School Management System JavaFX

Project Created using Java and JavaFX

PREPARED BY:

For name: Eshwanth Tokala

Project Documentation

Team Members: Eshwanth Tokala and Student #: 991704828

Summary

The School Management System JavaFX is a desktop application designed to facilitate the management of school-related activities. The application will provide a user-friendly interface for teachers, students, and administrators to manage their daily tasks, including attendance, grades, and schedules.

The application will be developed using Java 17 and the JavaFX framework, which allows for the creation of visually appealing and responsive user interfaces. The development environment used will be IntelliJ IDEA, a popular IDE for Java development.

To design the user interface, the application will utilize SceneBuilder, a drag-and-drop GUI builder for JavaFX. SceneBuilder makes it easy to create and modify the layout of the user interface.

To store data, the application will make use of file I/O. The application will save user data, including student and teacher profiles, attendance records, and grade data to files on disk. When the application starts up, it will load the data from the files to populate the various views and tables.

The application will utilize a dynamic hash map to store and manage student and teacher information efficiently. The hash map will allow for fast and easy retrieval of data based on the unique keys for each student or teacher. The hash map will be dynamically resized as the number of students or teachers increases or decreases.

In addition to the hash map, the application will also make use of dynamic arrays to store and manage data. For example, the application will use dynamic arrays to store and display student schedules and course offerings. The dynamic arrays will be resized as needed to accommodate changes in the number of courses or students.

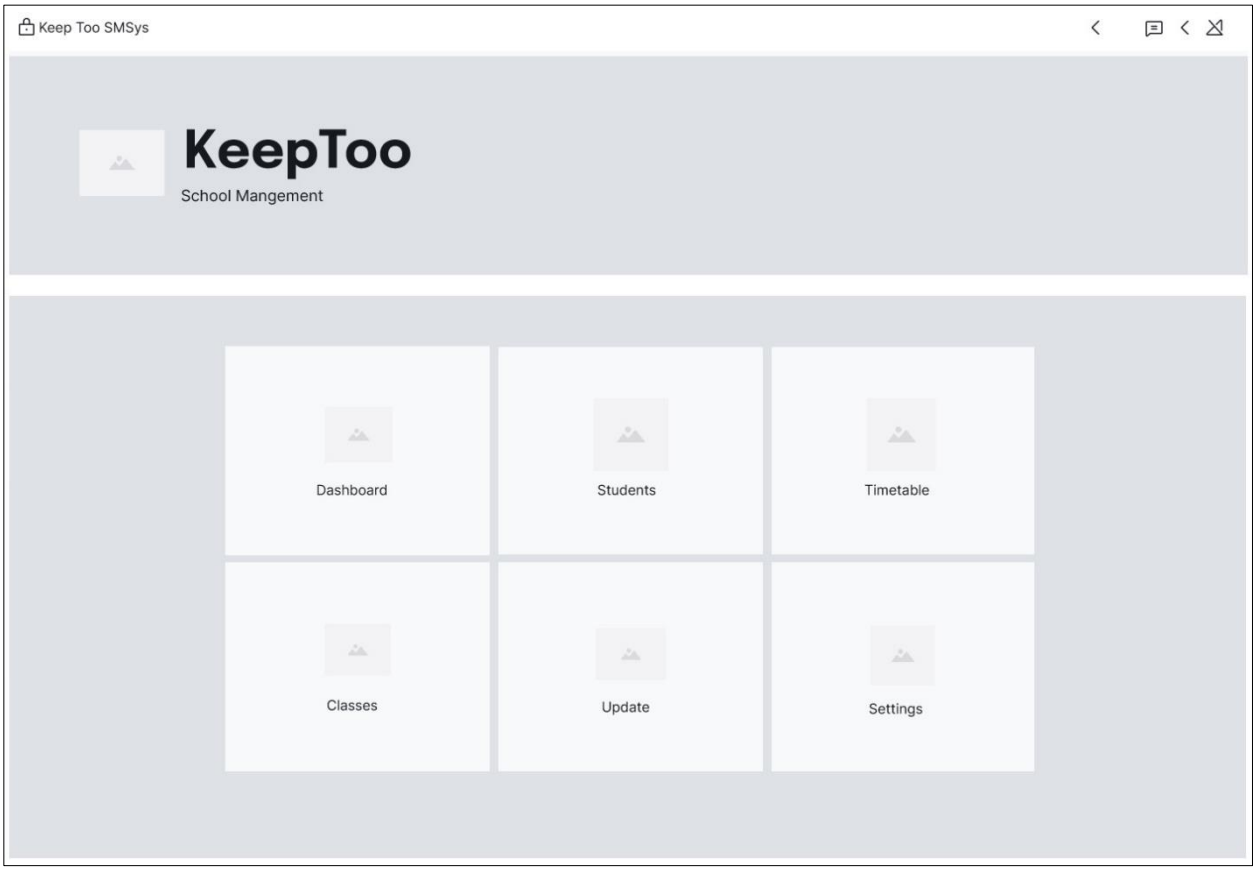
The application will also utilize graphics and charts to provide visual representation of data. For example, the application will generate charts and graphs to display student progress and performance over time. This will allow teachers and administrators to quickly identify areas

where students may need additional support. Overall, the School Management System JavaFX will provide a comprehensive solution for managing school-related tasks.

