Design Document

Materials List Estimator

Group 4

SDEV240 Fall 2023

Contents

[Introduction 3](#_heading=h.26in1rg)

[Architecture Design Diagram 4](#_heading=h.lnxbz9)

[Use Cases 6](#_heading=h.35nkun2)

[UML Diagram 8](#_heading=h.1ksv4uv)

[Test Cases 10](#_heading=h.44sinio)

[Summary 12](#_heading=h.2jxsxqh)

# Introduction

For this project, we will be building a materials list estimator for a construction company. It will have a GUI application that estimates the cost of each material and puts it into a table to see the grand total of the house materials. The user will be able to populate item description, number of pieces, and unit price for which the application will then calculate the total cost. The user can clear the form, save to a file, and press a calculate button for the total. Our project additionally will have category subtotals and an email link to send the quote. This project will be completed using C# and developed in Visual Studio.

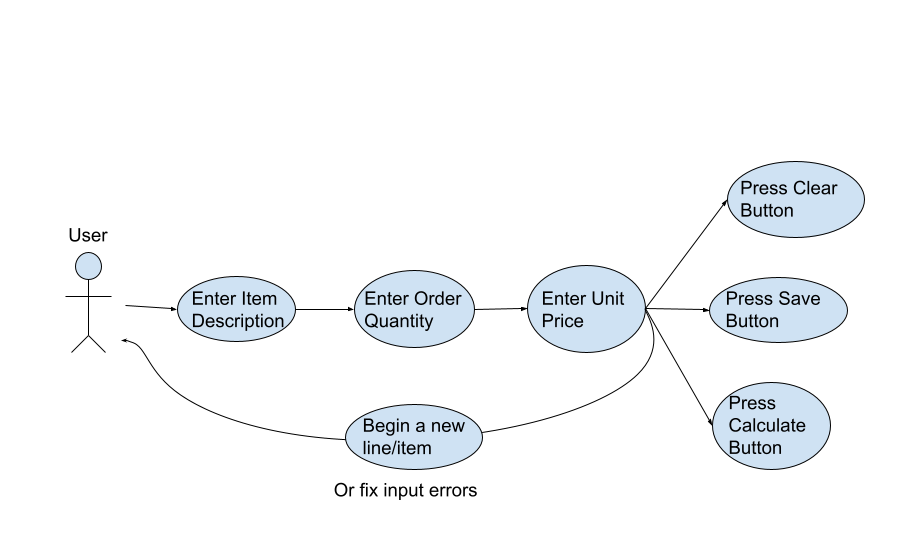
# Architecture Design Diagram

Below is the architecture design diagram we plan to use for the Materials List Estimator. This team is using a layered architecture approach. This consists of five layers. The first layer would be the system support, which would be from the Windows operating system. The next layer would be the data handling, which would consist of all the necessary calculations and actions that result from user input, such as the effects of buttons when they are pressed. The third layer be the even handlers layer, which consists of the users input via button clicks. Then there is the data structure layer, which would be the data table with all the inputted information about whatever materials have been added. The final layer would be the GUI, which is in the form of a windows form from Microsoft Visual Studio.



# Use Cases

The diagram below represents the use cases for the user. The user will first enter the name/description of the first material, then proceed to enter the quantity of that material that is being ordered, and finally enter in the unit price of the material. The user can then either begin a new line for a new material or correct any possible input errors, or press any of the three buttons. The user can press the clear button, which would clear the entirety of inputted information, the save button, which would save the table of inputted information to the user’s device, or the calculate button, which would calculate the total cost of the materials based on the users input.



# UML Diagram

The following is a unified modeling language diagram for how the user or actor interacts with the estimater application. The actor will fill in whatever information is needed or wanted into the data table on the form. The actor will then press any of the three buttons available. This will send the signal to the data handler to perform the action that corresponds with the button that was pressed, such as clearing the table; if the button that was pressed was one such as the calculate button, the information filled out in the table will also be pushed to the data handler to be processed there. Finally, this will then be sent to the GUI, a windows form, to be displayed to the user/actor.

# 

# Test Cases

The following table shows the different test cases that we will need to examine during the testing phase of our project. These include testing for all buttons, calculations, subtotals, import and export functionalities, and data validation.

| Test case ID | Test case description | Test steps | Expected result |
| --- | --- | --- | --- |
| TC\_Clear | See if pressing the clear button will blank out all inputs. | * Press the clear button. | All inputs are cleared. |
| TC\_Save | Verify that inserted values are saved after pressing the save buttons. | * Press the save button. * Close program. * Reopen program. | All values after reopening are the same as before closing. |
| TC\_Calculate | The calculate button creates the expected result. | * Input values * Hit calculate | Calculated values should match the manual calculation. |
| TC\_Import | Values from a file are loaded into the program | * Use import template. * Create file. * Upload using the import function. | All values after importing should reflect the original file. |
| TC\_Export\_01 | Values from the program should be saved to a file. | * Hit the export button * Verify values | All values from the file should reflect the ones in the program. |
| TC\_Export\_02 | Values from the program should be populated into a draft email. | * Hit export to email. * Verify values in email. | All values from the email should reflect the ones in the program. |
| TC\_Subtotals | Categories should have subtotals of the correct value. | * Enter values for different categories. | Subtoals for each category should match manual calculations. |
| TC\_DataValidation | All data fields should check that inputs are of the correct kind. | * Input data in all input boxes of the wrong kind. (EX. price = “green”) | An error message should appear. |

# Summary

This design document outlines the plan for a materials list estimation application intended for a construction company. It includes components like a UML diagram, architectural design diagram, test cases, use cases, and more. The application will feature a user-friendly Graphical User Interface (GUI) for inputting material details, quantities, and prices into a table. It will also offer convenient buttons for tasks such as clearing data, saving the table, and calculating the total cost. Development will follow a layered architecture approach, with distinct layers for system support, data management, event handling, data structures, and the user interface. The project will be implemented using C# in Visual Studio, with the user interface designed as a Windows Form.