Deliverable 2
Design Document

404 NOT FOUND

Matthew Chan
(matthew417@gmail.com)

Zhe Li
(lz897825130@gmail.com)

Gordon McCreary
(gord@uw.edu)

Ken Gil Romero
(kgmr@uw.edu)

Tammy Vo
(votammy97@gmail.com)

https://404github.github.io/404notfound_TCSS360/

Introduction

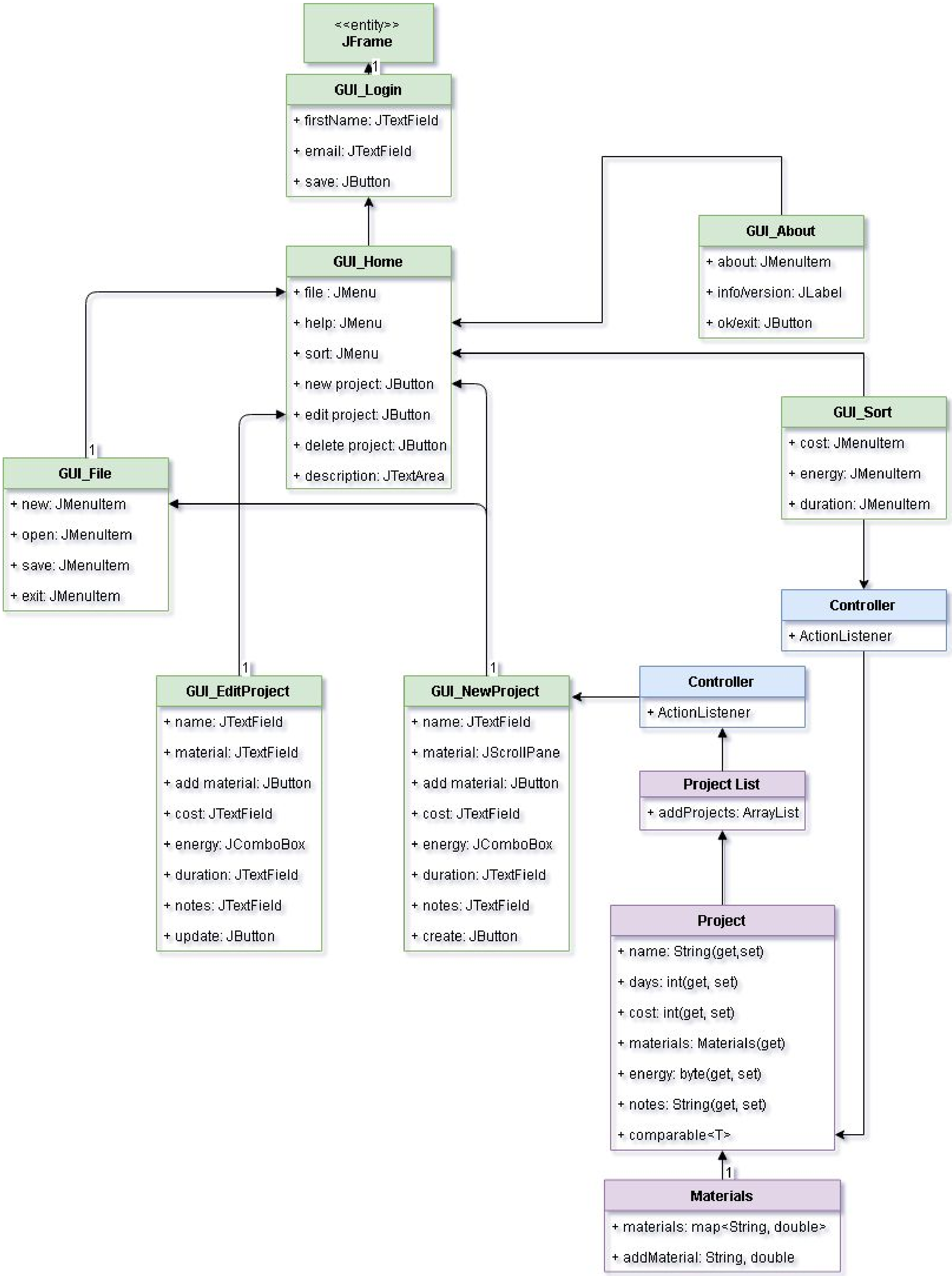
The purpose of our application is to allow DIYers to manage their projects. We will accomplish this by allowing the DIYer to create and delete projects which will be stored in a list that can be sorted by desired categories. Each project will store information about total cost and time cost as well as energy efficiency and required materials. This will enable the DIYer to compare and prioritize their projects.