To summarize, the School Management System JavaFX will provide a comprehensive solution for managing school-related tasks, utilizing technologies such as JavaFX, SceneBuilder, file I/O, dynamic arrays, hash maps, graphics, and charts. The application will include features such as user authentication and permissions, and will allow teachers, students, and administrators to manage their daily tasks with ease.

UI/UX Interface

Landing Screen



The landing screen of the School Management System JavaFX will be the first screen that users see when they launch the application. The landing screen will serve as the gateway to the

different functionalities of the application, providing users with quick and easy access to the features they need.

The landing screen will be designed using JavaFX and SceneBuilder. It will consist of various panels and widgets that will provide an overview of the different sections of the application. The layout of the landing screen will be simple and intuitive, allowing users to navigate to different sections of the application with ease.

The landing screen will include panels for managing attendance, grades, schedules, and student profiles, similar to the dashboard. Each panel will display relevant information in a summarized format, allowing users to quickly view and understand the state of their data.

The attendance panel will display a summary of the overall attendance status of the school, highlighting the number of present, absent, and tardy students. This will allow administrators to quickly identify any attendance issues and take corrective actions.

The grades panel will display a summary of the overall performance of students in the school. This will include average grades, performance trends, and comparisons with previous semesters. This will allow administrators and teachers to track student progress and identify areas for improvement.

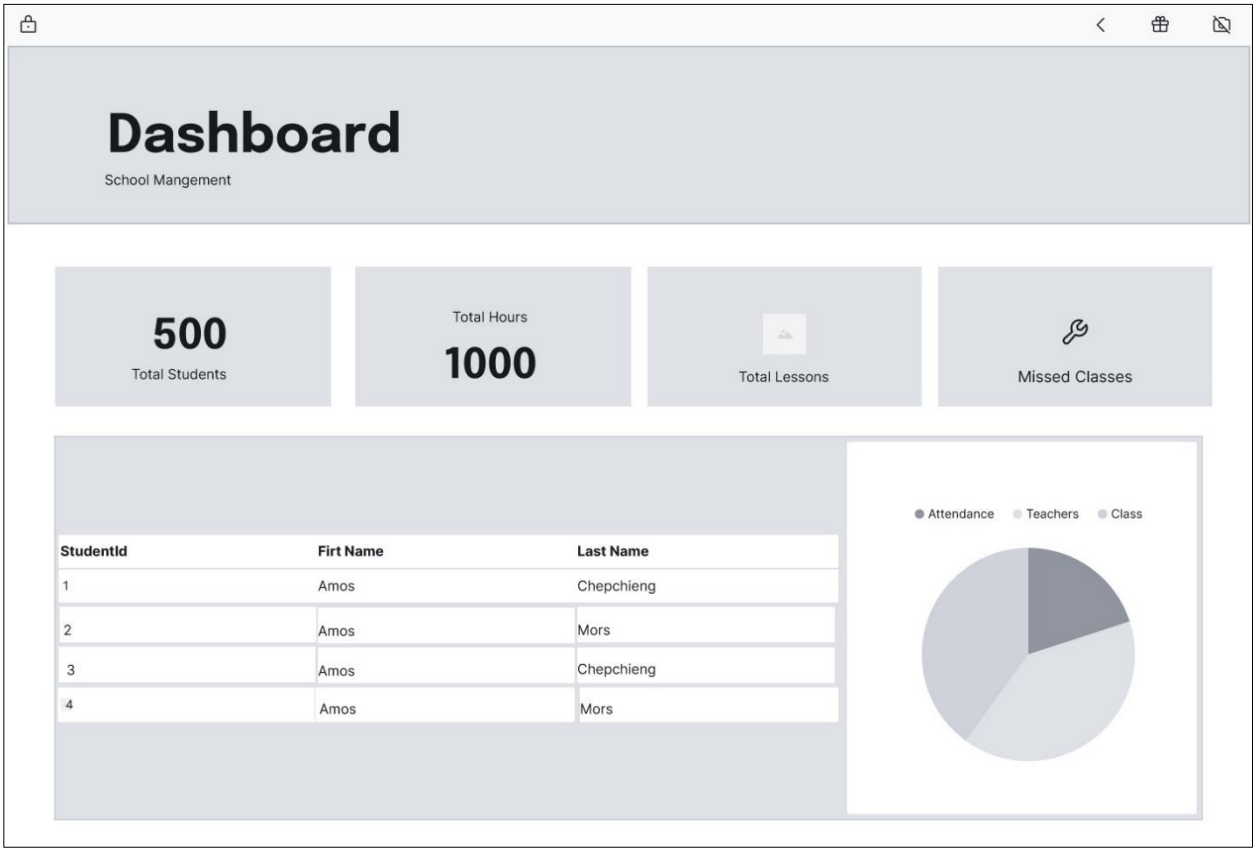
The schedules panel will display a summary of the course offerings and schedules for the current semester. This will include the number of courses, the number of students, and the number of teachers involved. This will allow administrators to quickly identify scheduling issues and take corrective actions.

