|  |  |
| --- | --- |
| Al-Quds University  Faculty of Engineering  Computer Engineering Department | sa.png |

Project name:

**Smart waste management system**

By:

Muhannad Taha

Haitham Ajaj

Supervisor:

Dr. Emad Hamadeh

This project is submitted in partial fulfilment of the requirements for the degree of BSc in Computer Engineering from Al-Quds University.

Palestine, 2020

**Table of contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Subject | | | | Page Number |
| Abstract | | | | III |
| Table of contents | | | | IV |
| List of figures | | | | VI |
| List of Tables | | | | VIII |
| Chapter One: Introduction | | | | |
| Section Number | | **Section Name** | **Page Number** | |
| 1.1 | | Introduction | 8 | |
| 1.2 | | Objectives | 8 | |
| 1.3 | | Problem Statement | 8 | |
| 1.4 | | Literature Review | 9+10 | |
| 1.5 | | Methodology and Methods | 10 | |
| Chapter Two: System Analysis | | | | |
| 2.1 | | Introduction | 12 | |
| 2.2 | | Project implementation options | 12 | |
| 2.3 | | The proposed system | 13 | |
| 2.4 | | System requirements | 13+14 | |
| 2.5 | | Development requirements and cost | 14+15 | |
| 2.6 | | Cost benefits | 15 | |
| 2.7 | | Feasibility study | 15-17 | |
| 2.8 | | Risk Evolutions | 17 | |
| 2.9 | | Project added values | 18 | |
| 2.10 | | Project management | 18 | |
| 2.11 | | Conclusion and recommendations | 18 | |
| Chapter Three: Requirement & Theory | | | | |
| 3.1 | | Introduction | 20 | |
| 3.2 | | Use-case Diagram | 20 | |
| 3.3 | | Functional requirements description | 21 | |
| 3.4 | | Non-Functional requirements description | 21 | |
| 3.5 | | Hardware component | 22-24 | |
| 3.6 | | Conclusion and recommendations: | 24 | |
| Chapter Four: System Design and Development | | | | |
| 4.1 | | **Project layout diagram** | **26** | |
| 4.2 | | Block Diagram | 27 | |
| 4.3 | | Flow Chart: | 28+29 | |
| 4.4 | | Circuit diagram | 30 | |
| 4.5 | | Sequence diagram | 30-32 | |
| 4.6 | | Activity diagrams | 33+34 | |
| 4.7 | | Class diagrams | 35 | |
| 4.8 | | Conclusion and Recommendations | 35 | |
| Chapter Five: Coding and Implementation | | | | |
| 5.1 | | Introduction | 37 | |
| 5.2 | | Technical approach | 37 | |
| 5.3 | | Coding Programming Languages | 37 | |
| 5.4 | | Data base system | 38+39 | |
| Chapter Six: Testing | | | | |
| 6.1 | Introduction | | 41 | |
| 6.2 | System testing plan | | 41 | |
| 6.3 | Software Testing | | 41-45 | |
| 6.4 | Hardware Testing | | 45-49 | |
| 6.5 | Conclusion and Recommendations | | 49 | |
| Chapter Seven: Conclusions and future works | | | | |
| 7.1 | Conclusions | | 51 | |
| 7.2 | Future works | | 51 | |
| 7.3 | References | | 51 | |
| 7.4 | Appendix Code C# | | 52-54 | |
| 7.5 | Appendix Arduino code | | 55-59 | |

**List of figure**

|  |  |
| --- | --- |
| Figures name | Page |
| Figure 2.1 Use Case | 20 |
| Figure 3.1: Ultra sonic Sensor | 22 |
| Figure 3.2: ESP8266 Wi-Fi module | 22 |
| Figure 3.3: Servo motor | 22 |
| Figure 3.4: Buzzer | 22 |
| Figure 3.5: RGB Led | 23 |
| Figure 3.6: Prototype | 23 |
| Figure 4.1: Project layout | 26 |
| Figure 4.2: Block Diagram | 27 |
| Figure 4.3: Flow chart | 28 |
| Figure 4.4: Schematics | 30 |
| Figure 4.5: Employee sequence diagram | 31 |
| Figure 4.6: Admin sequence diagram | 32 |
| Figure 4.8: Sign in activity | 33 |
| Figure 4.8: Employee activity | 34 |
| Figure 4.9: Class Diagram | 35 |
| Figure 6.1: Employee page testing | 42 |
| Figure 6.2: Admin page testing | 44 |
| Figure 6.3: Add trash testing | 45 |
| Figure 6.4: Open the trash automatically 1 | 47 |
| Figure 6.5: Open the trash automatically 2 | 48 |
| Figure 6.6: System alert | 48 |
| Figure 6.7: Monitoring trash | 49 |

**List of Tables**

|  |  |
| --- | --- |
| Table name | Page |
| Table 2.1: Development Software Cost. | 14+15 |
| Table 2.2Time table (Gantt table) | 16+17 |
| Table 3.1 Hardware Component | 24 |
| Table 5.1 Employee database table | 37 |
| Table 5.2 Admin database table | 37 |
| Table 5.3 Trash info database table | 38 |
| Table 6.1 Employee sign in | 41+42 |
| Table 6.2 Admin sign in | 43 |
| Table 6.3 Employee add trash info | 44 |

**ABSTRACT**

With the passage of years, the population increased dramatically, which also led to a greater increase in waste than it was, which was followed by several things, so we discussed the solution to this problem by designing a smart waste container. The smart bin is built on the WIFI module board microcontroller platform, interfacing with the WIFI module and the Ultrasonic sensor. The ultrasound sensor is placed at the top of the trash that will measure the position of the garbage. WIFI module will be programmed so that when the trash is filled, the height of the trash will be shown in it. Once the garbage has reached the ultrasound sensor it will trigger the WIFI module that will alert the employee that the garbage bin has been filled, if the container is completely filled and reaches 10 cm from the ultrasound sensor the LED will turn red This will inform the employee who will call the garbage collection vehicle to go to the garbage bin and empty it. Upon the arrival of the cleaning worker, the garbage will be opened automatically. In this way the environment will be preserved and the current waste management will be greatly appreciated.

**Chapter one: Introduction**

|  |  |  |
| --- | --- | --- |
| 1.1 | Introduction | 6 |
| 1.2 | Objectives | 6 |
| 1.3 | Problem Statement | 6 |
| 1.4 | Literature Review | 7+8 |
| 1.5 | Methodology and Methods | 8 |

**1.1 Introduction**

In many Palestinian cities, there are many problems in garbage collection, which generates environmental pollution in them, so they need to be managed in the right way. To overcome these problems, we discussed using the internet of things in our graduation project to serve public places and private places, and this garbage is under the supervision of a specific employee. We will talk about the methodology of this idea, and we will address the pieces that will be used and the goals to be reached, and we will clarify with the flow chart the way the project will work and the block diagram.

**1.2 Objectives**

The main objectives of the development of this project are:

* the protection of environment through effective waste management techniques.
* to protect health, wellbeing and environment.
* to prevent pollution.
* safe disposal of waste.
* to create awareness among the people about the impact of waste.
* ensure and promote proper solid waste management.

**1.3 Problem Statement**

The traditional garbage containers that exist today are not a practical solution for efficient waste management, which is very harmful to the environment, and ineffective ways to get rid of garbage, which are on the streets and in public places, which makes the environment unhealthy in addition to making a great effort for the workers as it is in some Sometimes they empty the garbage bins not completely full, this project is aim to design an effective waste management mechanism for proper disposal of waste and motivating people and influencing them to follow appropriate methods to get rid of waste.

**1.4 Literature Review**

**1.4.1 Ecube Labs.**



Established in 2011, Ecube Labs [1] is an innovative green technology company committed to providing eco-friendly waste management solutions for smart and sustainable cities of tomorrow. their current line of offerings includes an integrated, IoT-based solution that dramatically improves the efficiency of waste collection, all the while making our environment greener and communities cleaner.

They help cities and waste collection organizations alike to reduce operational costs by eliminating unnecessary pick-ups, providing dynamic collection routes and schedules for a complete optimization of the collection operations.

Essentially, they provide transparency and increase operational effectiveness in the waste collection chain, reducing costs and labor associated with such services. their eco-friendly waste management solutions also improve public cleanliness by reducing the impact of overflowing bins.

**1.4.2 NORDSENSE**



Nordsense [2] has created to waste management solution. We optimize the trash collection process by monitoring container levels, applying advanced data analytics, and delivering need based turn-by-turn directions to a driver. Nordsense offers a unique, easy to use, efficient, cost-effective and scalable smart waste solution for cities, waste operators, and trash haulers. The Nordsense solution provides a positive environmental impact by reducing both the number of vehicles on the road and fuel consumption, as well as preventing unattractive and unsanitary overflowing bins.

**1.4.3 Bigbelly**

****

Bigbelly [3] was founded in 2003 with the goal of transforming one of the least efficient and resource-intensive industries: waste collection. Cities were either collecting too often and wasting fuel and labor while emitting excess CO2, or alternatively, they were not able to keep up with the demands and overflowing trash cans that created litter, health, and safety issues.

In the following years, we evolved to offer a unique and complete solution by leveraging renewable solar energy and information technology. Bigbelly is a Smart Waste Management, Smart City, and Internet of Things (IoT) industry leader and is recognized as a C40 Cities Climate Leadership Group best practice. Industry knowledge & longevity in the market provides us with unparalleled insights backed by extensive data, analytics, and customer successes.

