Software Requirements Specification

for

Durak

Version 0.1 approved

Prepared by Danil Poluyanov, Noah Rutz, Eric Holenstein, Thomas Waldmann and Vito Maiocchi

Tubbeli Trupp

1.10.2024

Table of Contents

Revision History

Name	**Date**	**Release Description**	**Version**
Felix Friedrich	11.10.24	Template for Software Engineering Course in ETHZ.	0.3
Team	1.10.24	Draft of Durak	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

This project will implement a multiplayer game based on the most well-known Russian card game Durak. It will have the following functions:

- Starting a game server
- Let players join the server. For a match to start there must be at least two and at most six players.
- When all players have connected the host will be able to start the match
- The program will then:
  - Shuffle a deck of 52 (French) cards and generate a stack of cards which will then be dealt to players two at a time until all the players have six cards
  - Determine the Trump from the next card in the stack according to its suit (either Hearts, Diamonds, Clubs or Spades) and place it visibly on the playing field
  - Determine the first attacker based on the lowest trump card
  - By means of the GUI allow the players to make an attack, join an attack, defend an attack or pass on an attack (‚Weiterleiten')
  - Handle card drawing and discarding
  - Allow the player to end their turn and then automatically let the according player start the next turn
  - Automatically end a turn and start the next when a player cannot defend himself
  - Display the game state, including players cards (hidden from opponents), the number of cards every player has, the trump suit and the discard pile
  - At any time allow the player to overlay the rules of the game
  - Enforce the rules of the game, including valid attacks and defenses
  - Automatically end a game when the second to last person has played all their cards
  - Allow for a rematch or exit to the lobby at the end of a game

User Classes and Characteristics

New Player:

- This user class has never played Durak before
- User can access all game features and is encouraged to read the rules
- User does not need to have technical knowledge of the program, besides launching the program and navigating the GUI

Average Player:

- This user class is familiar with the rules

- User can access all game features and plays to the best of his abilities without going too deep into ‚Mind Games’
- User does not need to have technical knowledge of the program, besides launching the program and navigating the GUI

Expert Player:

- This user has experience in the game
- User can access all game features and may perform moves that are considered illogical at first but does so if it helps in the long run
- User does not need to have technical knowledge of the program, besides launching the program and navigating the GUI but will probably be the fastest to make his moves as he has routine

Operating Environment

The program will be developed to run on a UNIX-based system on a modern computer.

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).>

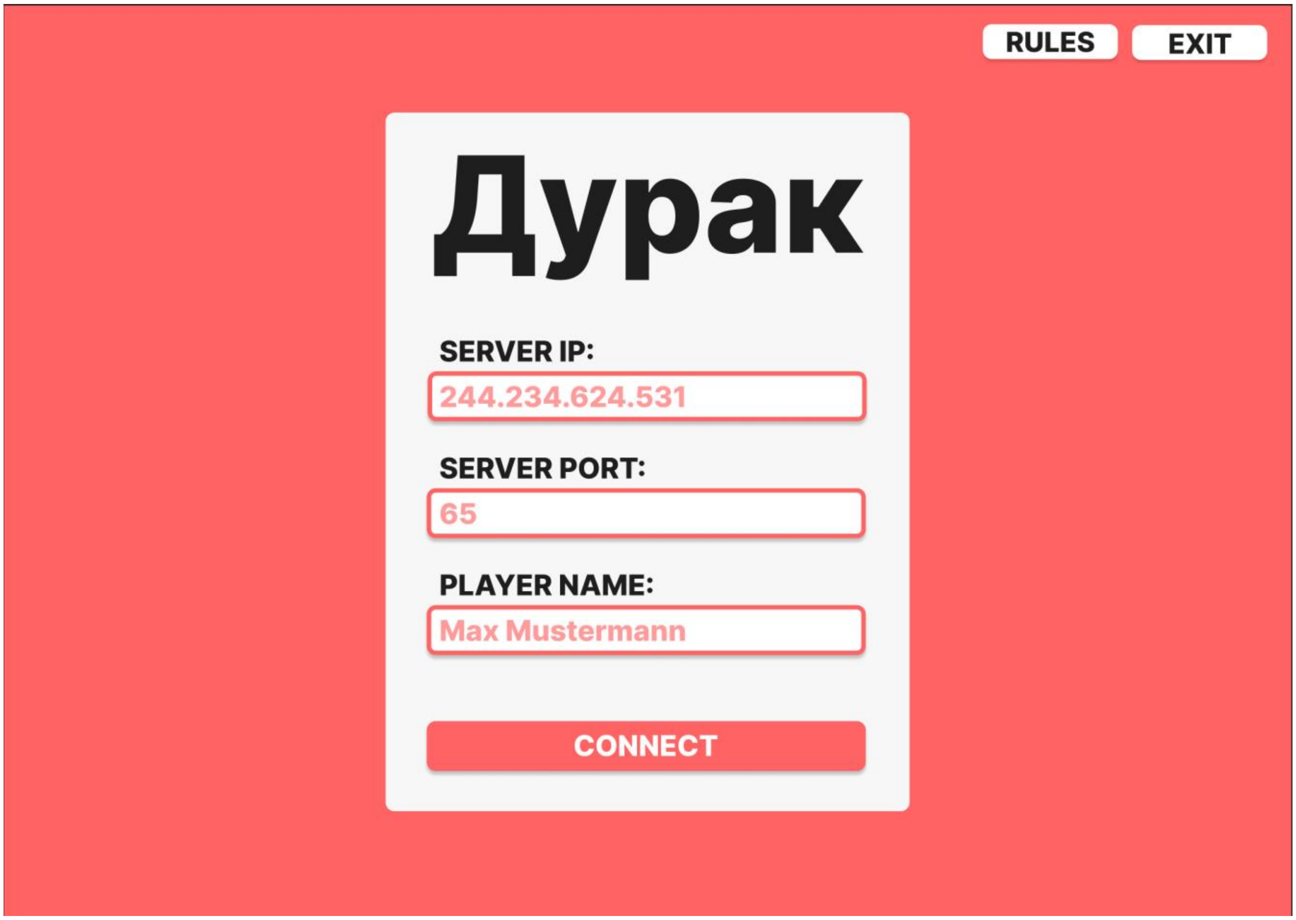
External Interfaces and Requirements

User Interface

1. **Login screen:**

The login screen is visible when starting the game client. There is a button that displays the game rules when clicked. This screen also includes fields to setup connection to the game server. The player will also be able to set a username and join a game.

2. **Lobby screen:**
   This screen appears once the user successfully joined a game. From here the user can go to settings. The user can press the 'START GAME' button when ready, the game starts once all players are ready.
3. **Settings:**
   Here the user will be able to change the language.
4. **Rules:**
   Here the user will be able to see the rules.
5. **Gameboard view:**
   The gameboard is a red background with six double tiles in the middle indicating where the cards will be. On the top right there are options to see the rules, give up or exit
6. **Attacking/Defending/Observing screen:**
   During the game all players can see their own cards. Additionally, they will be able to see: The usernames of all the other players, how many cards each of the other players hold, how many cards are left on the pile, the current trump (through the last card in the stack or the indicator on the right) and what cards are involved in the current attack. There is a text displayed that shows you if you are currently attacking defending or spectating, also little icons next to your opponent's names will show whether they are attacking defending or spectating. Depending if you are attacking, defending or observing there will be a button on the right of your cards, for attackers it displays 'DONE' they will press it to finish their turn, for defenders it will display 'PICK UP' to pick up the cards.
7. **End screen:**
   After the game ends, the end screen will give you the options to go to the lobby or rematch.

