**Use Case Scenario**

1. **Use case**: Join existing games
   1. **Brief description**: Allow the user to join existing games. The user selects active game then selects his money stack and “sit” in the table, while waiting to the next round.
2. **Actors**: Primary – registered user, secondary – Texas Hold’em system.
3. **Precondition**: The user is logged in.
4. **Postcondition**: The user joined to existing game.
5. **Flow of Events**:
   1. **Basic Flow**:
      1. The user selects existing table from active games list.
      2. The system check if his bankroll valid for the selected game.
      3. The user selects a free seat.
      4. The user selects his money stack.
   2. **Alternate Flows**:
      1. BAD GAME SELECT  
         The user try to enter to game that the minimal bankroll is greater the he has, then the system display an error message, and return to the beginning of the basic flow.
      2. BAD SEAT SELECT

The user try to sit in seated seat, return to 5.1.3

* + 1. EXIT

The user decided to exit the Application, the user disconnect from the system.

**Acceptance Test Scenarios**

**Use case**: Join existing games

1. **Good**:
   1. User success to join existing table: The user select valid table and select a seat and his money stack for the game.
   2. User exit the application: the system disconnects the user from the system
2. **Bad**:
   1. The user select bad table: the user try to enter to table that he can’t join, BAD TABLE SELECT is activated.
3. **Sad**:
   1. The user select not vacant seat: The user try to select a seat but the seat is not vacant the BAD SEAT SELECT is activated.

**Use Case Diagram**