The goal of our teams design is to keep the application simple and intuitive while providing all the desired functionality. We will do this by basing our user interface around a ‘home page’ that can’t be navigated away from, while minimizing pop up windows. The purpose of this is to keep the navigation of our app easy and keeping the DIYer from getting lost in the application. We ultimately want to make performing any task on our application instinctive. We will accomplish this by using common user interface elements and visual composition that guides use.

Table of Contents

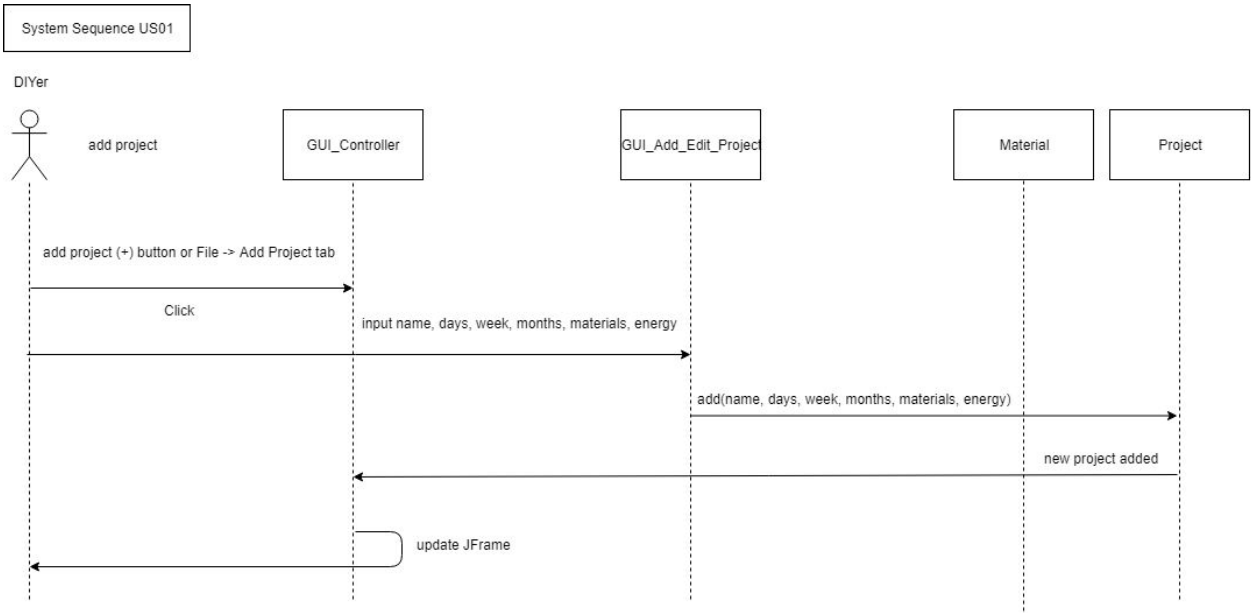
Rationale Summary:

Class Diagram:



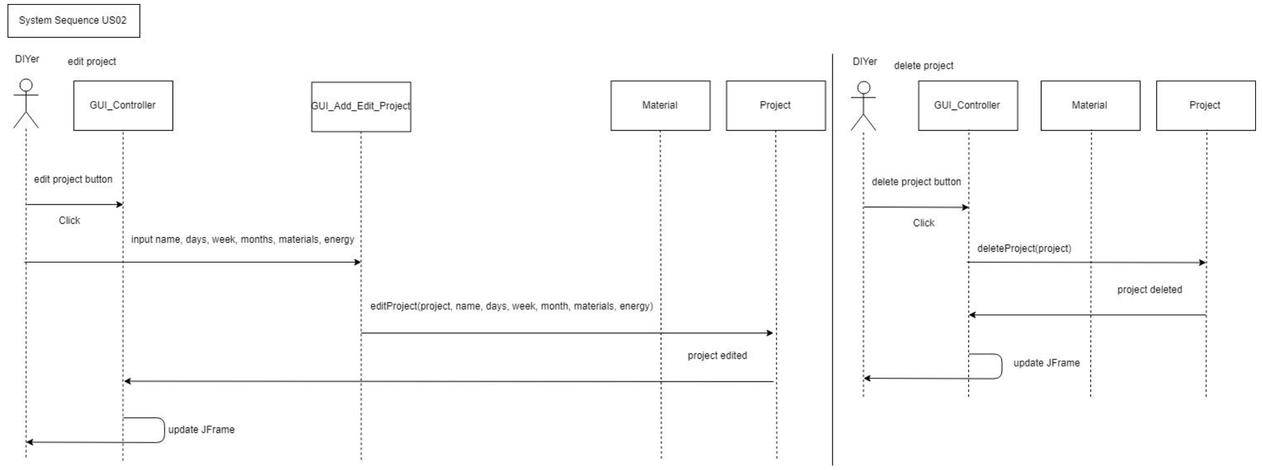
User Story Sequence Diagrams:

US01: As a DIYer, I want an app that collects project costs, duration, and other data that I want to put in.


System Sequence US01
DIYer
add project
GUI_Controller
GUI_Add_Edit_Project
Material
Project
add project (+) button or File -> Add Project tab
Click
input name, days, week, months, materials, energy
add(name, days, week, months, materials, energy)
new project added
update JFrame

The DIYer will initiate the create new project process by either selecting the add project button on the home page or through the File menu and selecting new project on the GUI_Controller. We chose to have this option in two places because we want the user to have their own style of using our application. The GUI_Add_Edit_Project will show up where the user will input a project name, the day, week, and months of the project's duration, the materials and their cost needed. The user will also choose an energy efficiency rating. The GUI_Add_Edit_Project will then send the information given to the GUI_Controller. It will then call the addProject method and pass all the project information through it. The project will be created and added to the list of Projects. After that it will notify the GUI_Controller to update and display the updated list of Projects. The diagram shows us going through the Material class because we will be storing the materials into the Material class, and then storing an instance of Material into Project.

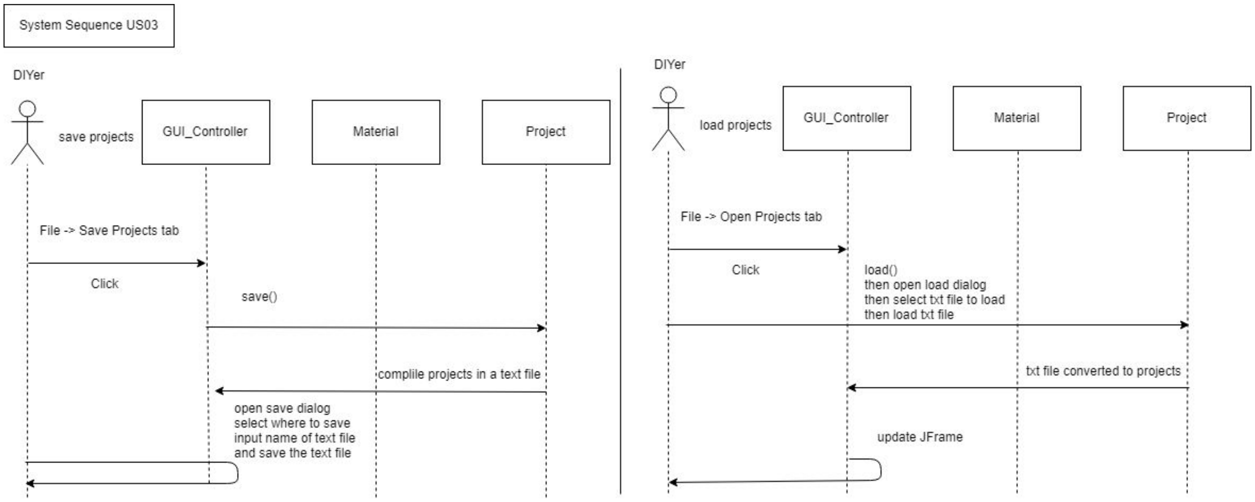
US02: As a DIYer, I want an app that organize my projects to compare many projects at once so I can choose one.



