Grade Summary App Technical Document

**TASK**

I was tasked to make a small application that would accept user input of marks store these and then display on a results screen showing the number of marks entered, average marks, total marks, the highest mark and the lowest mark. There should be a button on the results screen to navigate back to the main page.

**PSEUDOCODE**

1. ON BUTTON press ADD MARK

Store value in text box as int storedValue

Set text box to blank.

2. ON BUTTON press SUBMIT

IF total number of storedValues entered is = 0

PRINT error message

Else load results page

3. PRINT total number of storedValues entered

PRINT average number of storedValues entered

PRINT total sum of storedValues

SET text to green

PRINT highest storedValue entered

SET text to red

PRINT lowest storedValue

4.ON BUTTON press BACK

RETURN to Main Screen

**SCREEN DESIGNS**

Diagram

Description automatically generated

**OBJECT EXPLAINATIONS**

header – a label prompting user to enter marks

textbox – a text field for users to input data

addMark – a button to store the users input into an array list

submit – a button to take user to the second page

errorMessage – a blank label used to display an error message

resultsHeader – a label that displays message “Results”

numberOfMarks – a label to display the number of marks entered

averageMark - a label to display the average of marks entered

totalMark - a label to display the total sum of marks entered

highestMark - a label to display the highest mark entered

lowestMark - a label to display the lowest mark entered

back – a button to take the user back to the homescreen

**SCREENSHOTS**

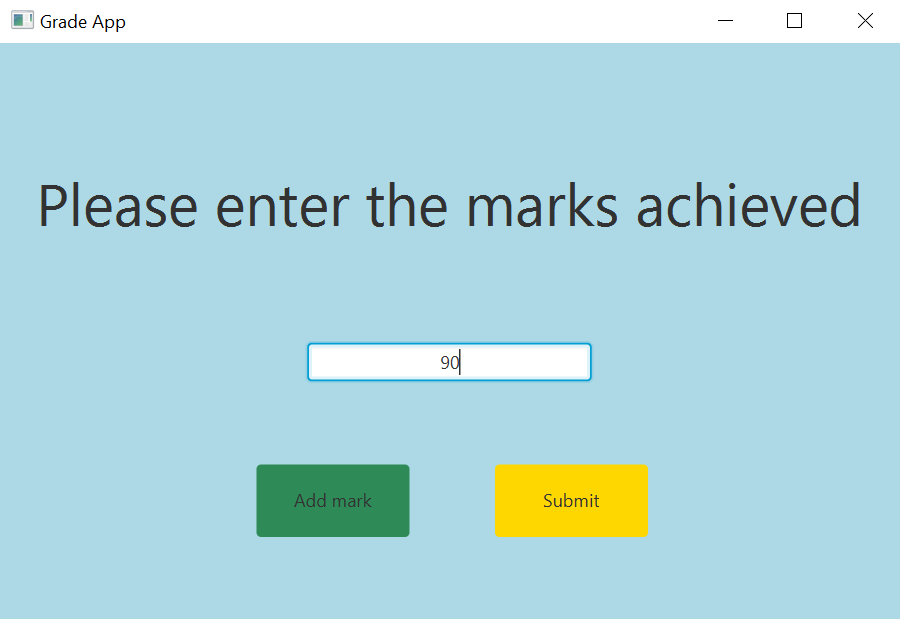


Figure Home screen

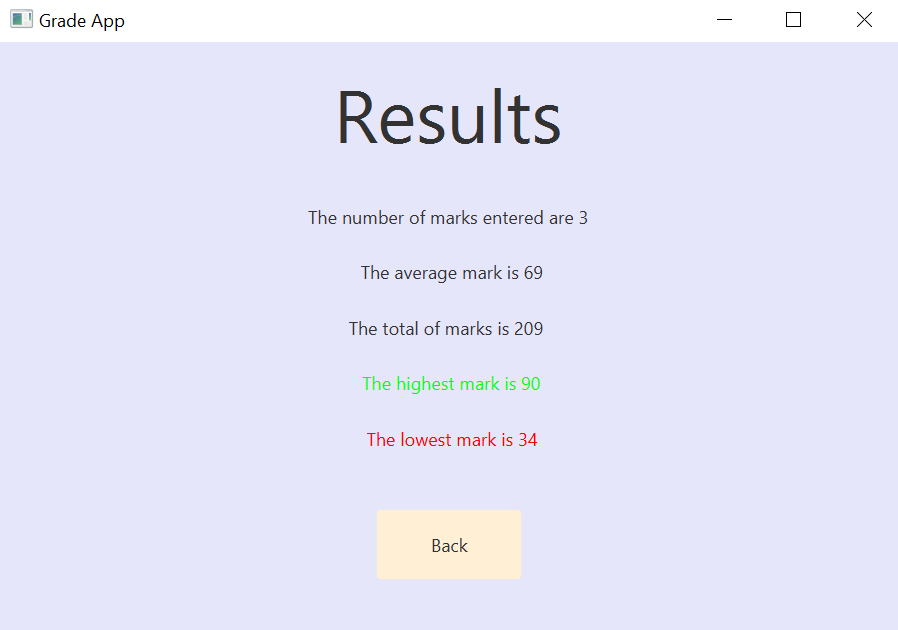


Figure Results screen

**TESTING**

|  |  |  |  |
| --- | --- | --- | --- |
| Actions | Expected Results | Actual Result | Pass/Fail |
| Run code | Home screen displayed | Home screen displayed | PASS |
| Enter letter instead of number | Error message | Error in console, no feedback for user | FAIL (can easily add a message for the user) |
| Enter negative numbers | Negative results displayed | Negative results displayed, | PASS |
| Enter extreme values (9999999) | Results displayed normally | Results displayed normally | PASS |
|  |  |  |  |
| Tested by – | Robert Lothian | 29/04/21 | PASSED |

The program works as it should although it does have room for some quality-of-life improvements for user, things like an error message for a letter being entered rather than a number and functionality for the cursor to move back to the text box after the add mark button is pressed (although functionality for add mark has been entered on the text field when the enter button is pressed).

