
* Valid Credentials: System authenticates the user and grants access to the admin interface.
* Invalid Credentials: System displays an error message indicating invalid login attempt.

## Functional Requirements

**REQ-SF1-1:** The system shall require users to enter a username and password for login.

**REQ-SF1-2:** The system shall authenticate user credentials against a secure user database.

**REQ-SF1-3:** The system shall grant access to the admin interface only upon successful authentication.

**REQ-SF1-4:** The system shall display an error message with appropriate details for invalid login attempts (e.g., incorrect username, password mismatch).

## Area and Camera Management

## Description and Priority

This feature with a high priority allows administrators to manage areas and cameras within the system through the web app. Administrators can create, edit, and delete areas and cameras to adapt the system to changing infrastructure and ensure efficient space management

## Stimulus/Response Sequences

* Administrator accesses the Area/Camera Management section of the web app.
* System displays a list of existing areas and their associated cameras (if any).
* **Create Area:**
* Administrator selects "Create Area."
* System displays a form for entering area details (e.g., area name, location).
* Administrator enters area details and submits the form.
* System creates the new area and displays it in the list.
* **Edit Area:**
* Administrator selects an existing area from the list.
* System displays the area details for editing.
* Administrator edits the details and submits the changes.
* System updates the area information and reflects the changes in the list.
* **Delete Area:**
* Administrator selects an existing area from the list.
* System prompts for confirmation to delete the area (including associated cameras, if any).
* Administrator confirms deletion.
* System deletes the area and its associated cameras from the system.
* **Manage Cameras:**
* Administrator selects an existing area from the list.
* System displays a list of cameras associated with that area (if any).
* Administrator can add, edit, or delete individual cameras within the area using similar actions as for managing areas.

.

## Functional Requirements

**REQ-AFM-1:** The system shall allow administrators to create new areas by specifying details like area name and location.

**REQ-AFM-2:** The system shall allow administrators to edit existing area details.

**REQ-AFM-3:** The system shall update the area list after any creation or editing of area information.

**REQ-AFM-4:** The system shall prompt for confirmation before deleting an area.

**REQ-AFM-5:** The system shall delete an area and its associated cameras upon administrator confirmation.

**REQ-AFM-6:** The system shall allow administrators to manage cameras within an area, including adding, editing, and deleting individual cameras. (Details of camera management functionalities can be specified in a separate section):

## Image Processing

## Description and Priority

This feature with high priority addresses the integration of the IoT box with the lab camera and leverages image processing for occupancy detection. The system shall display a live video stream from the camera and utilize image processing algorithms on the IoT box to detect human presence within designated Regions of Interest (ROIs). Real-time processing and monitoring are crucial for the system to make accurate decisions regarding power switch control based on occupancy detection within the ROIs.

## Stimulus/Response Sequences

* The system establishes a connection between the IoT Box and the camera.
* The system initiates a live video stream from the camera and displays it on the administrator interface when the system first starts.
* The administrator configures ROIs within the camera view for occupancy detection.
* The IoT box continuously receives live video frames from the camera.
* The IoT box applies image processing algorithms to the received stream, focusing on the defined ROIs.
* The image processing algorithms detect human presence within the ROIs.
* **Occupancy Detected:**
* The IoT box transmits an "occupancy detected" status to the IoT Platform.
* The IoT Platform receives and triggers appropriate actions
* **No Occupancy Detected:**
* The IoT box transmits a "no occupancy detected" " status to the IoT Platform.

## Functional Requirements

**REQ-LSIP-1:** The system shall establish a connection between the IoT box and the lab camera to access the live video stream.

**REQ-LSIP-2:** The system shall display a real-time image from the camera on the administrator when the system is first started.

**REQ-LSIP-3:** The system shall utilize image processing algorithms on the IoT box to detect human presence within defined ROIs.

**REQ-LSIP-4:** The IoT box shall continuously analyze live video frames for occupancy detection.

**REQ-LSIP-5:** The system shall transmit occupancy detection signals ("occupancy detected" or "no occupancy detected") from the IoT box to the IoT Platform for triggering appropriate actions.

## Smart Board Control

## Description and Priority

This feature is central to the system's functionality and is of high priority. It enables automated control of smart boards based on human presence detection within designated Regions of Interest (ROIs). This allows for dynamic power management, ensuring energy is used efficiently and consumption is reduced when areas are unoccupied, promoting electricity conservation.

## Stimulus/Response Sequences

* The IoT Platform receives occupancy detection signals from the IoT box
* The IoT Platform processes the "occupancy detected" status and maintains the current power state of the smart board (remains on).
* A timer is reset for the occupancy timeout period.
* The IoT Platform receives a "no occupancy detected" status from IoT Box.
* The IoT Platform initiates a power-saving action by sending a control signal to the smart board.
* The smart board responds to the status and switches off the connected appliance

## Functional Requirements

**REQ-SBC-1:** The backend server shall receive occupancy status from the IoT box.

**REQ-SBC-2:** The system shall maintain the current power state of the smart board when occupancy is detected within the ROI.

**REQ-SBC-3:** IoT Platform shall initiate a power-saving action by sending a control signal to the smart board.

**REQ-SBC-4:** The smart board shall respond to the status from the IoT Platform and switch off the connected appliance.

## IoT Platform

## Description and Priority

This requirement ensures seamless communication and data exchange between the core functionalities of the system and the chosen IoT platform and is of high priority. The IoT platform will serve as a central hub for device management, data collection, and communication protocols, enabling efficient system operation.

## Stimulus/Response Sequences

* The IoT Platform establishes a secure connection with the IoT Box.
* The IoT box transmits data from various system components to the IoT platform
* The IoT platform sends and receives relevant data to system operations, including configuration settings for areas, cameras, ROIs, occupancy status from IoT Box and the mode, analysis of energy consumption

## Functional Requirements

**REQ-IPI-1:** The IoT Platform shall establish a secure connection with the chosen IoT Box using appropriate protocols.

**REQ-IPI-2:** The IoT box shall transmit data from various system components to the IoT platform in a defined format.

**REQ-IPI-4:** The system shall leverage the IoT platform for device management of the IoT box and smart boards.

**REQ-IPI-5:** The system shall utilize the IoT platform for secure data storage and retrieval of analysis of energy consumption analysis.

## IoT Platform

## Description and Priority

This feature empowers administrators to analyze historical data and trends related to energy consumption events within the system. The web app interface will provide visualizations and reports, enabling insights into energy usage patterns and system performance over extended periods. This analysis functionality supports data-driven decision-making and helps administrators evaluate the long-term impact of the electricity conservation system on energy efficiency within the specific area.

## Stimulus/Response Sequences

* Administrator accesses the Energy Consumption Analysis section of the web app.
* System displays options to filter and visualize historical data
* System displays charts, graphs, or reports representing energy consumption trends.

## Functional Requirements

**REQ-ECA-1:** The system shall allow administrators to access historical data related to energy consumption events.

**REQ-ECA-3:** The system shall display visualizations (charts, graphs, reports) of energy consumption trends based on the applied filters.

**REQ-ECA-4:** The system shall provide insights into long-term energy usage patterns.

## Desktop Agent

## Description and Priority

This section outlines the system's functionality for managing PC power states based on occupancy detection, utilizing a Desktop Agent developed using Python. The Desktop Agent provides a user interface and facilitates communication with the IoT Platform. It is of high priority

## Stimulus/Response Sequences

* The Desktop Agent is installed and running on individual PCs within the lab environment.
* The Desktop Agent establishes a connection with the IoT Platform.
* The Desktop Agent receives the status from IoT Platform and does not shut down the PC if occupancy is detected.
* The Desktop Agent receives the status from IoT Platform and shuts down the PC if no occupancy is detected.
* The Desktop Agent sends manual or auto mode to the IoT Platform.

## Functional Requirements

**REQ-PCPM-1:** The Desktop Agent shall establish a connection with the backend server to receive power management instructions.

**REQ-PCPM-2:** The Desktop Agent receives the status from IoT Platform and does not shut down the PC if occupancy is detected.

**REQ-PCPM-3:** The Desktop Agent receives the status from IoT Platform and shuts down the PC if no occupancy is detected.

**REQ-PCPM-4:** The Desktop Agent sends manual or auto mode to the IoT Platform.

## Nonfunctional Requirements

## Performance Requirements

1. PER-1: Response Time

The system responds to user interactions within seconds to ensure a seamless user experience.

1. PER-2: Throughput

The system supports multiple simultaneous user logins without degradation in performance.

## Safety Requirements

* **SFT-1: Safety Regulations**

The system shall comply with relevant safety regulations and standards for electrical equipment and communication protocols.

* **SFT-2: Authorized Access**

The system shall implement secure authentication and authorization mechanisms to prevent unauthorized access and control of devices.

## Security Requirements

* SEC-1: Data Encryption

All communication between the system components, including the web interface, Raspberry Pi, and database, have been encrypted using industry-standard protocols.

* SEC-2: Access Control

The system has implemented authentication and authorization, ensuring that only authorized users have access to specific functionalities.

## Usability Requirements

* USE-1: User Interface Intuitiveness

The web interface is designed with a user-friendly layout and intuitive navigation to facilitate ease of use.

* USE-2: Documentation

Comprehensive documentation shall be provided to assist users and administrators in understanding system functionalities and configurations.

## Reliability Requirements

* REL-1: Up Time

The system shall maintain a high level of uptime, minimizing downtime due to system failures or errors.

* REL-2: Integrity

The system shall be resilient to network outages and ensure data integrity during such events.

* REL-3: Error Detection

The system shall provide mechanisms for error detection, logging, and reporting to facilitate troubleshooting and system maintenance.

## Maintainability/Supportability Requirements

* **MSR-1: Modular Components**

The deployed system is designed with modular components for ease of maintenance, troubleshooting, and future upgrades.

* **MSR-2: Diagnostic Tools**

The system provides comprehensive logging and diagnostic tools to assist administrators in identifying and resolving system issues.

* **MSR-3: Comprehensive**

The system documentation shall be comprehensive and up-to-date, including detailed instructions for system configuration, maintenance procedures, and user support.

## Portability Requirements

* **PORT-1: Portable**

The system, is designed with a flexible architecture to allow for potential future integration with different IoT platforms or hardware components.

## Efficiency Requirements

* **Eff-1:** Optimize Resources

The deployed system optimizes resource utilization on the IoT box and IoT Platform to minimize processing overhead and ensure efficient system operation.

* **Eff-2: Efficiency**

The system is designed for energy efficiency, considering factors like power consumption of the IoT box, communication protocols, and potential for optimizing smart board power usage based on occupancy detection.

## Domain Requirements

This section captures additional requirements specific to the IoT-based electricity conservation system that haven't been covered in previous sections.

## Database Requirements

* **DB-1:** Relational Database

The system shall utilize a relational database management system (RDBMS) to store data

* **DB-2:** Security Measures

The system shall implement data security measures for the database.

## Legal Requirements

* **LR-1: Privacy Regulations**

The system shall comply with all relevant data privacy regulations) concerning user data collection, storage, and access.

* **LR-2:** Ownership Access

The system shall clearly define data ownership and access rights within the user agreement.

# Chapter 3

# Use Case Analysis

**Chapter 3:** Use Case Analysis

This chapter dives into the functionalities of the IoT-based electricity conservation system by exploring various use cases. We will analyze how the system interacts with the administrator and how it delivers value by automating power management in buildings. Through a series of use case descriptions and their corresponding stimulus-response sequences, we will gain a clear understanding of the system's behavior and the interactions it facilitates.

