# Software Requirements Specification

# for

# Blackjack

Version 1.0 approved

Prepared by Severin Klapproth, Flavia Taras, Luca Wolfart,  
Pascal Engeler, Noé Canevascini, Stanislaw Piasecki

Jack Blacks

10.03.2022

Table of Contents

[Introduction 1](#_Toc98255220)

[Purpose 1](#_Toc98255221)

[Document Conventions 1](#_Toc98255222)

[Intended Audience and Reading Suggestions 1](#_Toc98255223)

[Product Scope 1](#_Toc98255224)

[References 1](#_Toc98255225)

[Overall Description 1](#_Toc98255226)

[Product Perspective 1](#_Toc98255227)

[Product Functions 2](#_Toc98255228)

[User Classes and Characteristics 2](#_Toc98255229)

[Operating Environment 2](#_Toc98255230)

[Design and Implementation Constraints 2](#_Toc98255231)

[User Documentation 3](#_Toc98255232)

[Assumptions and Dependencies 3](#_Toc98255233)

[External Interfaces and Requirements 3](#_Toc98255234)

[User Interface 3](#_Toc98255235)

[Communications Interfaces 6](#_Toc98255236)

[System Requirements 6](#_Toc98255237)

[Functional Requirements 6](#_Toc98255238)

[FREQ-1: Game Server 6](#_Toc98255239)

[FREQ-2: Connection 6](#_Toc98255240)

[FREQ-3: Lobby 6](#_Toc98255241)

[FREQ-4: GUI 7](#_Toc98255242)

[FREQ-5: Game Start 7](#_Toc98255243)

[FREQ-6.1: Round Start 7](#_Toc98255244)

[FREQ-6.2: Place Swapping 7](#_Toc98255245)

[FREQ-7.1: Turns 7](#_Toc98255246)

[FREQ-7.2: Insurance #why is this 7.2??? 8](#_Toc98255247)

[FREQ-8: Make a Move 8](#_Toc98255248)

[FREQ-9: User Input 8](#_Toc98255249)

[FREQ-10: Dealer’s turn / Dealer AI 8](#_Toc98255250)

[FREQ-11: Round End & Returns 9](#_Toc98255251)

[FREQ-12: Game End 9](#_Toc98255252)

[FREQ-13: User Messages/Restrictions 10](#_Toc98255253)

[FREQ-14: Player Exit 10](#_Toc98255254)

[System Scenarios 11](#_Toc98255255)

[Use-case Diagrams 11](#_Toc98255256)

[Scenarios 11](#_Toc98255257)

[SCN-1: Starting a game 11](#_Toc98255258)

[SCN-2: Starting a round 13](#_Toc98255259)

[SCN-3: Playing a turn 13](#_Toc98255260)

[SCN-4: Ending a round 15](#_Toc98255261)

[SCN-5: End of the game 16](#_Toc98255262)

[System Constraints 17](#_Toc98255263)

[Important Nonfunctional Requirements 17](#_Toc98255264)

[NFREQ-1: Speed 17](#_Toc98255265)

[NFREQ-2: Game Cards and Layout 17](#_Toc98255266)

[NFREQ-3: Language 17](#_Toc98255267)

[NFREQ-4: Reliability 17](#_Toc98255268)

[NFREQ-5: Ease of use 17](#_Toc98255269)

[Other Requirements 17](#_Toc98255270)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Release Description | Version |
| **Felix Friedrich** | 16.03.22 | Template for Software Engineering Course in ETHZ. | 0.3 |
| **Jack Blacks** | 16.03.22 | Requirements Specification for Blackjack | 1.0 |

# Introduction

## Purpose

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>*

## Document Conventions

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

## Intended Audience and Reading Suggestions

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

## Product Scope

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>*

## References

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

# Overall Description

## Product Perspective

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

## Product Functions

*The idea of this project is to implement a slightly modified version of the card game – Blackjack. Our implementation adapts it to be a “real-money free” multiplayer game for 2-5 people with clearly defined win conditions. Broadly speaking, we will provide the functionality to:*

1. *Allow players to host a game on a server.*
2. *Allow players to join hosted games using our client.*
3. *Start game and play Blackjack, if enough players joined the host (at least one).*
4. *Track and inform the player about the current state of the game (dealt cards, amount of money each player has, round count, bet sizes).*
5. *Allow players to interact with the game interface using a GUI.*
6. *Allow players to make bets before round, where minimal bet is set to 5$.*
7. *Take turns, where player can choose to either “hit”, “stand”, “insure”, “split” or “double-down”, but only if this move does not violate the game rules.*
8. *Notify the player if his count goes above 21 during his turn.*
9. *Have the game carried by dealer controlled by the AI, which represents the “house”.*
10. *Notify the player, when he is out of money (cannot afford minimal bet), which means he lost.*
11. *Start and end rounds until reaching round cap (so 100 rounds), when winner is decided (if game is not finished by now, player with the most money wins).*
12. *Notify all players when win condition is fulfilled for some player (“last man standing”), so the game ends.*

## User Classes and Characteristics

*User class – regular player:*

1. *Knows and understands the standard rules of Blackjack.*
2. *Can access all features of the software.*
3. *Has no technical knowledge besides understanding how to run an executable file on a UNIX system, how to use GUI to interact with the game and how to start/join games by creating/connecting to a server.*

## Operating Environment

*The game should run on any modern computer with a UNIX-based system.*

## Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

## User Documentation

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

## Assumptions and Dependencies

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# External Interfaces and Requirements

## User Interface

As a user you will interact with the game through the Graphical User Interface (GUI). There will be a login menu, waiting lobby and a game interface. The interaction between the game and the user can be done through clicking on buttons or inserting text in appropriate places.

In the login menu the following inputs are available:

* Option to begin a game and name it.
* Choosing a username.
* Option to join an existing game.
* Boxes to enter existing server information.

The gaming interface must provide the following:

* Usernames of the other players.
* Amount of money each player has.
* Current bet sizes.
* Visible cards drawn by all players.
* Whose turn is currently taking place.
* When it is the users turn, the players available actions must be shown correctly.

Figures 1-6 are sketches of the different interfaces a user can have while playing the game. The different events which the user has to be made aware of, winning or losing a hand, playing and having the possibility to leave the game are all present. If a player loses all his money he cannot continue to play and has to leave the match.

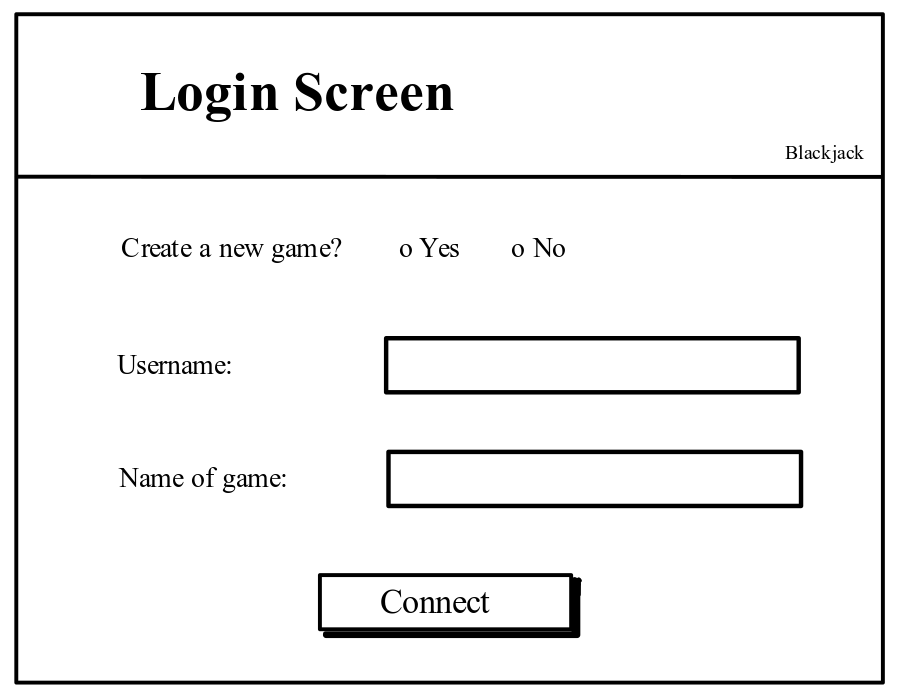
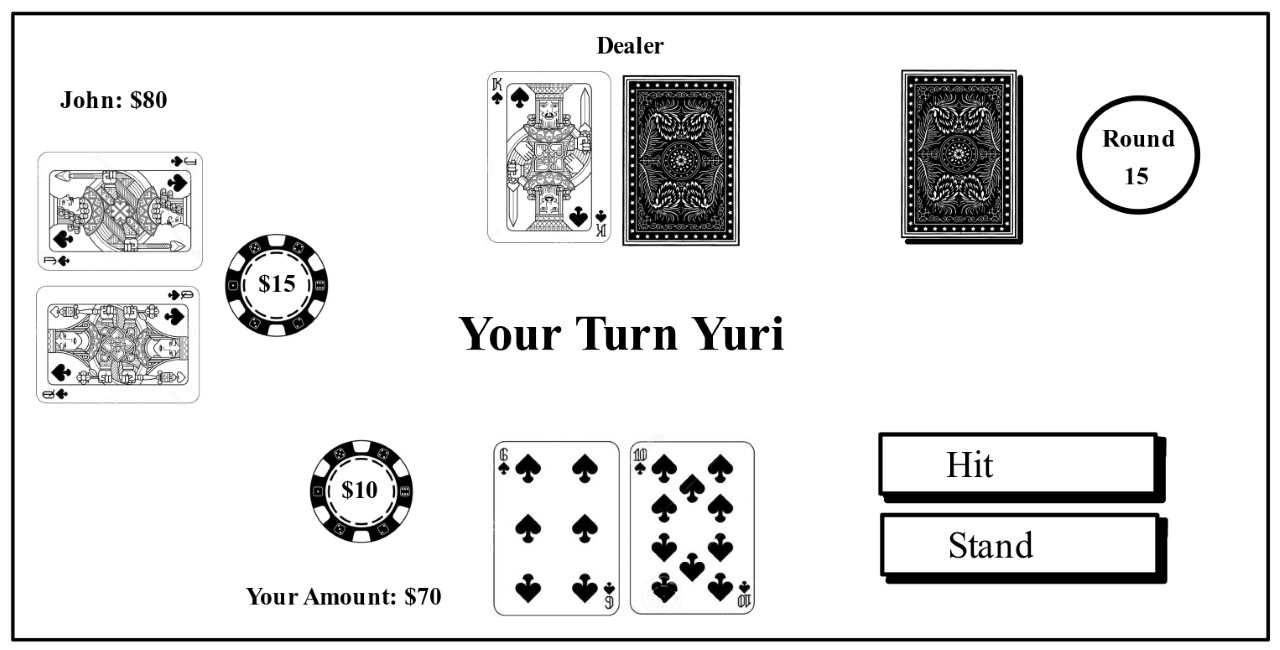


Figure 2: You are playing and it is your turn to decide.

Figure 1: The login screen.

Obraz zawierający tekst

Opis wygenerowany automatycznie

## Obraz zawierający tekst Opis wygenerowany automatycznie

Figure 4: An example of an error.

Figure 3: You lost and you’re all out of money. Your only option is to leave the game.

## Obraz zawierający tekst Opis wygenerowany automatycznieObraz zawierający tekst Opis wygenerowany automatycznie

Figure 6: You lost a round and you lose your bet.

Figure 5: You won this round and receive the money.

## Communications Interfaces

The interaction between the client and the server will be handled as follows: the communication will be between one server and multiple clients. Their communications will adhere to the TPC/IP protocol suite. The JSON format will be used to encode the data.

# System Requirements

## Functional Requirements

### FREQ-1: Game Server

The system runs a server to which up to 5 players can connect via the provided client.

**User Priority(5/5):** *the game cannot be played without this requirement.***Technical Priority(5/5):** *the game cannot be played without this requirement.*

### FREQ-2: Connection

The system provides a client which allows users to connect to the game lobby/server.

**User Priority(5/5):** *the users cannot join without this requirement.***Technical Priority(5/5):** *the game cannot be played without this requirement.*

### FREQ-3: Lobby

The system has a game lobby which can host up to 5 users at the same time. The lobby displays the currently joined players and allows the game to start if and only if at least 2 players are inside. The first player to join the lobby is considered the host.

**User Priority(5/5):** *the players need to be able to wait for each other before starting the game.***Technical Priority(2/5):** *the game cannot be played without at least two players. However, the lobby is technically not necessary to play the game.*

### FREQ-4: GUI

The system provides a GUI, displaying the game state (player and dealer cards, player pots, total player budgets), and possible actions (hit, stand, insure, split, double down).

**User Priority(5/5):** *makes it easier for the players to track the game.***Technical Priority(2/5):** *not strictly necessary in order to run the game.*

### FREQ-5: Game Start

The first round is initialized once the lobby host decides to start the game. Each player is provided with 100$ as a starting amount. A round counter is initialized.

**User Priority(5/5):** *the game cannot be started without this requirement.***Technical Priority(5/5):** *the game cannot be played without this requirement.*

### FREQ-6: Round Start

Each player has to place a bet or leave the game. The value of the bet has to be an integer, at least 5$ and is capped by each player's remaining money. Subsequently, each player receives 2 cards face-up, visible to all players. The dealer receives 2 cards, one facing up, one facing down. The person sitting next to the person who previously started, clockwise, starts the new round.

**User Priority(5/5):** *This requirement is necessary to play rounds.***Technical Priority(5/5):** *The game cannot be played without this requirement.*

### FREQ-7: Turns

The system keeps track of the player state and displays whose turn it is. The order in which the players perform actions will be clockwise. Only the player whose turn it is can choose to perform any actions (hit, stand, insure, split, double down). They can repeatedly do this until they choose to “stand” or the sum of their card values exceeds 21.

**User Priority(4/5):** *this makes it easier to keep track of whose turn it is.***Technical Priority(2/5):** *the users could keep track of this by themselves.*

### FREQ-8: Insurance

In the case that the up-card of the dealer is an ace, each player can decide whether they want to place an insurance bet.

**User Priority(3/5):** *this feature makes the* *game more enjoyable for users***Technical Priority(0/5):** *this is not relevant for the game functionality.*

### FREQ-9: Make a Move

During their turn, the system allows the user to repeatedly make a legal move until they decide to stand. After each legal move the game state is updated accordingly for everyone. Possible legal moves are:

• **Hit** (Take another card)

• **Stand** (Take no more cards and end the turn)

• **Split** (Create two hands from a starting hand where both cards are of the same rank. Each new hand gets another card so that the player has two starting hands. This requires an additional bet on the second hand. The two hands are played out independently and sequentially, and the wager on each hand is won or lost independently)

• **Double Down** (Increase the initial bet by 100% and take exactly one more card. The additional bet is placed next to the original bet. This can only be done at the beginning of a player’s turn.)

• **Insurance** (This is a side bet that the dealer has blackjack and can only be placed if the dealer’s visible card is an ace. Insurance bets are equal to half of the player's current bet. In case the dealer has a blackjack, insurance pays 2 to 1.)

(Sources for Player Decisions: <https://en.wikipedia.org/wiki/Blackjack#Player_decisions>)

**User Priority(5/5):** *the game cannot be played without this requirement.***Technical Priority(5/5):** *the game cannot be played without this requirement.*

### FREQ-10: User Input

The system allows the players to make inputs by clicking on the displayed move options in the GUI. It also allows the user to determine the value of the bet they want to place by interacting with the GUI through keyboard input/clicking.

**User Priority(4/5):** *this is very convenient.***Technical Priority(1/5):** *the inputs could also be delivered in a different way.*

### FREQ-11: Dealer’s turn / Dealer AI

After all players finish their turn, a program/AI decides the following moves of the dealer. First, the hole card of the dealer is now turned and visible to all players. They will continue to draw cards (hit) until they have 17 points or more. After that they will stand.

The dealer never splits or doubles down.

**User Priority(5/5):** *the game cannot be played without this requirement.***Technical Priority(5/5):** *the game cannot be played without this requirement.*

### FREQ-12: Round End & Returns

Once the dealer has ended their turn, the system calculates the return rates on the bets of every player. At this point, the system also needs to increment the internal round counter.

Bets have the following returns:

* If the player went over 21, had less points than the dealer or if the dealer had a blackjack and the player didn’t, the player loses the money they bet.
* If the player ties with the dealer (same amount of points or both dealer and player have blackjacks), the player receives their main bet back.
* If the player wins by having more points than the dealer, they receive two times their main bet.
* If the player wins with a blackjack, they receive 1.5 times their main bet.

Insurance:

* If the player made the insurance bet at the beginning of their turn and the dealer has blackjack he receives two times the insured amount. Otherwise, he loses the money invested in the insurance bet.

After having made the calculations, the system updates the budget of every player and displays the winnings of each player until a new round starts.

**User Priority(5/5):** *the game cannot continue without this requirement.***Technical Priority(5/5):** *the game cannot be played without this requirement.*

### FREQ-13: Game End

After each round the system checks…

* … if a player went bankrupt. If so, they have lost and can only continue the game as a spectator.
* … if all but one player went bankrupt. If yes, the game ends and the only player who has money left is declared the winner.
* … if the internal round counter has reached the chosen maximum number of rounds. If so, the game ends and the player with the most money wins.

**User Priority(4/5):** *this is an important part of the game flow and it is convenient for the players if the system keeps track of this.***Technical Priority(2/5):** *the game needs to end at some point, but the users could keep track of this event by themselves.*

### FREQ-14: User Messages/Restrictions

The system restricts the player by changing the mouse-interaction options inside the GUI, depending on their cards and budget, such that the selection of illegal moves is not even a possibility.

If the user tries to enter an illegal move via keyboard or mouse the system displays an error message.

**User Priority(3/5):** *gives the user more information why something is not possible.***Technical Priority(1/5):** *not necessary for the game to run.*

### FREQ-15: Player Exit

The system continues running even if one or more of the game clients exit, crash or disconnect.

**User Priority(3/5):** *it is inconvenient if your client crashes because someone else disconnected.***Technical Priority(3/5):** *necessary for the system in order to keep the game state, but the state might be irrelevant if the game cannot continue due to a missing player.*

# System Scenarios

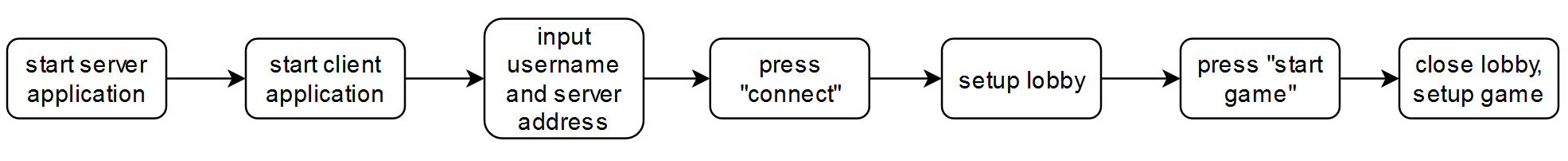
## Use-case Diagrams

## Scenarios

### SCN-1: Starting a game

|  |  |
| --- | --- |
| FREQ reference | *<Refer to the Functional Requirement numbers this scenario uses.>* |
| NFREQ reference | *<Refer to the Non Functional Requirement numbers this scenario uses. If none then write NA.>* |
| Short Description: | John, Jane and Peter want to play a game of Blackjack. John starts the server and communicates the address to Jane and Peter. All three start their clients, enter their usernames and connect to the server by filling in the address and pressing the “connect” button. Jane presses the “start game” button. |
| Activation action: | John starts the server, all players connect, Jane presses the “start game” button. |
| Precondition: | All players have the ability to run and control the game client, John also has the ability to run the server. All players have access to the same network as the server. |

