# Software Requirements Specification for ProgName: subtitle describing software

Team #25, The Crazy Four

Ruida Chen
Ammar Sharbat
Alvin Qian
Jiaming Li

30.09.2025

# Contents

# Revision History

Table 1: Revision History

| Date | Developer(s) | Change |
|---|---|---|
| 9.29 | Jiaming Li | Purpose of the Project |
| 9.30 | Jiaming Li | Scope of the Product |
| 9.30 | Jiaming Li | FR |

# 1 Purpose of the Project

## 1.1 User Business

The purpose of this project is to design and implement an educational card game based on the traditional *Crazy 8s* rule set, but adapted to integrate the **Dozenal (base-12) number system**.

- This project addresses the lack of accessible and engaging tools that introduce alternative number systems in a playful and intuitive way.
- By combining a familiar card game mechanic with Dozenal representations and operations, users can gradually build comfort and intuition with the base-12 system.
- The primary business value lies in providing a lightweight, fun, and interactive educational tool for students, hobbyists, and anyone interested in number systems beyond decimal.

Additionally, the game offers an opportunity to evaluate how gamification can support mathematical learning, and whether abstract concepts (such as base conversions or divisibility in Dozenal) can be effectively taught through play.

## 1.2 Goals of the Project

The goals of this project are:

- Educational Integration: Seamlessly incorporate Dozenal concepts (symbols 0–B, factorization, arithmetic) into the gameplay, ensuring that players learn by playing without requiring formal prior knowledge.
- Gameplay Design: Deliver a working digital version of *Crazy 8s* that is intuitive, responsive, and enjoyable, while maintaining the familiar flow of the original game and introducing Dozenal-specific mechanics (e.g., matching rules, scoring, or special cards).
- Accessibility and Engagement: Create a user-friendly interface that lowers the barrier to learning, accessible for casual users while offering depth for learners who want to explore Dozenal further.

- Scalability / Stretch Goals: Explore the potential for extending the system to other educational card or board games, and investigate how different number bases can be taught through similar game mechanics.

# 2 Stakeholders

## 2.1 Client

*Insert your content here.*

## 2.2 Customer

*Insert your content here.*

## 2.3 Other Stakeholders

*Insert your content here.*

## 2.4 Hands-On Users of the Project

*Insert your content here.*

## 2.5 Personas

*Insert your content here.*

## 2.6 Priorities Assigned to Users

*Insert your content here.*

## 2.7 User Participation

*Insert your content here.*

## 2.8 Maintenance Users and Service Technicians

*Insert your content here.*

# 3 Mandated Constraints

## 3.1 Solution Constraints

*Insert your content here.*

## 3.2 Implementation Environment of the Current System

*Insert your content here.*

## 3.3 Partner or Collaborative Applications

*Insert your content here.*

## 3.4 Off-the-Shelf Software

*Insert your content here.*

## 3.5 Anticipated Workplace Environment

*Insert your content here.*

## 3.6 Schedule Constraints

*Insert your content here.*

## 3.7 Budget Constraints

*Insert your content here.*

## 3.8 Enterprise Constraints

*Insert your content here.*

# 4 Naming Conventions and Terminology

## 4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

*Insert your content here.*

# 5 Relevant Facts And Assumptions

## 5.1 Relevant Facts

*Insert your content here.*

## 5.2 Business Rules

*Insert your content here.*

## 5.3 Assumptions

*Insert your content here.*

# 6 The Scope of the Work

## 6.1 The Current Situation

*Insert your content here.*

## 6.2 The Context of the Work

*Insert your content here.*

## 6.3 Work Partitioning

*Insert your content here.*

## 6.4 Specifying a Business Use Case (BUC)

*Insert your content here.*

# 7 Business Data Model and Data Dictionary

## 7.1 Business Data Model

*Insert your content here.*

## 7.2 Data Dictionary

*Insert your content here.*

# 8 The Scope of the Product

## 8.1 Product Boundary

The product to be developed is a digital card game application based on the traditional *Crazy 8s* rules, modified to integrate the **Dozenal (base-12) number system**. The system boundary includes:

- A game engine that supports core Crazy 8s mechanics (drawing, discarding, turn-taking, winning conditions).
- Adaptations of rules, card values, and scoring to incorporate Dozenal arithmetic and representations.
- A user interface allowing players to interact with the game (play cards, view scores, receive feedback).
- Educational prompts or visual aids to help players understand Dozenal concepts.

