The General UML Version 1. In version, panes are implemented using the functional approach. Look for the next page for properties and methods.

<<Interfaces>>

Loggable

authenticates



<<interface>>

Observer

<<interface>>

Observable

Attempt

MainApp

AuthenticationService

Wordle

UserAccount

UML Version 2. Here Panes are implemented as objects.

Diagram

Description automatically generated

UserAccount

|  |  |
| --- | --- |
| -data: Wordle | Keeps the last state of the wordle game the user played. |
| -password: String |  |
| -username: String |  |
|  |  |

|  |  |
| --- | --- |
| +UserAccount(string username, string password) | Create an account with the given password and username. |
| +getPassword(): string |  |
| +getUserName(): string |  |
| +getUserData(): Wordle |  |
|  |  |
|  |  |

You are free to call your helper methods as you want. No method should be over 50 lines of code.

**Storing user information:**  
We will store the user information in the HashMap<String, UserAccount>, where the key is the username. The hash is serialized and stored in usersDB.

Wordle

This is a grid that contains information about the attempts that user makes and grid state.

|  |  |
| --- | --- |
| -attempts: ArrayList<Attempt> | The attempts that user has made. |
| -attemptsLeft: int | How many attempts are left. |
| -gameOver: boolean | True if the game is over. |
| -win: boolean |  |
| -target: string | The word to be searched for. |
| -targetLength: int |  |

|  |  |
| --- | --- |
| +checkAttempt(string attempt): true |  |
| +setWin(boolean won): void |  |
| +isWin(): boolean |  |
| +setTarget (): string | Chooses a random target. |
| +getAttempts(): ArrayList<Attempt> |  |
| +isGameOver(): boolean |  |
| +getTarget(): string |  |
| +getTargetLength(): int |  |

You are free to call your helper methods as you want. No method should be over 50 lines of code.

Attempt

Each attempt represents the attempt that user has made.

|  |  |
| --- | --- |
| -data: string |  |
| -correctLetters: char[] |  |
|  |  |

|  |  |
| --- | --- |
| +Attempt() |  |
| +Attempt(string attempt) |  |
| +setAttempt(string attempt) |  |
| +getAttempt(): string |  |
| +getCorrectLetters(string target): char[] |  |
|  |  |
|  |  |
|  |  |
|  |  |

You are free to call your helper methods as you want. No method should be over 50 lines of code.

AuthenticationService

|  |  |
| --- | --- |
| -currentUser: string |  |
| -usersData: HashMap<String, UserAccount> |  |
| -DatabaseString | Name of the file that contains data. |

|  |  |
| --- | --- |
| +createAccount(UserAccount): int | 0 if the account already exists. 1 if the account is created. The user is not logged in automatically. |
| +logIn(String newUsername, String newPassword  ): int | 0 if someone is already logged in. 1 if the use does not exist. 2 passwords do not match. 3 if successfully logged in. |
| +logOutUser(): boolean | 0 if no one is logged. 1 if the user is logged out successfully. Their data is saved automatically. |

MainApp

This is the main GUI that coordinates that app activities.

|  |  |
| --- | --- |
| -funcPane: StackPane | GUI for attempts, guessed letters. |
| -gamePane: StackPane | GUI for showing boggle. |
| -keyboardPane: StackPane | GUI for the keyboard. |
| -gameOverPane: StackPane | GUI for win/lose. |
|  |  |
|  |  |

|  |  |
| --- | --- |
| +update() | Updates all observers. |
| +setFuncPane() | Sets up the functionality pane, where we will show attempts and letters that were guessed. |
| +setGamePane() | Sets up the game pane based on the current state. |
| +setKeyboardPane() | Shows the keyboard for typing in. |
| +setGameOverPane() | Shows the right text (success/error) after the game is over. |
| +loginUser() |  |
| +loginExistingUser() |  |
| +createNewAccount() |  |
|  |  |

You are free to call your helper methods as you want. No method should be over 50 lines of code.

Pane Management Approach:

Since we don’t really have repeated GUI elements, we may use procedures to create the appropriate panes and to manage them.