**GURU NANAK COLLEGE BUDHLADA**



**DEPARTMENT: COMPUTER**

**NAME OF PROJECT: notes App**

Submitted to: Submitted by:

HOD **Chushamdeep Singh(1103)**

**Dr. Rekha Kalra Amritpreet Singh(1145)**

**Munish Kumar(1130)**

Content and Chapters

[1. Introduction 3](#_Toc163213590)

[2. Project Overview 4](#_Toc163213591)

[3. System Architecture 4](#_Toc163213592)

[4. Technology Stack 5](#_Toc163213593)

[5. Implementation Details 5](#_Toc163213594)

[**Login Screen** 5](#_Toc163213595)

[**Main Menu** 5](#_Toc163213596)

[**Note Editing and Saving** 6](#_Toc163213597)

[**About Section** 6](#_Toc163213598)

[6. Code Structure 7](#_Toc163213599)

[7. References 29](#_Toc163213600)

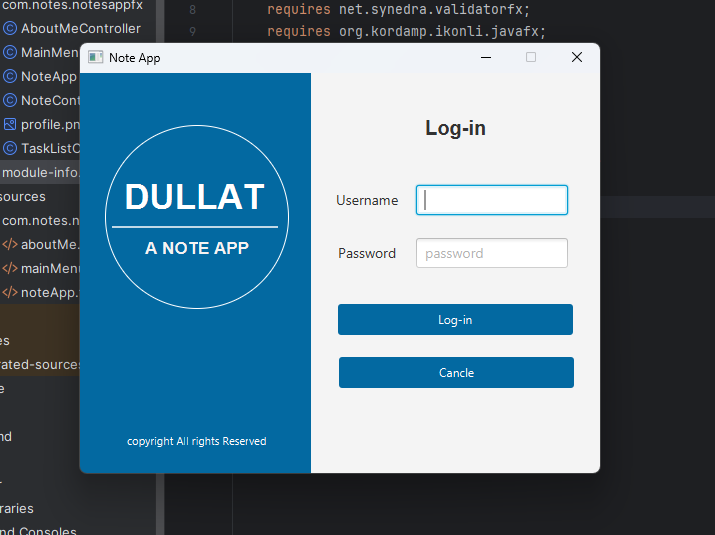
[8. LICENSE 29](#_Toc163213601)

# 1. Introduction

**The NotesAppFX** is a note-making application developed using **JavaFX framework**. It allows users to **create, store, load, edit, and update notes**. The application features a login screen for authentication and a main view for managing notes. Additionally, it includes an **"About"** section providing information about the developer and links to external projects.