External systems not included in the boundary are: general learning platforms, multiplayer servers beyond basic peer-to-peer/local play, and integrations with unrelated educational tools.

## 8.2 Product Use Case Table

The following table summarizes the primary product use cases (PUCs):

## 8.3 Product Use Case Table

The following table summarizes the primary product use cases (PUCs):

| **PUC #** | **Description** |
|---|---|
| PUC-1 | Player starts a new Crazy 8s game with Dozenal-enabled deck. |
| PUC-2 | Player takes a turn by drawing or discarding a card. |
| PUC-3 | System validates whether the played card is legal (same suit, same Dozenal value, or sum = 12). |
| PUC-4 | Player views scores and progress, displayed in both decimal and Dozenal. |
| PUC-5 | System provides hints or explanations to support Dozenal learning. |
| PUC-6 | Game ends when a player wins; final scores are calculated and displayed. |

## 8.4 Individual Product Use Cases (PUCs)

### UC1: Start a New Game

1. Player opens the application and selects "New Game".
2. The system initializes a Dozenal-enabled deck (0–B, 10).
3. The system shuffles the deck.
4. The system deals cards to each player.
5. The game state is displayed on the interface.

### UC2: Take a Turn

1. The system indicates that it is the player's turn.
2. The player chooses either to play a card or to draw from the deck.
3. If the player chooses a card, the system checks its validity against the discard pile.
4. If valid, the card is placed onto the discard pile.
5. If invalid, the system notifies the player and the card remains in the hand.

6. If the player chooses to draw, the system gives one card from the deck to the player.

**UC3: Validate Move**

1. The player selects a card to play.
2. The system retrieves the top card from the discard pile.
3. The system checks if the move is legal under Dozenal rules:
   (a) same suit, or
   (b) same Dozenal value, or
   (c) sum of the two values equals 12 (base-12).
4. If the move is legal, the system accepts the card and updates the discard pile.
5. If not, the system rejects the move and notifies the player.

**UC4: View Scores**

1. A round ends or the player requests to view scores.
2. The system calculates points for each player.
3. The system converts the scores into both decimal and Dozenal.
4. The system displays the results on screen.

**UC5: Provide Hints**

1. The player hovers over or selects a card.
2. The system analyzes the current game state.
3. The system provides a hint (e.g., "This card is valid because its sum with the top card equals 12 (base-12).").
4. The hint is displayed as text or a visual highlight.

**UC6: End Game**

1. A player discards their last card.

2. The system checks if the game-ending condition is satisfied.
3. If satisfied, the system declares the winner.
4. The system calculates final scores in both decimal and Dozenal.
5. The results are displayed on the final game screen.

# 9 Functional Requirements

## 9.1 Game Manager

1. **Start new game:** Game manager shall allow the player to start a new Crazy 8s game with a Dozenal-enabled deck. (FR-1)
   **Rationale:** Players need a way to initialize the game state; starting a new game is the foundation for all other gameplay functions.

2. **Turn management:** Game manager shall manage player turns, ensuring that each player either discards a valid card or draws a card. (FR-2)
   **Rationale:** Turn-taking enforces fairness and ensures game flow consistency.

3. **Rule validation:** Game manager shall validate that each played card is legal under Dozenal rules. A valid move is defined as either: (a) same suit as the previous card, (b) same Dozenal value as the previous card, or (c) the sum of the two card values equals 12 in Dozenal. (FR-3)
   **Rationale:** Prevents illegal moves, guarantees consistency, and introduces the core educational mechanic of Dozenal arithmetic.

4. **Special cards:** Game manager shall implement special card effects (e.g., 8s are wild) while supporting extensions with Dozenal-specific effects. (FR-4)
   **Rationale:** Special cards increase engagement and add flexibility in teaching Dozenal-based rules.

5. **End of game:** Game manager shall determine when the game ends (e.g., when a player runs out of cards) and declare the winner. (FR-5)
   **Rationale:** A clear end condition is required for meaningful gameplay and reinforcement of learning objectives.

## 9.2 Score Manager

1. **Calculate score:** Score manager shall calculate points for each round in both decimal and Dozenal. (FR-6)
   **Rationale:** Displaying scores in both systems reinforces learning by encouraging comparison between familiar decimal and new Dozenal formats.