**1.5 Methodology and Methods**

The proposed system will contain two units:

**The smart garbage container:** which will be a garbage prototype that contains the sensors, servos, and the microcontroller. In this unit the fullness status of the garbage containers is determined by calculating the distance between the lid of the container and the trash by using an ultrasonic sensor. A distance threshold will be set according to the container dimensions. When the distance measuring sensor indicates that the container is full, then a microcontroller board will send the measuring data to an IoT API in the cloud using a wifi module connected to the internet by a 3G or local wifi connections.

**The local health authority unit:** the data in the IoT API is then acquired by the local health authority web application that developed to display the status of each container in the local area. Employee can then decide to send the garbage car to the container and emptying it. The system will also contain a mechanism to open or close if there a user wants to put a garbage in it.

**Chapter two: System Analysis**

|  |  |  |
| --- | --- | --- |
|  | Introduction | 12 |
|  | Project implementation options | 12 |
|  | The proposed system | 13 |
|  | System requirements | 13+14 |
|  | Development requirements and cost | 14+15 |
|  | Cost benefits | 15 |
|  | Feasibility study | 15-17 |
|  | Risk Evolutions | 17 |
|  | Project added values | 18 |
|  | Project management | 18 |
|  | Conclusion and recommendations | 18 |

* 1. **Introduction**

This chapter describes the economic feasibility and the requirements of the system. The chapter also defines restrictions on the operation of the system and its implementation. Therefore, this chapter introduces the system requirements by describing the functional requirements and the non-functional requirements.

This chapter also shows the system specifications and the risks that faces the system and the system developers and the possible solutions.

* 1. **Project implementation options**
     1. **Software side**

For this project, there are three options that can be used to for the development; these options can be summarized as follows:

* **Standalone Application**
* **Web Application**
* **Mobile application**

Therefore, the selection of a suitable option depends on the business needs (requirements) and factors discussed in the comparison given above. Accordingly, we choose to develop this system as a Web application using ASP.net, C# as programming Language and SQL Server to build the Database.

* + 1. **Hardware side**

For this project, there are three options that can be used to as a controller; these options can be summarized as follows:

* **Arduino (Uno …)**
* **Raspberry PI**
* **Just WI-FI module**

We choose WI-FI module to Program this Controller using C as programming Language.

* 1. **The proposed solution**

The proposed solution will be an online portal allowing the employee and the administrator to monitor the trash cans that will be associated with the cloud that takes readings from the sensors that express the state of the trash.

**The portal contains two main interfaces**:

1. **Admin interface (district chief):** whose role is to view the trash cases.
2. **The employee interface:** through which he can add some general data for the trash bin and the bin also see the cases of the trash bin in order to inform the cleaning worker if it is full so that it can be emptied immediately.
   1. **System requirements**

It is well known that the main purpose of developing any system is to achieve several requirements and services known functions. These requirements can be classified into two types: functional requirements, and non-functional requirements which are explained below.

* + 1. **Functional requirements**

This section describes the main functions of the proposed model. More precisely, the proposed system includes the following actors and how each has its functional requirements as solution to the problem that have been discussed previously:

* **System:**
* **Admin:**
* **Employee**

These actors and their functional requirements will be explained in the next chapter.

* + 1. **Non-functional requirements**

In this section, the non-functional requirements of the application will be explained, which are the services that the system must provide. And how the system responds for specific inputs and how it acts. The aim of these requirements is to define the properties of the system Including

* **Extensibility**:
* **Portability**:
* **Usability**:
* **Reliability**:
* **Availability**:
* **Security**:

Also, these requirements will be explained in the next chapter.

**2.5 Development Requirements and Cost**

## 2.5.1 Software

The following table lists the software of that needed to develop this project and their costs:

**Table 2.1:** Development Software Cost.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Number of Units | Unit Cost | Available | Total |
| Visual studio 2015 community version | 1 | $0 | Yes | $0 |
| Photoshop cc student version | 1 | $0 | Yes | $0 |
| Website Hosting plan | 1 | 150$ | yes | 150$ |
| Total | | | | $150 |

## 2.5.3 Humans

The following table lists the costs that needed to develop this project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Member | Number | Cost/Month $ | Available | Total/months |
| Eng. Students | 2 | $300 | Yes | $600 |
| Total | | | | $600 |

|  |  |
| --- | --- |
| Type | Total |
| HW | $50 |
| SW | $150 |
| Humans | $600 |
| Total | $800 |

**2.6 Cost Benefit**

The main benefits of this project according to the cost is from hardware side is that the components are not expensive and the project has a few components, and from the side of the software, the web page is simple and it will not need a high cost to develop it, so when the system become in the industrial phase it will be cost effective.

* + 1. **Economic feasibility**

The portal is about building a portal website which will hosted on a hosting company server these company offers hosting plans which differ from company to the other. our website may use an economic plan because we think it will not have that big traffic, such plan details as follow:

**Web Hosting plan features:**

* Traffic: unlimited
* Size: unlimited
* Emails: unlimited
* Support: yes

**This Plan price is about: 250$ / year.**

**2.7.2 Technical feasibility**

This project requires professional programming capabilities, experience in web development and web design, and some knowledge about front end building, how build circuit and handling with components and experience in IoT field. All these requirements must be available for the programmers, developers and designers, who have an experience in programming languages such as C# and ASP.net technology, SQL server and C for microcontroller programing.

**2.7.3 Legal feasibility**

In the regular situation, there are no limitations or policies in building this project, so we do not need to take any license to implement this project, and there are no illegal issues.

* + 1. **Schedule feasibility**

This section will illustrate each step required in developing this system with the time needed to complete each step. Table 2.2 shows the project time planning for the semester of the academic year 2020/2021.

|  |  |  |  |
| --- | --- | --- | --- |
| MONTH | CONTENTS | | DETAILS |
| September | Components | | Buy component needed for the project |
| October | Realization | | Design and connection circuit transmitter & receiver |
| November | Implementation | | WI-FI module & prototype implementation |
| December | Results and faced  Problems | | Show the result & discussion in our project, put the problems we encountered in project design |
| January | Testing | Testing of the project in terms of connection, programming and output | |

Table 2.2 Time table (Gantt table)

* 1. **Risk Evaluations**

This section discusses the risks that may appear during project development and the possible solutions:

* **Hardware Failure:** To avoid this risk we will make a continuous daily backup of the project on flashes and other hard disks or on cloud.
* **Shortage of development time:** We all hope that the project will terminate in the determined time. The project team will divide the course time in good way to finish it. In case of shortage, working time will be doubled.
* **Closure (Covid-19):** We all hope that the current situation due to **COVID 19** will be finished and everything return to its normal conditions. During this closure the team will continue working online.
* **Illness of a project team member:** We all wish the healthiness for the project team, but in the illness case the other members should try to take his rule.
* **User’s dissatisfaction:** We all hope that all users could use the project in an easy way without any difficulties. To do this a continuous feedback is taken from them during the development time, and making sure that functions discussed with them before implementing.
  1. **Project added values**

The main goal of developing such a project is to build a portal to monitor trash can to preserve the safety of the environment and then to connect it through the cloud to the hardware circuit. Additionally, developing this project allows the developer and designer to take advantage of the following educational benefits:

* Acquire information about web development, web design and how to send data from Wi-Fi module to Cloud and how put this data in web site that we design
* Search and analysis skills.
* Gain experience in: C# programming language, ASP.net, SQL server and C.
* Documentation Skills.
  1. **Project management**

A project good management will support the project developer to clearly understand the problems of the project, and then successfully develop it. The developer needs first to gain research and analysis skills. learn web development technologies programming languages.

Therefore, the communication plan between the developer and the supervisor is usually made through:

* Email.
* Zoom app.
  1. **Conclusion and recommendations**

In this chapter, we have declared a number of options that can be used to develop such project. After identifying the advantages and disadvantages of each option, we have selected to implement our project as a web application and WI-FI module as Controller and sender in the same time. Then, we have described our proposed system in details, and how it is feasible to implement from economic, technical, scientific and legal perspectives.

After that, we have talked about the project added value to the work team and to the society. The next chapter will illustrate the system's functional and nonfunctional requirements in details.

**Chapter Three: Requirement & Theory**

|  |  |  |
| --- | --- | --- |
|  | Introduction | 20 |
|  | Use-case Diagram | 20 |
|  | Functional requirements description | 21 |
|  | Non-Functional requirements description | 21 |
|  | Hardware component | 22-24 |
|  | Conclusion and recommendations: | 24 |

**3.1 Introduction:**

Chapter 3 will introduce the main functional and non-functional requirements for developing the proposed system. In this chapter, the functional requirements will be explained in more technical terms, so the data will be collected and analyzed to fill the specification of the software requirements. More precisely, this chapter clarifies the Use-Case diagram is used to support requirements elicitation and to show the system functions. Then, the chapter covers the detailed functional description for the proposed system, we will also talk about the components used in the circuit electrical.

**3.2 Use Case Diagram:**

This section illustrates the use-case diagram, which is shown in figure 3.1 that is used to illustrate the system requirements and function. The use case diagram of the system is showing the system's functions that should be implemented to develop the system. A clear description of each one will be illustrated in the next sections.