## Use Case Model

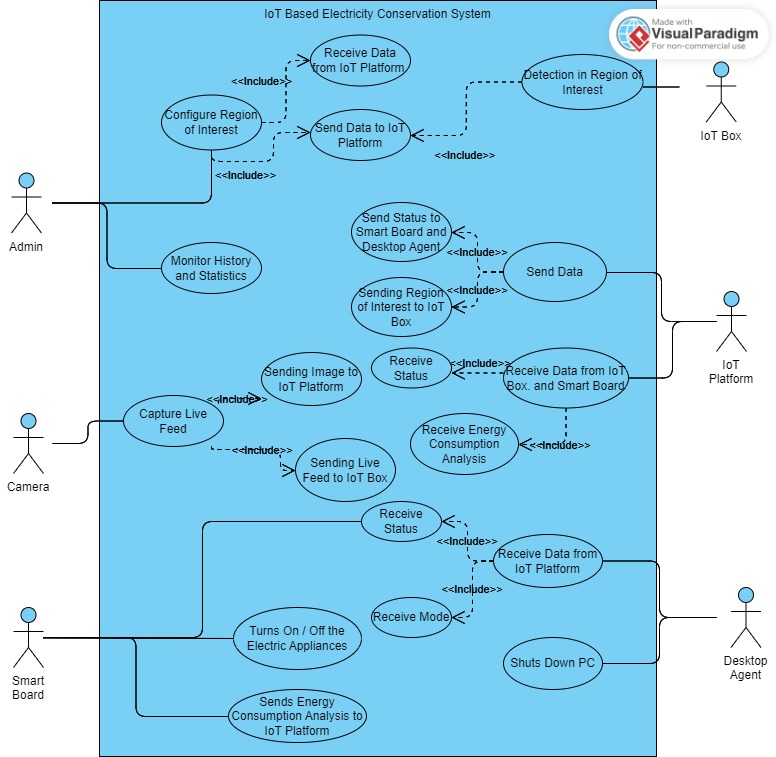


Figure 1: Use Case for IoT Based Electricity Conservation System

## Configure Region of Interest

Admin can add Region of Interest where the IoT Box will apply digital image processing to detect human on the selected camera.

## 3.2.1 Configure Region of Interest

|  |  |
| --- | --- |
| **Title** | Allow admin to add Region of Interest |
| **Requirement** | User must be logged in as Admin |
| **Rational** | Add ROI |
| **Restriction or Risk** | N/A |
| **Dependency** | Internet, PC on local network |
| **Priority** | Safety, timing |

Table 1: Configuration of ROI

**Use case 1**

|  |
| --- |
| **Configure Region of Interest** |
| **Actor** |
| * Admin |
| **Preconditions** |
| * User must be logged in as Admin |
| **Basic flow** |
| * Admin wants to add Region of Interest |
| **Alternate flows** |
| * N/A |
| **Post Condition** |
| * N/A |

Table 2: User Login Function: Use Case 1

## Monitor History and Statistics

Admin monitor statistics on the dashboard. The statistics show energy consumption and are generated from the Smart Board.

## 3.3.1 Monitor History and Statistics

|  |  |
| --- | --- |
| **Title** | Monitor History and Statistics |
| **Requirement** | User must be logged in as Admin |
| **Rational** | Analyze the energy consumption. |
| **Restriction or Risk** | N/A |
| **Dependency** | Internet, PC on local network |
| **Priority** | Safety, timing |

Table 3: Monitor History and Statistics

|  |
| --- |
| **Monitor History and Statistics** |
| **Actor** |
| * Admin |
| **Preconditions** |
| * User must be logged in as Admin |
| **Basic flow** |
| * Admin wants to see energy consumption statistics |
| **Alternate flows** |
| * N/A |
| **Post Condition** |
| * N/A |

Table 4: Monitor History and Statistics Use case 1

## Smart Board Control

Turn on or off the smart board based on status received from the IoT box.

## 3.4.1 Smart Board Control

|  |  |
| --- | --- |
| **Title** | Smart Board Control |
| **Requirement** | Smart Board must be associated with a ROI |
| **Rational** | Turn on or off the smart board |
| **Restriction or Risk** | N/A |
| **Dependency** | Internet, PC on local network |
| **Priority** | Safety, timing |

Table 5: Smart Board Control

|  |
| --- |
| **Smart Board Control** |
| **Actor** |
| * IoT Platform, IoT Box |
| **Preconditions** |
| * Smart Board must be associated with a ROI and with the IoT Platform |
| **Basic flow** |
| * Smart Switch must receive status from IoT Platform |
| **Alternate flows** |
| * N/A |
| **Post Condition** |
| * Switches Turn on or off |

Table 6: Smart Board Control Use case 1

## Detection in ROI

The human detection will take place inside the ROI at the IoT Box.

## 3.4.1 Detection in ROI

|  |  |
| --- | --- |
| **Title** | Detection in ROI |
| **Requirement** | IoT Box must be connected with camera and IoT Platform |
| **Rational** | Perform digital image processing |
| **Restriction or Risk** | N/A |
| **Dependency** | Camera, Raspberry Pi, IoT Box |
| **Priority** | Safety, timing |

Table 7: Detection in ROI

|  |
| --- |
| **Detection in ROI** |
| **Actor** |
| * IoT Box, Camera, IoT Platform |
| **Preconditions** |
| * IoT Box must be connected with camera and IoT Platform |
| **Basic flow** |
| * Human detection take place |
| **Alternate flows** |
| * N/A |
| **Post Condition** |
| * Status sent to IoT Platform |

Table 8: Detection in ROI Use case 1

## PCs Shutdown

Automatically shut down PCs

## 3.4.1 PC Shutdown

|  |  |
| --- | --- |
| **Title** | PC shutdown |
| **Requirement** | PCs must be associated with ROI |
| **Rational** | Shut down the PC |
| **Restriction or Risk** | N/A |
| **Dependency** | Internet, PC on local network, desktop agent |
| **Priority** | Safety, timing |

Table 9: PC Shutdown

|  |
| --- |
| **PC Shutdown** |
| **Actor** |
| * IoT Platform, IoT Box, PC |
| **Preconditions** |
| * PCs must be associated with a ROI and with the IoT Platform |
| **Basic flow** |
| * PCs must receive status from IoT Platform |
| **Alternate flows** |
| * N/A |
| **Post Condition** |
| * Switches shutdown if mode is auto |

Table 10: PC Shutdown Use case 1

|  |
| --- |
| **PC Shutdown** |
| **Actor** |
| * IoT Platform, IoT Box, PC |
| **Preconditions** |
| * PCs must be associated with a ROI and with the IoT Platform |
| **Basic flow** |
| * PCs must receive status from IoT Platform |
| **Alternate flows** |
| * N/A |
| **Post Condition** |
| * Switches stay on if mode is manual |

Table 11: PC Shutdown use case 2

# Chapter 4

# System Design

**Chapter 4:** System Design

This chapter delves into the intricate design and architecture of the IoT-based electricity conservation system. Here, we explore the sequence and flow of the system that works seamlessly to achieve efficient energy management in buildings, along with the database schema.

## Architecture Diagram

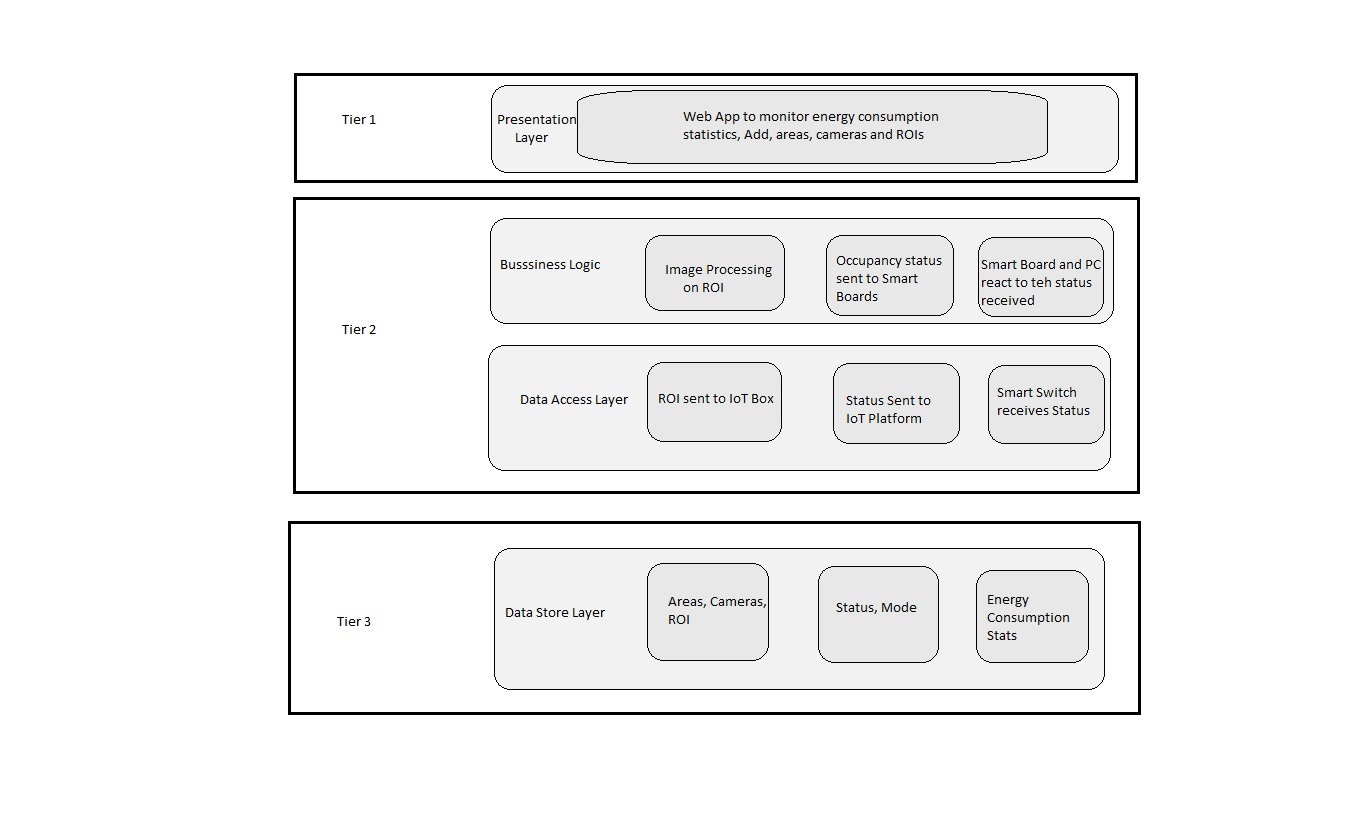


Figure 2: Architecture Diagram

## Entity Relationship Diagram with data dictionary

**Area Data:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| AreaName | nvarchar | Name of Area |
| AreaID | int (Primary Key) | Id of Building |
| Description | Nvarchar | Description about the area |
| Address | Nvarchar | Address of the Area |
| FocalPerson | Nvarchar | Name of Lab Administrator |
| Contact | BigInt | Contact number of Lab Administrator |

Table 12: Area Data

**Camera Data:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| AreaID | int (Foreign Key) | Id of Building |
| CameraID | int (Primary Key) | Id of Lab |
| CameraName | nvarchar | Name of Lab |
| Description | Nvarchar | Details about Lab |
| Contact | BigInt | Contact Information of area administrator |

