Unit Code : CIS016-2

Assignment : Control an Elevator

Unit Coordinator : Krishna Aryal

Student Name : Santosh Tamang

University ID : 2147440

University E-mail : [Santosh.Tamang@study.beds.uk.ac](mailto:Santosh.Tamang@study.beds.uk.ac)

Table of Contents

[Introduction 3](#_Toc150291683)

[Problem Statement 3](#_Toc150291684)

[Proposed Solution 3](#_Toc150291685)

[Aims and Objectives 4](#_Toc150291686)

[Project Plan 4](#_Toc150291687)

[System Requirement 5](#_Toc150291688)

[System Design 6](#_Toc150291689)

[Use Case Diagram 6](#_Toc150291690)

[Activity Diagram 7](#_Toc150291691)

[ER Diagram 8](#_Toc150291692)

[Language Used 8](#_Toc150291693)

[Database Used 9](#_Toc150291694)

[Prototype 9](#_Toc150291695)

[Testing and Evaluation Strategy 9](#_Toc150291696)

[Implementation 10](#_Toc150291697)

[Testing 10](#_Toc150291698)

[Critical Evaluation 12](#_Toc150291699)

[Conclusion 13](#_Toc150291700)

[References 13](#_Toc150291701)

[Appendix 14](#_Toc150291702)

## Introduction

The project is to build an object-oriented software control application to control an elevator for two -floor office building. Currently, all the employees use stair for any cases, so for making the life of the employees easier, the office wants to equip the building with an elevator.

The elevator consists of an elevator car in which there is a control panel having buttons and display-window, and elevator shift and request button panel one on each floor, a door that opens upon the arrival at the requested floor and close before the elevator departure from the floor. The door is closed to protect the passengers from being injured by brushing against the wall of the elevator shaft.

When the request button is pressed at either floor, the elevator will towards the floor and the door automatically opens. When the floor button is pressed, the door will close automatically, and elevator will move towards the floor. When the elevator reaches the destination floor, it will stop, and the doors will open automatically, and the door will close after a while.

## Problem Statement

The project is to build an elevator control software that is based on object-oriented approach. The software should have efficient passenger movement and scalability. There should be GUI of elevator. The GUI should work as an actual elevator would work. The log should be stored and display.

## Proposed Solution

For the solution of the above-mentioned problem statement, the proposed solution is:

* Task 1 (GUI Implementation)

For the testing of the logic that will be created for the elevator system, a GUI will be created. The GUI will consists of a call button, floor buttons, display panel and a log button to view the log.

* Task 2 (Main Logic)

To handle all the events that comes from the real elevator, the main logic will be created, and the logic will be tested in the created GUI.

* Task 3 (Database Integration)

To store and display all the logs a database will be integrated in the main logic. The database will be MySQL.

* Task 4 (Animation)

Create the effects that happens in real life elevators in the virtual elevator that we will create.

* Task 5 (Optimization)

Try to optimize the whole code if possible.

## Aims and Objectives

The main aim of this project is to develop capabilities and skills of solving real-world problems with C# programming language.

The objectives are :

* To create user-friendly GUI for the elevator system.
* To make the software responds according to the user’s interactions.
* To use database for the storing of the logs.

## Project Plan

|  |  |  |
| --- | --- | --- |
| Task No. | Tasks | Priority |
| 1 | Planning and GUI | MUST |
| 2 | Main Logic Development | MUST |
| 3 | Initial Testing | MUST |
| 4 | Optimization and Testing | MUST |
| 5 | Documentation | MUST |
| 6 | Review and Submission | MUST |

## System Requirement

Development Environment

1. Operating System

The software was developed using Window OS.

1. Integrated Development Environment

The IDE used was Visual Studio 2022

1. Framework

.Net Framework was used to create the system.

Functional Requirement

* The buttons should function properly.
* The doors should function properly.
* The display should show corresponding values.
* Log button should function properly.

Non-Functional Requirement

* Lift Management System should preform smoothly.
* Lift Management System should be scalable.
* Lift Management System should be portable.

Usability Requirement

* The GUI should be user-friendly.

