Semester Project

Chatting Application – CHADDING

**Chatting Application**

Chadding is a real time world chatting application where users can communicate via texting. Each user can have one account based upon their Google Account and chat with anyone in the world. Users can change the theme according to their likings to make theirself home in the software.

Technical Description:  
        Application uses Google Authentication to create accounts and accounts are being managed by Firebase's Real Time Database Firestore in order to manage their data and chatrooms with others. JavaFX is used to design the Front-End of the Software whereas backend is maintained by Third Party Libraries using the Maven Build System of Java to integrate multiple dependancies.

Authors:

* Ibraheem Bin Haseeb
* Marhaba Eman Ahmad

Classes Description:

* Hello Application:

It consists of the main method and initializes the firebase project.

* Animations:

Animations class includes the functionality of animations that are displayed on the Home Page. It consists of three built in java classes Path, PathTransition and RotateTransition.

* ShapesComponents:

Creates the shapes on which animations are applied and also specifies the path on which the animations would move.

* Home Page:

It contains the name of the application and sign up and log in buttons along with animations.

* Sign Up/Log In Page:

Sign up and Log In buttons opens a new window and has the sign in/log in options. User can create username and password in sign up screen and use the application right away. By pressing log in button, you can directly enter your existing username and password and chat.

* Hover In:

Hover In class changes styles on the button upon which you hover.

* Hover Out:

Hover out changes the button colour to the original colour when you hover out from that button.

* Inbox Users:

Inbox Users renders buttons from which user can open specific chats.

* Messages:

Messages renders messages of the specific user.

* **Chadding Page**:

User can interact with everything and make selections for different chats.

* Auth

Auth allows the user to create accounts as well as login

CHADDING PAGE

package com.example.semproject.Pages;  
  
import com.example.semproject.Events.HoverIn;  
import com.example.semproject.Events.HoverOut;  
import com.example.semproject.HelloApplication;  
import com.example.semproject.MyComponents.InboxUsers;  
import com.example.semproject.MyComponents.Messages;  
import com.google.api.core.ApiFuture;  
import com.google.cloud.firestore.\*;  
import com.google.cloud.firestore.EventListener;  
import com.google.firebase.cloud.FirestoreClient;  
import javafx.application.Platform;  
import javafx.event.ActionEvent;  
import javafx.event.Event;  
import javafx.event.EventHandler;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Cursor;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.input.KeyEvent;  
import javafx.scene.input.MouseEvent;  
import javafx.scene.layout.Background;  
import javafx.scene.layout.BorderPane;  
import javafx.scene.layout.HBox;  
import javafx.scene.layout.VBox;  
import javafx.scene.paint.Color;  
import javafx.stage.Stage;  
import javafx.stage.WindowEvent;  
  
import javax.annotation.Nullable;  
import java.util.\*;  
import java.util.concurrent.ExecutionException;  
  
public class ChaddingPage extends BorderPane {  
 *// ==================================* ArrayList<HashMap<String, Object>> messages = new ArrayList<>();  
 ArrayList<HashMap<Object, Object>> list = new ArrayList<>();  
 Firestore db = FirestoreClient.getFirestore();  
  
 {  
 *// messages data* ApiFuture<QuerySnapshot> future = db.collection("chat").get();  
  
 try {  
 int counter = 0;  
 for (QueryDocumentSnapshot d:future.get().getDocuments()) {  
 if (future.get().getDocuments().size() - 1 == counter++) break;  
 messages.add((HashMap<String, Object>) d.getData());  
 }  
 } catch (Exception e) {  
 throw new RuntimeException(e);  
 }  
 }  
  
 *// ==================================* public void getInboxList() {  
 *// inbox data* CollectionReference chats = db.collection("chat");  
 Query q = chats.whereArrayContains("users", new HashMap<String, Object>() { { put("uid", "1671123299609uid"); put("username", "afds"); } });  
 ApiFuture<QuerySnapshot> inboxChats = q.get();  
  
 try {  
 for (QueryDocumentSnapshot d:inboxChats.get()) {  
  
 HashMap<String, Object> temp = (HashMap<String, Object>) d.getData();  
 list.add(new HashMap<Object, Object>() {  
 {  
 if (HelloApplication.*userId*.equals(((HashMap)((ArrayList)temp.get("users")).get(0)).get("uid"))) {  
 put("name", ((HashMap)((ArrayList)temp.get("users")).get(1)).get("username"));  
 put("uid", ((HashMap)((ArrayList)temp.get("users")).get(1)).get("uid"));  
 put("room", temp.get("id"));  
 } else {  
 put("name", ((HashMap)((ArrayList)temp.get("users")).get(0)).get("username"));  
 put("uid", ((HashMap)((ArrayList)temp.get("users")).get(0)).get("uid"));  
 put("room", temp.get("id"));  
 }  
 }  
 });  
 }  
 } catch (Exception e) {  
 System.*out*.println(e.getMessage());  
 e.printStackTrace();  
 }  
 }  
  
 Scene chaddingPageScene;  
 public ChaddingPage(Stage stage, Scene scene){  
  
 getInboxList();  
  
 chaddingPageScene = new Scene(this, 800, 500);  
 this.setStyle("-fx-background-color: #fff");  
 *// Layouts Declarations* HBox topHbox = new HBox();  
 HBox welcomeContainer = new HBox();  
 HBox buttonsContainer = new HBox();  
 HBox centerHbox = new HBox();  
 VBox leftVbox = new VBox();  
 VBox rightVbox = new VBox();  
 HBox nameContainer = new HBox();  
 HBox sendContainer = new HBox();  
 ScrollPane scrollPane = new ScrollPane();  
 VBox messages = new VBox();  
  