Table 13: Camera Data

**Bounded Rectangle Data:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| RectangleID | int (Primary Key) | Id of Rectangle |
| CameraID | int (Foreign Key) | Id of Camera |
| x1 | int | Top Left (x) Coordinate of Desired Area |
| x2 | int | Bottom Right (x) Coordinate of Desired Area |
| y1 | int | Top Left (y) Coordinate of Desired Area |
| y2 | int | Bottom Right (y) Coordinate of Desired Area |

Table 14: Bounded Rectangle Data

**Admin:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| AdminEmail | nvarchar | Email of Admin |
| AdminID | int (Primary Key) | Id of Admin |
| AdminName | nvarchar | Name of Admin |
| AdminPassword | nvarchar | Password of Admin |
| AdminEmployeeID | Int | Employee ID of Admin |
| AdminRole | nvarchar | Role Type of Admin |
| Admin\_loggedin | Bit | Login Status of Admin |

Table 15: Admin Data

**Board Status:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| BoardID | Int (Primary Key) | ID Of of Board |
| RectangleID | int (Foreign Key) | Id of Rectangle associated with board |
| Relay1 | Bit | Status of Relay 1 |
| Mode1 | Bit | Automation Mode of Relay 1 |
| Relay2 | Bit | Status of Relay 2 |
| Mode2 | Bit | Automation Mode of Relay 2 |
| Relay3 | Bit | Status of Relay 3 |
| Mode3 | Bit | Automation Mode of Relay 3 |

Table 16: Board Status Data

**Energy Consumption:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| BoardID | Int (Primary Key) | ID Of of Board |
| Amp | Real | Ampere Consumption |
| Vol | Real | Voltage Consumption |
| Power | Real | Power consumption |
| Date | Date | Date of Statistics |
| Time | Date Time | Time of statistics |

Table 17: Energy Consumption Data

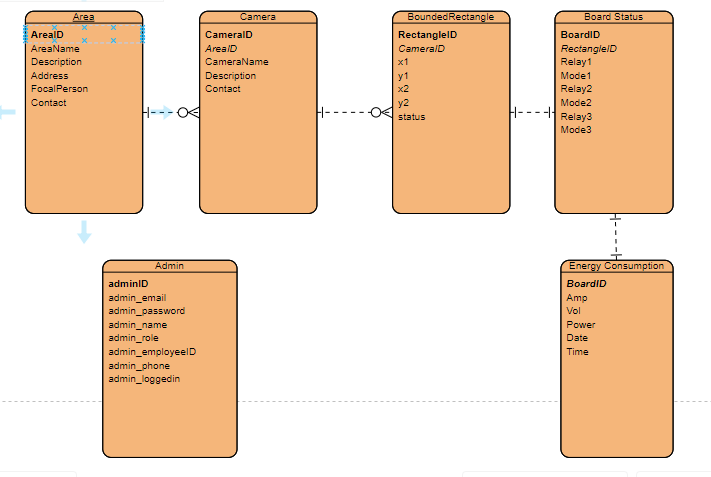


Figure 3: ER-Diagram

## Class Diagram

No classes were implemented

## Sequence / Collaboration Diagram

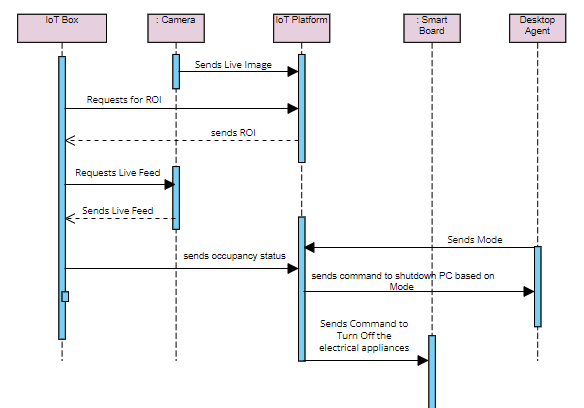


Figure 4: Sequence Diagram

## Activity Diagram

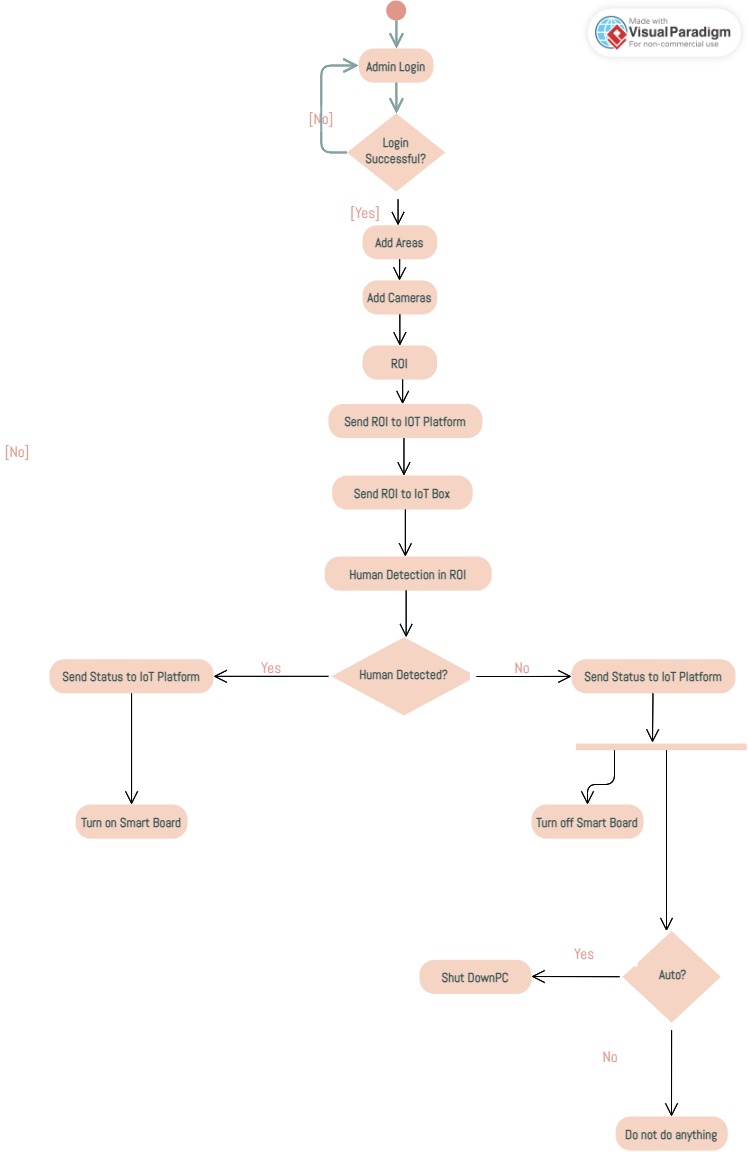


Figure 5: Activity Diagram

## State Transition Diagram

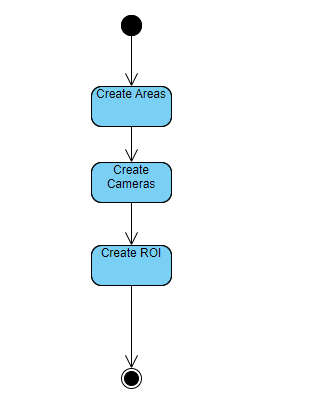
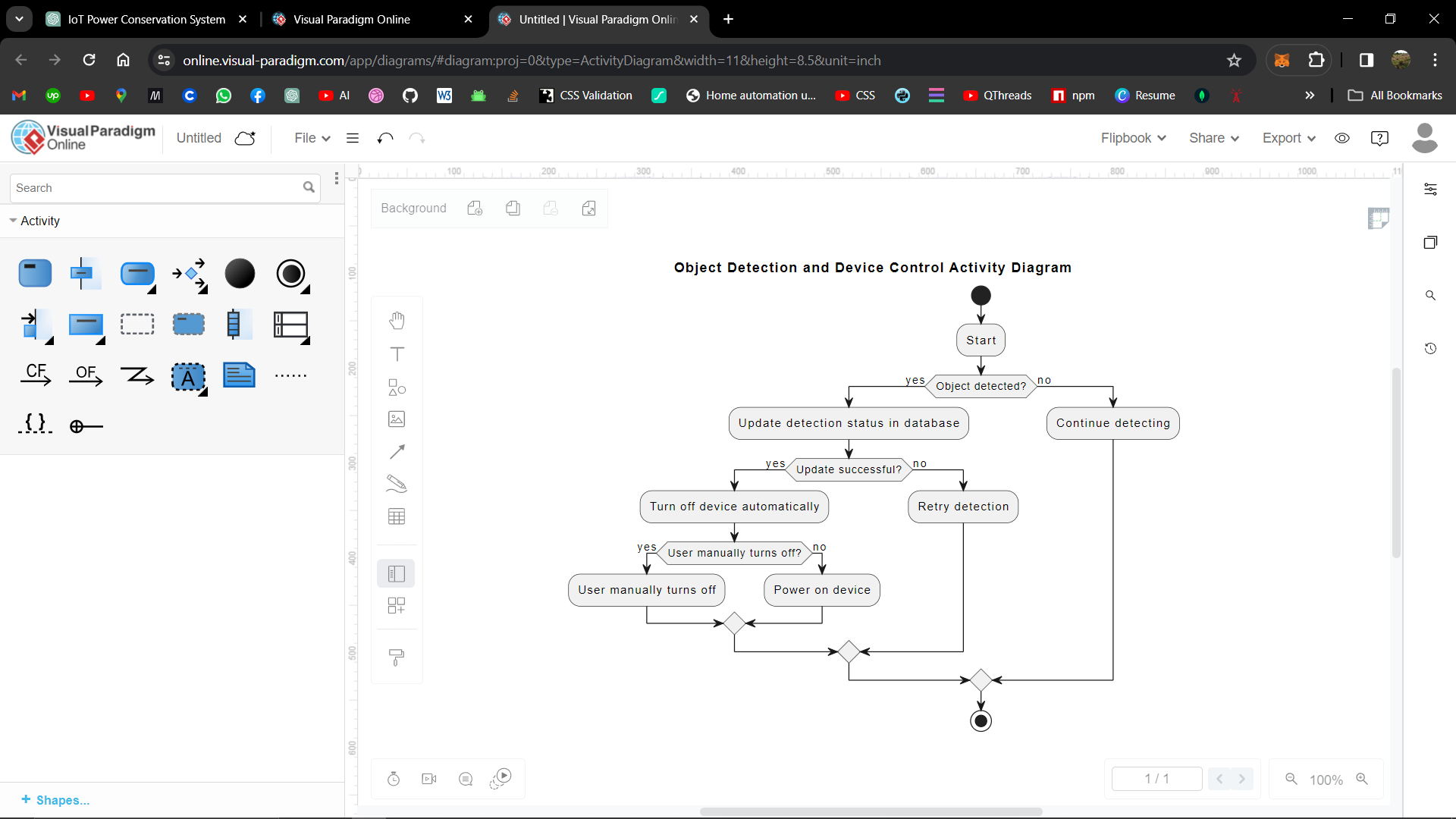


