Ain Shams University

Faculty of Engineering

Computer and system Engineering department

Advanced Computer Programming

CSE231

World Z!
2024

Presented for:

DR. Mahmoud Khalil

ENG. Mahmoud Sohail

Presentedby:

MohamedYehiaZakaria22P0064

Omar Mohamed Mostafa 22P0197

Ahmed Wael Raafat 22P0221

AhmedMohamedTalaat22P0176

George Hany Samy 22P0173

Ezzeldin Ismail Kaoud 22p0141

Table of Contents

Introduction

This project aims to create an interactive and user-friendly platform where users can connect, share posts, and communicate. Java OOP (Object-Oriented Programming) helps us organize our code efficiently using classes and objects, making it easier to manage and scale the application. Core OOP concepts like inheritance, encapsulation, and polymorphism will be essential in building a flexible application architecture.
JavaFX, our chosen framework for the user interface, provides the tools to create a modern and engaging user experience. It offers a wide range of UI controls, supports rich 2D and 3D graphics, animations, which are crucial for developing features like user profiles, news feeds, and messaging systems. Our goal is to leverage these technologies to build a fully functional social media application, enhancing our skills in Java programming and UI design while delivering a practicalplatform.

Project Description

Our social media platform, developed using Java, aims to connect users by allowing them to create and interact with posts. To achieve this, we have designed several key classes that handle different aspects of the platform's functionality. These classes include User, Comment, CreatePost, LikePost, and Database. Each class has a specific role and works together to ensure a seamless and engaging user experience.Absolutely, let's simplify the language:
We implement these classes using JavaFX GUI (that just means how everything looks and works) to make everything easy to use and nice to look at.

1. User Class:

- Purpose: Represents a person using the social media platform.
- Main Features:

1)Stores user information like username, password, email, and profile details
2)Methods to update profile, log in, and log out.

```
public user (){}
no usages
public user (String name, int password, String email, int phone , boolean tempgender, Strin
    this.name = name ;
    this.password = password;
    this.email = email ;
    if(tempgender == true){
        gender = "M";
    }
    else{
        gender = "F";
    }
    this.phone = phone ;
    this.birthday = birthday;
    LocalDate dateOfBirth = LocalDate.parse(birthday);
    this.age = calculateAge(dateOfBirth);
    this.id = publicid;
    publicid++;
    database.register( u: this);
}
```

Figure 1: User class

2. LikePost Class:

- Purpose: Represents a "like" given to a post by a user.
- Main Features:

1) Stores the user who liked the post and the post that was liked.
2) Methods to add and remove likes.

3. Comment Class:

- Purpose: Represents a comment made by a user on a post.
- Main Features:

1) Stores the text of the comment, the user who made the comment, and the time it was posted.
2) Methods to add, edit, and delete comments.

```
3 usages
public abstract class comment {

    2 usages
    String cmnt;
    2 usages
    user u;
    1 usage
    LocalDate time = LocalDate.now();
    no usages
    public comment (String cmnt) { this.cmnt = cmnt; }

}
```

Figure 1: Comment class

4.CreatePost

Class:

- Purpose: Allows users to create new posts.
- Main Features:

1)Stores details of the post like content, the user who created it, and the time it was posted.
2)Methods to create, edit, and delete posts.

Figure 2: create post class

5.Database Class:

- Purpose: Stores and manages all data related to users, posts, comments, and likes.
- Main Features:

1) Handles storage and retrieval of user details, posts, comments, and likes.
2)Methods to save and load data, ensuring the platform keeps track of all

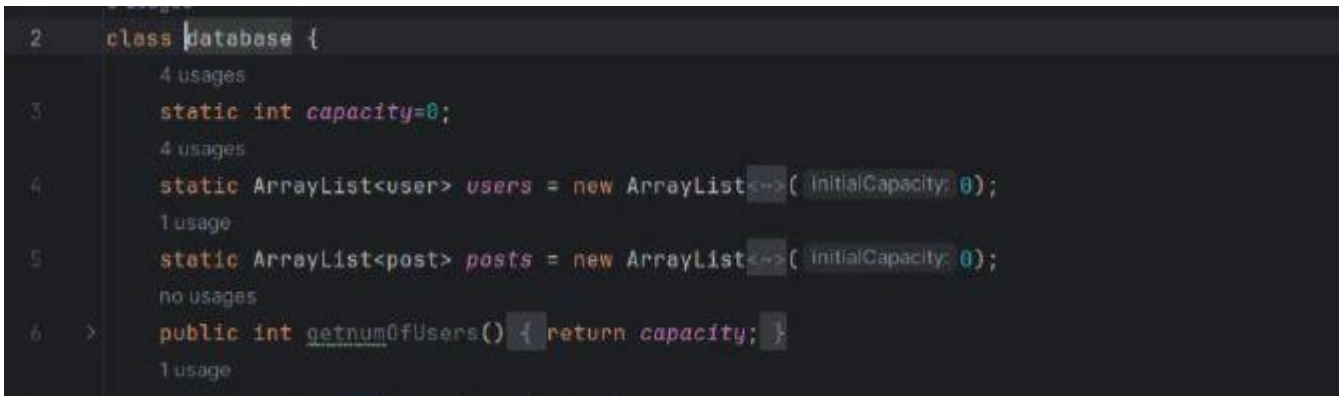


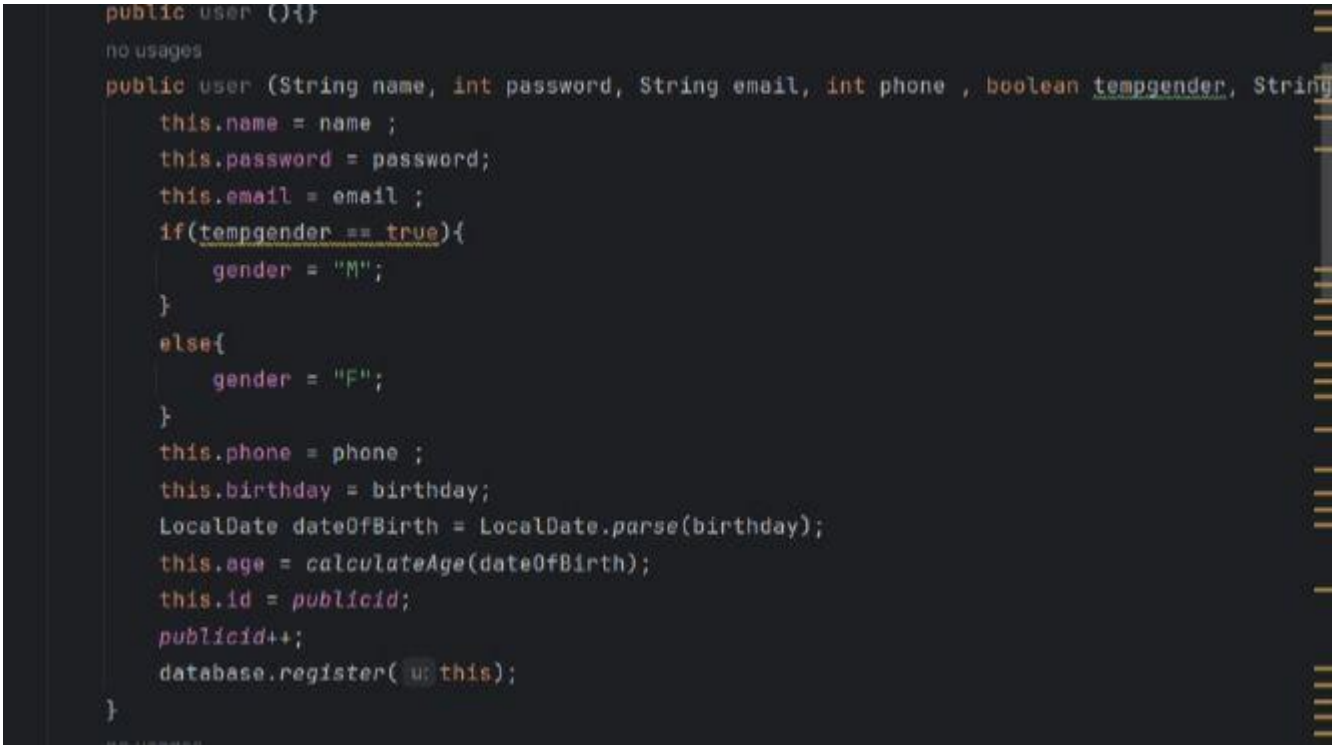
Figure 3:Database class

interactions and user information.