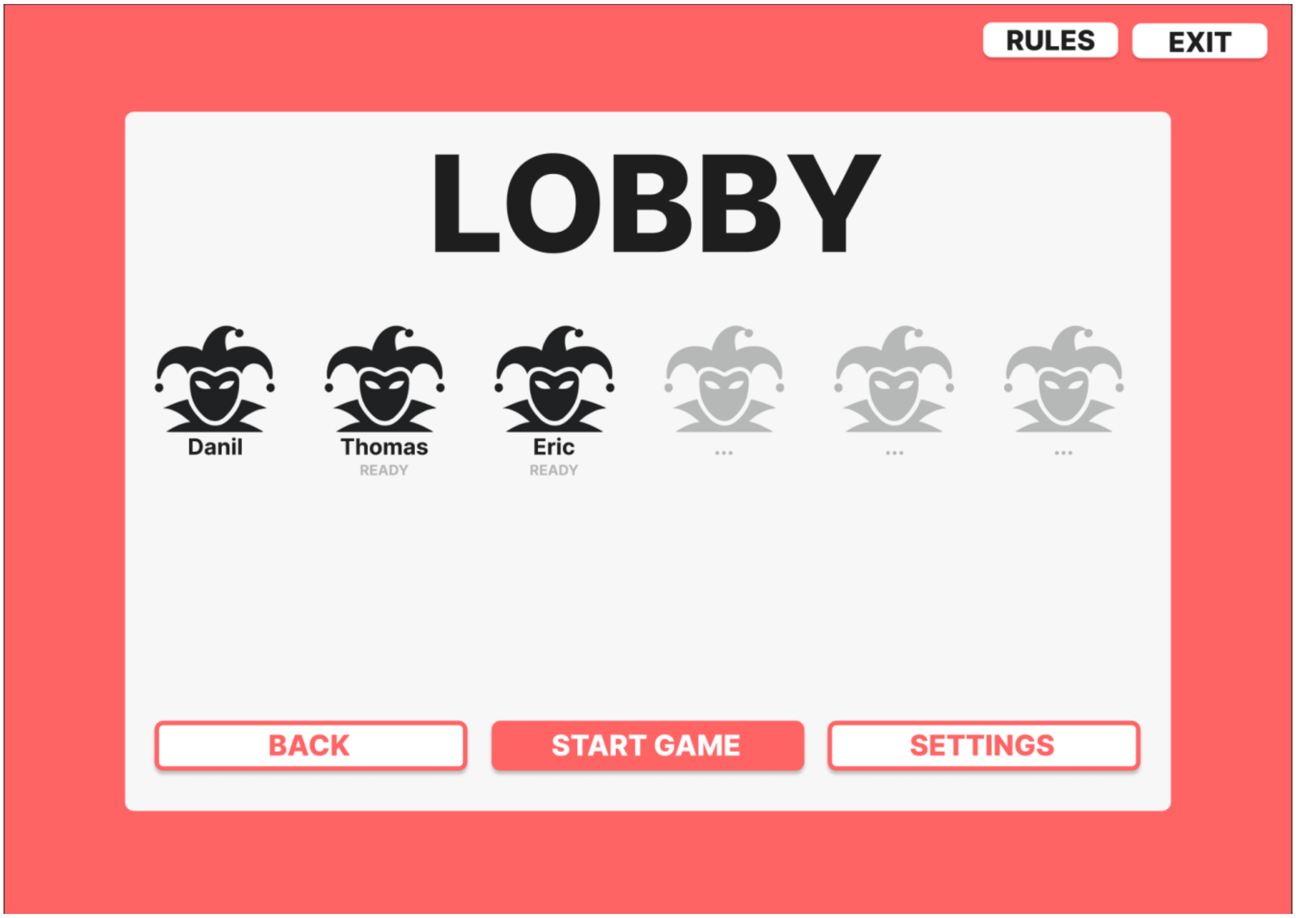


SERVER IP: 244.234.624.531

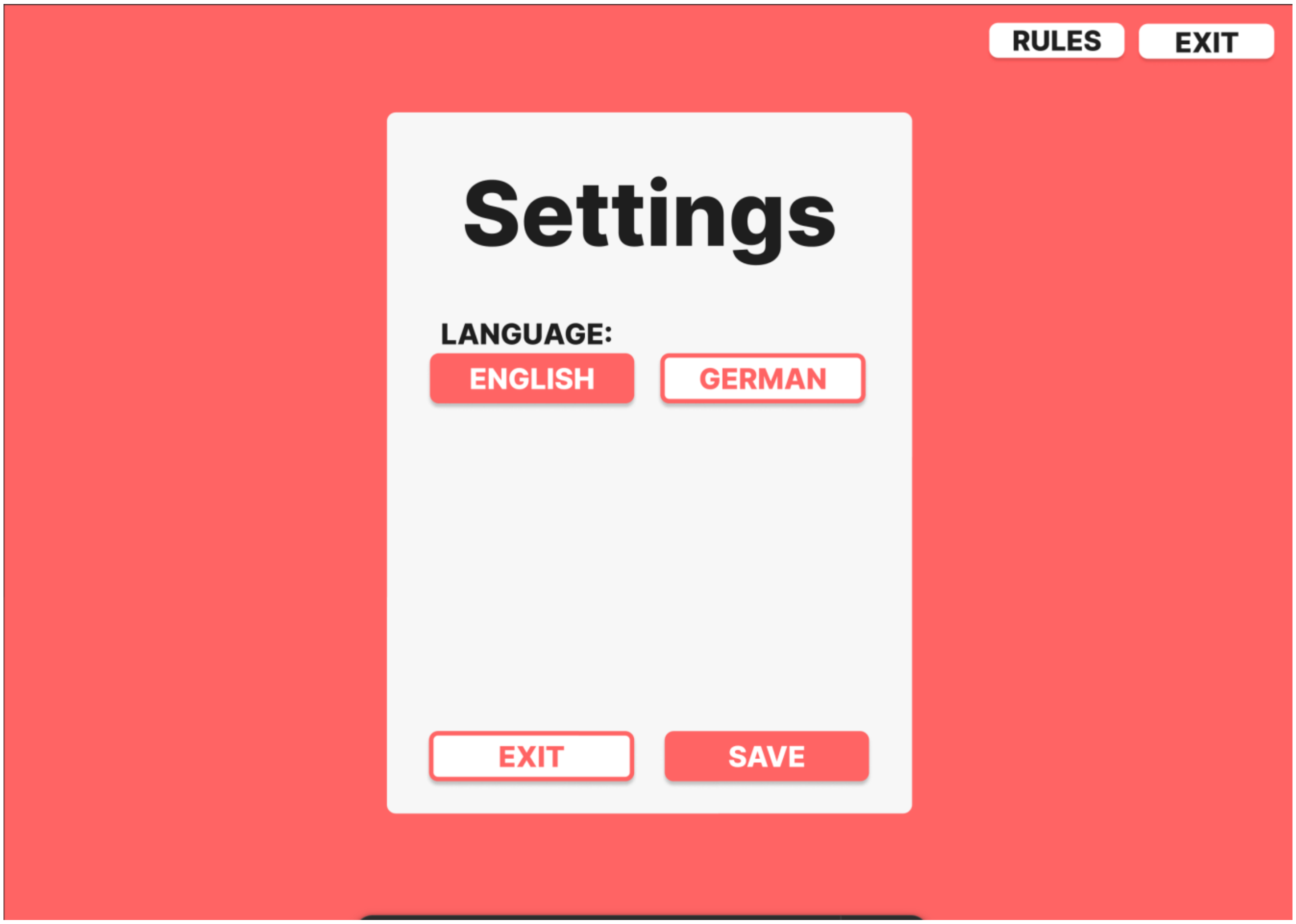
SERVER PORT: 65

PLAYER NAME: Max Mustermann

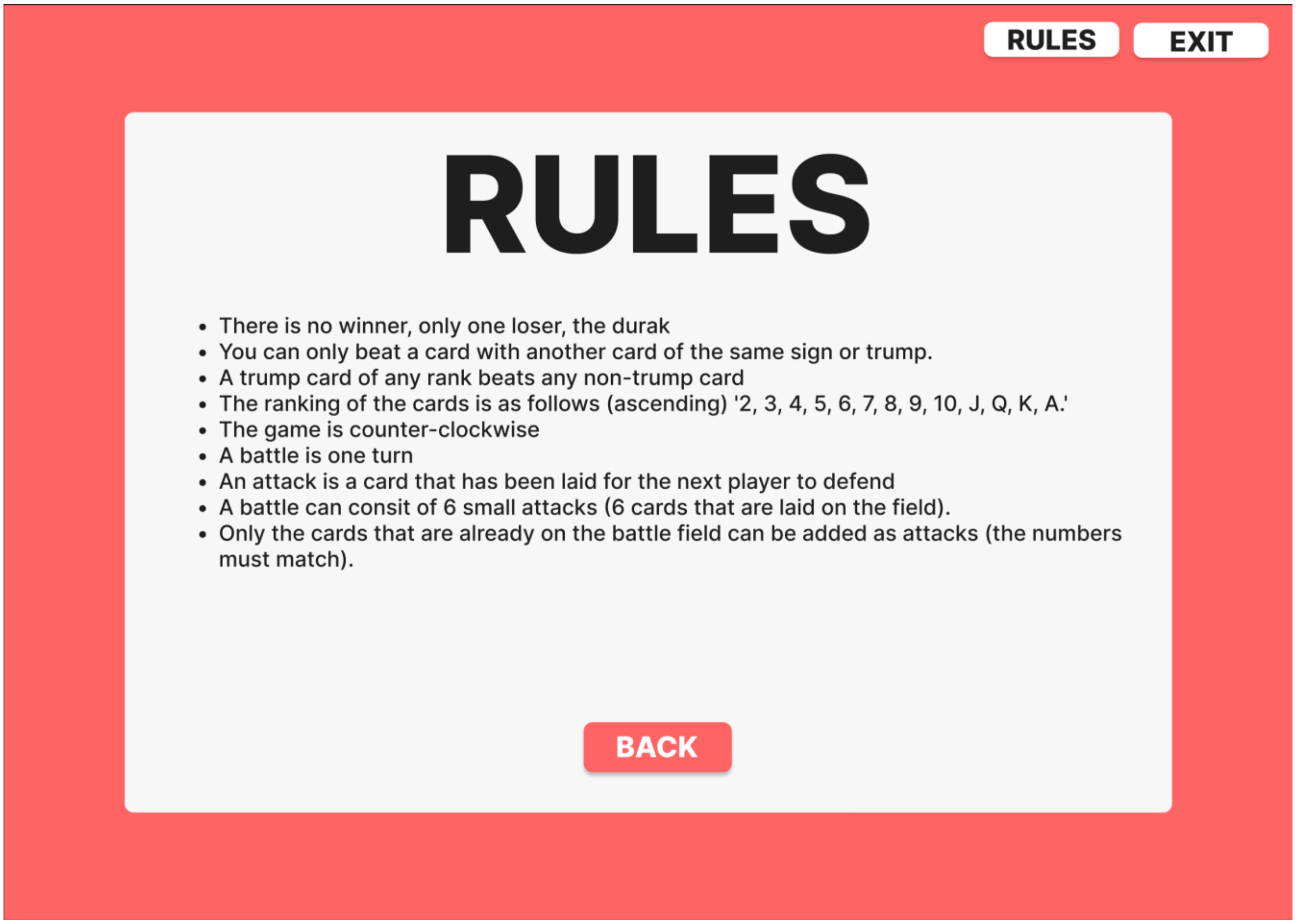
Login screen



Lobby screen



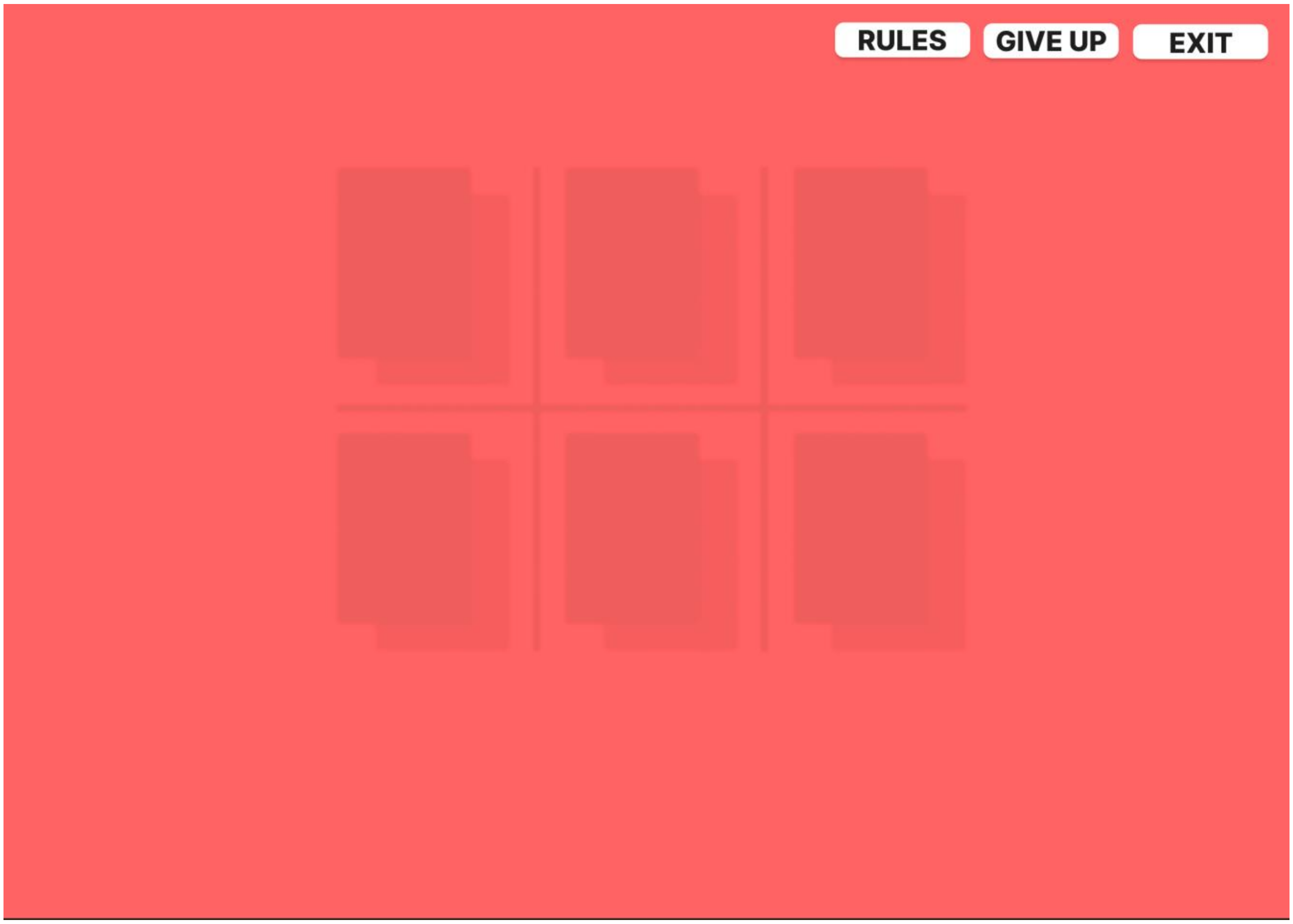
Setting screen



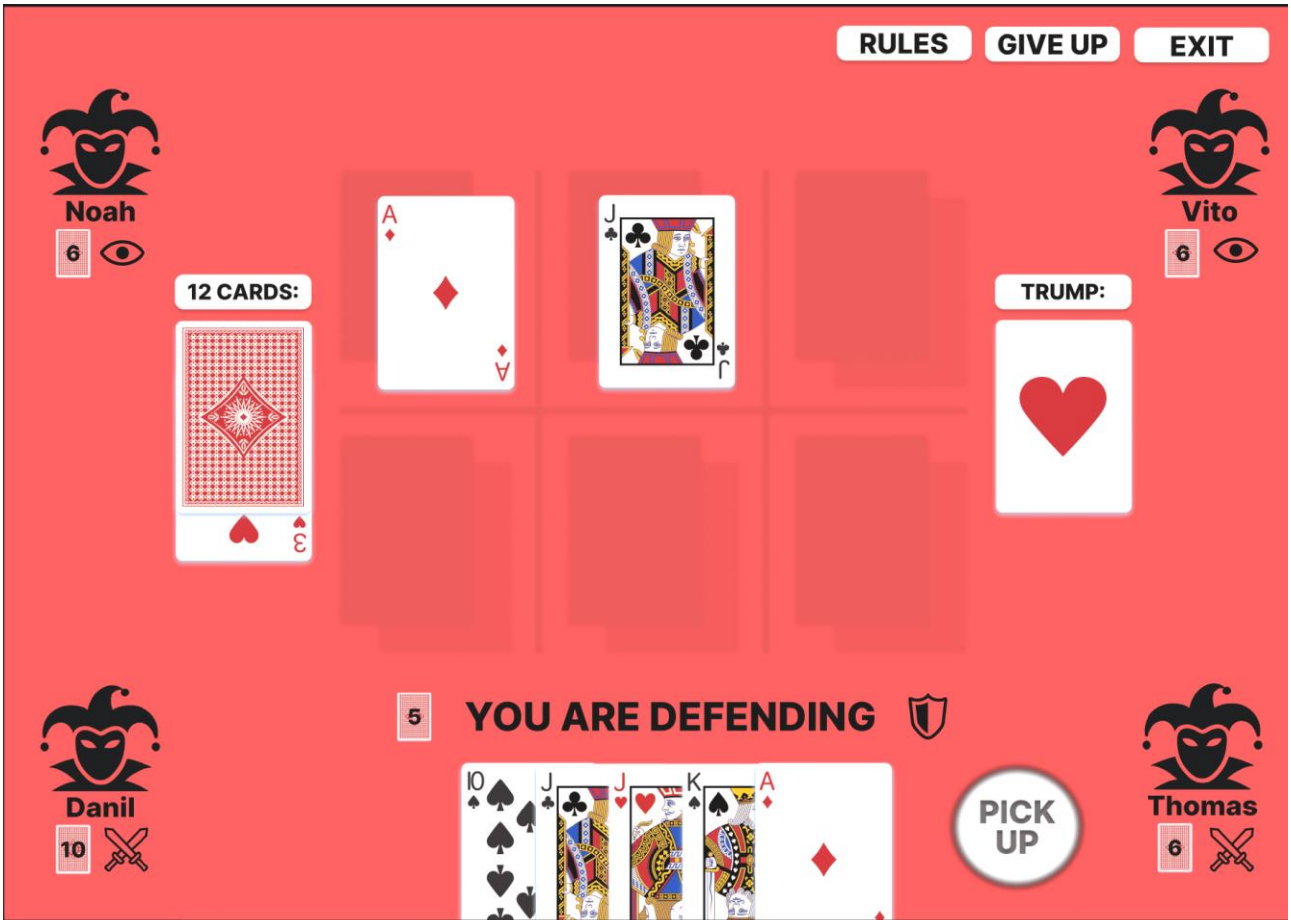
RULES

- There is no winner, only one loser, the durak
- You can only beat a card with another card of the same sign or trump.
- A trump card of any rank beats any non-trump card
- The ranking of the cards is as follows (ascending) '2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A.'
- The game is counter-clockwise
- A battle is one turn
- An attack is a card that has been laid for the next player to defend
- A battle can consit of 6 small attacks (6 cards that are laid on the field).
- Only the cards that are already on the battle field can be added as attacks (the numbers must match).