For editing a project, the DIYer will be clicking a project's edit button in the GUI_Controller. The GUI_Add_Edit_Project will show up where the user will change an input if he or she wants to change the project name, the day, week, and months of the project's duration, the materials or the cost needed. The user can also choose to change the energy efficiency rating. The GUI_Add_Edit_Project will then send the information given to the GUI_Controller. It will then call the editProject method and pass a project and all the edited project information through it. A project will be edited and updates the list of Projects. After that it will notify the GUI_Controller to update and display the updated list of Projects. The diagram shows us going through the Material class because we will be storing the materials into the Material class, and then storing an instance of Material into Project.

For deleting a project, the DIYer will be clicking a project's delete button in the GUI_Controller. It will then call the deleteProject method and pass a project through it. A project will be deleted from the list of Projects. After that it will notify the GUI_Controller to update and display the updated list of Projects.

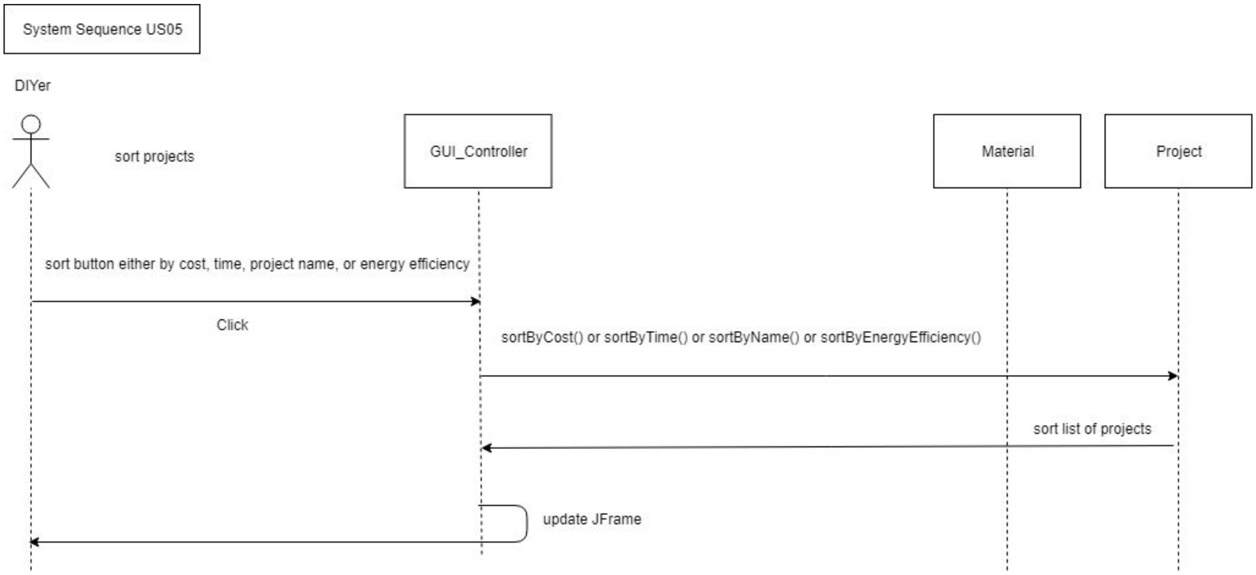
US03: As a DIYer, I want an app that load and saves the projects so I can export and then import to send the information on more than one device.



