

**Capstone Project Report**

**Report 3 – Software Requirement Specification**

– Ho Chi Minh, October 2025 –

**Table of Contents**

[I. Project Report 3](#_Toc72138553)

[1. Status Report 3](#_Toc72138554)

[2. Team Involvements 3](#_Toc72138555)

[3. Issues/Suggestions 3](#_Toc72138556)

[II. Software Requirement Specification 4](#_Toc72138557)

[1. Overall Description 4](#_Toc72138558)

[1.1 Product Overview 4](#_Toc72138559)

[1.2 Business Rules 5](#_Toc72138560)

[2. User Requirements 6](#_Toc72138561)

[2.1 System Actors 6](#_Toc72138562)

[2.2 Use Cases 6](#_Toc72138563)

[3. Functional Requirements 8](#_Toc72138564)

[3.1 System Functional Overview 8](#_Toc72138565)

[3.2 <<Feature Name 1>> 10](#_Toc72138566)

[3.3 <<Feature Name 2>> 10](#_Toc72138567)

[4. Non-Functional Requirements 11](#_Toc72138568)

[4.1 External Interfaces 11](#_Toc72138569)

[4.2 Quality Attributes 12](#_Toc72138570)

[5. Other Requirements 14](#_Toc72138571)

[5.1 Appendix1 - Messages List 14](#_Toc72138572)

[5.2 Appendix2 - … 14](#_Toc72138573)

[5.3 … 14](#_Toc72138574)

# I. Record of Changes

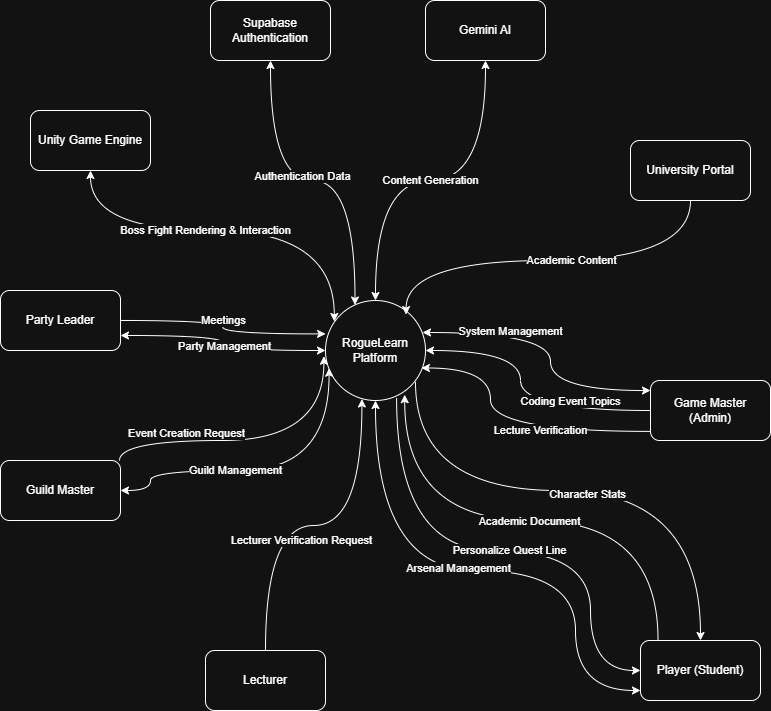
|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
| 10/10/2025 | A\* | Minh Anh | Initial document creation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. Software Requirement Specification

## 1. Product Overview

RogueLearn is a gamified learning platform that aligns university curriculum with career roadmaps. It converts academic progress into quests, skill trees, and boss fights, supporting students with AI-driven personalization and integrations (e.g., FPTU portal, roadmap.sh).