|  |  |  |  |
| --- | --- | --- | --- |
| Basic flow: Starting a game | | | |
| Step | User action | | System response |
| 1 | John starts the server application | | The server application starts and displays its address. |
| 2 | Players start the client | | The client application starts and shows text input fields for username, and for a server address |
| 3 | Players fill in the required information | |  |
| 4 | Players press the “connect” button | | All players are connected to the server and added to the lobby, the connected players are displayed |
| 5 | Jane presses the “start game” button | | The lobby is closed, a new game is initialized with all players that were present in the lobby, the amount of money of each player is set to 100, a shoe is initialized, players are assigned seats and the game starts by prompting each player to specify a bet. |
| Post-condition: | | All players are connected to the server, the game is initialized and players have the ability to specify their desired bets for the first round. | |

Scenario Diagram for SCN-1 Starting a game

### SCN-2: Starting a round

|  |  |
| --- | --- |
| FREQ reference | *<Refer to the Functional Requirement numbers this scenario uses.>* |
| NFREQ reference | *<Refer to the Non Functional Requirement numbers this scenario uses. If none then write NA.>* |
| Short Description: | Peter, Jane and John have just finished a round. All of them have a finite amount of money left. They each enter their bets for the next round. The next round starts, it’s Peter’s turn. |
| Activation action: | A round has just ended all players still have money. |
| Precondition: | John, Jane and Peter are playing Blackjack and a round has just ended. Peter sits on first base. |

|  |  |  |  |
| --- | --- | --- | --- |
| Basic flow: Starting a round | | | |
| Step | User action | | System response |
| 1 | All players specify their desired bets for the next round | | The bets are displayed, the bet of each player is subtracted from their money. |
| 2 |  | | Checks if a new shoe needs to be initialized. Initializes new shoe. |
| 3 |  | | Each player is dealt two cards face up, the dealer is dealt one card face up and one card face down. The dealer’s face up card is a 7 of spades. It’s displayed that Peter is to play. |
| Post-condition: | | A new round has started. Two cards are shown face up for each player, for the dealer one card is shown face up and one face down. It’s Peter’s turn. | |

