

Homework 2

ActivityTracker App

Software Technical Document

Student: **Şerban Iustinian-Bogdan**

Subject: **Industrial Informatics**

Contents

Figures.....	2
1. Introduction.....	3
2. Components and Data Flow	3
3. Data Format and Storage.....	3
4. Classes	4
4.1. UserForm	4
4.2. User.....	4
4.3. Session	5
4.4. PasswordHasher	5
5. Extras	5

Figures

Figure 1 - Data Flow.....	3
Figure 2 - Database Structure.....	3
Figure 3 - UserForm UI User Connected	4
Figure 4 - UserForm Class.....	4
Figure 5 - UserForm UI User Disconnected	4
Figure 6 - User Class	4
Figure 7 - Session Class.....	5
Figure 8 - PasswordHasher	5

1. Introduction

For Homework2, the project builds upon the previously developed *ActivityTracker App* from Homework1, updating it with several improvements to serve as the Windows Forms client. This document focuses solely on the new components and modifications introduced in Homework2. The original architecture and design of the application is presented in Homework1.

2. Components and Data Flow

The system consists of the previously implemented ActivityTracker application, which serves as the client, an ASMX web service that handles communication and logic, and a SQL database for persistent data storage.

Figure 1 illustrates the types of requests made by the client application to the ASMX server.

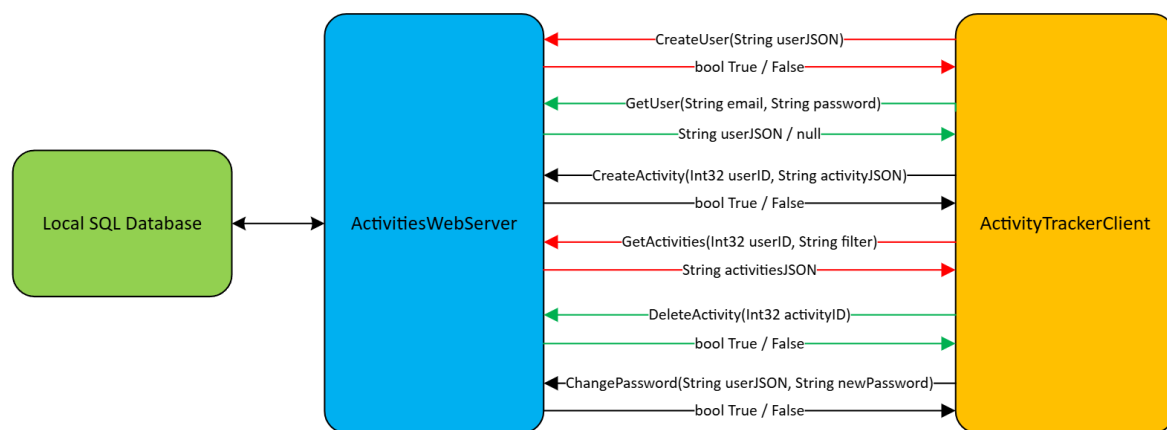


Figure 1 - Data Flow

3. Data Format and Storage

The activity data is now stored in the local SQL database, with its structure illustrated in Figure 2. To enhance security, user passwords are encoded using the PBKDF2 algorithm. This encoding process is handled on the server side, ensuring secure storage and authentication. In addition, the current session information is stored in a .txt file located in the *AppData/Local/ActivityTracker* directory. If either the folder or the file does not exist, the application automatically creates them at runtime to ensure consistent session tracking.

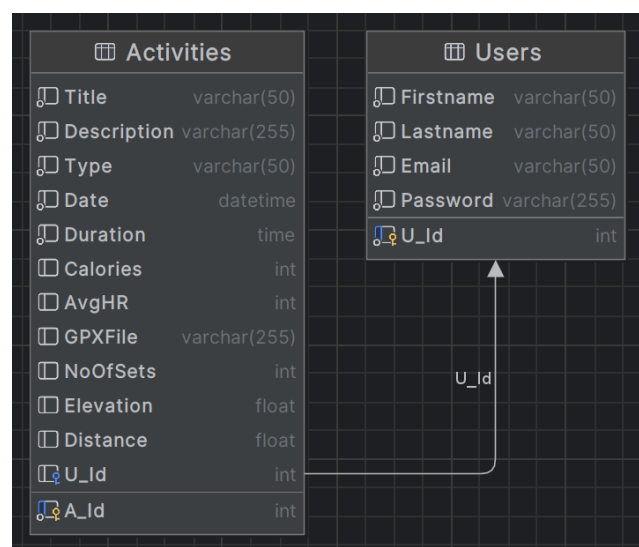


Figure 2 - Database Structure

4. Classes

4.1. UserForm

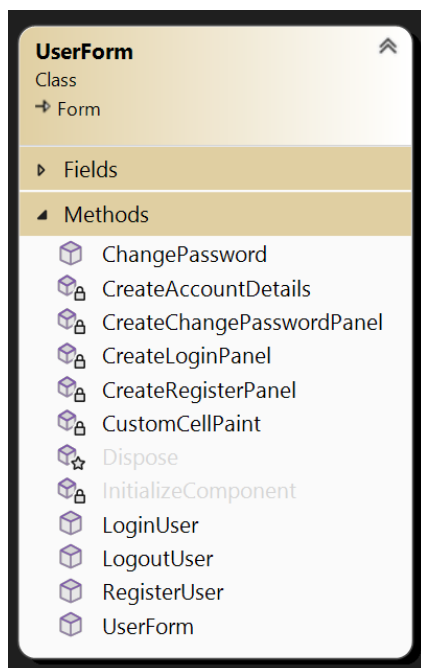


Figure 4 - UserForm Class

The UserForm allows users to log in or register if they do not already have an account. Once authenticated, users can view their account details, log out or update their password.

