Functional Specification: Server Component

# Program Flow

## Program Flow: Registration

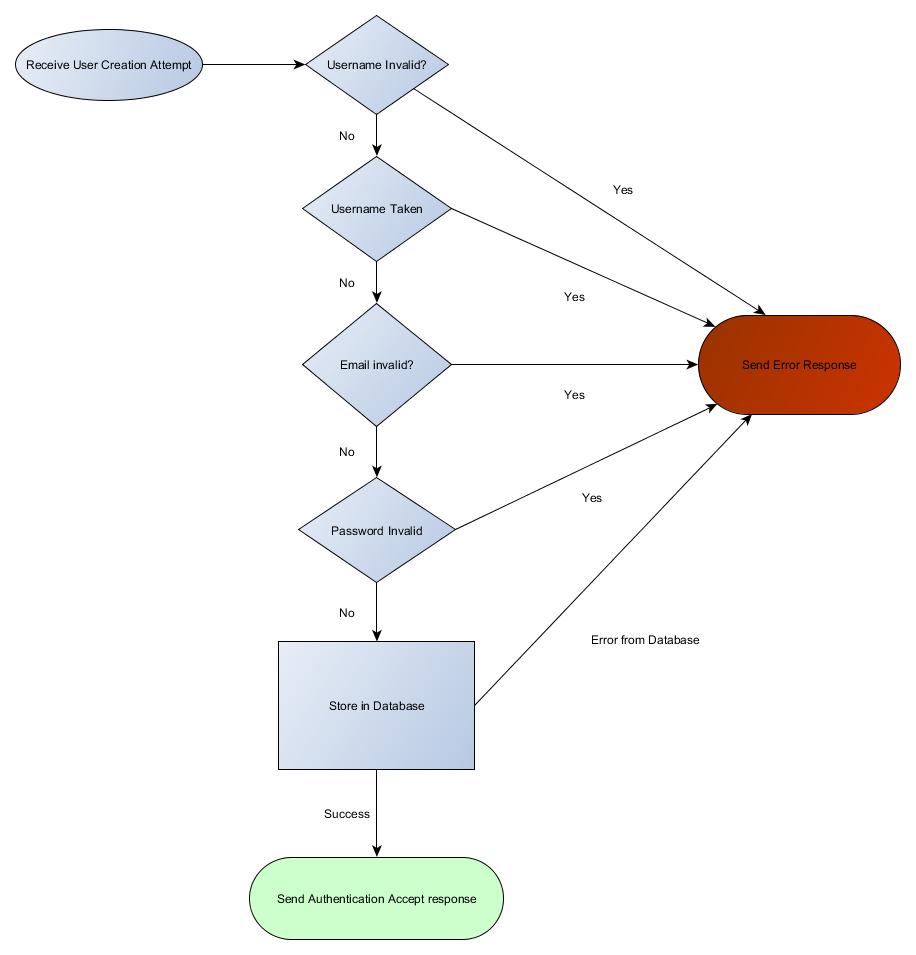


Figure : User Creation Attempt

Words coming soon.

## Program Flow: Login

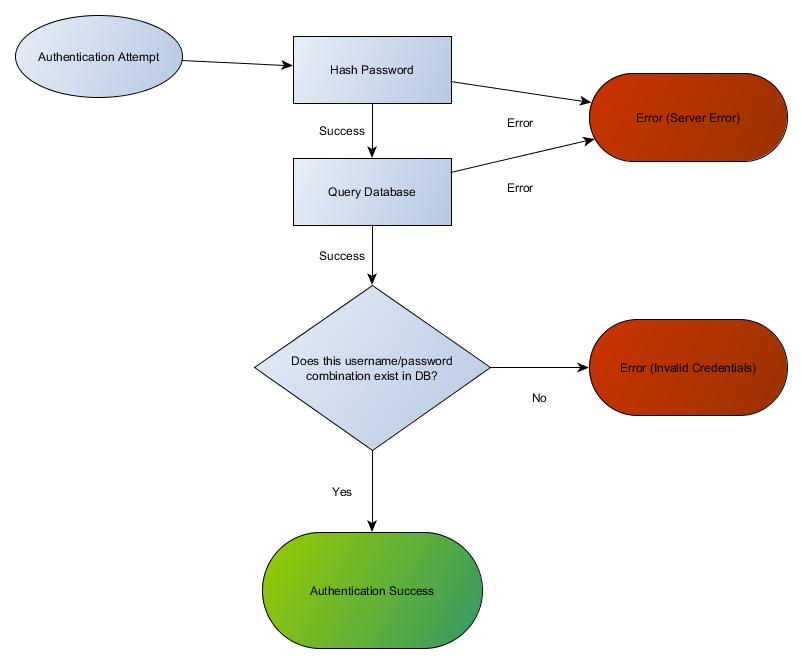


Figure : Authentication Attempt

Words coming soon.

# Components

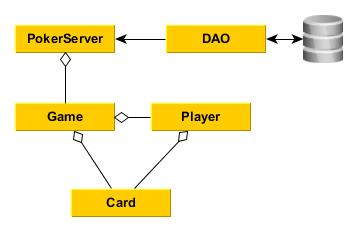


Figure : Class Diagram

## PokerServer

Instances of the class “PokerServer” shall represent a single concurrent server running. Each instance of the server shall listen on a specific port. It shall have the responsibility for managing connections and creating game objects as needed.

**Class Members:**

* ArrayList<Game> activeGames
* ArrayList<Player> connectedPlayers
* DAO Database

**Class Methods:**

* main
* route
* createGame

## Game

Instances of the class “Game” shall represent a single game currently in progress. It shall handle the turns of the game, including enforcing the business rules.

**Class Members**

* ArrayList<Player> players
* ArrayList<Card> tableCards
* Integer chipsInPot

**Class Methods:**

* attemptTurn
* dealCard

## Player

Instances of the class “Player” shall represent a single player, either in a game or in the lobby.

**Class Members**

* String username
* URL avatarURL
* Integer chipsRemaining
* ArrayList<Card> currentHand

**Class Methods:**

* addCardToHand
* subtractChips
* addChips
* getAvatar

## DAO

Instances of the class “DAO” shall represent individual databases in which data may be persisted. In practice, there should only be one instance per server.

**Class Members**

**Class Methods**

* updatePlayer
* checkCredentials
* addPlayer
* generateAuthToken