-Each class works together to provide the functionality needed for a social media platform, ensuring users can interact by posting content, commenting, and liking posts while keeping all their information organized and accessible.

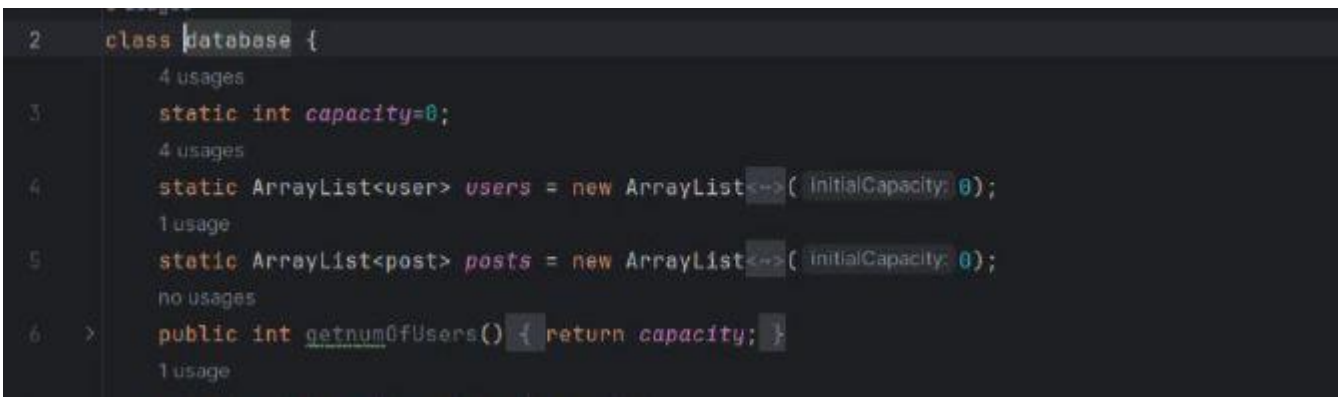
Detailed Analysis

We have class called user this class make the user to enter his name, password, email, select gender, his phone number, date of birth and his id. The class calculate his age using the date of birth entered by the user.
The class methods contain methods setter and getters for all the attribution of the user class as shown in the following:



```
public user (){}
no usages
public user (String name, int password, String email, int phone , boolean tempgender, String
    this.name = name ;
    this.password = password;
    this.email = email ;
    if(tempgender == true){
        gender = "M";
    }
    else{
        gender = "F";
    }
    this.phone = phone ;
    this.birthday = birthday;
    LocalDate dateOfBirth = LocalDate.parse(birthday);
    this.age = calculateAge(dateOfBirth);
    this.id = publicid;
    publicid++;
    database.register( u: this);
}
```

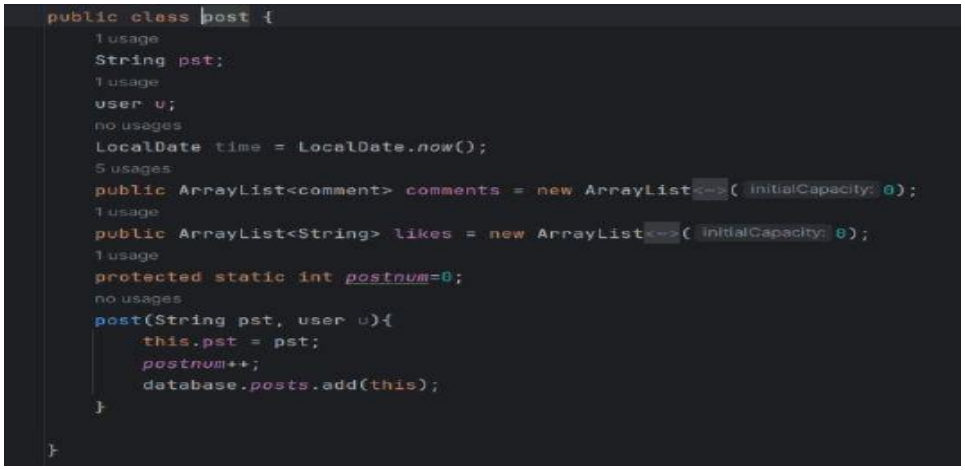
Then we have another class called database, this class stored all the data entered by the user in the user class.
Stores and manages all data related to users, posts, comments, and likes.



```
class database {
    4 usages
    static int capacity=0;
    4 usages
    static ArrayList<user> users = new ArrayList<>( initialCapacity: 0);
    1 usage
    static ArrayList<post> posts = new ArrayList<>( initialCapacity: 0);
    no usages
    public int getnumOfUsers() { return capacity; }
    1 usage
```

We have class called CreatePost this class allow the user to create new posts they also can edit these posts. The class has the following attributes:

post to create the post, u from type user to indicate the user who posted the post, time from type Local Date to indicate the date the post is published, comment from type array list to store the comments on this post in an array, likes from type array list to store the number of likes on the post and post num from type protected that count posts on the account.



```
public class post {
    1 usage
    String pst;
    1 usage
    user u;
    no usages
    LocalDate time = LocalDate.now();
    5 usages
    public ArrayList<comment> comments = new ArrayList<>( initialCapacity: 0);
    1 usage
    public ArrayList<String> likes = new ArrayList<>( initialCapacity: 0);
    1 usage
    protected static int postnum=0;
    no usages
    post(String pst, user u){
        this.pst = pst;
        postnum++;
        database.posts.add(this);
    }

}
```

Beneficiaries of the project

The beneficiaries of this social media platform, include :

1. Users: These are the people who sign up and use the platform to connect with friends and others online. They can share updates, photos, and thoughts, as well as interact with content posted by others.

2. Beginner Developers: Those who are learning to develop software or are interested in coding. They can use this platform as a practical project to understand concepts like user authentication, database, and interaction design.

Time plan

-In this project we moved on 3 phases:

Phase 1: Setting Up (2 weeks)

- Research and plan the project.
- Set up the development environment for JavaFX.
- Design the basic layout of the user interface.
- Implement user authentication (signup, login, profile setup).
- Create basic posting functionality

Phase 2: Adding Features (2 weeks)

- Enhance posting features (image uploads, tagging, hashtags).
- Implement interactive features (like, comment).
- Improve user profiles (profile picture).
- Develop the feed or timeline for users.

Phase 3: Testing and Deployment (1 week)

- Test all features thoroughly (usability, functionality).
- Gather user feedback and make necessary refinements.
- Perform final testing and optimization for performance.
- Prepare for deployment and launch the platform.

Id	Task	Team Member
1	Back end	Wael, ezz, omar
2	Front end	Wael, omar, George, Mohamed yehia
3	Documentation report	Mohamed yehia, Mohamed Talaat, George
4	Data-base connection	Wael, ezz
5	diagrams	Mohamed talaat

