SFWRENG 3XA3: Software Requirements Specification
Rummy For Dummies

Lab 2 Group 7, Rummy For Dummies
Joy Xiao, xiaoz18
Benson Hall, hallb8
Smita Singh, sings59

February 12, 2021

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

This document describes the requirements for .... The template for the Software Requirements Specification (SRS) is a subset of the Volere template (**?**). If you make further modifications to the template, you should explicity state what modifications were made.

1 Project Drivers

1.1 The Purpose of the Project

With the ongoing pandemic, people are seeking new ways to entertain themselves from the comfort and safety of their home. The intent is to recreate a single-player card game that is normally played with several people. This is to give users the experience of playing a multiplayer card game within their own homes.

Rummy is a popular card game. Although the currently-available Rummy apps on app stores are free, they have many in-game advertisements. Many users may find this an annoyance. The software being developed is to be played on the local machine, to avoid advertisements and the need for internet access.

The software will be an improved version of Rummy. The game will require little memory to play. The project will also be more user-friendly compared to the open-source implementation, as it will be extended to be accessible to the general public, even if they do not have a technical background.

1.2 The Stakeholders

1.2.1 The Client

The client for this project is Dr. Bokhari and the TAs of the course SFWRENG 3XA3.

1.2.2 The Customers

The customers for this project would be people looking to play a game of Rummy.A group of stakeholders would be users that are familiar with Rummy. Allowing the game to be played through the desktop will give more people access to enjoying the game. In particular, card game enthusiasts

would have a great deal of interest in this software because they actively seek out various card games to play.

1.2.3 Other Stakeholders

Another stakeholder are the developers. They are responsible for re implementing the open source project, testing and maintaining it.

1.3 Mandated Constraints

1.3.1 Solution Constrains

Description: The project shall be written in Java and unit tested using JUnit as the testing framework.

Rationale: This is chosen because the developers are already familiar with using both technologies.

Fit Criterion: The source code will be written completely in Java.

Description: The game shall be played through the CLI.

Rationale: It ensures that the game is simple for users to play with a minimalist interface which uses less of the computers resources.

Fit Criterion: The project will have no GUI, and will use ASCII characters to represent cards to create a more visual experience for the users on the CLI.

Description: The program will used on operating systems such as Windows, Mac Os and Linux

Rationale: These operating systems will be able to run the project without any necessary installations.

Fit Criterion: The project will be fit to compile and execute on the above stated operating systems.

1.3.2 Implementation Environment of the Current System

1.3.3 Partner or Collaborative Applications

This implementation does not have any partner or collaborative applications. It relies only on the operating system to be able to run a java application.

1.3.4 Off-the-Shelf Software

There aren't any off-the-shelf software, that the developers will use within the implementation.

1.3.5 Anticipated Workplace Environment

The anticipated workplace environment is the user's home. However, the product can run on most desktop or laptop computers anywhere. There is no need for an internet connection, and the game can be played locally on the user's machine.

1.3.6 Schedule Constraints

The project must be completed by the week of April 5th, 2021. This is when the final demonstration will take place.

1.3.7 Budget Constraints

The budget is $0 because there is no monetary funding. Creating this project should have no monetary cost, because there will be no costly resources used.

1.3.8 Enterprise Constraints

The game is free to download and free to play for any user to play on any CLI.

1.4 Naming Conventions and Terminology

Table 2: Naming Conventions and Terminology

Term	**Definition**
CLI	Command Line Interface
JRE	Java Runtime Environment
GUI	Graphical User Interface
Stock Pile	Remaining cards in the deck after dealing

1.5 Relevant Facts and Assumptions

The project will implement a specific version of Gin-Rummy. (Insert the instructions to the version here)

2 Functional Requirements

2.1 The Scope of the Work and the Product

Currently, to launch the original source project and play a game of Gin Rummy requires the code to be compiled and built before operation. As the code is built in Go, it requires an installation of Go to compile and execute.

As a command line and JRE generally come with a majority of desktop computers, the goal is to re-create Gin Rummy in Java for general use.

2.1.1 The Context of the Work

This project is to be developed for the SFWRENG 3XA3 course. It is developed for the professor of the course, Dr. Asghar Bokhari, and the teaching assistant assigned to our team, Mehdi Jafarizadeh.

