Software Requirements Specification
Version 2.0
March 19, 2014

# Online 28

Submitted by,

Punit Kumar Singh (M120527CA)(Leader)
Dipanjan Saha (M120359CA)
Niranjan Singh (M120377CA)
Sujay Kumar Mandal (M120361CA)

Submitted in partial fulfillment
Of the requirements of
**CS3004 Software Engineering**

## Table of Contents

## Table of Figures

# 1. Introduction

## 1.1. Purpose

The Software Requirements Specification provides a complete description of all the software design requirements and specifications of *online 28*, a card game. This document explains all features, functions and constraints of this program. *Online 28* is a program built to allow remote users to play with each other through an anonymous system. Its main focus is just to let people play. A new 4K+1 user will initiate a game, and the next three users arriving will pair up or pair against as they decide, where K is number of games that are already started.

## 1.2. Scope

*Online 28* is an online card-game which allows users to engage in a real-time game of cards. Anonymous users can login to the system and play game with other online users. The primary goal is to provide a card game that is intuitive and entertaining for players.

## 1.3. Definitions, acronyms, and abbreviations

| **Term** | **Definition** |
|---|---|
| Trick | All the cards played in a single round, one from each player. |
| Suit | For a standard 52 card deck, a suit is one of the four categories into which the deck is divided, that is spades, hearts, clubs or diamonds. |
| Trump Suit | A particular suit is set by the highest bidder called the trump suit. This suit is ranked above all the other suits, and automatically prevails over them, losing only to a higher |

| | |
|---|---|
| | trump if one is played to the same trick. |
| Trump Card | Any card belonging to the trump suit. |
| Trump | When a player plays a trump card in a particular trick(round) is called a trump. A trump card cannot be played if the player has any cards of the suit that led to the trick |
| Bid | An offer to make a specified number of points. In Online 28 a bid should start from 16 points. |

### *1.4. References*

[1.] Wikipedia-wikipedia.org/wiki/Twenty-eight_(card_game)

[2.] rummy.in/29.html

[3.] IEEE Std 830-1998

### *1.5. Document overview*

The rest of the document details the functional and non-functional requirements for *online 28* in two chapters. First providing a full description of the project, it lists all the functions performed by the system. The final chapter concerns details of each of the system functions and actions in full. Its format follows IEEE Std 830-1998 as closely as possible.

# 2. Overall description

## 2.1 Product Perspective

There is no shortage of computer-based card games. Online28 is not significantly different from these other card games.

The game allows four players to play a game of 28 over a network. Multiple games can occur concurrently. It provides a simple GUI.

### 2.1.1 System Interfaces

The web browser software is the interface that translates the user’s actions with the mouse into data. The server will be able to understand and relay to the other user’s computer, in order to actually perform the move on the other three computers.

### 2.1.2 User Interfaces

Online 28 web pages provide a simple GUI that the user operates with a mouse. It only requires a working Internet connection. Game data will be stored on a central server.

### 2.1.3 Hardware Interfaces

Online 28 will run on any computer with the following:

- An Internet connection
- A keyboard and mouse

### 2.1.4 Software Interface

- Any web browser

### 2.1.5 Communications Interfaces

Communication between the client software and the network software is facilitated by common network protocols.

### 2.1.6 Memory Constraints

The client has no memory constraint other than sufficient primary memory to run a web page. Server will have a memory constraint i.e the number of players who can log in simultaneously.

### 2.1.7 Operations

Not Applicable

### 2.1.8 Site Adaption Requirements

Not applicable

## 2.2 Product Functions

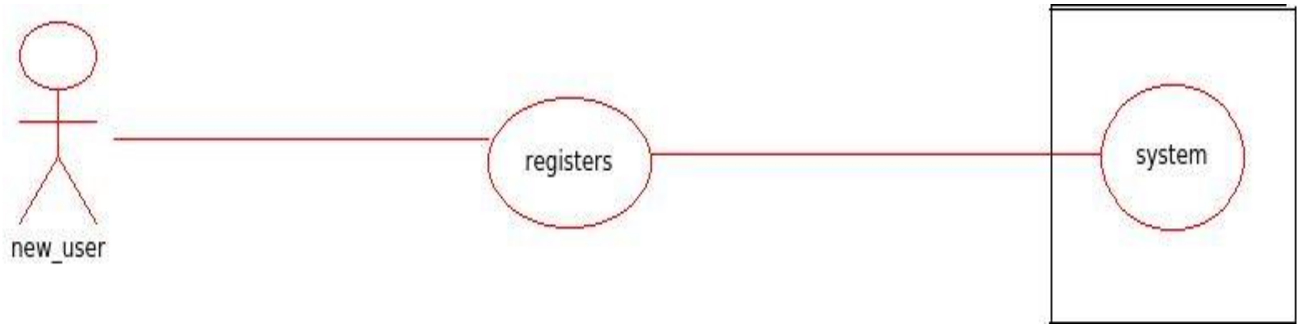

*Figure 1. <u>New User Registration</u>* – New user registers to the system.
*Xref : SRS 3.2.1*

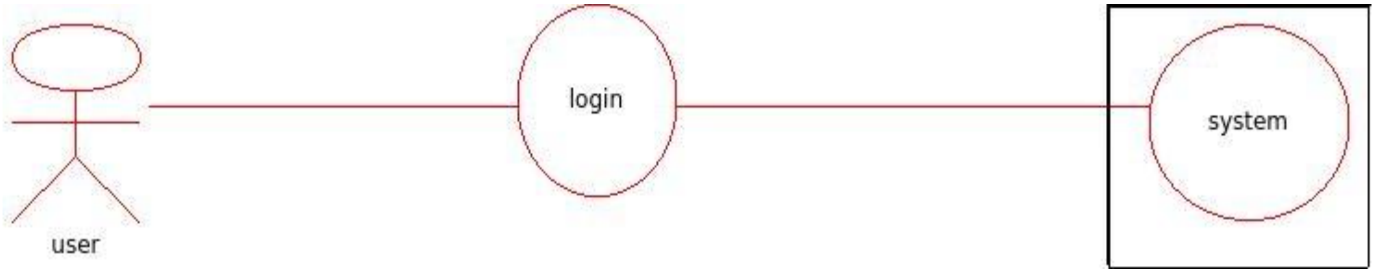

*Figure 2. <u>User Login</u>* – Registered user logins to the system.
*Xref : SRS 3.2.2*