## System Design

### Use Case Diagram

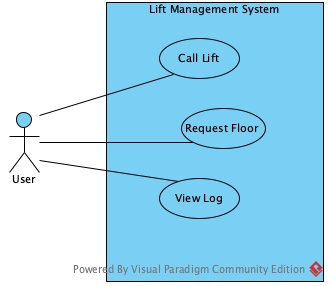


Fig.1 Use Case Diagram

(Paradigm, 2023)“A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped.”

The use case diagram shown in above figure that demonstrates the core functionality of the system. Here we can see, there is only one actor i.e., user. This image demonstrates that the user can call lift, request floor and view logs.

### Activity Diagram

A diagram of a process

Description automatically generated

Fig.2 Activity Diagram

(Paradigm, 2023)“Activity diagram is an advanced version of flow chart that modeling the flow from one activity to another activity”

Above figure shows the active activity diagram for the elevator management system. The user calls the elevator, requests floor and view logs, all the logic part is hidden from the user.

### ER Diagram

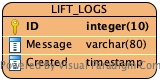


Fig.3 Entity Relation Diagram

(Biscobing, 2019)“An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.”

To make ERD following entity was created for elevator management system i.e., Lift\_Logs. Since there is only one entity there is no relation to maintain.

## Language Used

C# was used in the creation of the software. C# is a versatile high level programming language developed by Microsoft. Dot Net Framework was used which was also created by the Microsoft. Using the two, creation of application for the windows was easy.

## Database Used

For the creation of the software, MySQL was used to store and read the logs of the elevator. Tool to connect MySQL with C# was used.

## Prototype

The above figures are the ones that is opened and closed.

## Testing and Evaluation Strategy

To ensure the reliability and functionality of the system testing is required at various phases. The strategy to test and evaluate the system are :

* Door Operation Test

Ensure secure and smooth closing and opening of the door.

* Floor Transition Test

Confirming accurate floor transition upon selection.

* Historical Data Access Test

Access and display the historical data.

* Performance Test

Testing the system under different loads and conditions.

## Implementation

The project was to develop a software that controlled the elevator. For the development of the project, Microsoft Visual Studio 2022 IDE, C# language, Dot Net Framework and MySQL Database were used. To connect MySQL with the C# another tool was used.

First of a GUI for the lift was created. The IDE supports drag and drop to help create GUI faster and easier. The GUI consisted of a display panel, request button and floor buttons. The scalability of the floors was achieved that allowed users to have any number of floors without any problems.

Then started the logical parts, to handle all the events coming from the GUI and database was created with the required table. The database was integrated with the system to store and retrieve the data to display.

When the elevator call button was pressed, the elevator arrives, updates the floor in display and the doors open automatically if left untouched it will close automatically, else the elevator goes towards at the targeted floor, update floor in display and the doors open automatically and if left untouched it closes automatically. There is a log detail button when pressed displays all the log of the elevator.

## Testing

Test Case No : 1

Case Description : Call elevator to current floor

Created and Reviewed by : Santosh Tamang

Test By : Santosh Tamang

Test Result : Pass

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STEP | DESCRIPTION | EXPECTED OUTPUT | ACTUAL OUTPUT | RESULT | ERROR | ERROR STATUS |
| 1 | Start the elevator by pressing the request button | Elevator will come to current floor | Elevator arrived at current floor | Pass | No | No issues |
| 2 | Floor indicator update | Floor indicator display will current floor | Floor indicator displayed current floor | Pass | No | No issues |
| 3 | Door open | Door will open | Dorr opened automatically | Pass | No | No issues |
| 4 | Door close | Door will close if left untouched | Door closed automatically when left untouched | Pass | No | No issues |

Test Case No : 2

Case Description : Go to targeted floor

Created and Reviewed by : Santosh Tamang

Test By : Santosh Tamang

Test Result : Pass

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STEP | DESCRIPTION | EXPECTED OUTPUT | ACTUAL OUTPUT | RESULT | ERROR | ERROR STATUS |
| 1 | Pressing the floor button to go to that floor | Close the door  Arrive at the targeted floor | Door closed automatically  Arrived at the targeted floor | Pass | No | No issues |
| 2 | Floor indicator update | Floor indicator display will current floor | Floor indicator displayed current floor | Pass | No | No issues |
| 3 | Door open | Door will open automatically | Dorr opened automatically | Pass | No | No issues |
| 4 | Door close | Door will close if left untouched | Door closed automatically when left untouched | Pass | No | No issues |