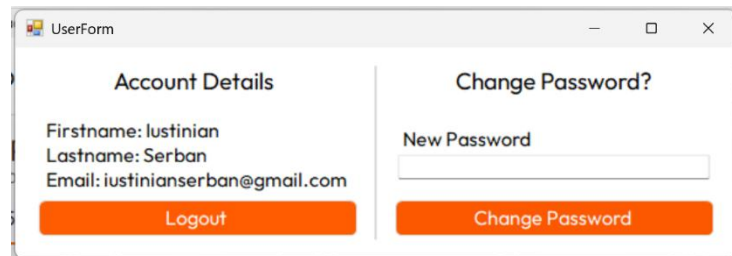


Figure 3 - UserForm UI User Connected

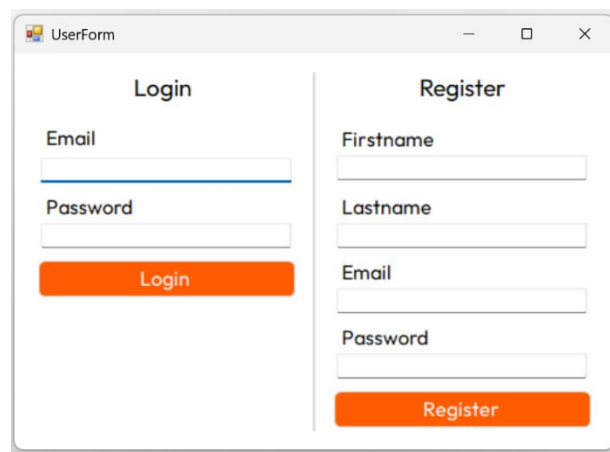


Figure 5 - UserForm UI User Disconnected

4.2. User

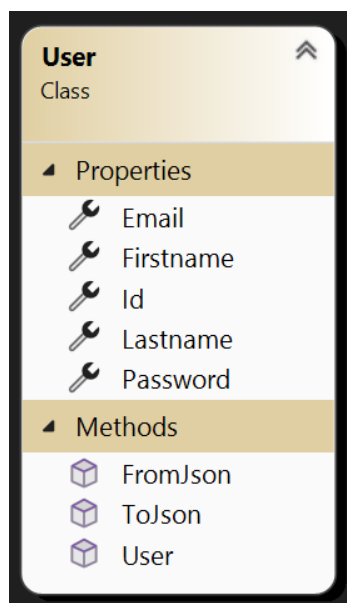


Figure 6 - User Class

The User class encapsulates user-related information, including the following properties: id, firstname, lastname, email and password.

In addition, the class includes methods for JSON serialization, enabling user data to be formatted and transmitted to the server efficiently.

It is implemented on both the client and the server.

4.3. Session

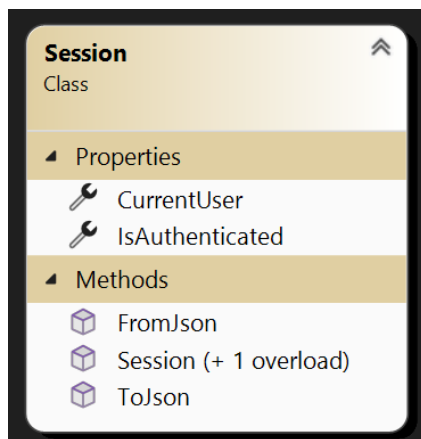


Figure 7 - Session Class

The Session class was introduced to manage the current session state within the client application. It keeps track of whether a user is currently logged in and stores the associated user details.

To mimic the behavior of a cookie, the current session information is stored in the *AppData/Local/ActivityTracker* folder. This allows the user to remain logged in between application restarts, eliminating the need to re-enter credentials each time the client app is launched.

4.4. PasswordHasher

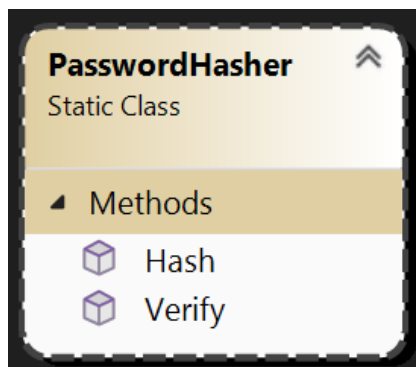


Figure 8 - PasswordHasher

The PasswordHasher is a server-side helper class responsible for encoding user passwords using the PBKDF2 algorithm. It also provides functionality to verify whether the password entered during login matches the securely stored hash.

5. Extras

In addition to this documentation, the archive includes the complete application source code as well as a CSV file populated with various activities to facilitate testing of the application's functionality.