The DIYer will save the projects by the File menu and selecting save projects on the GUI_Controller. The GUI_Controller will then call the save method. The list of projects will be compiled into a text file. After that the save dialog appears where the DIYer will select where to save the text file. Then, the DIYer will input what the name of the text file will be, and click the save button. The diagram shows us going through the Material class because we will be storing the materials into the Material class, and then storing an instance of Material into Project.

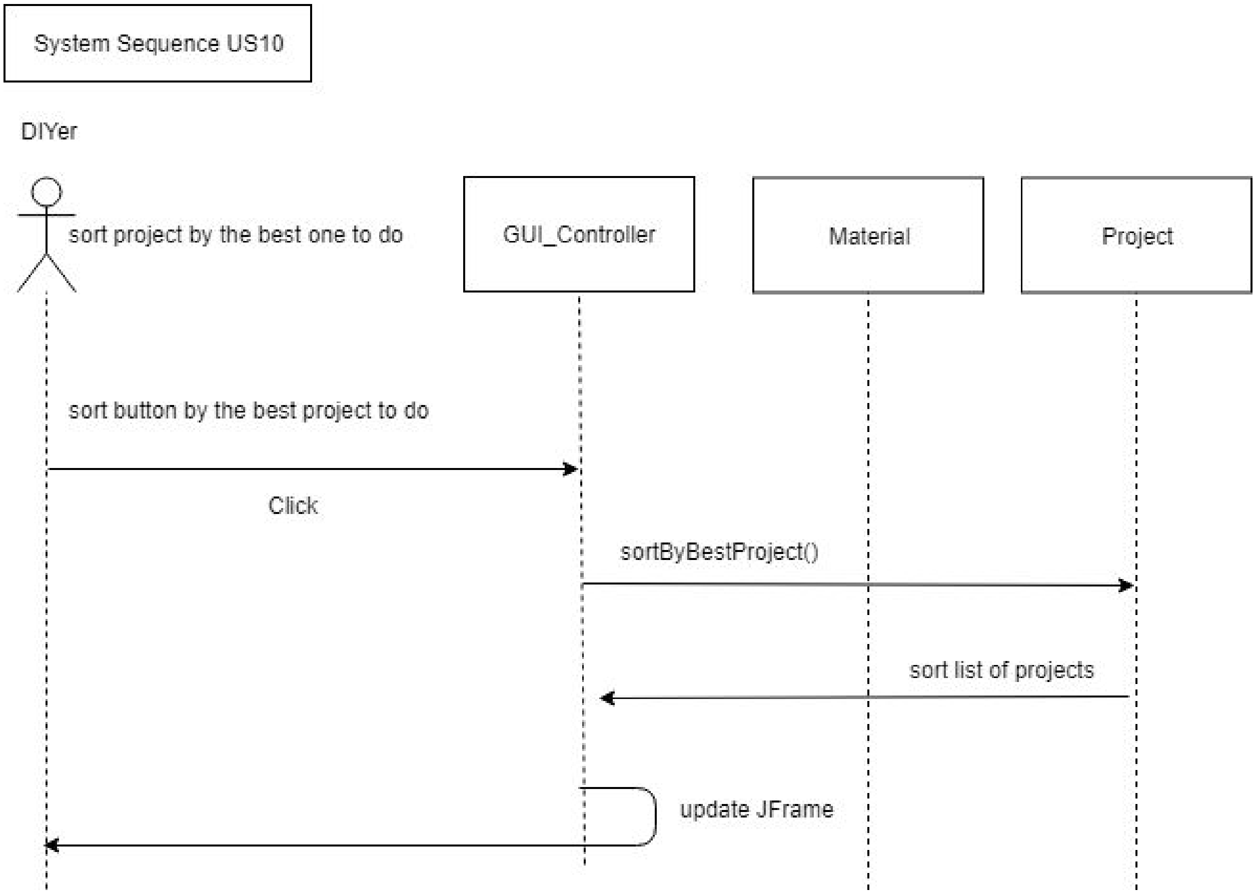
The DIYer will load projects by the File menu and selecting load projects on the GUI_Controller. The GUI_Controller will then call the load method. The load dialog appears where the DIYer can choose a file to load. The file will be converted to a list of projects but could result to an error where none will be loaded. After that the GUI_Controller and the display's list of projects will be updated. The diagram shows us going through the Material class because we will be storing the materials into the Material class, and then storing an instance of Material into Project.