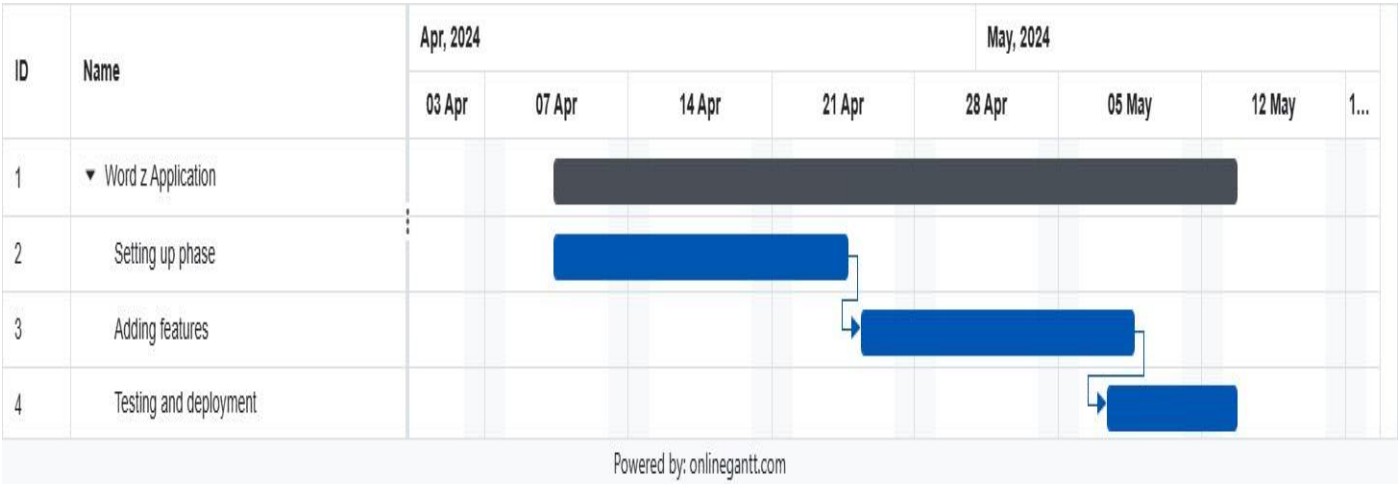


Figure 5: gantt chart

System architecture and design

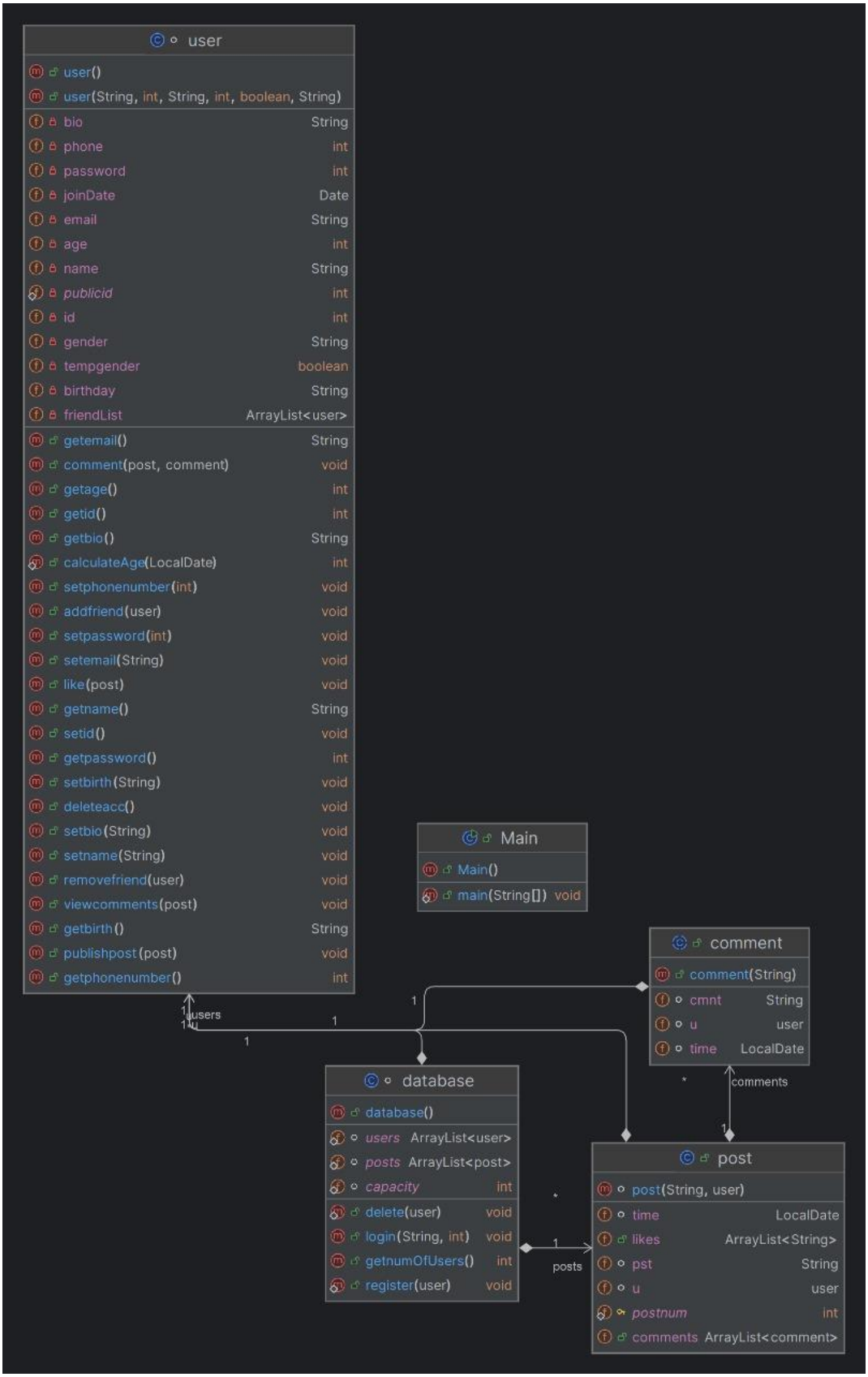


Figure 6:class diagram

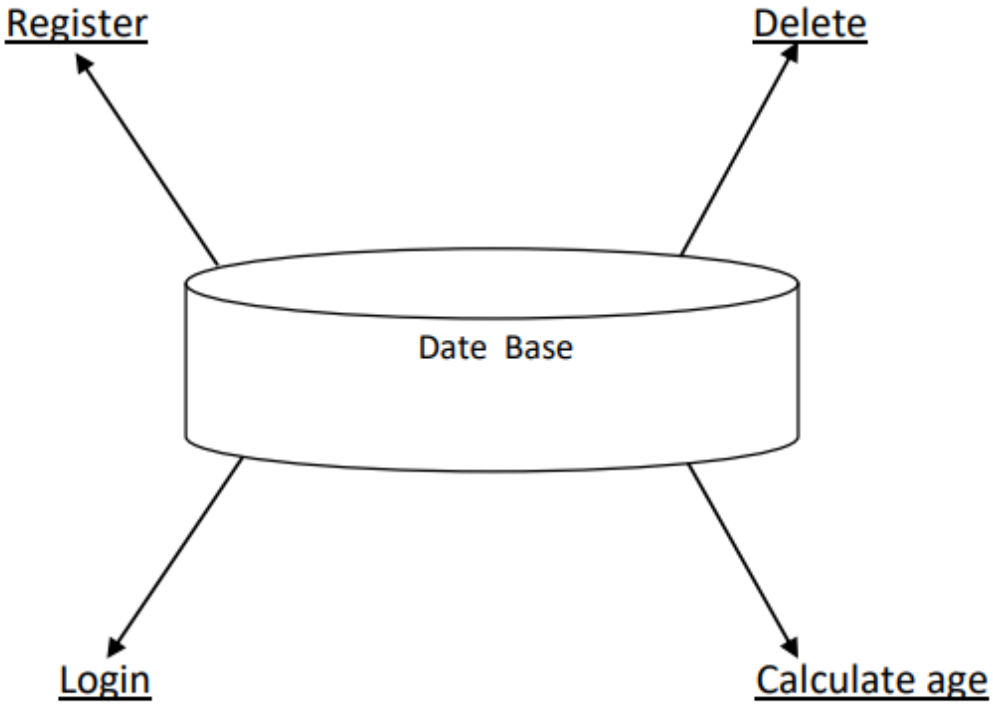


Figure 7: System architecture diagram

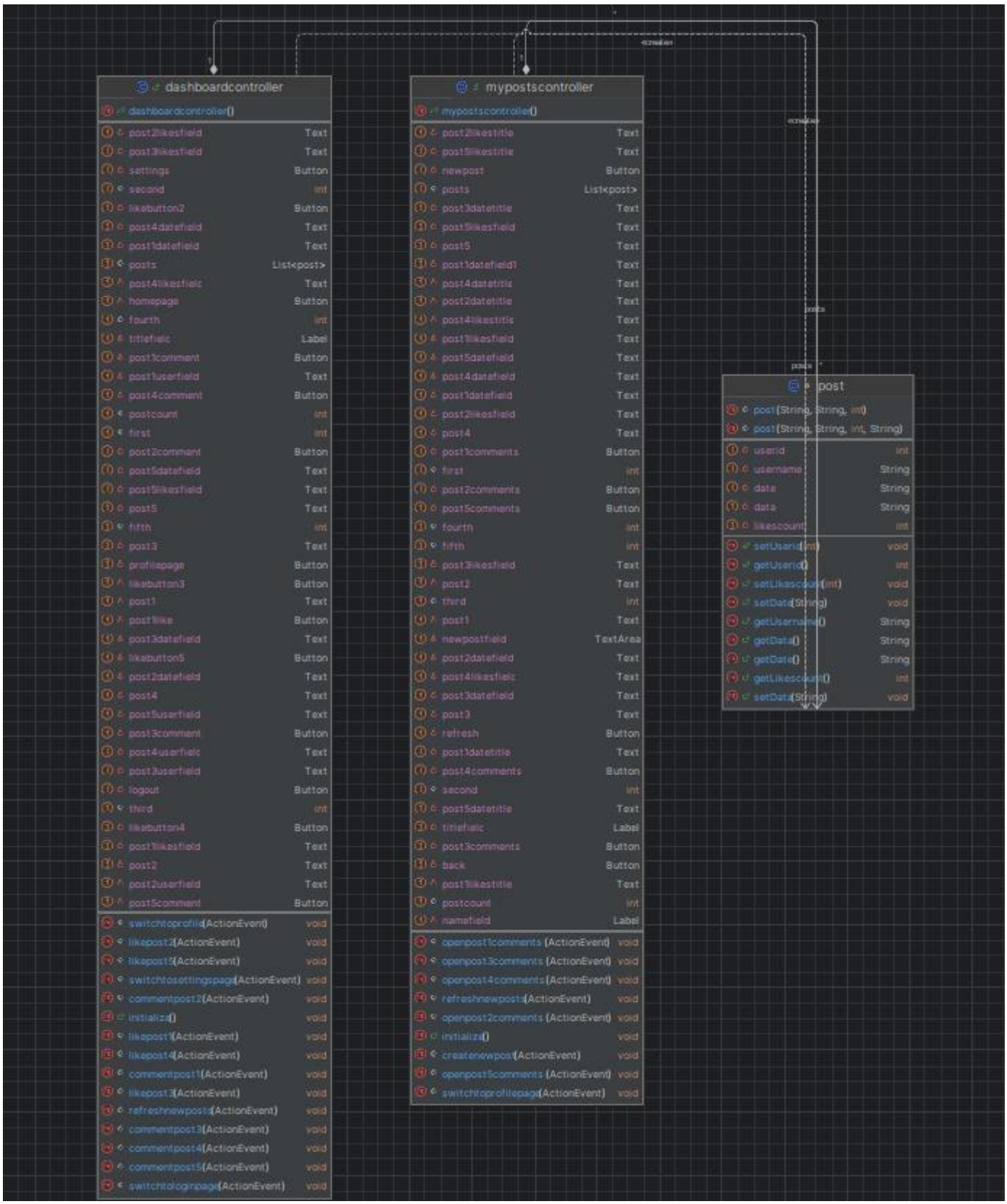


Figure 8:class diagram

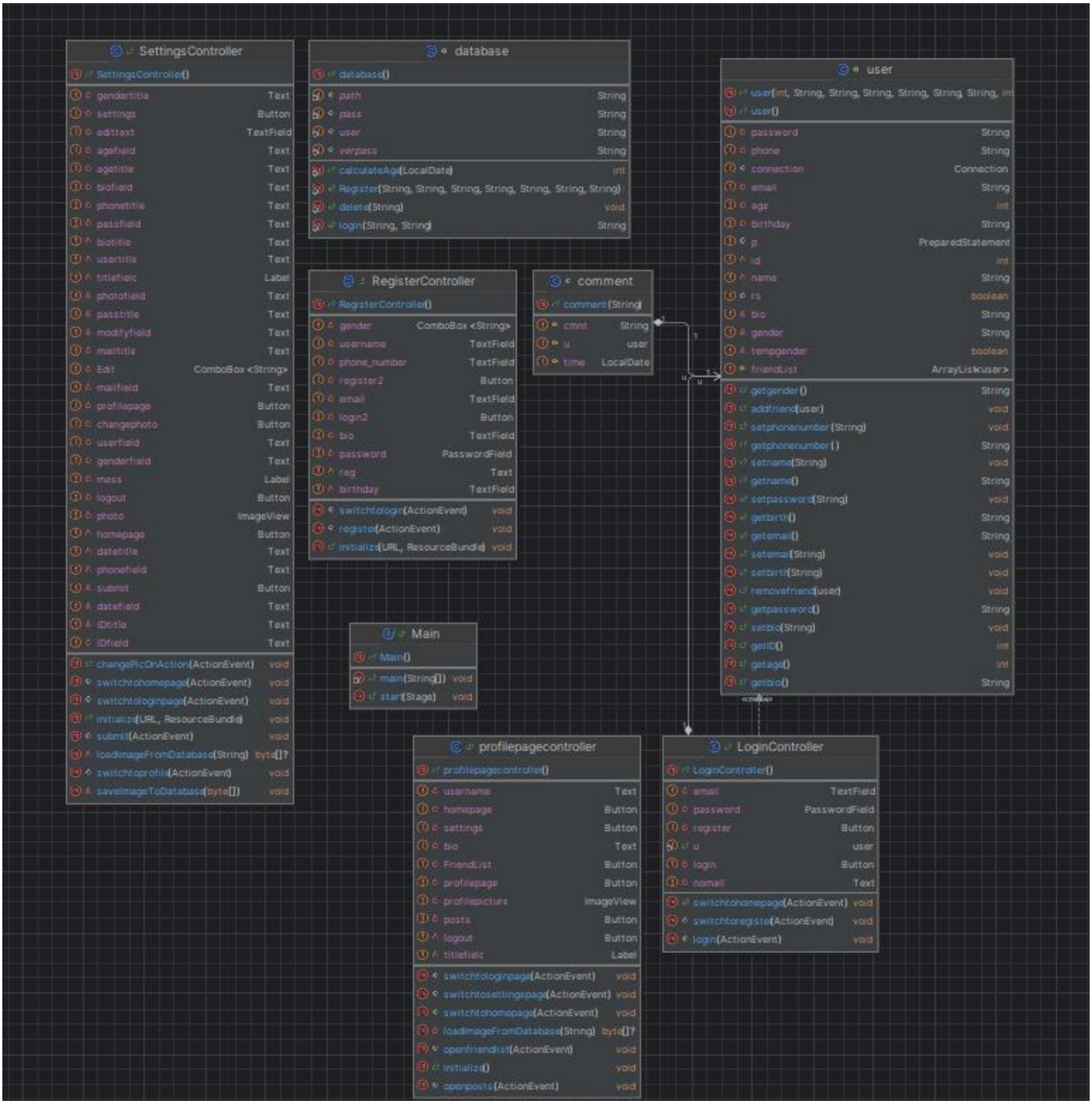


Figure 9:class diagram

Testing scenarios and results

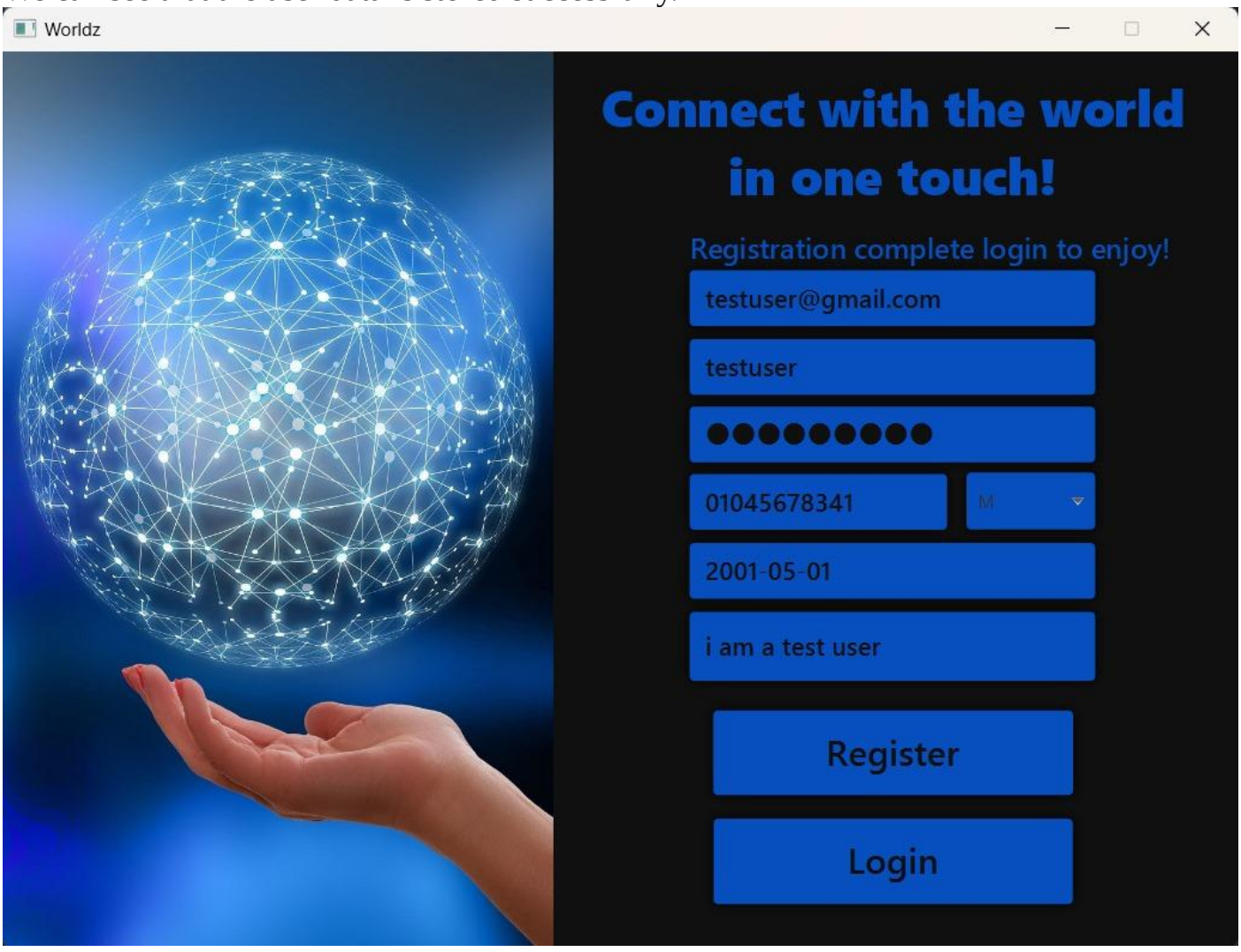
1. Test Scenario 1: Sign up for new accounts to verify the change in the database.

2. test Scenario 2: trying to sign up for an existing account.

Test case ID	Test case description	Test data	Expected result
