Contents

❖ Abstract :

The provided Java code presents a menu-driven program designed to manage users, software applications, and licenses. The program utilizes object-oriented principles to create and manipulate instances of users, applications, and licenses. It employs JSON file handling to persist data, enabling the storage and retrieval of user, application, and license information across sessions.

The program starts by initializing necessary components such as views, models, and controllers. It then enters a loop where users are prompted with a menu of options to perform various operations like adding, updating, displaying, and deleting users, applications, and licenses. The code demonstrates interactive object creation, allowing users to input data for the creation and modification of objects dynamically. Through switch-case statements, the program routes user choices to the appropriate functionalities, ensuring smooth navigation and management of the system's entities. Overall, the code encapsulates a comprehensive approach to user interaction, data management, and file handling in a Java application.

❖ Acknowledgments :

I would like to express my special thanks to our mentor Mr.Richard Joseph for his time and efforts he provided throughout the semester. Your useful advice and suggestions were really helpful to me during the project's completion. In this aspect, I am eternally grateful to you.

I would like to acknowledge that this project was completed entirely by our group.

Name Signature

❖ List of Abbreviations :

Here's a list of common abbreviations and acronyms related to Java programming:

1. JRE - Java Runtime Environment
2. JDK - Java Development Kit
3. API - Application Programming Interface
4. IDE - Integrated Development Environment
5. GUI - Graphical User Interface
6. JVM - Java Virtual Machine
7. OOP - Object-Oriented Programming
8. SQL - Structured Query Language
9. XML - Extensible Markup Language
10. JSON - JavaScript Object Notation
11. HTML - Hypertext Markup Language
12. CSS - Cascading Style Sheets
13. URL - Uniform Resource Locator
14. IO - Input/Output
15. SSL - Secure Sockets Layer
16. TLS - Transport Layer Security
17. HTTP - Hypertext Transfer Protocol

18. HTTPS - Hypertext Transfer Protocol Secure
19. POJO - Plain Old Java Object
20. DAO - Data Access Object
21. DTO - Data Transfer Object
22. MVC - Model-View-Controller
23. ORM - Object-Relational Mapping
24. REST - Representational State Transfer
25. SOAP - Simple Object Access Protocol

These are just a few examples, and there are many more abbreviations and acronyms used in the Java programming ecosystem. If you encounter any specific abbreviation that you're unsure about, feel free to ask!

❖ List of Figures :

Figure 1: GUI Mockup for the Main Screen
Description: Displays a mockup of the graphical user interface for the main screen of the application.

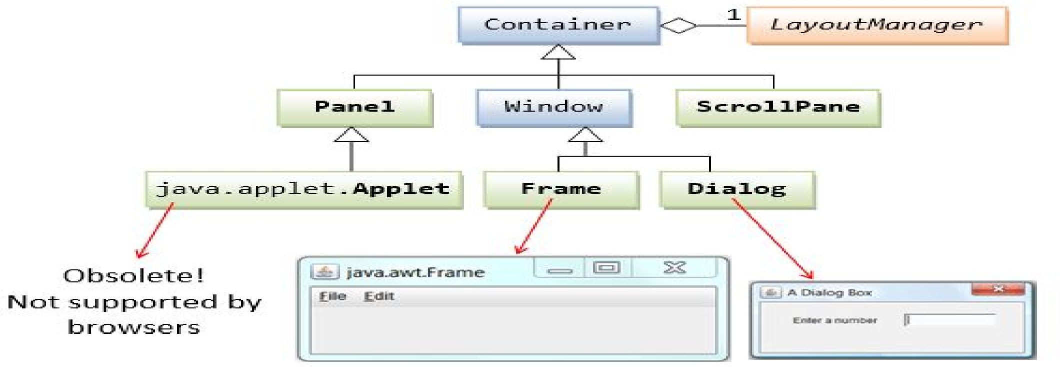
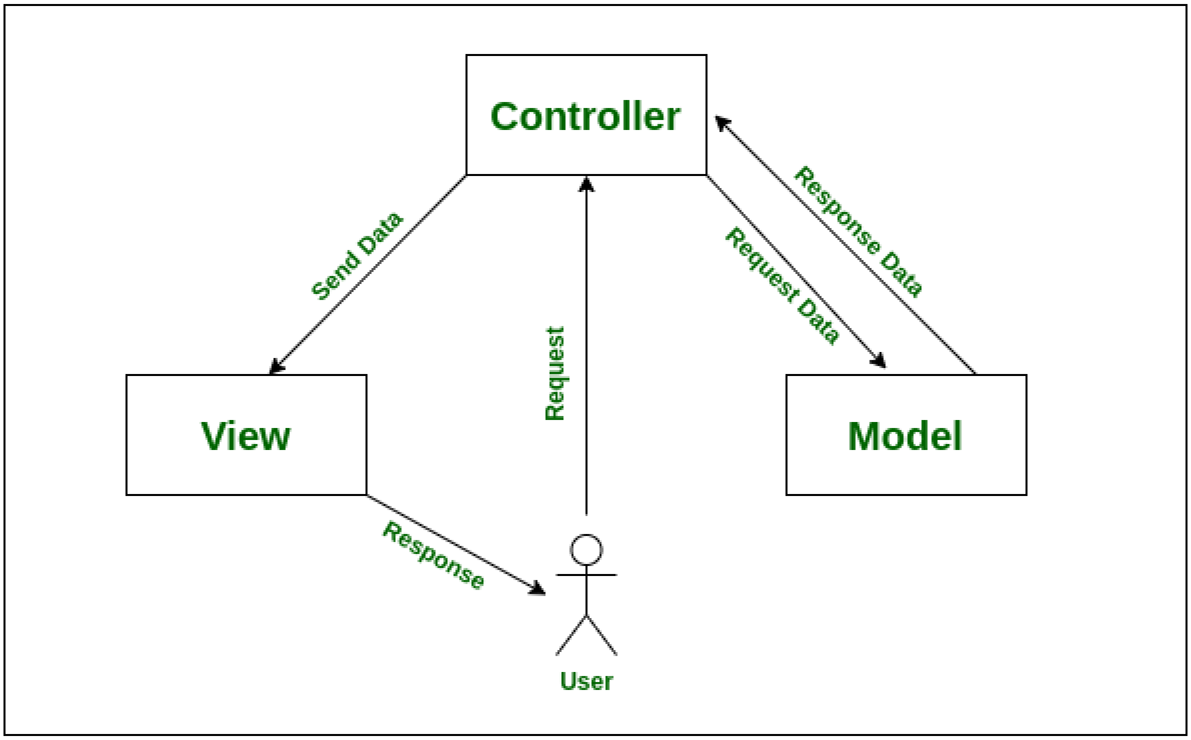


Figure 2: UML Diagram for Package Structure
Description: Presents a UML diagram illustrating the package structure of the Java project.



❖List of Symbols :

Here's a list of common symbols in Java programming along with examples:

Arithmetic Operators :

- ➔ + Addition: int sum = 5 + 3;
- ➔ - Subtraction: int difference = 5 - 3;
- ➔ * Multiplication: int product = 5 * 3;
- ➔ / Division: int quotient = 5 / 3;

- % Modulus: int remainder = 5 % 3;

Assignment Operator :

- = Assignment: int x = 10;
- Comparison Operators:
-
- == Equal to: if (x == 10) { ... }
- != Not equal to: if (x != 10) { ... }
- > Greater than: if (x > 10) { ... }
- < Less than: if (x < 10) { ... }
- >= Greater than or equal to: if (x >= 10) { ... }
- <= Less than or equal to: if (x <= 10) { ... }

Logical Operators:

- && Logical AND: if (x > 0 && x < 10) { ... }
- || Logical OR: if (x == 0 || y == 0) { ... }
- ! Logical NOT: if (!(x > 0)) { ... }

Increment/Decrement Operators:

- ++ Increment: x++;
- -- Decrement: x--;

Concatenation Operator:

- + Concatenation: String fullName = firstName + " " + lastName;

Bitwise Operators:

- & Bitwise AND: int result = 5 & 3;
- | Bitwise OR: int result = 5 | 3;
- ^ Bitwise XOR: int result = 5 ^ 3;
- ~ Bitwise NOT: int result = ~5;

Ternary Operator:

-

- ? : Ternary operator: String result = (x > 0) ? "Positive" : "Negative";
- Conditional Operator:
-
- : Conditional operator: String result = (x > 0) ? "Positive" : "Negative";

Miscellaneous Symbols:

- ; Semicolon: End of statement.
- . Dot: Access member of a class or object.
- [] Square Brackets: Used for arrays.
- () Parentheses: Used in method calls, expressions, and control flow statements.

These symbols are fundamental to Java programming and are used extensively in writing Java code.

Introduction

1.1 Introduction

The provided Java code exemplifies a modular approach to building a comprehensive management system for users, software applications, and licenses. With an interactive menu-driven interface, users can seamlessly navigate through a range of operations including adding, updating, displaying, and deleting entities within each category. The program leverages object-oriented programming principles to encapsulate the behavior and attributes of users, applications, and licenses, ensuring a flexible and scalable design. Utilizing JSON file handling, the system achieves persistence, allowing data to be stored and retrieved across sessions, thereby enhancing usability and continuity.

Structured around the Model-View-Controller (MVC) architectural pattern, the code embodies a clear separation of concerns, with distinct modules responsible for handling data (Model), user interface (View), and application logic (Controller). This architectural choice promotes code maintainability and extensibility, facilitating future enhancements or modifications to accommodate evolving requirements. Overall, the code showcases a robust and efficient solution for managing users, applications, and licenses, offering a versatile platform for various administrative tasks in a Java environment.

1.2 Motivation

The motivation behind the development of this code lies in the need for an efficient and user-friendly system to manage users, software applications, and licenses. Such a system can streamline administrative tasks, improve organization, and enhance productivity in various contexts, including businesses, educational institutions, or software development teams.

By providing a menu-driven interface, the code empowers users to interact with the system intuitively, enabling them to perform tasks such as adding new users, updating application information, or assigning licenses seamlessly. This approach not only simplifies the management process but also reduces the likelihood of errors by guiding users through predefined workflows.

Moreover, the utilization of JSON file handling ensures data persistence, enabling the system to store information across sessions. This feature enhances usability and reliability, as users can access previously entered data and continue their work uninterrupted.

Overall, the motivation behind this code is to create a robust, adaptable, and user-centric solution that addresses the complexities of managing users, applications, and licenses while promoting efficiency, organization, and ease of use.

1.3 Problem Statement & Objectives

Problem Statement:

In various organizational or software development contexts, there is often a need for a streamlined system to manage users, software applications, and licenses efficiently. Traditional manual methods of tracking and updating user information, application details, and license assignments can be time-consuming, error-prone, and lack scalability. Therefore, there is a demand for a software solution that automates these administrative tasks, provides a user-friendly interface for interaction, and ensures data integrity and persistence.

Objectives:

1. Develop a Java-based system that facilitates the management of users, software applications, and licenses through an interactive menu-driven interface.
2. Implement functionalities to allow users to perform operations such as adding, updating, displaying, and deleting users, applications, and licenses.
3. Utilize object-oriented programming principles to encapsulate the behavior and attributes of users, applications, and licenses, ensuring modularity, flexibility, and code reusability.
4. Incorporate JSON file handling to enable data persistence, allowing user-entered information to be stored and retrieved across sessions.
5. Ensure robust error handling and exception management to enhance the reliability and stability of the system.
6. Design the system following the Model-View-Controller (MVC) architectural pattern to ensure a clear separation of concerns and promote code maintainability and extensibility.
7. Provide a comprehensive solution that addresses the complexities of managing users, software applications, and licenses, catering to the needs of various organizational or development environments.

Proposed System (eg New Approach of Data Summarization)

2.1 Introduction

The Java menu-driven program presented herein is designed to manage users, software applications, and licenses. This project aims to provide a practical demonstration of object-oriented principles in Java programming, with a specific focus on creating and manipulating instances of users, applications, and licenses. By leveraging JSON file handling techniques, the program enables the persistent storage and retrieval of user, application, and license information, ensuring data integrity across sessions.

The primary purpose of this project is to showcase the practical application of object-oriented programming concepts in Java, particularly in the context of building interactive and data-driven applications. Through interactive object creation and manipulation, users are empowered to perform various operations such as adding, updating, displaying, and deleting users, applications, and licenses.

By employing switch-case statements, the program efficiently routes user choices to the appropriate functionalities, ensuring smooth navigation and management of the system's entities. This project serves as a valuable learning resource for both beginners seeking to grasp the fundamentals of Java programming and experienced developers looking to enhance their skills in object-oriented design and implementation.

In conclusion, this Java program encapsulates a comprehensive approach to user interaction, data management, and file handling. Whether you're embarking on your journey as a Java developer or seeking to deepen your understanding of object-oriented principles, this project offers valuable insights into building robust and maintainable applications in Java.

2.2 Architecture

MVC stands for Model-View-Controller, and it's a design pattern commonly used in software development, especially for building user interfaces. It aims to separate the concerns of an application into three interconnected components:

1. Model: The Model represents the data and the business logic of the application. It encapsulates the data, and it's responsible for managing access to that data. In simpler terms, the Model represents the 'what' of an application - what data it operates on and how it manipulates that data.

2. View: The View is responsible for rendering the user interface and displaying the data from the Model to the user. It represents the presentation layer of the application. Views are typically passive; they observe changes in the Model and update themselves accordingly. In other words, the View represents the 'how' of an application - how the data is presented to the user.

3. Controller: The Controller acts as an intermediary between the Model and the View. It receives input from the user via the View, processes that input (often by interacting with the Model), and updates the View accordingly. Controllers interpret user actions and decide how to respond to them. They bridge the communication between the Model and the View. Controllers represent the 'how the user interacts' with the application.

Here's how the components interact in a typical MVC flow:

- The user interacts with the View, such as clicking a button or entering data into a form.