US05: As a DIYer, I want to be able to sort by the project's costs, duration, and other data that have ranking order.


System Sequence US05
DIYer
sort projects
GUI_Controller
Material
Project
sort button either by cost, time, project name, or energy efficiency
Click
sortByCost() or sortByTime() or sortByName() or sortByEnergyEfficiency()
sort list of projects
update JFrame

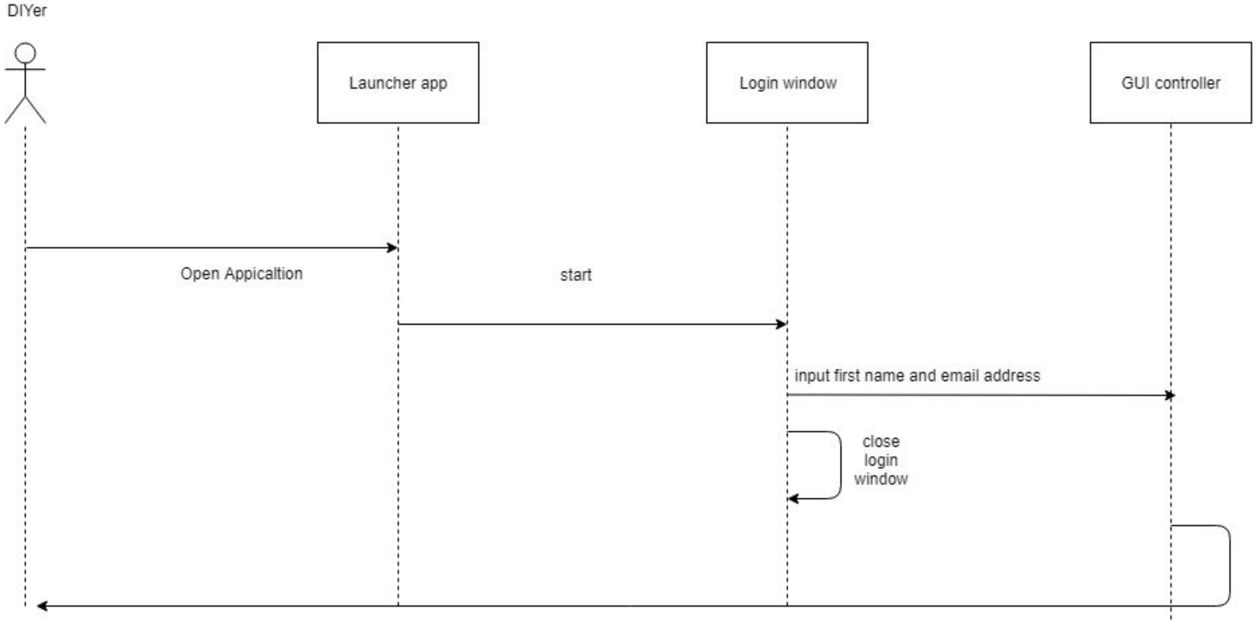
The DIYer will click any of the sort button on the GUI_Controller. The GUI_Controller will then call a sort method. The list of projects will be sorted by type of sorting that the DIYer want. After that it will notify the GUI_Controller to update and display the updated list of Projects. The diagram shows us going through the Material class because we will be storing the materials into the Material class, and then storing an instance of Material into Project.