**Projects and files** (<https://github.com/dullat/>)

**Preview 1: log-in SCREEN**

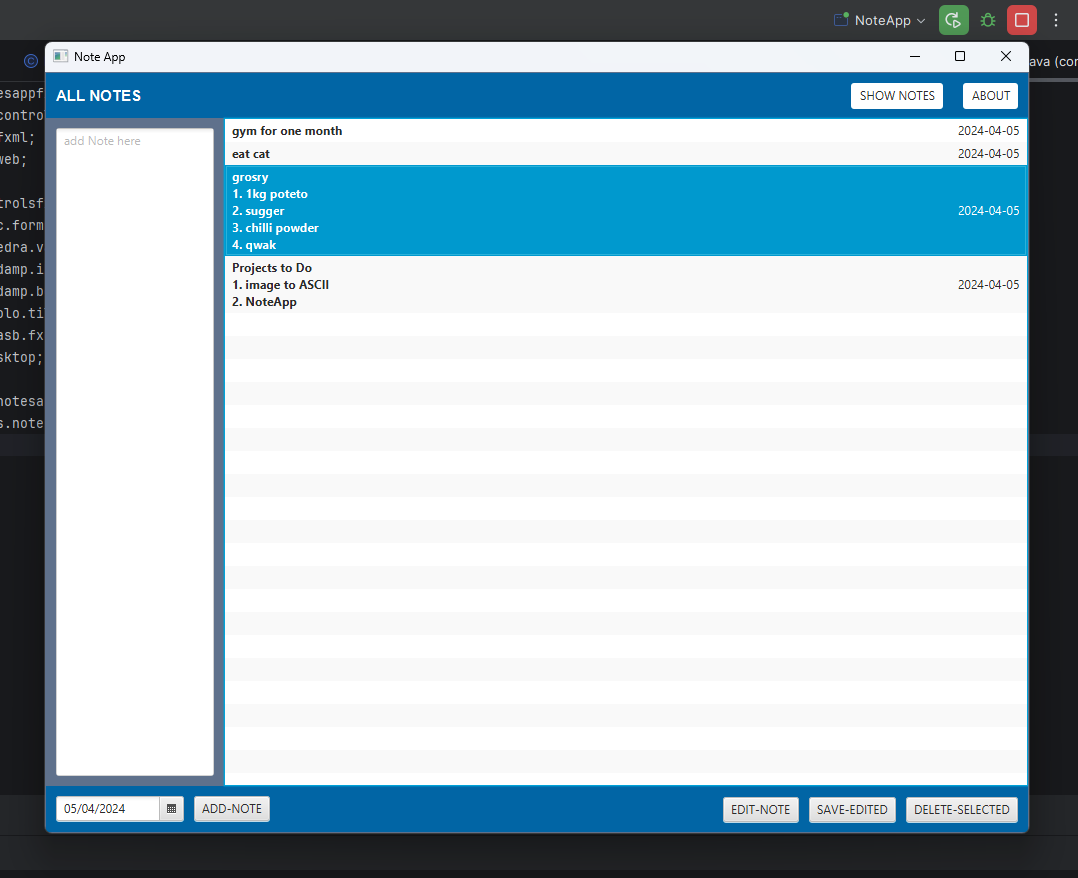


# 2. Project Overview

The purpose of the NotesAppFX is to provide users with a simple and intuitive platform for managing their notes. Key features include:

* **User authentication with a login screen.**
* **Ability to create, view, edit, and delete notes.**
* **Storage of notes in a serialized file format (.sr).**
* **Access to an "About" section with links to the developer's profiles and projects.**

**Preview 2: Main SCREEN**

****

# 3. System Architecture

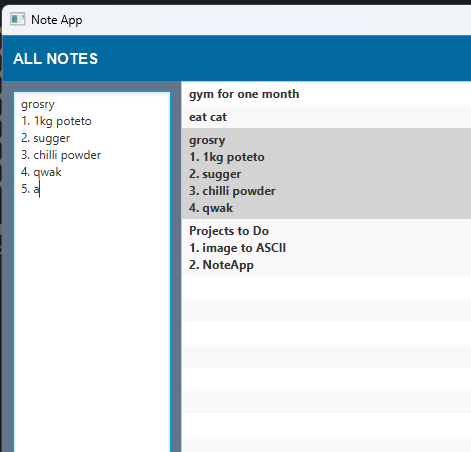
The system architecture of NotesAppFX follows a modular design, with separate components for the user interface, data management, and external integrations.

# 4. Technology Stack

The technology stack used in the project includes:

* [**JavaFX**](https://openjfx.io/) for building the graphical user interface.
* **ControlsFX, FormsFX, ValidatorFX, Ikonli,** and **BootstrapFX** for UI components and styling.
* **TilesFX** for displaying information tiles.
* **FXGL** for game development utilities.
* Java Desktop for accessing desktop functionalities.
* [**Scene Buide**](https://gluonhq.com/products/scene-builder/)**r** for building UI

**Preview 3: editing Notes**



# 5. Implementation Details

### **Login Screen**

The [**login screen**](#3vutfqi8y3yy) provides fields for entering a username and password. Upon successful authentication, users are directed to the main menu.

### **Main Menu**

The [**main menu**](#ak8ho2jek4a) displays a list of notes along with options to add, edit, and delete notes. It also includes a date picker for setting note dates.