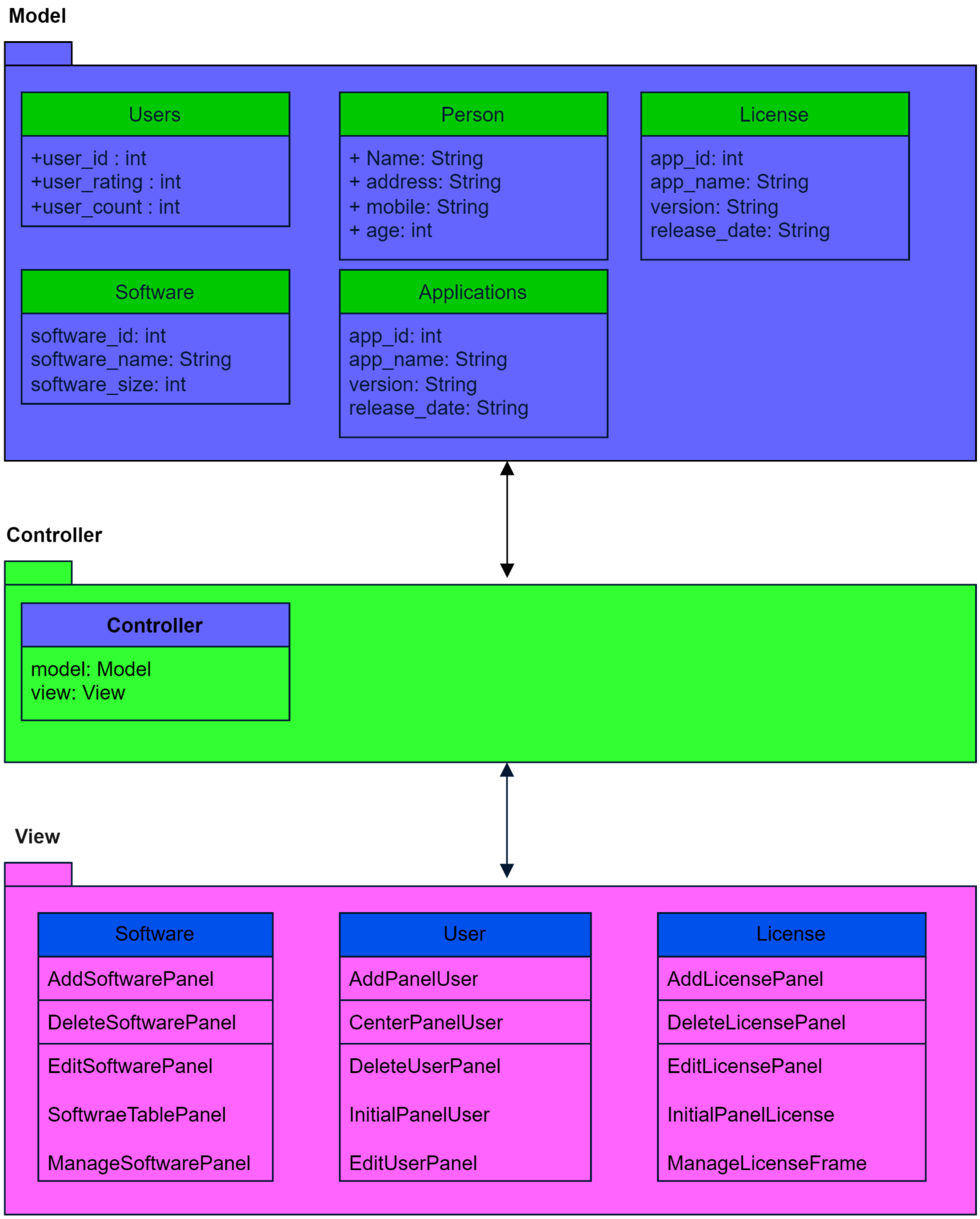
- The View sends the user's actions to the Controller.

- The Controller processes the user's actions, interacts with the Model to perform any necessary operations or retrieve data.

- The Model updates its state or performs operations based on the Controller's instructions.

- The Controller then updates the View with any changes in the Model's data.
- The updated View reflects these changes to the user.

One of the key benefits of using the MVC architecture is that it promotes separation of concerns, making the codebase easier to maintain and extend. Each component has a specific responsibility, which makes it easier to understand and modify the code without affecting other parts of the application. Additionally, it allows for better reusability of components and easier testing, as each component can be tested independently. Overall, MVC is a widely used architectural pattern that provides a structured and organized way to develop applications, particularly those with user interfaces.


Model
Users
+user_id : int
+user_rating : int
+user_count : int
Person
+ Name: String
+ address: String
+ mobile: String
+ age: int
License
app_id: int
app_name: String
version: String
release_date: String
Software
software_id: int
software_name: String
software_size: int
Applications
app_id: int
app_name: String
version: String
release_date: String
Controller
Controller
model: Model
view: View
View
Software
AddSoftwarePanel
DeleteSoftwarePanel
EditSoftwarePanel
SoftwraeTablePanel
ManageSoftwarePanel
User
AddPanelUser
CenterPanelUser
DeleteUserPanel
InitialPanelUser
EditUserPanel
License
AddLicensePanel
DeleteLicensePanel
EditLicensePanel
InitialPanelLicense
ManageLicenseFrame

2.2 Algorithm and process design

Certainly! Here's the course flow without any bold or highlighted text:

1. Initialization:

- The program initializes necessary components including View, Model, and Controller.
- It creates a Scanner object to take input from the user.
- It initializes arrays/lists to store users, applications, and licenses.

2. Menu Driven Interface:

- The program enters a loop to display a menu of options to the user.
- Options include operations for users, applications, and licenses such as adding, updating, displaying, and deleting entities.

3. User Operations:

- Users can choose operations related to managing users.
- Options include adding a user, updating user details, displaying a user, displaying all users, and deleting a user.

4. Application Operations:

- Users can choose operations related to managing software applications.
- Options include adding an application, updating application details, displaying an application, displaying all applications, and deleting an application.

5. License Operations:

- Users can choose operations related to managing licenses.
- Options include adding a license, deleting a license, and displaying all licenses.

6. Data Input and Validation:

- For operations that require user input (e.g., adding a user or application), the program prompts the user for relevant details.
- Input validation is performed to ensure the correctness of user-provided data (e.g.,

validating mobile numbers, application versions).

7. Exception Handling:

- Custom exceptions (e.g., mobileNumberException, appversionException) are thrown and caught to handle exceptional scenarios such as invalid input or incorrect data format.

8. Data Persistence:

- Data is read from and written to JSON files to achieve persistence.

- Methods like readUserJsonFile, writeUserJsonFile, readAppJsonFile, writeAppJsonFile, etc., handle file I/O operations for users, applications, and licenses.

9. Model-View-Controller (MVC) Architecture:

- The program follows the MVC architectural pattern, where Model represents the data, View represents the user interface, and Controller acts as an intermediary to handle user input and update the model accordingly.

10. Loop Continuation and Exit

The program continues to display the menu and prompt the user for options until the user chooses to exit.

Upon selecting the exit option, the program terminates.

Main.java

```
import Controller.Controller;

import Model.Licenses.License;

import Model.Licenses.manageLicense;

import Model.Model;

import Model.Softwares.Applications;

import Model.Softwares.appversionException;

//import Model.Users.Users;

import Model.Softwares.manageApplication;

import Model.Users.Users;

import Model.Users.manageUsers;

import Model.Users.mobileNumberException;

import View.View;

import
com.fasterxml.jackson.databind.deser.std.CollectionDeserializer;


import javax.swing.text.html.CSS;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Scanner;



/**

 * Identification comments:

 * Name: Harsh Tanwani

 * Experiment No: 05

 * Experiment Title:Interactive Object Creation: Empowering
Users to Generate Objects

 through Java Input

 * Experiment Date:23/01/2024

 * @version 1.0

 *

 *

 * Beginning comments:

 * Filename: Main.java

 * @author: Harsh Tanwani

 * Overview: This is the main class used to create objects for
Airports Class and Softwares

Class. In this file we have achieved the following

 * - Menu Driven Program to create Airports and Softwares

 *

 */

public class Main {

    private static
CollectionDeserializer.CollectionReferringAccumulator users;


    public static void main(String[] args) throws IOException {



        View view = new View();

        Model model = new Model();

        Controller controller = new Controller(model,view);


//Creating a Scanner Object to take input

        Scanner sc = new Scanner(System.in);

        ArrayList<Users> users =new ArrayList<>();

        manageUsers mu1=new manageUsers();

        manageApplication ma1=new manageApplication();


users=mu1.readUserJsonFile("src/Model/Users/User_Data.json");

        users = mu1.getTable();


        ArrayList<Applications> application =new ArrayList<>();


application=ma1.readAppJsonFile("src/Model/Softwares/Application_Data.json");

        application = ma1.getTable();


        ArrayList<License> licenses = new
ArrayList<License>();

        manageLicense ml1 = new manageLicense();

        licenses = ml1.getTable();

        int choice;
```

```
// display menu and get user's choice
    do {
      System.out.println("Menu");
      System.out.println("1.Add a User");
      System.out.println("2.Update a User");
      System.out.println("3.Display a User");
      System.out.println("4.Display all Users");
      System.out.println("5.Delete a User");
      System.out.println("6.Add a Software");
      System.out.println("7.Update a Software");
      System.out.println("8.Display a Software");
      System.out.println("9.Display all Software");
      System.out.println("10.Delete a Software");
      System.out.println("11.Add a License");
      System.out.println("12.Delete a License");
      System.out.println("13.Display all License");
      System.out.println("14.Exit");
//Taking input from user
      System.out.println("Enter a Choice");
      choice = sc.nextInt();
      sc.nextLine();
//Execute the selected operation
      switch (choice) {
//Taking user input for each Airports object in the array
        case 1:
          System.out.println("How do you wish to create User Id");
          System.out.println("1. Auto Generate Id and Add no other Details");
          System.out.println("2. Auto Generate Id and Add other Details");
          System.out.println("3. Provide User Id and Add other Details");
          System.out.print("Enter your choice: ");
          int pass_id_choice = sc.nextInt();
          sc.nextLine();

          if (pass_id_choice == 1) {
//            System.out.println("Hello");
            users.add(new Users());
mu1.writeUserJsonFile("src/Model/Users/User_Data.json", users);
          } else if (pass_id_choice == 2) {
//             Auto generating id of Users
            System.out.println("Enter First Name and Last Name");
            String name = sc.nextLine();
            String[] name_split = name.split(" ");

            System.out.println("Enter Address:");
            String add = sc.nextLine();

            System.out.println("Enter Id");
            int id = sc.nextInt();
            //Displaying Input in uppercase letters
            String s1 = name.toUpperCase();
            //Removing the spaces before and after the string using trim function

            System.out.println("Enter User rating:");
            int rating = sc.nextInt();

            System.out.println("Enter Mobile No. :");
            String mobile = sc.nextLine();
            try {
              if (mobile.length() == 10) {
                users.add(new Users(name_split[0],name_split[1], add, id, rating));
mu1.writeUserJsonFile("src/Model/Users/User_Data.json", users);
              } else {
                throw new mobileNumberException("Phone number should be of 10 numbers");
              }
            } catch (mobileNumberException e) {
              System.out.println("Custom Exception: " + e.getMessage());
            }
          } else if (pass_id_choice == 3) {
            System.out.println("Enter First Name and Last Name");
            String name = sc.nextLine();
            String[] name_split = name.split(" ");

            System.out.println("Enter Address:");
            String add = sc.nextLine();



            System.out.println("Enter Id:");
            int id = sc.nextInt();

            System.out.println("Enter User rating:");
            int rating = sc.nextInt();

            System.out.println("Enter Mobile No. :");
            String mobile = sc.nextLine();
            try {
                if (mobile.length() == 10) {
                    users.add(new Users(name_split[0],name_split[1], add, id, rating));
                    mu1.writeUserJsonFile("src/Model/Users/User_Data.json", users);
                } else {
                    throw new mobileNumberException("Phone number should be of 10 numbers");
                }
            } catch (mobileNumberException e) {
                System.out.println("Custom Exception: " + e.getMessage());
            }
          }



        case 2:
          //Taking user input to display specific Airport data
          System.out.print("Enter Users Index to Update: ");
          int user_idx = sc.nextInt();
          sc.nextLine();
          int choice_to_change;
          do {
```

```
        System.out.println("1. Change Name: ");
        System.out.println("2. Change Address: ");
        System.out.println("3. Change Mobile No. : ");
        System.out.println("4. Change Id: ");
        System.out.println("5. Change User Id: ");
        System.out.println("6. Change User rating: ");
        System.out.println("7. Exit Update");

        System.out.print("Enter your choice: ");
        choice_to_change = sc.nextInt();
        sc.nextLine();
        switch (choice_to_change){
            case 1:
                System.out.print("Enter Name: ");
                String name = sc.nextLine();
                users.get(user_idx).setName(name);
                mu1.writeUserJsonFile("src/Model/Users/User_Data.json",users);
                break;
            case 2:
                System.out.print("Enter Address: ");
                String add= sc.nextLine();
                users.get(user_idx).setAddress(add);
                mu1.writeUserJsonFile("src/Model/Users/User_Data.json",users);
                break;
            case 3:
                System.out.print("Enter Mobile No. : ");
                String mobile= sc.nextLine();

                users.get(user_idx).setMobile(mobile);
                mu1.writeUserJsonFile("src/Model/Users/User_Data.json",users);

                break;
            case 4:
                System.out.print("Enter Id: ");
                int id= sc.nextInt();
                users.get(user_idx).setId(id);
```

```
mu1.writeUserJsonFile("src/Model/Users/User_Data.json",us
ers);
                                break;
                            case 5:
                                System.out.print("Enter new User Id: ");
                                int user_id= sc.nextInt();
                                users.get(user_idx).setUser_id(user_id);

mu1.writeUserJsonFile("src/Model/Users/User_Data.json",us
ers);
                                break;
                            case 6:
                                System.out.print("Enter new User rating:
");
                                int user_rating= sc.nextInt();

users.get(user_idx).setUser_rating(user_rating);

mu1.writeUserJsonFile("src/Model/Users/User_Data.json",us
ers);
                                break;
                            case 7:
                                System.out.println("Thank You!");
                            default:
                                System.out.println("Invalid choice. Try
again.");
                                break;
                        }
                    } while (choice_to_change != 7);
                    break;
                case 3:

users=mu1.readUserJsonFile("src/Model/Users/User_Data.j
son");
                    System.out.println("Enter the index no. of
Users");
                    int ind=sc.nextInt();
                    users.get(ind).display();
                    break;
                case 4:

users=mu1.readUserJsonFile("src/Model/Users/User_Data.j
son");
                    for(int i=0;i<users.size();i++){
```

```
                        users.get(i).display();
                    }
                    break;



                case 5:

users=mu1.readUserJsonFile("src/Model/Users/User_Data.j
son");
                    System.out.println("Enter the index no. of
Users");
                    int indp=sc.nextInt();
                    users.remove(indp);

mu1.writeUserJsonFile("src/Model/Users/User_Data.json",us
ers);
                    break;
//Taking user input for each Software object in the array
                case 6:
                    System.out.println("How do you wish to create
Software  Id: ");
                    System.out.println("1. Auto Generate Id and Add
no other Details");
                    System.out.println("2. Auto Generate Id and Add
other Details");
                    System.out.println("3. Provide Software Id");
                    System.out.print("Enter your choice: ");
                    int al_id_choice = sc.nextInt();
                    sc.nextLine();
                    if (al_id_choice == 1) {
                        System.out.println("Auto Generating
Software...");
                        application.add(new Applications());

ma1.writeAppJsonFile("src/Model/Softwares/Application_Dat
a.json",application);


                    } else if (al_id_choice==2) {
                        System.out.println("Enter the Application
name: ");
                        String app_name = sc.nextLine();
                        String[] app_name_split = app_name.split("
");


                        System.out.println("Enter the Application
Release Date: ");
                        String app_date = sc.nextLine();


                        System.out.println("Enter the Software name:
");
                        String soft_name = sc.nextLine();


                        System.out.println("Enter the Software Id: ");
                        int soft_Id = sc.nextInt();


                        System.out.println("Enter the Software Size:
");
                        int soft_size = sc.nextInt();


                        System.out.println("Enter the Application
Version: ");
                        String app_ver = sc.nextLine();
                        try {
                            if (app_ver.length() == 2.6) {
                                application.add(new
Applications(app_name_split[0], app_ver, app_date, soft_Id,
soft_name, soft_size));

ma1.writeAppJsonFile("src/Model/Softwares/Application_Dat
a.json", application);
                            } else {
                                throw new
appversionException("Version should be 2.6");
                            }
                        } catch (appversionException e) {
                            System.out.println("Custom Exception: " +
e.getMessage());
                        }




                        //application.add(new
Applications(app_name,app_ver,app_date,soft_Id,soft_name
,soft_size));
                        //
ma1.writeAppJsonFile("src/Model/Softwares/Application_Dat
a.json", application);
                    } else if (al_id_choice==3) {
                        System.out.println("Enter Application Id: ");
                        int app_ID= sc.nextInt();
                        // To avoid unnecessary skip of input
                        sc.nextLine();


                        System.out.println("Enter the Application
name: ");
                        String app_name = sc.nextLine();
                        String[] app_name_split = app_name.split("
");


                        System.out.println("Enter the Application
Release Date: ");
                        String app_date = sc.nextLine();


                        System.out.println("Enter the Software name:
");
                        String soft_name = sc.nextLine();


                        System.out.println("Enter Software Id: ");
                        int soft_Id= sc.nextInt();
                        // To avoid unnecessary skip of input
                        sc.nextLine();


                        System.out.println("Enter the Software Size:
");
                        int soft_size = sc.nextInt();


                        System.out.println("Enter the Application
Version: ");
                        String app_ver = sc.nextLine();
                        try {
                            if (app_ver.length() == 2.6) {
                                application.add(new
Applications(app_ID,app_name_split[0], app_ver, app_date,
soft_Id, soft_name, soft_size));

ma1.writeAppJsonFile("src/Model/Teachers/teachers.json",
application);
                            } else {
                                throw new
appversionException("Version should be 2.6");
                            }
                        } catch (appversionException e) {
                            System.out.println("Custom Exception: " +
e.getMessage());
                        }
```

```
//System.out.println(Applications.getApp_count() + " ");
                        //application.add(new
Applications(app_ID,app_name,app_ver,app_date,soft_Id,so
ft_name,soft_size));



//ma1.writeAppJsonFile("src/Model/Softwares/Application_D
ata.json",application);
                    }
                    break;
                case 7:
                    //Taking user input to display specific Airport
data
                    System.out.print("Enter Software Index to
Update: ");
                    int app_idx = sc.nextInt();
                    sc.nextLine();
                    int choice_to_change_app;
                    do {
                        System.out.println("1. Change Software Id");
                        System.out.println("2. Change Software
Name");
                        System.out.println("3. Change Software
Size");
                        System.out.println("4.Exit");


                        System.out.print("Enter your choice: ");
                        choice_to_change_app = sc.nextInt();
                        sc.nextLine();
                        switch (choice_to_change_app){
                            case 1:
                                System.out.print("Enter Software Id: ");
                                int soft_id = sc.nextInt();

application.get(app_idx).setSoftware_id(soft_id);

ma1.writeAppJsonFile("src/Model/Softwares/Application_Dat
a.json",application);
                                break;
                            case 2:
                                System.out.print("Enter Software Name:
");
                                String soft_name = sc.nextLine();
```

```
application.get(app_idx).setSoftware_name(soft_name);

ma1.writeAppJsonFile("src/Model/Softwares/Application_Dat
a.json",application);
                                break;
                            case 3:
                                System.out.print("Enter Software Size:
");
                                int soft_size= sc.nextInt();

application.get(app_idx).setSoftware_size(soft_size);

ma1.writeAppJsonFile("src/Model/Softwares/Application_Dat
a.json",application);
                                break;


                            case 4:
                                System.out.println("Thank You!");
                            default:
                                System.out.println("Invalid choice. Try
again.");
                                break;
                        }
                    } while (choice_to_change_app != 4);
                    break;
                case 8:

application=ma1.readAppJsonFile("src/Model/Softwares/App
lication_Data.json");
                    System.out.println("Enter the index no. of
Users");
                    int inda=sc.nextInt();
                    application.get(inda).display();
                    break;
                case 9:

application=ma1.readAppJsonFile("src/Model/Softwares/App
lication_Data.json");
                    for (Applications applications : application) {
                        applications.display();
                    }
                    break;
                case 10:

                    application=ma1.readAppJsonFile("src/Model/Softwares/Application_Data.json");

                    System.out.println("Enter the index no. of Users");

                    int indai=sc.nextInt();

                    application.remove(indai);

                    ma1.writeAppJsonFile("src/Model/Softwares/Application_Data.json",application);

                    break;


                case 11:

                    application = ma1.readAppJsonFile("src/Model/Softwares/Application_Data.json");

                    users = mu1.readUserJsonFile("src/Model/Users/User_Data.json");


                    System.out.println("Enter Application Index to be Licensed: ");

                    int app_idx_lic = sc.nextInt();

                    sc.nextLine();

                    System.out.println("Enter User Index to be Licensed: ");

                    int user_idx_lic = sc.nextInt();

                    sc.nextLine();

                    System.out.println("Enter the License Key: ");

                    String lic_key = sc.nextLine();

                    System.out.println("Enter the License cost: ");

                    int lic_cost = sc.nextInt();

                    sc.nextLine();

                    License li = new License(application.get(app_idx_lic),users.get(user_idx_lic),lic_key,lic_cost);

                    licenses.add(li);

                    ml1.writeLicensesJsonFile("src/Model/Licenses/Licenses.json",licenses);

                case 12:

                    application = ma1.readAppJsonFile("Model/Licenses/Licenses.json");

                    users = mu1.readUserJsonFile("src/Model/Users/User_Data.json");

                    System.out.println("Enter Application Index to Delete License: ");

                    int app_idx_lic_del = sc.nextInt();

                    sc.nextLine();

                    System.out.println("Enter User Index to Delete License: ");

                    int user_idx_lic_del = sc.nextInt();

                    sc.nextLine();


                    for(int i =0;i<licenses.size();i++){

                        if(app_idx_lic_del == licenses.get(i).getA_temp().getApp_id() && user_idx_lic_del == licenses.get(i).getU_temp().getUser_id()){

                            licenses.remove(i);

                        }

                    }

                    break;


                case 13:

                    licenses = ml1.readLicensesJsonFile("Model/Licenses/Licenses.json");


                    for (int i=0; i<licenses.size();i++)

                    {

                        System.out.println("Application Name : "+licenses.get(i).getA_temp().getApp_name()+" "+licenses.get(i).getA_temp().getSoftware_name());

                        System.out.println("User Name : "+licenses.get(i).getU_temp().getName());

                        System.out.println("License Key : "+licenses.get(i).getLicense_key());

                        System.out.println("License Cost : "+licenses.get(i).getLicense_cost());

                    }


                default:

                    // invalid choice

                    System.out.println("Invalid choice, please try again.");

                    break;

            }

        } while (choice !=14);

    }

}
```

Controller.java

```
package Controller;

import Model.Model;

import Model.Users.Users;

import View.View;


import javax.swing.*;

import java.awt.event.*;

import java.io.IOException;


public class Controller {

    Model model;

    View view;

    public Controller(Model m,View v){

        model = m;

        view = v;

view.getFf().getManageSoftwareBtn().addActionListener(new
ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                System.out.println("Software Button Clicked");

                view.getFf().setVisible(false);

                view.getMsof().setVisible(true);

            }

        });

        view.getMsof().addWindowListener(new
java.awt.event.WindowAdapter(){

            @Override

            public void
windowClosing(java.awt.event.WindowEvent windowEvent) {

                view.getFf().setVisible(true);

            }

        });


        model.getMa().setLinesBeingDisplayed(20);

view.centerInitialSetupSoftware(model.getMa().getLinesBein
gDisplayed(),model.getMa().getHeaders().size());


        model.getMa().setFirstLineToDisplay(0);

view.centerUpdateSoftware(model.getMa().getLines(model.g
etMa().getFirstLineToDisplay(),model.getMa().getLastLineTo
Display()),model.getMa().getHeaders());



view.getMsof().getSoftware_ip().getStp().addMouseWheelLis
tener(new MouseWheelListener() {

            @Override

            public void mouseWheelMoved(MouseWheelEvent
e) {

                int units = e.getUnitsToScroll();

                System.out.println(units);

                int current_first_line =
model.getMa().getFirstLineToDisplay();

                int current_last_line =
model.getMa().getLastLineToDisplay();

                int no_of_courses =
model.getMa().getTable().size();

                int no_of_display_lines =
model.getMa().getLinesBeingDisplayed();

                if(units <= 0 && current_first_line == 0)
                {
                    model.getMa().setFirstLineToDisplay(0);
                }
                else if(units <= 0 && current_first_line > 0)
                {
                    int new_first_line = current_first_line + units;
                    if(new_first_line <= 0)
                    {
                        model.getMa().setFirstLineToDisplay(0);
                    }
                    else
                    {
model.getMa().setFirstLineToDisplay(new_first_line);
                    }
                }
                else if(units > 0 && current_last_line ==
no_of_courses-1)
                {
model.getMa().setFirstLineToDisplay(current_first_line);
                }
                else if (units > 0 && current_last_line <
no_of_courses-1)
                {
                    int new_first_line = current_first_line + units;
                    if(new_first_line > no_of_courses -
no_of_display_lines)
                    {
                        new_first_line =
no_of_courses-no_of_display_lines;
model.getMa().setFirstLineToDisplay(new_first_line);
                    }
                    else
                    {
model.getMa().setFirstLineToDisplay(new_first_line);
                    }
                }

view.centerUpdateSoftware(model.getMa().getLines(model.getMa().getFirstLineToDisplay(),
model.getMa().getLastLineToDisplay()),
model.getMa().getHeaders());
            }
        });


view.getMsof().getSoftware_ip().getAsp().getAddSoftwareBtn
().addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String txt_software_id =
view.getMsof().getSoftware_ip().getAsp().getTxt_software_id
().getText();
                String txt_software_name =
view.getMsof().getSoftware_ip().getAsp().getTxt_software_n
ame().getText();
                String txt_software_size =
view.getMsof().getSoftware_ip().getAsp().getTxt_software_si
ze().getText();
                String txt_application_id =
view.getMsof().getSoftware_ip().getAsp().getTxt_application
_id().getText();
                String txt_application_name =
view.getMsof().getSoftware_ip().getAsp().getTxt_application
_name().getText();
                String txt_application_version =
view.getMsof().getSoftware_ip().getAsp().getTxt_application
_version().getText();
                String txt_application_release_date =
view.getMsof().getSoftware_ip().getAsp().getTxt_application
_release_date().getText();
                try {

model.getMa().addNewApplication(Integer.valueOf(txt_applic
ation_id),txt_application_name,txt_application_version,txt_ap
plication_release_date,Integer.valueOf(txt_software_id),txt_s
oftware_name,Integer.valueOf(txt_software_size));
                }catch (IOException ex){
                    throw new RuntimeException(ex);
                }
            }
        });


view.getMsof().getSoftware_ip().getEsp().getGetSoftwareBtn
().addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                System.out.println("Get Software Details Clicked");

                String txt_software_idx = view.getMsof().getSoftware_ip().getEsp().getTxt_get_software_idx().getText();

                model.getMa().readAppJsonFile("src/Model/Softwares/Application_Data.json");

                int software_id = model.getMa().getTable().get(Integer.valueOf(txt_software_idx)).getSoftware_id();

                String software_name = model.getMa().getTable().get(Integer.valueOf(txt_software_idx)).getSoftware_name();

                int software_size = model.getMa().getTable().get(Integer.valueOf(txt_software_idx)).getSoftware_size();

                int application_id = model.getMa().getTable().get(Integer.valueOf(txt_software_idx)).getApp_id();

                String application_name = model.getMa().getTable().get(Integer.valueOf(txt_software_idx)).getApp_name();

                String application_version = model.getMa().getTable().get(Integer.valueOf(txt_software_idx)).getVersion();

                String application_release_date = model.getMa().getTable().get(Integer.valueOf(txt_software_idx)).getRelease_date();



                view.getMsof().getSoftware_ip().getEsp().getTxt_software_id().setText(String.valueOf(software_id));

                view.getMsof().getSoftware_ip().getEsp().getTxt_software_name().setText(software_name);

                view.getMsof().getSoftware_ip().getEsp().getTxt_software_size().setText(String.valueOf(software_size));

                view.getMsof().getSoftware_ip().getEsp().getTxt_application_id().setText(String.valueOf(application_id));

                view.getMsof().getSoftware_ip().getEsp().getTxt_application_name().setText(application_name);

                view.getMsof().getSoftware_ip().getEsp().getTxt_application_version().setText(application_version);

                view.getMsof().getSoftware_ip().getEsp().getTxt_application_release_date().setText(application_release_date);

            }

        });

        view.getMsof().getSoftware_ip().getEsp().getEditSoftwareBtn().addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                System.out.println("Edit Software Button Clicked");

                String txt_software_idx = view.getMsof().getSoftware_ip().getEsp().getTxt_get_software_idx().getText();

                String txt_software_id = view.getMsof().getSoftware_ip().getEsp().getTxt_software_id().getText();

                String txt_software_name = view.getMsof().getSoftware_ip().getEsp().getName().toUpperCase();

                String txt_software_size = view.getMsof().getSoftware_ip().getEsp().getTxt_software_size().getText();

                String txt_application_idx = view.getMsof().getSoftware_ip().getEsp().getTxt_application_idx().getText();

                String txt_application_id = view.getMsof().getSoftware_ip().getEsp().getTxt_application_id().getText();

                String txt_application_name = view.getMsof().getSoftware_ip().getEsp().getTxt_application_name().getText();

                String txt_application_version = view.getMsof().getSoftware_ip().getEsp().getTxt_application_version().getText();

                String txt_application_release_date = view.getMsof().getSoftware_ip().getEsp().getTxt_application_release_date().getText();

                try {

                    model.getMa().editApplication(Integer.valueOf(txt_application_idx),Integer.valueOf(txt_software_idx),Integer.valueOf(txt_application_id),txt_application_name,txt_application_version,txt_application_release_date,Integer.valueOf(txt_software_id),txt_software_name,Integer.valueOf(txt_software_size));

                } catch (IOException ex){

                    throw new RuntimeException(ex);

                }


            }

        });


        view.getMsof().getSoftware_ip().getDsp().getDeleteSoftwareBtn().addActionListener(new ActionListener() {
```

```
      @Override

      public void actionPerformed(ActionEvent e) {

        System.out.println("Delete Software Button
Clicked");

        String txt_application_idx =
view.getMsof().getSoftware_ip().getDsp().getTxt_del_applica
tion_id().getText();

        try {

          int a_id =
model.getMa().deleteApplication(Integer.valueOf(txt_applicat
ion_idx));

          for (int i=0;i<model.getMl().getTable().size();i++){

            if (a_id ==
model.getMl().getTable().get(i).getA_temp().getApp_id()){

              model.getMl().deleteLicense(i);

            }

          }

        } catch (IOException ex){

          throw new RuntimeException(ex);

        }

      }

    });



view.getFf().getManageUserBtn().addActionListener(new
ActionListener() {

      @Override

      public void actionPerformed(ActionEvent e) {

        System.out.println("User Button Clicked");

        view.getFf().setVisible(false);

        view.getMuf().setVisible(true);

      }

    });

    view.getMuf().addWindowListener(new
java.awt.event.WindowAdapter(){

      @Override

      public void
windowClosing(java.awt.event.WindowEvent windowEvent) {

        view.getFf().setVisible(true);

      }

    });



    model.getMu().setLinesBeingDisplayed(20);
```

```
view.centerInitialSetupUser(model.getMu().getLinesBeingDis
played(),model.getMu().getHeaders().size());


    model.getMu().setFirstLineToDisplay(0);


view.centerUpdateUser(model.getMu().getLines(model.getM
u().getFirstLineToDisplay(),model.getMu().getLastLineToDisp
lay()),model.getMu().getHeaders());



view.getMuf().getIu().getCpu().addMouseWheelListener(new
MouseWheelListener() {

      @Override

      public void mouseWheelMoved(MouseWheelEvent
e) {

        int units = e.getUnitsToScroll();

        System.out.println(units);

        int current_first_line =
model.getMu().getFirstLineToDisplay();

        int current_last_line =
model.getMu().getLastLineToDisplay();

        int no_of_courses =
model.getMu().getTable().size();

        int no_of_display_lines =
model.getMu().getLinesBeingDisplayed();

        if(units <= 0 && current_first_line == 0)

        {

          model.getMu().setFirstLineToDisplay(0);

        }

        else if(units <= 0 && current_first_line > 0)

        {

          int new_first_line = current_first_line + units;

          if(new_first_line <= 0)

          {

            model.getMu().setFirstLineToDisplay(0);

          }

          else

          {


model.getMu().setFirstLineToDisplay(new_first_line);

          }

        }

        else if(units > 0 && current_last_line ==
no_of_courses-1)
```

```
        {

model.getMu().setFirstLineToDisplay(current_first_line);
        }
        else if (units > 0 && current_last_line <
no_of_courses-1)
        {
          int new_first_line = current_first_line + units;
          if(new_first_line > no_of_courses -
no_of_display_lines)
          {
            new_first_line =
no_of_courses-no_of_display_lines;

model.getMu().setFirstLineToDisplay(new_first_line);
          }
          else
          {

model.getMu().setFirstLineToDisplay(new_first_line);
          }
        }



view.centerUpdateUser(model.getMu().getLines(model.getM
u().getFirstLineToDisplay(),
model.getMu().getLastLineToDisplay()),
model.getMu().getHeaders());
      }
    });



view.getMuf().getIu().getApu().getAddUserBtn().addActionLis
tener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        String txt_user_id =
view.getMuf().getIu().getApu().getTxt_user_id().getText();
        String txt_user_rating =
view.getMuf().getIu().getApu().getTxt_user_rating().getText();
        String txt_id =
view.getMuf().getIu().getApu().getTxt_id().getText();
        String txt_Name =
view.getMuf().getIu().getApu().getTxt_Name().getText();
        String txt_address =
view.getMuf().getIu().getApu().getTxt_address().getText();
```

```
        String txt_mobile =
view.getMuf().getIu().getApu().getTxt_mobile().getText();
        try {

model.getMu().addNewUser(txt_Name,txt_address,txt_mobil
e,Integer.valueOf(txt_id),Integer.valueOf(txt_user_id),Integer.
valueOf(txt_user_rating));
        }catch (IOException ex){
          throw new RuntimeException(ex);
        }
      }
    });



view.getMuf().getIu().getEup().getGetUserBtn().addActionLis
tener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        System.out.println("Get User Details Clicked");
        String txt_user_idx =
view.getMuf().getIu().getEup().getTxt_get_user_idx().getText
();

model.getMu().readUserJsonFile("src/Model/Users/User_Dat
a.json");
        int user_id =
model.getMu().getTable().get(Integer.valueOf(txt_user_idx)).
getUser_id();
        int user_rating =
model.getMu().getTable().get(Integer.valueOf(txt_user_idx)).
getUser_rating();
        int id =
model.getMu().getTable().get(Integer.valueOf(txt_user_idx)).
getId();
        String Name =
model.getMu().getTable().get(Integer.valueOf(txt_user_idx)).
getName();
        String Address =
model.getMu().getTable().get(Integer.valueOf(txt_user_idx)).
getAddress();
        String mobile =
model.getMu().getTable().get(Integer.valueOf(txt_user_idx)).
getMobile();



view.getMuf().getIu().getEup().getTxt_user_id().setText(Strin
g.valueOf(user_id));

view.getMuf().getIu().getEup().getTxt_user_rating().setText(S
tring.valueOf(user_rating));
```

```
view.getMuf().getIu().getEup().getTxt_id().setText(String.valueOf(id));

view.getMuf().getIu().getEup().getTxt_Name().setText(Name);

view.getMuf().getIu().getEup().getTxt_address().setText(Address);

view.getMuf().getIu().getEup().getTxt_mobile().setText(mobile);

      }

    });



view.getMuf().getIu().getEup().getEditUserBtn().addActionListener(new ActionListener() {

      @Override

      public void actionPerformed(ActionEvent e) {

        System.out.println("Edit User Button Clicked");

        String txt_user_idx = view.getMuf().getIu().getEup().getTxt_get_user_idx().getText();

        String txt_user_id = view.getMuf().getIu().getEup().getTxt_user_id().getText();

        String txt_user_rating = view.getMuf().getIu().getEup().getTxt_user_rating().getText();

        String txt_person_idx = view.getMuf().getIu().getEup().getTxt_person_idx().getText();

        String txt_id = view.getMuf().getIu().getEup().getTxt_id().getText();

        String txt_Name = view.getMuf().getIu().getEup().getTxt_Name().getText();

        String txt_address = view.getMuf().getIu().getEup().getTxt_address().getText();

        String txt_mobile = view.getMuf().getIu().getEup().getTxt_mobile().getText();

        try {

model.getMu().editUser(Integer.valueOf(txt_person_idx),Integer.valueOf(txt_user_idx),txt_Name,txt_address,txt_mobile,Integer.valueOf(txt_id),Integer.valueOf(txt_user_id),Integer.valueOf(txt_user_rating));

        } catch (IOException ex){

          throw new RuntimeException(ex);

        }



      }

    });



view.getMuf().getIu().getDup().getDeleteUserBtn().addActionListener(new ActionListener() {

      @Override

      public void actionPerformed(ActionEvent e) {

        System.out.println("Delete Software Button Clicked");

        String txt_user_idx = view.getMuf().getIu().getDup().getTxt_del_user_id().getText();

        try {

          int u_id = model.getMu().deleteUser(Integer.valueOf(txt_user_idx));

          for (int i=0;i<model.getMl().getTable().size();i++){

            if (u_id == model.getMl().getTable().get(i).getU_temp().getUser_id()){

              model.getMl().deleteLicense(i);

            }

          }

        } catch (IOException ex){

          throw new RuntimeException(ex);

        }

      }

    });

view.getFf().getManageLicenseBtn().addActionListener(new ActionListener() {

      @Override

      public void actionPerformed(ActionEvent e) {

        System.out.println("License Button Clicked");

        view.getFf().setVisible(false);

        view.getMlf().setVisible(true);

      }

    });


    view.getMlf().addWindowListener(new WindowAdapter() {

      @Override

      public void windowClosing(java.awt.event.WindowEvent windowEvent) {

        view.getFf().setVisible(true);

      }
    });


    model.getMl().setLinesBeingDisplayed(20);

view.centerInitialSetupLicense(model.getMl().getLinesBeing
Displayed(),model.getMl().getHeaders().size());


    model.getMl().setFirstLineToDisplay(0);

view.centerUpdateLicenses(model.getMl().getLines(model.g
etMl().getFirstLineToDisplay(),model.getMl().getLastLineToDi
splay()),model.getMl().getHeaders());



view.getMlf().getLicense_ip().getLtp().addMouseWheelListen
er(new MouseWheelListener() {
      @Override
      public void mouseWheelMoved(MouseWheelEvent
e) {
        int units = e.getUnitsToScroll();
        System.out.println(units);
        int current_first_line =
model.getMl().getFirstLineToDisplay();
        int current_last_line =
model.getMl().getLastLineToDisplay();
        int no_of_courses =
model.getMl().getTable().size();
        int no_of_display_lines =
model.getMl().getLinesBeingDisplayed();
        if(units <= 0 && current_first_line == 0)
        {
          model.getMl().setFirstLineToDisplay(0);
        }
        else if(units <= 0 && current_first_line > 0)
        {
          int new_first_line = current_first_line + units;
          if(new_first_line <= 0)
          {
            model.getMl().setFirstLineToDisplay(0);
          }
          else
          {

model.getMl().setFirstLineToDisplay(new_first_line);
          }
        }
        else if(units > 0 && current_last_line ==
no_of_courses-1)
        {

model.getMl().setFirstLineToDisplay(current_first_line);
        }
        else if (units > 0 && current_last_line <
no_of_courses-1)
        {
          int new_first_line = current_first_line + units;
          if(new_first_line > no_of_courses -
no_of_display_lines)
          {
            new_first_line =
no_of_courses-no_of_display_lines;

model.getMl().setFirstLineToDisplay(new_first_line);
          }
          else
          {

model.getMl().setFirstLineToDisplay(new_first_line);
          }
        }



view.centerUpdateLicenses(model.getMl().getLines(model.g
etMl().getFirstLineToDisplay(),
model.getMl().getLastLineToDisplay()),
model.getMl().getHeaders());
      }
    });



view.getMlf().getLicense_ip().getAlp().getAddLicenseBtn().ad
dActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        String txt_user_id =
view.getMlf().getLicense_ip().getAlp().getTxt_user_id().getTe
xt();
        String txt_app_id =
view.getMlf().getLicense_ip().getAlp().getTxt_app_id().getTex
t();
        String txt_license_key =
view.getMlf().getLicense_ip().getAlp().getTxt_license_key().g
```

```
etText();
            String txt_license_cost = view.getMlf().getLicense_ip().getAlp().getTxt_license_cost().getText();
            try {
            model.getMl().addNewLicense(Integer.valueOf(txt_user_id),Integer.valueOf(txt_app_id),txt_license_key,Integer.valueOf(txt_license_cost));
            } catch (IOException ex){
                throw new RuntimeException(ex);
            }
        }
    });


    view.getMlf().getLicense_ip().getElp().getGetLicenseBtn().addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            System.out.println("Get License Details Clicked");
            String txt_license_idx = view.getMlf().getLicense_ip().getElp().getTxt_get_license_idx().getText();
            model.getMl().readLicensesJsonFile("src/Model/Licenses/Licenses.json");
            int u_id = model.getMl().getTable().get(Integer.valueOf(txt_license_idx)).getU_temp().getUser_id();
            int a_id = model.getMl().getTable().get(Integer.valueOf(txt_license_idx)).getA_temp().getApp_id();
            String license_key = model.getMl().getTable().get(Integer.valueOf(txt_license_idx)).getLicense_key();
            int license_cost = model.getMl().getTable().get(Integer.valueOf(txt_license_idx)).getLicense_cost();
            view.getMlf().getLicense_ip().getElp().getTxt_app_id().setText(String.valueOf(a_id));
            view.getMlf().getLicense_ip().getElp().getTxt_user_id().setText(String.valueOf(u_id));
            view.getMlf().getLicense_ip().getElp().getTxt_license_key().setText(license_key);
            view.getMlf().getLicense_ip().getElp().getTxt_license_cost().setText(String.valueOf(license_cost));
        }
    });


    view.getMlf().getLicense_ip().getElp().getEditLicenseBtn().addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            System.out.println("Edit License Button Clicked");
            String txt_license_idx = view.getMlf().getLicense_ip().getElp().getTxt_get_license_idx().getText();
            String txt_user_idx = view.getMlf().getLicense_ip().getElp().getTxt_user_id().getText();
            String txt_app_idx = view.getMlf().getLicense_ip().getElp().getTxt_app_id().getText();
            String txt_license_key = view.getMlf().getLicense_ip().getElp().getTxt_license_key().getText();
            String txt_license_cost = view.getMlf().getLicense_ip().getElp().getTxt_license_cost().getText();
            try {
            model.getMl().editLicense(Integer.valueOf(txt_license_idx),Integer.valueOf(txt_user_idx),Integer.valueOf(txt_app_idx),txt_license_key,Integer.valueOf(txt_license_cost));
            } catch (IOException ex){
                throw new RuntimeException(ex);
            }
        }
    });


    view.getMlf().getLicense_ip().getDlp().getDeleteLicenseBtn().addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            System.out.println("Delete License Button Clicked");
            String txt_license_idx = view.getMlf().getLicense_ip().getDlp().getTxt_del_license_idx().getText();
            try {
                    model.getMl().deleteLicense(Integer.valueOf(txt_license_idx));
                } catch (IOException ex){
                    throw new RuntimeException(ex);
                }
            }
        });
    }
}
```

View.java

```
package View;
import View.Software.ManageSoftwareFrame;
import View.License.ManageLicenseFrame;
import java.awt.*;
import java.util.ArrayList;
import java.util.Scanner;


public class View {
    FirstFrame ff;
    ManageUserFrame muf;
    ManageSoftwareFrame msof;
    ManageLicenseFrame mlf;


    public View(){
        ff = new FirstFrame();
        muf = new ManageUserFrame();
        msof = new ManageSoftwareFrame();
        mlf = new ManageLicenseFrame();
    }
    public void centerInitialSetupUser(int linesBeingDisplayed,int size){
        muf.getIu().getCpu().setLayout(new GridLayout(linesBeingDisplayed+1,size));
        muf.getIu().getCpu().createButtons((linesBeingDisplayed+1) * size);
    }
    public void centerInitialSetupSoftware(int linesBeingDisplayed, int size){
        msof.getSoftware_ip().getStp().setLayout(new GridLayout(linesBeingDisplayed+1,size));
        msof.getSoftware_ip().getStp().createButtons((linesBeingDisplayed+1) * size);
    }
    public void centerInitialSetupLicense(int linesBeingDisplayed, int size){
        mlf.getLicense_ip().getLtp().setLayout(new GridLayout(linesBeingDisplayed+1,size));
        mlf.getLicense_ip().getLtp().createButtons((linesBeingDisplayed+1)*size);
    }
    public void centerUpdateSoftware(ArrayList<ArrayList<String>> lines, ArrayList<String> headers){
        for (int i = 0; i < headers.size(); i++)
        {
            msof.getSoftware_ip().getStp().getAllButtons().get(i).setText(headers.get(i));
        }


        for (int software_row_no = 0; software_row_no < lines.size(); software_row_no++)
        {
            for (int software_col_no = 0; software_col_no < headers.size(); software_col_no++)
            {
                int button_no = software_row_no * headers.size() + headers.size() + software_col_no;
                String button_txt = lines.get(software_row_no).get(software_col_no);


                msof.getSoftware_ip().getStp().getAllButtons().get(button_no).setText(button_txt);
            }
        }
    }


    public void centerUpdateUser(ArrayList<ArrayList<String>> lines, ArrayList<String> headers){
        for (int i = 0; i < headers.size(); i++)
        {
            muf.getIu().getCpu().getAllButtons().get(i).setText(headers.get(i));
        }

        for (int user_row_no = 0; user_row_no < lines.size();
user_row_no++)
        {
            for (int user_col_no = 0; user_col_no <
headers.size(); user_col_no++)
            {
                int button_no = user_row_no * headers.size() +
headers.size() + user_col_no;
                String button_txt =
lines.get(user_row_no).get(user_col_no);


muf.getIu().getCpu().getAllButtons().get(button_no).setText(b
utton_txt);
            }
        }
    }
    public void
centerUpdateLicenses(ArrayList<ArrayList<String>> lines,
ArrayList<String> headers) {
        for (int i = 0; i < headers.size(); i++)
        {

mlf.getLicense_ip().getLtp().getAllButtons().get(i).setText(hea
ders.get(i));
        }

        for (int license_row_no = 0; license_row_no <
lines.size(); license_row_no++)
        {
            for (int license_col_no = 0; license_col_no <
headers.size(); license_col_no++)
            {
                int button_no = license_row_no * headers.size() +
headers.size() + license_col_no;
                String button_txt =
lines.get(license_row_no).get(license_col_no);


mlf.getLicense_ip().getLtp().getAllButtons().get(button_no).s
etText(button_txt);
            }
        }
    }
    public void setFf(FirstFrame ff) {
        this.ff = ff;
    }
    public FirstFrame getFf() {
        return ff;
    }
    public void setMuf(ManageUserFrame muf) {
        this.muf = muf;
    }
    public ManageUserFrame getMuf() {
        return muf;
    }
    public void setMsof(ManageSoftwareFrame msof) {
        this.msof = msof;
    }
    public ManageSoftwareFrame getMsof() {
        return msof;
    }
    public void setMlf(ManageLicenseFrame mlf) {
        this.mlf = mlf;
    }
    public ManageLicenseFrame getMlf() {
        return mlf;
    }
}
```

