**OBJECT ORIENTED PROGRAMMING FINAL PROJECT :**

**HOUND**



**Student Informations:**

Fadhlan Muhammad Razan / 2440047463

**Class Informations:**

Class: L2AC

Lecturer: Jude Joseph Lamug Martinez / D5757

**BINUS UNIVERSITY INTERNATIONAL**

**SCHOOL OF COMPUTER SCIENCE**

**2021**

**Chapter I: Introduction**

For the object oriented programming final project in semester 2, students were tasked to create anything they want as long as it uses object oriented programming in their approach. Months before the due of the project, students were given lectures from their lectures both in the main class sessions and in the lab class sessions. In the main class, we were taught about the concepts of everything that we are doing starting from the basic ideas, logic, and simple exercises that uses the basic implementations of code that we would inevitably use in our final project. We were taught about the basic principles of object oriented programming. It started simple with just classes, inheritance, and class relationships in a program, but then it starts to get complicated when the lecturers taught us about the more complex concepts of object oriented programming. We are introduced to the concept of abstract classes, interfaces, and ways to use static and final keyword in object oriented programming.

For this semester’s final project, I decided to make a GUI application. There are no particular reason on the choosing of the project aside from the fact that I made a game in last semester’s final project. Since I created a game before for my final project, I decided to learn on making a GUI application to further expand my skillset as a programmer of sort. I am kinda familiar with showing graphic on the screen because of my previous final project so I thought that it is not going to be as hard as it actually is. I started to do the project around mid April since I thought it is going to be easier the second time when programming some kind of graphics to be shown on screen. But it turns out to be a hard project since the in the last project, A game menu is just a game menu. The library before already included the necessary functionality in order for an element of a menu screen is functioning. In this case however, I have to expand the basic functionality of the element from the library that I use to better suit my program because a GUI app could be anything while a game is a game if that makes sense.

With the type of project planned I started brainstorming on application ideas. I started with an application that could convert audio file to a midi file. But fiddling with Java libraries is harder than fiddling with python libraries at least for me. So I have to simplify my ideas. I realised that there is no planning for this final project, even though I thought that I planned it for a while. Then it struck to me that I want to create a task manager app or a tracker app. Then some ideas starting to come up and the project started in the making.

**CHAPTER II: PROJECT SPESIFICATION**

**Project purpose:**

To create a task/project tracking application that is created with Java programming language and uses the concepts of object oriented programming and familiarize myself to the workflow of a programmer when tasked on creating an application.

**Project audience:**

People that wanted to be organized by using an application to track their daily activities or any work that they wanted to do.

**Project Aim:**

To create a GUI application that can show tasks and works that the user tracked for them to do. The program should be able to store the information of each task onto each project that the user created. It also should have the basic feature that you expect when using such application such as editing the tasks, editing the projects, removing task, removing project and other things similar to that.

**Project Requirements:**

* A clear user interface so that the user knows what to do in the application
* A feature to add projects and their specific tasks in the program so that the user can separate their tasks based on their projects
* A feature to change the information on the screen seamlessly without the need to change scene in order to make the program look clean
* A feature to edit the information that is already on screen such as cahnging the information or deleting the information (information being the projects/tasks that the user has created)

**CHAPTER III: SOLUTION DESIGN**

**1. OVERVIEW**

****

“Hound” is a GUI application that is used for a user to log a project along with their task so that the user could track their activities in a more organized way.