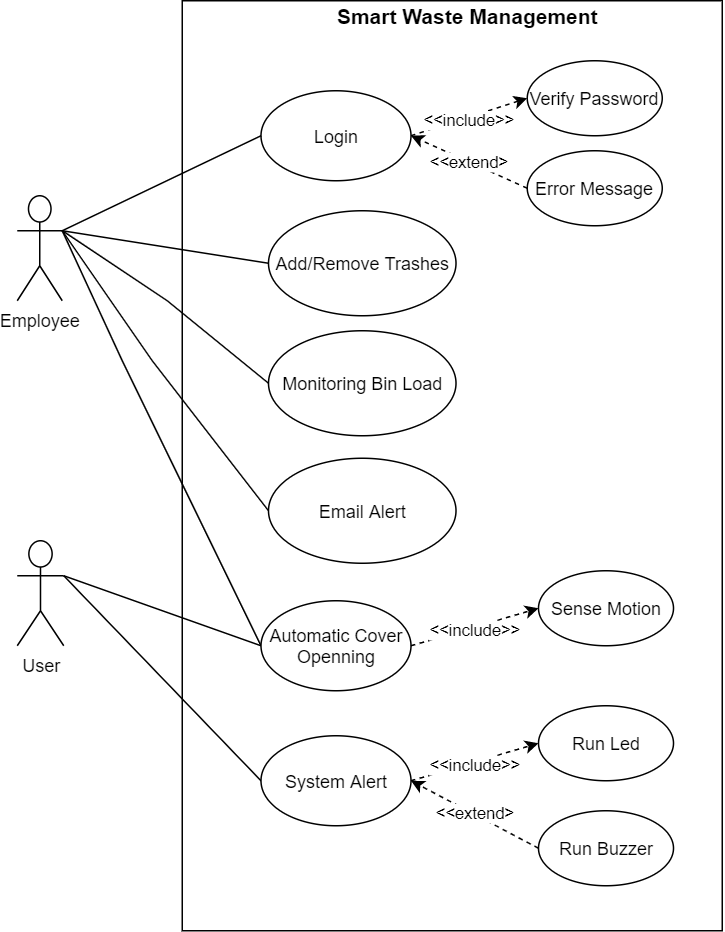


Figure 2.1 Use Case Diagram

**3.3 Functional Requirement**

## This project will help the employee to enhanced the health status of his local area with the least effort, fatigue and costs.

The proposed solution has three main functional entities:

1. **The system:** the garbage unit that contains all sensors and microcontrollers that send data to IoT API and the web application
2. **The employee:** the employee can access the web application and monitor the garbage container status and he can add trash in the web site page.
3. **The admin:** the admin can access the web application and monitor the garbage container status
4. **The user: the citizen:** can use the container and put the garbage inside.

## 3.4 Non-Functional Requirements:

## Performance: our project should have a good performance that means the system should response to the surrounding environment in a minimum time effectively.

## Usability: this means that is our project is easy to use for the different type of people.

## Availability: the project will be available to use all the time for to use.

## Security: a secure connection channel between each unit will be implemented.

## 

## 3.5 System components:

### 3.5.1 Hardware:

We used the following components in our project:

|  |  |  |  |
| --- | --- | --- | --- |
| **The name of piece** | **Picture** | **Number of figure** | **Datasheet** |
| An Ultrasonic Sensor | Ultrasonic Sensor, अल्ट्रासॉनिक सेंसर at Rs 700 ... | Figure 3.1 | <https://components101.com/ultrasonic-sensor-working-pinout-datasheet> |
| ESP8266 Wi-Fi module | C:\Users\user\Desktop\169034-91f-web.jpg | Figure 3.2 | <https://components101.com/wireless/esp8266-pinout-configuration-features-datasheet> |
| Servomotor | Servo Motor M MG995 (Metal Gear) | Smart Prototyping | Figure 3.3 | <https://education.ti.com/html/webhelp/EG_Innovator/EN/content/eg_innovsys/m_io-datasheets/io_ds_servomotor.HTML> |
| Buzzer | ÙØªÙØ¬Ø© Ø¨Ø­Ø« Ø§ÙØµÙØ± Ø¹Ù âªbuzzerâ¬â | Figure 3.4 | <https://components101.com/buzzer-pinout-working-datasheet> |
| RGB LED | C:\Users\user\Desktop\11120-Diffused_LED_-_RGB_10mm-01.jpg | Figure 3.5 | <https://components101.com/diodes/5mm-round-led> |

Table 3.1 Hardware Component

### 3.5.2 garbage model (System prototype)

We will make a small model to demonstrate the system operation so it can be used as a reference or prototype to a real type one which can be used in real life.



Figure 3.6 Prototype

### 3.5.3 Software:

We use the software to program the WI-FI module

We also will build a web application so it can read and display sensor reading that will acquire from IoT API.

**3.6 Conclusion and recommendations:**

These functions have been identified according to the problem of the system which has been explained previously, we drew the use case and we mentioned the pieces that we will use in making this project.

**Chapter Four:**

**System Design and Development**

|  |  |  |
| --- | --- | --- |
| 4.1 | Project layout diagram | 26 |
| 4.2 | Block Diagram | 27 |
| 4.3 | Flow Chart: | 28+29 |
| 4.4 | Circuit diagram: | 30 |
| 4.5 | Sequence diagram: | 30-32 |
| 4.6 | Activity diagrams: | 33+34 |
| 4.7 | Class diagrams: | 35 |
| 4.8 | Conclusion and Recommendations | 35 |

### 4.1 Project layout diagram:

The following diagram shows the main components of the system and the relationship between the work of each tool and how to connect them.

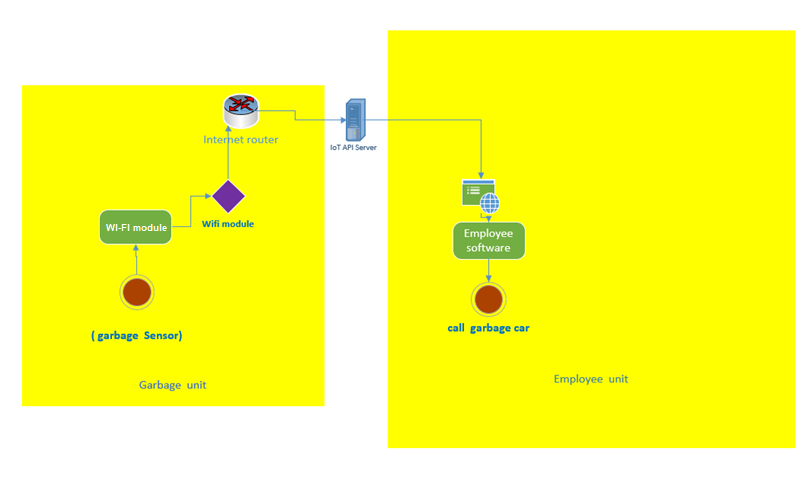


Figure 4.1 project layout

**4.2 Block Diagram**

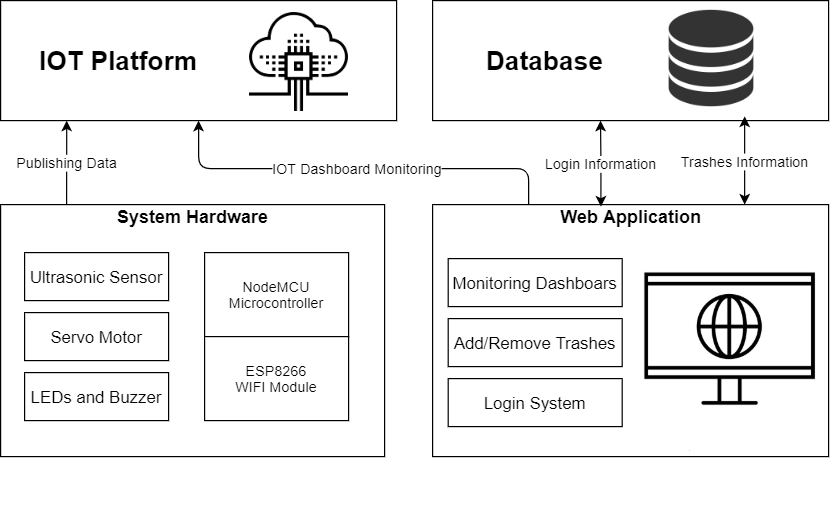
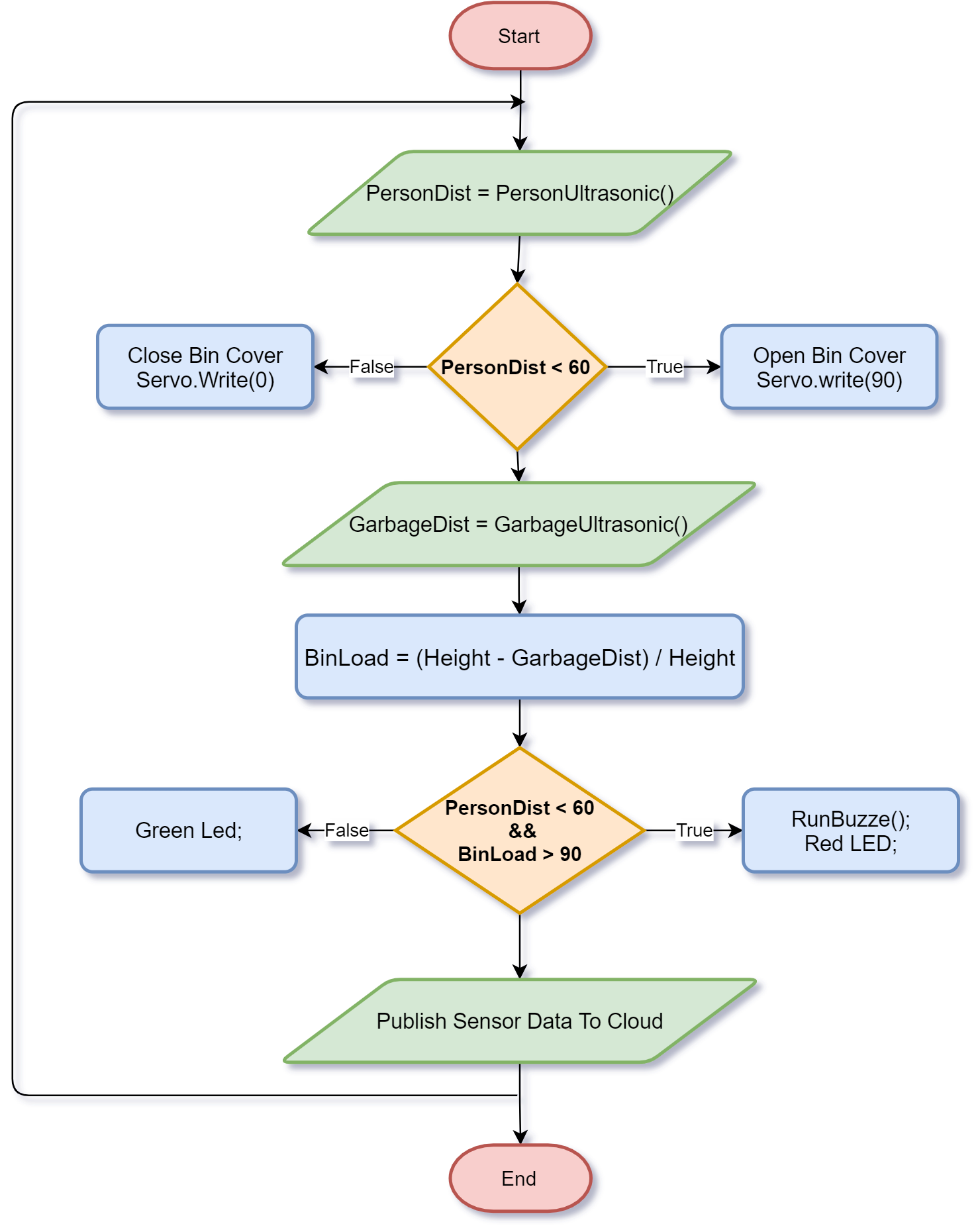


Figure 4.2 Block Diagram

### 4.3 Flow Chart:



The figure shows how the project works in a detailed and clear way from beginning to end

Figure 4.3 flow chart

When the start of project we want to connect between under garbage unit and employee unit through IoT API.

if the system is connected the Wi-Fi module will starts send sensor data to the IoT API which will send data to the employee software.

The figure below represents the flow chart of the smart bin and how it works, after we turn it on, we assume that it is empty for the first use and it waits a trigger of for opening its cover automatically when the ultrasonic sensor reads a distance mora than a predefined constant which is 60cm, if so, it will open the cover by giving a proper angle value to the servo motor, else it will remain closed.

The next step is to measure the bin load by measuring the distance between ultrasonic and top of bin content convert the distance to percentage value by this simple formula:

BinLoad = (Height - GarbageDist) / Height

In addition, store the value in a variable decide if the bin is full and a person close to use it, the LED will light red and the buzzer will be turn on, on the other hand, if there is a capacity, the LED will light green.

Finally, the bin load will be published to the IOT platform to put the data in a dashboard and use this dashboard in our website, and the whole procedure will be in infinite loop as long as the hardware module connected to a power.

### Circuit diagram:

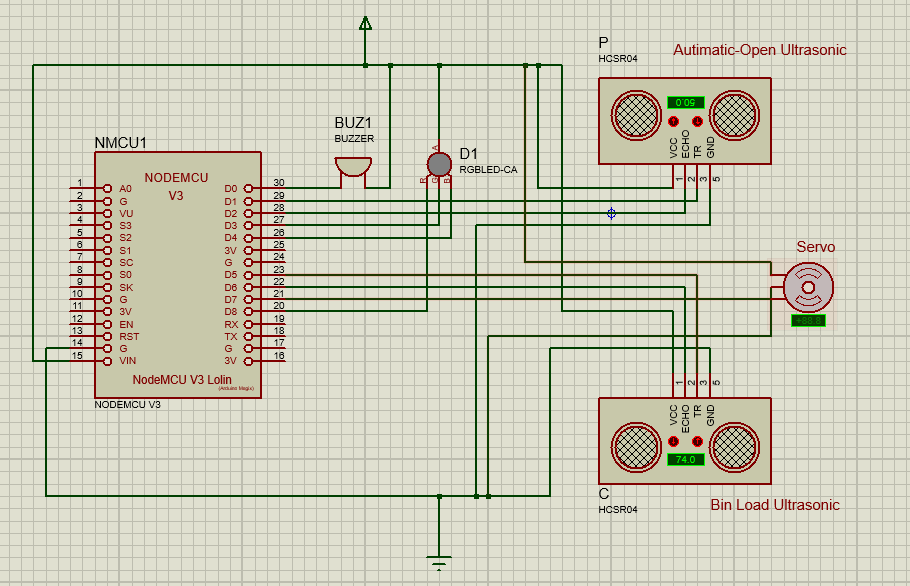


Figure 4.4 Schematics

* 1. **Sequence diagram:**

This section will illustrate another type of UML diagram which can be used to show the interactions between the actors of the system and the system components, and the interactions between these components themselves. This will help us understand whether the proposed system, structure could deliver the required system performance and dependability or not. The following figures show the sequence diagram for each function of the proposed system which will describe the interactions during a particular case.