US10: As a DIYer,, I want an app to make basic calculations and helps weigh costs versus benefits to size up projects.



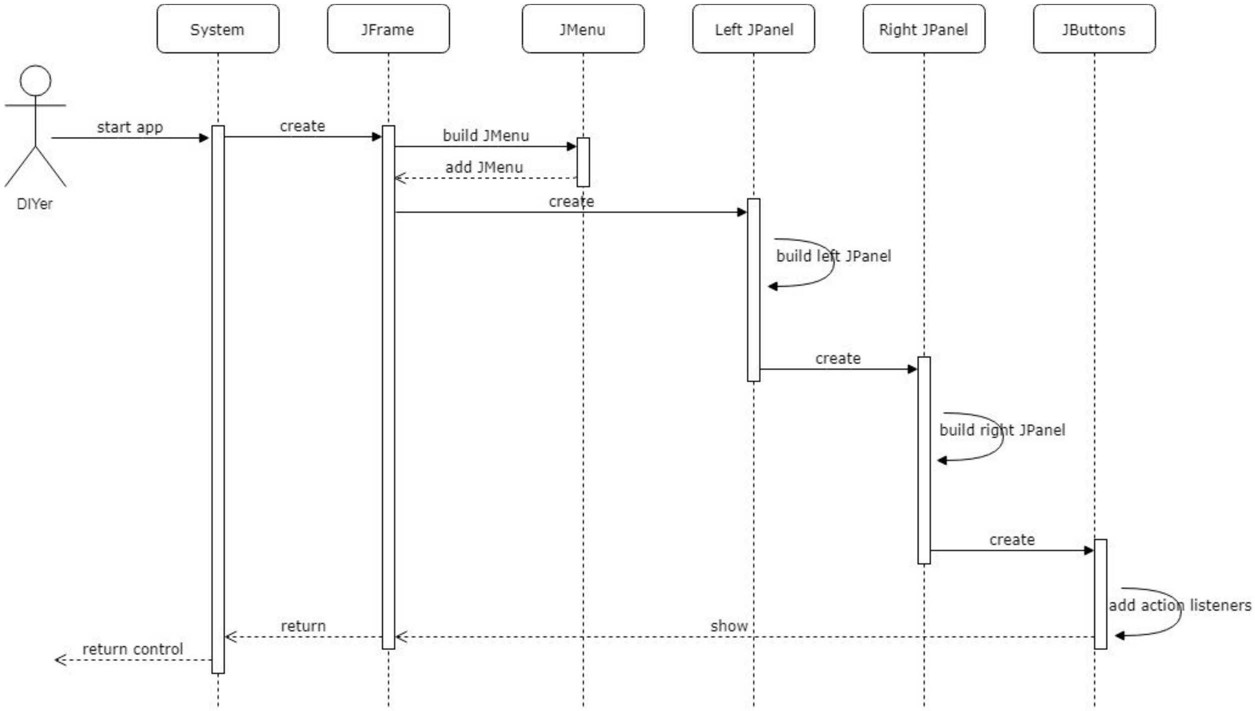
The DIYer will click sort button that sort by the best project on the GUI_Controller. The GUI_Controller will then call the sortByBestProject method. The list of projects will be sorted by the best project. After that it will notify the GUI_Controller to update and display the updated list of Projects. The diagram shows us going through the Material class because we will be storing the materials into the Material class, and then storing an instance of Material into Project.

US15: As a user I want to enter settings such as my first name and email address



The DIYer will open the application. The application will start the login window where the DIyer will input their first name and email address, or they can just close it without typing anything. This information will be sent to the GUI_Controller. The login window will be closed.

System Startup Sequence Diagram:



The DIYer will start the application. The System will create a JFrame for the application. The JFrame will build a JMenu, which will be added to the JFrame. The Left JPanel will be created and built, having all of its components created. Then the Right JPanel will be created and built, having all of its components created. Finally, all of the JButtons will be created and have action listeners added to them. The JFrame will be shown, which will return control to the System and then the DIYer.