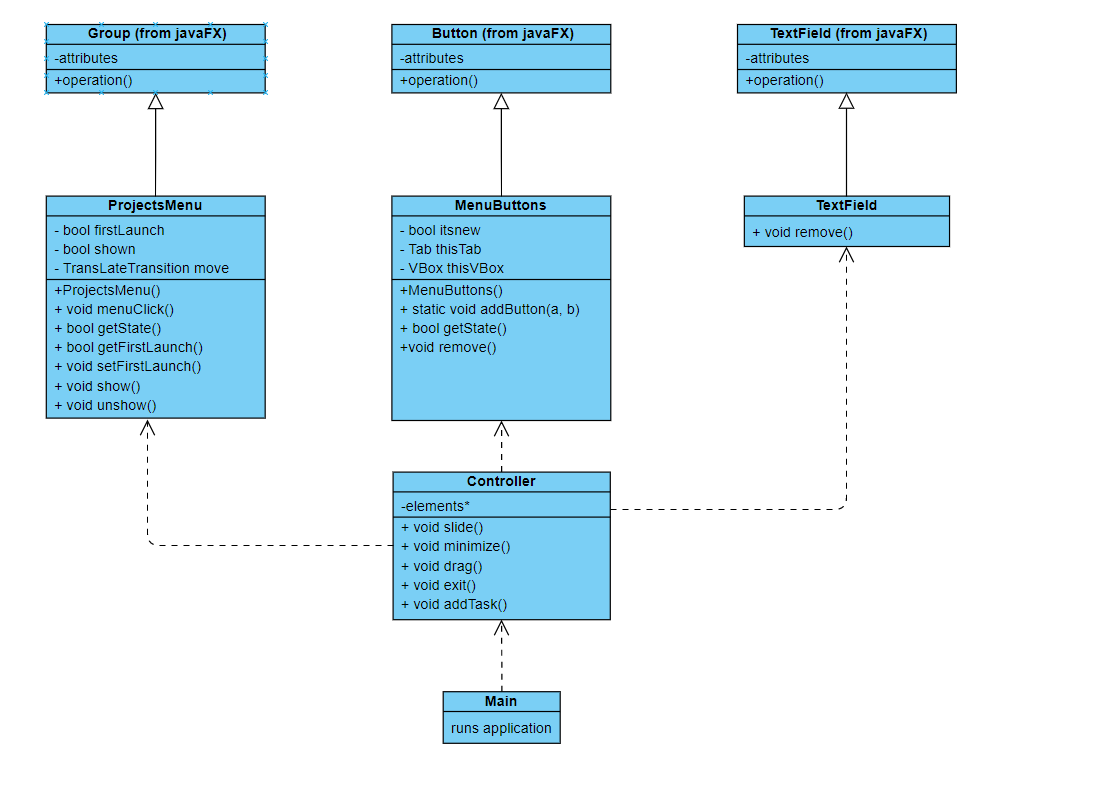
“Hound” is programmed completely in Java and uses the JavaFX library to show the graphics on the screen

The only external library that is used on the project is only the JavaFX library. It is used to display a GUI for the application. All the other functionalities that the program have is coded by myself.

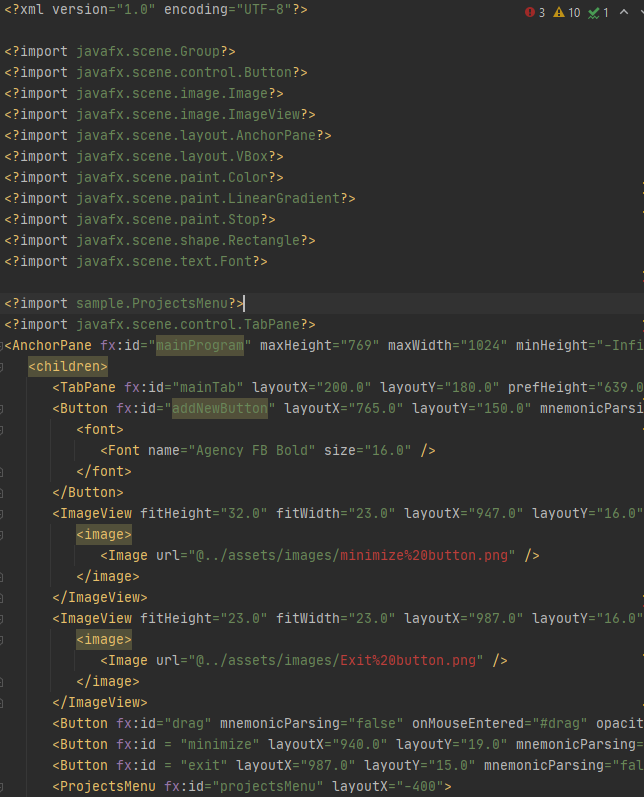
**2. PROGRAM FILE DEPENDENCIES**

The project itself only requires the javaFX library as its external file dependancy. If we are talking about the structure of the project, the code for the program uses a custom version of the already available classes of the javaFX library. For creating this app, I use a class that inherits the property of some class from the library and added extra attributes and methods that would suit the development of the program

Basic representation of the code (the code in a nutshell):



**3. Code Implementations**

To further elaborate from the class diagram, the program is going to be called to run from the Main.java. As for the creation of the UI, javaFX provides a way to design the UI using fxml. The fxml file itself is similar to a html file in the sense that it is used to arrange each of the elements that we wanna show in the final application.