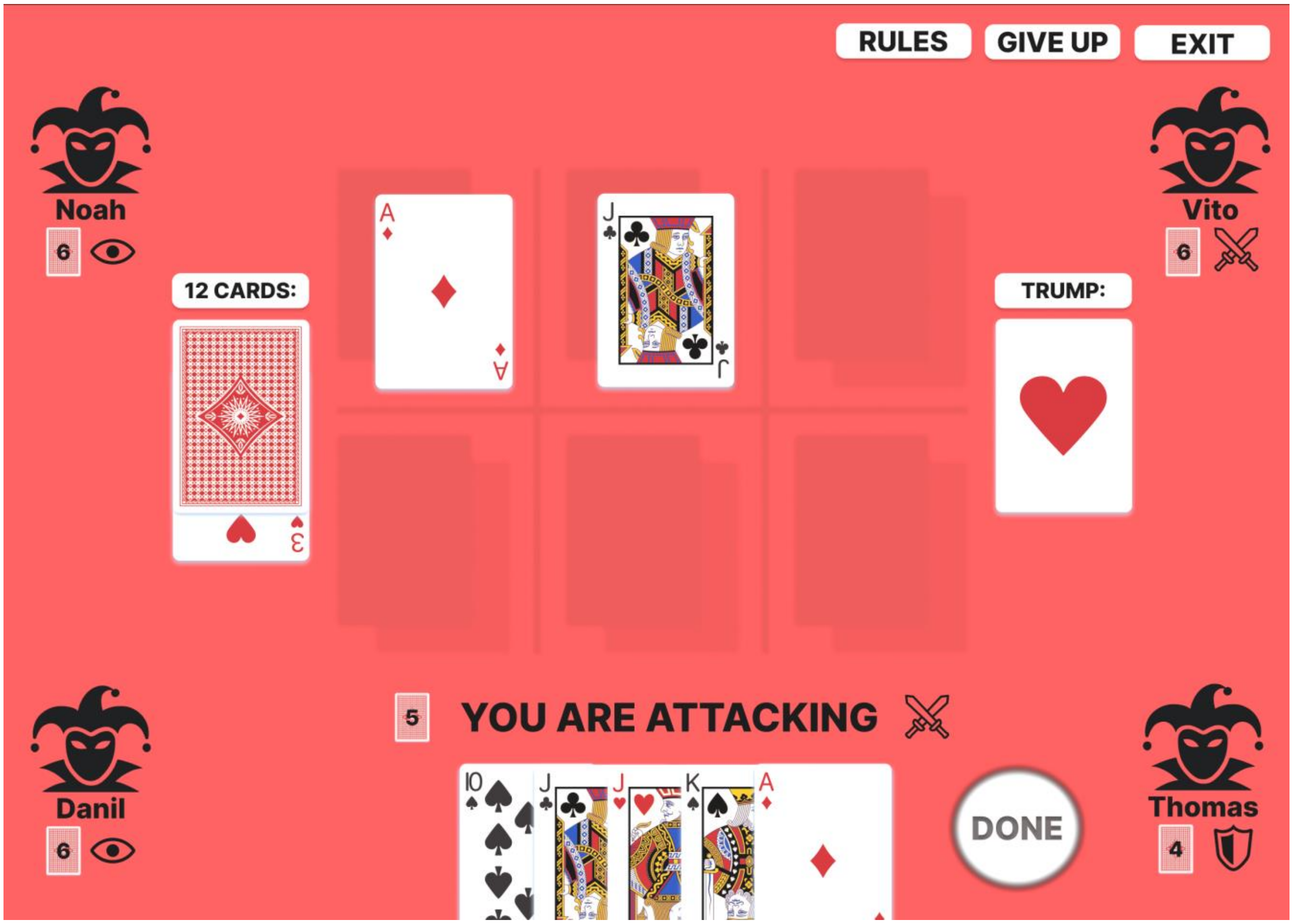
Rules screen



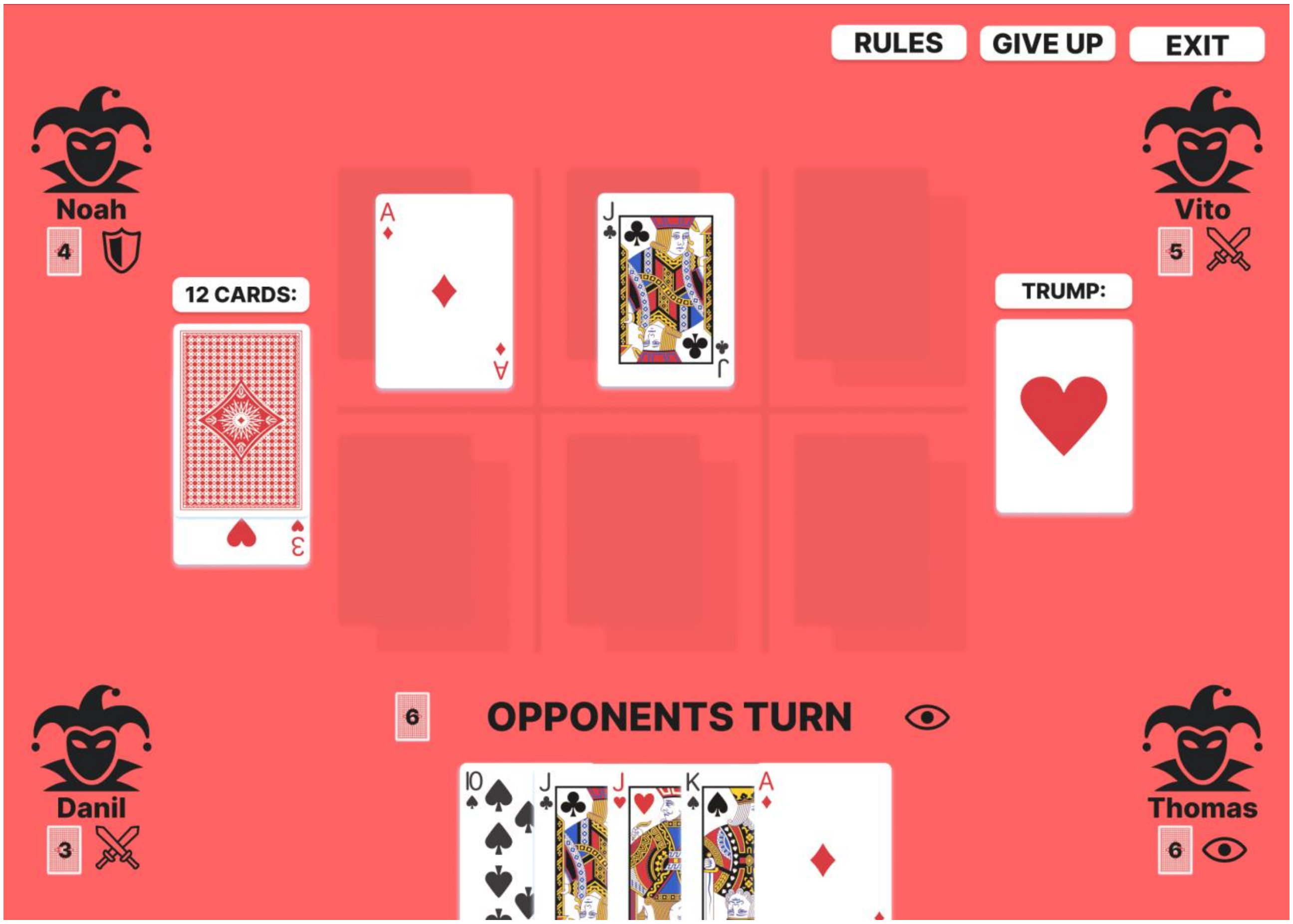
Gameboard screen



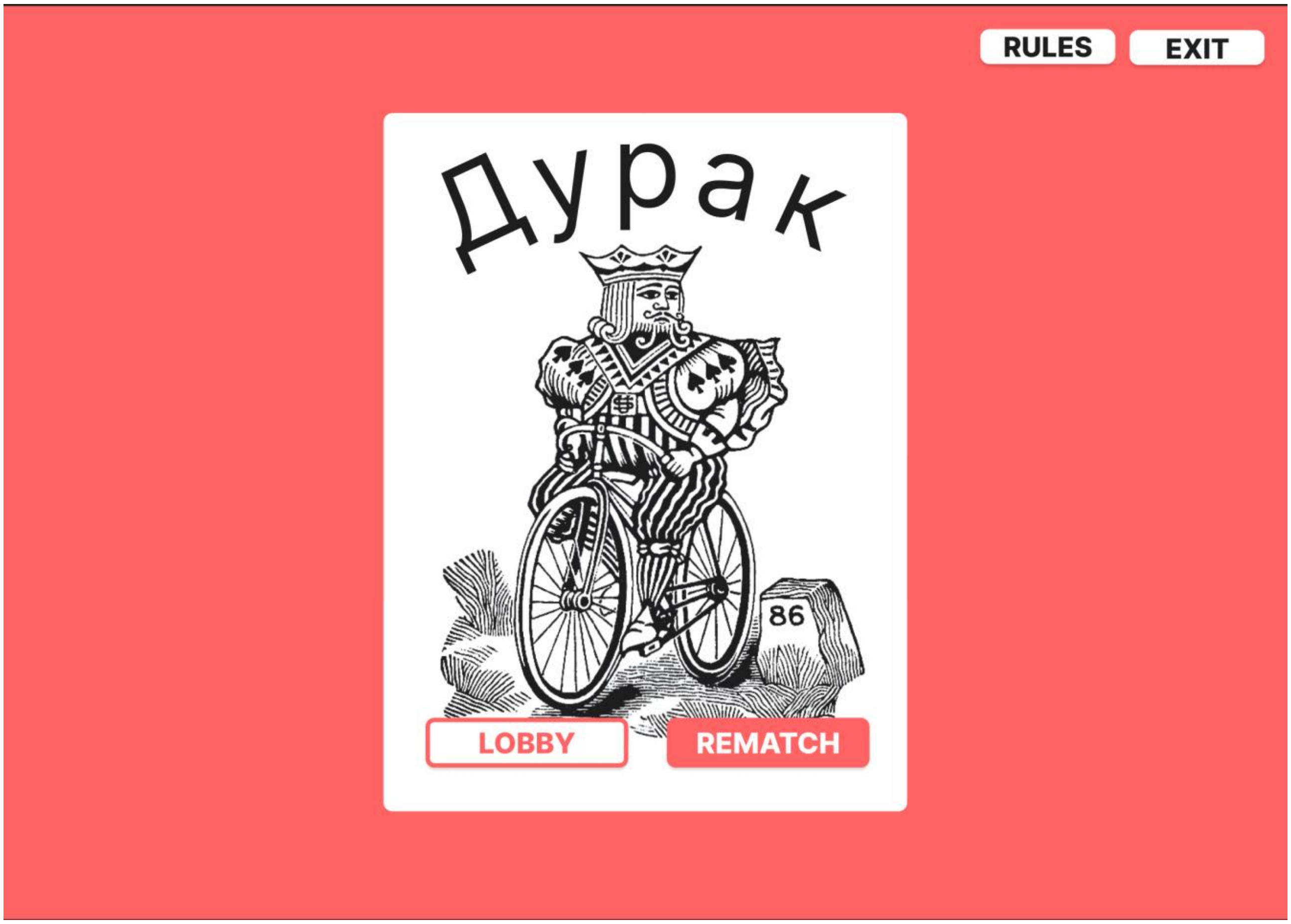
Screen during defending



Screen during attacking



Screen during opponents turn



End screen

Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

Communications Interfaces

The system will implement client-server communication between one server and multiple clients. Server-client communication will use TCP to ensure correct, reliable and in-order communication between both parties. The data will be encoded in the JSON format.

System Requirements

Functional Requirements

FREQ-1: Game Server

The system should offer the functionality to start and run a server to which players can connect via 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 should provide clients which allow the users to connect to the game server and join the game lobby.

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: GUI

The system should provide a GUI during the whole game for each user, displaying the game state (players, cards, piles and scores) and moves.

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-4: Lobby

The system should provide a game lobby with up to 6 users at the same time. The lobby should display the currently joined players. In the lobby the players can either close the game, go to settings, view the rules or start the game. The game starts by pressing the 'START GAME' button. The game can start if and only if there are 3-6 players and all players have pressed the 'START GAME' button.

User Priority(5/5): The players need to be able to wait for each other and press the 'START GAME' button.
Technical Priority(2/5): The game cannot be played without having enough players and thus waiting for at least three players is important. However, the lobby is technically not necessary to play the game.

FREQ-5: Game Start

The system should properly shuffle and deal six cards to every player at the start of each game. Also the game should choose the top card of the remaining stack to determine the suit that will be trump for this round. This card will be displayed for all users to see and will also be the last card to be drawn from the stack of cards.

User Priority (5/5): Randomness is needed for fairness and replayability of the game.

Technical Priority (5/5): The game cannot be played without this requirement.

FREQ-6: First Move

The system should automatically determine who is the first player to attack (the player with the lowest trump card) and who is the first player to defend (attacks counterclockwise always). If no player has a trump card the system should determine the first player to attack at random.

User Priority (5/5): The players need to know who starts to ensure that the game is played correctly.
Technical Priority (5/5): The game cannot be played if it isn't determined who is going to start.

FREQ-7: Turn-Based Gameplay with Time Limits

The system should enforce turn-based gameplay, where each player has a limited time to attack or defend. It should notify the players whose turn it is. The system should enforce time limits on each player's turn (if applicable), with a countdown visible. Since Durak is a game where cards can quickly be thrown in by multiple players, the system needs to be able to determine in what order cards were played.