* + 1. Employee sequence diagram

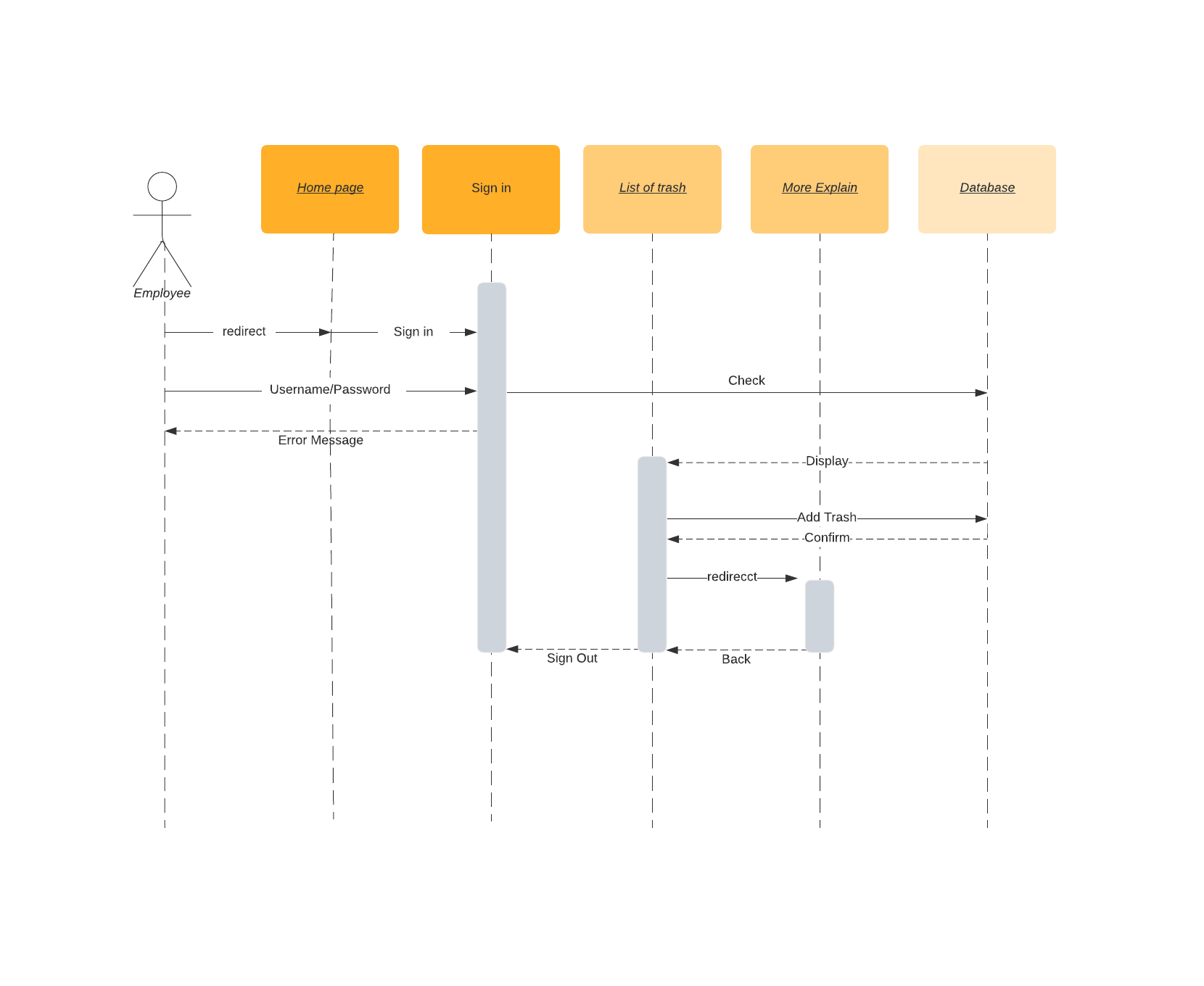
****

Figure 4.5: Employee sequence diagram

* + 1. Admin sequence diagram

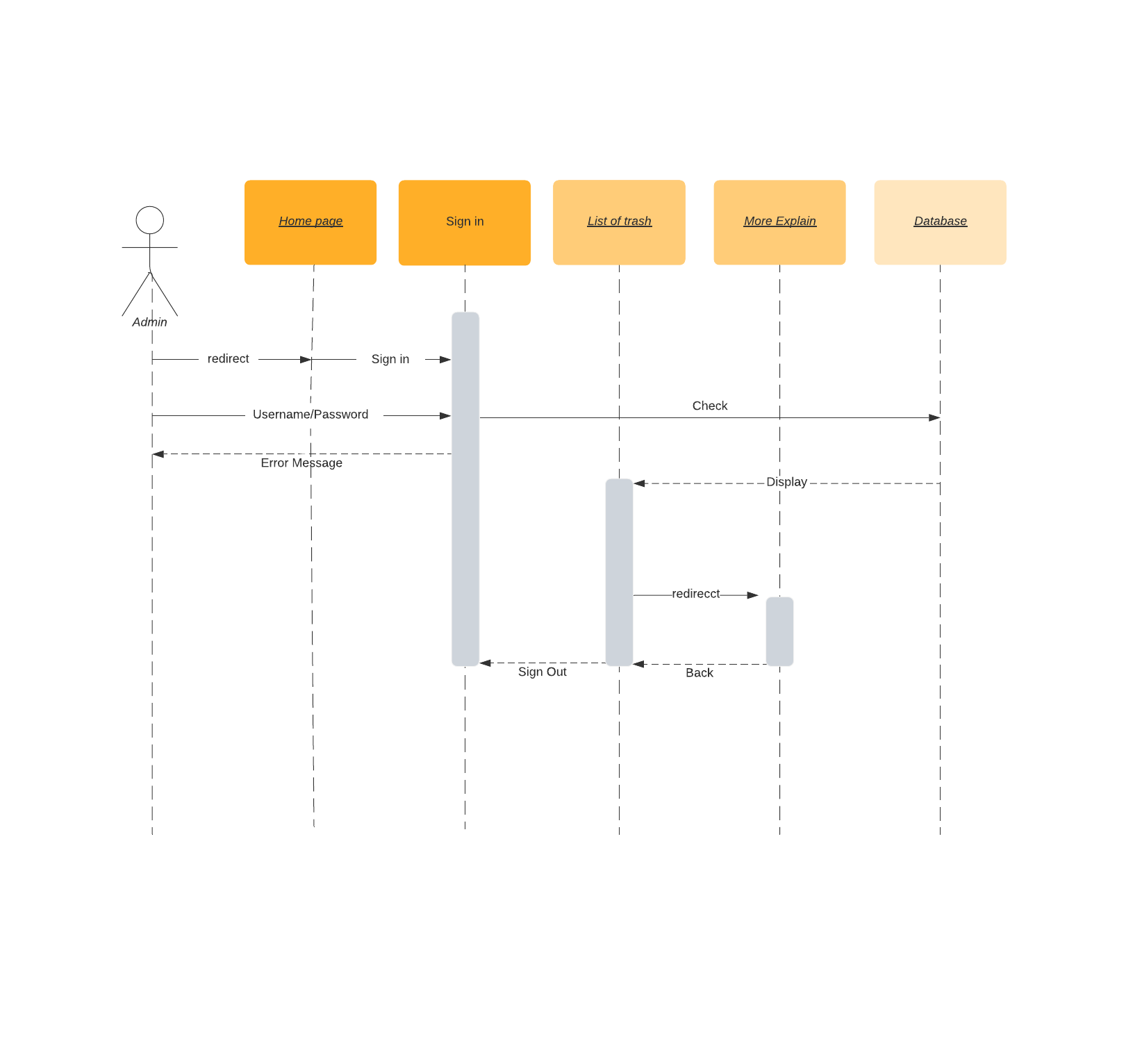
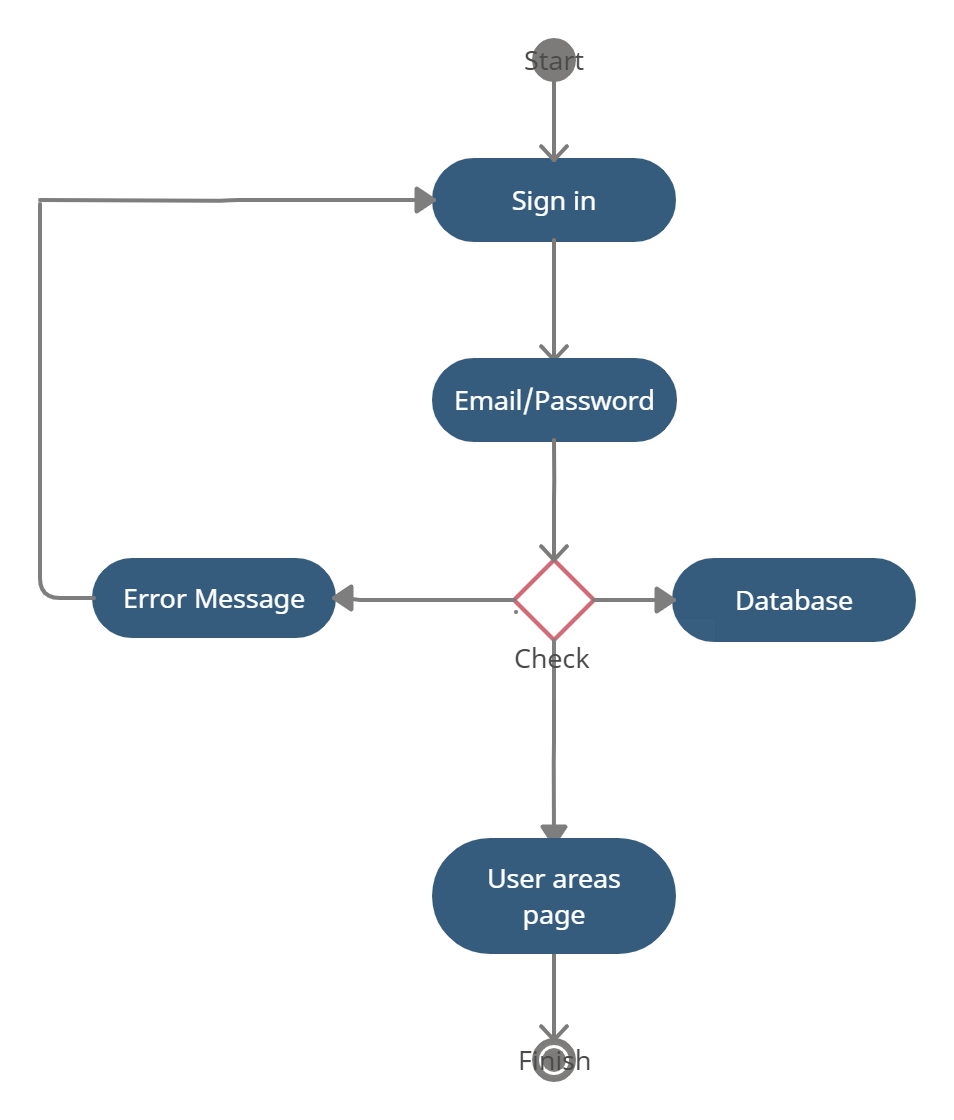
****