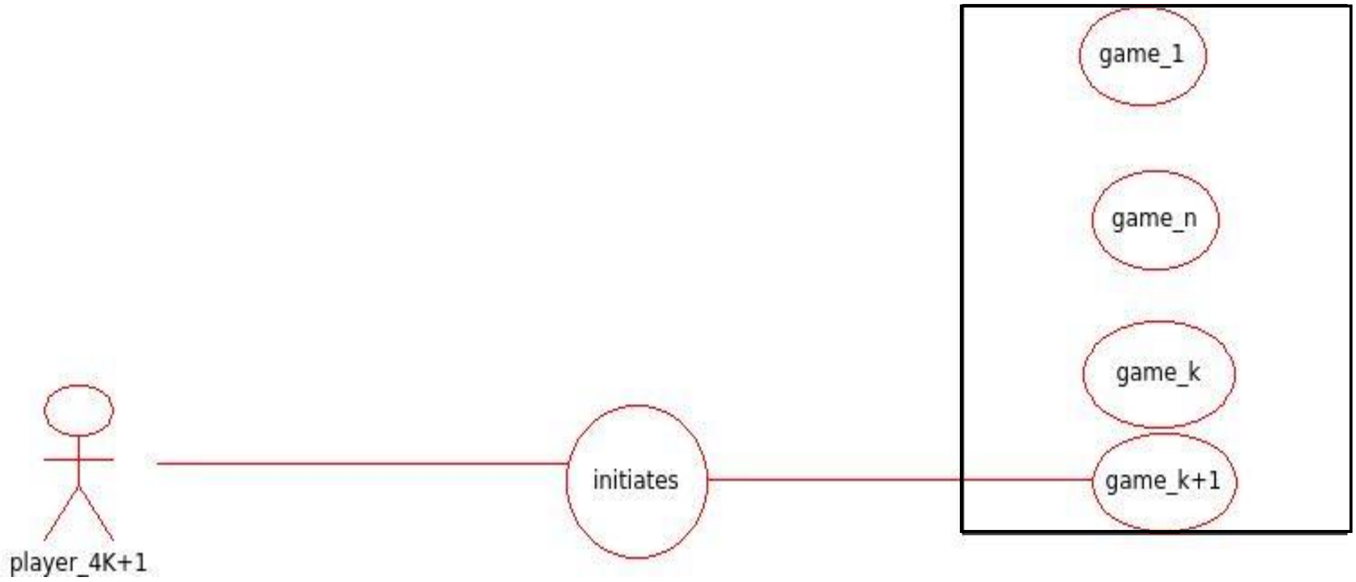

*Figure 3.* *<u>Player 4k + 1 initiates game k + 1</u>* - First player initiates new game.
*Xref : SRS 3.2.3*

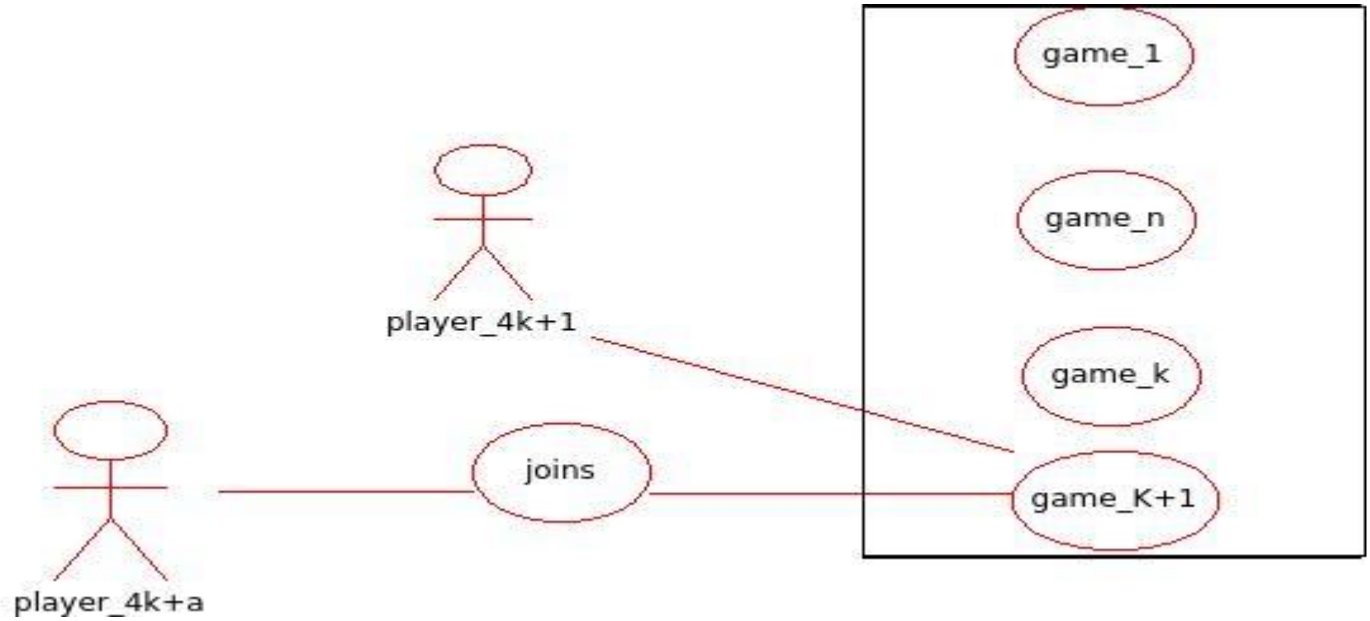

*Figure 4.* *<u>Player 4k + a joins game k+1</u>* – User 2 <= a < = 4 joins the game.
*Xref: SRS 3.2.4*

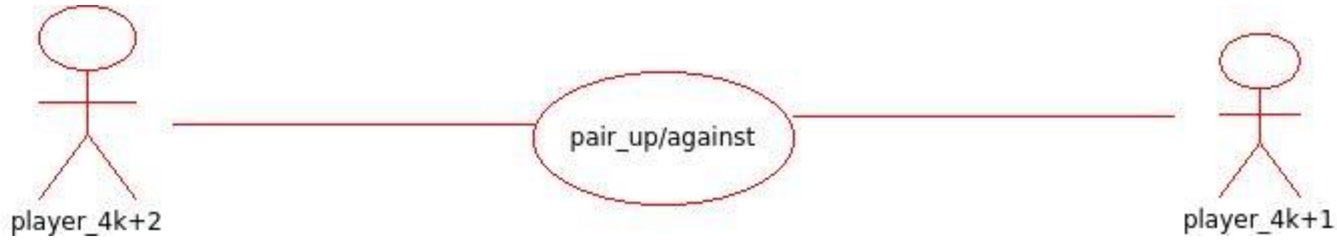

*Figure 5.* *<u>Player 4k + 2 pair_up or against player 4k + 1</u>* – Second player decides whether to pair up or pair against with first player. *Xref: SRS 3.2.5*

