
Software Requirements Specification

for

Super Rummy

Version 1.0 approved

Prepared by Kion Tupper, Gavin Palazzo, Gunnar Eastman, Matt Virgin

Rummy Team

17 February 2022

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
4. System Features	4
4.1 System Feature 1	4
4.2 System Feature 2 (and so on)	4
5. Other Nonfunctional Requirements	4
5.1 Performance Requirements	4
5.2 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	5
5.5 Business Rules	5
6. Other Requirements	5
Appendix A: Glossary	5
Appendix B: Analysis Models	5
Appendix C: To Be Determined List	6

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document specifies the software requirements for the web application Super Rummy. This SRS describes the entirety of the system.

1.2 Document Conventions

Currently, no special fonts or highlighting exist. Requirements are listed in no particular order.

1.3 Intended Audience and Reading Suggestions

This document is intended for the professor and TA of the course COS 420/520. The rest of this document contains information on our application's UI, functional requirements, and non-functional requirements. This SRS is organized according to the template posted on Brightspace, and sections 1, 3, 4, and 5 are currently the most important to read.

1.4 Product Scope

The Super Rummy software application is meant to allow users to play the card game Rummy with friends or AI opponents. The product will eliminate the need to carry a deck of cards around to enjoy the game. The app will teach users how to play Rummy and allow them to practice, encouraging the development of social card game skills.

1.5 References

This SRS does not currently refer to any other documents or Web addresses.

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

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

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).>

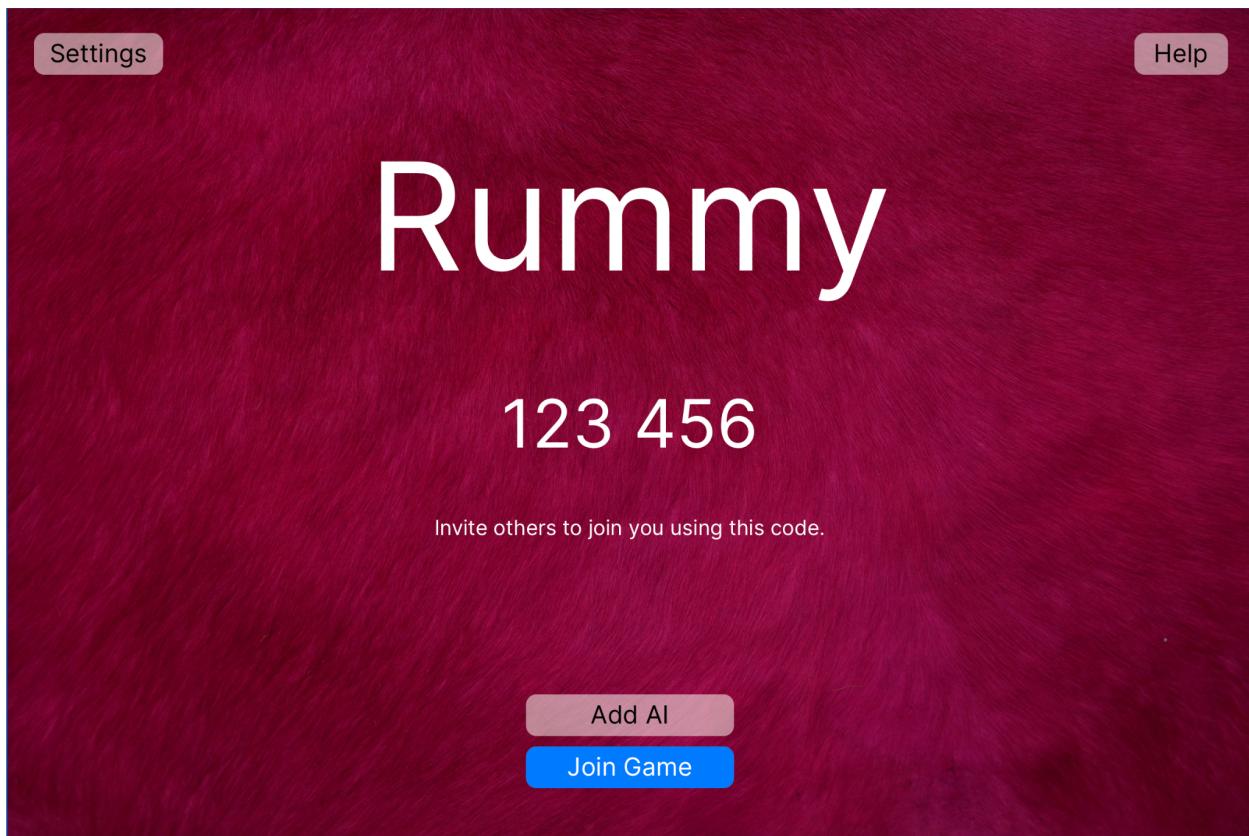
3. External Interface Requirements

3.1 User Interfaces

Two concepts have been developed for Super Rummy, referred to here as the Concept A and Concept B. The final interface will likely contain elements of both concepts, drawing from the strengths of each. A potential solution is described here, although this is highly subject to change.

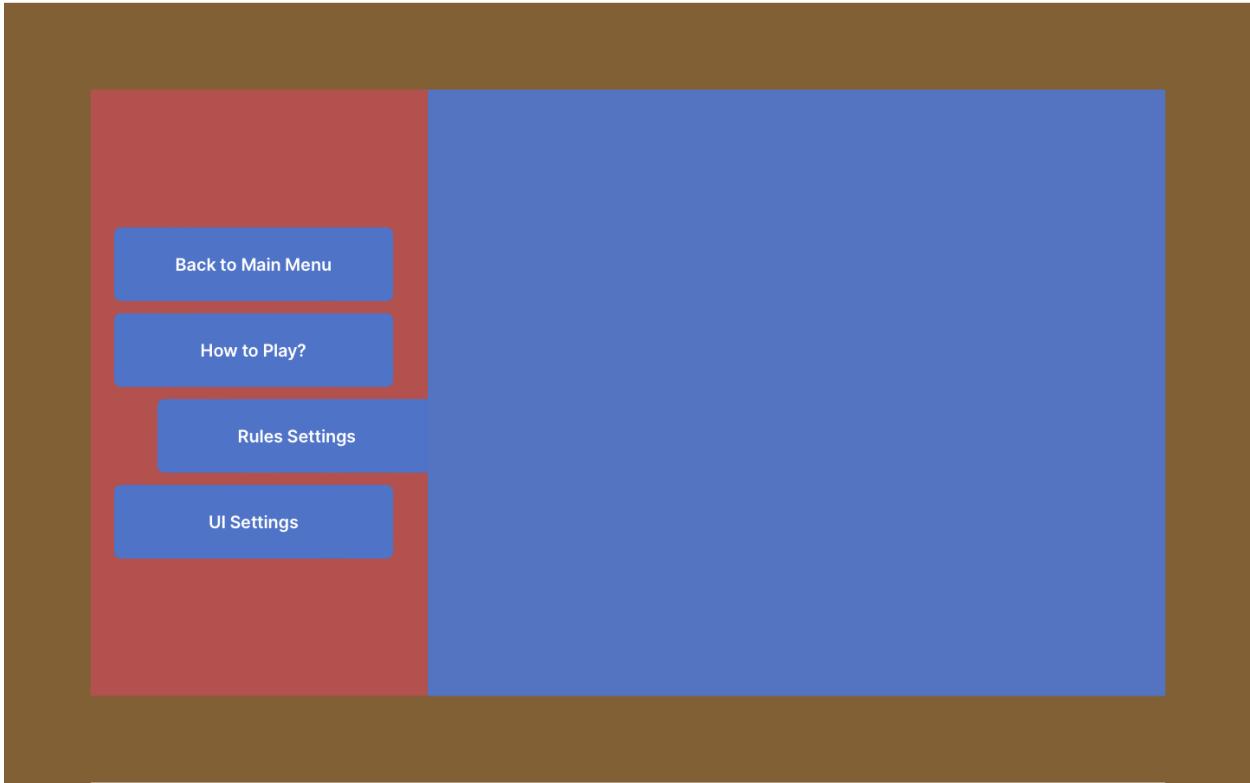
3.1.1 Overview of Screens

The first screen presented to the user depicts a new game automatically created for the user (Concept B):

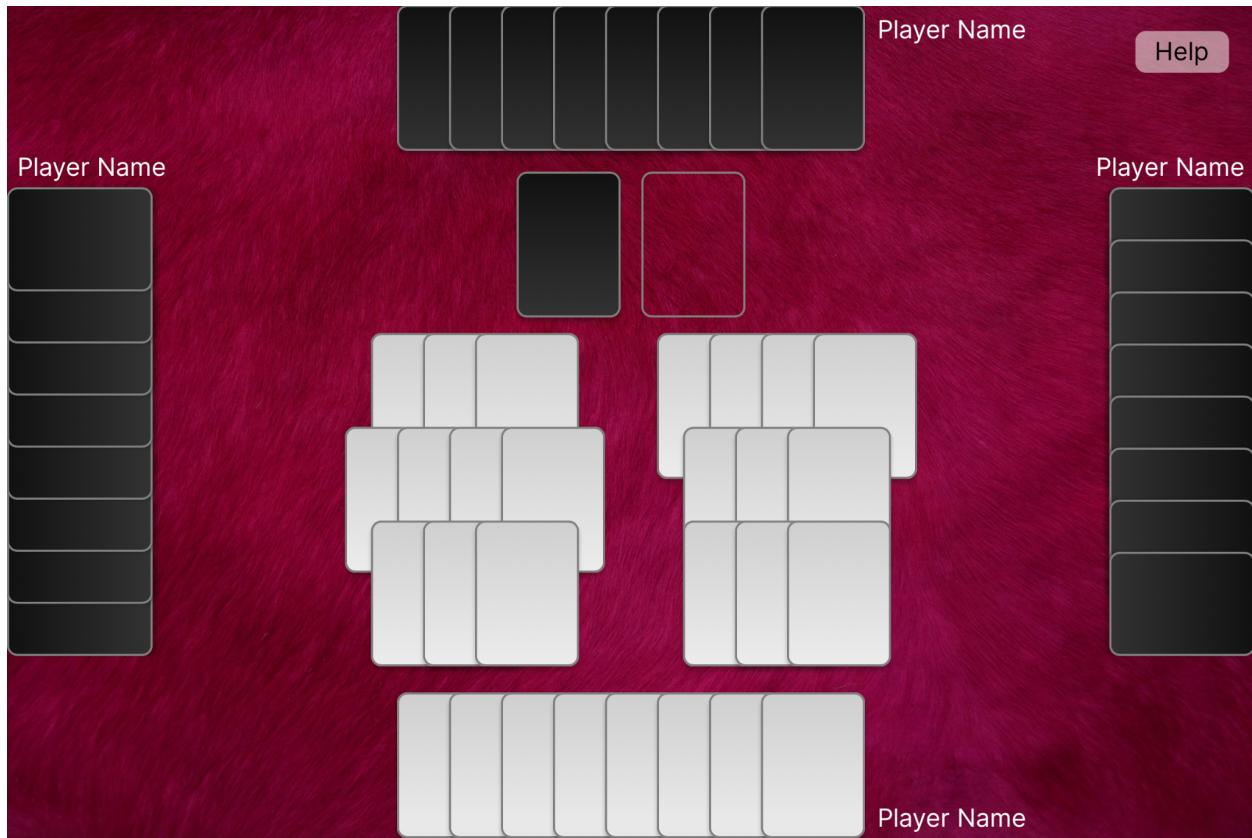


Functions shown are: a “Settings” button to present the settings screen (described below); an “Add AI” button which adds a computer player to the current game; a code (displayed here as “123 456”) which other users can use to join this game, and a “Join Game” button which allows the current user to join a different game.

The user will be able to adjust rules for the game (such as hand size, deck size, and options for legal melds), as well as interface settings including their theme (as in, the style of cards), in-game nickname, and other options (shown here from Concept A):



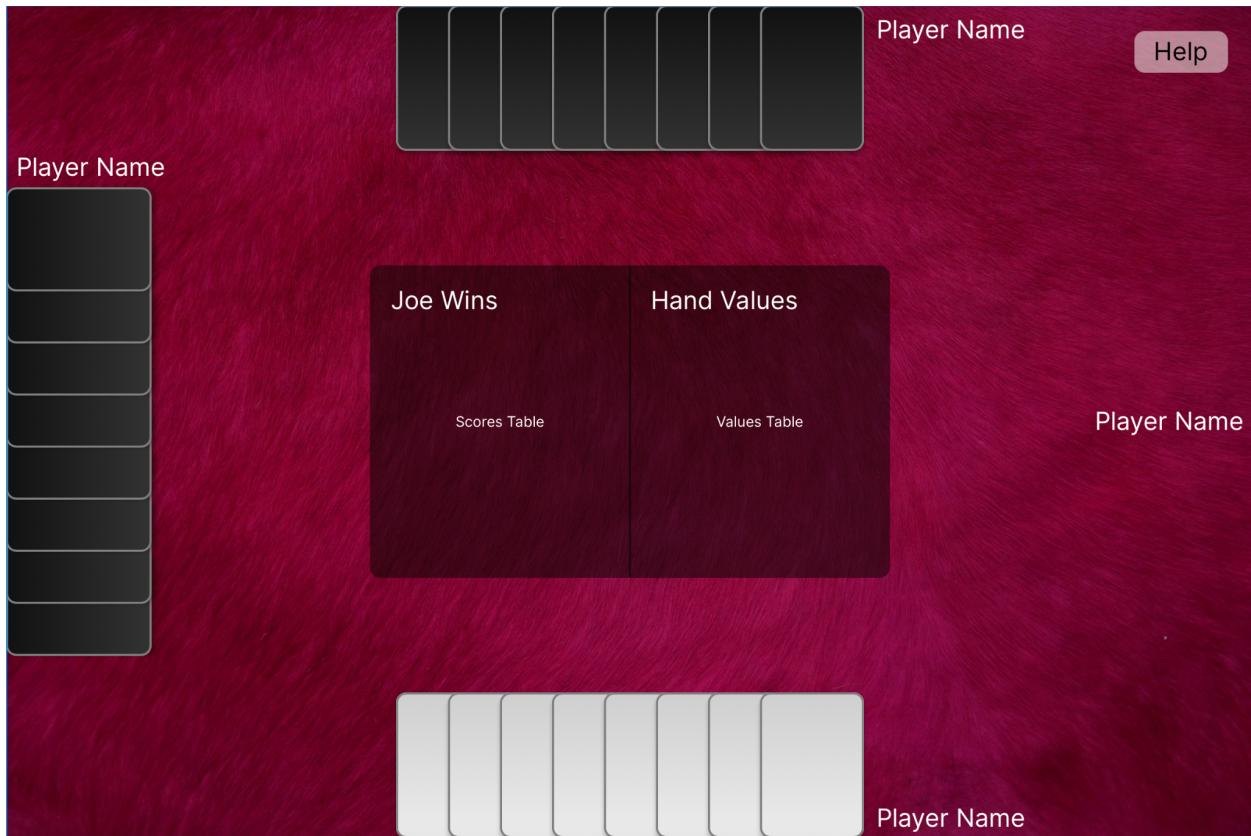
While in game, an interface such as shown below is presented (from Concept B):



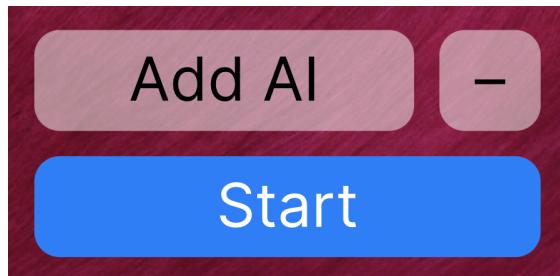
White cards are placeholders for face-up cards, while black cards are face-down.

The current hand is shown at the bottom, with the hands of other players shown on the sides and the top. The deck is shown near the top in the middle, as well as the location of the discard pile. Completed melds are shown face-up in the middle (6 are shown here).

At the completion of a round (when a player lays down all of their cards), a screen is shown depicting scores up to and including the round (Concept B):



3.1.2 Components



Buttons such as these are used throughout the interface for various actions. They are distinguished by the rounded border with a solid background.

A **Help** button is shown on every screen, which presents an interface explaining the rules and how to play the game.



Arguably the most important feature of a card game is, well, the cards. The current design uses placeholders for face-up and face-down cards. Art for these cards is a work in progress. Notably, users will be able to choose between several themed designs in the final product.

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 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.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic

forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 Functional Requirements

1. The system shall allow users to choose the color of their cards from at least 10 color options.
2. The system shall allow users to change the rules of the game, selecting from at least 2 different rulesets.
3. The system shall prevent users from making illegal plays
4. The system shall allow users to practice against AI opponents.
5. The system shall allow the user to change the number of AI opponents they face
6. The system shall allow users to play against up to 5 other users
7. The system shall allow users to start the game over at any point.
8. The system shall allow users to save the state of their game, should they wish to leave and continue playing later.
9. The system shall allow users to continue playing a previously saved game.
10. The system shall allow users to start a new game, should they not wish to continue a previously saved game or not have a previously saved game.
11. The system shall allow users to draw a card from the deck at the beginning of their turn.
12. The system shall allow users to draw the top card of the discard pile at the beginning of their turn.
13. The system shall allow users to discard a card from their hand at the end of their turn.
14. The system shall allow users to set their names at the beginning of the game.
15. The system shall autofill the names for each player so that the user does not need to type in their name at the beginning of each game.
16. The system shall allow users to select cards so that the selected card(s) can be played or discarded.

5. Other Nonfunctional Requirements

1. The system shall allow users to understand the game of rummy in less than 30 minutes.
2. The system's UI shall allow users to read their hand in less than 5 seconds.
3. The system shall work on at least 2 of the most popular browsers.
4. The system shall allow users to determine who is currently winning in less than 3 seconds.
 - a. The system shall provide a method to allow users to know who is winning the match within less than 5 seconds.
 - b. The system shall not provide a method to determine who is winning the round.
5. The system shall properly load the saved state of a user without error 99% of the time.
6. The system shall load a previously saved game state in less than 30 seconds 95% of the time.
7. The system shall load menus within 10 seconds, 90% of the time.
8. The system shall take no more than 30 seconds to undergo an AI opponent's turn 90% of the time.
9. The system shall not share user data with any third parties.
10. The system shall not cause user devices to reach temperatures above 60 degrees celsius.

6. 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.>