If we told the Main.java to run the file with the configuration of the fxml file, it will launch a window of the application with the UI set up like this:



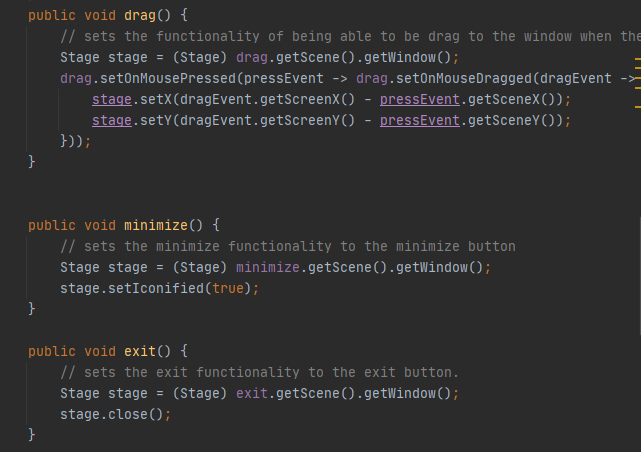
There would be nothing on the program as of yet (excluding the controls) since the user havent done anything with the application. Before explaining how the program is going to be used by the user, I will have to explain each of the elements inside the program and each code related to it first.



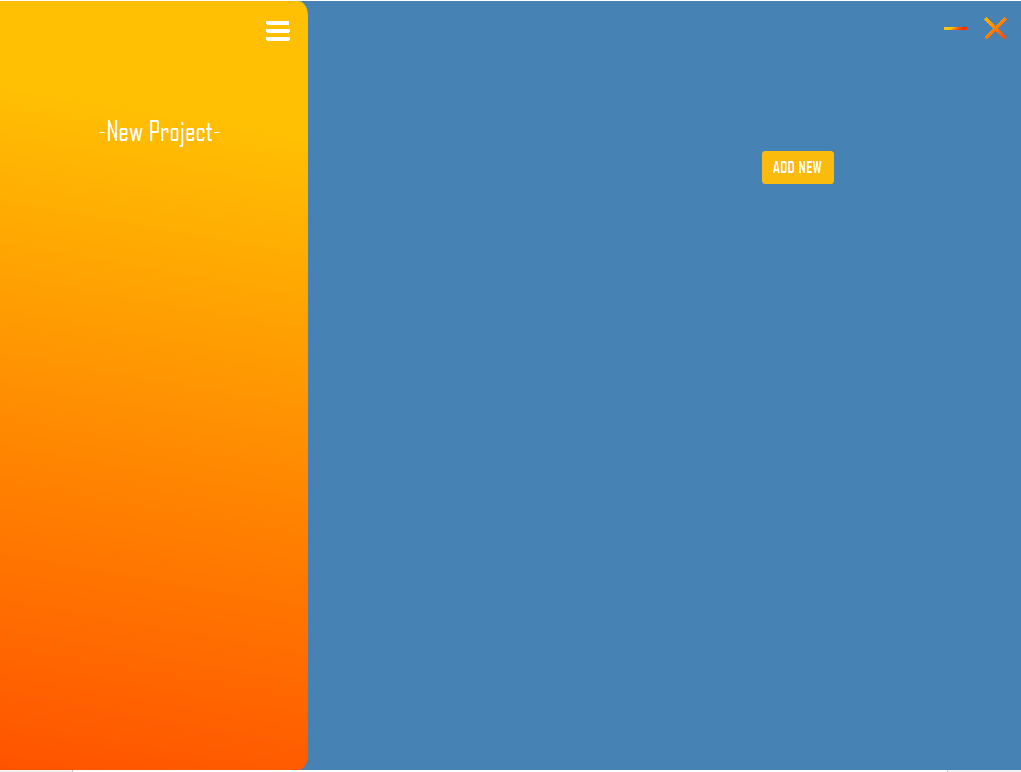


In the picture, the circeled elements are button that uses the Button class straight from javaFX. In the code, I just overrided each of their function using the Controller class by calling their instances and then use the already available method of assigning the onclick() method of the Button class. The 3 white bars button is used to show all of the projects that the user has inputted, the 2 buttons in the upper right corner is used to close and minimizes the application, and the add new button is used for the user to add a new task to a specific project.