The plan is to re-implement Gin Rummy in Java and to make it more accessible to the general public.

The project will be developed such that it can be easily compiled and run in the command line.

2.1.2 Work Partitioning

Table 3: Work Partitioning Events

Event Number	Event Name	Input	Output
1	420	420	420
2	69	69	69
3	snoop dogg	420.69	69420

Table 4: Work Partitioning Summaries

Event Number	Summary
1	0

2.1.3 Individual Product Use Cases

2.2 Functional Requirements

Business Event 1: The user wants to start a new game.

1. The system shall ask the user to input their name.
2. The system shall display a random set of cards as the users card hand, and it shall display the top card on the discard pile.
3. The system shall display the set of options the user can take.

Business Event 2: The user wants to draw a card from the stock pile.

1. The system shall add the top card of the stock pile to the user's hand.
2. The system shall display the new card to the user through the CLI.
3. The system shall prompt the user to discard one of the cards in their hand.

Business Event 3: The user wants to pick up a card from the discard pile.

1. The system shall add the top card on the discard pile to the user's hand.
2. The system shall display the new card to the user through the CLI.
3. The system shall prompt the user to discard one of the cards in their hand.

Business Event 4: The user wants to check for melds.

1. The system shall show the user any melds that can be created from the users hands.

Business Event 5: The user wants to play a meld.

1. The system shall allow the user to select the cards they wish to play for the meld.
2. The system shall place the cards in a meld if the user's entry is valid.

3. The system shall remove the cards used for the meld from the user's hand.

Business Event 6: The user wants to check total points of the hand.

1. The system shall calculate and display the user's points.

Business Event 7: The user wants to discard a card.

1. The system shall prompt the user to input which card the user wants to discard.
2. The system shall discard the chosen card if valid and display the new card hand.
3. The system shall place the card on top of the discard pile faced up.
4. The system shall make a move on behalf of the computer agent

Business Event 8: The user wants to knock.

1. The system checks if the user has less than 10 deadwood points.
2. If the user has 10 or fewer deadwood points, then the system shall end the round.
3. The system will calculate the difference in deadwood points between the user and AI and determine the winner
4. The system will award the points to the winner
5. The system will prompt the player if they want to play a new round.

3 Non-Functional Requirements

3.1 Look and Feel Requirements

LF.1 The system shall have a simplistic look.

3.2 Usability and Humanity Requirements

UH.1 The system shall be easy to play for users of any age.

UH.2 The system shall be easy to run for users of any age.

3.3 Performance Requirements

P.1 The system shall respond to valid user interactions within 0.5 seconds.

P.2 The system shall adhere to rules of the game Rummy.

3.4 Operational and Environmental Requirements

OE.1 The system shall be able to be operated on Windows, Linux, and MacOS operating systems.

3.5 Maintainability and Support Requirements

MS.1 The system shall be clearly documented and commented for ease of maintainability.

MS.2 The system shall be as modularized as possible.

3.6 Security Requirements

N/A

3.7 Cultural Requirements

C.1 The system shall not have any offensive images or text.

C.2 The system shall be available in English.

3.8 Legal Requirements

L.1 This system shall adhere to the same Open Source License as the original project: MIT License.

L.2 This system shall not violate any of the MIT License copyright properties.

3.9 Health and Safety Requirements

N/A

4 Project Issues

4.1 Open Issues

There is one open issue currently. The issue is to fix the test cases for checking deadwood cards.

4.2 Off-the-Shelf Solutions

4.3 New Problems

4.4 Tasks

The Gantt Chart for the project will be followed.

Table 5: Task due dates

Task	**Due Date**
Proof of Concept Demonstration	February 22, 2021
Test Plan Revision 0	March 5, 2021
Design and Document Revision 0	March 18, 2021
Revision 0 Demonstration	March 22, 2021
Final Demonstration	April 5, 2021
Final Documentation	April 12, 2021

4.5 Migration to the New Product

4.6 Risks

N/A

4.7 Costs

There should be no costs for this project.

4.8 User Documentation and Training

4.9 Waiting Room

4.10 Ideas for Solutions

References

5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.