License.java

```
package Model.Licenses;
import Model.Softwares.*;
import Model.Users.*;

public class License {
    Applications a_temp;
    Users u_temp;
    String license_key;
    int license_cost;
    public License(Applications a,Users u,String l_key,int
l_cost){
        setA_temp(a);
        setU_temp(u);
        setLicense_cost(l_cost);
        setLicense_key(l_key);
    }

    public void setA_temp(Applications a_temp) {
        this.a_temp = a_temp;
    }

    public void setLicense_cost(int license_cost) {
        this.license_cost = license_cost;
    }

    public void setLicense_key(String license_key) {
        this.license_key = license_key;
    }

    public void setU_temp(Users u_temp) {
        this.u_temp = u_temp;
    }

    public Applications getA_temp() {
        return a_temp;
    }

    public Users getU_temp() {
        return u_temp;
    }

    public int getLicense_cost() {
        return license_cost;
    }

    public String getLicense_key() {return license_key;}
}
```

manageLicense.java

```
package Model.Licenses;

import Model.Softwares.*;
import Model.Users.*;
import Model.Displayable;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;

import java.io.File;
import java.io.IOException;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class manageLicense extends FileHandlingLicense
implements Displayable {

    ArrayList<Applications> apps = new ArrayList<>();
    ArrayList<Users> users = new ArrayList<>();
    ArrayList<License> license_data = new ArrayList<>();

    ObjectMapper objectMapper = new ObjectMapper();

    private int linesBeingDisplayed;
    private int firstLineIndex;
    private int lastLineIndex;
    private int highlightedLine;

    public manageLicense() {
        manageApplication ma1=new manageApplication();
        apps=ma1.getTable();
        manageUsers mu1=new manageUsers();
        users=mu1.getTable();

readLicensesJsonFile("src/Model/Licenses/Licenses.json");
    }
    public ArrayList<License> readLicensesJsonFile(String file_path) {
        try {
            JsonNode rootNode = objectMapper.readTree(new File(file_path));

            if (rootNode.isArray()) {
                for (JsonNode node : rootNode) {
                    int a_temp = node.has("a_temp")?node.get("a_temp").asInt():0;
                    int u_temp = node.has("u_temp")?node.get("u_temp").asInt():0;
                    String license_key = node.has("license_key")?node.get("license_key").asText():null;
                    int license_cost=node.has("license_cost")?node.get("license_cost").asInt():0;

                    Applications a_temp_obj=null;
                    Users u_temp_obj=null;

                    for(int i = 0; i < apps.size(); i++)
                    {
                        if(a_temp == apps.get(i).getApp_id())
                        {
                            a_temp_obj = apps.get(i);
                        }
                    }
                    for(int i = 0; i < users.size(); i++)
                    {
                        if(u_temp == users.get(i).getUser_id())
                        {
                            u_temp_obj = users.get(i);
                        }
                    }

                    License l_temp=new License(a_temp_obj,u_temp_obj,license_key,license_cost);
                    license_data.add(l_temp);
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return license_data;
    }

    public void writeLicensesJsonFile(String file_path, ArrayList<License> license_al) throws IOException {
        ArrayList<Map<String, Object>> license_to_be_written = new ArrayList<>();

        for (int i = 0; i < license_al.size(); i++) {
            HashMap<String, Object> data = new HashMap<>();
            data.put("a_temp", license_al.get(i).getA_temp().getApp_id());
            data.put("u_temp", license_al.get(i).getU_temp().getUser_id());
            data.put("enroll_date", license_al.get(i).getLicense_key());
            data.put("course_fees", license_al.get(i).getLicense_cost());

            license_to_be_written.add(data);
        }

        objectMapper.writeValue(Paths.get(file_path).toFile(), license_to_be_written);
    }

    public ArrayList<String> getHeaders() {
        ArrayList<String> headers = new ArrayList<String>();
        headers.add("Application Name");
        headers.add("User Name");
        headers.add("License Key");
        headers.add("License Cost");

        return headers;
    }

    public void setAppsTable(ArrayList<Applications> apps) {
        this.apps = apps;
    }


    @Override
    public ArrayList<String> getLine(int line) {
        ArrayList<String> license_datails = new ArrayList<String>();

license_datails.add(license_data.get(line).getA_temp().getApp_name());

license_datails.add(license_data.get(line).getU_temp().getName());

license_datails.add(license_data.get(line).getLicense_key());

license_datails.add(String.valueOf(license_data.get(line).getLicense_cost()));
        return license_datails;
    }


    @Override
    public ArrayList<ArrayList<String>> getLines(int firstLine, int lastLine) {
        ArrayList<ArrayList<String>> license_subset = new ArrayList<>();


        for (int i = firstLine; i <= lastLine; i++) {
            license_subset.add(getLine(i));
        }
        return license_subset;
    }


    @Override
    public int getFirstLineToDisplay() {
        return firstLineIndex;
    }


    @Override
    public int getLineToHighlight() {
        return highlightedLine;
    }


    @Override
    public int getLastLineToDisplay() {
        setLastLineToDisplay(getFirstLineToDisplay() + getLinesBeingDisplayed() - 1);
        return lastLineIndex;
    }


    @Override
    public int getLinesBeingDisplayed() {
        return linesBeingDisplayed;
    }


    @Override
    public void setFirstLineToDisplay(int firstLine) {
        this.firstLineIndex = firstLine;
    }


    @Override
    public void setLineToHighlight(int highlightedLine) {
        this.highlightedLine = highlightedLine;
    }


    @Override
    public void setLastLineToDisplay(int lastLine) {
        this.lastLineIndex = lastLine;
    }


    @Override
    public void setLinesBeingDisplayed(int numberOfLines) {
        this.linesBeingDisplayed = numberOfLines;
    }


    public ArrayList<License> getTable() {
        return license_data;
    }


    public void addNewLicense(int user_id,int Software_id,String license_key,int license_cost)throws IOException{

readLicensesJsonFile("src/Model/Licenses/Licenses.json");
        License temp_license = new
```

```
License(apps.get(Software_id), users.get(user_id),license_key,license_cost);
        license_data.add(temp_license);

writeLicensesJsonFile("src/Model/Licenses/Licenses.json",license_data);
    }


    public void editLicense(int edit_license_idx,int user_id,int Software_id,String license_key,int license_cost) throws IOException{

readLicensesJsonFile("src/Model/Licenses/Licenses.json");

license_data.get(edit_license_idx).setLicense_cost(license_cost);

license_data.get(edit_license_idx).setLicense_key(license_key);

license_data.get(edit_license_idx).setA_temp(apps.get(Software_id));

license_data.get(edit_license_idx).setU_temp(users.get(user_id));

writeLicensesJsonFile("src/Model/Licenses/Licenses.json",license_data);
    }


    public void deleteLicense(int delete_license_idx) throws IOException{

readLicensesJsonFile("src/Model/Licenses/Licenses.json");
        license_data.remove(delete_license_idx);

writeLicensesJsonFile("src/Model/Licenses/Licenses.json",license_data);
    }
}
```

FileHandlingLicense.java

```
package Model.Licenses;



import java.io.IOException;
import java.util.ArrayList;
public abstract class FileHandlingLicense{
    protected abstract ArrayList<License> readLicensesJsonFile(String file_path);
    protected abstract void writeLicensesJsonFile(String file_path, ArrayList<License> enrolls) throws IOException;
}
```

Applications.java

```
package Model.Softwares;


public class Applications extends Software{
    private static int app_count=0;
    int app_id;
    String app_name;
    String version;
    String release_date;


    public static int getApp_count(){ return app_count;}
    public static void setApp_count(int app_count){ Applications.app_count=app_count;}


    public Applications(){
        setApp_count(getApp_count()+1);
        this.setApp_id(app_count);
    }
    public  Applications(String app_name,String version,String release_date,int software_id,String Software_name, int Software_size){
        //calling the constructor of superclass(Software)
        super( software_id,Software_name,Software_size);
        //Incrementing Application_cnt and initializing App_id
        setApp_count(getApp_count()+1);
        this.setApp_id(app_count);
        this.setApp_name(app_name);
        this.setVersion(version);
        this.setRelease_date(release_date);
    }
    public  Applications(int app_id,String app_name,String version,String release_date,int software_id,String Software_name, int Software_size){
        //calling the constructor of superclass(Software)
        super( software_id,Software_name,Software_size);
        //Incrementing Application_cnt

        setApp_count(getApp_count()+1);

        this.setApp_id(app_id);

        this.setApp_name(app_name);

        this.setVersion(version);

        this.setRelease_date(release_date);

    }

    public void setApp_id(int app_id) { this.app_id=app_id;}

    public void setApp_name(String app_name){ this.app_name=app_name;}

    public void setVersion(String version){ this.version=version;}

    public void setRelease_date(String release_date){ this.release_date=release_date;}


    public int getApp_id(){ return app_id;}

    public String getApp_name(){ return app_name;}

    public String getVersion(){ return version;}

    public String getRelease_date(){ return release_date;}


    public void display()

    {

        System.out.println("App ID:"+getApp_id());

        System.out.println("App name:"+getApp_name());

        System.out.println("App version:"+getVersion());

        System.out.println("App release data:"+getRelease_date());

        super.display();

    }



}
```

Software.java

```
package Model.Softwares;


public class Software {

    int Software_id;

    String Software_name;

    int Software_size;

    public Software()

    {

        System.out.println("Creating a Software");

    }

    public Software(int software_id,String Software_name, int Software_size)

    {

        this.setSoftware_id(software_id);

        this.setSoftware_name(Software_name);

        this.setSoftware_size(Software_size);

    }

    public void setSoftware_id(int software_id) {

        this.Software_id = software_id;

    }

    public void setSoftware_name(String software_name) {

        Software_name = software_name;

    }

    public void setSoftware_size(int software_size) {

        Software_size = software_size;

    }


    public int getSoftware_id() {

        return Software_id;

    }

    public int getSoftware_size() {

        return Software_size;

    }

    public String getSoftware_name() {

        return Software_name;

    }

    public void display(){

        System.out.println("Software id: " + getSoftware_id());

        System.out.println("Software Name : " + getSoftware_name());

        System.out.println("Software size:" + getSoftware_size());

    }

}
```

manageApplication.java

```
package Model.Softwares;
```

```
import Model.Displayable;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;

import java.io.File;
import java.io.IOException;
import java.nio.file.Paths;
import java.util.ArrayList;

public class manageApplication extends FileHandlingApplications implements Displayable {

    private ArrayList<Applications> apps = new ArrayList<>();
    private ObjectMapper objectMapper = new ObjectMapper();

    private int linesBeingDisplayed;
    private int firstLineIndex;
    private int lastLineIndex;
    private int highlightedLine;

    public manageApplication() {
        readAppJsonFile("src/Model/Softwares/Application_Data.json");
    }

    public ArrayList<Applications> readAppJsonFile(String file_path) {
        try {
            JsonNode rootNode = objectMapper.readTree(new File(file_path));

            if (rootNode.isArray()) {
                for (JsonNode node : rootNode) {
                    int Software_id = node.has("Software_id")?node.get("Software_id").asInt():0;
                    String Software_name = node.has("Software_name")?node.get("Software_name").asText():null;
                    int Software_size = node.has("Software_size")?node.get("Software_size").asInt():500;
                    int app_id = node.has("app_id")?node.get("app_id").asInt():0;
                    String app_name = node.has("app_name")?node.get("app_name").asText():null;
                    String version = node.has("version")?node.get("version").asText():null;
                    String release_date = node.has("release_date")?node.get("release_date").asText():null;

                    apps.add(new Applications(app_id, app_name, version, release_date, Software_id, Software_name, Software_size));
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return apps;
    }

    public void writeAppJsonFile(String file_path, ArrayList<Applications> apps) throws IOException {
        objectMapper.writeValue(Paths.get(file_path).toFile(), apps);
    }

    public void setAppsTable(ArrayList<Applications> apps) {
        this.apps = apps;
    }

    public ArrayList<String> getHeaders() {
        ArrayList<String> headers = new ArrayList<>();
        headers.add("App ID");
        headers.add("App Name");
        headers.add("Version");
        headers.add("Release Date");
        headers.add("Software ID");
        headers.add("Software Name");
        headers.add("Software Size");
        return headers;
    }
    @Override
    public ArrayList<String> getLine(int line) {
        ArrayList<String> appDetails = new ArrayList<>();

        Applications app = apps.get(line);
        appDetails.add(String.valueOf(app.getApp_id()));
        appDetails.add(app.getApp_name());
        appDetails.add(app.getVersion());
        appDetails.add(app.getRelease_date());
        appDetails.add(String.valueOf(app.getSoftware_id()));
        appDetails.add(app.getSoftware_name());

appDetails.add(String.valueOf(app.getSoftware_size()));

        return appDetails;
    }

    @Override
    public ArrayList<ArrayList<String>> getLines(int firstLine,
int lastLine) {
        ArrayList<ArrayList<String>> appsSubset = new
ArrayList<>();

        for (int i = firstLine; i <= lastLine; i++) {
            appsSubset.add(getLine(i));
        }
        return appsSubset;
    }

    @Override
    public int getFirstLineToDisplay() {
        return firstLineIndex;
    }

    @Override
    public int getLineToHighlight() {
        return highlightedLine;
    }
```

```
    @Override
    public int getLastLineToDisplay() {
        setLastLineToDisplay(getFirstLineToDisplay() +
getLinesBeingDisplayed() - 1);
        return lastLineIndex;
    }

    @Override
    public int getLinesBeingDisplayed() {
        return linesBeingDisplayed;
    }

    @Override
    public void setFirstLineToDisplay(int firstLine) {
        this.firstLineIndex = firstLine;
    }

    @Override
    public void setLineToHighlight(int highlightedLine) {
        this.highlightedLine = highlightedLine;
    }

    @Override
    public void setLastLineToDisplay(int lastLine) {
        this.lastLineIndex = lastLine;
    }

    @Override
    public void setLinesBeingDisplayed(int numberOfLines) {
        this.linesBeingDisplayed = numberOfLines;
    }

    public ArrayList<Applications> getTable() {

readAppJsonFile("src/Model/Softwares/Application_Data.jso
n");
        return apps;
    }

    public void addNewApplication(int app_id,String
app_name,String app_version,String release_date,int
Software_id,String Software_name,int Software_size) throws IOException{

        readAppJsonFile("src/Model/Softwares/Application_Data.json");
        Applications temp_user = new Applications(app_id,app_name,app_version,release_date,Software_id,Software_name,Software_size);
        apps.add(temp_user);

        writeAppJsonFile("src/Model/Softwares/Application_Data.json",apps);
    }


    public void editApplication(int edit_software_idx,int edit_app_idx,int app_id,String app_name,String app_version,String release_date,int Software_id,String Software_name,int Software_size)throws  IOException{

        readAppJsonFile("src/Model/Softwares/Application_Data.json");
        apps.get(edit_app_idx).setApp_id(app_id);
        apps.get(edit_app_idx).setApp_name(app_name);
        apps.get(edit_app_idx).setVersion(app_version);
        apps.get(edit_app_idx).setRelease_date(release_date);

        apps.get(edit_software_idx).setSoftware_id(Software_id);

        apps.get(edit_software_idx).setSoftware_name(Software_name);

        apps.get(edit_software_idx).setSoftware_size(Software_size);

        writeAppJsonFile("src/Model/Softwares/Application_Data.json",apps);
    }


    public int deleteApplication(int del_app_idx) throws IOException{

        readAppJsonFile("src/Model/Softwares/Application_Data.json");
        apps.remove(del_app_idx);
        int a_id = apps.get(del_app_idx).getApp_id();

        writeAppJsonFile("src/Model/Softwares/Application_Data.json",apps);
        return a_id;
    }
}
```

FileHandling.java

```
package Model.Softwares;


import java.io.IOException;
import java.util.ArrayList;


public abstract class FileHandlingApplications {
    protected abstract ArrayList<Applications> readAppJsonFile(String file_path);


    protected abstract void writeAppJsonFile(String file_path, ArrayList<Applications> apps) throws IOException;
}
```

Users.java

```
package Model.Users;


public class Users extends Person {
    private static int user_count = 0;
    private int user_id;
    private int user_rating;


    public static int getUser_count() {
        return user_count;
    }




    public Users() {
        setUser_count(getUser_count() + 1);
        this.setUser_id(user_count);
    }


    public Users(String name, String address, String mobile, int id, int user_rating) {
        super(name, address, mobile, id);
        setUser_count(getUser_count() + 1);
        this.setUser_id(user_count);
        this.user_rating = user_rating;
    }

    public Users(String name, String address, String mobile,
int id, int user_id, int user_rating) {
        super(name, address, mobile, id);
        setUser_count(getUser_count()+1);
        this.setUser_id(user_id);
        this.user_rating = user_rating;
    }

    public static void setUser_count(int user_count) {
        Users.user_count = user_count;
    }
    public void setUser_id(int user_id) {
        this.user_id = user_id;
    }

    public int getUser_id() {
        return user_id;
    }

    public int getUser_rating() {
        return user_rating;
    }

    public void setUser_rating(int user_rating) {
        this.user_rating = user_rating;
    }

    @Override
    public void display() {
        System.out.println("User Id: " + getUser_id());
        super.display();
        System.out.println("User Rating: " + getUser_rating());
    }
}
```

Person.java

```
package Model.Users;

public class Person {
    private int id;
    private String Name;
    private String address;
    private String mobile;
    // Default constructor
    public Person() {
        System.out.println("Creating a Person");
    }
    // Parametrized constructor
    public Person(String Name,String address,String
Mobile,int id)
    {
        this.setName(Name);
        this.setAddress(address);
        this.setMobile(Mobile);
        this.setId(id);

    }
    //Getters and Setters
    public void setName(String Name) {this.Name = Name;}
    public void setAddress(String
address){this.address=address;}
    public void setMobile(String mobile) {this.mobile = mobile;}
    public void setId(int id){this.id=id;}
    public String getName() {return Name;}
    public String getAddress() {return address;}
    public String getMobile() {return mobile;}
    public int getId(){return id;}
    public void display() {
        System.out.println("Name: " + getName());
        System.out.println("Address: " + getAddress());
        System.out.println("Mobile: " + getMobile());
        System.out.println("Id:" + getId());
    }
}
```

FileHandling.java

```
package Model.Users;


import java.io.IOException;
import java.util.ArrayList;
/**
 * Identification comments:
 *   Name: Harsh Tanwani
 *   Experiment No: 06
 *   Experiment Title: Implementation of Abstract Class and Abstract Method for the entities of the relationship
 *   Experiment Date: 13/02/2024
 *   @version 1.0
 *
 *
 * Beginning comments:
 * Filename: FileHandlingPerson.java
 * Overview: This abstract class serves as a blueprint for handling application data through file operations.
 * In this file, we have achieved the following:
 * - Created abstract methods for reading and writing airline information to/from a JSON file
 */


//An abstract class for handling airline data through file operations.
public abstract class FileHandlingPerson {


    //Reads application information from a JSON file.
    protected abstract ArrayList<Users> readUserJsonFile(String file_path);


    //Writes airline information to a JSON file.
    protected abstract void writeUserJsonFile(String file_path, ArrayList<Users> users) throws IOException;
}
```

manageUsers.java

```
package Model.Users;


import Model.Displayable;
import Model.Softwares.Applications;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;


import java.io.File;
import java.io.IOException;
import java.nio.file.Paths;
import java.util.ArrayList;


public class manageUsers extends FileHandlingPerson implements Displayable {


    ArrayList<Users> users = new ArrayList<>();
    ObjectMapper objectMapper = new ObjectMapper();


    private int linesBeingDisplayed;
    private int firstLineIndex;
    private int lastLineIndex;
    private int highlightedLine;


    public manageUsers() {
        readUserJsonFile("src/Model/Users/User_Data.json");
    }


    public ArrayList<Users> readUserJsonFile(String file_path) {
        try {
            JsonNode rootNode = objectMapper.readTree(new File(file_path));


            if (rootNode.isArray()) {
                for (JsonNode node : rootNode) {
                    String name = node.has("Name") ? node.get("Name").asText() : null;
                    String address = node.has("address") ? node.get("address").asText() : null;
                    String mobile = node.has("mobile") ? node.get("mobile").asText() : null;
                    int id = node.has("id") ? node.get("id").asInt() : 0; // Or any default value
                    int user_id = node.has("user_id") ?
node.get("user_id").asInt() : 0;
                    int rating = node.has("user_rating") ?
node.get("user_rating").asInt() : 0; // Or any default value


                    users.add(new Users(name, address,
mobile,id,user_id,rating));
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }


        return users;
    }


    public void writeUserJsonFile(String file_path,
ArrayList<Users> users) throws IOException {
        objectMapper.writeValue(Paths.get(file_path).toFile(),
users);
    }


    @Override
    public ArrayList<String> getLine(int line) {
        ArrayList<String> user_details = new ArrayList<>();


        Users user = users.get(line);
        user_details.add(String.valueOf(user.getUser_id()));
        user_details.add(user.getName());
        user_details.add(user.getAddress());
        user_details.add(user.getMobile());
        user_details.add(String.valueOf(user.getId()));
        user_details.add(String.valueOf(user.getUser_rating()));


        return user_details;
    }


    @Override
    public ArrayList<ArrayList<String>> getLines(int firstLine,
int lastLine) {
        ArrayList<ArrayList<String>> users_subset = new
ArrayList<ArrayList<String>>();
```

```
        for (int i = firstLine; i <= lastLine; i++) {
            users_subset.add(getLine(i));
        }
        return users_subset;
    }


    @Override
    public int getFirstLineToDisplay() {
        return firstLineIndex;
    }


    @Override
    public int getLineToHighlight() {
        return highlightedLine;
    }


    @Override
    public int getLastLineToDisplay() {
        setLastLineToDisplay(getFirstLineToDisplay() +
getLinesBeingDisplayed() - 1);
        return lastLineIndex;
    }


    @Override
    public int getLinesBeingDisplayed() {
        return linesBeingDisplayed;
    }


    @Override
    public void setFirstLineToDisplay(int firstLine) {
        this.firstLineIndex = firstLine;
    }


    @Override
    public void setLineToHighlight(int highlightedLine) {
        this.highlightedLine = highlightedLine;
    }


    @Override
```

```
  public void setLastLineToDisplay(int lastLine) {
      this.lastLineIndex = lastLine;
  }

  @Override
  public void setLinesBeingDisplayed(int numberOfLines) {
      this.linesBeingDisplayed = numberOfLines;
  }

  public void setUsersTable(ArrayList<Users> users) {
      this.users = users;
  }

  public ArrayList<Users> getTable() {
      return users;
  }

  public ArrayList<String> getHeaders() {
      ArrayList<String> headers = new ArrayList<>();
      headers.add("Id");
      headers.add("Name");
      headers.add("Address");
      headers.add("Mobile");
      headers.add("User Id");
      headers.add("Rating");
      return headers;
  }
  public void addNewUser(String Name,String
Address,String mobile,int id,int user_id,int user_rating)
throws IOException{
      readUserJsonFile("src/Model/Users/User_Data.json");
      Users temp_user = new
Users(Name,Address,mobile,id,user_id,user_rating);
      users.add(temp_user);

writeUserJsonFile("src/Model/Users/User_Data.json",users);
  }

  public void editUser(int edit_person_idx,int
edit_user_idx,String Name,String Address,String mobile,int
id,int user_id,int user_rating)throws  IOException{
      readUserJsonFile("src/Model/Users/User_Data.json");
      users.get(edit_person_idx).setId(id);
      users.get(edit_person_idx).setName(Name);
      users.get(edit_person_idx).setAddress(Address);
      users.get(edit_person_idx).setMobile(mobile);
      users.get(edit_user_idx).setUser_id(user_id);
      users.get(edit_user_idx).setUser_rating(user_rating);

writeUserJsonFile("src/Model/Users/User_Data.json",users);
  }

  public int deleteUser(int del_user_idx) throws
IOException{
      readUserJsonFile("src/Model/Users/User_Data.json");
      users.remove(del_user_idx);
      int u_id = users.get(del_user_idx).getUser_id();

writeUserJsonFile("src/Model/Users/User_Data.json",users);
      return u_id;
  }
}
```

AddLicense.java

```
package View.License;
import javax.swing.*;
public class AddLicensePanel extends JPanel {
  JTextField txt_user_id;
  JTextField txt_app_id;
  JTextField txt_license_key;
  JTextField txt_license_cost;
  JButton addLicenseBtn;

  public AddLicensePanel(){
      txt_user_id = new JTextField();
      txt_app_id = new JTextField();
      txt_license_key = new JTextField();
      txt_license_cost = new JTextField();
      addLicenseBtn = new JButton("Add License");

      txt_user_id.setText("txt_user_id");
      txt_app_id.setText("txt_application_id");
```

```
        txt_license_key.setText("txt_license_key");
        txt_license_cost.setText("txt_license_cost");

        add(txt_user_id);
        add(txt_app_id);
        add(txt_license_key);
        add(txt_license_cost);
        add(addLicenseBtn);
    }

    public JTextField getTxt_user_id() {
        return txt_user_id;
    }

    public JTextField getTxt_license_key() {
        return txt_license_key;
    }

    public JTextField getTxt_license_cost() {
        return txt_license_cost;
    }

    public JButton getAddLicenseBtn() {
        return addLicenseBtn;
    }

    public void setTxt_user_id(JTextField txt_user_id) {
        this.txt_user_id = txt_user_id;
    }

    public void setTxt_license_key(JTextField txt_license_key)
{
        this.txt_license_key = txt_license_key;
    }

    public void setTxt_license_cost(JTextField
txt_license_cost) {
        this.txt_license_cost = txt_license_cost;
    }

    public void setAddLicenseBtn(JButton addLicenseBtn) {
        this.addLicenseBtn = addLicenseBtn;
    }

    public JTextField getTxt_app_id() {
        return txt_app_id;
    }

    public void setTxt_app_id(JTextField txt_app_id) {
        this.txt_app_id = txt_app_id;
    }
}
```