User Priority (5/5): The turn-based mechanic defines Durak, and time limits ensure a smooth flow of the game.
Technical Priority (5/5): Core game mechanics must work consistently to maintain gameplay integrity.

FREQ-8: Attacking

When a player is allowed to attack (either by starting an attack or joining someone else's attack) the player should be able to:
- Play any card to start an attack (if it is the players turn to attack) or multiple cards of the same number
- Add a number-wise matching card to the pool of cards currently in the game
- Signal that the attacker does not want to make any more moves using the 'DONE' button

The system should check that a player is not attacked by more than 6 cards per round.

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-9: Defending

When a player is being attacked, the player should be able to:
- Pass on a card as long as it is the first attack on the player, the player being attacked has a number-wise matching card to the card played, and the next player has enough cards to receive the attack.
- "Defending" an attack by playing a higher card of the same suit or by playing a trump card
- At any time picking up the cards currently in the pool because the player can't defend them or doesn't want to defend them. This can be done using the 'PICK UP' button
- If the defender chooses to pick up the cards, the two attackers can still throw in number-wise matching cards until a timer expires as long as the total amount of attacks doesn't exceed 6.

It should not be possible to be attacked with more than 6 cards per round.

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: Observing:

While a player is neither attacking nor defending (either because the player is not involved in the current attack or because the player has already finished the game by playing all cards) the player should be able to observe an attack on another player. The player should still be able to see all cards played and every move made by other players (pass on, pick up etc.)

User Priority (4/5): For tactical reasons it is important that the player knows what's going on while the other players are making moves
Technical Priority (3/5): The game could be played without fully implementing this feature.

FREQ-11: After Turn End

If the battle is fully defended the defended cards should be burnt away. The system should automatically draw cards after every turn for every player if they have fewer than 6 cards, until the deck is depleted. The system should make sure that the correct order is held up when drawing cards. System should automatically determine whose turn it is next (attacking and defending).

User Priority (5/5): Drawing cards ensures that players maintain a playable hand and is a core part of the game's mechanics.
Technical Priority (5/5): Managing the deck and drawing cards is essential, the game cannot be played without.

FREQ-12: Game End

The system should end the game when only one player remains with cards (with the last player being the "Durak"). The system should display a message indicating who the "Durak" is at the end of the game. The system should offer an option to restart the game or exit to the lobby.

User Priority (5/5): Players need a clear determination of game outcomes.
Technical Priority (5/5): Properly identifying the game end state and handling final conditions is essential.

FREQ-13: User Messages

The system should display a message to the user, for example when the chosen game moves are illegal, when there is a problem in setting up the connection, when the user tries to start a game with the wrong number of players, and when a player is disconnected during an ongoing game.

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.

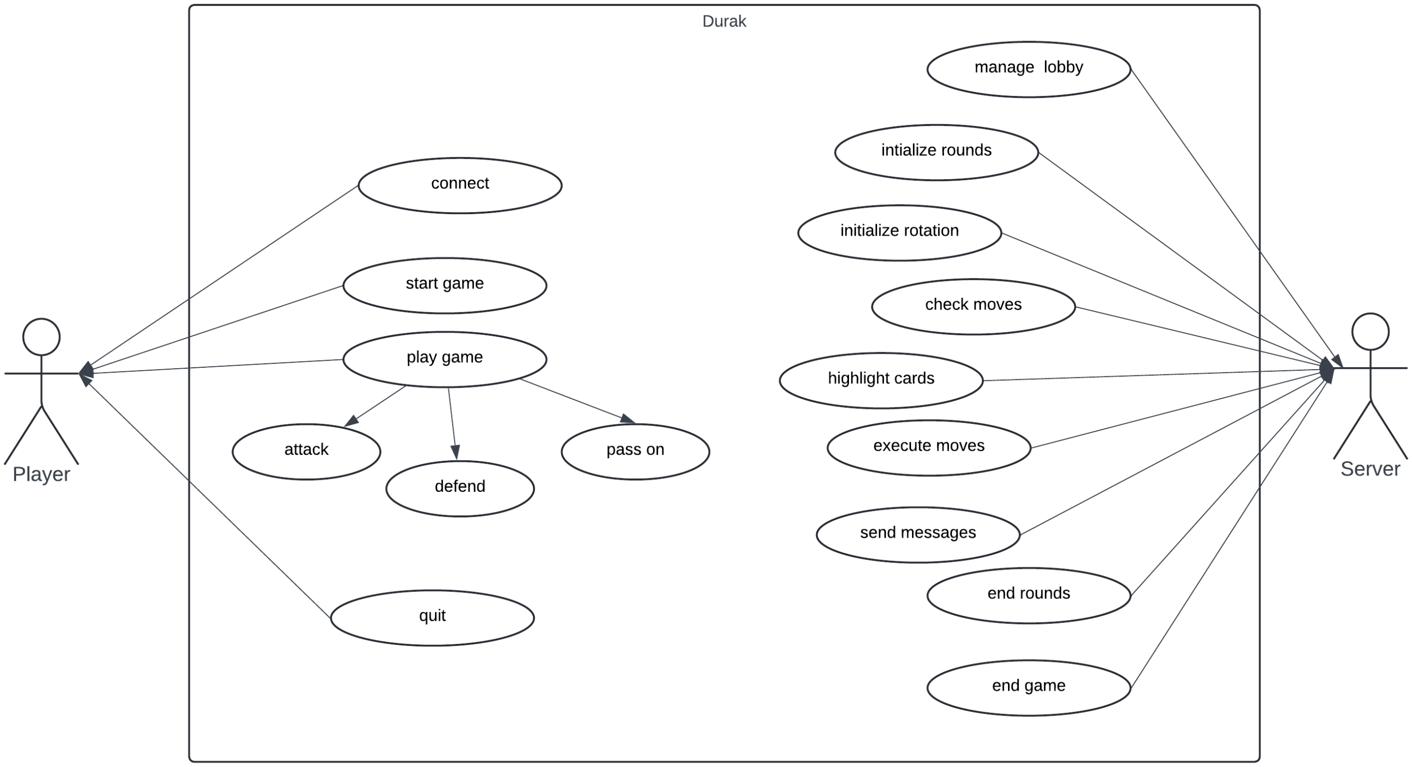
FREQ-14: Fairness:

The game absolutely must prevent cheating or unauthorized access.

User Priority (5/5): The user will not enjoy a game where other users are cheating.
Technical Priority (2/5): Although enforcement of the rules does not directly influence if the game runs at all, it surely must be considered when designing how the game is implemented.

System Scenarios

Use-case Diagrams



Scenarios

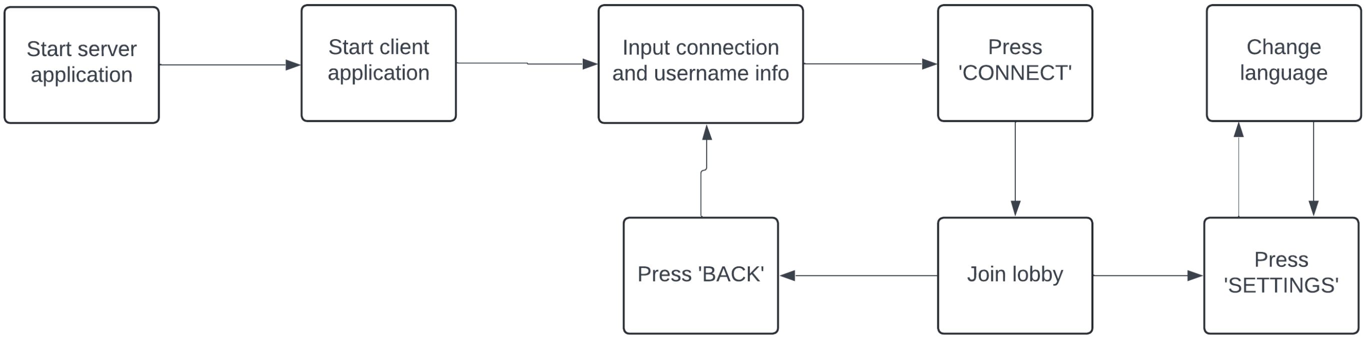
SCN-1: Setting up a game

FREQ reference	1, 2, 3, 4, 13
NFREQ reference	1, 2, 4, 5
Short Description:	Gregorij, Klara, Margret, Frank01 and Natasha gathered for a game of Durak. Gregorij starts the game server. Now all the other players can start their client. After the client is started each player must fill out the required information and their usernames. Once a player clicks the 'CONNECT' button the player joins the lobby, where the player waits for everyone else to join. Margret doesn't speak English very well and wants to have the instructions in a different language, she presses the 'SETTINGS' button and picks German.