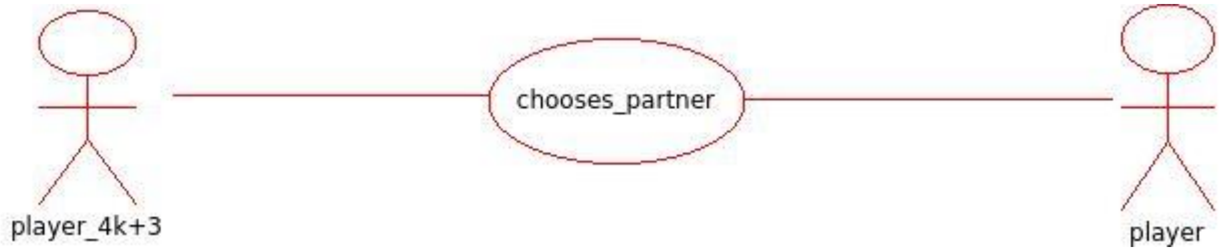

*Figure 6.* *<u>Player 4k + 3 chooses partner between player 4k +1 or player 4k + 2</u>*

– Third player chooses whether to pair up with first player or second player. *Xref: SRS 3.2.6*

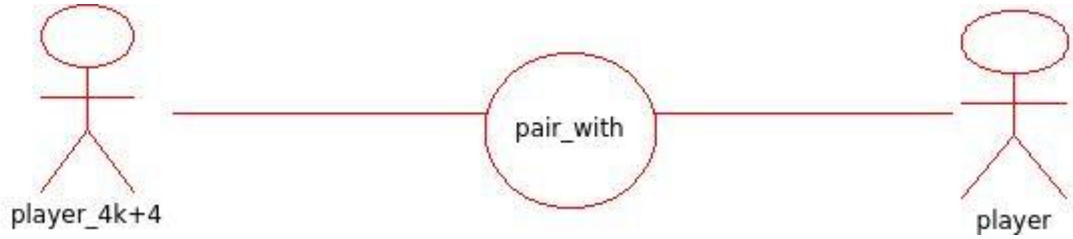

*Figure 7.* *<u>Player 4k + 4 pairs up with unpaired player</u>* – Fourth player pairs up. *Xref: SRS 3.2.7*

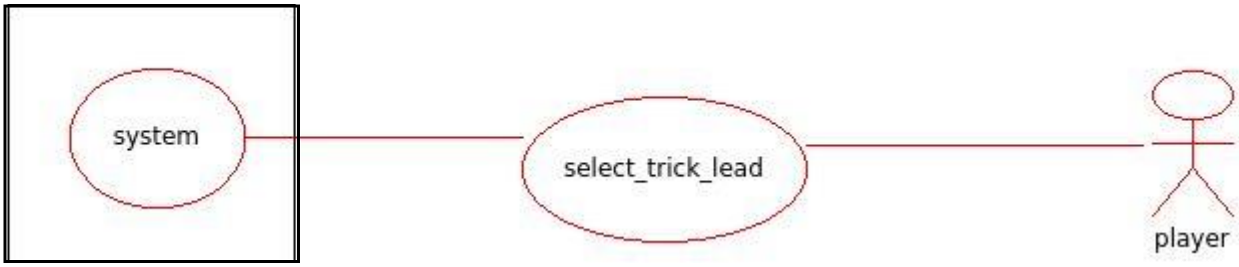

*Figure 8.<u>System selects trick lead</u>* – Trick leader is to initiate the coming trick.

*Xref: SRS 3.2.8*

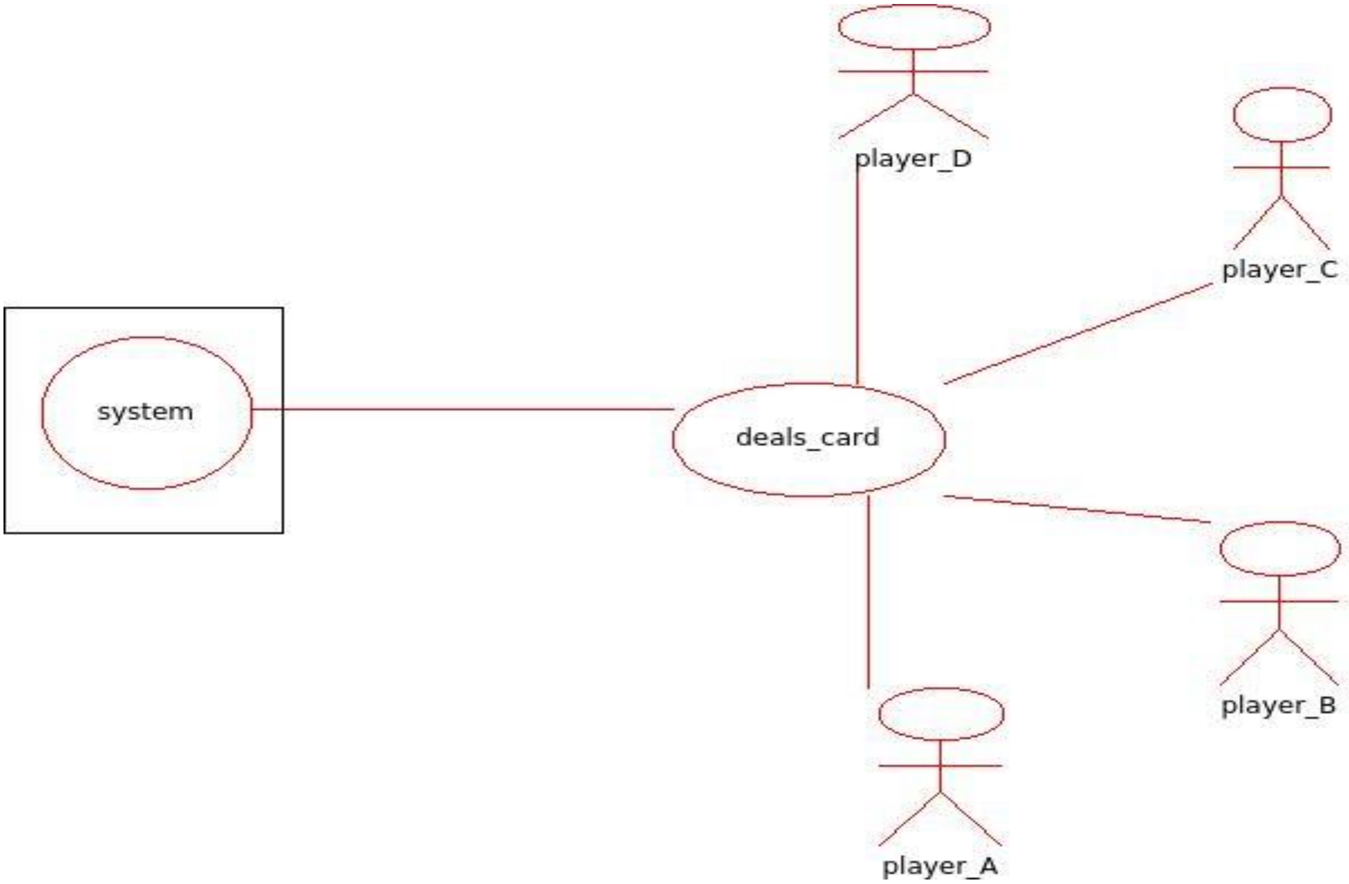

*Figure 9. <u>System deals cards to all four players</u>* – Dealing starts with player 4k+1. *Xref: SRS 3.2.9*