DeleteLicense.java

```
package View.License;

import javax.swing.*;

public class DeleteLicensePanel extends JPanel{
    JTextField txt_del_license_idx;
    JButton deleteLicenseBtn;

    public DeleteLicensePanel(){
        txt_del_license_idx = new JTextField();
        deleteLicenseBtn = new JButton("Delete License");
        txt_del_license_idx.setText("txt_license_idx");
        add(txt_del_license_idx);
        add(deleteLicenseBtn);
    }

    public void setTxt_del_license_idx(JTextField
txt_del_license_idx) {
        this.txt_del_license_idx = txt_del_license_idx;
    }

    public void setDeleteLicenseBtn(JButton
deleteLicenseBtn) {
        this.deleteLicenseBtn = deleteLicenseBtn;
    }
    public JTextField getTxt_del_license_idx() {
        return txt_del_license_idx;
    }

    public JButton getDeleteLicenseBtn() {
        return deleteLicenseBtn;
    }
}
```

EditLicense.java

```
package View.License;

import javax.swing.*;

public class EditLicensePanel extends JPanel {
    JTextField txt_get_license_idx;
    JButton getLicenseBtn;
    JTextField txt_app_id;
    JTextField txt_user_id;
    JTextField txt_license_key;
    JTextField txt_license_cost;
    JButton editLicenseBtn;

    public EditLicensePanel(){
        txt_app_id= new JTextField();
        txt_user_id = new JTextField();
        txt_license_cost = new JTextField();
        txt_license_key = new JTextField();
        txt_get_license_idx = new JTextField();
        editLicenseBtn = new JButton("Edit License");
        getLicenseBtn = new JButton("Get License to Edit");

        txt_user_id.setText("txt_user_id");
        txt_app_id.setText( "txt_app_id");
        txt_license_cost.setText("txt_license_cost");
        txt_license_key.setText("txt_license_key");
        txt_get_license_idx.setText("txt_get_license_idx");

        txt_user_id.setEditable(false);
txt_app_id.setEditable(false);

        add(txt_user_id);
        add (txt_app_id);
        add(txt_license_key);
        add(txt_license_cost);
        add(editLicenseBtn);
    }

    public void setTxt_app_id(JTextField txt_app_id) {
        this.txt_app_id = txt_app_id;
    }

    public void setTxt_user_id(JTextField txt_user_id) {
        this.txt_user_id = txt_user_id;
    }

    public void setTxt_license_cost(JTextField
txt_license_cost) {
        this.txt_license_cost = txt_license_cost;
    }

    public void setTxt_license_key(JTextField txt_license_key)
{
        this.txt_license_key = txt_license_key;
    }

    public void setTxt_get_license_idx(JTextField
txt_get_license_idx) {
        this.txt_get_license_idx = txt_get_license_idx;
    }

    public void setEditLicenseBtn(JButton editLicenseBtn) {
        this.editLicenseBtn = editLicenseBtn;
    }

    public void setGetLicenseBtn(JButton getLicenseBtn) {
        this.getLicenseBtn = getLicenseBtn;
    }
    public JTextField getTxt_app_id() {
        return txt_app_id;
    }

    public JTextField getTxt_user_id() {
        return txt_user_id;
    }

    public JTextField getTxt_license_cost() {
        return txt_license_cost;
    }

    public JTextField getTxt_license_key() {
        return txt_license_key;
    }

    public JButton getEditLicenseBtn() {
        return editLicenseBtn;
    }

    public JTextField getTxt_get_license_idx() {
        return txt_get_license_idx;
    }

    public JButton getGetLicenseBtn() {
        return getLicenseBtn;
    }
}
```

InitialPanelLicense.java

```
package View.License;
import View.License.AddLicensePanel;
import View.License.LicenseTablePanel;
import javax.swing.*;

public class InitialPanelLicense extends JPanel {
    private LicenseTablePanel ltp;
    private AddLicensePanel alp;
    private EditLicensePanel elp;
    private  DeleteLicensePanel dlp;
    public InitialPanelLicense(){
        super();
        ltp = new LicenseTablePanel();
        add(ltp);
        alp = new AddLicensePanel();
        add(alp);
        elp = new EditLicensePanel();
        add(elp);
        dlp = new DeleteLicensePanel();
        add(dlp);
    }

    public void setAlp(AddLicensePanel alp) {
        this.alp = alp;
    }

    public void setElp(EditLicensePanel elp) {
        this.elp = elp;
    }

    public void setDlp(DeleteLicensePanel dlp) {
        this.dlp = dlp;
    }

    public void setLtp(LicenseTablePanel ltp) {
        this.ltp = ltp;
    }

    public AddLicensePanel getAlp() {
        return alp;
    }

    public EditLicensePanel getElp() {
        return elp;
    }

    public DeleteLicensePanel getDlp() {
        return dlp;
    }

    public LicenseTablePanel getLtp() {
        return ltp;
    }
}
```

LicenseTablePanel.java

```
package View.License;
import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;
public class LicenseTablePanel extends JPanel {
    ArrayList<JButton> license_buttons = new ArrayList<>();

    public LicenseTablePanel()
    {
        super();
    }

    public void createButtons(int count)
    {
        for (int i = 1; i <= count; i++)
        {
            JButton b = new JButton();
            b.setBackground(Color.cyan);
            b.setSize(500,50);
            license_buttons.add(b);
            this.add(b);
            validate();
            repaint();
        }


    }

    public void setButtonText(int button_no, String button_text)
{
        license_buttons.get(button_no).setText(button_text);
    }

    public ArrayList<JButton> getAllButtons()
    {
        return license_buttons;
    }
}
```

ManageLicenseFrame.java

```
package View.License;

import View.ManageUserFrame;

import javax.swing.*;

public class ManageLicenseFrame extends JFrame {
    InitialPanelLicense license_ip;
    public ManageLicenseFrame(){
        super("Manage License Dashboard");
        license_ip = new InitialPanelLicense();
        add(license_ip);
        pack();
        setSize(500,500);
    }

    public void setLicense_ip(InitialPanelLicense license_ip) {
        this.license_ip = license_ip;
    }
    public  InitialPanelLicense getLicense_ip(){
        return license_ip;
    }
}
```

AddSoftwarePanel.java

```
package View.Software;

import javax.swing.*;
import java.awt.*;
```

```
public class AddSoftwarePanel extends JPanel {
    JTextField txt_software_id;
    JTextField txt_software_name;
    JTextField txt_software_size;
    JTextField txt_application_id;
    JTextField txt_application_name;
    JTextField txt_application_version;
    JTextField txt_application_release_date;
    JButton addSoftwareBtn;

    public AddSoftwarePanel(){
        txt_software_id = new JTextField();
        txt_software_name = new JTextField();
        txt_software_size = new JTextField();
        txt_application_id = new JTextField();
        txt_application_name = new JTextField();
        txt_application_version = new JTextField();
        txt_application_release_date = new JTextField();
        addSoftwareBtn = new JButton("Add Software");


        txt_software_id.setText("txt_software_id");
        txt_software_name.setText("txt_software_name");
        txt_software_size.setText("txt_software_size");
        txt_application_id.setText("txt_application_id");
        txt_application_name.setText("txt_application_name");
txt_application_version.setText("txt_application_version");
txt_application_release_date.setText("txt_application_release_date");


        add(txt_software_id);
        add(txt_software_name);
        add(txt_software_size);
        add(txt_application_id);
        add(txt_application_name);
        add(txt_application_version);
        add(txt_application_release_date);
        add(addSoftwareBtn);
    }
```

```
    public JTextField getTxt_software_id() {
        return txt_software_id;
    }


    public JTextField getTxt_software_name() {
        return txt_software_name;
    }


    public JTextField getTxt_software_size() {
        return txt_software_size;
    }


    public JButton getAddSoftwareBtn() {
        return addSoftwareBtn;
    }


    public JTextField getTxt_application_id() {
        return txt_application_id;
    }


    public JTextField getTxt_application_name() {
        return txt_application_name;
    }


    public JTextField getTxt_application_version() {
        return txt_application_version;
    }


    public JTextField getTxt_application_release_date() {
        return txt_application_release_date;
    }


    public void setTxt_software_id(JTextField txt_software_id) {
        this.txt_software_id = txt_software_id;
    }


    public void setTxt_software_name(JTextField txt_software_name) {
        this.txt_software_name = txt_software_name;
    }

    public void setTxt_software_size(JTextField
txt_software_size) {
        this.txt_software_size = txt_software_size;
    }

    public void setTxt_application_id(JTextField
txt_application_id) {
        this.txt_application_id = txt_application_id;
    }

    public void setTxt_application_name(JTextField
txt_application_name) {
        this.txt_application_name = txt_application_name;
    }

    public void setTxt_application_version(JTextField
txt_application_version) {
        this.txt_application_version = txt_application_version;
    }

    public void setTxt_application_release_date(JTextField
txt_application_release_date) {
        this.txt_application_release_date =
txt_application_release_date;
    }
    public void setAddSoftwareBtn(JButton addSoftwareBtn) {
        this.addSoftwareBtn = addSoftwareBtn;
    }
}
```

DeleteSoftwarePanel.java

```
package View.Software;

import javax.swing.*;

public class DeleteSoftwarePanel extends JPanel {
    JTextField txt_del_software_id;
    JTextField txt_del_application_id;
    JButton deleteSoftwareBtn;

    public DeleteSoftwarePanel(){
        txt_del_software_id = new JTextField();
        txt_del_application_id = new JTextField();
        deleteSoftwareBtn = new JButton("Delete Software");

        txt_del_software_id.setText("txt_software_id");
        txt_del_application_id.setText("txt_del_application_id");
        add(txt_del_software_id);
        add(txt_del_application_id);
        add(deleteSoftwareBtn);
    }

    public void setTxt_del_software_id(JTextField
txt_del_software_id) {
        this.txt_del_software_id = txt_del_software_id;
    }

    public void setTxt_del_application_id(JTextField
txt_del_application_id) {
        this.txt_del_application_id = txt_del_application_id;
    }

    public void setDeleteSoftwareBtn(JButton
deleteSoftwareBtn) {
        this.deleteSoftwareBtn = deleteSoftwareBtn;
    }

    public JTextField getTxt_del_software_id() {
        return txt_del_software_id;
    }

    public JTextField getTxt_del_application_id() {
        return txt_del_application_id;
    }

    public JButton getDeleteSoftwareBtn() {
        return deleteSoftwareBtn;
    }
}
```

EditSoftwarePanel.java

```
package View.Software;


import javax.swing.*;


public class EditsoftwarePanel extends JPanel{
    JTextField txt_get_software_idx;
    JButton getSoftwareBtn;
    JTextField txt_software_id;
    JTextField txt_software_name;
    JTextField txt_software_size;
    JTextField txt_application_idx;
    JTextField txt_application_id;
    JTextField txt_application_name;
    JTextField txt_application_version;
    JTextField txt_application_release_date;
    JButton editSoftwareBtn;


    public EditsoftwarePanel(){
        txt_software_id = new JTextField();
        txt_software_name = new JTextField();
        txt_software_size = new JTextField();
        txt_get_software_idx = new JTextField();
        txt_application_idx = new JTextField();
        txt_application_id = new JTextField();
        txt_application_name = new JTextField();
        txt_application_version = new JTextField();
        txt_application_release_date = new JTextField();
        getSoftwareBtn = new JButton("Get Software to Edit");
        editSoftwareBtn = new JButton("Edit Software");


        txt_software_id.setText("txt_software_id");
        txt_software_name.setText("txt_software_name");
        txt_software_size.setText("txt_software_size");
        txt_get_software_idx.setText("Get software id");
        txt_application_idx.setText("Get application id");
        txt_application_id.setText("txt_application_id");
        txt_application_name.setText("txt_application_name");
txt_application_version.setText("txt_application_version");

txt_application_release_date.setText("txt_application_release_date");


        add(txt_get_software_idx);
        add(txt_application_idx);
        add(getSoftwareBtn);
        add(txt_software_id);
        add(txt_software_name);
        add(txt_software_size);
        add(txt_application_id);
        add(txt_application_name);
        add(txt_application_version);
        add(txt_application_release_date);
        add(editSoftwareBtn);
    }


    public void setTxt_software_id(JTextField txt_software_id) {
        this.txt_software_id = txt_software_id;
    }


    public void setTxt_software_name(JTextField txt_software_name) {
        this.txt_software_name = txt_software_name;
    }


    public void setTxt_software_size(JTextField txt_software_size) {
        this.txt_software_size = txt_software_size;
    }


    public void setGetSoftwareBtn(JButton getSoftwareBtn) {
        this.getSoftwareBtn = getSoftwareBtn;
    }


    public void setTxt_get_software_idx(JTextField txt_get_software_idx) {
        this.txt_get_software_idx = txt_get_software_idx;
    }
    public void setTxt_application_idx(JTextField
txt_application_idx) {
        this.txt_application_idx = txt_application_idx;
    }

    public JTextField getTxt_application_id() {
        return txt_application_id;
    }

    public JTextField getTxt_application_idx() {
        return txt_application_idx;
    }

    public JTextField getTxt_application_name() {
        return txt_application_name;
    }

    public JTextField getTxt_application_version() {
        return txt_application_version;
    }

    public JTextField getTxt_application_release_date() {
        return txt_application_release_date;
    }

    public void setTxt_application_id(JTextField
txt_application_id) {
        this.txt_application_id = txt_application_id;
    }

    public void setTxt_application_name(JTextField
txt_application_name) {
        this.txt_application_name = txt_application_name;
    }

    public void setTxt_application_version(JTextField
txt_application_version) {
        this.txt_application_version = txt_application_version;
    }

    public void setTxt_application_release_date(JTextField
txt_application_release_date) {
        this.txt_application_release_date =
txt_application_release_date;
    }

    public void setEditSoftwareBtn(JButton editSoftwareBtn) {
        this.editSoftwareBtn = editSoftwareBtn;
    }

    public JTextField getTxt_software_id() {
        return txt_software_id;
    }

    public JTextField getTxt_software_name() {
        return txt_software_name;
    }

    public JTextField getTxt_software_size() {
        return txt_software_size;
    }

    public JTextField getTxt_get_software_idx() {
        return txt_get_software_idx;
    }

    public JButton getGetSoftwareBtn() {
        return getSoftwareBtn;
    }

    public JButton getEditSoftwareBtn() {
        return editSoftwareBtn;
    }
}
```