	Frank01 doesn't like his username and wants to go back to change it. He presses the 'BACK' button and changes it to Frank, then he connects again to the lobby.
Activation action:	Gregorij starts the game server; every player starts their own client; everyone presses 'CONNECT'; Margret presses 'SETTINGS'; Frank01 presses 'BACK'.
Precondition:	All players have access to a computer where the game Durak is installed and a stable internet connection.

Basic flow: Setting up a game		
Step	**User action**	**System response**
1	Gregorij executes server application.	Starts server, displays open port.
2	Players execute client applications.	Starts clients, opens and displays login screen.
3	Players fill in connection information and usernames.	
4	Players press ‘CONNECT’	Connects clients, adds players to the lobby and displays lobby.
5	Margret clicks ‘SETTINGS’	Settings window opens.
6	Margret selects language	The language of the text displayed changes to the selected language.
7	Frank01 clicks ‘BACK’	Opens and displays the login screen.
8	Frank01 changes his username to Frank and presses ‘CONNECT’ again	Connects his client to the other clients, adds Frank to the lobby and displays lobby.
Post-condition:	All five players joined the lobby. Everyone sees each other. The language has been set for every player individually.	

Scenario Diagram for SCN-1 Setting up a game



SCN-2: Starting a game

FREQ reference	1, 2, 3, 5, 6, 13, 14
NFREQ reference	1, 2, 3, 4
Short Description:	The lobby is filled with 5 players and the settings have been set. Gregorij, Klara, Margret and Frank press the ‘START GAME’ button.

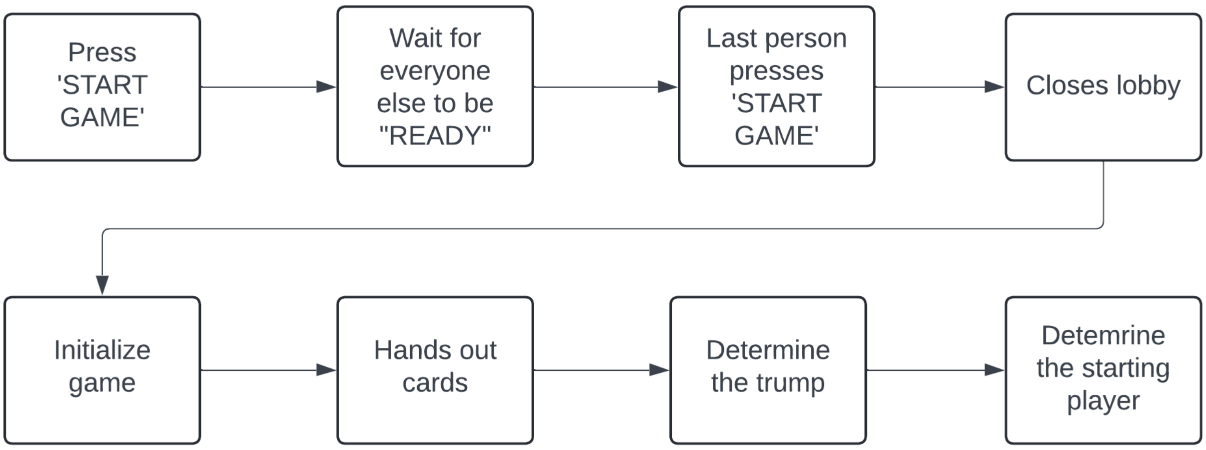
	Under their names a text appears saying “READY”. Finally, player Natasha presses the ‘START GAME’ button. The screen switches to the playing field and the lobby disappears. Each player receives 6 cards. The trump suit is determined, by drawing the top card from the pile and placing it on the according place. The player with the lowest trump card is determined from the system to start the first round.
Activation action:	Gregorij, Klara, Margret and Frank press the ‘START GAME’ button. Natasha clicks the ‘START GAME’ button
Precondition:	At least 3 players are in the lobby, max 6.

Basic flow: Starting a game

Step	**User action**	**System response**
1	Gregorij, Klara, Margret and Frank click the ‘START GAME’ button.	Under each name (Gregorij, Klara, Margret and Frank) the text “READY” appears.
2	Natasha clicks the ‘START GAME’ button.	Closes lobby, a new game is initialized. Each player gets 6 cards, and the trump suit is determined by the system. Then it determines who starts, based on who has the lowest trump card.

Post-condition:	All players have 6 cards, the trump suit is set, the player with the lowest trump card can start his/her turn.

Scenario Diagram for SCN-2 Starting a game


Press 'START GAME'
Wait for everyone else to be "READY"
Last person presses 'START GAME'
Closes lobby
Initialize game
Hands out cards
Determine the trump
Detemrine the starting player

SCN-3: Starting a new battle by ending a battle

FREQ reference	1, 2, 3, 6, 7, 8, 9, 10, 11, 13, 14
NFREQ reference	1, 2, 3, 5
Short Description:	A game with 5 players (Gregorij, Klara, Margret, Frank and Natasha) is ongoing, some turns have already been played. Gregorij was the last to defend himself, against Natasha and Klara. He successfully defended all attacks by playing higher cards. The cards that were in play are burned, and everyone who was involved and has less than 6 cards on hand, receives the number of cards missing (until 6 on hand again). It is now Gregorij's turn, he lays down a '4 of hearts', starting the attack on Klara, Klara defends herself by laying down '8 of hearts'. Gregorij attacks with another '4 of clubs'. Klara doesn't have clubs, but she defends with a '3 of spades' (the spades are trump cards). Margret (in this case the second attacker) has a '3 of clubs', which she puts down, co-attacking Klara. Klara chooses to pick up all cards that were part of the battle, as she doesn't want to waste more trump cards. Now the system hands out cards again until everyone has at least 6 cards on hand. Seeing that Klara picked up all the cards, she doesn't need any more. The system now determines who's turn is next, Klara is being skipped, and it is now Margret's turn to attack, because Klara lost her battle. The new battle starts.
Activation action:	Gregorij clicks on the '4 of hearts' card, Klara clicks on the '8 of hearts' card. Gregorij clicks on the '4 of clubs' card. Klara clicks on the '3 of spades' card. Margret presses the '3 of clubs' card. Klara presses the 'PICK UP' button. Gregorij and Margret press the 'DONE' (i.e. no more attack) button.
Precondition:	Gregorij, Klara, Margret, Frank and Natasha are in the middle of a Durak game. Gregorij successfully defended himself. Everyone that needed cards received them. Turn was passed to Gregorij. The card count for the pile and players was updated.
Basic flow: Starting a new battle by ending a battle	

Step	**User action**	**System response**
1	Gregorij clicks on the '4 of hearts' card and then on the playing field.	The '4 of hearts' is put on the playing field on one of the open slots, the display shows that Gregorij and Margret are attacking, and Klara is defending. Gregorij's card count is updated.
2	Klara clicks on the '8 of hears' card and then on the '4 of hearts' card which is laying on the playing field	The '8 of heart' card is put on top of the '4 of hearts'. Klara's card count is updated.