Scenario Diagram for SCN-2 Starting a round

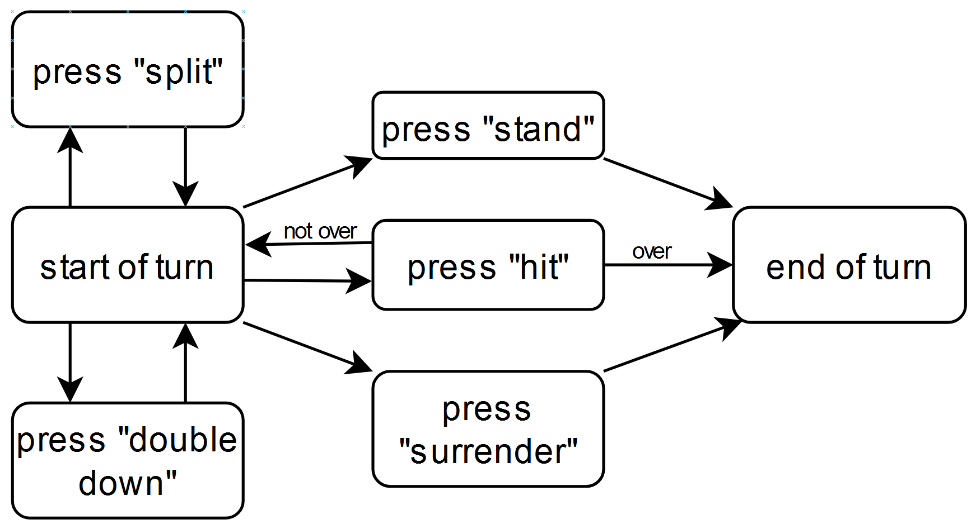
Obraz zawierający tekst

Opis wygenerowany automatycznie

### SCN-3: Playing a turn

|  |  |
| --- | --- |
| FREQ reference | *<Refer to the Functional Requirement numbers this scenario uses.>* |
| NFREQ reference | *<Refer to the Non Functional Requirement numbers this scenario uses. If none then write NA.>* |
| Short Description: | Jane, John and Peter are playing a round of Blackjack. Peter sits on first base and has already finished his turn by pressing “stand”. It’s Jane’s turn, she sits at second base. Her two face up cards are a 6 of spades and a 2 of hearts. She presses ‘hit’ and receives an ace of hearts. She presses the “Stand” button. Her turn ends, it’s John’s turn. |
| Activation action: | Peter presses “stand”. |
| Precondition: | Jane, John and Peter are playing a round of Blackjack. Peter’s turn has just ended, it’s Jane’s turn. |

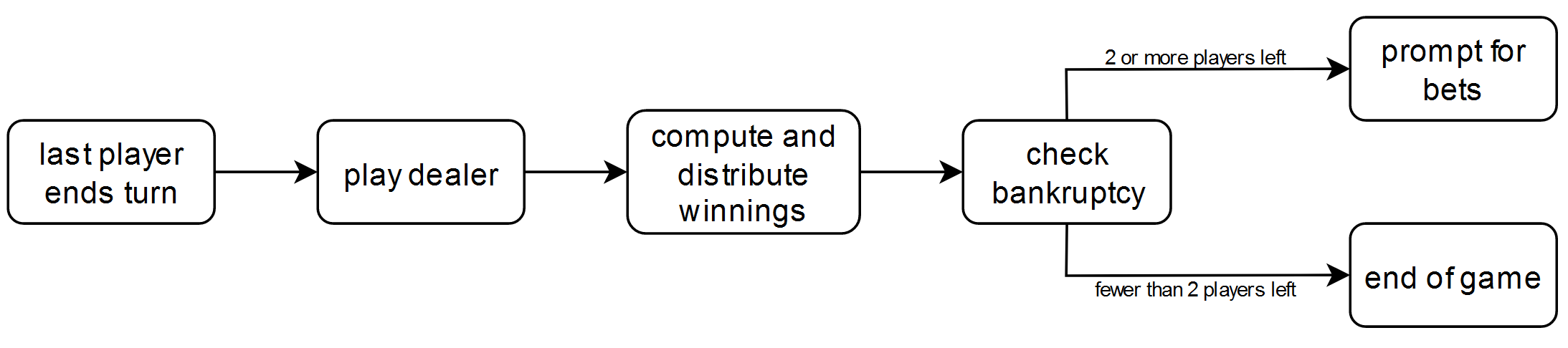
|  |  |  |  |
| --- | --- | --- | --- |
| Basic flow: Playing a turn | | | |
| Step | User action | | System response |
| 1 | Peter presses “stand” | | Peter’s score is calculated and displayed. Checks who’s turn it is and displays that it’s Jane’s turn. |
| 2 | Jane presses “hit” | | The next card is drawn from the shoe and dealt to Jane. The card is displayed face up. Jane’s intermediate score is calculated, it’s less than or equal to 21, so her turn continues. |
| 3 | Jane presses “stand” | | Jane’s final score is calculated and displayed. Checks whose turn it is next and displays that it’s John’s turn. |
| Post-condition: | | It’s John’s turn. Jane’s score is displayed. | |