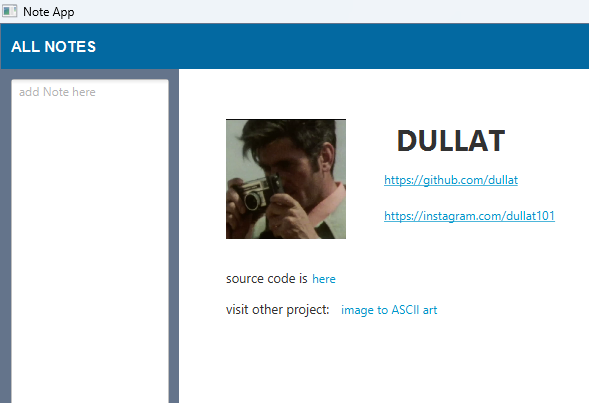
### **Note** [**Editing**](#dhqeiseqjcvl) **and Saving**

Users can add new notes, edit existing notes, and save changes. Notes are stored in a serialized file format for persistence.

### **About Section**

The "About" section displays information about the developer and includes hyperlinks to external profiles and projects.

**Preview 4: about screen**



# 6. Code Structure

The project follows a structured directory layout, with separate packages for controllers, models, and views. Key classes include **NoteApp, MainMenuController, TaskListCell,** and **AboutMeController.**

* **Main Class** (**NoteApp.java**):
  + Entry point of the application.
  + Loads the login screen (**noteApp.fxml**).
  + Sets up the UI elements and handles login functionality.
  + Upon successful login, loads the main menu screen (mainMenu.fxml).
* **FXML Files:**
  + **noteApp.fxml:** Defines the login screen layout.
  + **mainMenu.fxml:** Defines the main menu screen layout.
* **Controller Classes:**
  + **NoteControls.java:** Controller for the login screen.
  + **MainMenuController.java:** Controller for the main menu screen.
* **Helper Classes:**
  + **TaskListCell.java:** Custom list cell renderer for displaying tasks in the main menu screen.
* **CSS Files:**
  + No CSS files are explicitly mentioned in the provided code, but they could be used for styling the UI components defined in the FXML files.
* **Dependencies:**
  + Imports necessary JavaFX classes for UI components.
  + Imports **ImageIO** and **BufferedImage** for handling image loading.
  + Utilizes **java.awt.Desktop** and **java.net.URI** for opening hyperlinks.
  + Requires specific JavaFX libraries and modules for additional functionality (**org.controlsfx, com.dlsc.formsfx,** etc.).
* **Error Handling:**
  + Uses **try-catch** blocks for handling exceptions during image loading and hyperlink navigation.

**NoteApp.java**

package com.notes.notesappfx;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.image.Image;

import javafx.scene.layout.BorderPane;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

import javax.imageio.ImageIO;

import java.awt.image.BufferedImage;

import java.io.IOException;

public class NoteApp extends Application {

private BorderPane root;

private Button loginButton;

private TextField usernameTextField;

private TextField passwordTextField;

private Label warnText;

@Override

public void start(Stage primaryStage) throws IOException {

FXMLLoader loader = new FXMLLoader(getClass().getResource("noteApp.fxml"));

root = loader.load();

// primaryStage.initStyle(StageStyle.UNDECORATED);

primaryStage.setTitle("Note App");

primaryStage.setScene(new Scene(root, 520, 400));

primaryStage.setResizable(false);

primaryStage.show();

// get login btn directly using its fx id

loginButton = (Button) loader.getNamespace().get("loginButton");

usernameTextField = (TextField) loader.getNamespace().get("usernameTextField");

passwordTextField = (TextField) loader.getNamespace().get("passwordTextField");

warnText = (Label) loader.getNamespace().get("warnText");

loginButton.setOnAction(event -> {

if(usernameTextField.getText().equals("user") && passwordTextField.getText().equals("user")){

warnText.setText("welcome");

try {

// load main menu FXML

FXMLLoader loader2 = new FXMLLoader(getClass().getResource("mainMenu.fxml"));

BorderPane mainMenuRoot = loader2.load();

// set main menu as center of the root borderpane

root.setCenter(mainMenuRoot);

root.setLeft(null);

root.setRight(null);

primaryStage.setResizable(true);

primaryStage.setHeight(800);

primaryStage.setWidth(1000);