Figure 4.6: Admin sequence diagram

* 1. **Activity diagrams:**
     1. Sign in activity

****

Yes

No

Figure 4.7: Sign in activity

* + 1. Employee add trash activity

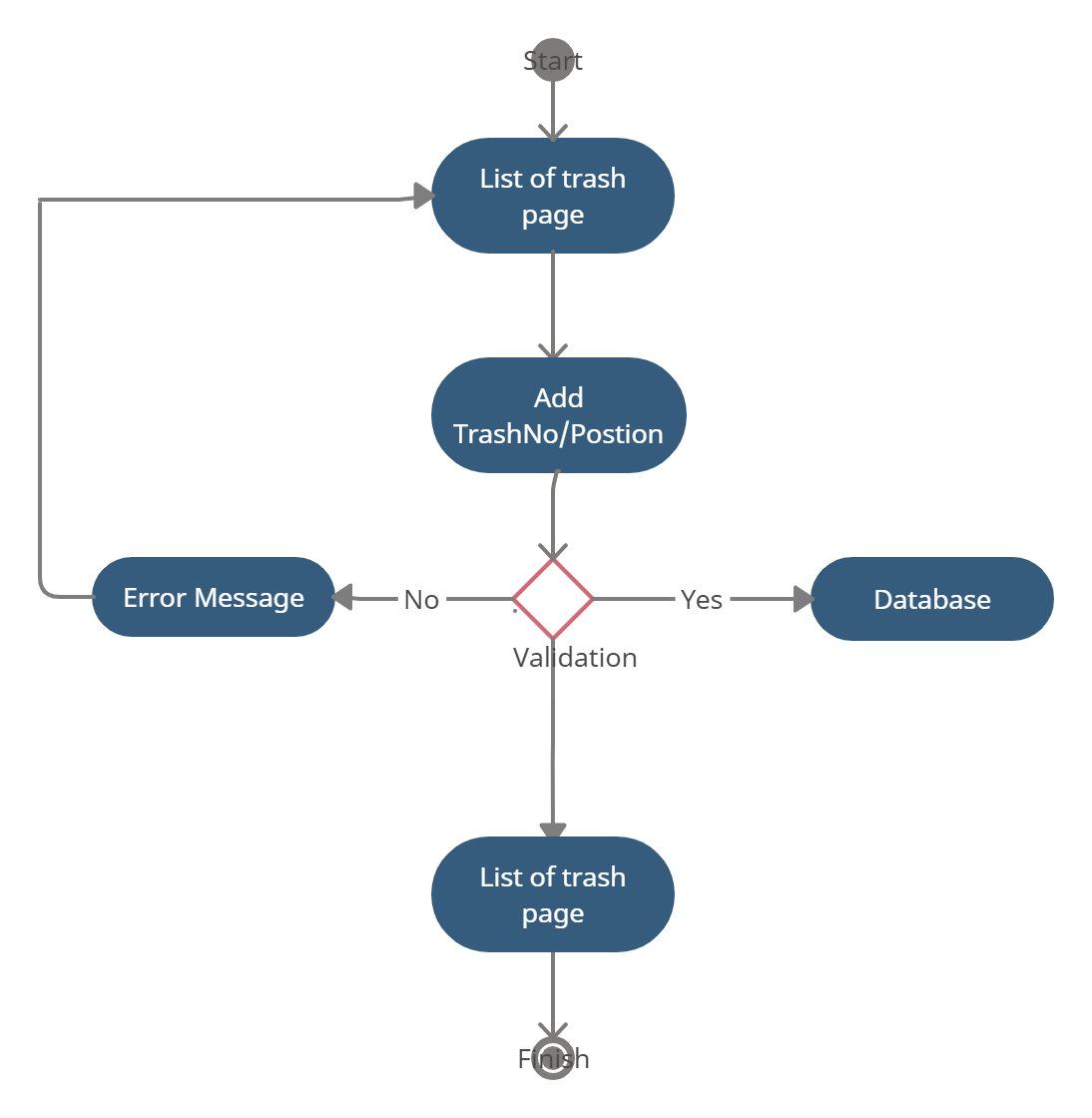
****

Figure 4.8: Employee activity

* 1. **Class Diagram:**

Class diagram is one of the diagram types in the unified modeling language (UML) and it is used to show the organization of the system in terms of object classes that build up the system, and the associations that indicate the relationships between these classes

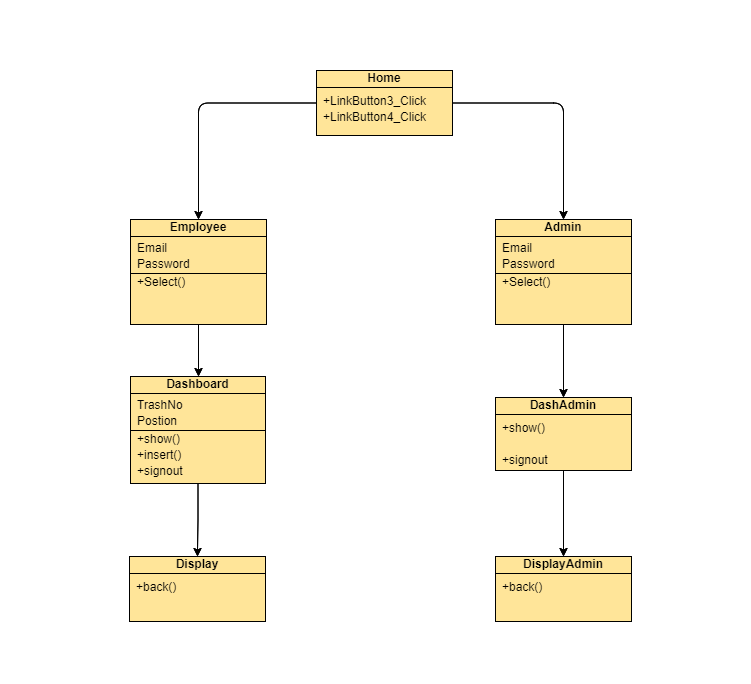
****

Figure 4.9: Class Diagram

* 1. **Conclusion and Recommendations**

We have drawn by making several illustrative diagrams that pertain to the software side, including the Sequence Diagram and Activity Diagrams, which also pertain to the hardware side such as Flowchart and Circuit Diagram.

**Chapter five**

**Coding and Implementation**

|  |  |  |
| --- | --- | --- |
| 5.1 | Introduction | 36 |
| 5.2 | Technical approach | 36 |
| 5.3 | Coding Programming Languages | 36 |
| 5.4 | Data base system | 37+38 |
| 5.5 | Conclusion and Recommendations | 38 |

**5.1 Introduction**

In chapter four, we have described several system modeling diagrams for representing the system functions from different perspectives. These diagrams include the system class diagram, activity diagrams and sequence diagrams.

This chapter will identify the actual coding and implementation for the system’s functions according to the steps that have been described in the previous chapter using design modeling diagrams. The programming and coding environments that have been used for developing the system functions will also be described with their establishment procedures in this chapter.

**5.2 Technical approach:**

We choose to build our project as a web application: using asp.net v4.5 technologies with the support of C# programming language as the runtime language

We build our application using Visual studio 2015 IDE. Visual studio helps us to build front end and back end, write code, link to database, run and debug our application locally.

We also Use the Photoshop to design the logo and optimize photo. Our application can be opened through computers

**5.3 programming language:**

C# programming language:

C# (pronounced see sharp) is a general-purpose, programming language, strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed around 2000 by Microsoft as part of its .NET initiative. C# is one of the programming languages designed for the Common Language Infrastructure (CLI).

**5.4 database:**

A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information.

A database-management system (DBMS) is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases.

Our data base is designed and managed using MS SQL management studio 2015.

MS SQL database has the following features:

• High Security.

• Easy to use and understand code.

• High performance: it’s more efficient by its quick functionalities

• high scalability

The following figure show the MS SQL database for our application.

**5.4.1 Database tables:**

**Table 5.1 Employee database table**

|  |  |  |
| --- | --- | --- |
| no | Field name | data type |
|  | **UserID** | Varchar(50) |
|  | Email | Varchar(50) |
|  | Password | Varchar(50) |

**Table 5.2 Admin database table**

|  |  |  |
| --- | --- | --- |
| no | Field name | data type |
|  | **UserID** | Varchar(50) |
|  | Email | Varchar(50) |
|  | Password | Varchar(50) |

**Table 5.3 Trash info database table**

|  |  |  |
| --- | --- | --- |
| no | Field name | data type |
|  | **TrashNo** | Varchar(50) |
|  | Postion | Varchar(50) |

**5.5 Conclusion and Recommendations**

In this chapter, we explain the technical approach and programming language that is used to build our system. The chapter also describe the different technologies and tool are used with the visual studio 2015 to make the system ready for use.

**Chapter SIX**

**Testing**

|  |  |  |
| --- | --- | --- |
| 6.1 | Introduction | 41 |
| 6.2 | System testing plan | 41 |
| 6.3 | Software Testing | 41-45 |
| 6.4 | Hardware Testing | 45-49 |
| 6.5 | Conclusion and Recommendations | 49 |

**6.1 Introduction**

In chapter five, we have clarified the coding and implementation of the system functions using the ASP.net technology and C# programming language with the help of Visual studio 2015 IDE.

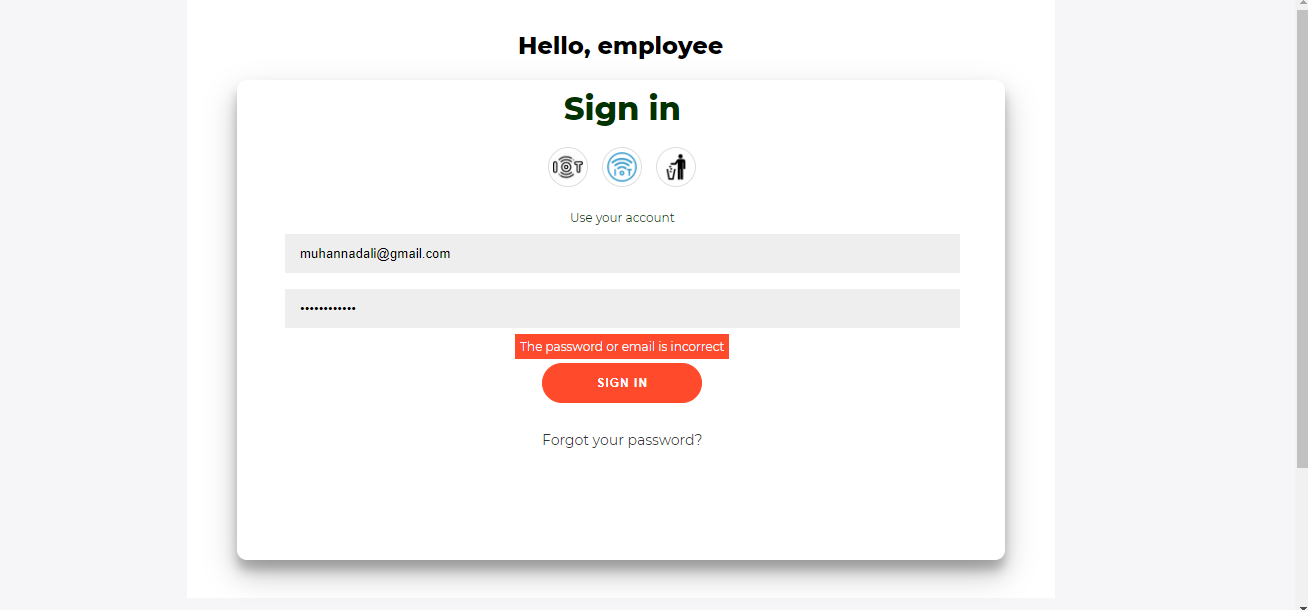
This chapter will illustrate the system testing process. First, the system units’ functions must be tested separately to ensure that it performs as expected, then testing the system components integration to ensure that the system is working fine as a whole. We will explain each test case for each function with all possibility inputs and procedures.

**6.2 Testing plan:**

* **Module and Unit Code Testing:** In this type of testing, each function was tested individually to ensure that it operates as expected, so that each function, code, and operation, will be tested by itself to make sure that this is the right implementation of each function.
* **System Integration Testing:** In this type of testing, the integration of all objects was tested to ensure that the whole system performs as expected, so that the system will tested as a unit to make sure that all operations are integrated with each other and there is a match between all of them.
  1. **Software Testing:**
     1. **Test cases unit testing:**

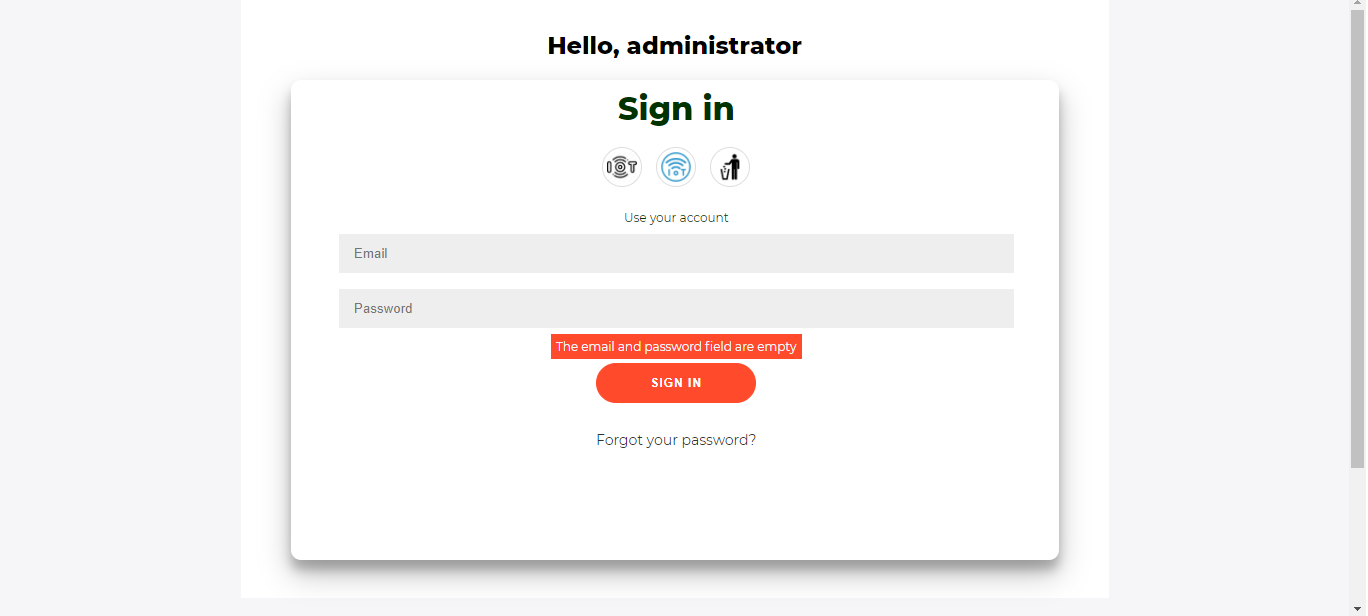
1. Table 6.1 Employee sign in

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Case | Description | Example input | Excepted Output |
| 1 | NC  (Normal Case) Validation testing | Enter correct email  and password | Email: muhnnad.taha@gmail.com  Password: Muhannad#157 | “Welcome To Al Quds  Dormitories!” |
| 2 | EC (error case)  defect testing | User log in  without enter  any data | Email:”  Password: “” | Error message  (The email and  password field  are empty) |
| 3 | EC (error case)  defect testing | User doesn’t enter  any input at  email entry | Email:  Password: Muhannad#157 | Error message  (The email is empty) |
| 4 | EC (error case)  defect testing | User doesn’t enter  any input at  password entry | Email: muhnnad.taha@gmail.com  Password: | Error message  (The password is empty) |
| 5 | EC (error case)  defect testing | User enters invalid  data as email | Email: muh. taha.1548  Password: 164895 | Error message  (The email  must contain only  letters) |
| 6 | EC (error case)  defect testing | User enter email  not exist in DB or  Password incorrect | Email:muhannad[ali@gmail.com](mailto:ali@gmail.com)  Password: Muhannad#157 | Error message  (The password or  email is incorrect) |

Figure 6.1 employee page testing

1. Table 6.2 Admin sign in

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Case | Description | Example input | Excepted Output |
| 1 | NC  (Normal Case) Validation testing | Enter correct email  and password | Email: muhnnad.taha@gmail.com  Password: Muhannad#157 | “Welcome To Al Quds  Dormitories!” |
| 2 | EC (error case)  defect testing | User log in  without enter  any data | Email:”  Password: “” | Error message  (The email and  password field  are empty) |
| 3 | EC (error case)  defect testing | User doesn’t enter  any input at  email entry | Email:  Password: Muhannad#157 | Error message  (The email is empty) |
| 4 | EC (error case)  defect testing | User doesn’t enter  any input at  password entry | Email: muhnnad.taha@gmail.com  Password: | Error message  (The password is empty) |
| 5 | EC (error case)  defect testing | User enters invalid  data as email | Email: muh. taha.1548  Password: 164895 | Error message  (The email  must contain only  letters) |
| 6 | EC (error case)  defect testing | User enter email  not exist in DB or  Password incorrect | Email:muhannad[ali@gmail.com](mailto:ali@gmail.com)  Password: Muhannad#157 | Error message  (The password or  email is incorrect) |

Figure 6.2 Admin page testing

1. Table 6.3 Employee add trash info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Case | Description | Example Input | Excepted Output |
| 1 | NC  (Normal Case) Validation testing | Employee enter all  attribute corrects in system | TrashNo: 11  Position: Ramallah | Successes |
| 2 | EC (error case)  defect testing | Position field  is empty | TrashNo: 11  Position: | Error message:  (Position field  is empty) |
| 5 | EC (error case)  defect testing | Trash number field  is empty | TrashNo:  Position: Ramallah | Error message:  (Trash number field  is empty) |
| 6 | EC (error case)  defect testing | Trash number existing  Already in database | TrashNo:11  Postion: Al-Quds | Error message:  (Trash number existing  Already) |
| 7 | EC (error case)  defect testing | Trash number and Position  field is empty | TrashNo:  Position: | Error message:  ) All field is  empty( |

Figure 6.3: Add trash bins testing

* 1. **Hardware Testing:**
     1. **Development Testing:**

In this stage of testing, the system is tested during development to discover bugs and defects.

1. **Unit testing**: in this low level testing we have tested the functionality of individual system units, these units include a specific method and functions and individual hardware components such as:

* Testing the ultrasonic sensors and how to read distance from it using a specified formula to convert the time to distance considering many test cases such as what to do when the reading is above or below a predefined value like the distance to open the trash and the test case when the object moving is out of the sensor measurable range which leads to incorrect distance values.
* Testing the servo motor and the required current that we need to provide to operate appropriately, for example firstly we have provided a power source with 1 amp which was not enough to powering the motor so we have increased the supplied current to 1.8 amp, another thing we have tested in servo is the angle of rotation when the trash is open and when the trash is closed, after trying many times we conclude that it should be 0 when trash is opened and 90 when the trash is closed.
* Testing the other small components such as LED and the colors that we need and whether it is active low or active high that inform us that our RGB LED is common anode which means that it turns on when it takes LOW signal, and Buzzer and its best tone to choose for it when running.

1. **Component testing:** in this stage, the testing where done to components that consist of several integrated individual units as follows:

* Ultrasonic and servo as a component: testing of this component functionality were done after ensuring that each unit operate properly, this were done by letting the servo moving according to the ultrasonic reading, the first test case is that what it happen if the ultrasonic returned an incorrect value due to far object, this leads to unwanted behavior of the servo, so modified the method to trigger an action if and only if the reading in a specified range.

Another test case is observing the best distance from the trash to trigger an event of opening the trash which we find that 60 cm is the best value that make the person not very close which makes him dirty, or very far which makes a difficulty for the sensor to detect motion.

* Internal ultrasonic and IOT platform as a component: this test involves measuring the level of the bin content and upload the value to the IOT platform in real-time manner, the test case here the whether the data uploaded properly to cloud and this testing was successful when giving the correct network name, password, and the API key.

1. **System testing:** this test will include most of the system components integrated to each other, we have done this testing when we connected all the system part but before installing the hardware to the trash body, note that still part of the development testing, this test was done according to the following test cases:

* What happen if there is no WIFI connection? this system must not work at all if there is no connection to the internet and the IOT platform, after inspection of this point we have recognized that this consideration has successful satisfied and the system will continuously try to connect to internet before go through the code of opening the trash cover for example.
* When the buzzer will run? The buzzer was intended to be run as a response to the internal sensor only, in other words, when the trash become full, after doing test we have find that it should run when the person come and the trash is full, this condition include two components of the system which is the two ultrasonic sensors, this condition was successfully tested.
  + 1. **Release Testing:**

This stage of testing occurs on a particular release of a system that is intended to use after the developing complete, this type of testing is a black-box testing where tests are derived the system specification.

In our project we have done this testing after installing all the hardware components on the system prototype and test according to the specified requirements as following tests:

* **Open the trash automatically:** the following two pictures show how the trash will open automatically when a person closes to it and the sensor detect a distance of less than 60 cm.