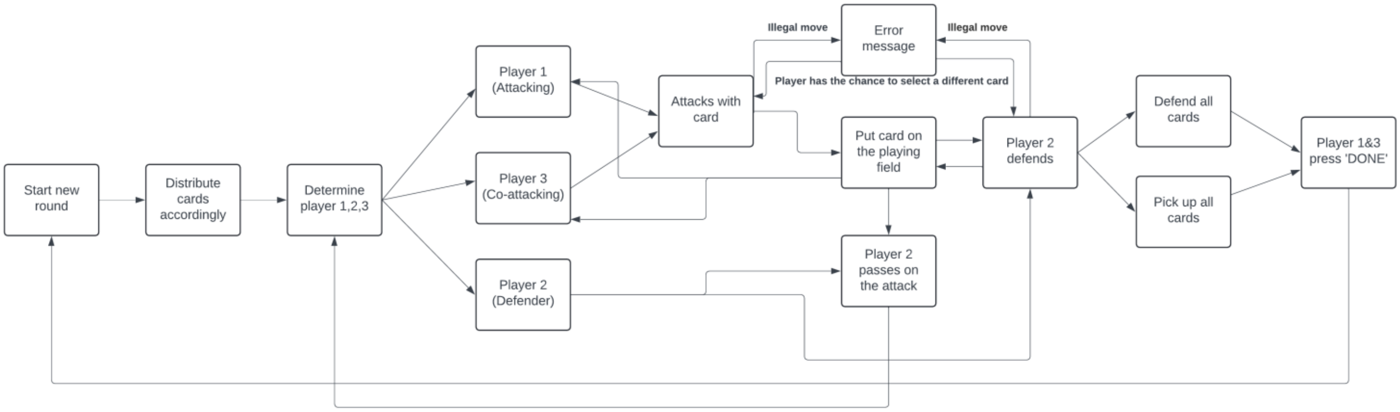
3	Gregorij clicks on the '4 of clubs' card and then on the playing field.	The '4 of hearts' is put on the playing field next to the other attack (cards) on one of the open slots. Gregorij's card count is updated.
4	Klara clicks on the '3 of spades' card and then on the '4 of clubs' card.	The '3 of spades' is put on top of the '4 of clubs' card. Klara's card count is updated.
5	Margret clicks on the '3 of clubs' card and then on the playing field.	The '3 of clubs' is put on the playing field, next to the other attacks (cards) on one of the open slots. Margret's card count is updated.
6	Klara presses the button 'PICK UP"	All the cards on the playing field are now put into Klara's hand, her card count gets updated.
7	Gregorij and Margret press the button 'DONE'.	The turn is done, Gregorij is missing 2 cards and Margret is missing 1 card to have 6 on hand, the system hands out the missing cards. The card count on the pile is updated. Klara's turn gets skipped and it is Margret's turn now.
Post-condition:	All players have at least 6 cards on hand. The system has determined who's turn is next. All of the card counts are updated accordingly.	

SCN-4: Playing the wrong card

FREQ reference	1, 2, 3, 6, 7, 8, 9, 10, 13, 14	
NFREQ reference	1, 2, 3, 5	
Short Description:	A game with 5 players (Gregorij, Klara, Margret, Frank and Natasha) is ongoing. Frank is new to the game. He must defend himself against a '7 of spades', with trump being heart. Frank clicks on his '10 of diamonds' and then on the playing field. An error message appears telling Frank that he tried to use an invalid card in this scenario.	
Activation action:	The '7 of spades' is the attacking card.	
Precondition:	The defending player still has cards on hand, the attacking cards have not exceeded 6 cards. The game is ongoing.	
Basic flow: Playing a wrong card		
Step	**User action**	**System response**
1	Frank clicks on the '10 of diamonds' and then clicks on the playing field.	The system checks if the suit matches and if it is a trump card, it doesn't match or if it's not big enough the system will show an error message.
2	Frank clicks a card of spades, namely '9 of spades'.	The system checks if the card selected is valid. ('9 of spades' > '7 of spades' => valid)

3	-	Removes the card from Frank's hand and places it on the other card. Updates Frank's card count.
Post-condition:	Frank sees that there was an error message when he tried playing the wrong card, and when he played the correct card, it was removed from his hand and placed on the playing field. His card count is updated.	

Scenario Diagram for SCN-3 Starting a new battle by ending a battle & SCN-4 Playing the wrong card



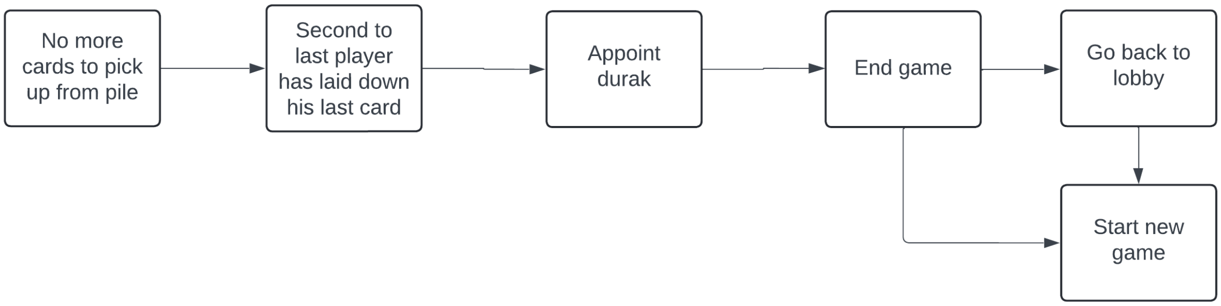
The card count was left out of this diagram due to it being too cluttered anyways. The card count update happens after every move or every time cards get distributed.

SCN-5: Endgame

FREQ reference	1, 2, 3, 7, 8, 9, 10, 11, 12, 13, 14
NFREQ reference	1, 2, 3
Short Description:	A game with 5 players (Gregorij, Klara, Margret, Frank and Natasha) is ongoing. The cards in the middle have been distributed among the players and there are no more cards left to pick up from the pile (card count of pile = 0). Slowly one player finishes after the next, only Gregorij (3), Klara (2) and Frank (6) are left still with cards. It's Frank's turn to attack. He can place at most 3 cards in his battle against Gregorij, because Gregorij only has 3 left in his hand. Frank plays the '10 of hearts' and the '10 of clubs'. Klara also has a '10' and attacks Gregorij with it as well. Gregorij beats all the cards with his trump cards and finishes. Now it is Klara's turn. She puts down her last card attacking Frank and also finishes. Frank is now the Durak (i.e. loser).
Activation action:	Klara places down her last card, while Frank still has cards.
Precondition:	Only one person has still cards on hand. Card count > 0.
Basic flow: Ending a game	

Step	User action	System response
1	Klara places her last card.	Updates Klara's card count to 0. Checks if Frank still has cards. Declares him as the Durak.
Post-condition:	Frank is Durak and the game has ended.	

Scenario Diagram for SCN-5 Endgame


No more cards to pick up from pile
Second to last player has laid down his last card
Appoint durak
End game
Go back to lobby
Start new game

System Constraints

Important Nonfunctional Requirements

NFREQ-1: Performance:

The game should be responsive and provide a smooth user experience. Displaying a move after a player has executed it and updating the game state should not take more than 1 second. The server should process requests within less than 1 second.

User Priority (5/5): This is of vital importance for the user as it drastically improves enjoyability of the game.
Technical Priority (2/5): Not necessary for the game to run at all, although time delays could influence a players move if it restricts his possibilities.

NFREQ-2: Reliability:

The game should be stable and reliable, with minimal crashes or errors. The game should crash at most 1 out of 100 times due to system failure.

User Priority (5/5): A crashed game is not enjoyable, thus a reliable system is necessary in order to play.
Technical Priority (5/5): A reliable system is necessary to play the game.

NFREQ-3: Scalability:

The performance of the game should not be unreasonably influenced by the number of players.

User Priority (3/5): It is in the interest of the user that the number of players does not drastically change his user experience.
Technical Priority (3/5): Ensuring scalability could prove important and streamlines further development process.

NFREQ-4: Learnability:

The game should be easy for new players to learn and understand. This should be achieved by intuitive user interface design and clear explanations of the game rules.

User Priority (4/5): New players are more likely to continue playing a game if they can easily learn the rules and mechanics. A steep learning curve could be a deterrent.
Technical Priority (1/5): This is not crucial for the games core functionality.

NFREQ-5: Maintainability:

The game codebase should be structured and documented to facilitate future maintenance and updates. This includes using modular design principles, clear coding conventions, and comprehensive documentation.

User Priority (1/5): This is not a primary concern from a user’s perspective.

Technical Priority (4/5): Maintainability is crucial for eventual further development of the game. Maintaining the codebase in a proper and structured way makes debugging and expanding easier, thus saving a lot of time in the long run.

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.>