} catch (IOException e) {

e.printStackTrace();

}

}else

warnText.setText("user: user / pass: user");

});

}

public static void main(String[] args) {

launch(args);

}

}

**NoteControls.java**

package com.notes.notesappfx;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextArea;

import javafx.scene.control.TextField;

import javafx.stage.Stage;

public class NoteControls extends NoteApp {

@FXML

private Button cancleButton;

public void cancleButtonAction(ActionEvent e) {

Stage stage = (Stage) cancleButton.getScene().getWindow();

stage.close();

}

}

**NoteApp.fxml**

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.PasswordField?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.layout.AnchorPane?>

<?import javafx.scene.layout.BorderPane?>

<?import javafx.scene.shape.Circle?>

<?import javafx.scene.shape.Line?>

<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="520.0" xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.notes.notesappfx.NoteControls">

<left>

<AnchorPane prefHeight="200.0" prefWidth="228.0" style="-fx-background-color: #0369a1;" BorderPane.alignment="CENTER">

<children>

<Circle fill="#ffffff00" layoutX="117.0" layoutY="144.0" radius="92.0" stroke="WHITE" strokeType="INSIDE" />

<Label alignment="CENTER" layoutY="103.0" prefHeight="26.0" prefWidth="228.0" text="DULLAT" textFill="WHITE">

<font>

<Font name="Arial Bold" size="35.0" />

</font>

</Label>

<Label alignment="CENTER" layoutX="7.0" layoutY="360.0" prefHeight="16.0" prefWidth="220.0" text="copyright All rights Reserved" textFill="WHITE">

<font>

<Font size="11.0" />

</font>

</Label>

<Label alignment="CENTER" layoutX="3.0" layoutY="165.0" prefHeight="20.0" prefWidth="228.0" text="A NOTE APP" textFill="#f5f4f4">

<font>

<Font name="Arial Bold" size="17.0" />

</font>

</Label>

<Line endX="83.0" endY="-1.0" fill="WHITE" layoutX="114.0" layoutY="155.0" opacity="0.75" startX="-81.0" startY="-1.0" stroke="WHITE" strokeWidth="2.0" />

</children>

</AnchorPane>

</left>

<right>

<AnchorPane prefHeight="400.0" prefWidth="332.0" BorderPane.alignment="CENTER">

<children>

<Label alignment="CENTER" layoutX="1.0" layoutY="38.0" prefHeight="32.0" prefWidth="288.0" text="Log-in">

<font>

<Font name="Arial Bold" size="20.0" />

</font>

</Label>

<Label layoutX="25.0" layoutY="117.0" text="Username">

<font>

<Font size="14.0" />

</font>

</Label>

<Label layoutX="27.0" layoutY="170.0" text="Password">

<font>

<Font size="14.0" />

</font>

</Label>

<TextField fx:id="usernameTextField" layoutX="105.0" layoutY="112.0" prefWidth="152.0" promptText="user-name">

<font>

<Font size="14.0" />

</font>

</TextField>

<PasswordField fx:id="passwordTextField" layoutX="105.0" layoutY="165.0" prefWidth="152.0" promptText="password">

<font>

<Font size="14.0" />

</font>

</PasswordField>

<Button fx:id="loginButton" layoutX="27.0" layoutY="231.0" mnemonicParsing="false" prefHeight="30.0" prefWidth="235.0" style="-fx-background-color: #0369a1;" text="Log-in" textFill="WHITE" />

<Button fx:id="cancleButton" layoutX="28.0" layoutY="284.0" mnemonicParsing="false" onAction="#cancleButtonAction" prefHeight="30.0" prefWidth="235.0" style="-fx-background-color: #0369a1;" text="Cancle" textFill="WHITE" />

<Label fx:id="warnText" alignment="CENTER" layoutX="3.0" layoutY="80.0" prefHeight="18.0" prefWidth="285.0" textFill="#0369a1">

<font>

<Font size="10.0" />

</font>

</Label>

</children>

</AnchorPane>

</right>

</BorderPane>

**MainMenuController.java**

package com.notes.notesappfx;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.scene.control.Button;

import javafx.scene.control.DatePicker;

import javafx.scene.control.ListView;

import javafx.scene.control.TextArea;

import javafx.scene.control.TextField;

import java.io.\*;

import java.net.URL;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.List;

import java.util.ResourceBundle;

import javafx.fxml.Initializable;

import javafx.scene.layout.BorderPane;

public class MainMenuController implements Initializable {

@FXML

private TextArea textField;

@FXML

private DatePicker datePicker;

@FXML

private ListView<String> listView;

@FXML

private Button addNewNoteButton;

@FXML

private BorderPane mainPane;

private List<String> tasks = new ArrayList<>();

@Override

public void initialize(URL url, ResourceBundle resourceBundle) {

listView.setCellFactory(param -> new TaskListCell());

loadTasks();

}

@FXML

protected void onAddButtonClickAction(ActionEvent e){

String description = textField.getText();

LocalDate selectedDate = datePicker.getValue();

if (!description.isEmpty()) {

String task = description + " - " + selectedDate.toString();

listView.getItems().add(task);

tasks.add(task);

saveTasks();

textField.clear();

}

}

private void saveTasks() {

try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("tasks.ser"))) {

oos.writeObject(tasks);

} catch (IOException e) {

e.printStackTrace();

}

}

private void loadTasks() {

File file = new File("tasks.ser");

if (file.exists()) {

try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(file))) {

tasks = (List<String>) ois.readObject();

// Populate ListView with loaded tasks

listView.getItems().addAll(tasks);

} catch (IOException | ClassNotFoundException e) {

e.printStackTrace();

}

} else {

tasks = new ArrayList<>();

}

}

@FXML

private void deleteTask() {

int selectedIndex = listView.getSelectionModel().getSelectedIndex();

if (selectedIndex >= 0) {

listView.getItems().remove(selectedIndex);

tasks.remove(selectedIndex);

saveTasks();

}

}

@FXML

private void editNoteBottonAction() {

int selectedIndex = listView.getSelectionModel().getSelectedIndex();

if (selectedIndex >= 0) {

String selectedTask = tasks.get(selectedIndex);

// Extracting the description part without the date

String[] parts = selectedTask.split(" - ");

textField.setText(parts[0]); // Set the description to the text field

// Clear the date picker selection

datePicker.setValue(null);

}

}

@FXML

private void saveEditedButtonAction() {

int selectedIndex = listView.getSelectionModel().getSelectedIndex();

if (selectedIndex >= 0) {

String editedTaskDescription = textField.getText();

LocalDate selectedDate = datePicker.getValue();

if (selectedDate == null) {

// If no new date is specified, use the old date from the selected task

String selectedTask = tasks.get(selectedIndex);

// Extracting the date part

String[] parts = selectedTask.split(" - ");

String oldDate = parts.length > 1 ? parts[1] : "";

// Reconstruct the task with the edited description and the old date

editedTaskDescription += " - " + oldDate;

} else {

// If a new date is specified, use it

editedTaskDescription += " - " + selectedDate.toString();

}

// Update the task list view and the tasks list

tasks.set(selectedIndex, editedTaskDescription);

listView.getItems().set(selectedIndex, editedTaskDescription);

// Save the tasks to file

saveTasks();

// Clear the text field

textField.clear();

// Clear the date picker selection

datePicker.setValue(null);

}

}

@FXML

private void aboutButtonAction() throws IOException {

try {

// Load the main menu FXML

FXMLLoader loader = new FXMLLoader(getClass().getResource("aboutMe.fxml"));

BorderPane aboutMe = loader.load();

mainPane.setCenter(aboutMe);

} catch (IOException e) {

e.printStackTrace();

}

}

@FXML

private void showNotesButtonAction(){

mainPane.setCenter(listView);

}

}

**mainMenu.fxml**

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.DatePicker?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.ListView?>

<?import javafx.scene.control.SplitPane?>

<?import javafx.scene.control.TextArea?>

<?import javafx.scene.layout.BorderPane?>

<?import javafx.scene.layout.HBox?>

<?import javafx.scene.text.Font?>

<BorderPane prefHeight="475.0" prefWidth="639.0" xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.notes.notesappfx.MainMenuController">