## 2. User Requirements

### 2.1 Actors

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Player (Student) | Primary user who onboards, views skill tree, completes quests, attempts boss fights, and manages personal notes; may become Party Leader or Guild Master based on actions. |
| 2 | Party Leader | Player with leadership role over a Party; invites members, manages roles, schedules activities, and can start group Boss Fights. |
| 3 | Guild Master | Owner/admin of a Guild; manages membership, roles, guild page, events, and moderation. |
| 4 | Verified Lecturer (Special Status) | Instructor verified by Game Master; can create and manage Guilds and official events, review student progress, and provide feedback. |
| 5 | Game Master (Admin) | Platform administrator; owns Elective Library and University Curriculum; configures system, verifies lecturers, and moderates community/leaderboards. |

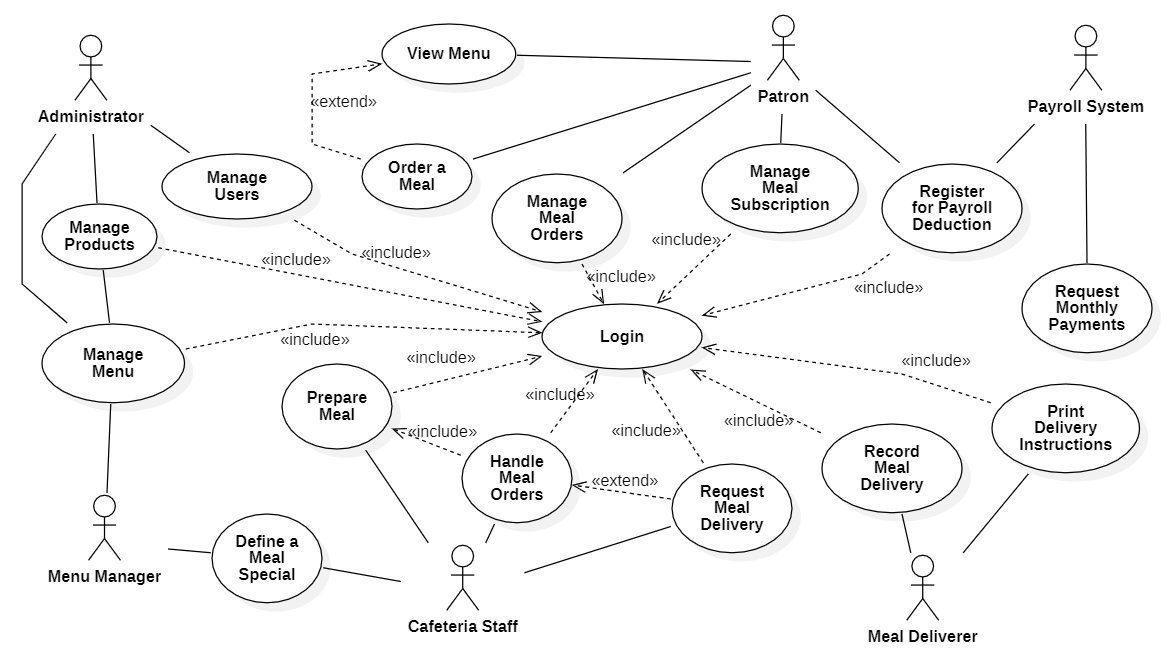
### 

### 2.2 Use Cases

*[A use case (UC) describes a sequence of interactions between a system and an external actor that results in the actor being able to achieve some outcome of value. The names of use cases are always written in the form of a verb followed by an object. Select strong, descriptive names to make it evident from the name that the use case will deliver something valuable for some user]*

#### 2.2.1 Diagram(s)

*[Provide the UC diagram(s) to show the actor-UCs and UC-UC relationships like the sample below. You can have multiple UC diagrams for the system]*