1	Saving new accounts to the database.	Information is saved to the database successfully with the expect output.	Data is stored in the database.
2	Trying to save a new account with the information of the existing one.	Didn't overwrite, nor delete old data or lead to errors.	Printing error duplicated entry.

Test scenario 1:

We can see that the user data is stored successfully.



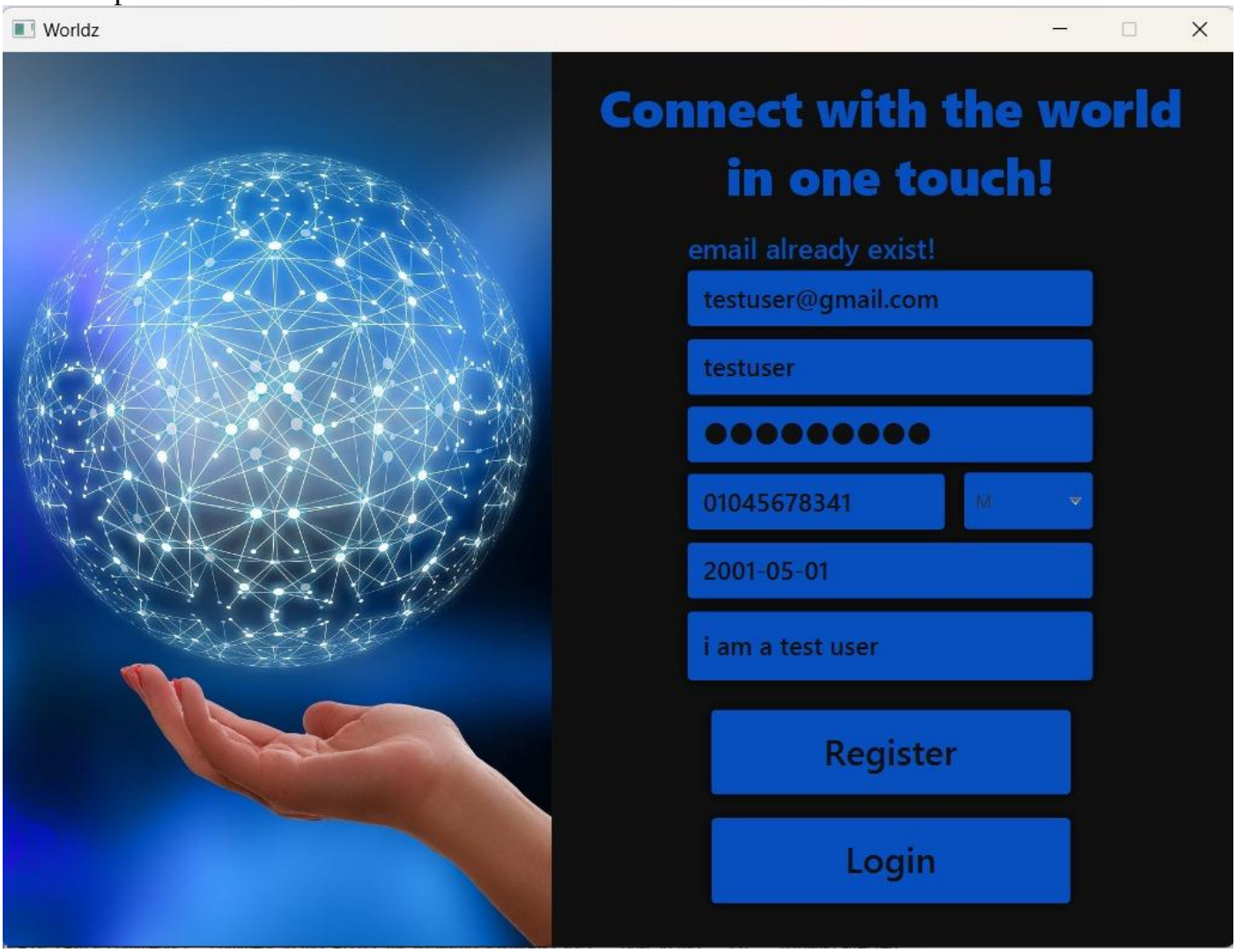
7	testuser@gmail.com	dyduw9und	testuser	01045678341	M	2001-05-01	23	i am a test user	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

id	email	password	username	phone_number	gender	birthday	age	bio	profilepic
1	ahmed@gmail.com	d30ds728	ahmed	01060442281	M	2005-12-19	18	alo alo	BLOB
2	emailismail@gmail.com	ezz	ezz	01068488157	M	2004-12-01	19	abyaad	NULL
3	ezzismail@gmail.com	E1234567	ezz	01068488157	M	2004-12-04	19		NULL
4	ezzismail1@gmail.com	E1234567	ezz	01068488157	M	2004-12-04	19		NULL
5	ezzismail2@gmail.com	E1234567	ezz	01068488157	M	2004-12-04	19		NULL
6	ezzismail3@gmail.com	E1234567	ezz	01068488157	M	2004-12-04	19		NULL
7	testuser@gmail.com	dyduw9und	testuser	01045678341	M	2001-05-01	23	i am a test user	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

We can see here that when the user is going to sign up for the first time, the information entered by him is stored in a database and the database added his information so he will not need to enter the information when he logins next time.

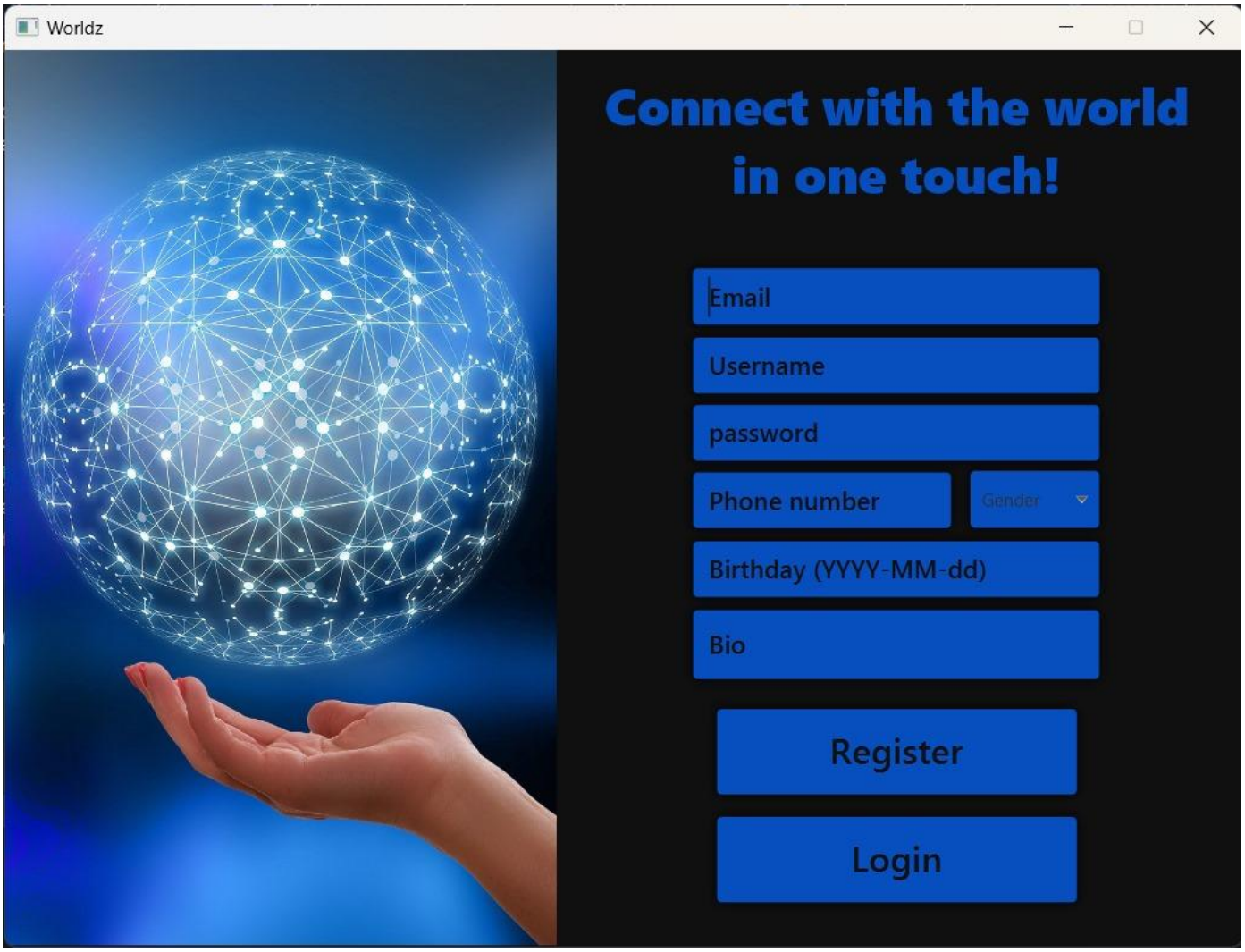
Test scenario 2:

No unexpected errors occurred.

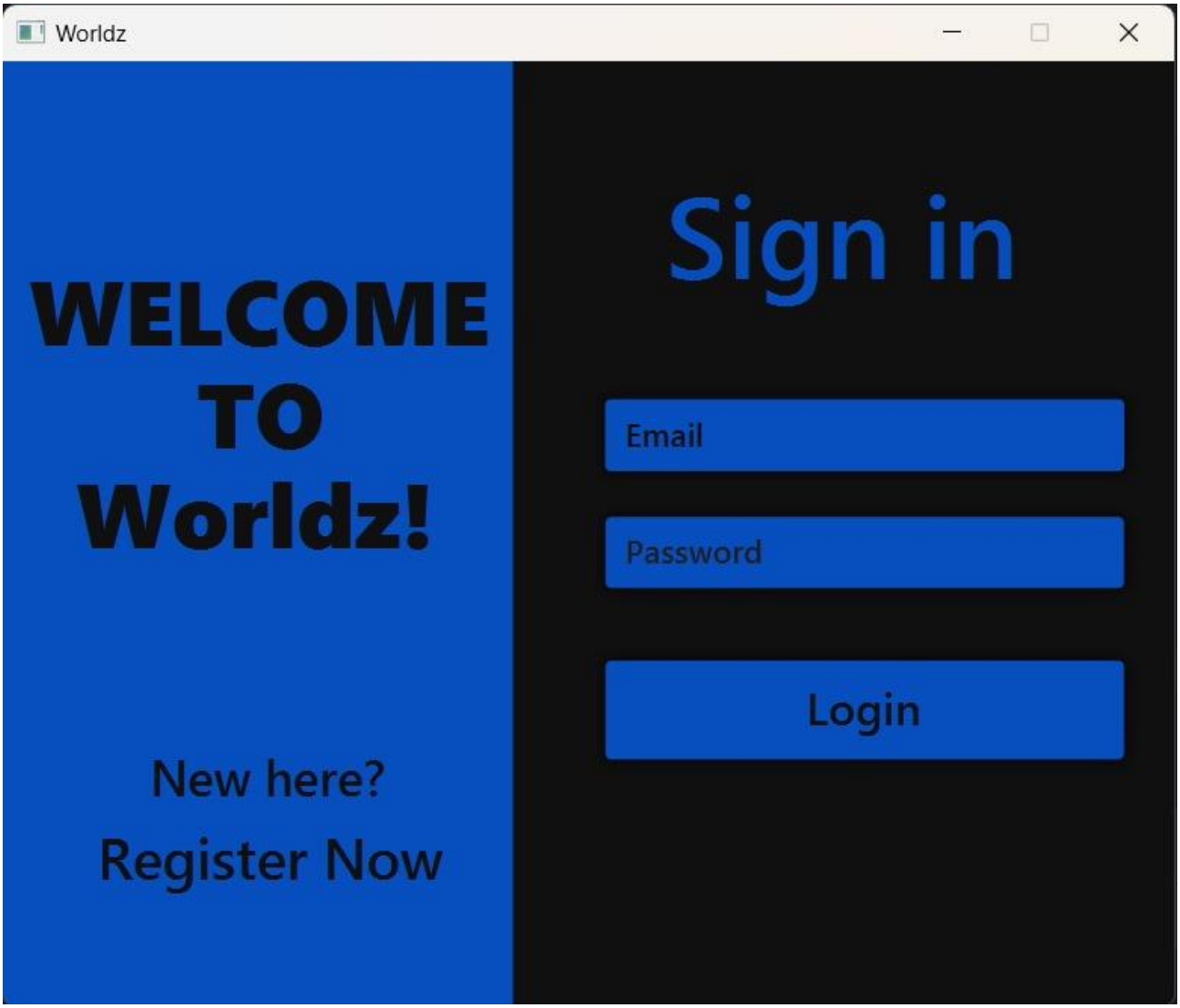


As we see in the previous photo, when the user tried to login again the message with email already exist is displayed to him and no unexpected errors occurred because the entered information is already previously stored in the database, so he doesn't need to sign up again.

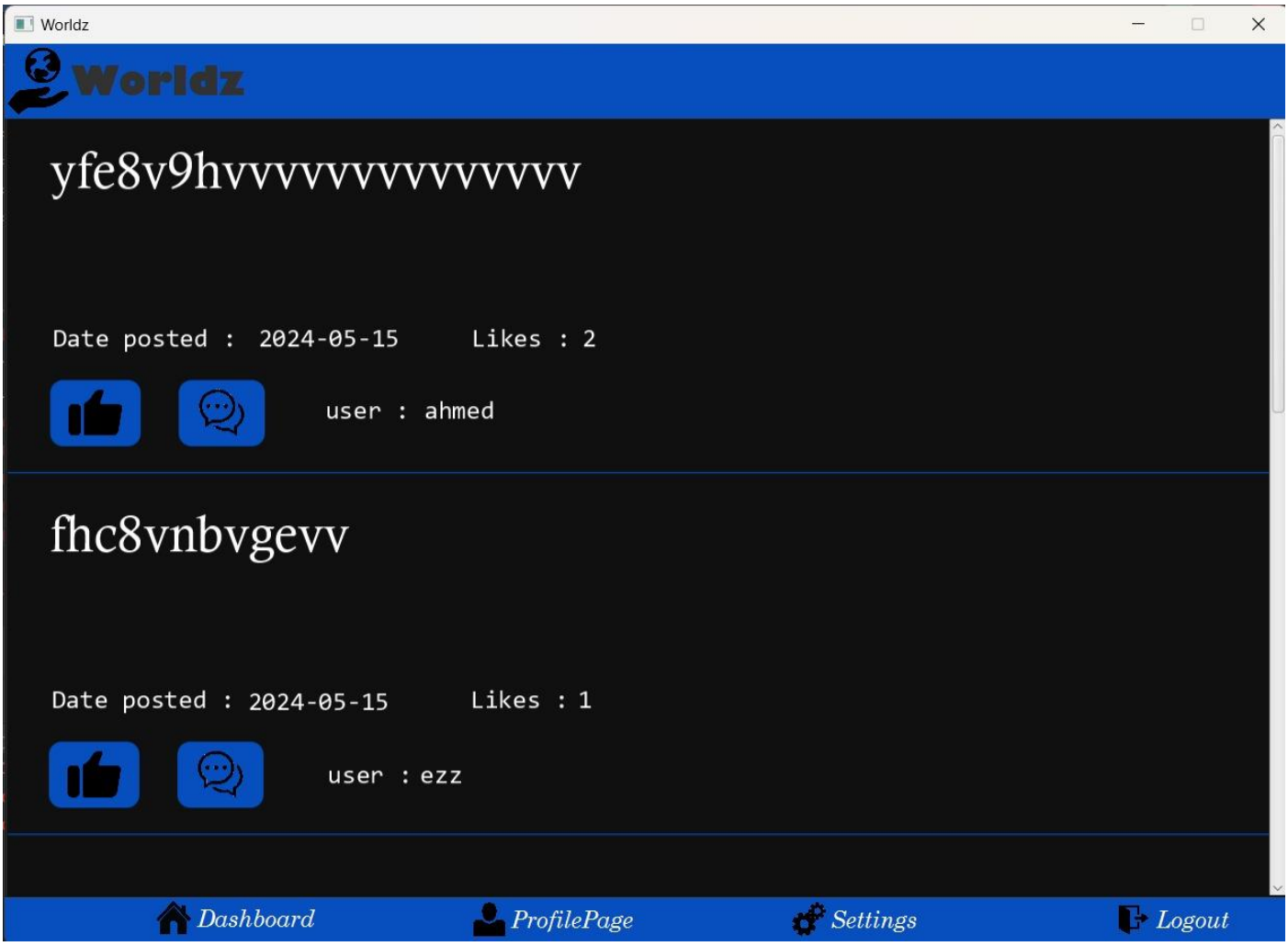
End-user guide
1.signup page: you need to fill the required information for making new account in application.



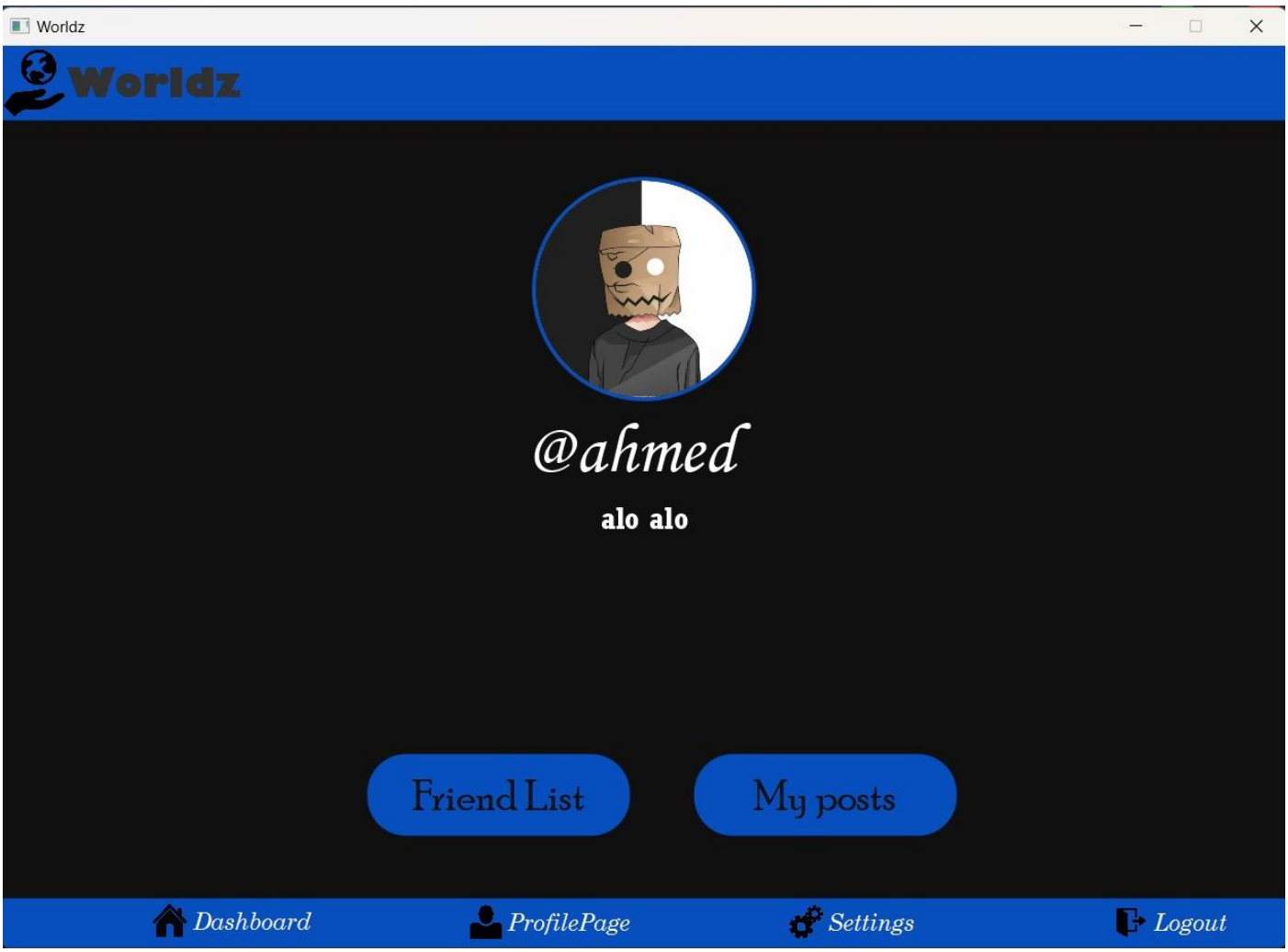
2.login page: enter the email address, the password you registered with.



