**Functional**

1. Connect to the game server by I.P. address and port number
2. Each player must sign up for an account
3. Players must login to access lobby
4. Players able to log off from game server
5. Game has a lobby
6. Ability to view currently active rooms in the lobby
7. Lobby has the option to create new tables, view different games/table
8. Ability to set minimum wager during room creation
9. Each table can accommodate up to 5 players per game but any number of players not playing can view the game
10. Ability to read and modify player account information
11. Add or subtract funds from player account
12. GUI incorporates player’s ability to: Sit, Hit, Stand, Split, Double
13. Shuffling system to minimize card counting
14. Logic to determine winner/loser
15. Add/subtract from account winnings/losses
16. Time limit for players to make their wager and choose additional cards/options
17. Ability for players to leave game room
18. Players able to log off server

Use Case ID: RegAcct

Use Case Name: Registers for an account, logs on to game lobby, then adds $1000 dollars to account, then logs off.

Relevant Requirements: \* Connect to the game server by I.P. address and port number, each player must sign up for an account, each player must login to access lobby, ability to read and modify player account information, add or subtract funds from player account, players able to log off from game server.

Primary Actor: New player

Pre-conditions: The new player must have the IP address and port number of the server.

Post-conditions: Player has added $1000 dollars to account.

Basic Flow or Main Scenario:

1. Connect to server by I.P. address and port number.
2. Displays connection error message if connection fails.
3. Player registers for an account.
4. Server verifies user information.
5. Server adds player to user database.
6. Player logs on to game lobby.
7. Server verifies login.
8. Server displays login error message if incorrect information provided
9. Server initializes player state information
10. Player visits account information page.
11. Server displays account information
12. Player adds $1000 dollars to account.
13. Update server account balance
14. Server displays account information
15. Player logs off.
16. Server uninitialize player state information

Extensions or Alternate Flows:

1. Connect to server by I.P. address and port number.
2. Displays connection error message if connection fails.
3. Player registers for an account.
4. Server verifies user information.
5. Server adds player to user database.
6. Player logs on to game lobby.
7. Server verifies login.
8. Server displays login error message if incorrect information provided
9. Server initializes player state information
10. Player logs off.
11. Server uninitialize player state information

Exceptions: Server is down, server fails to register player, player types in wrong registration information, player types in wrong login information, server fails to add money to account.

Related Use Cases: UserPlays

Use Case ID: UserPlays

Use Case Name: Player creates empty room, plays a hand, leaves room, log off

Relevant Requirements: \* Ability to create a new room or join a pre-existing room, ability to choose sit, hit, stand, split, and double, ability for players to leave game room, players able to log off from game server.

Primary Actor: User/player

Pre-conditions: Player has signed in, entered the lobby, can view the currently available rooms, GUI gives player choice to join one of the currently available rooms or create a new room

Post-conditions: Player has successfully created a room, account balance reflects wager and earning/loss

Basic Flow or Main Scenario:

1. Player creates room
2. Player chooses minimum wager amount
3. Update player state
4. Server initializes game state information
5. Player sits down to play
6. Update player state
7. Update game state information - add active player
8. Update player state information - playing in room x
9. Start Timer
10. Place bet
11. Deal card(s)
12. Update game state information - player issued card(s) y
13. Run game logic
14. Player chooses hit
15. Deal card(s)
16. Update game state information - player issued card(s) y
17. Run game logic
18. Update player balance
19. Player chooses stand
20. Player leaves room
21. Uninitialize game state information
22. Update player state
23. Player logs off
24. Terminate connection

Extensions or Alternate Flows:

1. Player joins room
2. Player sits down to play
3. Update game state information - add active player
4. Update player state information - playing in room x
5. Place bet
6. Deal card(s)
7. Update game state information - player issued card(s) y
8. Run game logic
9. Update player balance
10. Player chooses hit
11. Deal card(s)
12. Update game state information - player issued card(s) y
13. Run game logic
14. Update player balance
15. Player chooses double
16. Deal card(s)
17. Update game state information - player issued card(s) y
18. Run game logic
19. Update player balance
20. Player chooses split
21. Deal card(s)
22. Update game state information - player issued card(s) y
23. Run game logic
24. Update player balance
25. Player chooses stand
26. Update player balance
27. Update game state information - remove active player

Exceptions: Server is down, player fails to join room, server fails to initialize game, server fails to issue card(s), server fails to update player’s account balance, game logic incorrect,

Related Use Cases: RegAcct