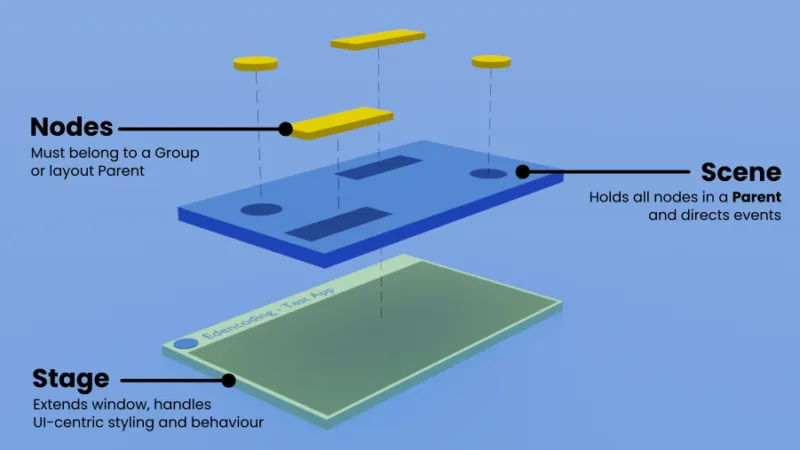
**ABOUT JAVAFX**

JavaFX is a library or a framework, however you like to call it, that developers can use to create desktop applications. It was also developed to replace *Swing* and *AWT*, two toolkits that were also used to create applications.

A **library**is a set or collection of prewritten code that other developers can use to help them with their tasks. In the case of JavaFX, the library had already dealt with the horrors of creating the code for interacting with the OS itself to give us the Graphical User Interface that we can manipulate however we like, and we do not even need to know it (Abstraction).

[*https://edencoding.com/javafx-scene/*](https://edencoding.com/javafx-scene/)

**How do JavaFX Applications work?**

The application is made up of four major parts: **Stage, Scene,** and the **Parent** and **Children Nodes.**

The stage is the window itself, where we can set scenes into.

Stage:



Scenes:

A picture containing logo

Description automatically generatedGraphical user interface

Description automatically generatedNodes:

**Source Code:**

Package:

*package* com.example.lms;

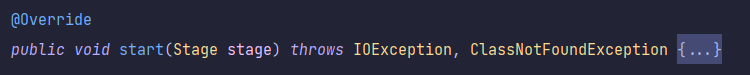
This is the first line of code for every class in the project. This package includes everything inside this folder.

*public static void* main(String[] args) {  
 *launch*();  
}

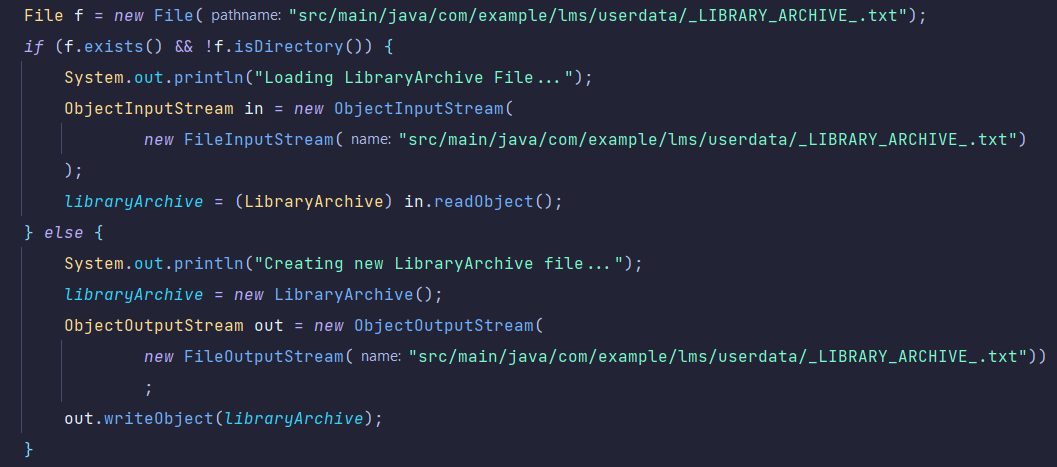
Inside the Main class, a generic psvm (public static void main) method is created to call the *launch()* method which will start the application. This method will call the start() method inside the abstract **Application** class included in the JavaFX library, which will be overridden in this project.

*public class* Main *extends* Application

The Main class extends to the Application class (an Abstract class), which means we have to override the start() method before continuing.



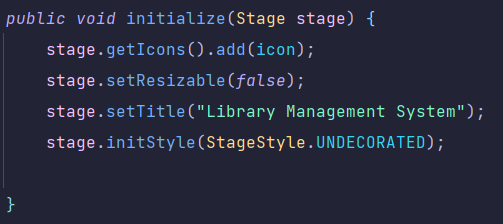
The first blocks of code inside the overridden start() method is handling the instantiation of the LibraryArchive object which will be explained later.



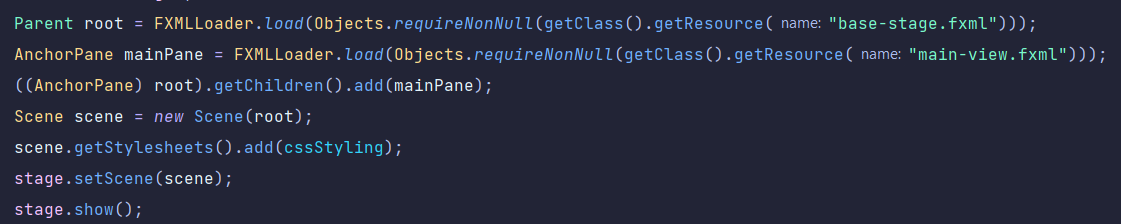
It’s basically only saying that if the **\_LIBRARY\_ARCHIVE\_.txt** exists in the said directory, then load it, if not, then create one.



And then the program will call the initialize() method that I only created inside the class, with the stage as its argument.



This method sets the icon for the program, makes the window not resizable, sets the title, and makes it undecorated. Making the stage **undecorated** means removing the upper bar in the application which has the maximize, minimize, and exit control buttons.



Here is where we will set the stage for our application. Remember that nodes need a **Group** or **Parent** to be passed to a **Scene**, which will also be passed to the **Stage** before it can be shown.

Children Nodes -> Parent Nodes -> Scene -> Stage

mainPane (child Node) -> root (parent Node) -> scene (Scene) -> stage (Stage)