The landing screen will also include quick links to the different functionalities of the application, such as the Landing Main Page, Dashboard panel, Student Panel, Timetable Panel. This will allow users to quickly access the features they need without having to navigate through different menus.

Overall, the landing screen of the School Management System JavaFX will provide an easy-to-use interface that will allow users to quickly access the different functionalities of the application. The landing screen will utilize technologies such as JavaFX and SceneBuilder to

create a visually appealing and responsive user interface, making it a powerful tool for school management.

Dashboard Panel



The dashboard of the School Management System JavaFX will be the central hub for accessing various functionalities of the application. The dashboard will provide a user-friendly interface that will allow teachers, students, and administrators to manage their daily tasks with ease.

The dashboard will be designed using JavaFX and SceneBuilder. It will consist of various panels and widgets that will provide quick access to different features of the application. The layout of the dashboard will be intuitive, making it easy for users to navigate the various panels and widgets.

The dashboard will include panels for managing attendance, grades, schedules, and student profiles. Each panel will display relevant information in a tabular format, allowing users to quickly view and modify data.

The attendance panel will display a list of students and their attendance records for the selected date range. Teachers will be able to mark attendance for individual students, and the application will automatically calculate the total number of present, absent, and tardy students.

The grades panel will allow teachers to enter and view grades for individual assignments and assessments. The grades panel will also display a summary of student performance, including average grades and overall class performance.

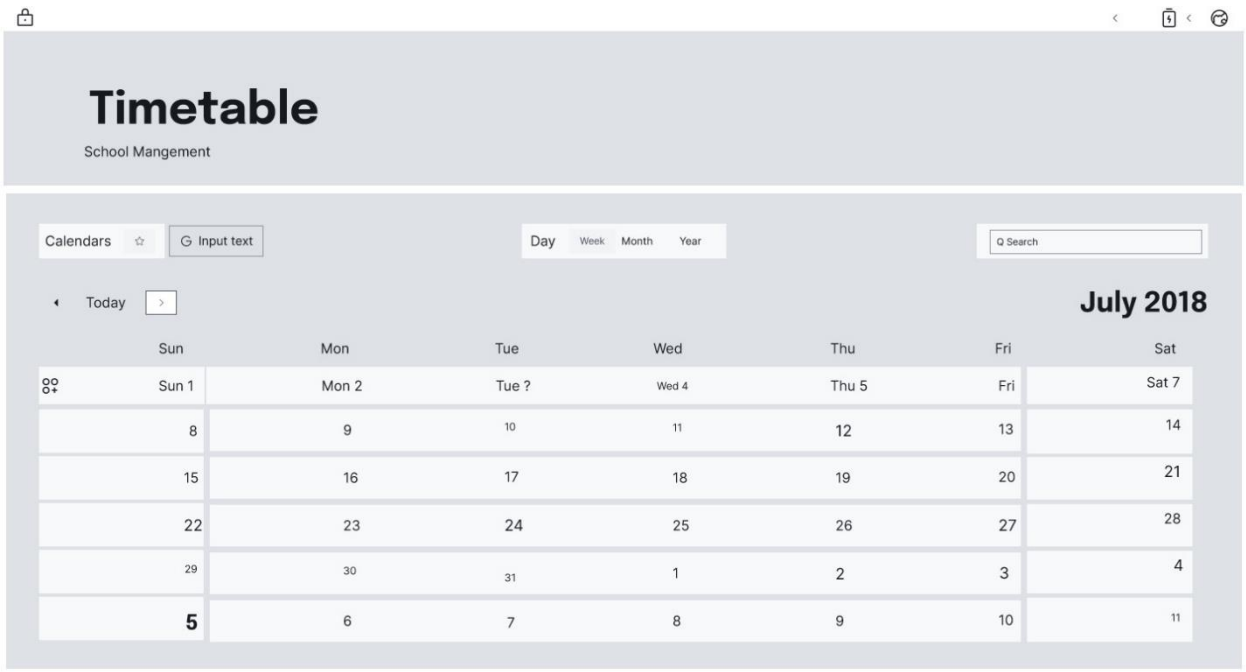
The schedules panel will display the course offerings and schedules for the selected semester. Students will be able to view their own schedules, while teachers and administrators will be able to view schedules for all students in their respective classes.

The student profiles panel will display detailed information about each student, including personal information, contact details, and academic performance. The panel will also allow teachers and administrators to view and modify student data as needed.

The dashboard will also include a search function that will allow users to quickly retrieve data based on various criteria, such as student name, ID number, or course code.

Overall, the dashboard of the School Management System JavaFX will provide a user-friendly interface that will allow teachers, students, and administrators to manage their daily tasks with ease. The dashboard will utilize technologies such as JavaFX and SceneBuilder to create a visually appealing and responsive user interface, making it a powerful tool for school management.

TimeTable



The timetable screen is an essential part of any school management system. It displays the schedule of classes and activities for students and teachers, making it easier for them to keep track of their daily routines. The timetable screen is typically accessed by students, teachers, and administrators.

The project overview for the timetable screen in a school management system using JavaFX involves designing a user-friendly interface that displays the class schedule for the day or week. The timetable screen should allow users to view, add, and edit schedules for each class, teacher, and room.

The timetable screen should have the following features:

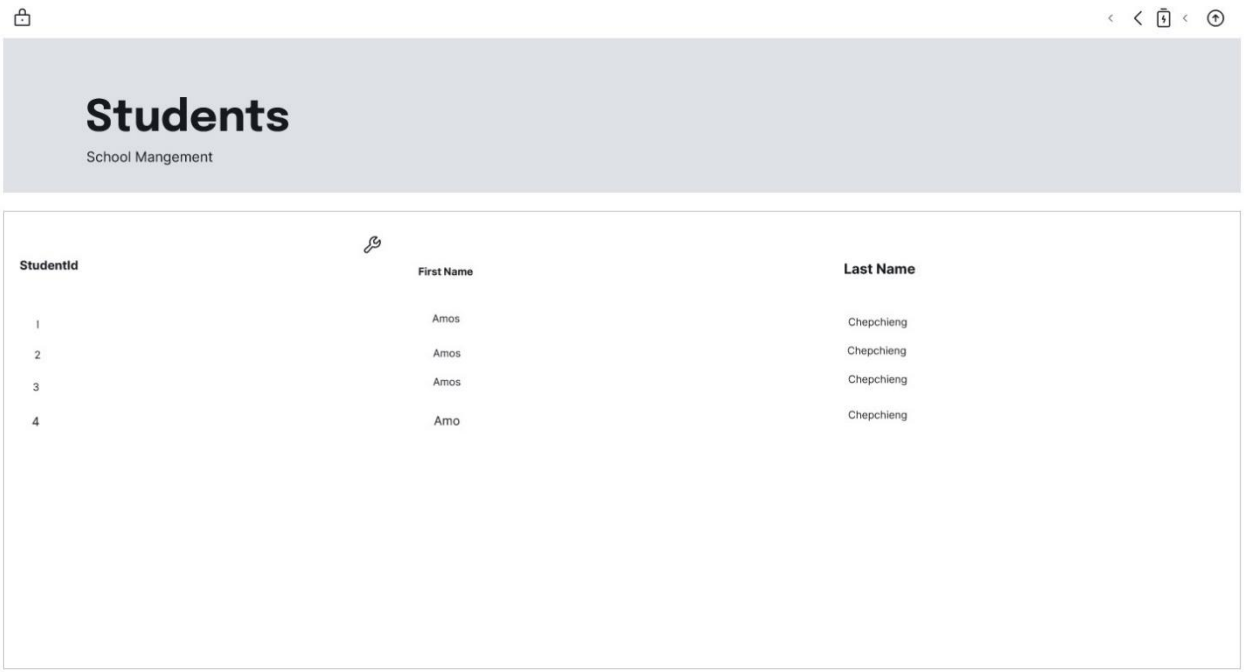
1. Navigation menu - This menu should be located at the top of the screen and allow users to access other areas of the school management system.

2. Date selection - This feature allows users to select the date for which they want to view the timetable.
3. Class schedule display - The timetable screen should display the schedule for each class, including the class name, teacher, and room number.
4. Add schedule - This feature allows users to add new schedules to the timetable. Users should be able to select the class, teacher, and room number for the schedule.
5. Edit schedule - This feature allows users to edit existing schedules in the timetable. Users should be able to select the schedule they want to edit and make changes to the class, teacher, or room number.
6. Delete schedule - This feature allows users to delete schedules from the timetable. Users should be able to select the schedule they want to delete and confirm the action.
7. Print timetable - This feature allows users to print the timetable for the selected date. The printed timetable should display the class name, teacher, and room number for each schedule.

The JavaFX framework provides the necessary tools to create a user-friendly and responsive interface for the timetable screen. The application should be developed using a Model-View-Controller (MVC) architecture to ensure separation of concerns and maintainability.

Overall, the timetable screen in a school management system using JavaFX is an essential tool for students, teachers, and administrators to manage their daily routines effectively.

Students



The student screen is an essential part of any school management system. It allows students to access their personal information, academic records, and other school-related information. The student screen is typically accessed by students after they log in to the system using their credentials.

