

|  |  |
| --- | --- |
| Name: | Hiba Noor |
| Roll No.: | SP-24-BSE-044 |
| Submitted to: | Sir Shahid Bhatti |

This JavaFX program creates a graphical user interface (GUI) for a data entry system, specifically for a registration form. The structure of the program can be broken down into two primary stages: the welcome screen and the main form screen. Here's a more detailed breakdown of how the program works:

**1. Welcome Window:**

* Upon running the application, a Stage (window) named welcomeStage is created.
* The window contains a VBox layout with a welcome message and a button ("Start Registration").
* When the user clicks the "Start Registration" button, the welcome window closes, and the main form is opened by calling the openMainForm() method.

**2. Main Form (Data Entry Screen):**

The main form is made up of several key sections:

* **Top Banner:**
  + A HBox layout is used to place a logo image (bannerImageView) and a title label (bannerLabel) horizontally in the top section of the window.
  + The image is loaded from a specified file path, and the text label is styled with a dark blue font.
* **Form Fields (Left Section):**
  + The form fields are arranged using a GridPane, which helps organize the form inputs in rows and columns.
  + The fields include:
    - **Name, Father’s Name, CNIC, Date of Birth:** Standard text input fields (TextField and DatePicker).
    - **Gender:** A pair of radio buttons ("Male" and "Female"), grouped together using a ToggleGroup.
    - **City:** A combo box (ComboBox) with predefined city options.
* **Image Upload (Right Section):**
  + A VBox layout is used for displaying the image upload section.
  + A FileChooser is provided, allowing the user to select an image file (such as .jpg, .png) from their system.
  + The chosen image is previewed using an ImageView, which resizes the image to fit within the specified dimensions.
* **Save and Exit Buttons (Bottom Section):**
  + Two buttons are provided at the bottom of the form:
    - **Save Button:** When clicked, it checks if all the required fields are filled. If they are, the entered data is stored in an ArrayList<Person> and a success message is shown. If any fields are missing, an error alert is displayed.
    - **Exit Button:** Closes the form and exits the application.

The saveButton is responsible for validating user input. If any field is empty, the program displays an error alert asking the user to fill all fields. If all fields are completed, the entered data is collected into a new Person object and added to the personList. A success message is then displayed.

**3. Event Handling and User Interaction:**

* **Button Event Handling:** Each button (Start Registration, Save, Exit) is linked to an event handler (using setOnAction) that defines the action taken when the user clicks the button.
* **Image Selection:** The FileChooser allows users to select an image file, and the selected image is displayed in the image preview section.

**4. Scene Setup:**

* The entire layout (root node) is a BorderPane, with the main content area split into top, left, right, and bottom sections.
* The scene is created with this layout, and it is displayed in the Stage, which represents the main application window.

**5. Error and Success Alerts:**

* Alerts are used to provide feedback to the user:
  + **Error Alert:** Displays when the user attempts to save the form with missing fields.
  + **Success Alert:** Shows when the user successfully saves the data (i.e., all fields are filled).

**6. Main Method and Launching the Application:**

* The main method calls launch(args), which is a built-in JavaFX method to start the JavaFX application. This method in turn calls the start() method, which sets up the primary stage and triggers the display of the welcome window.

**Summary of Key JavaFX Components Used:**

* Stage: Represents a window in JavaFX.
* Scene: Holds all the components for a window.
* VBox, HBox, GridPane: Layout containers for organizing UI elements.
* Button, TextField, ComboBox, RadioButton, DatePicker: Various form controls for user input.
* Alert: Used to show success or error messages.
* FileChooser: Allows the user to select a file from their system.

This GUI-based system is a user-friendly form where users can enter and save personal data, along with uploading an image, all within an organized, structured interface.

Here i have shown the outputs:A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated