Software Requirements Specification

for

Dominion

Version 1.0

Prepared by Nicola Bruhin, Mathieu Martin,
Aaron Mazzetta, Philipp Obrist,
Nicholas Schaller, Caspar Schucan

doMINIONS

11.10.2024

Table of Contents

Revision History

Name	Date	Release Description	Version
Felix Friedrich	25/09/24	Template for Software Engineering Course in ETHZ.	0.0.3
Team	03/10/24	Initial draft SRS document for Dominion	0.1
Team	11/10/24	The finalised version of the SRS document for Dominion	1.0

1. Introduction

1.1 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.>

1.2 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.>

1.3 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 organised. 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.>

1.4 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.>

1.5 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.>

2. Overall Description

2.1 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.>

2.2 Product Functions

This project implements the multiplayer game Dominion, created by Donald X. Vaccarino and published by Rio Grande Games for 2-4 players.

This concretely includes functionality to:

- Start a game server
- Join a game as a player using a client
- Start and play games once enough players have joined

During a running game, the software will:

- Track and display the state of the game. This includes in particular:
    - The cards of the player
    - The cards available for purchase
- For the player currently on the turn:
    - Display the possible actions the player can take and allow them to make moves via the GUI
    - Track and display state related to the player's current turn. This includes:
        - Remaining actions
        - Available funds
        - Remaining buys
- Make sure no move violates the rules
- Notify the players when the game has ended and there is a winner

2.3 User Classes and Characteristics

User class: Normal player

- They know how to play Dominion
- No technical expertise except for the basic capability to run an executable file on the defined operating system and interact with the game via GUI
- Can access all product functionality

2.4 Operating Environment

The software must be compatible with, compile and run on a fresh install of **Ubuntu 22.04**. It should be developed using modern software development tools to ensure compatibility with the specified environment.

2.5 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).>

2.6 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.>

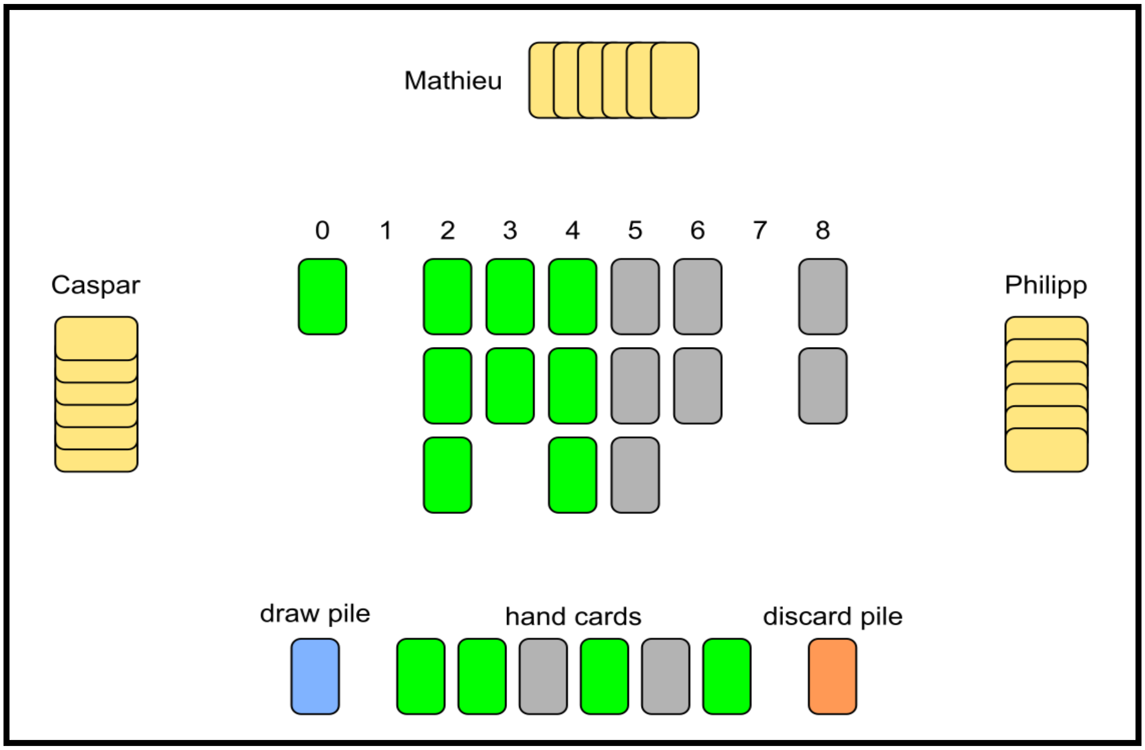
2.7 Assumptions and Dependencies

<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).>

3. External Interfaces and Requirements

3.1 User Interface

The following is a sketch of how the game user interface may be laid out. Currently, the player we are watching is playing. The playable or buyable cards are highlighted in green, and the others are greyed out. Note that many details have been omitted in this sketch to focus on the primary layout.



3.1.1 The login screen

- Visible when starting the game clientsetup, i.e. fields for TCP port and IP address.
- Provides a field for entering the username.
- Provides a button for creating a game. The player who created a game will be called the " **game master**" from here on.
- Provides a button for joining a game.

3.1.2 The card selection screen

- Allows the game master to choose a set of 10 kingdom cards by either
  - choosing from a preset
  - selecting 10 cards from the card library
  - generating a random configuration
- Lists the players who joined the game
- Provides a button for the game master to start the game. This button shall only be clickable once 2 or more players have joined the game.

3.1.3 The game screen

- Opens for all players in a lobby, after the game master has started the game
- The game screen will at least provide the following information
  - Other players
  - Cards available for purchase
  - The player's hand, including the draw and discard pile.
- Provides a way for the player to obtain an enlarged view of a card.

3.1.4 For the player currently doing his turn

- Allows the player to left-click on cards in his hand to play them.
- Allows the player to left-click on cards he is allowed to purchase to buy them.
- Provides buttons for actions other than buying and playing cards. Such as e.g. prematurely ending their turn.
- Allowed actions must be visually distinguishable from the forbidden actions, e.g. by being greyed out.
- Allows the player to use keyboard shortcuts to perform the following actions:
  - Buy **kingdom card**
  - Buy **treasure card: (**copper, silver, gold)
  - Buy **victory card:** (estate, duchy, province)

3.1.5 Errors

If the software encounters any unexpected errors, it shall inform the user about the problem using a popup message. The software shall try to recover from and restore its state to the last valid state before the error occurs. In case of an unrecoverable error, the software shall inform the player and may then abort the game or return to the login screen.

The software may show no popup message if the player attempts a forbidden action that is visually distinguishable from allowed actions.

3.2 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.>

3.3 Communications Interfaces

The system will implement client-server communication between one server and multiple clients. Server-client communication will ensure correct, reliable and in-order communication between both parties.

4. System Requirements

4.1 Functional Requirements

We have divided the functional requirements into three parts: server, client and GUI. This seems suitable, as those are the three major components of the game. A few rules-based requirements are listed at the top.

4.1.1.1 FREQ-01: Implemented Cards
Description: The game shall provide at least 10 playable kingdom cards, as this is the minimum required to play the game as intended. Many of the hundreds of cards that the whole game provides are not trivially implemented, because of complex interactions that are specific to single cards. Our target is to implement as many cards as possible, but we will require at least 10.
User Priority(5/5): *The game cannot be played without this requirement*
Technical Priority(5/5): *The game cannot be played without this requirement*

4.1.1.2 FREQ-02: Card selection
Description: If more than 10 kingdom cards are available, the user shall be able to select 10 of them either manually, from a few presets or at random
User Priority(3/5): *This allows many more facets to the game*
Technical Priority(1/5): Not required

4.1.2 Server Requirements

4.1.2.1 FREQ-S-01: Game Session Management
Description: The server shall manage the creation, joining, and closing of game sessions, allowing multiple clients to participate in the same game instance.
User Priority(5/5): *The game cannot be played without this requirement*
Technical Priority(5/5): *The game cannot be played without this requirement*

4.1.2.2 FREQ-S-02: Player Authentication
Description: The server shall authenticate players when they connect to ensure each player has a unique identity within the game.
User Priority(5/5): *The game cannot be played without this requirement*
Technical Priority(5/5): *The game cannot be played without this requirement*

4.1.2.3 FREQ-S-03: Game State Synchronisation
Description: The server shall maintain the game state and synchronise it across all clients after each player's move.
User Priority(5/5): *The game cannot be played without this requirement*
Technical Priority(5/5): *The game cannot be played without this requirement*

4.1.2.4 FREQ-S-04: Turn Management

Description: The server shall enforce turn order, ensuring each player plays only during their turn, and notify all clients of the current player's turn.
User Priority(3/5): The users could keep track of this themselves.
Technical Priority(1/5): Not required for core functionality.

4.1.2.5 FREQ-S-05: Rule Enforcement

Description: The server shall enforce the game rules according to the Dominion game logic, including card abilities, victory conditions, and any restrictions.
User Priority(4/5): Improves player experience by eliminating many forms of cheating.
Technical Priority(1/5): *The players can enforce the rules themselves*

4.1.2.6 FREQ-S-06: Disconnection Handling

Description: In the case that a player loses connection to the server, the server shall respond in a way such that the game can proceed. The server may remove said player from the game.
User Priority(2/5): *Improves the user experience.*
Technical Priority(1/5): Not required for normal gameplay

4.1.3 Client Requirements

4.1.3.1 FREQ-C-01: Player Interface

Description: The client shall provide an interface that allows players to interact with the game, including drawing cards, playing cards, and managing their deck.
User Priority(5/5): *The game cannot be played without this requirement*
Technical Priority(3/5): This can be done through raw data manipulation.

4.1.3.2 FREQ-C-02: Server Communication

Description: The client shall communicate player actions and game state updates in real-time with the server.
User Priority(5/5): *The game cannot be played without this requirement*
Technical Priority(5/5): *The game cannot be played without this requirement*

4.1.3.3 FREQ-C-03: Player Notifications

Description: The client shall notify players of important events, such as turn changes, card effects, and victory points tally, using visual (or potentially audio) cues.
User Priority(4/5): *Greatly improves playability.*
Technical Priority(2/5): *Players could manage this themselves.*

4.1.3.4 FREQ-C-04: API for GUI

Description: The client shall provide a game-state API, which the GUI can use to display relevant information. This way we can lower the technical priority of GUI requirements, as those will just beautify the information provided by the API.
User Priority(2/5): *This does not concern the players as long as a GUI is provided.*
Technical Priority(3/5): *The GUI can access raw data instead of a well-defined interface.*

4.1.3.5 FREQ-C-05: Client Reconnection

Description: The client shall attempt to reconnect to the server in case of connection loss and re-synchronize its game state.

User Priority(2/5): *Not essential, but may improve the multiplayer experience in case of connection losses.*
Technical Priority(2/5): *This can affect how the server manages communication between clients.*

4.1.3.6 FREQ-C-06: Invalid Action Feedback
Description: The client shall provide immediate feedback when a player attempts an invalid action, explaining why the action is not allowed.
User Priority(4/5): *Greatly improves player experience.*
Technical Priority(2/5): *Feedback on invalid actions is not required for core functionality.*

4.1.4 GUI Requirements

4.1.4.1 FREQ-G-01: Interactive Game Representation
Description: The GUI shall provide an interactive view of the game state. This includes the player's hand, the supply deck, and other relevant game elements such as action, buy and discard piles.
Unavailable actions, such as sold-out piles, or unaffordable cards shall be visually distinguishable from available actions. E.g. by being greyed out.
User Priority(5/5): *The game cannot be played without this requirement.*
Technical Priority(1/5): *This does not affect the core functionality of the game.*

4.1.4.2 FREQ-G-03: Real-Time Updates
Description: The GUI shall update in real-time based on the server's game state, reflecting changes in player decks, the supply pile, and other game elements.
User Priority(5/5): *The GUI cannot be used without this requirement.*
Technical Priority(1/5): *This does not affect the core functionality of the game.*

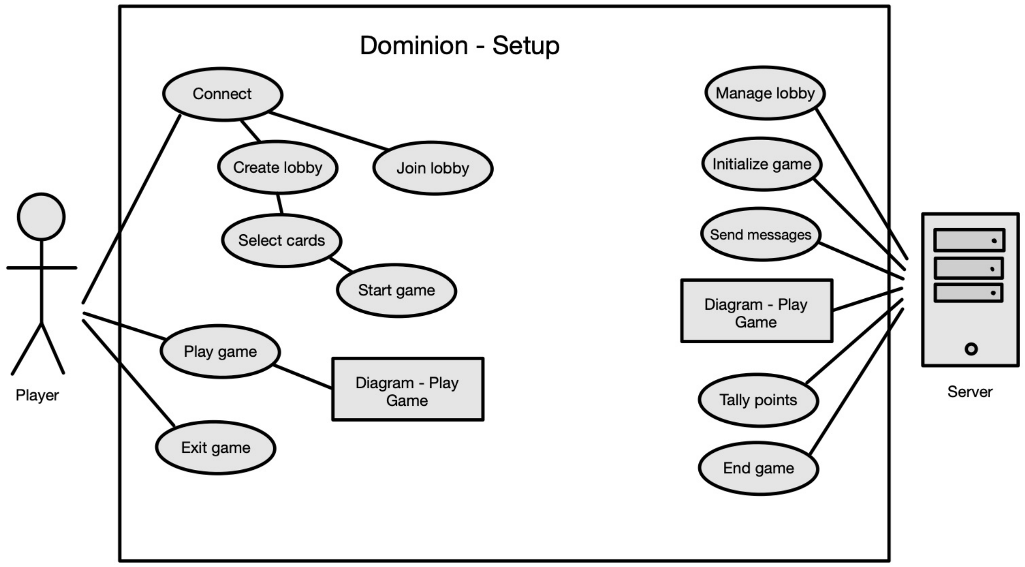
4.1.4.3 FREQ-G-04: Victory Screen
Description: The GUI shall display the victory points and notify all players when the game has ended, showing a final score breakdown.
User Priority(4/5): *The game ending is unsatisfactory without this requirement.*
Technical Priority(2/5): *The game can just stop after a winner is determined.*

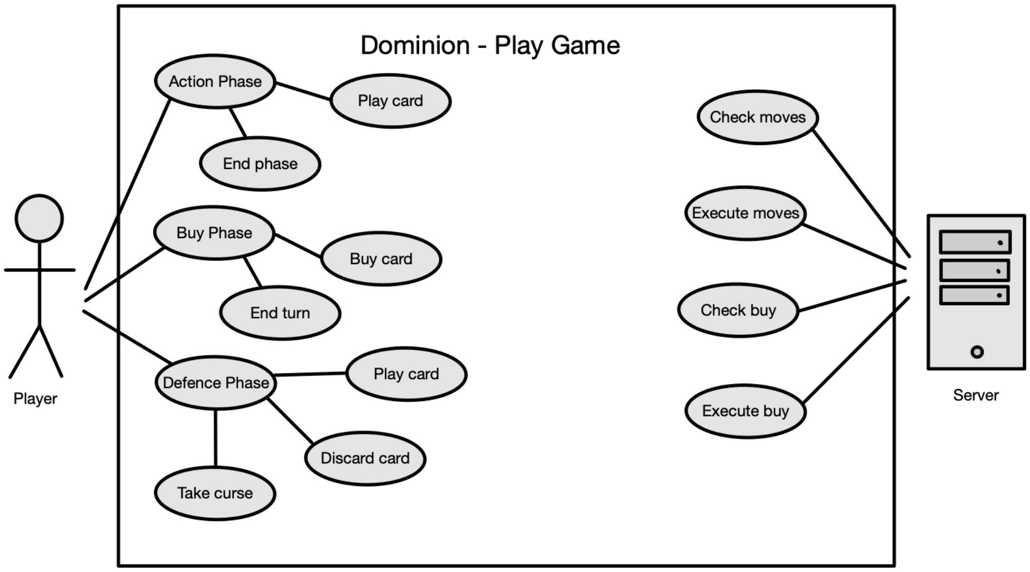
4.1.4.4 FREQ-G-05: Card Animations
Description: The GUI shall provide animations for card draws, plays, and other actions to enhance the user experience.
User Priority(2/5): *Not necessary for gameplay, but improves the experience.*
Technical Priority(1/5): *Adds complexity but is not critical to core functionality.*

4.1.4.5 FREQ-G-06: Minion mode
Description: The GUI shall provide a way for the player to change the GUI to become minion-themed.
User Priority(1/5): Funny joke, not essential
Technical Priority(1/5): Funny joke, not essential

5. System Scenarios

5.1 Use-case Diagrams





5.2 Scenarios

Any linked requirements are ones that immediately concern the scenario.

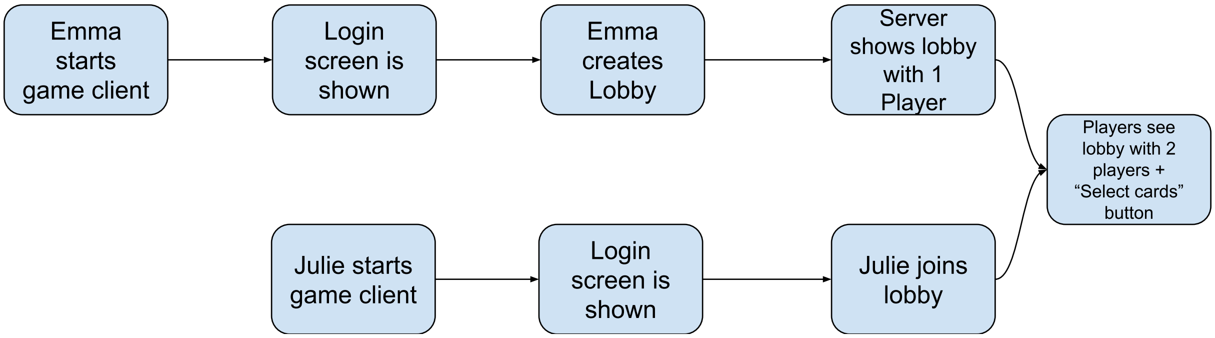
5.2.1 SCN-1: Setting up the lobby

FREQ reference	**FREQ-S-01, FREQ-S-02**

NFREQ reference	***NFREQ-01***
Short Description:	*Emma and Julie want to play a round of Dominion. Emma starts a client, enters her name and presses "Create Lobby". Julie starts a second client, enters her name and presses "Join Lobby"*
Activation action:	*Emma and Julie start game clients*
Precondition:	*The game server is already running*

Basic flow: Setting up the lobby		
Step	**User action**	**System response**
1	*Emma starts a game client*	*The game client starts and connects to the game server. Shows a login screen to Emma with a field for entering a name as well as a button to create a lobby.*
2	*Emma enters her name and presses "Create Lobby"*	*The system creates a game lobby with one player Emma.*
3	*Julie starts a game client*	*The game client starts and connects to the game server. Shows a login screen to Emma with a field for entering a name as well as a button to join the lobby.*
4	Julie enters her name and presses "Join Lobby".	*The game server adds Julie to the game lobby that Emma created.*
Post-condition:	*All players see the names of all players in the lobby as well as a button "Select cards"*	

Scenario Diagram for SCN-1 Setting up the lobby



5.2.2 SCN-2: Card selections

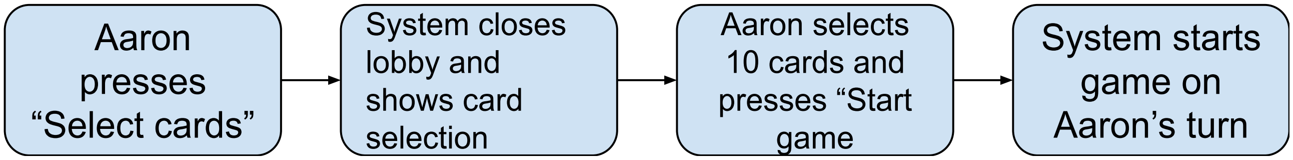
FREQ reference	***FREQ-C-01***
NFREQ reference	***NFREQ-02***
Short Description:	*Aaron, Nicola, Mathieu and Philipp are waiting in the game lobby. As Aaron presses "select cards" everyone is shown the selection of action cards. Aaron selects ten cards and then presses "start game".*
Activation action:	*Aaron presses "select cards"*

Precondition:	*Aaron, Nicola, Mathieu and Philipp are waiting in the game lobby.*

d

Basic flow: Card selection		
Step	**User action**	**System response**
1	*Aaron presses "select cards"*	*The system closes the Lobby and shows everyone the selection of action cards. Aaron can select 10 of them*
2	*Aaron selects 10 cards*	*Selected cards are marked and can be unselected by clicking again. When 10 cards are selected a "Start game" button appears*
3	*Aaron presses "start game"*	*The system distributes the starting cards to all players. Specifically, those are 7 Bronze and 3 estate cards. The cards are shuffled randomly, but the same for every player. The first 5 cards are the starting hand cards. The selected action cards as well as the treasure and victory cards are displayed in the center.*
Post-condition:	*The game starts with Aaron's turn and everyone has the same hand cards as well as a draw pile.*	

Scenario Diagram for SCN-2 Selecting cards


Aaron presses "Select cards"
System closes lobby and shows card selection
Aaron selects 10 cards and presses "Start game
System starts game on Aaron's turn

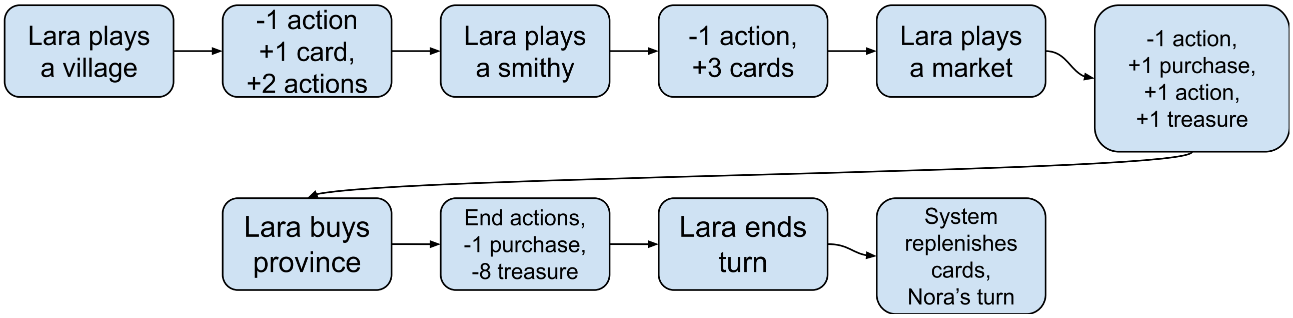
5.2.3 SCN-3: Standard turn

FREQ reference	***FREQ-G-01, FREQ-G-02, FREQ-G-05, FREQ-C-01, FREQ-S-03, FREQ-S-04, FREQ-S-05***
NFREQ reference	***NFREQ-04***
Short Description:	*Lara plays a village, then a smithy and a market. With the 2 buys and 9 gold Lara now has, she buys a province. Then, Lara ends her turn.*
Activation action:	*Freddy finishes his turn.*
Precondition:	*Freddy, Lara, Nora and Thomas are playing Dominion. Freddy has just ended his turn.*

Basic flow: Standard turn		
Step	**User action**	**System response**

1	*Lara plays a village.*	*The system gives Lara the uppermost card from the deck to the hand and leaves her with 2 remaining actions.*
2	*Lara plays a smithy*	*The system gives Lara the first three cards from the deck to the hand. The available actions are reduced by 1.*
3	*Lara plays a market*	*The system gives Lara another card from the deck as well as adds one to the number of purchases Lara can make. The available actions stay the same. Her funds have increased by 1.*
4	Lara buys a province	*The system enters Lara into the buying phase. All actions are disabled. Her funds are decreased by 8 and her purchases by 1. She has 1 purchase left.*
5	Lara ends her turn	*The system discards all of Lara's hand cards and takes the first 5 cards from the deck to her hand. Nora's turn begins.*
Post-condition:	*Freddy, Lara, Nora and Thomas are playing Dominion. Nora's turn has just begun.*	

Scenario Diagram for SCN-3 Standard turn

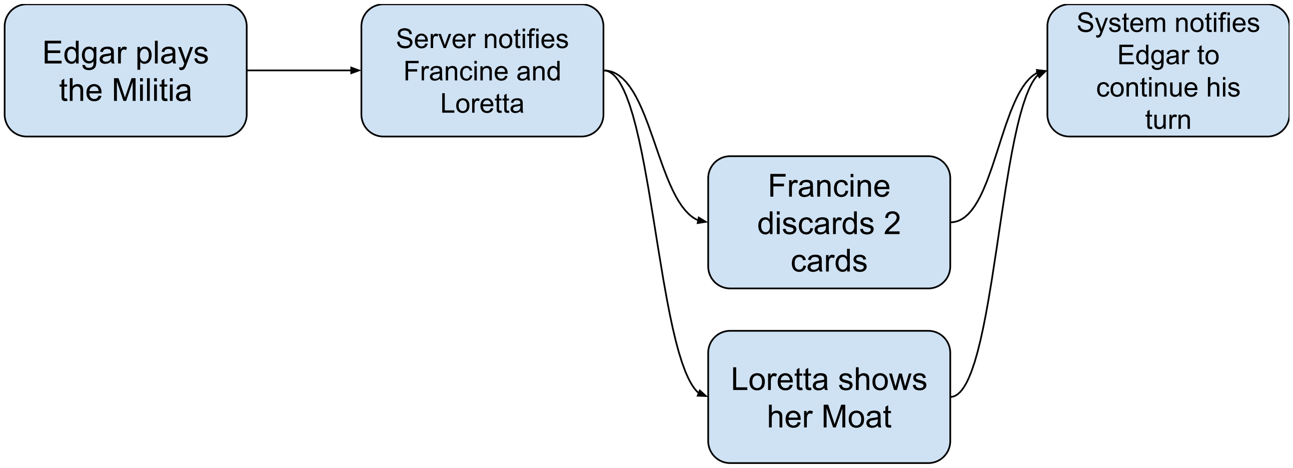

Lara plays a village
-1 action +1 card, +2 actions
Lara plays a smithy
-1 action, +3 cards
Lara plays a market
-1 action, +1 purchase, +1 action, +1 treasure
Lara buys province
End actions, -1 purchase, -8 treasure
Lara ends turn
System replenishes cards, Nora's turn

5.2.4 SCN-4: Attack

FREQ reference	***FREQ-G-01, FREQ-G-02, FREQ-G-05, FREQ-C-01, FREQ-S-03, FREQ-S-04, FREQ-S-05***
NFREQ reference	***NA***
Short Description:	*Edgar, Francine and Loretta are playing Dominion. Edgar plays the card "Militia". Francine now both have to discard all of their handcards except for 3. After this is satisfied Edgar's turn continues*
Activation action:	*Edgar plays the Militia*
Precondition:	*Edgar, Francine and Loretta are playing a game of dominion. It is Edgar's turn and he has at least one action left over.*

Basic flow: Attack		
Step	**User action**	**System response**
1	*Edgar plays the Militia*	*The system notifies the other players. They need to place all but 3 of their hand cards on the discard pile or use a reaction card*
2	*Francine has 5 cards and discards 2 of them*	*The system marks that Francine is done and waits for Loretta's response*
3	*Loretta uses her Moat*	*The system notifies the players that Francine has discarded cards and Loretta has a Moat to defend Edgar's turn and continues*
Post-condition:	*Edgar, Francine and Loretta are playing a game of dominion. It is Edgar's turn.*	

Scenario Diagram for SCN-4 Attack


Edgar plays the Militia
Server notifies Francine and Loretta
Francine discards 2 cards
Loretta shows her Moat
System notifies Edgar to continue his turn

5.2.5 SCN-5: End of game

FREQ reference	***FREQ-G-04, FREQ-C-02, FREQ-C-03, FREQ-S-03, , FREQ-S-04***
NFREQ reference	***NA***
Short Description:	*Claire was the first to play. Frank who's last to play has just ended his turn.*
Activation action:	*George buys the last province*
Precondition:	*Claire, Frank and George are playing Dominion. Claire is the first to play in a round and Frank last.*

Basic flow: End of game		
Step	**User action**	**System response**
1	*Frank ends his turn with no province's left*	*Since there are no provinces left, one of the end-of-game conditions has been met.*

		The system counts points for everyone and displays the victory screen.
Post-condition:	*The game is over*	

6. System Constraints

6.1 Important Nonfunctional Requirements

6.1.1 NFREQ-1: Server Speed

Description: *Any changes that have to be verified by the server and communicated to all clients should be completed within 1 second after the action took place.*
User Priority(5/5): *A fast server is crucial to the enjoyment of the game.*
Technical Priority(2/5): *The system is fully functional even without a fast server.*

6.1.2 NFREQ-2: Scalability

Description: *The system should be able to handle up to four players playing without a noticeable decrease in performance or an increase in delays. This game is best played with multiple people. Therefore additional players mustn't put a strain on performance.*
User Priority(5/5): *The game becomes substantially more fun if more than 2 people are playing together.*
Technical Priority(2/5): *The game itself is functional even if more players cause bad performance.*

6.1.3 NFREQ-3: Reliability

Description: *The game should not crash 99.9% of the time.*
User Priority(5/5): *If the game crashes often, this makes the game feel frustrating or unplayable.*
Technical Priority(3/5): *Whilst the game is technically functional even if it occasionally crashes, this should be avoided.*

6.1.4 NFREQ-4: Usability

Description: *The interface should allow players to interact with the game purely using their mouse except where text input is required. The interface shall provide a screen that lists the available keyboard shortcuts. The usage of the mouse should make the game feel more intuitive since it more closely resembles real-life play.*
User Priority(4/5): *Whilst some controls might be intuitive or found by trial and error if the players are unable to figure out the controls, they will not be able to play the game.*
Technical Priority(1/5): *From a technical standpoint, the game is fully functional without this requirement.*

6.1.5 NFREQ-5: Extensibility

Description: *The code should be well documented, easily modifiable and expandable, allowing developers to modify and add new features, especially new cards.*
User Priority(1/5): *This does not affect the players except for possible future additions to the game.*
Technical Priority(3/5): *This will allow for easier bug fixes and maintenance of the game.*

6.1.6 NFREQ-6: Anti-Cheating Measures

Description: *The software ensures that players cannot cheat by impersonating other players. If the software is implemented without this measure, a player could send a message to the server to prematurely end the turn for the other players forcing them to skip their turn.*
User Priority(1/5): *We assume that players of the game usually know each other personally and want to honestly win the game. Thus they have little interest in cheating in this manner.*
Technical Priority(1/5): *The game is fully functional without this requirement.*

6.1.7 NFREQ-7: Error Logging

Description: *The game should record exceptional events. This drastically accelerates the process of finding the source of bugs and thus facilitates debugging.*
User Priority(1/5): *This does not affect the player experience.*
Technical Priority(2/5): *The game is fully functional without this requirement, but it can be helpful in the debugging process.*

7. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalisation 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 organisation, 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.>