If this project was to be expanded, I would recommend adding in these features however as it stands it functions and fits the brief.

**CODE LISTING**

Main

package sample;  
  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.stage.Stage;  
  
public class Main extends Application {  
  
 @Override  
 public void start(Stage primaryStage) throws Exception{  
 Parent root = FXMLLoader.*load*(getClass().getResource("FirstPage.fxml"));  
 primaryStage.setTitle("Grade App");  
 primaryStage.setScene(new Scene(root, 600, 400));  
 primaryStage.show();  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

Controller

package sample;  
  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Node;  
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.stage.Stage;  
import java.io.IOException;  
  
  
public class Controller {  
 @FXML Button addMark, submit;  
 @FXML TextField textBox;  
 @FXML Label errorMessage;  
  
  
 public void initialize() {  
  
 addMark.setOnAction((event) -> { //on press of add mark button execute following code  
 String storedValue = textBox.getText(); //store value from text box  
 Integer.*parseInt*(storedValue); //change value from string to int  
 StoredMarks.*add*(storedValue);//adds the entered value into the array  
  
  
 textBox.setText("");//clears the text box  
 });  
  
 submit.setOnAction((event) -> { //on button press load second page  
  
 if (StoredMarks.*getTotal*() == 0) //code to write error message if user hits submit without entering a mark  
 {  
 errorMessage.setText("You have not entered a mark, try again.");  
 }  
  
 else  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource("SecondPage.fxml"));  
 Parent root = loader.load();  
  
 Scene scene = new Scene(root);  
 Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();  
 stage.setScene(scene);  
 stage.show();  
 }  
 catch (IOException e) {  
 }  
 });  
 }  
  
  
}

Stored Marks

package sample;  
  
import java.util.ArrayList;  
import java.util.Collections;  
  
public class StoredMarks {  
  
 private static ArrayList<Integer> *storedMarks* = new ArrayList<>();//creates new array list to store marks  
  
 public static void add(String s) { //adds value entered into array and parses as an int  
  
 *storedMarks*.add(Integer.*valueOf*(s));  
 }  
  
  
 public static int getTotal(){ //get total number of marks entered  
  
 return *storedMarks*.size();  
 }  
  
 public static int getAverage(){ //get average number of marks entered  
  
 return *getSum*()/*getTotal*();  
 }  
  
 public static int getSum(){ //get total of marks entered  
  
 int sum = 0;  
 for (Integer element : *storedMarks*)  
 {  
 sum += Integer.*parseInt*(String.*valueOf*(element));  
 }  
 return sum;  
 }  
  