The project overview for the student screen in a school management system using JavaFX involves designing a user-friendly interface that displays the student's personal information, academic records, attendance records, and other relevant information.

The student screen should have the following features:

1. Navigation menu - This menu should be located at the top of the screen and allow students to access other areas of the school management system.
2. Personal information - This feature allows students to view their personal information, such as their name, address, phone number, email address, and date of birth.

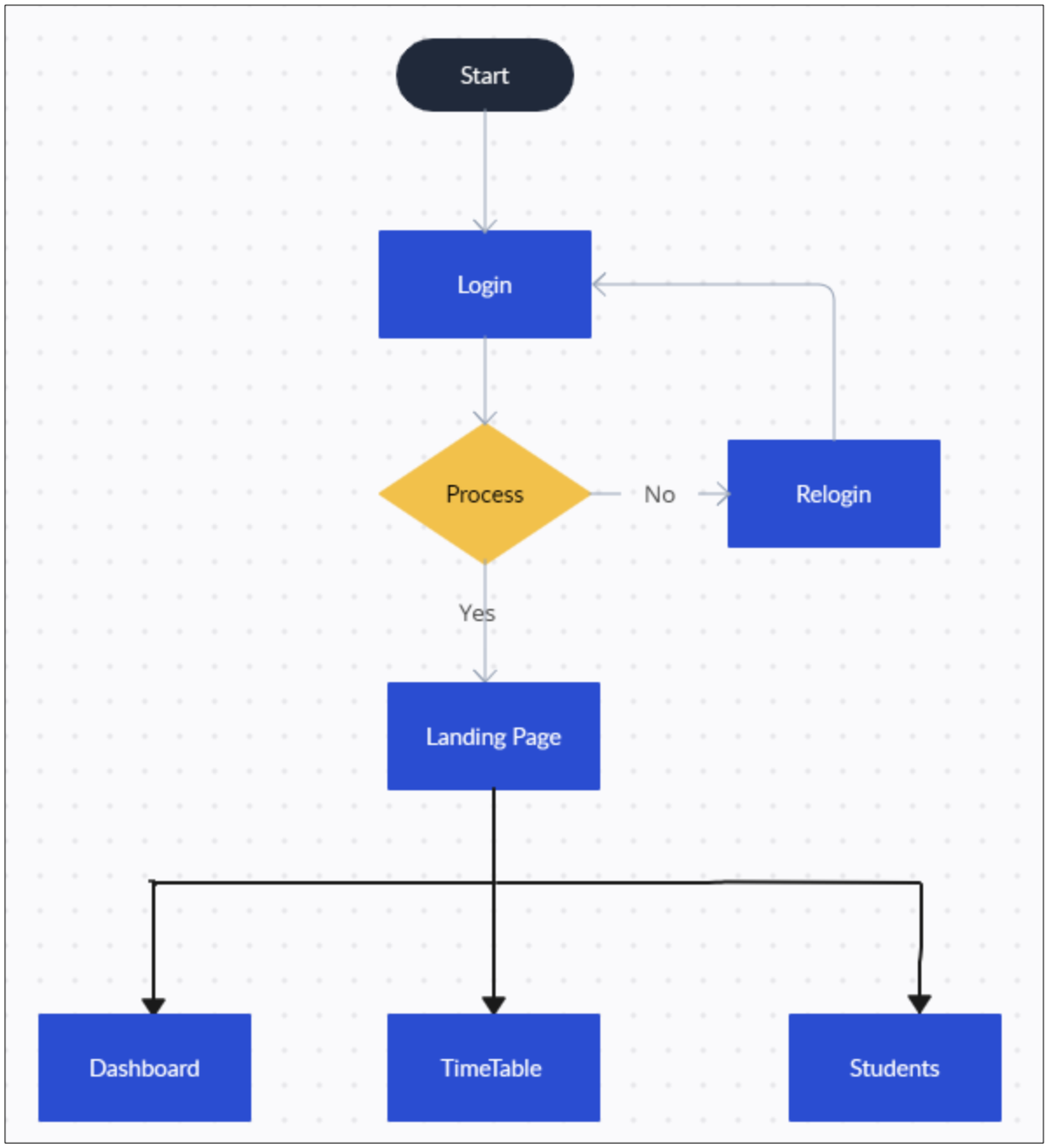
3. Academic records - This feature allows students to view their academic records, including their grades, GPA, and class rank.

The JavaFX framework provides the necessary tools to create a user-friendly and responsive interface for the student screen. The application should be developed using a Model-View-Controller (MVC) architecture to ensure separation of concerns and maintainability.