Figure 6.4: Open the trash automatically 1



Figure 6.5: Open the trash automatically 2

* **System alert:** as shown in the previous images the green LED indicates that the trash is not full, but in the following images the LED lights red when the inside ultrasonic indicates a level of more than 90%, and if anyone closes to the trash when it in full state, the buzzer will generate a sound to alert the user that the trash is full.



Figure 6.6: System alert

* **Monitoring the system:** we have tested that the level of the trash uploaded and varying continuously in real-time to the web page as shown in the following figure.



Figure 6.7: Monitoring the system

**6.5 Conclusion and Recommendations**

In this chapter the system was tested against its requirements specifications and ensured that each operation is operating as expected, the integration between all objects is operate as expected, the system is working correctly and this is clear according to its results and the system is operating as a unit and performing as expected.

**Chapter Seven**

**Conclusions and future works**

|  |  |  |
| --- | --- | --- |
| 7.1 | Conclusions | 51 |
| 7.2 | Future works | 51 |
| 7.3 | References | 51 |
| 7.4 | Appendix: C # code | 52-54 |
| 7.5 | Appendix Arduino code | 55-59 |

**7.1 Conclusions**

This web application has been presented, described and evaluated successfully. and installed suitable circuit After developing its functions, we conclude that it satisfies the requirements and the objectives correctly as intended.

Such project challenging for us to design and build, the whole thing is about learning new skills and time management especially during Covid 19 pandemic.

Finally, this web application can be hosted online with any host company so that employees and competent authorities such as mayors can use it immediately, and smart garbage bins can be placed in smart cities and neighborhoods that need it most.

* 1. **Future works**

For future work, we recommend and suggest the following:

* Online hosting and domain registering for this web application
* Front end enhancement and make it more attractive
* Plan and implement Marketing campaign to increase popularity and traffic of the site
* Native mobile apps development for the site (iOS, android)
* Make trash can underground.
* The trash can send its geolocation.
  1. **References**

1. <https://en.wikipedia.org/wiki/C_Sharp_(programming_language)>
2. <https://www.youtube.com/results?search_query=asp.net+web+application+tutorial>
3. <https://www.ecubelabs.com/>
4. <https://nordsense.com/>
5. <https://bigbelly.com/about/>
   1. **Appendix: C # code**

**Sign in code:**

public partial class \_Default : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

LblErMe.Visible = false;

}

protected void Button1\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS12;AttachDbFilename=C:\Project\App\_Data\Database.mdf;Integrated Security=True;User Instance=True");

con.Open();

SqlCommand cmd = new SqlCommand("SELECT COUNT(1) FROM tblUs WHERE username =@username AND password=@password", con);

cmd.Parameters.AddWithValue("@username", TextBox1.Text.Trim());

cmd.Parameters.AddWithValue("@password", TextBox2.Text.Trim());

int count = Convert.ToInt32(cmd.ExecuteScalar());

if (count == 1)

{

Session["username"] = TextBox1.Text.Trim();

Response.Redirect("Dashboard.aspx");

}

else if(TextBox1.Text=="" && TextBox2.Text == ""){

LblErMe.Text = "The email and password field are empty";

LblErMe.Visible = true;

}

else if (TextBox1.Text == "")

{

LblErMe.Text = "The email is empty";

LblErMe.Visible = true;

}

else if (TextBox2.Text == "")

{

LblErMe.Text = "The password is empty";

LblErMe.Visible = true;

}

else if (count == 0)

{

LblErMe.Text = "The password or email is incorrect";

LblErMe.Visible = true;

}

else

LblErMe.Visible = true;

}

}

**List of trash:**

public partial class Dashboard : System.Web.UI.Page

{

SqlDataAdapter da;

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)

{

Response.Redirect("Default.aspx");

}

protected void Button1\_Click(object sender, EventArgs e)

{

Response.Redirect("Display.aspx");

}

protected void Button3\_Click(object sender, EventArgs e)

{

Response.Redirect("Default.aspx");

}

protected void DataList1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

protected void Button1\_Click1(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS12;AttachDbFilename=C:\Project\App\_Data\Database.mdf;Integrated Security=True;User Instance=True");

SqlDataAdapter da = new SqlDataAdapter("SELECT TrashNo FROM TrashDe WHERE TrashNo='" + TextBox1.Text + "' ", con);

DataTable dt = new DataTable();

da.Fill(dt);

if (TextBox1.Text == "" && TextBox2.Text == "")

{

Label1.Text = "All field are empty";

Label1.Visible = true;

}

else if (TextBox1.Text == "")

{

Label1.Text = "Trash number field is empty";

Label1.Visible = true;

}

else if (TextBox2.Text == "")

{

Label1.Text = "Postion field is empty";

Label1.Visible = true;

}

else if (dt.Rows.Count >= 1)

{

Label1.Text = "Trash number existing already";

Label1.Visible = true;

}

else

{

SqlCommand cmd = new SqlCommand("INSERT INTO TrashDe (TrashNo,Postion) VALUES (@TrashNo,@Postion)", con);

con.Open();

cmd.Parameters.AddWithValue("@TrashNo", TextBox1.Text);

cmd.Parameters.AddWithValue("@Postion", TextBox2.Text);

cmd.ExecuteNonQuery();

Label1.Text = "Successed";

Label1.Visible = true;

}

}

}

* 1. **Appendix Arduino code:**

#include "UbidotsESPMQTT.h"

#include <Servo.h>

#define TOKEN "BBFF-t3H037T2SSs2aVkuZuLKJKMJmfBYqT" // Your Ubidots TOKEN

#define WIFINAME "Redmi" // Your SSID

#define WIFIPASS "12312312313" // Your Wifi Pass

const byte SERVO\_PIN = D7;

const byte BUZZER\_PIN = D0;

const byte PERSON\_TRIG\_PIN = D1; //can be D3

const byte PERSON\_ECHO\_PIN = D2;

const byte CONTENT\_TRIG\_PIN = D5; //can be D4

const byte CONTENT\_ECHO\_PIN = D6;

const byte RED\_PIN = D4;

const byte GREEN\_PIN = D3;

const byte TRASH\_HEIGHT = 50;

const byte DISTANCE\_TO\_OPEN = 60;

Ubidots client(TOKEN);

Servo myservo;

int binLoad;

void setup() {

Serial.begin(115200);

client.setDebug(true); // Pass a true or false bool value to activate debug messages

client.wifiConnection(WIFINAME, WIFIPASS);

client.begin(callback);

pinMode(CONTENT\_TRIG\_PIN, OUTPUT);

pinMode(CONTENT\_ECHO\_PIN, INPUT);

pinMode(PERSON\_TRIG\_PIN, OUTPUT);

pinMode(PERSON\_ECHO\_PIN, INPUT);

pinMode(BUZZER\_PIN, OUTPUT);

pinMode(RED\_PIN, OUTPUT);

pinMode(GREEN\_PIN, OUTPUT);

myservo.attach(SERVO\_PIN);

binLoad = 0;

//client.ubidotsSubscribe("nodemcu-1", "content-level"); // Insert the dataSource and Variable's Labels

}

void loop() {

contentLevel();

if (!client.connected()) {

client.reconnect();

//client.ubidotsSubscribe("nodemcu-1", "content-level"); // Insert the dataSource and Variable's Labels

}

client.add("content-level", binLoad);

client.ubidotsPublish("nodemcu-1");

client.loop();

delay(1000);

senseMotion();

runLED();

}

void contentLevel() {

long duration, distance;

//int level;

digitalWrite(CONTENT\_TRIG\_PIN, LOW);

delayMicroseconds(2);

digitalWrite(CONTENT\_TRIG\_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(CONTENT\_TRIG\_PIN, LOW);

duration = pulseIn(CONTENT\_ECHO\_PIN, HIGH);

distance = (duration/2) / 29.1;

if (distance > 50 || distance <= 0){

Serial.println("Person out of range");

}

else{

binLoad = (TRASH\_HEIGHT - distance) \* 2;

}

}

void senseMotion(){

// the distance result in centimeters:

long duration, cm;

digitalWrite(PERSON\_TRIG\_PIN, LOW);

delayMicroseconds(2);

digitalWrite(PERSON\_TRIG\_PIN, HIGH);

delayMicroseconds(20);

digitalWrite(PERSON\_TRIG\_PIN, LOW);

duration = pulseIn(PERSON\_ECHO\_PIN, HIGH);

// convert the time into a distance

cm = (duration/2) / 29.1;

// the condition for the distance

if ( cm <= DISTANCE\_TO\_OPEN && binLoad < 90 ){

myservo.write(0);

delay(3000);

}

else if(cm <= DISTANCE\_TO\_OPEN && binLoad >= 90 ){

myservo.write(0);

runBuzzer();

delay(3000);

}

else{

myservo.write(90);

delay(100);

}

Serial.print(cm);

Serial.print("cm");

Serial.println();

delay(100);

}

void runBuzzer(){

digitalWrite(BUZZER\_PIN, HIGH);

delay(1000);

digitalWrite(BUZZER\_PIN, LOW);

delay(500);

digitalWrite(BUZZER\_PIN, HIGH);

delay(1000);

digitalWrite(BUZZER\_PIN, LOW);

delay(3000);

}

void runLED(){

if(binLoad < 90){

digitalWrite(RED\_PIN, HIGH);

digitalWrite(GREEN\_PIN, LOW);

}

else{

digitalWrite(RED\_PIN, LOW);

digitalWrite(GREEN\_PIN, HIGH);

}

}