  
  
 public static int highestMark() //get highest mark  
 {  
 Collections.*sort*(*storedMarks*);  
 return *storedMarks*.get(0+ *storedMarks*.size()-1);  
 }  
  
 public static int lowestMark() //get lowest mark  
 {  
  
 Collections.*sort*(*storedMarks*);  
 return *storedMarks*.get(0);  
 }  
}

FirstPage.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="400.0" prefWidth="600.0" style="-fx-background-color: lightblue;" xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="sample.Controller">  
 <children>  
 <Label fx:id="header" layoutX="24.0" layoutY="80.0" style="-fx-background-color: lightblue;" text="Please enter the marks achieved">  
 <font>  
 <Font name="Gadugi" size="39.0" />  
 </font>  
 </Label>

<TextField fx:id="textBox" alignment="CENTER" layoutX="205.0" layoutY="200.0" prefHeight="25.0" prefWidth="189.0" promptText="Enter mark numerically eg. 54" />

<Button fx:id="addMark" defaultButton="true" layoutX="171.0" layoutY="281.0" mnemonicParsing="false" prefHeight="47.0" prefWidth="102.0" style="-fx-background-color: seagreen;" text="Add mark" />

<Button fx:id="submit" layoutX="330.0" layoutY="281.0" mnemonicParsing="false" prefHeight="47.0" prefWidth="102.0" style="-fx-background-color: gold; -fx-border-style: line;" text="Submit" />

<Label fx:id="errorMessage" alignment="CENTER" layoutX="148.0" layoutY="356.0" prefHeight="17.0" prefWidth="304.0" />  
 </children>  
</AnchorPane>

Second Page Controller

package sample;  
  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Node;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.paint.Color;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
  
  
public class SecondPageController {  
  
 @FXML private Button back;  
 @FXML private Label numberOfMarks, averageMark, totalMark, lowestMark, highestMark;  
 @FXML private Color red = Color.*color*(1,0,0); //define colours used  
 @FXML private Color green = Color.*color*(0,1,0);  
  
 public void initialize()  
 {  
 //the following displays the results  
  
 numberOfMarks.setText("The number of marks entered are " + StoredMarks.*getTotal*());  
 averageMark.setText("The average mark is " + StoredMarks.*getAverage*());  
 totalMark.setText("The total of marks is " + StoredMarks.*getSum*());  
 highestMark.setTextFill(green);  
 highestMark.setText("The highest mark is " + StoredMarks.*highestMark*());  
 lowestMark.setTextFill(red);  
 lowestMark.setText("The lowest mark is " + StoredMarks.*lowestMark*());  
  
 back.setOnAction((event) -> { //on button press load first page  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource("FirstPage.fxml"));  
 Parent root = loader.load();  
  
 Scene scene = new Scene(root);  
 Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();  
 stage.setScene(scene);  
 stage.show();  
 }  
 catch (IOException e) {  
 }  
 });  
 }  
}

SecondPage.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="400.0" prefWidth="600.0" style="-fx-background-color: lavender;" xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="sample.SecondPageController">  
 <children>  
 <Button fx:id="back" layoutX="252.0" layoutY="312.0" mnemonicParsing="false" prefHeight="46.0" prefWidth="96.0" style="-fx-background-color: papayawhip;" text="Back" />  
 <Label fx:id="resultsHeader" layoutX="223.0" layoutY="14.0" text="Results">  
 <font>  
 <Font name="Gadugi" size="49.0" />  
 </font></Label>  
 <Label fx:id="numberOfMarks" layoutX="219.0" layoutY="98.0" prefHeight="37.0" prefWidth="153.0" />  
 <Label fx:id="numberOfMarks" alignment="CENTER" layoutX="55.0" layoutY="98.0" prefHeight="37.0" prefWidth="489.0" text="Number of Marks" />  
 <Label fx:id="averageMark" alignment="CENTER" layoutX="74.0" layoutY="135.0" prefHeight="37.0" prefWidth="455.0" text="Average" />  
 <Label fx:id="totalMark" alignment="CENTER" layoutX="48.0" layoutY="172.0" prefHeight="37.0" prefWidth="500.0" text="Total" />  
 <Label fx:id="highestMark" alignment="CENTER" layoutX="46.0" layoutY="209.0" prefHeight="37.0" prefWidth="512.0" text="Highest" />  
 <Label fx:id="lowestMark" alignment="CENTER" layoutX="46.0" layoutY="246.0" prefHeight="37.0" prefWidth="512.0" text="Lowest" />  
 </children>  
</AnchorPane>