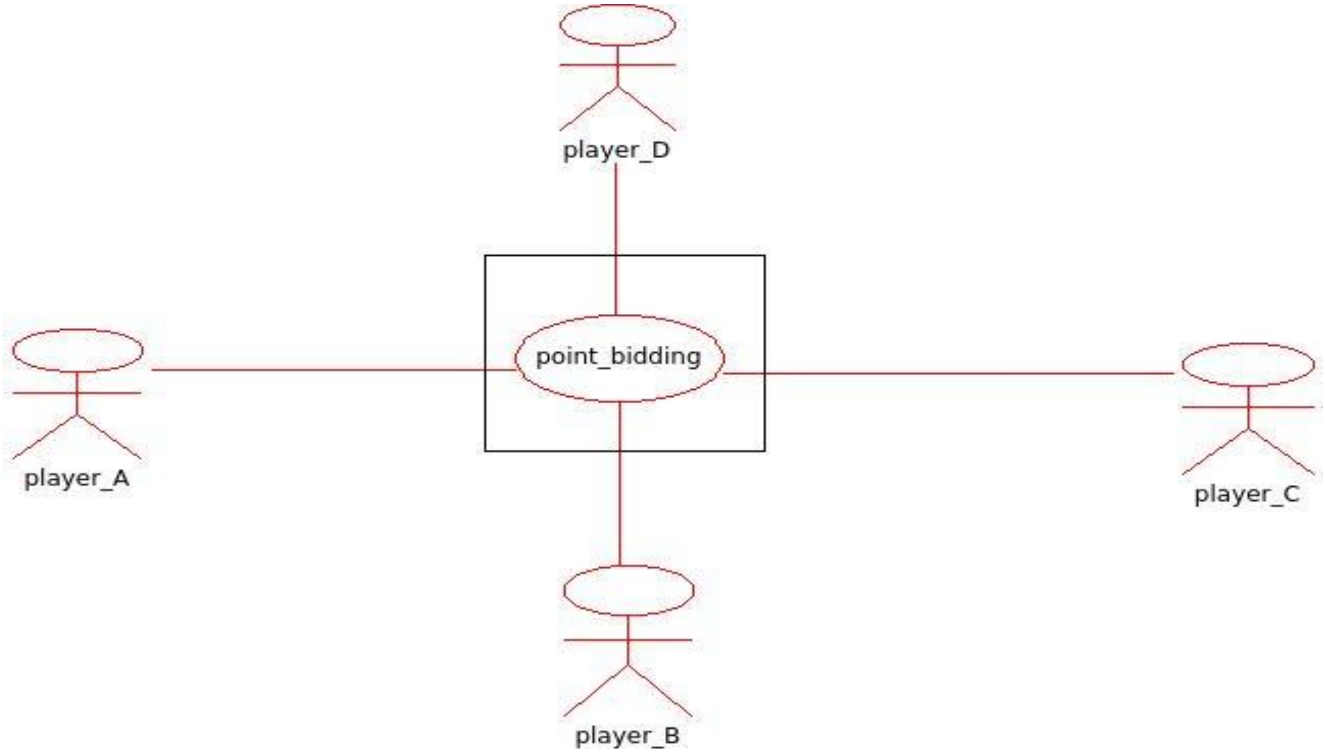

*Figure 10.* *Point bidding* – System organizes point bidding among four players in a particular order. *Xref: SRS 3.2.10*

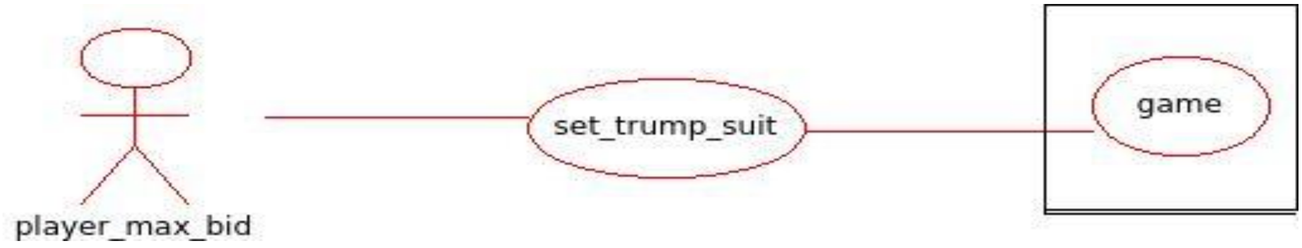

*Figure 11.* *Set trump suit* – The max bidder sets trump suit.

*Xref: SRS 3.2.11*

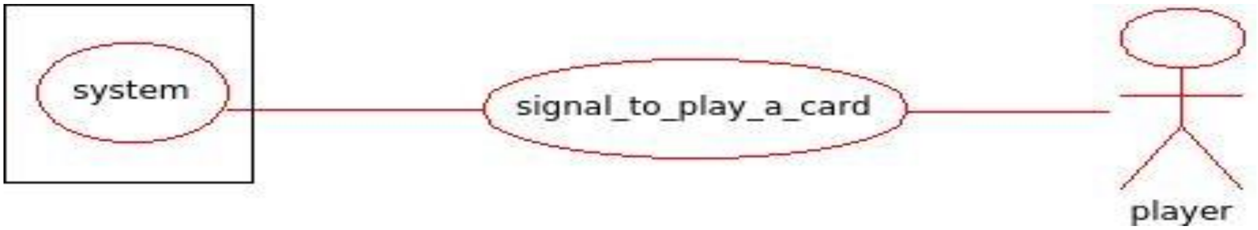

*Figure 12.* *<u>System signals a player to play a card</u>* – Players can play cards in a particular order. *Xref: SRS 3.2.12*