Scenario Diagram for SCN-3 Playing a turn

### SCN-4: Ending a round

|  |  |
| --- | --- |
| FREQ reference | *<Refer to the Functional Requirement numbers this scenario uses.>* |
| NFREQ reference | *<Refer to the Non Functional Requirement numbers this scenario uses. If none then write NA.>* |
| Short Description: | Peter, Jane and John are playing a round of Blackjack. All players have a bet of 50. Peter and Jane have already finished their turns and achieved scores of 19 and 20, respectively. It’s John’s turn. John has a jack of hearts and a king of spades. He presses “hit” and receives a 6 of clubs. John loses, and his turn ends. The dealer’s face down card is turned around. The dealer’s score is 16, so she is dealt another card. It’s a 2 of spades. The dealer has a score of 18 and the round ends. 100 is added to Peter’s and Jane’s money. All players still have money. The players are prompted for their next bets. |
| Activation action: | John presses “hit”. |
| Precondition: | John is the last to play in the round and starts with a score of 20. All players have money left. |

|  |  |  |  |
| --- | --- | --- | --- |
| Basic flow: Ending a round | | | |
| Step | User action | | System response |
| 1 | John presses “hit” | | The next card off the shoe is dealt face up to John, it’s a 6 of clubs. The intermediate score of John is calculated and compared to 21. It’s greater, a notification is shown. John’s score is displayed and his turn ends. All players have finished their turns. |
| 2 |  | | The dealer’s face down card is displayed. The dealer’s score is calculated. It’s 16. The dealer is dealt another card. It’s a 2 of spades. The dealer’s score is calculated, it’s 18. The dealer is finished, her score is displayed. |
| 3 |  | | All players’ scores are ranked. 100 is added to Jane and Peter’s money. A notification of who won how much is shown. |
| 4 |  | | The system checks if a player has gone bankrupt. The round ends and players are prompted to specify their next bets. |
| Post-condition: | | 100 has been added to Jane’s and Peter’s money. The bet entry prompt is shown. | |

Scenario Diagram for SCN-4 Ending a round

### SCN-5: End of the game

|  |  |
| --- | --- |
| FREQ reference | *<Refer to the Functional Requirement numbers this scenario uses.>* |
| NFREQ reference | *<Refer to the Non Functional Requirement numbers this scenario uses. If none then write NA.>* |
| Short Description: | *John and Jane are playing a game of Blackjack. Both have a bet of 5, Jane’s remaining money is at 200 while John’s is at 0. In the current round, John has already finished playing and achieved 15 points. Jane was dealt 20 points and decides to stand. The dealer achieves 19 points. Jane’s money is increased by 10. John lost and Jane is the only player to have money left. The game ends. Jane wins the game. John closes the game client.* |
| Activation action: | *Jane presses “stand”* |
| Precondition: | *John has no more money left.* |

# System Constraints

## Important Nonfunctional Requirements

### NFREQ-1: Speed

Every action of the players should be processed in less than one second.

**User priority(4/5):** speed is essential for fun gameplay.

**Technical priority(1/5):** speed doesn’t influence the functioning of the system

### NFREQ-2: Game Cards and Layout

A classic poker deck should be displayed. The images displayed should be similar to those seen in a casino, but there can be some deviations. For example, if the players cards are displayed around a semicircle in the casino, in the GUI the card can be displayed in a linear fashion.

**User priority(3/5):**  visual inputs/outputs make the game more enjoyable.

**Technical priority(1/5):** visuals do not influence functioning of the system.

### NFREQ-3: Language

The system should display all texts in English.

**User priority(4/5):** the product is intended for an international audience, so understandability is important. **Technical priority(2/5):** texts are designed for the end user but could be useful to solve problems if the system has issues.

### NFREQ-4: Reliability

The system should crash at most once every 50 games.

**User priority(4/5):** reliability is important as users would be unsatisfied with the product if it crashed often.

**Technical priority(5/5):** if the game crashes all the other component of the software can’t work

### NFREQ-5: Ease of use

A new player should be able to find the button to execute the next action within 5 seconds. This action doesn’t have to be a tactically good one, so the 5 seconds should be used only to find the button to proceed in the game.

**User priority(4/5):** ease of use promotes fun gameplay.

**Technical priority(1/5):** ease of use doesn’t influence other components of the system

# Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

Appendix A: Glossary

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

Appendix B: Analysis Models

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>*

Appendix C: To Be Determined List

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*