SWE206

Readme File

Farah Hammad 202257640

Shatha Alharbi 202283660

Leen Ghazi 202223660

Renad Adel 202276760

Contents

1 Description

The KFUPM Registration System is a software application developed as part of the SWE-206: Introduction to Software Engineering course project. It enables students, staff, and faculty at KFUPM to create, join, and view reservations. The system aims to streamline the reservation process and provide a user-friendly interface for efficient management of activities.

2 Features

1. **Edit reservation:** The edit reservation feature allows users to easily modify their existing reservations. Users can update details such as date, time, and location, ensuring flexibility and adaptability. This feature streamlines the reservation management process, providing convenience and enhancing user satisfaction.
2. **Incredible user interface:** Our application features an incredible user interface designed to deliver an exceptional user experience. With its visually appealing and modern design elements, the interface is both aesthetically pleasing and easy to navigate. The layout is thoughtfully organized, ensuring that key features and functionalities are readily accessible. It is also designed to be responsive and adaptive, providing a seamless experience across different devices and screen sizes. By prioritizing usability, aesthetics, and responsiveness, our incredible user interface enhances user satisfaction and promotes a delightful interaction with our application.
3. **Scene Switching User Interface:** Our application boasts an intuitive user interface that seamlessly facilitates scene switching. This dynamic interface enables the application to adapt to user interactions and specific functionalities, providing clear and logical navigation between different

sections. The scene switching feature ensures a smooth and polished user experience.

4. **Account Deletion Confirmation**: We have introduced a confirmation prompt to the "Delete" button functionality, prioritizing user safety and preventing accidental deletion of user profiles. This additional step ensures that users are prompted to confirm their intention to delete their account before proceeding, offering a user-centric approach to account management.
5. **Email notification**: Our application includes an email notification feature that automatically sends emails to users who have made reservations in the event that the department cancels their reservation. This functionality ensures efficient and timely communication between the department and the users, keeping them informed about any changes or cancellations. By promptly notifying users via email, we aim to minimize inconvenience and provide transparent communication, allowing users to make alternative arrangements if necessary. This email notification feature enhances the overall user experience and facilitates effective reservation management.

3 Design Overview

3.1 Common Class

Purpose

This class serves as the stage and scene for this project.

Variables

- Stage (stage): used in the start method and switch scenes.
- Scene (scene): used in the start method and switch scenes.

3.2 Controller_CreateReservation Class

Purpose

The Controller_CreateReservation class is a controller for the creation of reservations in the application. It handles user input, validation, and reservation creation based on the selected options.

Variables

- locationsBox: A ChoiceBox for selecting the location of the reservation.
- attendeeText: A Label for displaying information about the number of attendees.
- datePicker: A DatePicker for selecting the date of the reservation.
- reasonField: A TextField for entering the reason for the reservation.
- startingTimesList: An ObservableList containing starting times for reservations.
- durationList: An array of integers representing possible durations for reservations.
- durationChoiceBox: A ChoiceBox for selecting the duration of the reservation.
- startingTimeChoiceBox: A ChoiceBox for selecting the starting time of the reservation.
- attendeeNumberBox: A ChoiceBox for selecting the number of attendees for the reservation.
- confirm_button: A Button for confirming the reservation.
- cancel_button: A Button for canceling the reservation.
- labs_classrooms_radioButton: A RadioButton for selecting lab or classroom reservation.
- facilities_radioButton: A RadioButton for selecting facilities reservation.
- timeText: A Text object for displaying reservation time information.
- noteText: A Text object for displaying notes or error messages.

- currentUserName: A Text object for displaying the current user's name.
- currentUserPosition: A Text object for displaying the current user's position.

Methods

- switchTo_home_page_scene(ActionEvent event)

  This method switches the scene to the home page when the cancel button is clicked or confirms the reservation when the confirm button is clicked. It handles input validation and reservation creation.

- setLabs_classrooms_radioButton(ActionEvent event)

  This method sets up the UI for lab or classroom reservation based on user eligibility and selection.

- setfacilities_radioButton (ActionEvent event)

  This method sets up the UI for facilities reservation based on user eligibility and selection.

- initialize()

  This method initializes the UI components and sets default values for various fields.

- isValidEndTime(int duration, String inputStartingTime)

  This method checks if the end time of the reservation is valid based on the selected duration and starting time.

- returnReservedHours(int starting, int ending)

This method returns a list of reserved hours based on the starting and ending times of the reservation.

3.3 Controller_Department Class

Purpose

This class serves as the controller for the department page in a reservation management system.

Variables

FXML Fields:

- reservationDataAnchorPane : AnchorPane to display reservation
- tableView : TableView to displayreservations.
- `bookingID`, `date`, `gender`, `location`, `reason`, `timeFormat`: TableColumns for various reservation properties.
- cancel_button: Button for canceling a reservation.
- bookingID_choiceBox : ChoiceBox for selecting reservation IDs.