Figure 6: State Transition Diagram

## Data Flow diagram

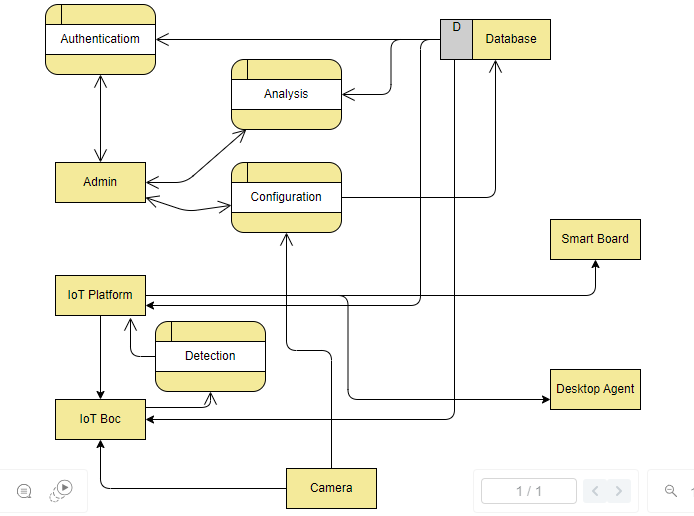


Figure 7: DFD 0

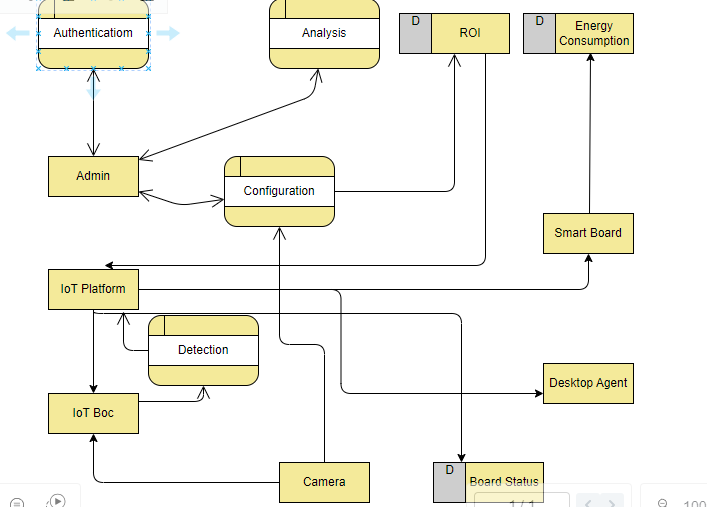


Figure 8: DFD 1

# 

# Chapter 5

# Implementation

**Chapter 5:** Implementation

This chapter delves into the technical details of the IoT-based electricity conservation system. It outlines the chosen hardware and software components, their functionalities, and how they work together to achieve the system's goals. We will explore the system architecture, communication protocols, and provide pseudocode examples for key functionalities like occupancy detection and power management logic. This chapter provides a comprehensive understanding of the system's inner workings and serves as a valuable resource for developers and system administrators.

## Important Flow Control/Pseudo codes

In our project, important pseudo codes include creating rectangles on the website for defining regions of interest, detecting human presence, controlling sockets on the smart board, and managing the PC's power state based on occupancy. The process begins with the administrator using the website to draw and define regions of interest, which are stored for subsequent image processing. The human detection algorithm, running on the Raspberry Pi, processes images captured by the CCTV camera, checking if any human is within the defined regions. If a person is detected, the Raspberry Pi sends a signal to the IoT server, which then instructs the smart board to turn on the necessary sockets, such as those controlling lights and fans. Conversely, if no human is detected for a specified period, the system sends a command to turn off the PC to save energy, provided this action is permissible within the defined user preferences. Below is an example of how these operations are structured in pseudo code:

## 5.1.1. Website:

* Administrator Interface

function drawRectangleInterface():

initialize drawing tool

event listener for mouse down:

start drawing rectangle at (x1, y1)

event listener for mouse up:

finish drawing rectangle at (x2, y2)

save coordinates (x1, y1, x2, y2) in regionsOfInterest array

end events

function saveCoordinates():

regionsOfInterest = get all drawn rectangles coordinates

save regionsOfInterest to database

## 5.1.2. Raspberry Pi:

* Capture Image and Detect Human

function captureImage():

camera = initialize CCTV camera

image = camera.capture()

return image

function detectHuman(image, regionsOfInterest):

for each region in regionsOfInterest:

croppedImage = crop image to region coordinates

detectionResult = apply human detection algorithm on croppedImage

if detectionResult contains human:

return true

return false

* Send Occupancy Status

function sendOccupancyStatus(status):

if status == "occupied":

send status "occupied" to IoT server

else:

send status "no-occupancy" to IoT server

function main():

regionsOfInterest = fetch regions of interest from database

while true:

image = captureImage()

humanDetected = detectHuman(image, regionsOfInterest)

if humanDetected:

sendOccupancyStatus("occupied")

else:

sendOccupancyStatus("no-occupancy")

wait for next image capture interval

## 5.1.3. IoT Server:

* Receive Occupancy Status

function receiveOccupancyStatus():

while true:

status = listen for status from Raspberry Pi

if status == "occupied":

instructSmartBoard("on")

reset no-occupancy timer

else:

increment no-occupancy timer

if no-occupancy timer exceeds threshold:

checkUserSettings()

* Check User Settings and Turn Off PC

function checkUserSettings():

userSettings = fetch user settings from database

if userSettings.allowPCShutdown:

send command to turn off PC

## 5.1.4. Smart Board:

* Control Relays

function controlRelays(command):

if command == "on":

turn on sockets (fan, light)

else:

turn off sockets (fan, light)

function receiveCommands():

while true:

command = listen for command from IoT server

controlRelays(command)

## Components, Libraries, Web Services and stubs

In our IoT-Based Electricity Conservation System, we integrated a variety of components, libraries, and web services to ensure a seamless and efficient operation. The core components include hardware elements like the Raspberry Pi, which serves as the main processing unit, the CCTV camera for capturing real-time images, the smart board configured with an Arduino for controlling electrical sockets, and the PC, which is used to demonstrate the system's functionality. To facilitate the development process, we utilized several libraries.

For image processing and human detection, we employed YOLOv5, a state-of-the-art object detection model. Flask, a lightweight Python web framework, was used to set up the server for handling HTTP requests. PyQt was utilized for developing graphical user interfaces. The requests library facilitated HTTP requests within our Python code.

On the front end, we leveraged React and its associated libraries, such as apexcharts for data visualization, toastify for notifications, tanstack table for data tables, icones for graphical icons, and react-spinners for loading indicators. Web services played a critical role in enabling communication between different parts of the system. We employed RESTful APIs to manage data exchange between the Raspberry Pi, the IoT server, and the website, ensuring real-time updates and control. The server side was built using Express, a web application framework for Node.js, and mssql for our connection to database. Multer was used for handling file uploads, and Nodemon was employed for automatic server reboots during development, enhancing efficiency and productivity.

## Deployment Environment

[Paragraph Text 12 pt, Times New Roman, 1.5 Line Spacing, Justified]

Deployment is yet to be done.

## Tools and Techniques

Our IoT-Based Electricity Conservation System utilizes a range of tools and technologies to ensure effective performance and ease of development. We employed the Raspberry Pi 3B+ as the central processing unit, leveraging its versatility and connectivity capabilities. The front-end was developed using React, providing a dynamic and responsive user interface. For the back-end, we used Node.js along with the Express framework to handle server-side operations and API endpoints. MSSQL served as our relational database management system for storing configuration and operational data. Python was chosen for its extensive libraries and was used alongside Flask for managing HTTP requests. YOLOv5, a state-of-the-art object detection model, was implemented for accurate human detection in image processing tasks. These tools and technologies collectively contributed to the robustness and functionality of our system.

## Best Practices / Coding Standards

We diligently applied best practices and coding standards throughout our IoT-Based Electricity Conservation System to ensure its quality, maintainability, and performance. We established a clear and organized folder structure, facilitating easy navigation and management of the project. Industry-standard coding conventions were strictly followed, including consistent variable naming and comprehensive commenting, to enhance code readability and maintainability. State management techniques were employed in our React application to efficiently handle and update the application's state. We prioritized writing clean, modular code that supports future modifications and scalability. Security best practices, such as input validation, secure authentication, and data encryption, were implemented to safeguard the system. Additionally, we focused on performance optimization to keep the system fast and reliable. These best practices collectively ensured the robustness and adaptability of our project.

## Version Control

For version control, we utilized GitHub to manage and collaborate on our IoT-Based Electricity Conservation System project. GitHub provided a centralized platform where we could track changes, manage code versions, and collaborate effectively as a team. We followed a branching strategy, creating feature branches for new functionalities and merging them into the main branch after thorough code reviews and testing. This approach allowed us to maintain a stable codebase while simultaneously developing new features. GitHub's issue tracking and project management tools helped us organize tasks, track progress, and address bugs promptly. By leveraging GitHub's capabilities, we ensured a smooth and efficient development process, facilitating collaboration and maintaining high code quality throughout the project lifecycle.

# Chapter 6

# Business Plan

**Chapter 6:** Business Plan

This chapter outlines the comprehensive business strategy for the IoT-based electricity conservation system. We will conduct a thorough market analysis to identify target customers, assess industry trends, and evaluate potential market size. A competitive analysis will compare our solution to existing options, highlighting our unique value proposition. Furthermore, a SWOT analysis will be conducted to assess the system's Strengths, Weaknesses, Opportunities, and Threats, providing a clear picture of the business landscape. This chapter establishes a roadmap for successful market entry and long-term growth of the electricity conservation system.

* 1. **Business Description**

Our project, the IoT-Based Electricity Conservation System, addresses the pressing need for sustainable energy practices in commercial settings. With a focus on leveraging Internet of Things (IoT) technology, our system offers an innovative solution to optimize electricity consumption and reduce energy wastage. By integrating smart sensors, image processing algorithms, and intelligent control mechanisms, our system enables real-time monitoring and management of electrical devices based on occupancy detection. This not only enhances energy efficiency but also contributes to cost savings for consumers and promotes environmental sustainability. Our target market includes educational institutions, office buildings, and industrial facilities seeking to adopt smart energy management solutions. Our business aims to deliver reliable, scalable, and user-friendly products and services, positioning ourselves as a trusted partner in the transition towards a greener and more efficient energy future.

## Market Analysis & Strategy

Our market analysis reveals a growing demand for energy conservation solutions, driven by increasing awareness of environmental sustainability and rising electricity costs. In commercial sectors, there is a notable trend towards adopting smart technologies to optimize energy usage and reduce expenses. Our IoT-Based Electricity Conservation System offers a competitive advantage by providing a comprehensive solution that integrates IoT sensors, image processing, and intelligent control mechanisms. We have identified key target segments such as educational institutions, office buildings, and industrial facilities, each presenting unique opportunities for implementation. To penetrate these markets effectively, our strategy involves targeted marketing campaigns, partnerships with utility companies and building management firms, and participation in industry events and exhibitions to showcase our innovative solution. Additionally, we will offer flexible pricing models and customization options to cater to diverse customer needs. By focusing on delivering value, reliability, and sustainability, we aim to capture a significant share of the growing energy conservation market.

## Competitive Analysis

In conducting a competitive analysis, we have identified several key players in the energy conservation market offering similar solutions. While some competitors focus on specific aspects such as smart thermostats or lighting controls, others offer comprehensive IoT-based systems like ours. Each competitor brings unique strengths and weaknesses to the table, with factors such as pricing, scalability, and user experience influencing market positioning. Our system distinguishes itself by its robust integration of image processing for occupancy detection, providing granular control over electrical devices based on real-time occupancy status. Additionally, our emphasis on user-friendly interfaces, customizable configurations, and scalability sets us apart from competitors. By continuously monitoring market trends and customer feedback, we aim to stay agile and responsive, ensuring our competitive edge in the dynamic energy conservation landscape.