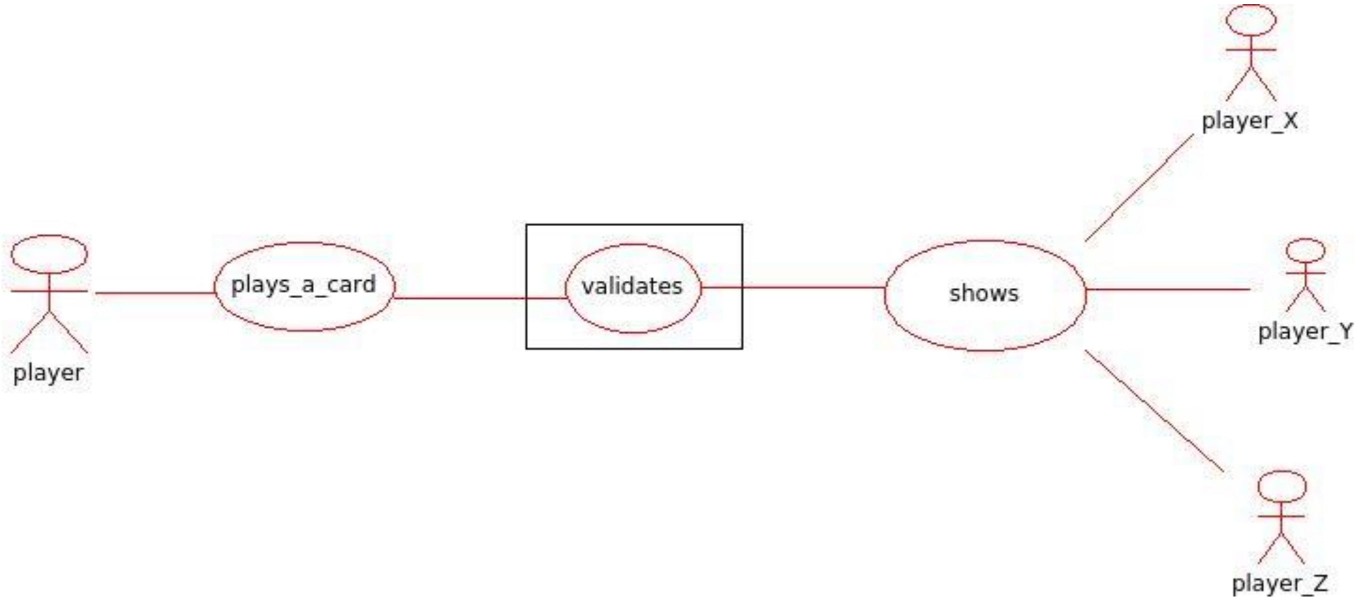

*Figure13.* *<u>System validates the card played by user</u>* – After validation only the player card is shown to other players. *Xref: SRS 3.2.13*

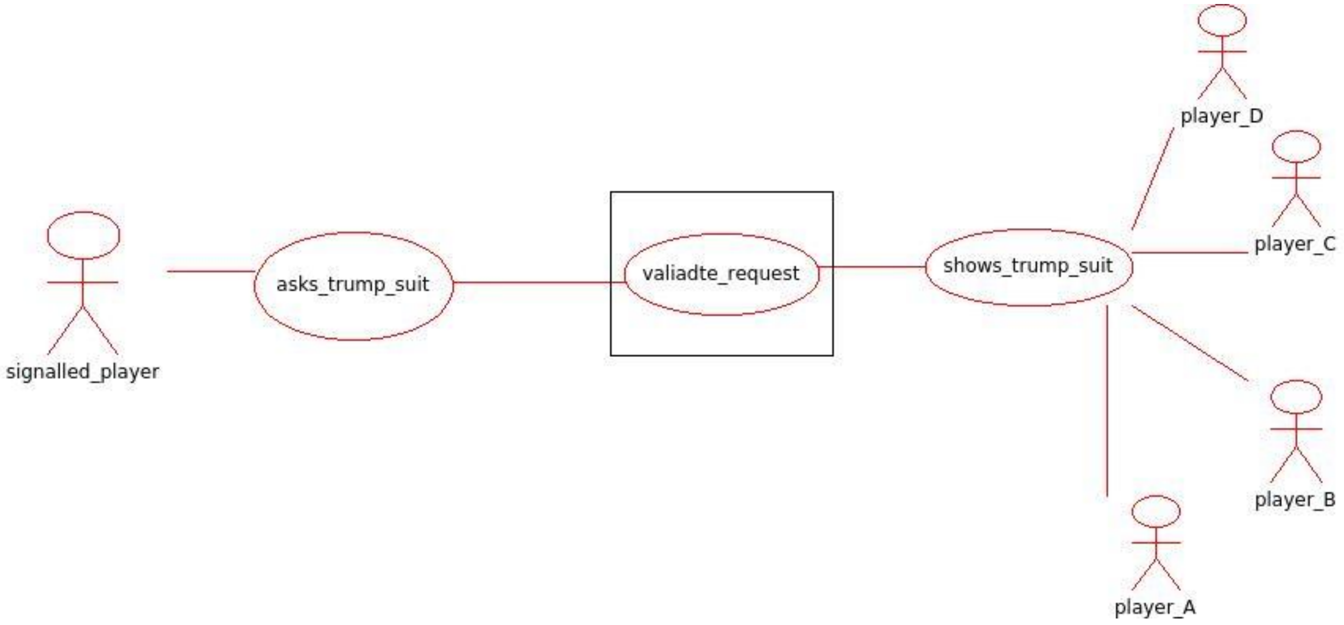

*Figure 14.* *<u>Player requests system for the trump suit</u>* – System shows the trump suit to all four players after validating the request. *Xref: SRS 3.2.14*

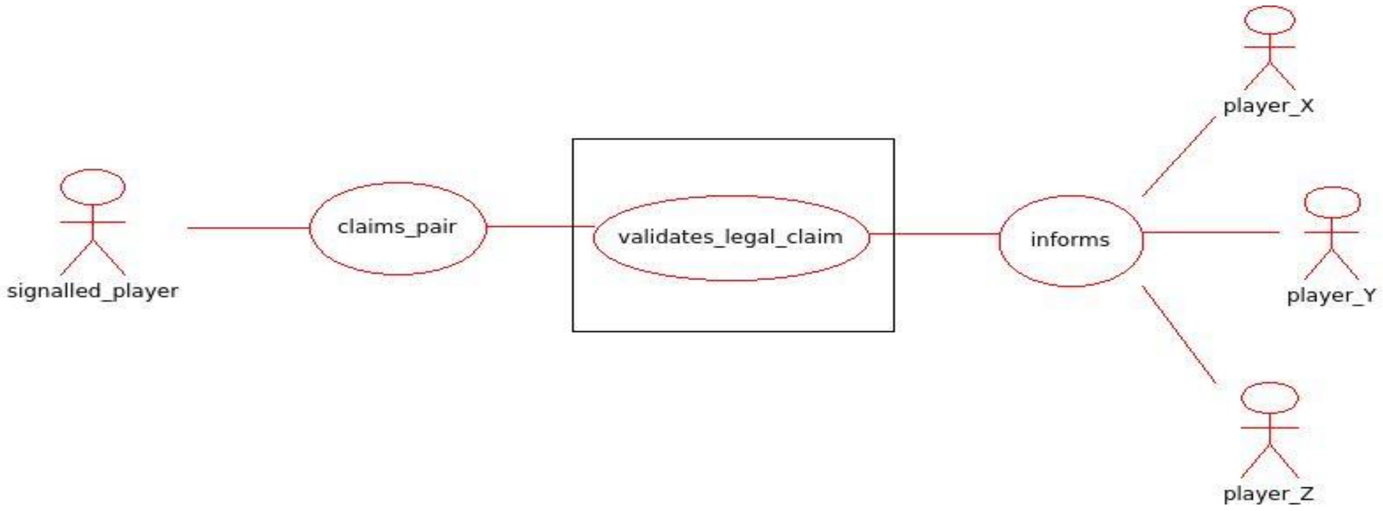

*Figure 15.* *<u>Player claims possession of trump suit pair</u>* – System checks whether the claim is legal or not. *Xref: SRS 3.2.15*