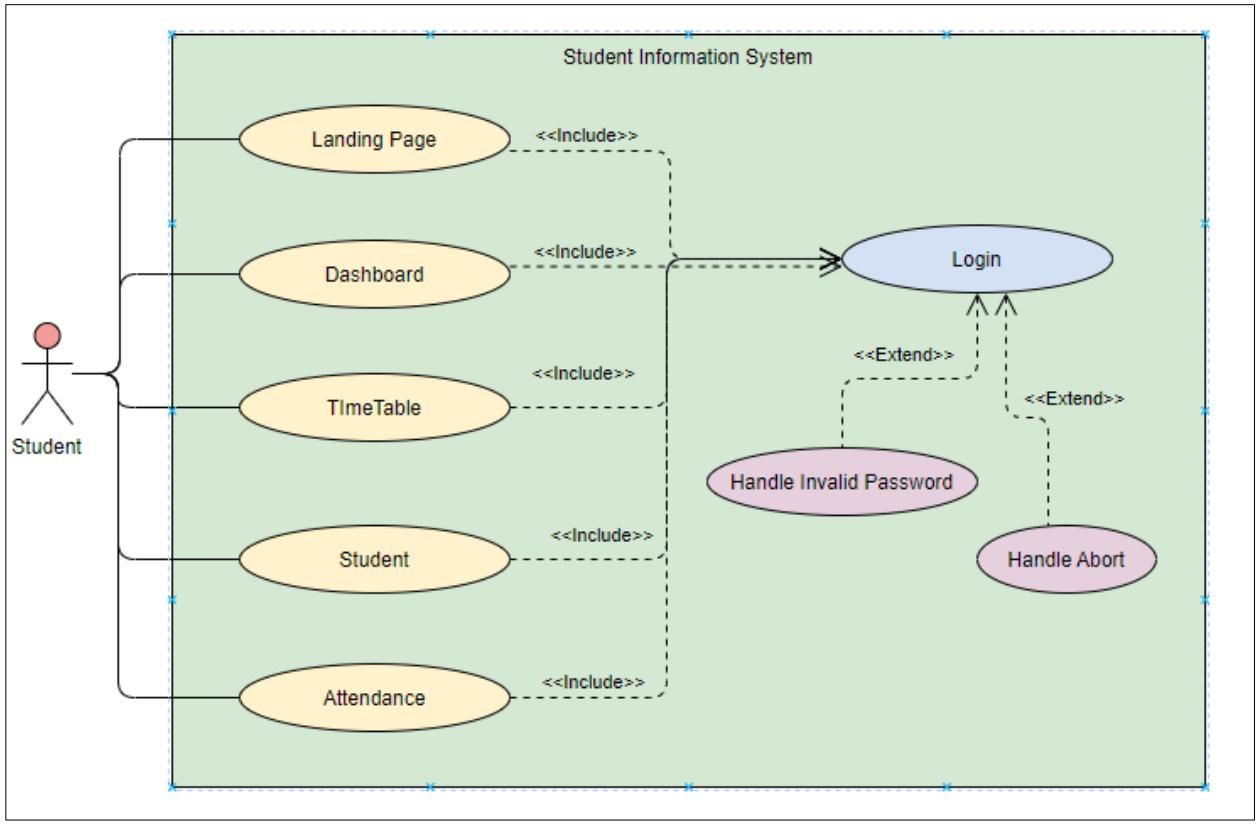
Overall, the student screen in a school management system using JavaFX is an essential tool for students to access their personal information, academic records, attendance records, and other relevant information.

UML class diagram

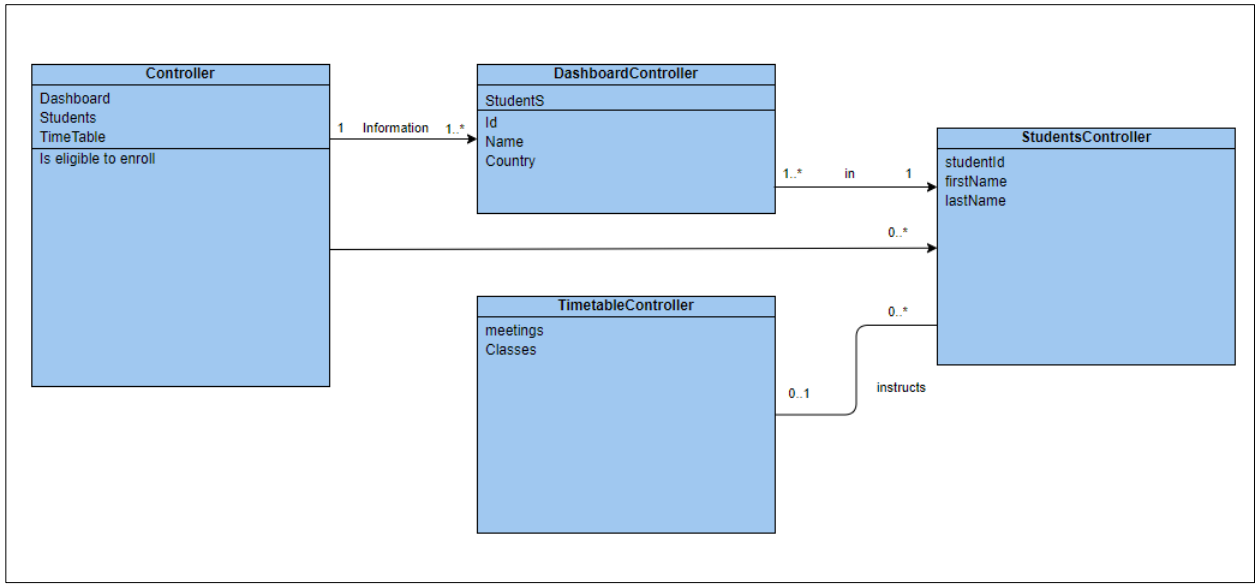
Flow Diagram:



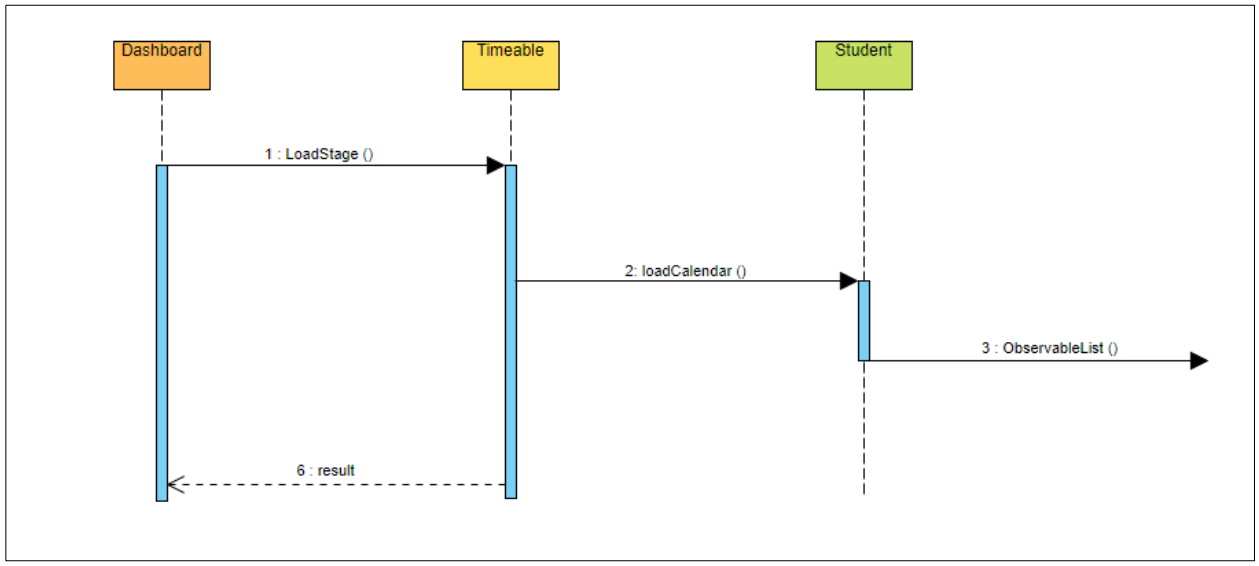
Use Case Diagram



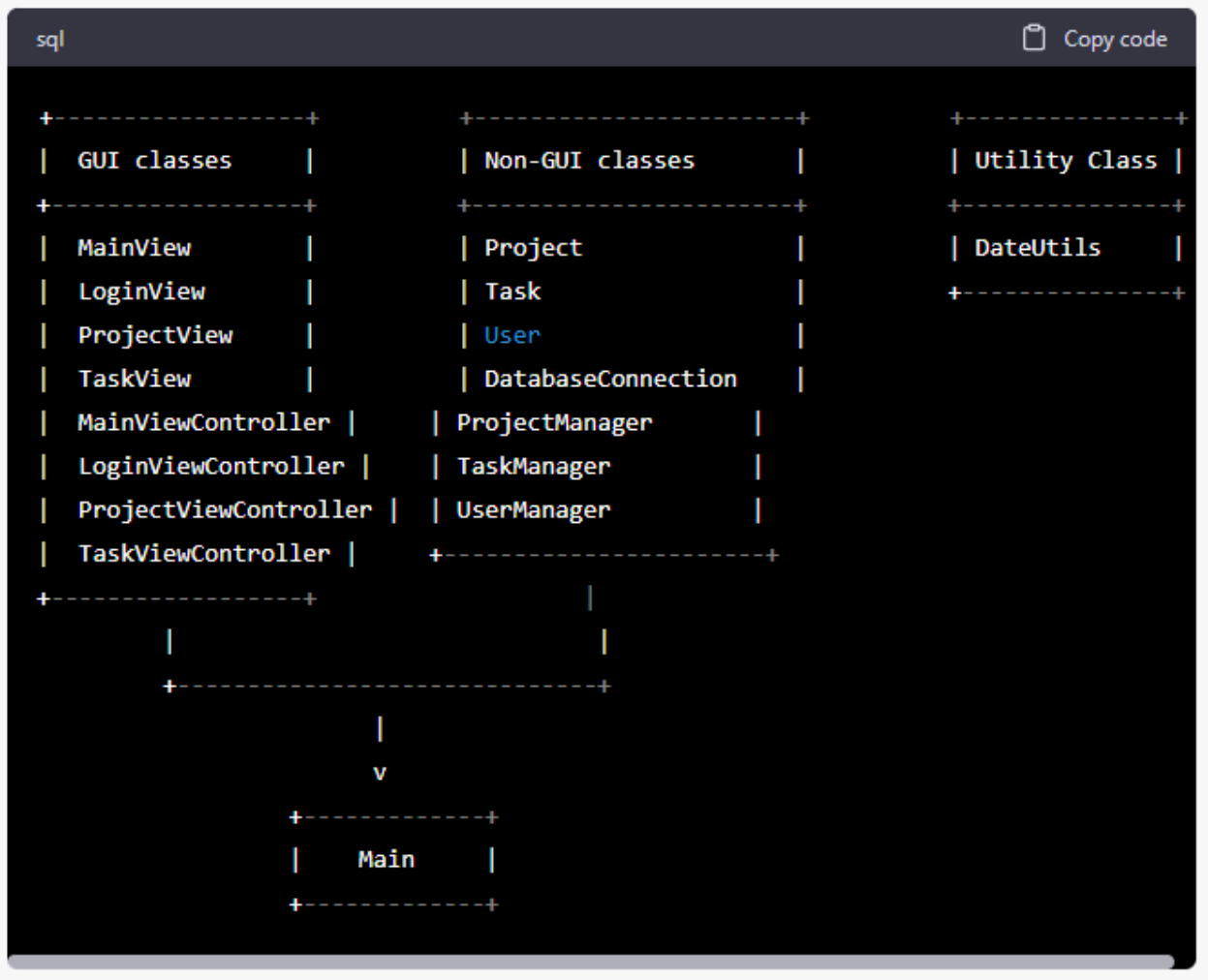
Class Diagram:



Sequence Diagram:



GUI classes and which classes are non-GUI



In this updated diagram, we can see that the GUI classes are MainView, LoginView, ProjectView, TaskView, MainViewController, LoginViewController, ProjectViewController, and TaskViewController. These classes are responsible for displaying the user interface and handling user input. The controller classes have names ending with "Controller," as requested.

The non-GUI (data management) classes are Project, Task, User, DatabaseConnection, ProjectManager, TaskManager, and UserManager. These classes are responsible for managing the data of the application, such as user accounts, projects, tasks, and connections to the database.

Finally, the utility class is DateUtils, which provides various utility methods for working with dates.

The Main class is not a GUI class, but it is the entry point of the application. It initializes the GUI and non-GUI classes, and handles the overall control flow of the application.

File Information

Based on the information provided, it appears that you are referring to a JavaFX application architecture that follows the Model-View-Controller (MVC) pattern. In this pattern, the GUI layout is separated into a view component (represented by the FXML files), the controller component handles the event handling logic, and the model component handles data management.

Here's a brief description of each component:

1. FXML files for the GUI layout: FXML files are XML-based files that describe the layout and structure of a JavaFX user interface. Each FXML file represents a separate window in the application.

   Files:

   - Dashboard.fxml
   - Home.fxml
   - Students.fxml
   - Timetable.fxml
