

Sprint 3 – GUI Development Report

Student Name: Phone Pyae Kyaw

Student ID: A00091367

1. Introduction

Sprint 3 focuses on developing a Graphical User Interface (GUI) for the Booking System using JavaFX. This sprint enhances usability by allowing users to interact visually with the system.

2. System Overview

The Booking System manages Facilities, Equipment, and Hobbies. Users can view availability, book items, and release them through the GUI.

3. GUI Design

The GUI is built using JavaFX components such as ListView, Buttons, and Labels. It displays real-time booking status and system messages.

4. User Stories

- View all bookable items
- Book an available item
- Release a booked item
- View booking status
- Receive feedback messages

5. Class Design

The system follows object-oriented principles. BookableItem is an abstract class extended by Facility, Equipment, and Hobby. BookingSystem handles business logic while BookingSystemGUI manages the interface.

6. Implementation Details

JavaFX event handlers are implemented using lambda expressions. ObservableList is used to ensure automatic updates in the ListView when booking states change.

7. Testing & Results

The system was tested for booking, releasing, UI refresh, and error handling. All functionality worked correctly as shown in the screenshots below.

Screenshots

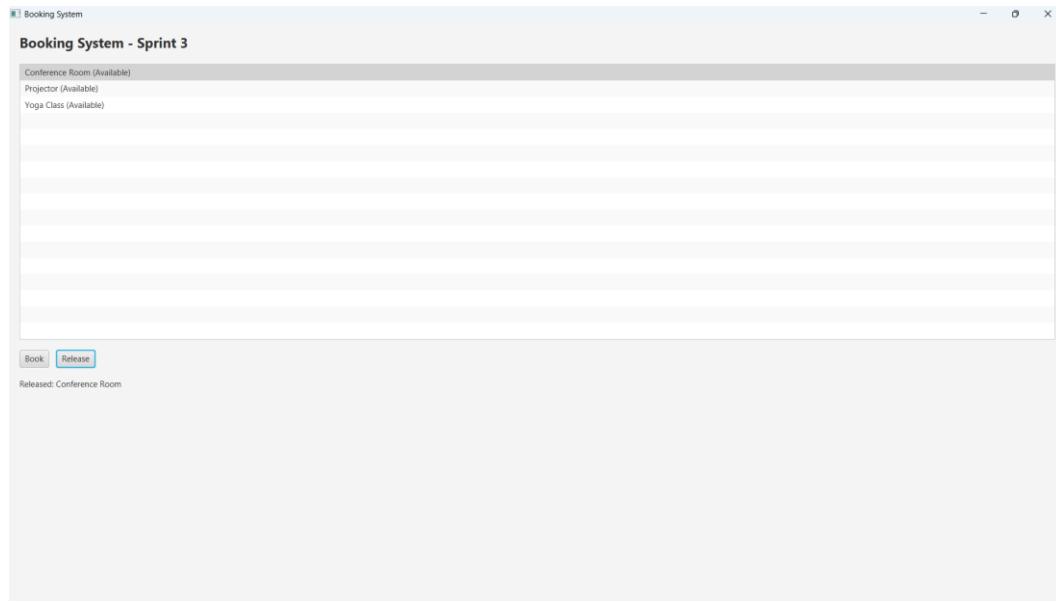


Figure: Screenshot 2026-01-14 030116.png

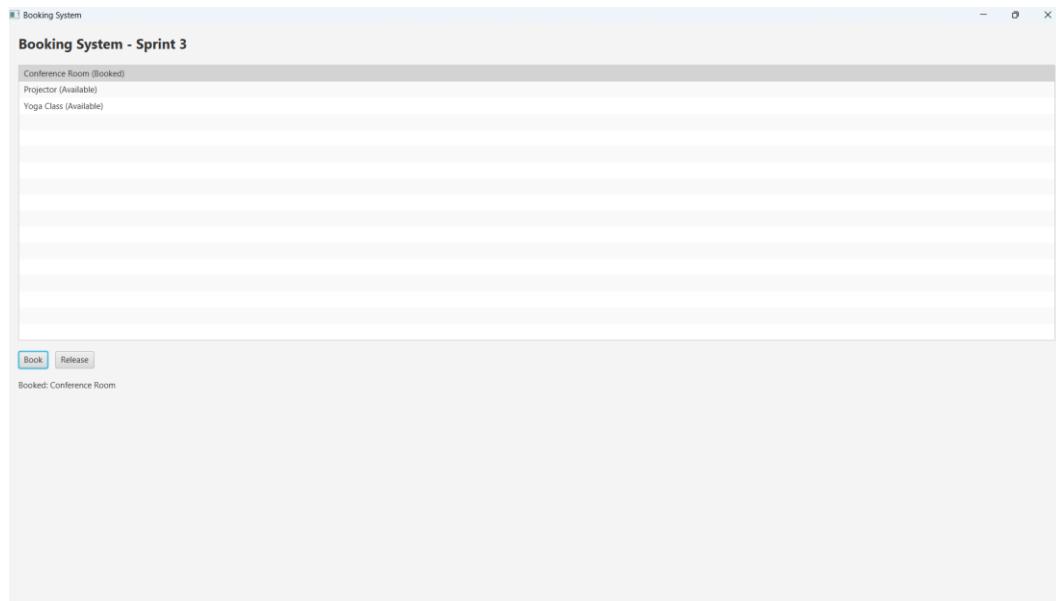


Figure: Screenshot 2026-01-14 030122.png

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. The main area displays the `BookingSystemGUI.java` file. The code is as follows:

```
public class BookingSystemGUI extends Application {
    public void start(Stage stage) {
        bookBtn.setOnAction(e -> {
            BookableItem selected = listView.getSelectionModel().getSelectedItem();
            if (selected == null) {
                status.setText("Select an item first");
                return;
            }
            if (selected.book(currentUser)) {
                status.setText("Booked: " + selected.getName());
            } else {
                status.setText("Already booked by " + selected.getBookedBy().getName());
            }
            listView.refresh();
        });
        releaseBtn.setOnAction(e -> {
            BookableItem selected = listView.getSelectionModel().getSelectedItem();
            if (selected == null) {
                status.setText("Select an item first");
            }
        });
    }
}
```

In the bottom left corner of the code editor, there is a red circular icon with a '1' inside, indicating one error. The bottom status bar shows the file path `IdeaProjects > Sprint3 > src > BookingSystemGUI` and encoding information `UTF-8 4 spaces`.

Figure: Screenshot 2026-01-14 030207.png

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. The main area displays the `Main.java` file. The code is as follows:

```
public class Main {
    public static void main(String[] args) {
        BookingSystemGUI.main(args);
    }
}
```

In the bottom left corner of the code editor, there is a red circular icon with a '1' inside, indicating one error. The bottom status bar shows the file path `IdeaProjects > Sprint3 > src > Main > main` and encoding information `UTF-8 4 spaces`.

Figure: Screenshot 2026-01-14 030520.png

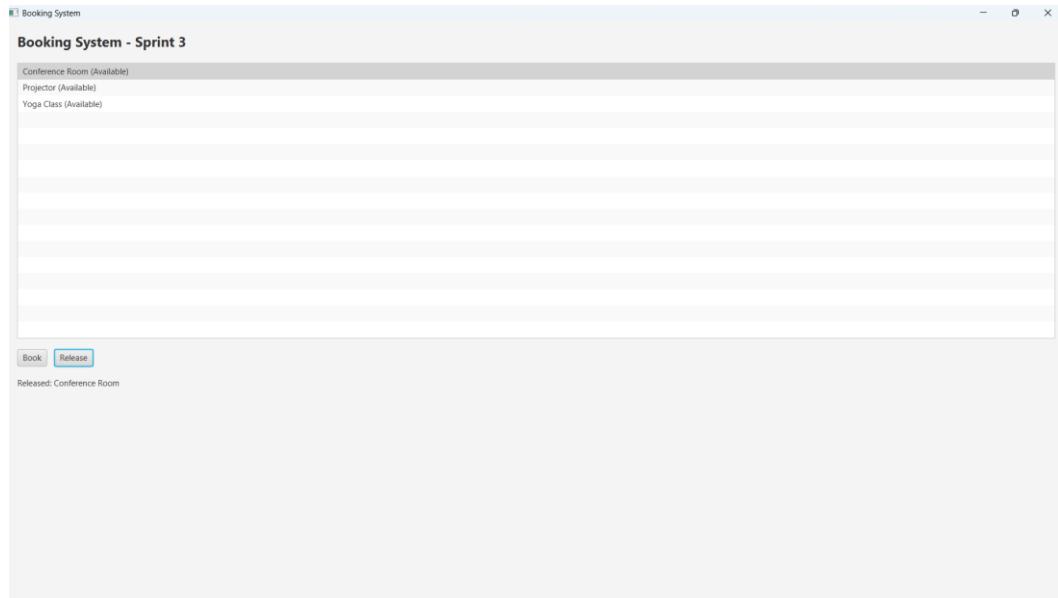


Figure: Screenshot 2026-01-14 030942.png

A screenshot of the IntelliJ IDEA IDE. The left side shows the project structure with a tree view containing "IdeaProjects", "Sprint2 Booking System", "Sprint3", and "src" folder. Inside "src", there are packages for "BookableItem", "BookingSystem", "Hobby", "Equipment", "Facility", and "User". The right side is a code editor window displaying the "BookingSystemGUI.java" file. The code is a Java class that extends Application. It imports various JavaFX classes and defines a main method that starts the application, initializes a BookingSystem, adds a user, and adds several Facility, Equipment, and Hobby objects to the system. The code editor shows syntax highlighting and some annotations like @Override and @FXML.

Figure: Screenshot 2026-01-14 031107.png

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure under "IdeaProjects". The "src" folder contains "BookingSystemGUI.java", "BookingSystem.java", "Hobby.java", "User.java", "BookableItem.java", "Facility.java", and "Equipment.java".
- Code Editor:** The main window displays the content of `BookingSystemGUI.java`. The code is a JavaFX application for a booking system. It includes imports for `javafx.application.Application`, `javafx.scene.Scene`, `javafx.scene.control.Button`, `javafx.scene.control.ListView`, `javafx.scene.control.Label`, `javafx.scene.layout.VBox`, `javafx.stage.Stage`, and `java.util.List`.
- Code Content:** The code defines a `BookingSystemGUI` class that extends `Application`. It sets up a stage, creates a title label, a list view for items, and two buttons for booking and releasing items. It handles the selection of items from the list view and the corresponding actions on the buttons.

```
public class BookingSystemGUI extends Application {
    public void start(Stage stage) {
        Label title = new Label("Booking System - Sprint 3");
        title.setStyle("-fx-font-size: 20px; -fx-font-weight: bold");

        ListView<BookableItem> listView = new ListView<>(items);

        Button bookBtn = new Button("Book");
        Button releaseBtn = new Button("Release");

        Label status = new Label("Ready");

        bookBtn.setOnAction(e -> {
            BookableItem selected = listView.getSelectionModel().getSelectedItem();
            if (selected == null) {
                status.setText("Select an item first");
                return;
            }
            if (selected.book(currentUser)) {
                status.setText("Booked: " + selected.getName());
            } else {
                status.setText("Already booked by " + selected.getBookedBy().getName());
            }
            listView.refresh();
        });

        releaseBtn.setOnAction(e -> {
            BookableItem selected = listView.getSelectionModel().getSelectedItem();
            if (selected == null) {
                status.setText("Select an item first");
                return;
            }
        });
    }
}
```

