Gin Rummy, With a Twist

Team 17

Qixuan Zhong
Zizeng Li
Qiwen Jiao
Chenrui Hao
Yunze(Figo) Li

October 11, 2024

Content

1 Versions, Roles and Contributions

1.1 Revision History

Version Number	Description	Date
Rev 0	The initial draft of SRS	October 9th, 2024

1.2 Personnel of the project

Team Member	Email	Role
Qixuan Zhong	zhongq7@mcmaster.ca	Developer
Zizeng Li	li124@mcmaster.ca	Backend Developer
Qiwen Jiao	jiaoq2@mcmaster.ca	Frontend Developer
Chenrui Hao	haoc3@mcmaster.ca	Full Stack Developer
Yunze(Figo) Li	li561@mcmaster.ca	Product Manager & Developer

1.3 Table of Contributions

Team Member	Sections Worked On
Qixuan Zhong	1 Versions, Roles and Contributions 2 Introduction 6 Data and Metrics 7 Functional Requirements
Zizeng Li	3 Stakeholders 4 Mandated Constraints 5 Relevant Facts and Assumption
Qiwen Jiao	11 Operational and Environmental Requirements 13 Security Requirements
Chenrui Hao	8 Look and Feel Requirements 9 Usability and Humanity Requirements 10 Operational and Environmental Requirements
Yunze(Figo) Li	12 Maintainability and Support Requirements 14 Legal Requirements 15 Risks and issues predicted.

2 Introduction

2.1 Purpose of the project

Gin Rummy has been one of the most popular two-person card games for over a century. This project recreates the game electronically but with a difference. The entire game is played and scored in the dozen number base —“base twelve”— with a dozenal deck of 64 (5 dozen 4) cards. Dozenal has been researched for a few centuries as a number base superior to decimal (“base ten”) mainly because of its abundance of non-trivial factors, 2, 3, 4 and 6, compared to those of decimal, only 2 and 5, with five not being particularly useful.

2.2 Goal of the project

The project involves dealing with the cards with animation and sorting them with mild animation, enabling the play of the cards, and scoring and presenting the results. The focus will be on implementing the new game logic, perfect user interactions and seamless user experience.

2.3 Glossary

Term	Definition
Dozenal number	A number expressed in base 12.
GUI	Graphic User Interface
Tooltip	A small pop-up box that appears when a user hovers over or focuses on an element in a user interface
FPS	Frames Per Seconds, the number of images displayed each second in an animation, determining how smooth the motion appears
Codebase	The entire collection of source code for a software project, including all code files and dependencies

3 Stakeholders

3.1 Client

- A card game enthusiast organization or platform is a key stakeholder in this updated version of Gin Rummy, as they require an accessible and user-friendly platform to provide a diverse selection of card games to their community.

3.2 Customer

- People who interested in the dozenal number base, because that is the main difference from the usual game
- A card game enthusiast serves as a primary stakeholder for the online Gin Rummy game, as it offers them a unique and strategic gameplay experience. This version of the game allows them to explore diverse strategies, appealing to their interest in card-based tactics and innovative game mechanics.
- A player of a traditional Gin Rummy game can also be a stakeholder of this Gin Rummy game since they may think about approaches to improve their thinking patterns and come up with new strategies of winning this new version of game.

- Individuals who enjoy card games but may not always have a physical deck on hand are valuable stakeholders for the online Gin Rummy game. This group benefits from the convenience and accessibility of the digital version, allowing them to play anytime, anywhere without the need for physical cards.

3.3 Other Stakeholders

- Paul Rapoport served as the project supervisor, guiding us in modifying the rules of Gin Rummy and enhancing the user experience for this digital version of the game.
- Our capstone team members are the primary developers of the website, contributing extensively to its design, functionality, and overall development to bring this Gin Rummy game to life.

3.4 Personas

- Individuals who enjoy card games casually and are looking for new and engaging types of card games to explore.
- Dedicated Gin Rummy players who seek an online platform where they can compete with other enthusiasts, hone their skills, and evaluate their gameplay level.
- People looking for a fun, accessible game to enjoy online, offering entertainment and a way to unwind through an interactive digital experience.

3.5 User Participation

- Before releasing the official version of the Gin Rummy game, introduce users to the original visual effects and a beginner-friendly tutorial. Collect user feedback to enhance the game's visual appeal and make the rules more understandable and accessible for new players.

3.6 Maintenance Users and Service Technicians

- Developers are responsible for updating game visuals and rules, fixing bugs, maintaining server stability, addressing user-reported issues such as third-party cheating and lag, ensuring database integrity, and managing overall performance.

4 Mandated Constraints

4.1 Solution Constraints

- Accessible across major web browsers and compatible with desktops, mobiles, and tablets.
- Implement secure user authentication and data protection with a safe connection proxy.
- Support multiple concurrent users with minimal latency, ensuring a real-time multiplayer experience.