2. Controller classes for the GUI event handlers: Controller classes are Java classes that handle events triggered by user interactions with the GUI. They are responsible for managing the application logic and communicating with the model component to perform any necessary updates.

   Files:

   - Controller.java
   - DashboardController.java
   - StudentsController.java
   - TimetableController.java
3. Data management classes: These classes represent the model component in the MVC pattern. They are responsible for managing the data used by the application and communicating with the controller component to perform any necessary updates to the view.

4. Model: The model represents the data and business logic of the application. It can include classes that represent the data entities, such as customer or order, as well as classes that implement business logic, such as validation or calculation.

   Files:

   - StudentsModel.java
5. View: The view is responsible for rendering the user interface to the user. It is typically implemented using FXML files in a JavaFX application, but can also be implemented using other technologies, such as HTML in a web application.
6. Benefits of the MVC pattern: The MVC pattern provides several benefits, such as separation of concerns, maintainability, and testability. By separating the user interface, data management, and business logic into separate components, each component can be developed, tested, and maintained independently, which can help to reduce complexity and improve code quality.
7. Drawbacks of the MVC pattern: While the MVC pattern can be a useful architecture for developing applications, it is not without its drawbacks. One common issue with the pattern is that it can result in a large number of classes, which can make the codebase difficult to navigate and understand. Additionally, the pattern can be overkill for small or simple applications, which may not require the level of abstraction provided by the MVC pattern.