<center>

<SplitPane nodeOrientation="LEFT\_TO\_RIGHT" prefHeight="200.0" prefWidth="200.0" style="-fx-background-color: #64748b;" BorderPane.alignment="CENTER">

<items>

<BorderPane fx:id="mainPane" prefHeight="400.0" prefWidth="606.0">

<top>

<BorderPane prefHeight="40.0" prefWidth="631.0" style="-fx-background-color: #0369a1;" BorderPane.alignment="CENTER">

<left>

<Label text="ALL NOTES" textFill="WHITE" BorderPane.alignment="CENTER">

<font>

<Font name="Arial Bold" size="15.0" />

</font>

</Label>

</left>

<opaqueInsets>

<Insets />

</opaqueInsets>

<padding>

<Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />

</padding>

<right>

<HBox nodeOrientation="RIGHT\_TO\_LEFT" prefHeight="100.0" prefWidth="200.0" spacing="20.0">

<children>

<Button mnemonicParsing="false" onAction="#aboutButtonAction" style="-fx-background-color: white;" text="ABOUT" />

<Button mnemonicParsing="false" onAction="#showNotesButtonAction" style="-fx-background-color: white;" text="SHOW NOTES" />

</children>

</HBox>

</right>

</BorderPane>

</top>

<bottom>

<BorderPane prefHeight="46.0" prefWidth="455.0" style="-fx-background-color: #0369a1;" BorderPane.alignment="CENTER">

<padding>

<Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />

</padding>

<left>

<HBox prefHeight="37.0" prefWidth="295.0" spacing="10.0" BorderPane.alignment="CENTER">

<children>

<DatePicker fx:id="datePicker" prefHeight="25.0" prefWidth="128.0" promptText="set date" />

<Button fx:id="addNewNoteButton" mnemonicParsing="false" onAction="#onAddButtonClickAction" text="ADD-NOTE" />

</children>

</HBox>

</left>

<right>

<HBox alignment="CENTER\_RIGHT" prefHeight="26.0" prefWidth="311.0" spacing="10.0" BorderPane.alignment="CENTER">

<children>

<Button fx:id="editNoteButton" mnemonicParsing="false" onAction="#editNoteBottonAction" text="EDIT-NOTE" />

<Button fx:id="saveEditedButton" mnemonicParsing="false" onAction="#saveEditedButtonAction" text="SAVE-EDITED" />

<Button fx:id="deleteButton" mnemonicParsing="false" onAction="#deleteTask" text="DELETE-SELECTED" />

</children>

</HBox>

</right>

</BorderPane>

</bottom>

<left>

<TextArea fx:id="textField" prefHeight="311.0" prefWidth="158.0" promptText="add Note here" BorderPane.alignment="CENTER">

<BorderPane.margin>

<Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />

</BorderPane.margin>

</TextArea>

</left>

<center>

<ListView fx:id="listView" prefHeight="200.0" prefWidth="200.0" BorderPane.alignment="CENTER" />

</center>

</BorderPane>

</items>

</SplitPane>

</center>

</BorderPane>

**TaskListCell.java**

package com.notes.notesappfx;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.control.Label;

import javafx.scene.control.ListCell;

import javafx.scene.layout.GridPane;

import javafx.scene.layout.HBox;

import javafx.scene.layout.Priority;

import javafx.scene.text.Text;

public class TaskListCell extends ListCell<String> {

@Override

protected void updateItem(String item, boolean empty) {

super.updateItem(item, empty);

if (empty || item == null) {

setText(null);

setGraphic(null);

} else {

String[] parts = item.split(" - ");

String description = parts[0];

String date = parts[1];

// custom layout

Label descriptionLabel = new Label(description);

Label dateLabel = new Label(date);

// align labels

descriptionLabel.setAlignment(Pos.CENTER\_LEFT);

dateLabel.setAlignment(Pos.CENTER\_RIGHT);

// padding

descriptionLabel.setPadding(new Insets(0, 10, 0, 0));

dateLabel.setPadding(new Insets(0, 0, 0, 10));

descriptionLabel.setStyle("-fx-font-weight: bold;");