4.2 Implementation Environment of the Current System

- The project utilizes Spring Boot as the backend framework and Next.js for the frontend, hosted on an AWS cloud server with a MySQL database for data management.

5 Relevant Facts And Assumptions

5.1 Relevant Facts

- Supports experienced players and beginners with interactive tutorials.
- First-time users must create an account to access the platform.
- Players can join games via system-pairing or by entering a user ID to host a match.

5.2 Business Rules

- Apply a ranking system to boost user engagement and assess user levels.
- Implement encryption for user actions to prevent cheating and ensure fair play.
- Ensure that user data and gameplay information are securely stored and protected.

5.3 Assumptions

- Players can access the game on devices with compatible browsers.
- The platform is designed for gameplay, with no betting features or in-game currency.
- A stable connection is recommended for optimal performance.

6 Data and Metrics

6.1 Data Storage

- All data should be stored in the server, and not editable by any user(Except for the login credentials).
- The following data should be stored and publicly available:
    - The global leaderboard
- The following data should be encrypted and stored as sensitive entries for each user:
    - Login credentials for verifying users' identity
    - Tracking in-game data turn times, win/loss ratio, and gameplay patterns for each user.
    - Storing the gameplay history for the user to review the history of their gameplay.
- The following data should be encrypted and only available for admins' analytical purposes:
    - Number of registered users
    - Number of games played, in both decimal/dozenal modes

7 Functional Requirements (Require Data for each Feature)

P0 (Minimum Viable Product)	Backend: ● Implement basic game logic in dozenal scheme Frontend ● Basic interaction with the game logic. Could start from command-line based for proof of concept
P1 (Features that significantly improved experience)	Backend: ● Implement a local network server that connects two clients for a match ● Implement basic game logic in the regular (decimal) scheme and allow the user to select between the schemes. ● Implement login feature and user management system ● Implement a lobby feature for inviting two users to the same match Frontend

	● Implement a GUI for the game, preferably a web-based game
P2 (Non-critical features to the functions of the application)	Backend: ● Pushing the server to the internet, allowing two remote clients to match ● Implement a matchmaking feature to pair players of similar skill levels. Frontend ● Highlight the changed cards in dozenal scheme ● Implement animation for dealing cards, sorting cards and other necessary actions. ● Ensure the game is playable across web browsers on various devices (desktop, mobile, tablets).
P3 (Nice to have, not required)	Backend: ● Implement a leaderboard and ranking system ● Implement game rules in other number bases(octal for example) Frontend ● Implement In-game chat and communication system. ● Implement a verifying process for users to recover their account ● Implement skin and customization for user preferences

8 Look and Feel Requirements

8.1 Appearance Requirements

- Using the card deck from Professor Paul
- The game should be responsive and accessible across various devices, including desktop, tablets and smartphones. UI elements should adjust seamlessly to different screen sizes without losing functionality to clarity
- Animations for card movements, shuffling, and sorting should be smooth. There should have visual feedback, such as highlighting the selected cards and notifying players of game events

8.2 Style Requirements

- Use clean and readable fonts. Font sizes should be adaptive to screen size, ensuring that text remain legible across different devices
- The overall style of the game should be modern and minimalist. The goal is providing a focused and immersive gameplay experience
- Use specific shades to indicate error messages and success messages. Ensures that alerts and notifications remain consistent with the overall design
- Loading animations, such as spinners or progress bar, and it should match the website’s theme

9 Usability and Humanity Requirements

9.1 Ease of Use Requirements

- User authentication should be streamlined, allowing new users to register and existing users to login quickly. Social login options could be available to minimize registration time
- The state of the game should be visible at a glance. The interface should update in real-time, keeping players informed without refreshing the page.

9.2 Personalization and Internationalization Requirements

- Users could be able to customize their profiles with avastars, display names and game preferences. Customizations should be saved and accessible across all devices upon login
- Users should have the option to adjust gameplay settings, such as background music volume, sound effects and card deck designs. It should also be easily accessible and persistent across sessions.

9.3 Learning Requirements

- New users should have access to an optional guided tutorial that introduces the basics of the game, including rules, gameplay mechanics, and scoring in base twelve. This tutorial should be interactive and brief, helping users get started quickly
- Tooltips should be available for key game elements, such as card functions, game rules, and scoring calculation. Hovering over an element should display a brief description to aid users in understanding the game
- Users should have easy access to detailed game rules and base twelve explanations from within the game interface. It should also include examples of common game scenarios to help users understand the rules.

9.4 Understandability and Politeness Requirements

- Error messages should be clear and polite, explaining the issue in simple terms and offering guidance on how to resolve it (e.g. “Unable to connect. Please check your internet connection and try again.”)
- When a user wishes to leave a game or log out, there should be a polite prompt, such as “Are you sure you want to leave the game?”. This ensures that users have the opportunity to save their progress or confirm their decisions.

9.5 Accessibility Requirements

- Users with slower reaction times should have the option to extend time limits for making a move during gameplay.
- All images, icons should include descriptive tooltip alt text to support screen readers.