2. **Display score:** Score manager shall display both decimal and Dozenal results on the user interface. (FR-7)
   **Rationale:** Visual feedback supports player understanding and helps users internalize Dozenal representations.

## 9.3 Education Support

1. **Hints:** System shall provide hints or explanations when a player performs an action involving Dozenal arithmetic. (FR-8)
   **Rationale:** On-demand guidance lowers the learning curve and supports players with varying levels of familiarity.

2. **Highlight valid moves:** System shall visually highlight valid moves based on Dozenal rules. (FR-9)
   **Rationale:** Reduces frustration, ensures players stay engaged, and reinforces Dozenal rules through visual learning.

# 10 Look and Feel Requirements

## 10.1 Appearance Requirements

*Insert your content here.*

## 10.2 Style Requirements

*Insert your content here.*

# 11 Usability and Humanity Requirements

## 11.1 Ease of Use Requirements

*Insert your content here.*

## 11.2 Personalization and Internationalization Requirements

*Insert your content here.*

## 11.3 Learning Requirements

*Insert your content here.*

## 11.4 Understandability and Politeness Requirements

*Insert your content here.*

## 11.5 Accessibility Requirements

*Insert your content here.*

# 12 Performance Requirements

## 12.1 Speed and Latency Requirements

*Insert your content here.*

## 12.2 Safety-Critical Requirements

*Insert your content here.*

## 12.3 Precision or Accuracy Requirements

*Insert your content here.*

### 12.4 Robustness or Fault-Tolerance Requirements

*Insert your content here.*

### 12.5 Capacity Requirements

*Insert your content here.*

### 12.6 Scalability or Extensibility Requirements

*Insert your content here.*

### 12.7 Longevity Requirements

*Insert your content here.*

## 13 Operational and Environmental Requirements

### 13.1 Expected Physical Environment

*Insert your content here.*

### 13.2 Wider Environment Requirements

*Insert your content here.*

### 13.3 Requirements for Interfacing with Adjacent Systems

*Insert your content here.*

### 13.4 Productization Requirements

*Insert your content here.*

### 13.5 Release Requirements

*Insert your content here.*

## 14 Maintainability and Support Requirements

### 14.1 Maintenance Requirements

*Insert your content here.*

### 14.2 Supportability Requirements

*Insert your content here.*

### 14.3 Adaptability Requirements

*Insert your content here.*

## 15 Security Requirements

### 15.1 Access Requirements

*Insert your content here.*

### 15.2 Integrity Requirements

*Insert your content here.*

### 15.3 Privacy Requirements

*Insert your content here.*

### 15.4 Audit Requirements

*Insert your content here.*

### 15.5 Immunity Requirements

*Insert your content here.*

## 16 Cultural Requirements

### 16.1 Cultural Requirements

*Insert your content here.*

## 17 Compliance Requirements

### 17.1 Legal Requirements

*Insert your content here.*

### 17.2 Standards Compliance Requirements

*Insert your content here.*

## 18 Open Issues

*Insert your content here.*

## 19 Off-the-Shelf Solutions

### 19.1 Ready-Made Products

*Insert your content here.*

### 19.2 Reusable Components

*Insert your content here.*

### 19.3 Products That Can Be Copied

*Insert your content here.*

# 20 New Problems

## 20.1 Effects on the Current Environment

*Insert your content here.*

## 20.2 Effects on the Installed Systems

*Insert your content here.*

## 20.3 Potential User Problems

*Insert your content here.*

## 20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

*Insert your content here.*

## 20.5 Follow-Up Problems

*Insert your content here.*

# 21 Tasks

## 21.1 Project Planning

*Insert your content here.*

## 21.2 Planning of the Development Phases

*Insert your content here.*

# 22 Migration to the New Product

## 22.1 Requirements for Migration to the New Product

*Insert your content here.*

## 22.2 Data That Has to be Modified or Translated for the New System

*Insert your content here.*

# 23 Costs

*Insert your content here.*

# 24 User Documentation and Training

## 24.1 User Documentation Requirements

*Insert your content here.*

## 24.2 Training Requirements

*Insert your content here.*

# 25 Waiting Room

*Insert your content here.*

# 26 Ideas for Solution

*Insert your content here.*

# Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?

2. What pain points did you experience during this deliverable, and how did you resolve them?

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

5. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

6. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?