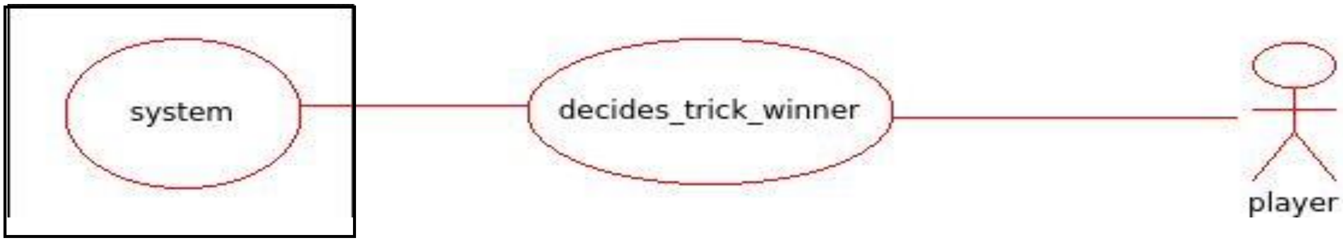

*Figure 16.* *System decides trick winner* – Decision is taken only after completion of the trick.
*Xref: SRS 3.2.16*

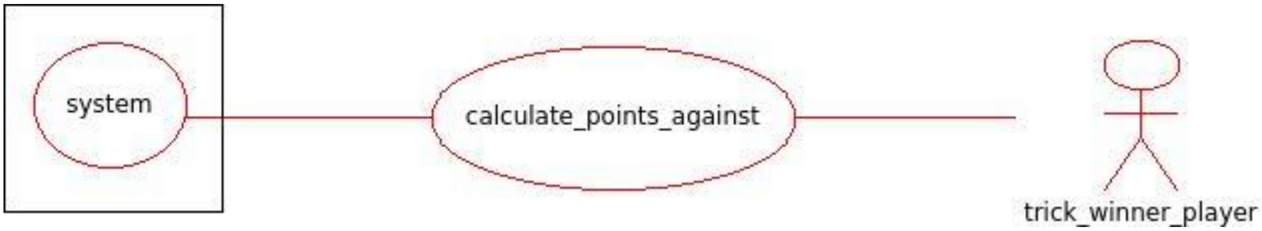

*Figure 17.* *System calculates points against the trick winner* – Points in the all four cards of the trick are added up. *Xref: SRS 3.2.17*

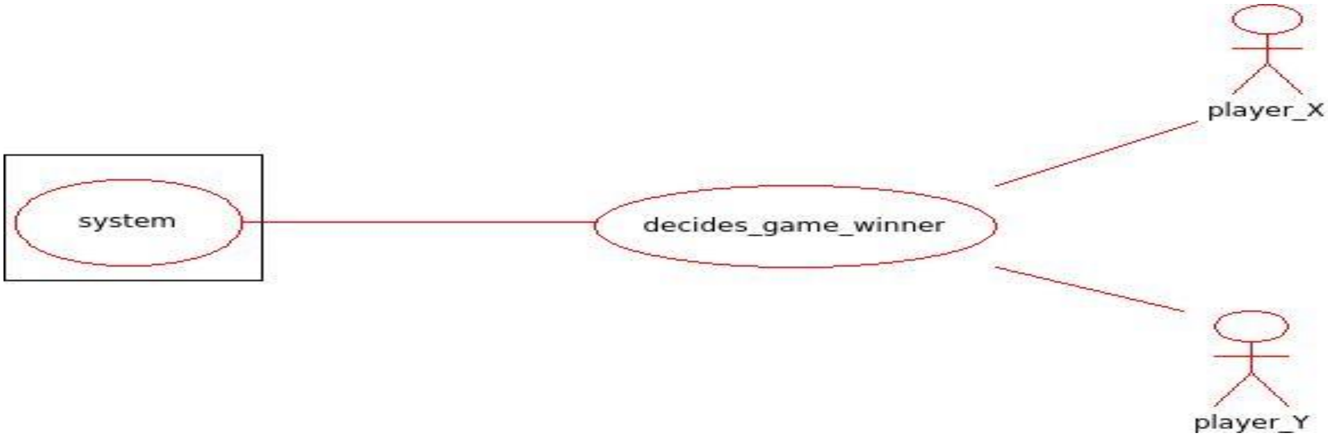

*Figure 18.* *System decides the winning team* – After completion of all 8 tricks the game completes. *Xref: SRS 3.2.18*

## 2.3 User Characteristics

The main users of this program will be people who are familiar with the game of 28 and wish to play it in an online environment.

## 2.4 Constraints

The number of users is restricted because of the server's capacity. Response time of the web service is a performance constraint.

## 2.5 Assumptions and Dependencies

Not Applicable.

## 2.6 Apportioning of requirements

Not Applicable.

## 3. Requirement specifications

### *3.1. External interface specifications*

Not Applicable.