10 Performance and Speed Requirements

10.1 Speed and Latency Requirements

- The game should respond to user actions within 300 milliseconds to provide a smooth and responsive user experience
- The initial game page including login and main menu should load within 5 seconds on a standard internet connection.
- If a user disconnects during a match and then reconnects, the game state should be synced and reloaded within 5 seconds to ensure they can resume play.
- Card animations and transitions should run at a minimum of 60 FPS to provide smooth visuals.

10.2 Safety-Critical Requirements

- All backend API endpoints should require proper authentication and authorization checks to prevent unauthorized access. Request rate limit should be applied to API requests.
- Data encryption and decryption processes (e.g. for user data) should be optimized to avoid introducing significant latency in real-time interactions.

10.3 Precision or Accuracy Requirements

- The scoring system must be accurately calculate points using base twelve rules to ensure fairness and transparency in determining the winner of each game
- User actions such as playing a card or entering data, should be validated accurately to prevent invalid actions that could disrupt the game flow.

10.4 Robustness or Fault-Tolerance Requirements

- When database latency or failure, the game should use cached data to serve critical information (e.g. game rules) without noticeable delay.
- There should be a health check run at frequent intervals to check the backend server is running correctly.

10.5 Capacity Requirements

- The system should support at least 50 concurrent players across various games without performance degradation.
- The database should be capable of storing user accounts information, game histories data and game states, with an initial capacity to store at least 50 users data.

10.6 Scalability or Extensibility Requirements

- The game codebase should follow software design patterns, allowing easily adding new features without significant changes to the existing structure
- The backend API should be designed with extensibility in mind, allowing third-party services (e.g., social media sharing, sign in with google etc) to be integrated with minimal effort

10.7 Longevity Requirements

- The game should be designed to allow easy maintenance and updates, ensuring compatibility with new web standards and technologies.

11 Operational and Environmental Requirements

11.1 Expected Physical Environment

- The game server should operate in a stable network environment to ensure a smooth gaming experience.

11.2 Wider Environment Requirement

- The game should be operated in different devices from different users, it should be work fluently in different operating systems (like Windows, macOS)

11.3 Requirement for Interfacing with Adjacent Systems

- The game should integrate with a user account system, allowing players to register, log in, and add friends. It should also enable players to join game rooms using unique room numbers and match with friends.

11.4 Productization Requirements

- The game should be fully tested and discussed with Professor Paul, to ensure that there are no critical bugs

11.5 Release Requirements

- Before release, all features must be developed and tested to ensure the game meets expected performance standards, and approved to release by Professor Paul

12 Maintainability and Support Requirements

12.1 Maintenance Requirements

- The codebase should include a suite of automation tests to ensure changes are not breaking major system functionality.
- Adopt clear versioning policy to manage updates of the game
- Maintain an updated list of libraries, frameworks and dependencies version.

12.2 Supportability Requirements

- Developer team should maintain a comprehensive technical documentation, including API references, system architecture diagrams, and deployment procedures.

12.3 Adaptability Requirements

- The game should be adaptable to different devices and operating systems.
- The game should be deployable on any cloud platforms with minimal changes

13 Security Requirements

13.1 Access Requirements

- Implement a secure login system, such as username and password combinations, to ensure that some user functions can only be performed under login

13.2 Integrity Requirements

- Ensure that the game is not interrupted during play, and all game data is preserved in its entirety to prevent data loss or corruption

13.3 Privacy Requirements

- Ensure that each player's cards remain private during the game, players only have access to their own cards, and it will not be revealed to other players

14 Legal Requirements

- There are no legal or compliance requirements for this project.

15 Risks and issues predicted.

15.1 Risks

- **Development inexperience**: Lack of experience working with game developing and its framework. This inexperience may cause more time in the beginning to build up the structure. Furthermore, it may be a struggle to implement core features effectively, as we are unfamiliar with the tools and libraries available for game development.
- **Lack of Visual Design Skills**: Due to limited experience with visual design tools and animation software, the resulting visuals may not achieve the desired quality. This may lead to dull animations and graphics, which may not deliver a perfect visual experience.
- **Testing**: With all developers are beginners in the Gin Rummy game, there is a risk regarding the testing phase. The lack of experience may hinder the ability to anticipate edge cases, these tricky points will be hard to test out

15.2 Open Issues

- **Interface Design**: It is significant to develop an intuitive and user-friendly interface for our users. Generating feasible animations during the game. This process may involve conducting preliminary user testing to optimize the design and enhance its overall usability.
- **Rubric Design**: We will create a new game system based on dozenal, which will involve developing unique card types, a scoring system, and specific game mechanics tailored to this base. We will establish a clear and engaging scoring system that ensures the game remains balanced and competitive.
- **Scalability Challenges:** As the game gains popularity, the server may struggle to handle a larger number of concurrent users. It may lead to slow response time or even crash the server.