Others:

- reservationObservableList: ObservableList to hold reservations.

Methods

- initialize(URL url, ResourceBundle resourceBundle):

  Initializes the controller.

  Sets up TableView, ChoiceBox, and UI colors.

  Populates TableView and ChoiceBox with data from reservations.

- switchTo_singIn(ActionEvent event):

  Switches the scene to the sign-in page when called.

Loads the FXML file for sign-in and sets it as the scene.

- getReservationObservableList():

  Retrieves reservations for the department.

  Adds department reservations to the reservationObservableList.

- cancelReservation(ActionEvent event):

  Handles the event of canceling a reservation.

  Retrieves the selected booking ID from the choice box.

  Searches for the corresponding reservation in the reservationObservableList.

  Removes the reservation from the list and updates the UI accordingly.

  Removes the booking ID from the choice box and clears the selection.

  Removes the reservation from the appropriate person's reservation list.

  Sends a cancellation email to the person associated with the reservation.

3.4 Controller_EditReservation Class

3.5 Controller_HomePage Class

Purpose

This class serves as the controller for the home page in a reservation management system.

Variables

- AnchorPane (homePageAnchorPane): for the Home page.
- Button (create_button): for creating reservations.
- Button (edit_delete_button): for editing or deleting reservations.
- Button (view_button): for viewing reservations.
- Button (join_button): for joining reservations.
- Button (log_out_button): for logging out.

- Text (currentUserName): for displaying the current user's name.
- Text currentUserPosition: for displaying the current user's position.

Methods

- initialize():

  Sets initial configurations for currentUserName, currentUserPosition, homePageAnchorPane, create_button, edit_delete_button.

- switch_scenes(ActionEvent event):

  Handles scene transitions based on the button selected by the user.

3.6 Controller_JoinReservation Class

Purpose

This class serves as the controller for joining reservations for events in a reservation management system.

Variables

FXML Fields:

- tableView: TableView to display reservations.
- attendeeNumber, bookingId, date, location, timeFormat: TableColumns for various reservation properties.
- messageLable: Label to display messages to the user.
- boockingID_choiceBox: ChoiceBox for selecting reservation IDs.
- join_button, back_button: Buttons for joining events and navigating back.

Others:

- reservationObservableList: ObservableList to hold reservations.
- currentUser: Current user of the system.

Methods

- initialize(URL url, ResourceBundle resourceBundle):

  Initializes the controller.

  Sets up TableView and ChoiceBox.

  Populates them with data from reservations.

- switchTo_home_page_scene(ActionEvent event):

  Switches the scene to the home page when called.

  Loads the FXML file for the home page and sets it as the scene.

- getReservationObservableList():

  Retrieves valid reservations for the current user.

  Filters reservations based on conflicts and maximum attendee number.

- joinEvent(ActionEvent event):

  Handles the event of joining a reservation.

  Checks if the user is already assigned to the event.

  Attempts to join the event and updates the message label accordingly.

3.7 Controller_ModifyReservation Class

3.8 Controller_SignIn Class

Purpose

This class handles user sign-in functionality in a reservation management system.

Variables

FXML Fields:

- UserNameTextField: TextField for entering the username.

- passwordTextField: PasswordField for entering the password.
- invalidLable: Label to display invalid username or password message.

Methods:

- switchTo_singUp(ActionEvent event):

  Switches the scene to the sign-up page when called.

  Loads the FXML file for sign-up and sets it as the scene.

- switchTo_Home_page(ActionEvent event):

  Handles the event of switching to the home page.

  Retrieves username and password from the text fields.

  Validates the credentials against the list of users.

  If credentials match, sets the current user and navigates to the appropriate page (either the department page or home page).

  Displays an error message for incorrect username or password.

  Displays a message for missing information if fields are empty.

3.9 Controller_SignUp Class

Purpose

This class manages the sign-up process for new users in a reservation management system.

Variables

FXML Fields:

- passwordTextField2: PasswordField for entering the password.
- confirmTextField2: PasswordField for confirming the password.
- invalidLable: Label to display invalid information messages.
- nameTextField2: TextField for entering the username.
- emailTextField2: TextField for entering the email.

- genderChoiceBox: ChoiceBox for selecting gender.
- positionChoiceBox: ChoiceBox for selecting position.
- register: Button to register new users.

Others:

- gender: Array of gender options.
- position: Array of position options.

Methods

- initialize():

  Initializes the choice boxes with gender and position options.

- registerButtonOnAction(ActionEvent event):

  Handles the event of registering a new user.

  Retrieves user information from the text fields and choice boxes.

  Validates the information, such as checking for empty fields, matching passwords, and unique username/email.

  Creates a new user object and adds it to the list of users.

  Navigates to the appropriate page based on the user's position.

- switchTo_singIn(ActionEvent event):

  Switches the scene to the sign-in page when called.