 *// Components* Label welcomeLabel = new Label("Welcome Back, " + HelloApplication.*userName*);  
 Button back = new Button("Back");  
 Button search = new Button("Search");  
 Button logout = new Button("Logout");  
 Label name = new Label("Name");  
 TextArea textArea = new TextArea();  
 Button send = new Button("Send");  
  
 *// Adding components to layouts* welcomeContainer.getChildren().add(welcomeLabel);  
 buttonsContainer.getChildren().addAll(back, search, logout);  
 topHbox.getChildren().addAll(welcomeContainer, buttonsContainer);  
  
 centerHbox.getChildren().addAll(leftVbox, rightVbox);  
 rightVbox.getChildren().addAll(nameContainer, scrollPane, sendContainer);  
 nameContainer.getChildren().add(name);  
 sendContainer.getChildren().addAll(textArea, send);  
  
 *// adding inbox list* for (HashMap<Object, Object> d:list) {  
 leftVbox.getChildren().add(new InboxUsers(chaddingPageScene, (String)d.get("name"), (String)d.get("room"), this.messages, messages, send, textArea, scrollPane, name));  
 }  
  
 scrollPane.setContent(messages);  
  
 this.setTop(topHbox);  
 this.setCenter(centerHbox);  
  
 *// Styling* topHbox.setMinWidth(chaddingPageScene.getWidth());  
 welcomeContainer.setMinWidth(800\*0.4);  
 buttonsContainer.setMinWidth(800\*0.6);  
 welcomeLabel.setStyle("-fx-padding: 20; -fx-font-weight: 700; -fx-font-size: 25");  
 back.setStyle(HomePage.*buttonStyles*);  
 search.setStyle(HomePage.*buttonStyles*);  
 search.setCursor(Cursor.*HAND*);  
 logout.setCursor(Cursor.*HAND*);  
 back.setCursor(Cursor.*HAND*);  
 logout.setStyle("-fx-border-radius: 5px; -fx-background-radius: 5px; -fx-background-color: #ff6060; -fx-text-fill: #fff; -fx-margin: 10px; -fx-font-size: 15");  
 topHbox.setStyle("-fx-border-width: 1; -fx-border-color: #ff0000; -fx-border-style: hidden hidden solid hidden;");  
  
 centerHbox.setMinWidth(chaddingPageScene.getWidth());  
 leftVbox.setMinWidth(chaddingPageScene.getWidth()\*0.25);  
 leftVbox.setSpacing(10);  
 leftVbox.setPadding(new Insets(10));  
 rightVbox.setMinWidth(chaddingPageScene.getWidth()\*0.75);  
 buttonsContainer.setAlignment(Pos.*TOP\_RIGHT*);  
 buttonsContainer.setPadding(new Insets(20));  
 buttonsContainer.setSpacing(10);  
 leftVbox.setStyle("-fx-border-width: 1; -fx-border-color: #ff0000; -fx-border-style: hidden solid hidden hidden;");  
  
 name.setStyle("-fx-padding: 10; -fx-font-weight: 700; -fx-font-size: 20;");  
 nameContainer.setStyle("-fx-border-width: 1; -fx-border-color: #ff0000; -fx-border-style: hidden hidden solid hidden");  
  
 scrollPane.setMinHeight(315);  
 textArea.setWrapText(true);  
 textArea.setPromptText("Type your message here");  
 rightVbox.setStyle("-fx-background-color: white");  
  
 sendContainer.setAlignment(Pos.*BOTTOM\_LEFT*);  
 send.setStyle("-fx-background-color: #fff; -fx-border-width: 1; -fx-border-color: #000; -fx-border-radius: 5; -fx-padding: 10 18 10 18");  
 send.setCursor(Cursor.*HAND*);  
 sendContainer.setSpacing(10);  
 sendContainer.setMinWidth(chaddingPageScene.getWidth() \* 0.75);  
 sendContainer.setStyle("-fx-padding: 10; -fx-border-width: 1; -fx-border-style: solid hidden hidden hidden; -fx-border-color: #ff0000");  
 textArea.setMinWidth(chaddingPageScene.getWidth() \* 0.63);  
 textArea.setStyle("-fx-background-color: #fff; -fx-border-width: 1; -fx-border-color: #000; -fx-border-radius: 5;");  
  
 *// event handling* back.addEventHandler(MouseEvent.*MOUSE\_ENTERED*, new HoverIn(back));  
 back.addEventHandler(MouseEvent.*MOUSE\_EXITED*, new HoverOut(back));  
 back.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent actionEvent) {  
 stage.setScene(scene);  
 }  
 });  
  
 search.addEventHandler(MouseEvent.*MOUSE\_ENTERED*, new HoverIn(search));  
 search.addEventHandler(MouseEvent.*MOUSE\_EXITED*, new HoverOut(search));  
  
 logout.addEventHandler(MouseEvent.*MOUSE\_ENTERED*, new HoverIn(logout));  
 logout.addEventHandler(MouseEvent.*MOUSE\_EXITED*, new HoverOut(logout));  
 logout.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent actionEvent) {  
 HelloApplication.*isAccount* = false;  
 stage.setScene(scene);  
 }  
 });  
 }  
}

Images Of Outputs:



