RideBuddies

-Semester project documentation-

# Overview

RideBuddies is an app dedicated to those who are passionate about a sport but, due to different circumstances, didn’t have the opportunity to get in touch with the other people practicing that sport in a specific area.

Maybe they have just moved to a new town or they just went on a trip and want to check out local spots and meet people that share the same passions and interests.

This app offers its users the solution to these problems by gathering everyone on a platform where they can showcase their interests, level of experience and so on, while combining the ability to chat and meet with other fellow enthusiasts of a sport.

# Functionalities

### Create account/Register

The user is able to create their account by simply typiyng in the required information. The „Register” page has fields for:

-First Name;

-Last Name;

-Username;

-Password and Confirm password;

-Email;

-Age;

To successfully register an account, users must meet these requirements:

1. **Password Confirmation**: The passwords entered in the "Password" and "Confirm Password" fields must match.
2. **Minimum Age**: Users must be at least 18 years old to create an account.
3. **Email Validation**: The email address provided must follow a valid email format.

After inserting all the neccesarry data, a message to valide the Registration will appear, and the „Go to Login” button will pop up and redirect the user to the Login page.

### Log in

The user must insert the username and password and if they are correct, they will be redirected to the Main Page. If the credentials are incorrect, an error notification will appear, prompting the user to enter valid login information. The error message will persist until the correct details are provided.

### Friend search

The user can search for new friends by their username. A list of other users will appear, if they have that username or if it encompases that name. After the search is finished, they can select the preffered user and add them to their friend list.

### Edit account/save/delete

On the Profile page, users can manage their account details.

 **Edit Account**: Users can update certain profile information (first name, last name and age).

 **Save Changes**: After editing, users can save their changes to update their profile.

 **Delete Account**: If desired, users can permanently delete their account

# Structure and workflow

### User class

The user class contains information about the user: First Name, Last Name, password, age, username and email;

Methods used:

1. Getters and setter used for their access and modifying;
2. Validate the age and email format logic;

### Logged user

The LoggedUser class manages the currently authenticated user session. It make sure that only one user is logged and makes that users’ information accesible throughout the application.

### Database connection

The DatabaseConnection class is responsible for managing connections to the application's database.

* The method getConnection() is responsible for creating and returning a connection to the specified database.
* If the connection is successful, the method returns the Connection object. If unsuccessful, the method still returns databaseLink (which will be null if the connection fails).
* If an exception occurs the stack trace and error message is printed.

### Controllers for UI

 **Event Handling**: Controllers respond to user actions such as button clicks or text input by invoking backend methods to process these actions.

 **Data Binding**: They populate UI elements with backend data and update backend data when users modify the interface.

 **Navigation**: Controllers manage screen transitions, such as moving from the login page to the main page, often using FXMLLoader to load views.

 **Validation and Feedback**: They validate user inputs and provide feedback through error messages and notifications.

# Future development

As the current status of the project requires updates, in the future, some of the following functionalities will be added:

### Status updates

In order to boost the interaction within users, status updates will be implemented in order for them to express their thoughts and feelings about their progress in their activities and maybe ask for advice and plan future activities.

### Google Maps API integration

The app is focused on finding users in a specific area so Google Maps will be integrated to yield the exact location of an user and also provide the ability to adjust the area in which other users can be found based on their location.

### Chat integration

Users will be able to easily communicate with each other using an in-app chat

# 