## Products/Services Description

Our IoT-Based Electricity Conservation System offers a comprehensive suite of products and services designed to optimize energy usage and promote sustainability. At its core, our system comprises hardware components such as the Raspberry Pi, CCTV camera, and smart board with Arduino, seamlessly integrated with software solutions developed using React, Node.js, and Python. The system's key features include real-time occupancy detection using image processing algorithms, intelligent control of electrical devices based on occupancy status, and a user-friendly web interface for configuration and monitoring. In addition to the core system, we offer customization services to tailor the solution to the specific needs of our clients, as well as ongoing support and maintenance to ensure optimal performance. Our goal is to provide reliable, scalable, and user-friendly products and services that empower our customers to achieve significant energy savings and contribute to a greener future.

## SWOT Analysis

Our SWOT analysis highlights the strengths, weaknesses, opportunities, and threats of our IoT-Based Electricity Conservation System.

**Strengths**: Our system boasts robust image processing capabilities for accurate occupancy detection, intuitive user interfaces for easy configuration, and seamless integration with existing electrical infrastructure. Additionally, our focus on scalability and customization ensures flexibility to meet diverse customer needs.

**Weaknesses**: One potential weakness is the reliance on internet connectivity for real-time data transmission and control, which could pose challenges in areas with poor network coverage. Additionally, the complexity of the system may require specialized technical expertise for installation and maintenance.

**Opportunities**: The increasing emphasis on energy conservation and sustainability presents a significant opportunity for market growth. By targeting emerging sectors such as educational institutions and industrial facilities, we can expand our customer base and market reach. Collaborations with utility companies and government initiatives promoting energy efficiency further enhance our market opportunities.

**Threats**: Market competition from established players and rapid technological advancements pose threats to our market position. Additionally, regulatory changes and shifts in consumer preferences could impact demand for our solution. Addressing these threats requires ongoing innovation, strategic partnerships, and proactive market monitoring.

By leveraging our strengths, addressing weaknesses, capitalizing on opportunities, and mitigating threats, we aim to position our IoT-Based Electricity Conservation System for long-term success in the competitive energy conservation market.

# Chapter 7

# Testing & Evaluation

**Chapter 7:** Testing and Evaluation

This chapter details the comprehensive testing strategy for the IoT-based electricity conservation system. We will employ various testing methodologies, including use case testing, boundary value analysis (BVA), and equivalence partitioning, to ensure the system functions as intended under diverse conditions. The chapter will outline the test plan, test cases, and expected results. Evaluation procedures will also be covered, focusing on performance metrics like system responsiveness, accuracy of occupancy detection, and energy savings achieved. By implementing a rigorous testing and evaluation process, we aim to ensure the system's reliability, robustness, and effectiveness in real-world buildings.

## 7.1 Use Case Testing

**Test Case - 1**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | BU\_001 | **Test Case Description** | | Test the functionality of creating regions of interest (ROI) on the website | | | | | |
| **Created By** | | Azfar | **Reviewed By** | | Junaid | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Azfar | **Date Tested** | | 1-Jan-2017 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | Website is operational | | |  | 1 | Top-Left (x1, y1): 100, 100 | | | | |
| 2 | Logged in as administrator | | |  | 2 | Bottom-Right (x2, y2): 300, 300 | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | The administrator uses website to create and save ROI on the displayed image | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Logged in as admin | | Site should open | | As Expected | | | Pass | | |  |
| 2 | Navigate to Configuration Page | | Page should open with image | | As Expected | | | Pass | | |  |
| 3 | Draw ROI | | ROI are created | | As Expected | | | Pass | | |  |
| 4 | Save ROI | | ROI are save | |  | | |  | | |  |

**Test Case - 2**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | BU\_001 | **Test Case Description** | | Test the functionality of turning devices on or off based on occupancy status | | | | | |
| **Created By** | | Azfar | **Reviewed By** | | Junaid | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Azfar | **Date Tested** | | 1-Jan-2017 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | All devices are connected | | |  | 1 | Occupancy status (occupied, unoccupied) | | | | |
| 2 | System is operational | | |  | 2 | Devices (fan, light, PC) | | | | |
| 3 |  | | |  | 3 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | The system should turn on or off devices based on detected occupancy status | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Simulate occupancy in defined ROI | | Occupancy status should be shown | | As Expected | | | Pass | | |  |
| 2 | Verify devices turn on (light, fan, PC) | | Devices should turn off | | As Expected | | | Pass | | |  |
| 3 | Simulate no occupancy in defined ROI | | No occupancy status should be shown | | As Expected | | | Pass | | |  |
| 4 | Verify devices turn off (light, fan, PC) | | Devices should turn on | | As Expected | | | Pass | | |  |

**Test Case - 3**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | BU\_001 | **Test Case Description** | | Test the functionality of displaying usage data on website interface | | | | | |
| **Created By** | | Azfar | **Reviewed By** | | Junaid | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Azfar | **Date Tested** | | 1-Jan-2017 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | System is operational | | |  | 1 | Sample usage data | | | | |
| 2 | Usage data is being stored in database | | |  | 2 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | The admin views the usage data for the devices through the website | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Navigate to analytics | | Page should Open | | As Expected | | | Pass | | |  |
| 2 | Select the smart board | | Data should be displayed | | As Expected | | | Pass | | |  |

**Test Case - 4**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | BU\_001 | **Test Case Description** | | Test the functionality of detecting a human within a ROI using our model | | | | | |
| **Created By** | | Azfar | **Reviewed By** | | Junaid | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Azfar | **Date Tested** | | 1-Jan-2017 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | Model is deployed and integrated | | |  | 1 | Sample image with human | | | | |
| 2 | Website can send ROIs | | |  | 2 | Sample image without human | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | The system should correctly detect the presence of human in the defined ROI | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Simulate occupancy in defined ROI | | Human should be detected | | As Expected | | | Pass | | |  |
| 2 | Simulate no occupancy in defined ROI | | Human should not be detected | | As Expected | | | Pass | | |  |

## 7.2 Equivalence Partitioning

Creation of Regions of Interest (ROI):

Valid ROIs within boundaries

|  |  |  |
| --- | --- | --- |
| **Valid** | **Invalid** | **Invalid** |
| (50, 50, 150, 150) | (150, 50, 50, 150) | Coordinates outside image boundary |

Device Shutting On/Off Based on Occupancy Status:

Valid occupancy status and devices

|  |  |  |
| --- | --- | --- |
| **Valid** | **Invalid** | **Invalid** |
| Occupancy Status: true  Devices: Light, Fan, PC | Occupancy Status: null  Devices: Light, Fan, PC | Occupancy Status: true  Devices: non-existent |

Showing Devices Usage Data:

Valid time range and devices

|  |  |  |
| --- | --- | --- |
| **Valid** | **Invalid** | **Invalid** |
| Time Range: Start time – End time  Devices: Light, Fan, PC | Time Range: End time – Start time  Devices: Light, Fan, PC | Time Range: Start time – End time  Devices: non-existent |

Detection of Human in ROI:

Clear image with human in ROI

|  |  |  |
| --- | --- | --- |
| **Valid** | **Invalid** | **Invalid** |
| Image: clear\_image.jpg  Coordinates: (50, 50, 150, 150) | Image: clear\_image.jpg  Coordinates: (150, 50, 50, 150) | Coordinates outside image boundary |

## 7.3 Boundary Value Testing

Creation of Regions of Interest (ROI):

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Valid** | **Invalid** |
| ROI Coordinates | (1,1,1,1) | (0,0,0,0), (max\_x, max\_y, max\_x, max\_y), (max\_x + 1, max\_y + 1, max\_x + 1, max\_y + 1) |

Devices Shutting On/Off Based on Occupancy Status:

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Valid** | **Invalid** |
| Occupancy Status | False | True, Null |
| Devices | Fan, Light, PC | Non-existent device |

Showing Devices Usage Data:

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Valid** | **Invalid** |
| Time Range | (min\_time + 1, min\_time + 1) to (max\_time, max\_time) | (min\_time, min\_time) to (max\_time + 1, max\_time + 1) |
| Devices | Fan, Light, PC | Non-existent device |

Detection of Human in ROI:

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Valid** | **Invalid** |
| Image | Recognizable Features | Empty, Blurry |
| ROI Coordinates | Just inside image boundaries | (0,0,0,0), (max\_x, max\_y, max\_x, max\_y) |

## 7.4 Unit Testing

**Unit Testing 1:** Creation of Regions of Interest (ROI)

**Testing Objective:** To ensure the creation of ROI works correctly with valid and invalid coordinates.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check the ROI coordinates to validate proper coordinates within boundaries | Coordinates: (50, 50, 150, 150) | Creates ROI successfully | Pass |
| 2 | Check the ROI coordinates to validate it displays an error for x1 >= x2 | Coordinates: (150, 50, 50, 150) | Highlights field and displays error message | Pass |
| 3 | Check the ROI coordinates to validate it displays an error for out of bound coordinates | Coordinates: (-10, 50, 150, 150) | Highlights field and displays error message | Pass |

**Unit Testing 2:** Device Shutting On/Off Based on Occupancy Status

**Testing Objective:** To ensure devices shut on/off correctly based on occupancy status.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check the occupancy status with valid inputs | Occupancy: True  Devices: Fan, Light, PC | Creates ROI successfully | Pass |
| 2 | Check the occupancy status with null inputs | Occupancy: Null  Devices: Fan, Light, PC | Displays error message | Pass |
| 3 | Check the occupancy status with non-existent devices | Occupancy: True  Devices: Non-existent | Displays error message | Pass |

**Unit Testing 3:** Showing Devices Usage Data

**Testing Objective:** To ensure the usage of data display works correctly for valid and invalid time ranges and devices.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check the time range and devices with valid inputs | Time range: (‘2024-05-20T10:00:00’, ‘2024-05-20T11:00:00’)  Devices: Light, Fan, PC | Displays usage data | Pass |
| 2 | Check the time range and devices with valid inputs | Time range: (‘2024-05-20T12:00:00’, ‘2024-05-20T11:00:00’)  Devices: Light, Fan, PC | Displays error message | Pass |
| 3 | Check the time range and devices with valid inputs | Time range: (‘2024-05-20T10:00:00’, ‘2024-05-20T11:00:00’)  Devices: Non-existent | Displays error message | Pass |

**Unit Testing 4:** Detection of Human in ROI

**Testing Objective:** To ensure human detection in the ROI works correctly with valid and invalid images and coordinates.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check human detection with a clear image and valid ROI coordinates | Image: clear\_image.jpg  Coordinates: (50, 50, 150, 150) | Detects human successfully | Pass |
| 2 | Check human detection with a blurry image and valid ROI coordinates | Image: blurry\_image.jpg  Coordinates: (50, 50, 150, 150) | No human detected or displays error message | Pass |
| 3 | Check human detection with outside image detection | Image: clear\_image.jpg  Coordinates: (-10, 50, 150, 150) | Displays error message | Pass |

## 7.5 Integration Testing

**Integration Test 1:** IoT Device Communication with Web Server

**Objective:** Verify that IoT devices (e.g., Raspberry Pi) can communicate with web server to send occupancy status and receive commands.

**Methods Used:** REST API calls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Components Involved** | **Expected result** | **Result** |
| 1 | Send occupancy status from IoT device to web server | Raspberry Pi, Web Server | Server receives and logs occupancy status | Pass |
| 2 | Web server sends command to IoT device based on occupancy status | Web Server, Raspberry Pi | IoT devices command and executes action (e.g., turns on light) | Pass |

**Integration Test 2:** Image Processing and ROI Creation

**Objective:** Ensure that images captured by the CCTV camera are correctly processed and regions of interest (ROI) are created via website.

**Methods Used:** Image capture, HTTP POST requests, Flask API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Components Involved** | **Expected result** | **Result** |
| 1 | Capture image and send to Flask server for processing | CCTV Camera, Flask Server | Image received and processed, ROI data returned | Pass |
| 2 | Create ROI via web portal and verify it is stored in the database | Website, Database (MSSQL) | ROI data saved and retrievable from database | Pass |

**Integration Test 3:** Data Transfer between Web Portal and Backend Database

**Objective:** Ensure that usage data is accurately recorded in the database and displayed on the website.

**Methods Used:** RESTful APIs, Express

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Components Involved** | **Expected result** | **Result** |
| 1 | Log device usage data to database | Web Server, Database (MSSQL) | Data is logged accurately and retrievable | Pass |
| 2 | Display usage data on the website | Database (MSSQL), Web Portal | Usage data is displayed correctly on the web portal | Pass |

**Integration Test 4:** Device Control Commands

**Objective:** Ensure that commands issued from the website (e.g., turning devices on/off) are correctly executed by the IoT devices.

**Methods Used:** HTTP Requests

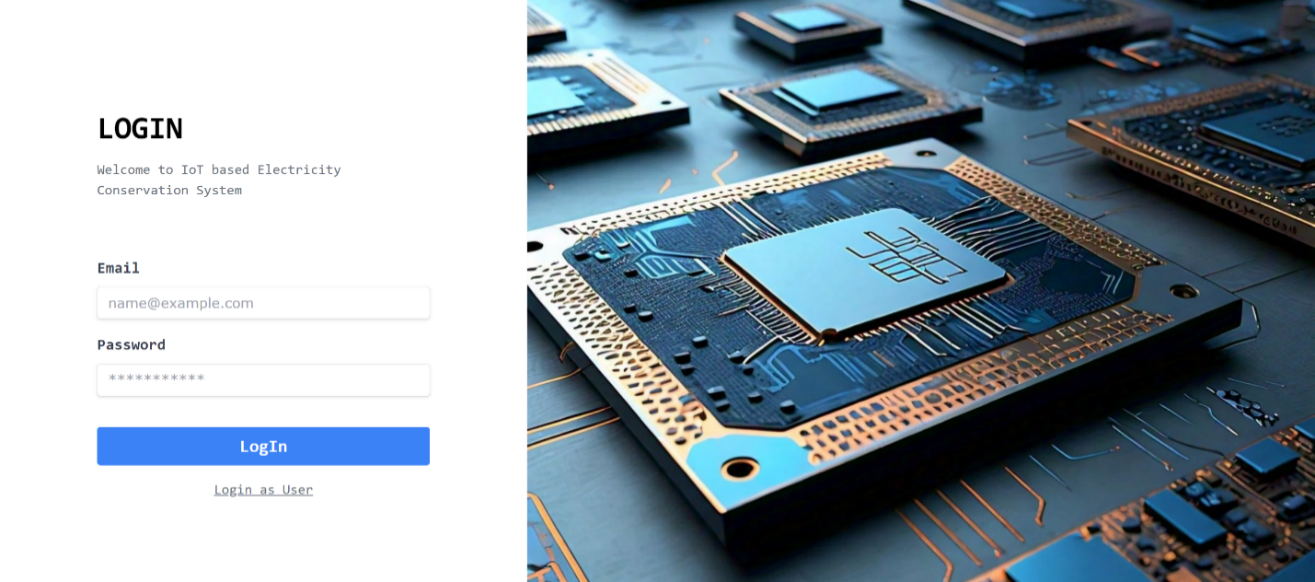
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Components Involved** | **Expected result** | **Result** |
| 1 | Send command to turn on a device from the web server | Web Server, IoT Device | Devices turn on as expected | Pass |
| 2 | Send command to turn on a device from the web server | Web Server, IoT Device | Devices turn off as expected | Pass |

## 7.6 Output and Screenshots

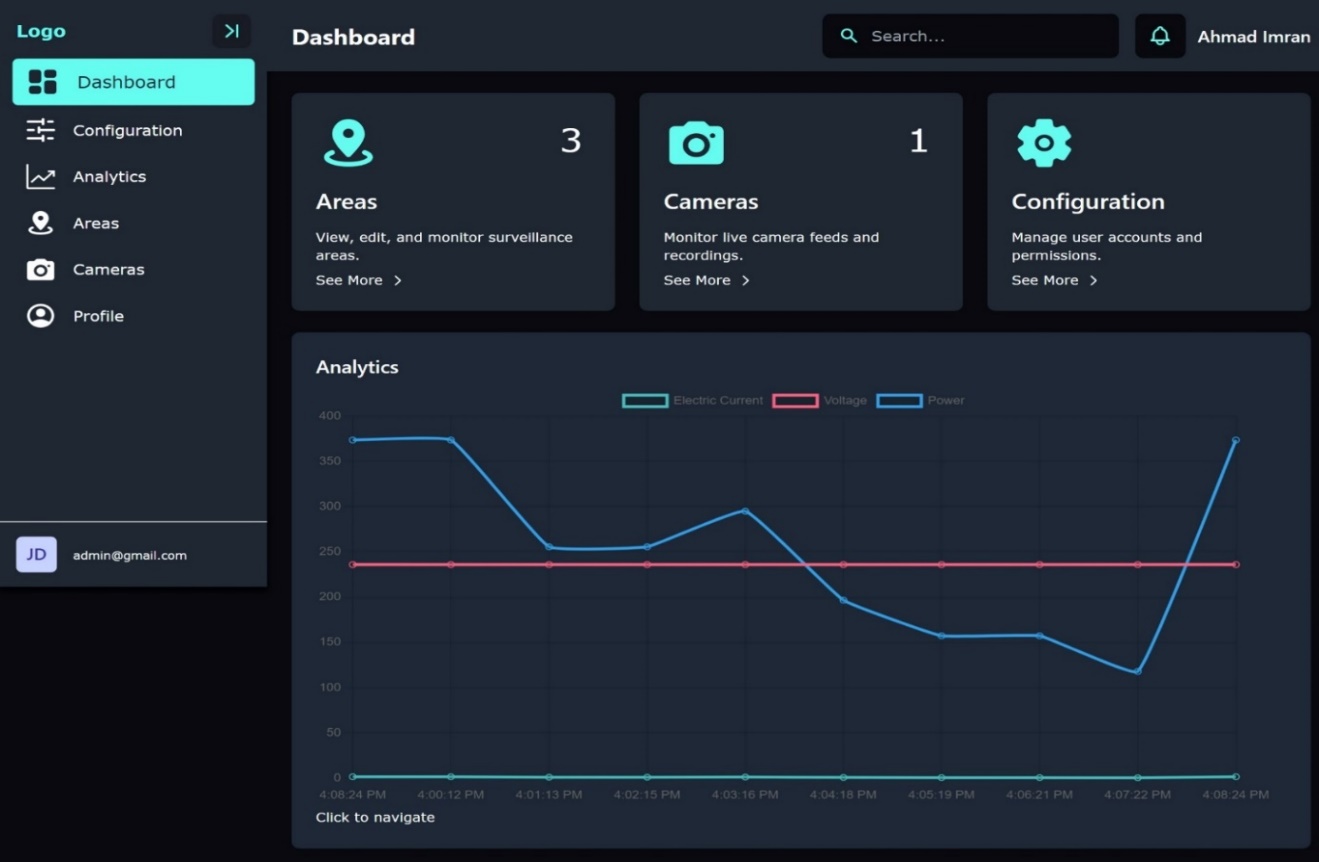
**Web Application:**

Following are the UIs of the web application where admin can login to add, configure and monitor the areas. Furthermore analytics regarding electricity consumption can be viewed:

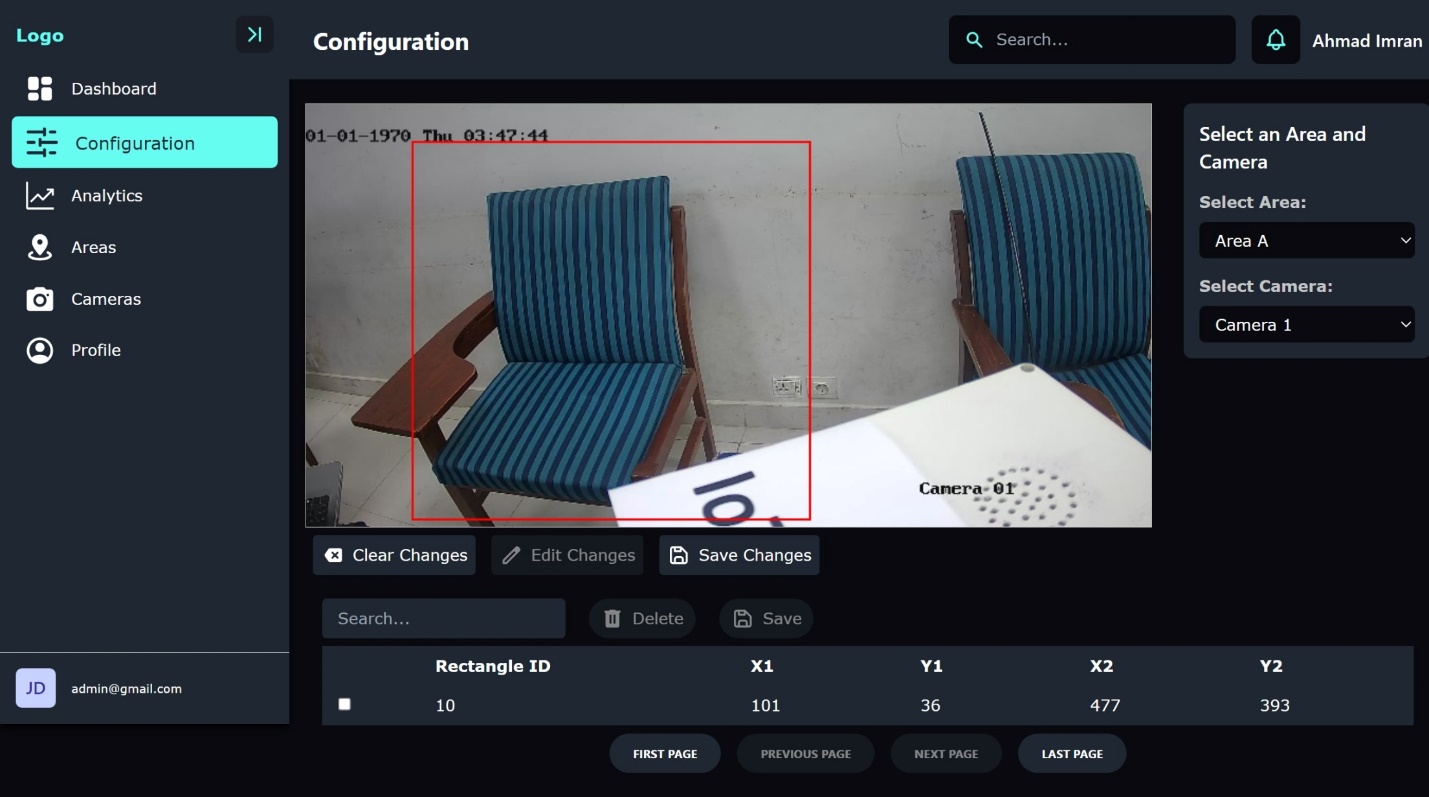
1. **Login Screen**

****

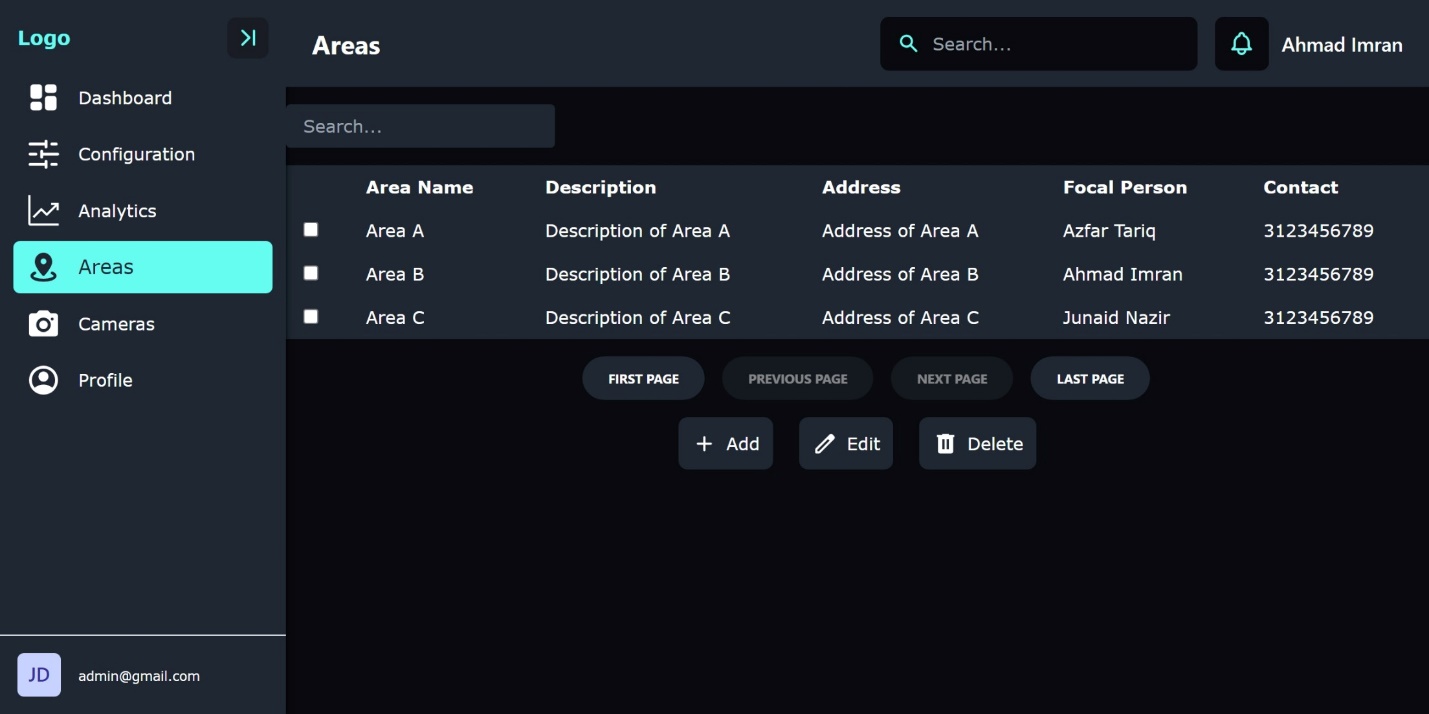
1. **Dashboard**

****

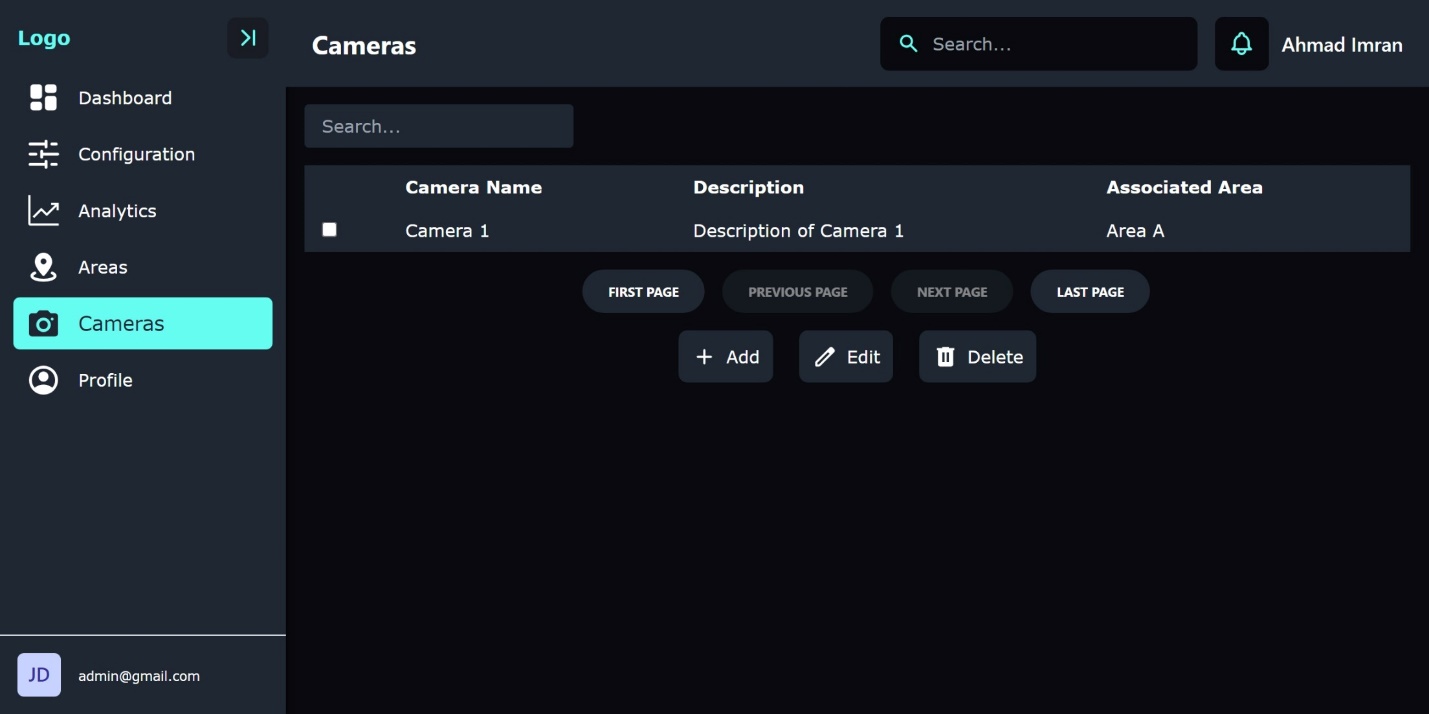
1. **Configuration**

****

1. **Areas**

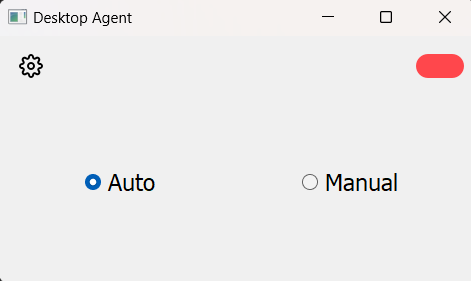
****

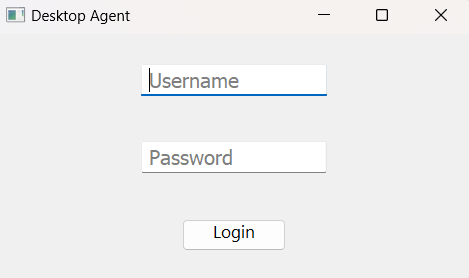
1. **Cameras**

****

**Desktop Agent:**

Following are the UIs of desktop agent that will be installed on the systems to ensure smooth shutdown in case there is no person present in the ROI. Furthermore the user can choose to keep the system running even if there is no one in the ROI in case he /she has to go away for a while.

****

****

**Prototype:**

Following is picture of working prototype:

****

# 

# Chapter 8

# Conclusion & Future Enhancements

**Chapter 8:** Conclusion & Future Enhancements

## 8.1 Achievements and Improvements

Throughout the development of our IoT-Based Electricity Conservation System, we have achieved significant milestones and made notable improvements to enhance its functionality and usability. Key achievements include the successful integration of image processing algorithms for accurate occupancy detection, the implementation of a user-friendly web interface for intuitive configuration and monitoring, and the establishment of robust communication protocols for seamless interaction between system components. Additionally, our system has garnered positive feedback from initial pilot testing, demonstrating its effectiveness in reducing energy consumption and promoting sustainable practices. Moving forward, we are committed to continuous improvement, with plans to further optimize performance, expand compatibility with additional devices and platforms, and enhance user experience through feedback-driven refinements. By prioritizing innovation and responsiveness to customer needs, we aim to maintain our position as a leader in the evolving landscape of energy conservation solutions.

## 8.2 Critical Review

A critical review of our IoT-Based Electricity Conservation System reveals both strengths and areas for improvement. While our system excels in its integration of image processing for accurate occupancy detection and its user-friendly web interface, there are areas that warrant attention. One such area is the complexity of installation and setup, which may pose challenges for users without technical expertise. Additionally, the reliance on internet connectivity for real-time data transmission could be a limitation in certain environments with unreliable network coverage. Furthermore, while our system offers comprehensive energy management capabilities, there is potential for further refinement and optimization to maximize energy savings and efficiency. Moving forward, addressing these challenges will be crucial to ensuring the continued success and adoption of our system in the competitive energy conservation market.

## 8.3 Lessons Learnt

The development of our IoT-Based Electricity Conservation System has provided valuable insights and lessons that have shaped our approach to future projects. One key lesson learned is the importance of user-centered design and simplicity in user interfaces, ensuring accessibility for a wide range of users. Additionally, effective communication and collaboration among team members are essential for project success, highlighting the significance of clear roles, responsibilities, and regular progress updates. Furthermore, the iterative nature of development underscores the importance of flexibility and adaptability to accommodate changing requirements and feedback. Another crucial lesson is the significance of thorough testing and validation throughout the development process to identify and address potential issues early on. By applying these lessons, we are better equipped to navigate future challenges and deliver innovative, user-centric solutions that meet the needs of our customers.

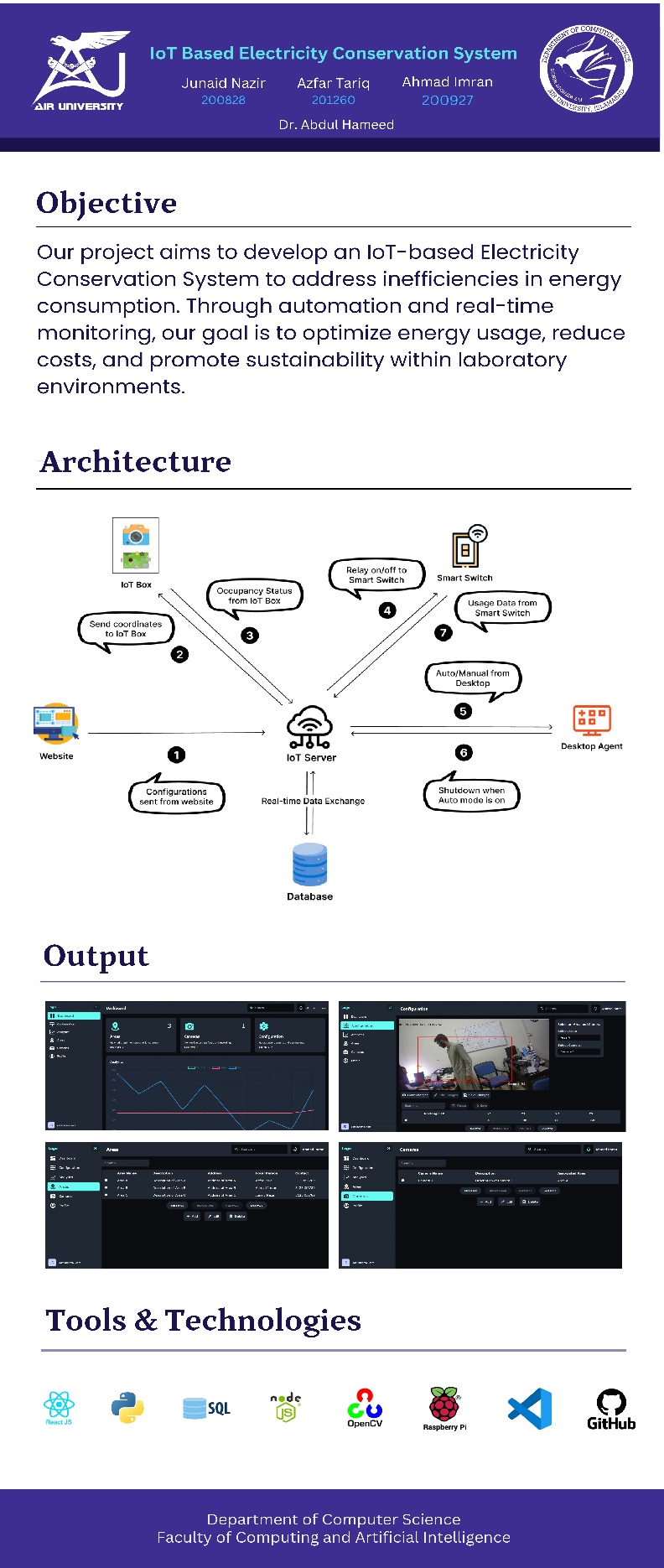
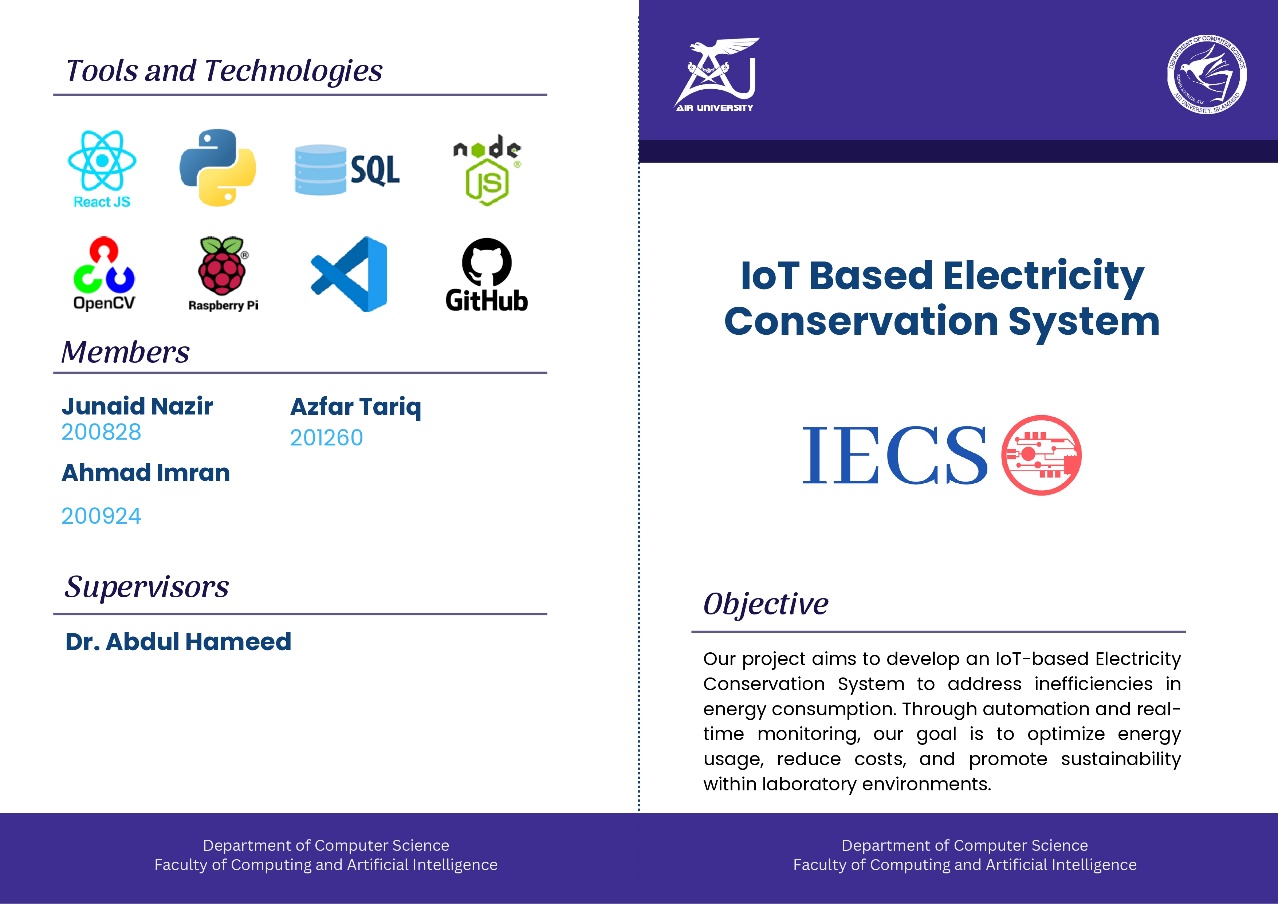
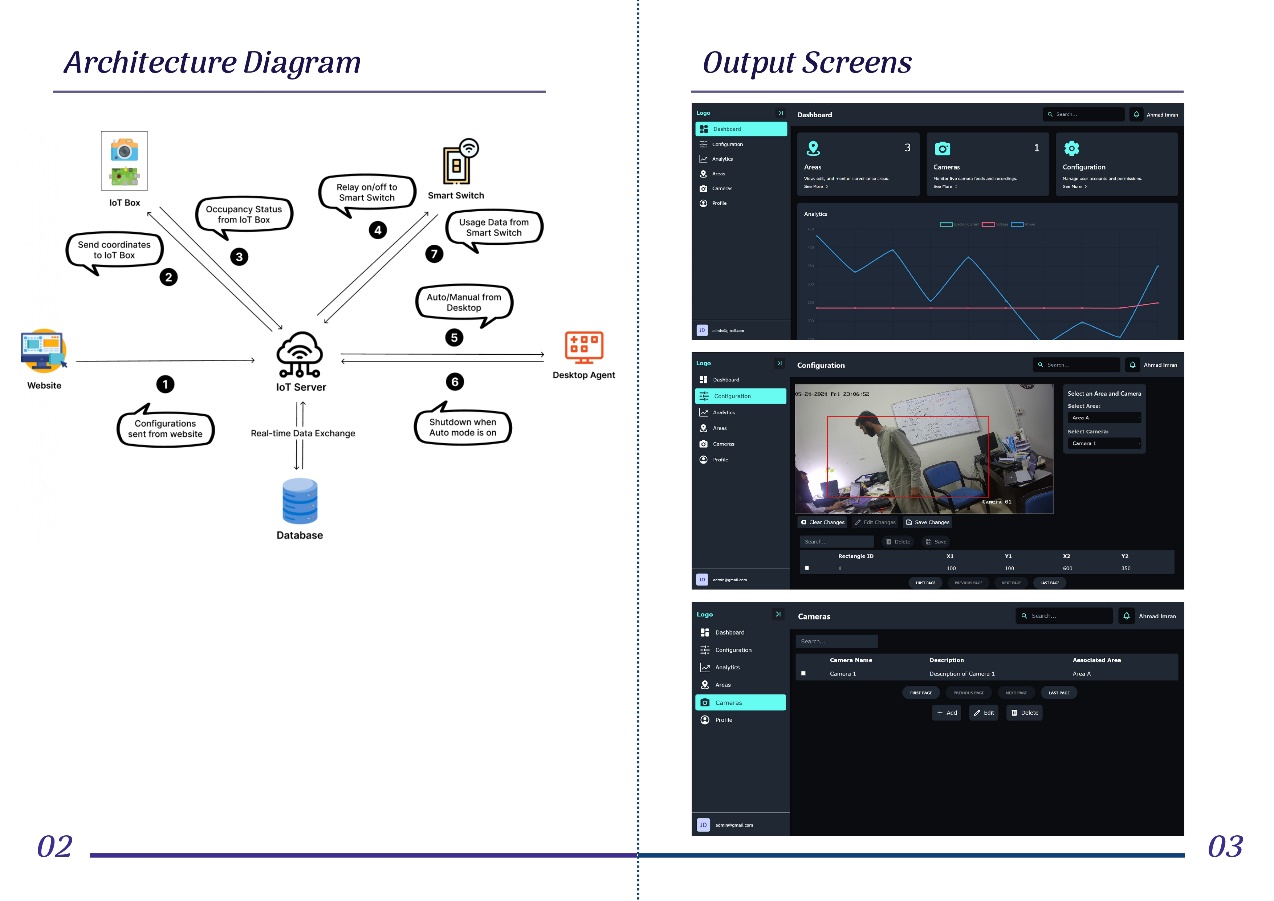
## 8.4 Future Enhancements/Recommendations

Expanding upon the collection and analysis of usage data to enhance accuracy and provide actionable insights represents a significant improvement for our IoT-Based Electricity Conservation System. Implementing advanced algorithms capable of identifying peak usage times and opportunities for energy conservation would empower users with valuable information to optimize their energy consumption patterns effectively. By leveraging machine learning techniques, the system could learn from historical data to predict future usage trends and recommend personalized strategies for energy conservation. Additionally, enhancing the system's compatibility with a wider range of home setups and smart devices would increase its accessibility and appeal to a broader audience. Through these enhancements, we aim to provide users with a more intelligent and adaptable energy management solution that delivers tangible benefits in terms of both energy savings and environmental sustainability.

# Appendices

# Appendix A: Information / Promotional Material

This appendix contains supplementary materials that provide additional context and support for the IoT-Based Electricity Conservation System project. It includes brochures, flyers, and posters designed for promotional purposes, aiming to raise awareness about the project and its benefits. These materials highlight the system's features, advantages, and potential applications, targeting stakeholders such as university administrators, faculty, and students. The appendix also features technical diagrams and infographics to visually communicate the system's functionality and impact on energy conservation.

* 1. ******Standee**
  2. **Broacher**

**Reference and Bibliography**

**Note:** Use Mendeley or Endnote

[1] Raspberry Pi Documentation, Raspberry Pi Foundation, Available at: <https://www.raspberrypi.org/documentation/>

[2] YOLOv5 Documentation, Available at: <https://docs.ultralytics.com/yolov5/>

[3] Flask Documentation, Pallets Projects, Available at: <https://flask.palletsprojects.com/>

[4] PyQt Documentation, Riverbank Computing, Available at: <https://riverbankcomputing.com/software/pyqt/intro>

[5] Node.js Documentation, OpenJS Foundation, Available at: <https://nodejs.org/en/docs/>

[6] Express Documentation, OpenJS Foundation, Available at: <https://expressjs.com/en/starter/basic-routing.html>

[7] React Documentation, Meta Platforms, Available at: <https://reactjs.org/docs/getting-started.html>

[8] OpenCV Documentation, OpenCV team, Available at: <https://docs.opencv.org/>