

PasswordHash

```
+PBKDF2_ALGORITHM: string
+SALT_BYTE_SIZE: int
+HASH_BYTE_SIZE: int
+PBKDF2_ITERATIONS: int
+ITERATION_INDEX: int
+SALT_INDEX: int
+PBKDF2_INDEX: int

+createHash(password: string): string
+createHash(password: char[]) string
+validatePassword(password: string, correctHash: string): boolean
+validatePassword(password: char[], correctHash: string): boolean
+slowEquals(a: byte[], b: byte[]): boolean
+pbkdf2(password: char[], salt: byte[], iterations: int, bytes: int): byte[]
+fromHex(hex: string): byte[]
+toHex(array: byte[]): string
```

MainGUIController

```
controller: ScreensController

+initialize(url: URL, rg: ResourceBundle): void
+setScreenParent(screenParent: ScreensController): void
+goToLoginScreen(event: ActionEvent): void
```

LoginGUIController

```
controller: ScreensController
dbMgr: DBMgr

+initialize(url: URL, rb: ResourceBundle): void
+setScreenParent(screenParent: ScreensController): void
+registerNewUser(event: ActionEvent): void
+clearRegisterForm(event: ActionEvent): void
+login(event: ActionEvent): void
+clearLoginForm(event: ActionEvent): void
+goToHomeScreen(event: ActionEvent): void
```

TablesGUIController

```
controller: ScreensController

+initialize(url: URL, rb: ResourceBundle): void
+addNewHoldemTable(tableName: string, blind: string, maxBuyIn: string, capacity: string): void
+addNewFiveCardDrawTable(tableName: string, ante: string, maxBet: string, capacity: string): void
+addNewBlackjackTable(tableName: string, minBet: string, maxBet: string, capacity: string): void
+createNewTableEntry(game: Game, name: string, info1: string, info2: string, capacity: string): AnchorPane
+setScreenParent(screenParent: ScreensController): void
+goToLoginScreen(event: ActionEvent): void
+showHoldemTables(event: ActionEvent): void
+showFiveCardDrawTables(event: ActionEvent): void
+showBlackjackTables(event: ActionEvent): void
+goToAccountSettingsScreen(event: ActionEvent): void
```

ManageAccountGUIController

```
controller: ScreensController
dbMgr: DBMgr

+initialize(url: URL, rb: ResourceBundle): void
+setScreenParent(screenParent: ScreensController): void
+showChangePassword(event: ActionEvent): void
+showChangeDisplayName(event: ActionEvent): void
+showChangeEmail(event: ActionEvent): void
+goToTablesScreen(event: ActionEvent): void
+goToLoginScreen(event: ActionEvent): void
+changePassword(event: ActionEvent): void
+clearPasswordForm(event: ActionEvent): void
+changeDisplayName(event: ActionEvent): void
+clearDisplayNameForm(event: ActionEvent): void
+changeEmail(event: ActionEvent): void
+clearEmailForm(event: ActionEvent): void
```

DesktopCardGameGUI

```
+homeScreen: string
+homeScreenFile: string
+loginScreen: string
+loginScreenFile: string
+tablesScreen: string
+tablesScreenFile: string
+accountScreen: string
+accountScreenFile: string
+holdemScreen: string
+holdemScreenFile: string
+fiveCardDrawScreen: string
+fiveCardDrawScreenFile: string
+blackjackScreen: string
+blackjackScreenFile: string
+statisticsScreen: string
+statisticsScreenFile: string
+start(stage: Stage): void
+main(args: string[]): void
```

ScreensController

```
+screens: HashMap<String, Node>
+ScreensController()
+addScreen(name: string, screen: Node): void
+getScreen(name: string): Node
+loadScreen(name: string, resource: string): boolean
+setScreen(name: string): boolean
+unloadScreen(name: string): boolean
```

UserStatisticsGUIController

```
controller: ScreensController

+initialize(url: URL, rb: ResourceBundle): void
+setScreenParent(screenParent: ScreensController): void
+showOverallStats(event: ActionEvent): void
+showFiveCardStats(event: ActionEvent): void
+showBlackjackStats(event: ActionEvent): void
+showFiveCardDrawStats(event: ActionEvent): void
```

FiveCardDrawGUIController

```
controller: ScreensController

+initialize(url: URL, rb: ResourceBundle): void
+setScreenParent(screenParent: ScreensController): void
+goToTablesScreen(event: ActionEvent): void
+goToLoginScreen(event: ActionEvent): void
+bet(event: ActionEvent): void
+call(event: ActionEvent): void
+raise(event: ActionEvent): void
+check(event: ActionEvent): void
+fold(event: ActionEvent): void
+sendMessage(event: ActionEvent): void
```

HoldEmGUIController

```
controller: ScreensController

+initialize(url: URL, rb: ResourceBundle): void
+setScreenParent(screenParent: ScreensController): void
+goToTablesScreen(event: ActionEvent): void
+goToLoginScreen(event: ActionEvent): void
+bet(event: ActionEvent): void
+call(event: ActionEvent): void
+raise(event: ActionEvent): void
+check(event: ActionEvent): void
+fold(event: ActionEvent): void
+sendMessage(event: ActionEvent): void
```

EuclreGUIController

```
controller: ScreensController

+initialize(url: URL, rb: ResourceBundle): void
+setScreenParent(screenParent: ScreensController): void
+goToTablesScreen(event: ActionEvent): void
+goToLoginScreen(event: ActionEvent): void
+showUserStatisticsScreen(): void
+pass(event: ActionEvent): void
+call(event: ActionEvent): void
+sendMessage(event: ActionEvent): void
```

BlackjackGUIController

```
controller: ScreensController

+initialize(url: URL, rb: ResourceBundle): void
+setScreenParent(screenParent: ScreensController): void
+goToTablesScreen(event: ActionEvent): void
+goToLoginScreen(event: ActionEvent): void
+insurance(event: ActionEvent): void
+split(event: ActionEvent): void
+doubleDown(event: ActionEvent): void
+hit(event: ActionEvent): void
+stand(event: ActionEvent): void
+outlander(event: ActionEvent): void
+sendMessage(event: ActionEvent): void
```

ChatClient

```
+HOST: string
+PORT: int
+chatServer: ChatServer
+registry: Registry
+chatBoxString: StringProperty

+ChatClient(chatBoxString: StringProperty)
+sendMessage(message: string): void
+chatEventSent(ev: ChatEvent): void
+closeConnection(): void
```

**<<interface>>
ControlledScreen**

```
+setScreenParent(screenPage: ScreensController): void
```