An example of overriding the onclick function with our own function is shown in the example picture below:



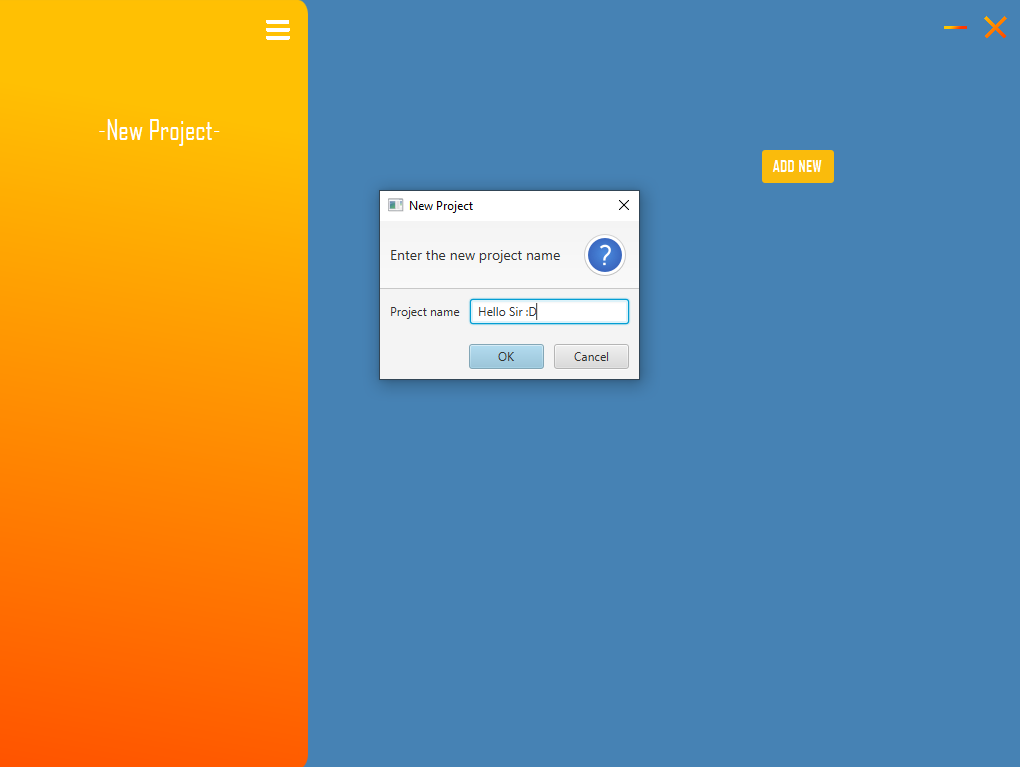
There are also other elements in the program that are only visible after we click the 3 white bar button as shown in this picture:



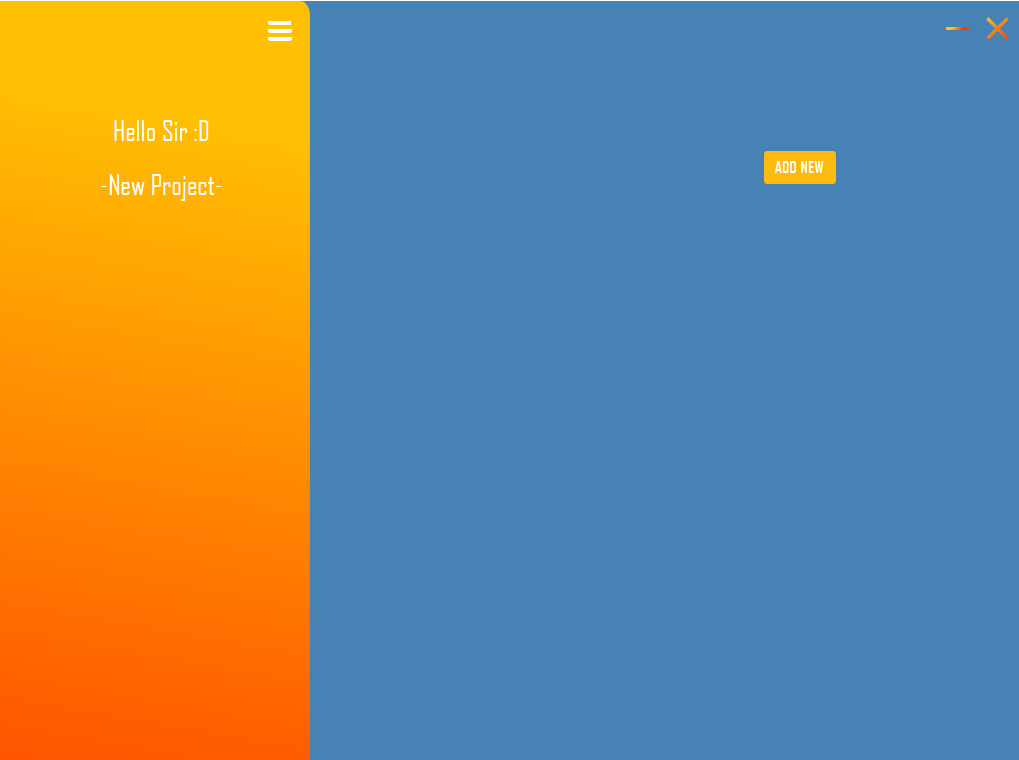
In the image above, there are few elements. Some are visible to the user, while some are not.

For example there is a new Project button that appears but there is also a vBox that contains the new Project button and there is also a tabPane that has tabs that are associated to the new Project Button.

As you see in the UML diagram the MenuButton class has the attribute thisTab and thisVBox. It is used to assign a new tab and a new vBox that is associated to the menuButton. This is used so whenever the user pressed the button, it will then switch the currently active tab to the tab associated with the menuButton. And it will also tell the Controller class that the currently active vBox that will be used in the process of adding new tasks.

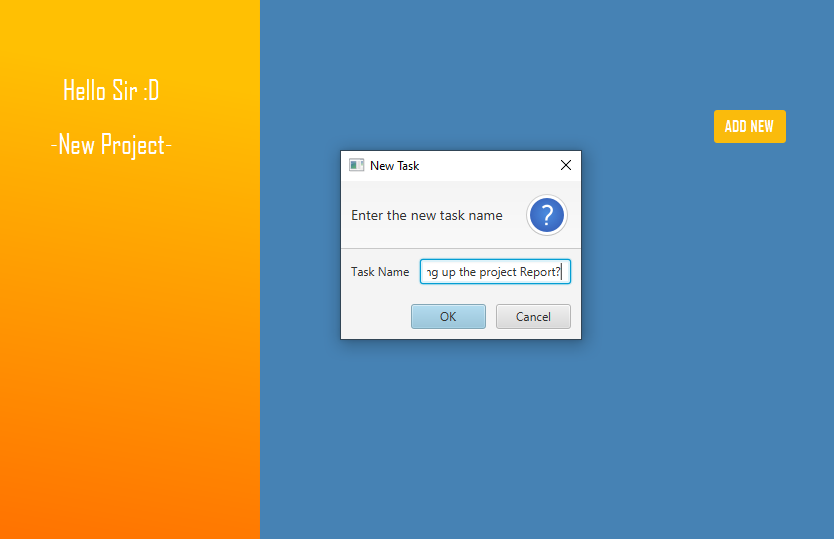
How it works now is that when the user click the new project button, it is going to ask an input to the user for the name of their new project.

If it is a valid text input and that the user doesn’t outright close the new project window or press cancel, then the new project text in the button is going to be changed to the user inputted text. Then the addbutton static method of the menuButton class is going to be called and it will add the button to the projectMenuButton vbox.



Everytime a new button is added it will make a new tab in the tabPane and then associating it to the new button.

Right now the tabPane is going to show the tab that is associated with the newly named project. So now, everytime you click the add new button, it will ask the user for another input of tasks and then inserting it if it is a valid input



In a program like this a user alse expect for the program to be able to correct things by deleting or editing certain project or task. If a user want to delete a task or a project, they only need to click it and then pressing alt on the keyboard. But if the user want to edit it, they only need to click the task or right click the project.

**CHAPTER IV: PROJECT DEMO**

<https://youtu.be/2FxFAfzhgoo>