  Loads the FXML file for sign-in and sets it as the scene.

3.10 Controller_ViewReservation Class

Purpose

This class serves as the controller for the view reservation page in a reservation management system.

Variables

- Anchore (viewReservationAnchor): for the View Reservation page.
- TableView (reservationData): for displaying reservation information.
- TableColumn (bookingID, location, date, timeFormat, reason, attendeeNumber): for displaying specific some reservation attributes.
- Button (back_button): for returning to the home page.

Methods

- initialize(URL url, ResourceBundle resourceBundle):

  Initializes the TableView with TableColumn of some reservation attributes.

- switchTo_home_page_scene(ActionEvent event):

  Handles scene transition to the home page.

3.11 HelloApplication Class

3.12 Person Class

Purpose

This class serves as the Person object that represents each user for the department page in a reservation management system.

Variables

- String (contactEmail): Contact email of the person.
- String (gender): Gender of the person.

- ArrayList<Reservation> (reservationList): List of reservations associated with the person.
- String (username): Username of the person.
- String (password): Password of the person.
- String (position): Position of the person (Student, staff…).

Methods:

Getters:

- getPosition(): Returns the position of the person.
- getGender(): Returns the gender of the person.
- getContactEmail(): Returns the contact email of the person.
- getUserName(): Returns the username of the person.
- getPassword(): Returns the password of the person.
-
- joinASportEvent(int choosedID):

  This method is to verify if the user joined an event successfully by traversing the reservation list and add it to the user's reservation list.

- createEvent(LocalDate date, ArrayList<Integer> time, String location, String gender, String reason, int maximumAttendeeNumber):

  Creates a new event reservation and add it to the system reservation list and the user reservation list after checking if a conflict occurs with his/her schedule as well as the system list.

- cancelReservation(Reservation reservation) – Not Used:

  This method supposed to cancel a reservation from the user by choosing the ID and cancel it.

- ConfirmReservation(Reservation reservation) – Not Used:

  Confirms a reservation.

- reserveLabOrCLassroom(LocalDate date, ArrayList<Integer> time, String location, String gender, String reason):

  This method is Similar to CreateEvent, it reserves a lab of classroom after confirming there is no conflict.

- setCurrentUser(Person person): Sets the current user of the system.
- getCurrentUser(): Gets the current user of the system.

3.13 Reservation Class

Purpose

The Reservation class represents a reservation made in the system. It encapsulates details such as booking ID, location, date, time, attendee number, gender, reason, and reservation type.

Constructors

Labs/Classrooms Reservation Constructor:

- public Reservation(LocalDate date, ArrayList<Integer> time, String location, String reason)

  Initializes a reservation for labs/classrooms.

  Does not include gender or maximum attendee number.

Facilities Reservation Constructor:

- public Reservation(LocalDate date, ArrayList<Integer> time, String location, String gender, int maximumAttendeeNumber, String reason)

Initializes a reservation for facilities.

Includes gender and maximum attendee number.

Getters and Setters

Getters and setters are provided for each attribute of the reservation, such as booking ID, location, date, time, attendee number, gender, reason, and time format.

Methods

- equalReservation(Object that)

  Checks if two reservations have the same booking ID.

- checkConflictingTime(ArrayList<Integer> otherArrayList)

  Checks if the reservation conflicts with another reservation in the system based on time.

- checkSystemConflict(Reservation systemReservation)

  Checks if the reservation conflicts with a system-wide reservation based on date, location, and time.

- checkTimeAndDateConflict(Reservation currentUserReservationList)

  Checks if the reservation conflicts with another reservation made by the current user based on date and time.

- checkConflictingTimeForSameUser(ArrayList<Integer> otherArrayList)

  Checks if the reservation conflicts with another reservation made by the same user based on time.

3.14 RunApplication Class

Purpose

This class is responsible for running the registration system, loading, and saving all the data of the user.

Variables

FXML Fields:

- people : An ArrayList of Person objects to store user information.
- reservations : An ArayList of Reservation objects to store Reservation information.

Methods

- main(Sttring[] args):

  Entry point of the application.

  Launches the JavaFX application by calling the launch method.

- start(Stage stage):

  Overridden from the Application class.

  Sets up the initial scene and the application's main scene.

- loadData():

  Reads data from the "people.txt" file.

  Populates the people and reservations lists.

- saveData():

  Saves the data from the people and reservations lists to the "people.txt" file.

  Uses a BufferedWriter to write the data in a specific format, separating each attribute with the "=" delimiter.

- sendEmail(String recipient, String subject, String body):

  Allows sending an email using the default email client on the system.

  Constructs a mail to URI with recipient email address, subject, and body to open the email client with pre-filled information.

- stop():

  Overridden from the Application class.

  Called when the application is about to exit.

  Invokes the saveData method to save the data before exiting.

4 Technologies Used

- intelliJ IDEA
- SceneBuilder