

```
class LoginWindow {
• LoginWindow(parent : QWidget* = nullptr)
• ~LoginWindow()
• stackedWidget : QStackedWidget
• loginPage : QWidget
• usernameLoginEdit : QLineEdit
• passwordLoginEdit : QLineEdit
• loginButton : QPushButton
• signupPageButton : QPushButton
• playAsGuestButton : QPushButton
• signupPage : QWidget
• firstNameEdit : QLineEdit
• lastNameEdit : QLineEdit
• dobEdit : QDateEdit
• genderComboBox : QComboBox
• usernameSignupEdit : QLineEdit
• passwordSignupEdit : QLineEdit
• passwordStrengthLabel : QLabel
• browseImageButton : QPushButton
• signupButton : QPushButton
• backToLoginButton : QPushButton
• setupLoginPage() : void
• setupSignupPage() : void
• validateSignupForm() : bool
• startGame(user : User) : void
• onLoginClicked() : void
• onSignupClicked() : void
• onPlayAsGuestClicked() : void
• validatePassword(password : QString) : void }
```

Collects and validates the user data

Evan Gallagher

```
class User {
• firstName : QString
• lastName : QString
• dateOfBirth : QDate
• gender : QString
• username : QString
• password : QString
• profilePicture : QString
• isGuest : bool
• scoreHistory : QVector<GameScore>
• getFirstName() : QString
• getLastName() : QString
• getFullName() : QString
• getDateOfBirth() : QDate
• getGender() : QString
• getUsername() : QString
• getPassword() : QString
• getIsGuest() : bool
• getScoreHistory() : QVector<GameScore>
• getBestScore() : int
• isBirthday() : bool
• setFirstName(value: QString)
• setLastName(value: QString)
• setDateOfBirth(value: QDate)
• setGender(value: QString)
• setUsername(value: QString)
• setPassword(value: QString)
• setProfilePicture(value: QString)
• setIsGuest(value: bool)
• addScore(score : int) : void }
```

User database helps add information to the user class.

Evan Gallagher

```
class UserDatabase {
• m_users : QHash<QString, User>
• m_dbFilePath : QString
• m_instance : UserDatabase*
• instance() : UserDatabase*
• addUser(user : User) : bool
• authenticateUser(username : QString, password : QString, user : User) : bool
• updateUser(user : User) : bool
• userExists(username : QString) : bool
• getGlobalBestScore() : int
• loadDatabase() : void
• saveDatabase() : void
• UserDatabase(parent : QObject* = nullptr)
• ~UserDatabase() }
```

```
class GameScore {
• score : int
• playedAt : QDateTime
• GameScore()
• GameScore(score : int, playedAt : QDateTime) }
```

Game window will use user information for to be displayed in the game window.

Evan Gallagher

```
class GameWindow {
• GameWindow(user : User, parent : QWidget* = nullptr)
• ~GameWindow()
• m_user : User
• gameTimer : QTimer*
• score : int
• gameRunning : bool
• ball : QLabel*
• paddle : QLabel*
• blocks : QVector<QLabel*>
• ballSpeedX : int
• ballSpeedY : int
• paddlePos : QPoint
• stackedWidget : QStackedWidget*
• menuPage : QWidget*
• gamePage : QWidget*
• scoreHistoryPage : QWidget*
• welcomeLabel : QLabel*
• birthdayLabel : QLabel*
• userInfoLabel : QLabel*
• currentDateLabel : QLabel*
• scoreLabel : QLabel*
• playButton : QPushButton*
• scoreHistoryButton : QPushButton*
• logoutButton : QPushButton*
• scoreHistoryTable : QTableWidgetItem*
• backToMenuButton : QPushButton*
• setupMenuPage() : void
• setupGamePage() : void
• setupScoreHistoryPage() : void
• updateScoreHistory() : void
• createBlocks() : void
• moveBall() : void
• checkCollisions() : void
• showBirthdayGreeting() : void
• updateUserInfo() : void
• closeEvent(event : QCloseEvent*) : void
• keyPressEvent(event : QKeyEvent*) : void
• startGame() : void
• gameLoop() : void
• endGame() : void
• showMainMenu() : void
• showScoreHistory() : void
• logoutUser() : void }
```

Game window might update the user data base with scores.

Evan Gallagher