InitialPanelSoftware.java

```
package View.Software;
import javax.swing.*;
public class InitialPanelSoftware extends JPanel{
    private SoftwareTablePanel stp;
private AddSoftwarePanel asp;
private EditsoftwarePanel esp;
private DeleteSoftwarePanel dsp;

public InitialPanelSoftware(){
    super();
    stp = new SoftwareTablePanel();
    add(stp);
    asp = new AddSoftwarePanel();
    add(asp);
    esp = new EditsoftwarePanel();
    add(esp);
    dsp = new DeleteSoftwarePanel();
    add(dsp);
}

public void setStp(SoftwareTablePanel stp) {
    this.stp = stp;
}

public void setEsp(EditsoftwarePanel esp) {
    this.esp = esp;
}

public void setDsp(DeleteSoftwarePanel dsp) {
    this.dsp = dsp;
}

public void setAsp(AddSoftwarePanel asp) {
    this.asp = asp;
}

public SoftwareTablePanel getStp() {
    return stp;
}

public EditsoftwarePanel getEsp() {
    return esp;
}

    public AddSoftwarePanel getAsp() {
        return asp;
    }

    public DeleteSoftwarePanel getDsp() {
        return dsp;
    }
}
```

ManageSoftwareFrame.java

```
package View.Software;
import View.InitialPanelUser;


import javax.swing.*;
public class ManageSoftwareFrame extends JFrame {
    InitialPanelSoftware software_ip;


    public ManageSoftwareFrame(){
        super("Manage Software Dashboard");
        software_ip = new InitialPanelSoftware();
        add(software_ip);
        pack();
        setSize(500,500);
    }

    public void setSoftware_ip(InitialPanelSoftware
software_ip) {
        this.software_ip = software_ip;
    }

    public InitialPanelSoftware getSoftware_ip() {
        return software_ip;
    }
}
```

SoftwareTablePanel.java

```
package View.Software;
```

```
import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;

public class SoftwareTablePanel extends JPanel {
    ArrayList<JButton> software_buttons = new ArrayList<>();

    public SoftwareTablePanel()
    {
        super();
    }

    public void createButtons(int count)
    {
        for (int i = 1; i <= count; i++)
        {
            JButton b = new JButton();
            b.setBackground(Color.YELLOW);
            b.setSize(100,100);
            software_buttons.add(b);
            this.add(b);
            validate();
            repaint();
        }


    }

    public void setButtonText(int button_no, String button_text)
{
        software_buttons.get(button_no).setText(button_text);
    }

    public ArrayList<JButton> getAllButtons()
    {
        return software_buttons;
    }
}
```

AddUserPanel.java

```
package View;


import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;
public class AddPanelUser extends JPanel {
    JTextField txt_Name;
    JTextField txt_address;
    JTextField txt_mobile;
    JTextField txt_id;
    JTextField txt_user_id;
    JTextField txt_user_rating;
    JButton addUserBtn;
    public AddPanelUser() {
        txt_Name = new JTextField();
        txt_address = new JTextField();
        txt_mobile = new JTextField();
        txt_id = new JTextField();
        txt_user_id = new JTextField();
        txt_user_rating = new JTextField();
        addUserBtn = new JButton("Add a User");

        txt_Name.setText("Name: ");
        txt_address.setText("Address: ");
        txt_mobile.setText("Mobile: ");
        txt_id.setText("Id: ");
        txt_user_id.setText("User Id: ");
        txt_user_rating.setText("User Rating: ");

        add(txt_Name);
        add(txt_address);
        add(txt_mobile);
        add(txt_id);
        add(txt_user_id);
        add(txt_user_rating);
        add(addUserBtn);
    }

    public void setTxt_Name(JTextField txt_Name) {
        this.txt_Name = txt_Name;
    }

    public void setTxt_address(JTextField txt_address) {
        this.txt_address = txt_address;
    }

    public void setTxt_mobile(JTextField txt_mobile) {
        this.txt_mobile = txt_mobile;
    }

    public void setTxt_id(JTextField txt_id) {
        this.txt_id = txt_id;
    }

    public void setTxt_user_id(JTextField txt_user_id) {
        this.txt_user_id = txt_user_id;
    }

    public void setTxt_user_rating(JTextField txt_user_rating)
{
        this.txt_user_rating = txt_user_rating;
    }

    public void setAddUserBtn(JButton addUserBtn) {
        this.addUserBtn = addUserBtn;
    }

    public JTextField getTxt_Name() {
        return txt_Name;
    }

    public JTextField getTxt_address() {
        return txt_address;
    }

    public JTextField getTxt_mobile() {
        return txt_mobile;
    }
    public JTextField getTxt_id() {
        return txt_id;
    }

    public JTextField getTxt_user_id() {
        return txt_user_id;
    }

    public JTextField getTxt_user_rating() {
        return txt_user_rating;
    }

    public JButton getAddUserBtn() {
        return addUserBtn;
    }
}
```

CenterPanelUser.java

```
package View;
import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;



public class CenterPanelUser extends JPanel {
    ArrayList<JButton> buttons = new ArrayList<>();

    public CenterPanelUser()
    {
        super();
    }

    public void createButtons(int count)
    {
        for (int i = 1; i <= count; i++)
        {
            JButton b = new JButton();
            b.setBackground(Color.green);
            b.setSize(500,50);
            buttons.add(b);

            this.add(b);

            b.validate();

            b.repaint();

        }


    }


    public void setButtonText(int button_no, String button_text) {

        buttons.get(button_no).setText(button_text);

    }


    public ArrayList<JButton> getAllButtons()

    {

        return buttons;

    }

}
```

DeleteUserPanel.java

```
package View;


import javax.swing.*;


public class DeleteUserPanel extends JPanel{

    JTextField txt_del_user_id;

    JTextField txt_del_person_id;

    JButton deleteUserBtn;


    public DeleteUserPanel(){

        txt_del_user_id = new JTextField();

        txt_del_person_id = new JTextField();

        deleteUserBtn = new JButton("Delete User");


        txt_del_user_id.setText("txt_del_user_id");

        txt_del_person_id.setText("txt_del_person_id");

        add(txt_del_person_id);

        add(txt_del_user_id);

        add(deleteUserBtn);

    }


    public void setTxt_del_person_id(JTextField txt_del_person_id) {

        this.txt_del_person_id = txt_del_person_id;

    }


    public void setTxt_del_user_id(JTextField txt_del_user_id) {

        this.txt_del_user_id = txt_del_user_id;

    }


    public void setDeleteUserBtn(JButton deleteUserBtn) {

        this.deleteUserBtn = deleteUserBtn;

    }


    public JTextField getTxt_del_person_id() {

        return txt_del_person_id;

    }


    public JTextField getTxt_del_user_id() {

        return txt_del_user_id;

    }


    public JButton getDeleteUserBtn() {

        return deleteUserBtn;

    }

}
```

EditUserPanel.java

```
package View;


import javax.swing.*;


public class EditUserPanel extends JPanel {

    JTextField txt_get_user_idx;

    JButton getUserBtn;

    JTextField txt_user_id;

    JTextField txt_user_rating;

    JTextField txt_person_idx;
JTextField txt_id;
JTextField txt_Name;
JTextField txt_address;
JTextField txt_mobile;
JButton editUserBtn;

public EditUserPanel(){
    txt_user_id = new JTextField();
    txt_user_rating = new JTextField();
    txt_get_user_idx = new JTextField();
    txt_id = new JTextField();
    txt_Name = new JTextField();
    txt_address = new JTextField();
    txt_mobile = new JTextField();
    txt_person_idx = new JTextField();
    getUserBtn = new JButton("Get User to Edit");
    editUserBtn = new JButton("Edit User");

    txt_user_id.setText("txt_user_id");
    txt_user_rating.setText("txt_user_rating");
    txt_get_user_idx.setText("Get User id");
    txt_id.setText("txt_id");
    txt_Name.setText("txt_Name");
    txt_address.setText("txt_address");
    txt_mobile.setText("txt_mobile");
    txt_person_idx.setText("Get Person id");

    add(txt_get_user_idx);
    add(txt_person_idx);
    add(txt_user_id);
    add(txt_user_rating);
    add(txt_id);
    add(txt_Name);
    add(txt_address);
    add(txt_mobile);
    add(editUserBtn);
    add(getUserBtn);
}
```

```
    public void setTxt_user_id(JTextField txt_user_id) {
        this.txt_user_id = txt_user_id;
    }

    public void setTxt_user_rating(JTextField txt_user_rating)
{
        this.txt_user_rating = txt_user_rating;
    }

    public void setTxt_get_user_idx(JTextField
txt_get_user_idx) {
        this.txt_get_user_idx = txt_get_user_idx;
    }

    public void setTxt_id(JTextField txt_id) {
        this.txt_id = txt_id;
    }

    public void setTxt_Name(JTextField txt_Name) {
        this.txt_Name = txt_Name;
    }

    public void setTxt_person_idx(JTextField txt_person_idx) {
        this.txt_person_idx = txt_person_idx;
    }

    public void setTxt_mobile(JTextField txt_mobile) {
        this.txt_mobile = txt_mobile;
    }

    public void setTxt_address(JTextField txt_address) {
        this.txt_address = txt_address;
    }

    public void setEditUserBtn(JButton editUserBtn) {
        this.editUserBtn = editUserBtn;
    }

    public void setGetUserBtn(JButton getUserBtn) {
        this.getUserBtn = getUserBtn;
    }

    public JTextField getTxt_user_id() {
        return txt_user_id;
    }

    public JTextField getTxt_user_rating() {
        return txt_user_rating;
    }

    public JTextField getTxt_get_user_idx() {
        return txt_get_user_idx;
    }

    public JTextField getTxt_person_idx() {
        return txt_person_idx;
    }

    public JTextField getTxt_id() {
        return txt_id;
    }

    public JTextField getTxt_mobile() {
        return txt_mobile;
    }

    public JTextField getTxt_Name() {
        return txt_Name;
    }

    public JTextField getTxt_address() {
        return txt_address;
    }

    public JButton getEditUserBtn() {
        return editUserBtn;
    }

    public JButton getGetUserBtn() {
        return getUserBtn;
    }
}
```

FirstFrame.java

```
package View;

import javax.swing.*;
import java.awt.*;

public class FirstFrame extends JFrame {
    JButton manageUserBtn;
    JButton manageSoftwareBtn;
    JButton manageLicenseBtn;
    JPanel firstPanel;

    FirstFrame(){
        super("Main Dashboard");
        manageUserBtn = new JButton("Manage User");
        manageSoftwareBtn = new JButton("Manage Software");
        manageLicenseBtn = new JButton("Manage License");

        firstPanel = new JPanel();
        firstPanel.setLayout(new GridLayout(3,1,20,20));
        firstPanel.add(manageUserBtn);
        firstPanel.add(manageSoftwareBtn);
        firstPanel.add(manageLicenseBtn);
        add(firstPanel);
        pack();
        setSize(500,500);
        setVisible(true);
    }

    public void setFirstPanel(JPanel firstPanel) {
        this.firstPanel = firstPanel;
    }

    public void setManageUserBtn(JButton manageUserBtn) {
        this.manageUserBtn = manageUserBtn;
    }

    public void setManageSoftwareBtn(JButton
manageSoftwareBtn) {
        this.manageSoftwareBtn = manageSoftwareBtn;
    }

    public void setManageLicenseBtn(JButton
manageLicenseBtn) {
        this.manageLicenseBtn = manageLicenseBtn;
    }

    public JPanel getFirstPanel() {
        return firstPanel;
    }

    public JButton getManageUserBtn() {
        return manageUserBtn;
    }

    public JButton getManageSoftwareBtn() {
        return manageSoftwareBtn;
    }

    public JButton getManageLicenseBtn() {
        return manageLicenseBtn;
    }

}
```

InitialPanelUser.java

```
package View;
import javax.swing.*;

public class InitialPanelUser extends JPanel{
    private CenterPanelUser cpu;
    private AddPanelUser apu;
    private EditUserPanel eup;
    private DeleteUserPanel dup;
    public InitialPanelUser(){
        super();
        cpu = new CenterPanelUser();
        add(cpu);
        apu = new AddPanelUser();
        add(apu);
        eup  = new EditUserPanel();
        add(eup);
        dup = new DeleteUserPanel();
        add(dup);
    }
    public CenterPanelUser getCpu(){return cpu;}
    public EditUserPanel getEup(){return eup;}

    public DeleteUserPanel getDup(){return dup;}

    public void setEup(EditUserPanel eup) {
        this.eup = eup;
    }

    public void setDup(DeleteUserPanel dup) {
        this.dup = dup;
    }

    public void setCpu(CenterPanelUser cpu) {
        this.cpu = cpu;
    }

    public void setApu(AddPanelUser apu) {
        this.apu = apu;
    }

    public AddPanelUser getApu() {
        return apu;
    }
}
```

ManageUserFrame.java

```
package View;
```

```
import javax.swing.*;
public class ManageUserFrame extends JFrame {
    InitialPanelUser iu;
    public ManageUserFrame(){
        super("Manage User Dashboard");
        iu = new InitialPanelUser();
        add(iu);
        pack();
        setSize(500,500);
    }

    public void setIu(InitialPanelUser iu) {
        this.iu = iu;
    }

    public InitialPanelUser getIu() {
        return iu;
    }
}
```

2.6 Conclusion and Future work

In conclusion, the provided program offers a comprehensive solution for managing users, software applications, and licenses through a menu-driven interface. By adhering to object-oriented principles and employing a Model-View-Controller (MVC) architectural pattern, the program ensures modularity, flexibility, and maintainability.

Through interactive functionalities, users can perform various operations such as adding, updating, displaying, and deleting entities with ease. Exception handling mechanisms guarantee robustness by gracefully managing errors and ensuring data integrity. Additionally, the utilization of JSON file handling facilitates data persistence, allowing users to store and retrieve information across sessions seamlessly.

Overall, this program provides an effective and user-friendly platform for administrative tasks in environments requiring efficient management of users, applications, and licenses. Its structured design and intuitive interface make it a valuable tool for enhancing productivity and organization in diverse settings.

References

https://www.javatpoint.com/

https://www.mockaroo.com/

Object-oriented Modeling and design with UML by Modelsoft Consulting Corporation