Test Case No : 3

Case Description : View all the logs

Created and Reviewed by : Santosh Tamang

Test By : Santosh Tamang

Test Result : Pass

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STEP | DESCRIPTION | EXPECTED OUTPUT | ACTUAL OUTPUT | RESULT | ERROR | ERROR STATUS |
| 1 | Pressing log detail button displays all log | Displays all logs | Displayed all logs | Pass | No | No issues |

## Critical Evaluation

The project was completed on time, as the required knowledge was given to us prior to the task assignment. The project to control an elevator was created and all the requirements were fulfilled.

The main problem I faced was that Dot Net Framework was not support in MacOS. As I have been using MacOS for more than a year, the unsupported Dot Net Framework possessed a great threat. I tried various options like download and creating a virtual which failed, using XCode interface while coding in Visual Studio which also failed, etc. So, to solve all these problems, I used my friends device (a Window device) and would you believe me it was so much easier than it looked while I was searching for the ways to use Dot Net Framework in my device (a Mac device).

First of all, the GUI was created by drag and drop feature of the Visual Studio 2022 IDE. As GUI was created using drag and drop, the time required for the creation of the GUI was very less i.e., one day. Then started the logical part, headache part. The logical part used most of the time. The logical was rewritten many times so as to refactor and optimize.

The concept of multi-threading was new nevertheless, it was used and was the root of all problems and bugs. The help form my friends was required, even as they were doing their own, they helped me. While doing the project, we all helped each other in positive way.

As the elevator was moving as it was supposed to move, the storing of the logs was required, a MySQL database was created. The connect between the system and the database was created and the storing and reading the stored log was possible.

## Conclusion

The project to create elevator management system in object-oriented approached for two floor office building was completed. For the project C# language, Dot Net Framework, Visual Studio 2022 and MySQL database was used. The allotted time enough to complete the project. The project can more be improvised in the future.

Overall the project was created on time with all the requirement fulfilled and even if there were problems in my way I solved them

# References

Paradigm, V., 2023. *Visual Paradigm.* [Online]   
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/  
[Accessed 2 Nov 2023].

Paradigm, V., 2023. *Visual Paradigm.* [Online]   
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/  
[Accessed 2 Nov 2023].

Biscobing, J., 2019. *Tech Target.* [Online]   
Available at: https://www.techtarget.com/searchdatamanagement/definition/entity-relationship-diagram-ERD  
[Accessed 2 Nov 2023].

## Marking Matrix With Assessment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task Number** | **Sub-tasks** | **Possible Marks** | **Self-assessment (completed Yes/No)** | **Reference to your testing report** | **Mark Awarded** |
| **Task 1** | Complete GUI for Task 1 | 10 | YES |  |  |
| Skeleton of event handlers in place for all buttons | 10 | YES |  |  |
| **Task 2** | All event handlers are functional | 10 | YES |  |  |
| **Task 3** | Database (DB) is designed and can be connected | 5 | YES |  |  |
| Log Information can be retrieved from DB and displayed in the GUI | 5 | YES |  |  |
| When the log button is pressed, log information is sent to and stored in the DB | 5 | YES |  |  |
| Use the disconnected model rather than connected model (Data source is updated via DataAdapters Update() method instead of ExecuteNonQuery() method) | 5 | NO |  |  |
| Using relative path instead of absolute path | 5 | YES |  |  |
| Avoiding any duplication among the event handlers over the database related functions | 5 | YES |  |  |
| **Task 4** | Events described in Task 2 animated using delegation and timer | 10 | YES |  |  |
| **Task 5** | Eliminating logical errors and handling exceptions with try and catch | 5 | NO |  |  |
| Optimise the efficiency of GUI by implementing multiple tasks concurrently via BackgroundWorker | 5 | NO |  |  |
| Use state patterns instead of if-else statements to accommodate future changes of the requirement | 10 | NO |  |  |
| **Task 6** | Testing report | 10 | YES |  |  |
| **Total** |  | 100 |  |  |  |

## Appendix