Blackjack

Online Game

24-Apr-15

Benjamin Umbach, Stoyan Staynov, Popa Vlad

Table of Contents

[Description of the Game 2](#_Toc417590129)

[Requirements 2](#_Toc417590130)

[Use-Cases 3](#_Toc417590131)

[1.1: Login 3](#_Toc417590132)

[1.2: Start a game 3](#_Toc417590133)

[1.3: Joining a game 4](#_Toc417590134)

[1.4: Invite a Player 4](#_Toc417590135)

[1.5: Playing with one player 5](#_Toc417590136)

[1.6: Adding money to the pool 5](#_Toc417590137)

[1.7: Hit 6](#_Toc417590138)

[1.8: Stay 6](#_Toc417590139)

[1.9: Leaving a game 6](#_Toc417590140)

[1.10: Logout 7](#_Toc417590141)

[Glossary 8](#_Toc417590142)

[Use Case Diagram 8](#_Toc417590143)

[UI Design 9](#_Toc417590144)

[Lobby 9](#_Toc417590145)

[Game Field 10](#_Toc417590146)

[Non-functional Requirements 11](#_Toc417590147)

[Constraints 11](#_Toc417590148)

[External Interface Requirements 11](#_Toc417590149)

[Performance 11](#_Toc417590150)

[Quality Attributes 11](#_Toc417590151)

# 

# Description of the Game

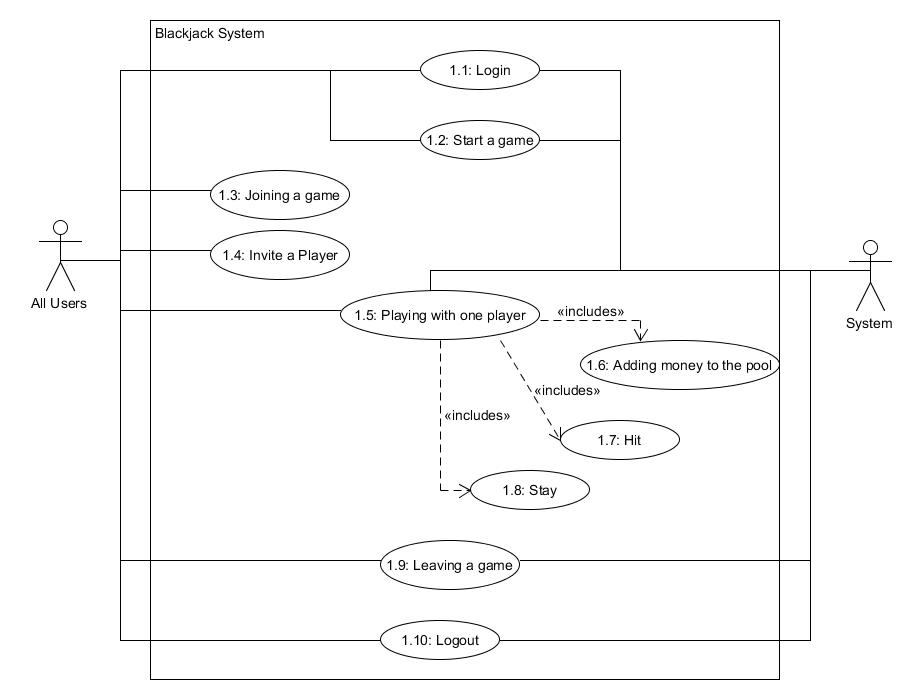
Blackjack is a gambling game in which the goal is to reach as close as possible to 21 points without going over. The players start off with 2 cards, one is visible to all players, and the other is only visible to that player. As the game is played, players will be able to choose to gain another card or stay with the cards they have; they will also have the ability to chat with the other players during the game. At the end the points are compared and the closest to 21 without going over wins. This will be a portal type game in which there is a lobby where players can find and play with others as well as take part of a public chat. To participate they will have to login to the portal.

# Requirements

|  |  |
| --- | --- |
| Requirements | MoSCoW |
| Start a Game | M |
| Invite Players | M |
| Chat with Opponents | M |
| Chat with all online | S |
| Play vs one Player | M |
| Play vs multiple players | C |
| Spectate other games | C |
| Login/logout | M |
| Moves in game | M |
| Private Games | S |
| Leave a game | M |
| Join a game | M |
| (un)Friend Player | C |

|  |
| --- |
| Moves in Game |
| Adding money to the pool |
| Hit |
| Stay |

# Use Case Diagram



# Use-Cases

## 1.1: Login

Actors: User

Description: Logging in

Pre – Condition: The program must be started.

MSS: 1. The system prompts the user to login.

2. The user inputs his/her username and password

3. The user clicks the login button or presses enter

4. The System redirects the user to the main logged in page

Post Condition: The user is logged in.

Exceptions:

3a. The user inputs wrong username or password.

3b. A messages pops up informing the user that the login info is incorrect.

## 1.2: Start a game

Actors: User

Description: Starting a new game.

Pre – Conditions: The program must be running and the user logged in.

MSS: 1. The user clicks the start new public game button.

2. The system generates a game in which the user is a player.

3. The system waits for more players.

Post Condition: The game is created.

Exceptions:

Non

## 1.3: Joining a game

Actors: User

Description: Joining an already started game.

Pre – Conditions: The program must be running and the user logged in and a game running.

MSS: 1. The system displays a list of all games

2. The user selects the game they wish to join.

3. The user clicks the join game button.

4. The system connects the user to the game.

Post Condition: The user joins the game as a player.

Exceptions:

2ac. If password is correct continue MSS, else a message pops up indicating wrong password.

2ba.The game is full.

2bb. The system displays a message that the game is full

3a. The hand is not finished.

3ab. The system notifies the user he will have to wait till the end of the hand

3ac. The system waits until the hand is finished

## 1.4: Invite a Player

Actors: User 1, User 2

Description: User 1 invites User 2 into a game.

Pre – Conditions: User 1 must be in a game.

MSS: 1. User 1 right clicks on User 2’s name.

2. User 1 then clicks on invite to game option.

3. The system asks user 2 if he/she wants to join the game.

4. User 2 selects yes.

5. The system adds user 2 to the game of user 1

Post Condition: User 2 joins the game.

Exceptions:

3a. User 2 is already in a game.

3b. User 1 receives a message that User 2 is in a game.

4a. User 2 selects no.

4b. User 1 receives a message that User 2 did not want to join.

## 1.5: Playing with one player

Actors: User 1 and User 2

Description: User 1 is playing against User 2

Pre – Conditions: Both User 1 and User 2 must be in the same game.

MSS: 1. Both User 1 and User 2 select the amount of money they wish to bet go to **UC 1.6**.

2. System gives both players two cards.

3. User 1 and User 2 choose from hit(**UC 1.7**), stay(**UC 1.8**)

4. Step 3 repeats until either both select stay or are bust.

5. The system determines the winner.

6. Winner receives money.

7. Repeat.

Exceptions:

6a. It is a tie.

6b. Money gets returned to its respective owner.

7a. User 1 or User 2 leaves game.

7b. System waits for a new player.

## 1.6: Adding money to the pool

Actors: User

Description: User changes the amount of money being but into the pool

Pre – Conditions: User one must be in a game.

MSS: 1. User selects the amount of money he/she wishes to bet.

2. User clicks the ready button.

3. The system takes the amount of money from the user and puts it in the pool.

Post – Condition: money is placed in the pool.

Exceptions:

1a. User does not have the amount of money.

1b. ready button is disabled until the amount of money is viable.

## 1.7: Hit

Actor: User

Description: User wished to add another card.

Pre – Conditions: User must be in a game and have at least 2 cards and not be bust.

MSS: 1. User clicks hit.

2. System adds a card.

3. System calculates new value and shows it to all users in the game.

## 1.8: Stay

Actor: User

Description: User to stay with the current value.

Pre – Conditions: User must be in a game and have at least 2 cards and not be bust.

MSS: 1. User clicks stay.

2. System gives turn to the next player.

Post Condition: The system disables the hit button

## 1.9: Leaving a game

Actors: User

Description: User wishes to leave the game he/she is currently playing in.

Pre – Conditions: User must be in a game.

MSS: 1. User clicks on the Leave game option.

2. System prompts user if he/she is sure.

3. User selects yes.

4. System returns user to the main logged in page

Post Condition: User is removed from the game.