// add items

GridPane gridPane = new GridPane();

gridPane.addRow(0, descriptionLabel, dateLabel);

GridPane.setHgrow(descriptionLabel, Priority.ALWAYS); // Expand description to fill space

GridPane.setHgrow(dateLabel, Priority.NEVER); // Keep date label fixed size

setGraphic(gridPane);

}

}

**aboutMeController.java**

package com.notes.notesappfx;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.scene.input.MouseEvent;

import java.awt.Desktop;

import java.net.URI;

public class AboutMeController {

@FXML

public void mouseClickedInsta(ActionEvent e) {

try {

Desktop.getDesktop().browse(new URI("https://instagram.com/dullat101"));

} catch (Exception ex) {

ex.printStackTrace();

}

}

@FXML

public void mouseClickedGithub(ActionEvent e) {

try {

Desktop.getDesktop().browse(new URI("https://github.com/dullat"));

} catch (Exception ex) {

ex.printStackTrace();

}

}

@FXML

public void visitOtherProject(ActionEvent e) {

try {

Desktop.getDesktop().browse(new URI("https://github.com/dullat/java-project-ASCIIart"));

} catch (Exception ex) {

ex.printStackTrace();

}

}

@FXML

public void visitSourceCode(ActionEvent e) {

try {

Desktop.getDesktop().browse(new URI("https://github.com/dullat/java-project-NotesApp"));

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

**aboutMe.fxml**

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Hyperlink?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.image.Image?>

<?import javafx.scene.image.ImageView?>

<?import javafx.scene.layout.AnchorPane?>

<?import javafx.scene.layout.BorderPane?>

<?import javafx.scene.text.Font?>

<BorderPane prefHeight="400.0" prefWidth="600.0" style="-fx-background-color: white;" xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.notes.notesappfx.AboutMeController">

<center>

<BorderPane prefHeight="200.0" prefWidth="200.0" BorderPane.alignment="CENTER">

<center>

<AnchorPane prefHeight="372.0" prefWidth="562.0" BorderPane.alignment="CENTER">

<children>

<ImageView fitHeight="126.0" fitWidth="120.0" layoutX="47.0" layoutY="50.0" pickOnBounds="true" preserveRatio="true">

<image>

<Image url="file:/C:/Users/chusa/IdeaProjects/notesAppFx/src/main/java/com/notes/notesappfx/profile.png" />

</image>

</ImageView>

<Label layoutX="217.0" layoutY="50.0" text="DULLAT">

<font>

<Font name="System Bold" size="29.0" />

</font>

</Label>

<Hyperlink fx:id="mouseClickedLink" layoutX="201.0" layoutY="99.0" onAction="#mouseClickedGithub" text="https://github.com/dullat" underline="true" />

<Hyperlink fx:id="mouseClickedLink1" layoutX="201.0" layoutY="135.0" onAction="#mouseClickedInsta" text="https://instagram.com/dullat101" underline="true" />

<Hyperlink layoutX="158.0" layoutY="229.0" onAction="#visitOtherProject" prefHeight="23.0" prefWidth="229.0" text="image to ASCII art" />

<Label layoutX="47.0" layoutY="231.0" text="visit other project:">

<font>

<Font size="13.0" />

</font>

</Label>

<Label layoutX="47.0" layoutY="200.0" text="source code is">

<font>

<Font size="13.0" />

</font>

</Label>

<Hyperlink layoutX="129.0" layoutY="198.0" onAction="#visitSourceCode" prefHeight="23.0" prefWidth="229.0" text="here" />

</children>

</AnchorPane>

</center>

</BorderPane>

</center>

</BorderPane>

# 7. References

* JavaFX Documentation:<https://openjfx.io/>
* ControlsFX Documentation: <https://controlsfx.github.io/>
* BootstrapFX Documentation:<https://github.com/kordamp/bootstrapfx>
* My GitHub profile (<https://github.com/dullat/>)

# 8. LICENSE

**MIT License**

Copyright (c) 2024 DULLAT

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.