### *3.2. Functional Requirements*

3.2.1 New user should be able to register in the system.

| | |
|---|---|
| **Use Case** | New user registration |
| **Precondition** | User not already registered |
| **Basic Path** | 1. User chooses a distinct user-name<br>2. User chooses a password<br>3. User goes for registration<br>4. System successfully registers user |
| **Alternate Path** | N/A |
| **Postcondition** | New user entry created in system database |
| **Exception Path** | 1. User chooses an user-name that already Exists<br>2. User goes for registration<br>3. Registration unsuccessful<br>4. User is informed about reason |
| **Reference:** | SRS 2.2 / Figure 1 |

### 3.2.2 Registered user should be able to login to system.

| | |
|---|---|
| **Use Case Name:** | User login |
| **Precondition** | User need to be registered |
| **Basic Path** | 1. User enters user-name<br>2. User enters corresponding password<br>3. User goes for login<br>4. System matches the password against the user-name stored in database<br>5. Upon successful matching system allows user to Login |
| **Alternate Path** | N/A |
| **Postcondition** | User logs in into the system |
| **Exception Path 1** | 1. User-name doesn't exist<br>2. Login unsuccessful<br>3. User redirected to new user registration Option |
| **Exception Path 2** | 1. Password doesn't match<br>2. User is informed<br>3. User redirected to login option |
| **Reference:** | SRS 2.2 / Figure 2 |

3.2.3 Player_4K + 1 should be able to initiate a new game_K + 1, if K games is in progress.

| | |
|---|---|
| **Use Case** | Player_4K+1 initiates game_K+1 |
| **Precondition** | User had successful login into system |
| **Basic Path** | 1. User wants to play game<br>2. System checks limits of concurrent games<br>2. User initiates a new game<br>3. User as a player waits for 3 more players |
| **Alternate Path** | N/A |
| **Postcondition** | One player is assigned to the game_K+1 |
| **Exception Path 1** | 1. Maxi-limit of concurrent games reached<br>2. User can't initiate a new game<br>3. User waits for a new game |
| **Exception Path 2** | 1. User stop waiting & quits<br>2. Game_K+1 cancelled & players removed |
| **Reference:** | SRS 2.2 / Figure 3 |

### 3.2.4 Player_4K+2, Player_4K+3, Player_4K+4 should be able to join the game_K+1

| | |
|---|---|
| **Use Case** | Player_4K+A joins game_K+1 |
| **Precondition** | 1. User had successful login into System<br>**2.** 2 <= A <= 4 |
| **Basic Path** | 1. User wants to play game<br>2. User joins the game_K+1 |
| **Alternate Path** | N/A |
| **Postcondition** | User is assigned to the game_K+1 as player |
| **Exception Path** | N/A |
| **Reference:** | SRS 2.2 / Figure 4 |

### 3.2.5 Player_4k+2 have the option to pair up or pair against with player_4K+1.

| | |
|---|---|
| **Use Case** | Player_4K+2 pair up / against player_4K+1 for game_K+1 |
| **Precondition** | Both players are assigned to the game_K+1 |
| **Basic Path** | 1. System asks player_4K+2 to pair up with player_4K+1<br>2. Player_4K+2 accepts the proposal<br>3. Player_4K+2 pair up with Player_4K+1 |
| **Alternate Path** | 1. System asks player_4K+2 to pair up with player_4K+1<br>2. Player_4K+2 denies the proposal<br>3. Player_4K+2 pair against Player_4K+1 |
| **Postcondition** | Both players either pair up or pair against each other |
| **Exception Path** | Any of two players quits & game_K+1 cancelled , players removed |
| **Reference:** | SRS 2.2 / Figure 5 |

### 3.2.6 Player_4K+3 should have option to choose partner to pair up for game_K+1

| | |
|---|---|
| **Use Case** | Player_4K+3 chooses partner to pair up for game_K+1 |
| **Precondition** | Player_4K+1 and Player_4K+2 paired against each other |
| **Basic Path** | 1. System asks player_4K+3 to pair up with player_4K+1<br>2. Player_4K+3 accepts the proposal<br>3. Player_4K+3 pair up with Player_4K+1 |
| **Alternate Path** | 1. System asks player_4K+3 to pair up with player_4K+1<br>2. Player_4K+3 denies  the proposal<br>3. Player_4K+3 pair against Player_4K+1 |
| **Postcondition** | Player_4K+3 pair up with either player_4K+1 or player_4K+2 |
| **Exception Path** | Any of three players quits & game_K+1 cancelled , players removed |
| **Reference:** | SRS 2.2 / Figure 6 |

### 3.2.7 Player_4K+4 should be paired up with unpaired player for game_K+1.

| | |
|---|---|
| **Use Case** | Player_4K+4 pair up with unpaired player for game_K+1 |
| **Precondition** | Player_4K+4 joins the game & one among other 3 players waiting for partner to pair up |
| **Basic Path** | Player_4K+4 paired up with the unpaired player by the system |
| **Alternate Path** | N/A |
| **Postcondition** | All 4 players paired up forming 2 teams |
| **Exception Path** | Any of four players quits & game_K+1 cancelled , players removed |
| **Reference:** | SRS 2.2 / Figure 7 |

3.2.8 System should be able to select the player to lead a trick.

| | |
|---|---|
| **Use Case** | System selects the player to lead a trick |
| **Precondition** | Either the game is about to start or a new trick of the game is to begin. |
| **Basic Path** | Player_4K+1 is chosen to lead the first trick of the game_K+1 |
| **Alternate Path** | The player who won the last trick is chosen to lead the next trick |
| **Postcondition** | Trick leader signaled to lead the next trick also, bidding & dealing starts with trick lead in case of a fresh game |
| **Exception Path** | N/A |
| **Reference:** | SRS 2.2 / Figure 8 |

### 3.2.9 System should able to deal cards to all 4 players.

| | |
|---|---|
| **Use Case** | System deals card to all 4 players |
| **Precondition** | 1. Player to lead a trick is selected and teams are formed<br>2. Ordering of players is done<br>3. In case of 2nd dealing the trump suit is set by the max bidder |
| **Basic Path** | 1. System shuffles 32 cards (J,9,A,10,K,Q,8,7 of 4 suits) in random<br>2.First 4 cards shown to the trick lead<br>3. Next 4 cards shown to the next player in order, who is in opponent team of trick lead<br>4. Next 4 cards shown to the next player in order, who is paired up with trick lead<br>5. Next 4 cards shown to the last player in order, who is in opponent team of trick lead |
| **Alternate Path** | 1. System starts with rest 16 cards remaining without shuffling them again<br>2. System follows the same order to distribute 4 cards at a time to all 4 players |
| **Postcondition** | 4 cards are dealt at a time to all 4 players |
| **Exception Path** | N/A |
| **Reference:** | SRS 2.2 / Figure 9 |

3.2.10 System should be able to organize point bidding among 4 players and should be able to find the max bidder.

| | |
|---|---|
| **Use Case** | System organizes point bidding among 4 players to find the highest bidder |
| **Precondition** | All 4 players have been dealt 4 cars only |
| **Basic Path** | 1. Minimum bid is 15 , maximum 28<br>2. Players can either Pass or bid higher point than last one<br>3. System takes primary point bid from all 4 players in the order the cards dealt .<br>4. If first 3 players in order call Pass then 4th player starts game with 15 automatically<br>5. Next round of bidding takes place in circular order leaving those who called Pass in last round<br>6. Bidding goes on till any player calls 28 or at least 3 player call Pass |
| **Alternate Path** | N/A |
| **Postcondition** | The player bids maximum point without calling Pass wins the bidding |
| **Exception Path** | N/A |
| **Reference:** | SRS 2.2 / Figure 10 |