Exceptions:

1a. There is a current hand in play.

1b. A message pops up telling the user he/she will have to wait until after the hand is played.

3a. The User selects no.

3b. The User is not removed from the game.

## 1.10: Logout

Actors: User

Description: Logging out

Pre – Condition: The program must be running.

MSS: 1. User clicks on Logout.

2. The system prompts the User if he/she is sure.

3. The User selects yes.

4. The system returns the user to the main logged out page.

Post Condition: The user is logged out.

Exceptions:

1a. The User is in the middle of the game.

1b. The User is prompted that he/she needs to leave the game first (**UC 1.9**).

3a. The User selects no.

3b. The User is not logged out.

# Glossary

MSS: Main Success Scenario

Pool: Where the money is being held until a winner is determined.

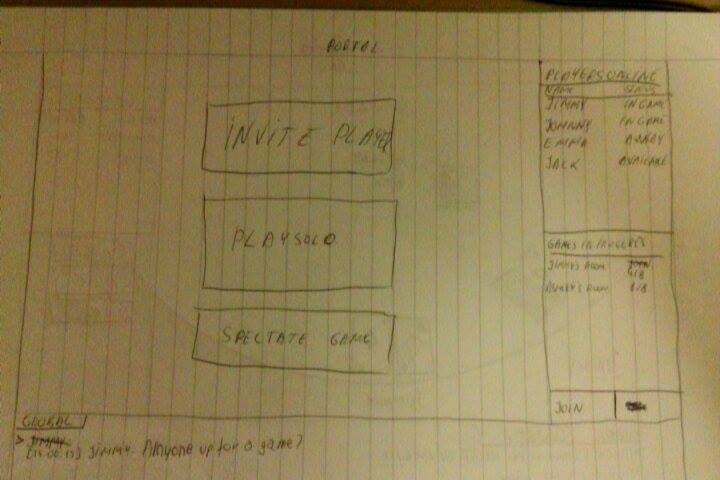
Hit: The action of getting another card.

Stay: The action of abstaining from getting another card until the end of the hand.

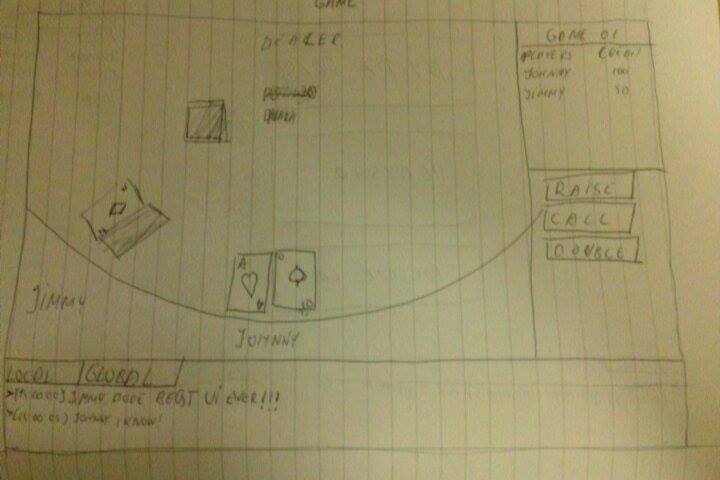
Hand: The round being played.

# UI Design

## Lobby



## Game Field



# Non-functional Requirements

Non-functional requirements are not concerned with the functions of the system. Instead, they look at the criteria to which the software or website is expected to conform to. Non-functional requirements can include things like response time and reliability. It can also be closely tied to user satisfaction.

## Constraints

Which imposes limits on the choices available to the architects and engineers. There are two different kinds of constraints, they could be design constraints and implementation constraints.

**Design constraints**: The application is going to be for desktop computer with windows 7/8 operating systems explicitly.

**Implementation constraints**:

C# programming language.

## External Interface Requirements

Application shows changes on the screen instantly, and changes in hand and credits.

## Performance

The game will be a simple application requiring no more than a normal program worth of memory.

## Quality Attributes

* Usability: The UI will be designed in such a way that it is easy to read and understand.
* Install-ability:  Our software has everything compiled into one file (.exe). An internet connection is needed in order to play the game.
* Integrity: the two different parts are easy to transit.
* Availability:  all functionalities are available after entering a game.