

Class: "CredentialsDatabase"

- Fields:

- HashMap <String, ArrayList<Object>> userData
Key would be the username and values would be an ArrayList of objects where the list contains email, password, ranking, connect4Stats, ticTacToeStats, checkerStats, recent matches played?

- Constructor → public CredentialsDatabase()

- We will create a new HashMap
- Then we will loadDatabase → invoke the method (so basically we will be storing the database in a database.txt and whenever we start a new session of login page (creating a new instance of loginPage in our main function) we will be creating a database instance too which will load the previous data/information about players from a database.txt file into the instance of the database and pass it on to the loginPage instance as a parameter)

- Methods:

- usernameLookup(username) : boolean
- addCredentials (username, email, password) : boolean
- deleteCredentials (username, email, password) : boolean
- findPlayerByUsername(username) : Player
(It will look up the username in the hashmap and its details, create a new player object with that username and then initialize all the fields of the player like username, ranking, connect4Stats, ticTacToeStats and checkedStats. Then it is gonna return the user object with those fields)
- updateUsername(old username, new username)
- updateEmail(username, email)
- updatePassword(username, password)
- saveDatabase : void
(It will save all the information to the database.txt file before every logout session)
- loadDatabase : void
(It will load all the information from the database.txt file into the hashmap everytime main is run and main creates a new instance of Database class)

Class: "Player"

- Fields:

- Username / playerId : String
- Email : String
- Password : String
- Ranking : Rank (Enum)
- connect4Stats : Connect4Stats
- ticTacToeStats : TicTacToeStats

- checkerStats : CheckerStats
- **Constructor** → **public Player (username)**
- **Methods :**
 - getStats(gameType : String) : GeneralStats
 - All the other getter and setter methods for the fields

Class: “LoginPage”

//instance of LoginPage will be created by our main function with database as a parameter

- **ENUM CLASS** : State (Enum)
 - UsernameTaken
 - EmailFormatWrong
 - VerificationCodeWrong
 - Success
- **Fields :**
 - database : CredentialsDatabase
- **Constructor** → LoginPage(CredentialsDatabase : database)
- **Methods:**
 - login(username, password) : HomePage
 - *(We will basically be doing a lookup in the database object like Player player = database.findPlayerByUsername(username) and then we will check if the player is not null (the username exists in the database and the password entered is the same as player.getPassword and everything is correct, we will create a new Homepage with the player and database as its parameter and return that, so basically the homepage will be associated to that player, or if not then we will return null. So the existence of homePage will determine if the login was successful or not. Something like:*
HomePage home = loginPage.login("player1", "password123");
if (home != null) { System.out.println("Login successful!"); }
 - register (username, password, email) : ENUM
(Just check if entered code is not empty for verification purpose)
 - forgot password (username) : boolean