### 3.2.11 The player who win the bidding should be able to set the trump suit for the game.

| | |
|---|---|
| **Use Case** | The player who wins points bidding, sets the trump suit for the game |
| **Precondition** | 1. Bidding is over<br>2. Max bidder is selected |
| **Basic Path** | 1. Max bidder sets the trump suit ( one among diamond, heart, spade ,club)<br>2. Only system knows the suit, not exposed to the rest 3 players |
| **Alternate Path** | 1. Max bidder sets the trump suit as NONE i.e. No Trump suit |
| **Postcondition** | Trump suit is set either one of 5 options by max bidder |
| **Exception Path** | N/A |
| **Reference:** | SRS 2.2 / Figure 11 |

3.2.12 System should be able to signal a particular player to play a card.

| | |
|---|---|
| **Use Case** | System signals a particular player to play a card |
| **Precondition** | 1. Trump suit is set<br>2. If in middle of a trick then the previous player in order has played its turn |
| **Basic Path** | 1.  System signal the player to play a card<br>2. System allots a time limit to the player to play its turn |
| **Alternate Path** | N/A |
| **Postcondition** | The player signaled is allowed to play a card |
| **Exception Path** | N/A |
| **Reference:** | SRS 2.2 / Figure 12 |

3.2.13 System should be able to validate the card played by a player before showing it to others.

| | |
|---|---|
| **Use Case** | Player plays a card and system validates the play before showing it to other 3 players |
| **Precondition** | The player who plays a card is signaled by system to play a card |
| **Basic Path** | 1. Player plays a card<br>2. System validates the play, whether the correct suit is played or not<br>3. If validated system shows the card to all rest 3 players |
| **Alternate Path** | N/A |
| **Postcondition** | All players can see the card played |
| **Exception Path** | 1. Player plays a wrong suit intentionally<br>2. On calling for trump suit the player play a different suit intentionally<br>3. System invalidate the play<br>4. Player forced to play again |
| **Reference:** | SRS 2.2 / Figure 13 |

### 3.2.14 Player should be able to request the system for showing up trump suit and system should be able to validate the request.

| | |
|---|---|
| **Use Case** | Player asks the trump suit to the system & system validates the request before showing up trump suit to all players |
| **Precondition** | 1. The Player asks for trump suit need to be signaled to play a card<br>2. Trump suit should not be exposed already<br>3. The player should not be a trick lead for that particular trick |
| **Basic Path** | 1. Player requests for trump suit<br>2. System validates the request<br>3. If validated system shows the suit to all 4 players |
| **Alternate Path** | N/A |
| **Postcondition** | Trump suit is exposed to all 4 players |
| **Exception Path** | 1. Player has the current suit card to play, but yet requests for trump suit<br>2. System denies the request |
| **Reference:** | SRS 2.2 / Figure 14 |

3.2.15 Player should be able to claim the possession of King-Queen pair of the trump suit and system need to validate the claim before notifying other players and changing the bid.

| | |
|---|---|
| **Use Case** | Player claims the possession of King – Queen pair of the trump suit & system validates legality of claim before informing other players |
| **Precondition** | 1. The player who claims is signaled by system to play a card<br>2. Trump suit is already exposed to all players<br>3. Player's team already won a trick |
| **Basic Path** | 1. Player claims pair<br>2. System validates the claim<br>3. If validated system informs rest 3 players<br>4. Points bided are reduced or increased according to side of the player claims |
| **Alternate Path** | N/A |
| **Postcondition** | All players are informed & points bided are recalculated |
| **Exception Path** | 1. Player raises wrong claim<br>2. Bided points doesn't support pair show up by the player<br>3. System invalidate the claim |
| **Reference:** | SRS 2.2 / Figure 15 |

3.2.16 System should be able to decide the trick winner.

| | |
|---|---|
| **Use Case** | System decides trick winner |
| **Precondition** | All 4 players played their turn in the trick |
| **Basic Path** | 1. Ranking of cards for any suit is J,9,A,10,K,Q,8,7 in decreasing order of weightage , same for trump suit also<br>2. Any card of trump suit holds more weightage than any other suit card if trump suit is exposed at time of playing the card<br>3. System decides the player who played the highest weightage card to be the trick winner |
| **Alternate Path** | N/A |
| **Postcondition** | The player among 4 players who wins the trick is decided |
| **Exception Path** | N/A |
| **Reference:** | SRS 2.2 / Figure 16 |

3.2.17 System should be able to calculate points won by the trick winner.

| | |
|---|---|
| **Use Case** | System calculates points against trick winner |
| **Precondition** | Trick winner is decided |
| **Basic Path** | 1. Points given to cards as J -> 3 , 9 -> 2, A,10 -> 1, K,Q,8,7 -> 0<br>2. System adds up points for all 4 cards played in the trick<br>3. System stores the points against the trick winner |
| **Alternate Path** | N/A |
| **Postcondition** | Points won by the trick winner is stored against the player |
| **Exception Path** | N/A |
| **Reference:** | SRS 2.2 / Figure 17 |

### 3.2.18 System should be able to decide the winner of the game.

| | |
|---|---|
| **Use Case** | System decides the team which wins the game |
| **Precondition** | All 8 tricks should be completed |
| **Basic Path** | 1. System adds up points stored against the max point bidder and its partner<br>2. System compares the total points by them with the points set by the max bidder or updated point bid, if it is changed by pair show-up<br>3. If the total points won are equal or more then they wins the game otherwise looses |
| **Alternate Path** | If any player quits while game is on, then if the player belong to max bidders team the team looses otherwise opponent team wins |
| **Postcondition** | Two players who win the game is decided |
| **Exception Path** | N/A |
| **Reference:** | SRS 2.2 / Figure 18 |

### *3.3. Non-functional requirements*

1. Online 28 will work on any modern computer, and will thus be operating system independent.

2. Online 28 will be built using modules/layers that can interface with each other, thus allowing for extensibility and upgrades with minimal time involved.

3. Online 28 will be intuitive to use and will feature adequate instructions for specific implementations of the card game (meaning that it will not describe rules in depth, but will describe specific instructions for how to play specifically Online 28, for instance notifying the user on a wrong move). The game will have no more than a 30 second lag between user input and response.

4. The network should be available.

5. If the network connection fails in the middle of a game, Online 28 shall retry a fixed number of times and then display an error message.

### *3.4. System Evolution*

As a future scope the system can be extended to meet a few categories of interest of the users:

1. Pair-wise anonymous(pair challenge mode): this feature would enable 2 players to join the game as a pair and challenge any other anonymous pair.

2. GroupWise login: this feature would meet the category of interest of people who knows each other wants to play amongst themselves, enabling four people to engage in a game of Online 28 as a group.