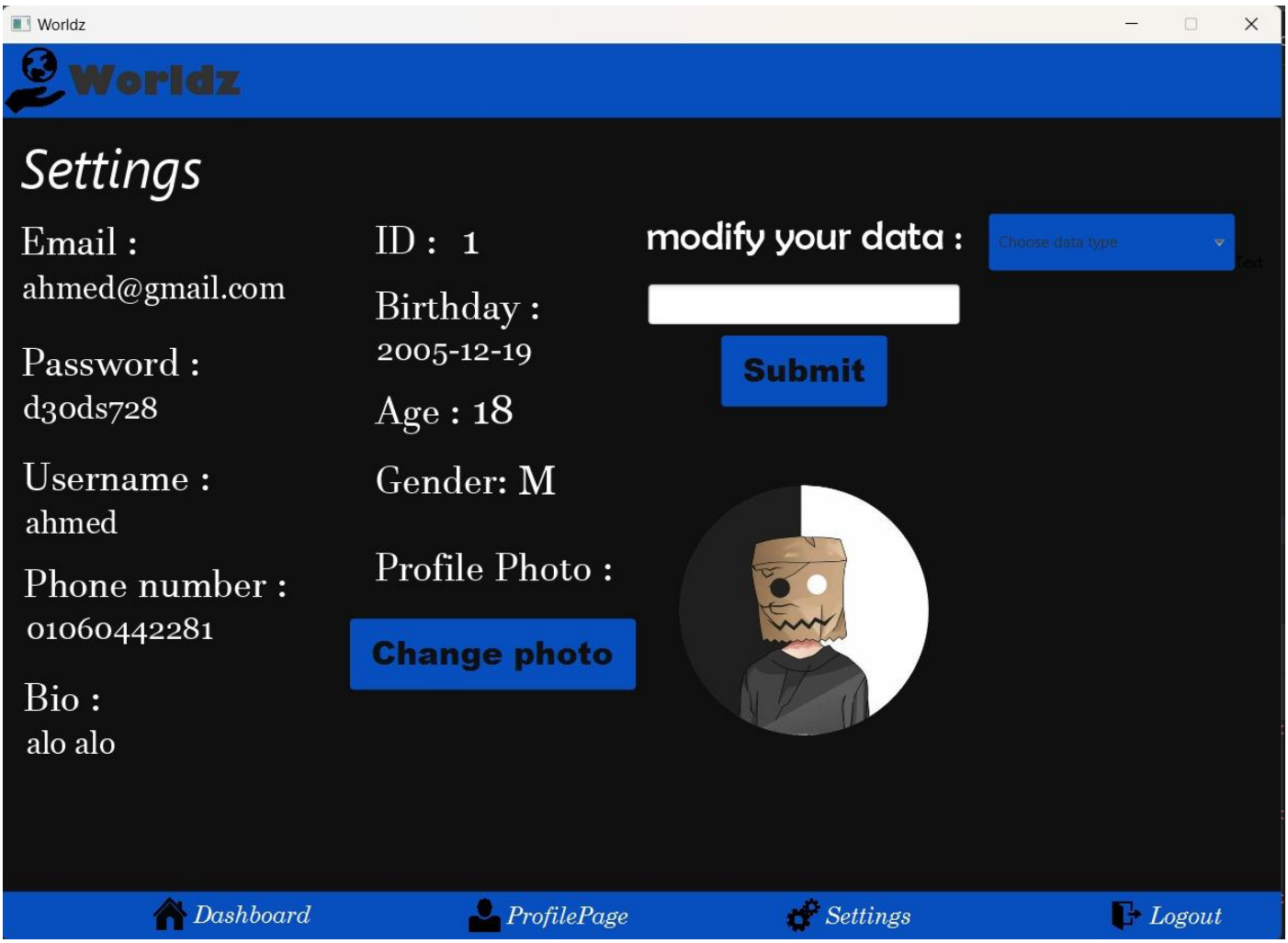
3.the home page where you can view others posts.



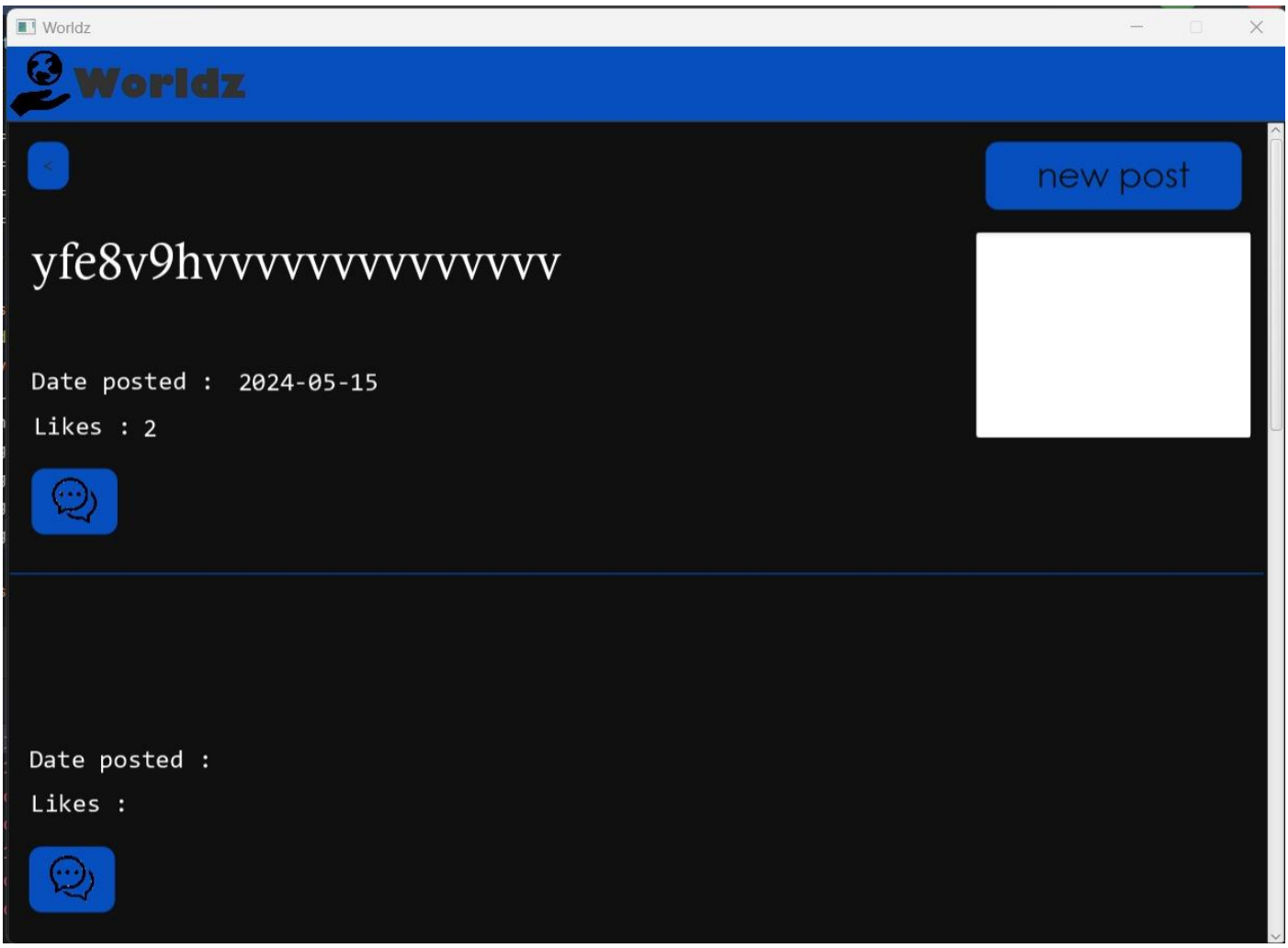
4.profile page.



5.settings scene where you can edit your information.



6.viewmy posts.



Conclusion

To sum up our social project: We've made a cool application where people can share stuff and make and interact with friends online.
We've built it using Java classes, which helped us organize everything neatly and make sure it runs smoothly. Plus, we've stored all the important stuff securely in a database.
With JavaFX and Java classes, we've made it easy for users to navigate and enjoy the platform. Whether they're posting updates or exploring new interests.
And with a solid database foundation, we've ensured that user data is accessible whenever needed.
In conclusion, our social media platform stands as a testament to the power of JavaFX, Java classes, and database technology in creating a dynamic and user-friendly online community. We're excited to continue refining and improving our platform to provide the best possible experience for all users.

References

[1] Daniel Liang, Introduction to Java Programming, 10th Edition, Pearson, 2014.