Figure: Screenshot 2026-01-14 031122.png

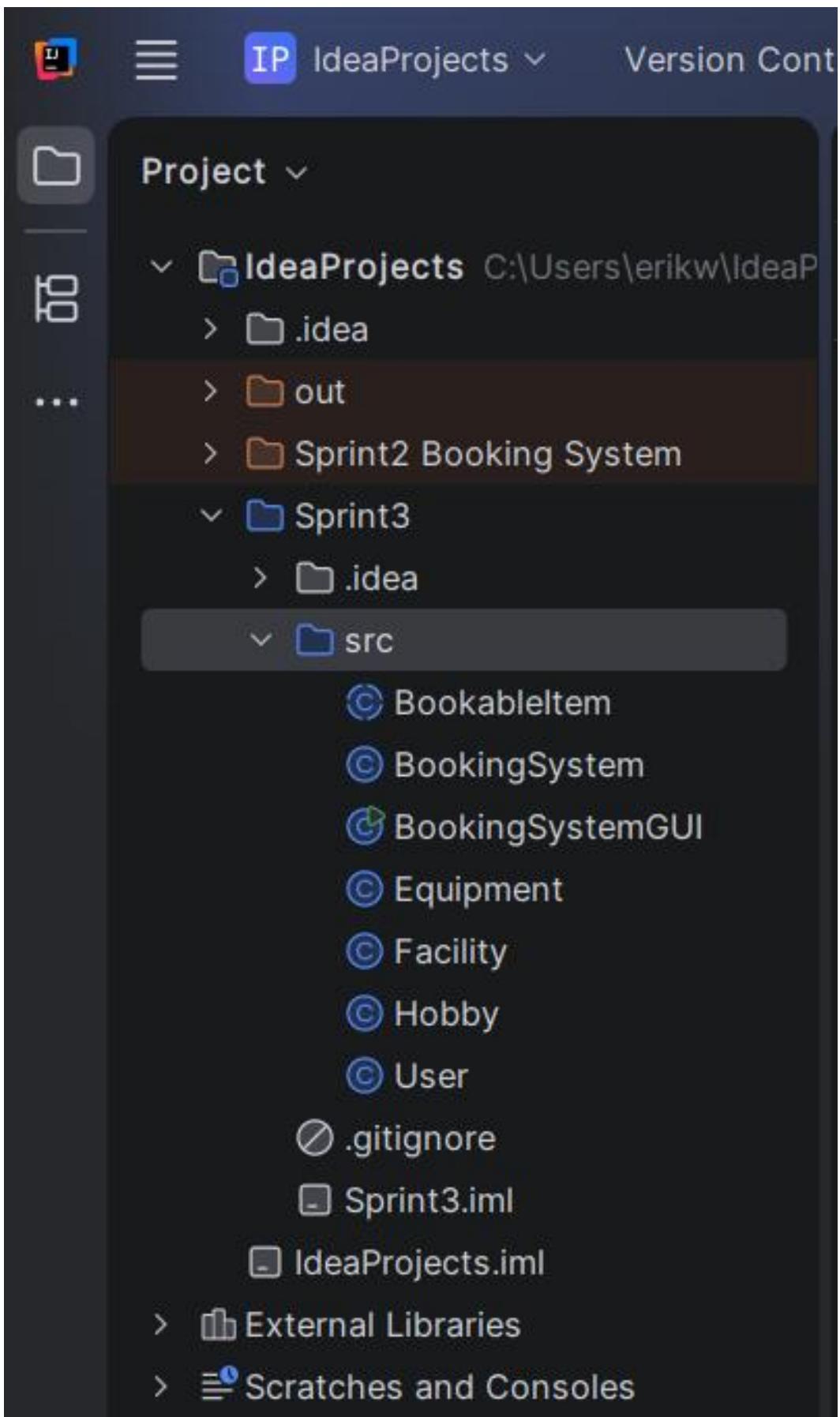


Figure: Screenshot 2026-01-14 031134.png

8. Conclusion

Sprint 3 was completed successfully with a fully functional JavaFX GUI. The system meets all requirements and is ready for submission.