#### 2.2.2 Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Actors** | **Use Case Description** |
| UC-01 | User Registration and Curriculum-Career Onboarding | Player (Student) | Register and complete the 3-step Character Creation flow (Curriculum → Career Path → Skill-based roadmap). |
| UC-02 | Academic Document Enhancement and Skill Tree Integration | Player (Student) | Upload academic documents to enhance skill tree visualization, Arsenal personalization, and enable FPTU verification. |
| UC-03 | Curriculum-Based Skill Tree Visualization and Career Navigation | Player (Student) | View interactive skill tree mapped to curriculum; explore prerequisites, dependencies, and career navigation. |
| UC-04 | AI-Powered Curriculum Quest Generation and Career Enhancement | Player (Student) | Generate primary quest line from curriculum and supplementary gap quests aligned to career path. |
| UC-05 | Curriculum-Based Party Creation and Career Collaboration | Player (Student) | Create a party, invite members or open for join; configure study/collaboration rules. |
| UC-06 | Meeting Scheduling and Management | Party Members | Schedule and manage study meetings; record content and generate summaries. |
| UC-07 | Browser Extension Integration | Player (Student) | Extract academic info/web content; organize into Arsenal; provide context-aware suggestions linked to quests. |
| UC-08 | Guild Management System | Guild Master, Verified Lecturer | Create and manage guilds; share materials; basic analytics and announcements; educational governance retained by Admin. |
| UC-09 | Curriculum-Career Boss Fight Assessment System | Player (Student) | Attempt boss fights (WebGL), receive readiness assessments, and convert scores to skill XP. |
| UC-10 | Collaborative Study Sessions | Party Members | Conduct collaborative study sessions with shared objectives and resource sharing. |
| UC-11 | Event Management with Game Master Approval | Guild Master, Game Master (Admin) | Create and manage competitive events via wizard; submit for admin approval when required. |
| UC-12 | Code Battle Participation | Player (Student) | Participate in real-time code battles; submit solutions with automated scoring and live rankings. |
| UC-13 | Real-time Notifications and Updates | Player (Student) | Receive in-app/email/push notifications with configurable preferences for quest and system updates. |
| UC-14 | Performance Analytics and Monitoring | Verified Lecturer | Visualize performance analytics and monitor progress across students/classes. |
| UC-15 | System Integration and API Management | Game Master (Admin) | Manage system integrations, APIs, logging, and synchronization. |
| UC-16 | Elective Library Curation & Approval (Admin-Owned) | Game Master (Admin) | Curate, review, approve, and publish elective content aligned to skill nodes. |
| UC-17 | University Curriculum Import & Administration (Admin-Owned) | Game Master (Admin) | Import curriculum via JSON (MVP), manage versions/activations, and reporting exports. |

## 3. Functional Requirements

### 3.1 System Functional Overview

*[Provide functionality overview of software system: screen flow, screen descriptions, system user roles, screen authorization, non-screen functions, ERD]*

#### 3.1.1 Screens Flow

*[This part shows the system screens and the relationship among screens. You can draw the Screens Flow for the system in the form of diagram as below. Please note that beside the normal flat screen, we might have the oval notation for pop-up screen (Import Order) or a screen with multiple information tab (Order Details), etc. You may also use text or background format for different visuality purpose]*



#### 3.1.2 Screen Descriptions

*[Provide the descriptions for the screens in the Screens Flow above]*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Screen** | **Description** |
| 1 | Order Meals | Create Order | <<Screen Brief description>> |
| 2 | Order Meals | Change Order |  |
| 3 | .. |  |  |

#### 3.1.3 Screen Authorization

*[Provide the system roles authorization to the system features (down to screens, and event to the screen activities if applicable) in the table form as below – replace Role1, Role2,… with your specific system user role names]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Role-Name1** | **Role-Name2** | **Role-Name3** | **…** |
| <<Screen Name1>> | X |  | X | X |
| <<Screen Activity>> |  |  | X | X |
| <<Screen Name2>> | X |  | X |  |
| Query All Data | X |  |  |  |
| Query Own Data |  |  | X |  |
| Query Managed Data |  |  | X |  |
| Add New Data |  |  | X | X |
| Update All Data |  |  |  | X |
| Update Own Data |  |  |  | X |
| Update Managed Data |  |  |  | X |
| Delete Data |  |  |  |  |
| … |  |  |  |  |

#### 3.1.4 Non-Screen Functions

*[Provide the descriptions for the non-screen system functions, i.e batch/cron job, service, API, etc.]*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **System Function** | **Description** |
| 1 | <<Feature Name>> | <<Function Name1>> | <<Function Name1 Description>> |
| 2 | … |  |  |

#### 3.1.5 Entity Relationship Diagram

*[Provide the entity relationship diagram and the entity descriptions in the table format as below]*



**Entities Description**

|  |  |  |
| --- | --- | --- |
| **#** | **Entity** | **Description** |
| 1 | User |  |
| 2 | Meal |  |
| 3 | Meal Subscription |  |
| 4 | … |  |

### 3.2 <<Feature Name 1>>

#### 3.2.1 <<Function Name 1>>

*[A function can be a screen or a non-screen function (listed in the part 3.1.5 above). In this part, you need to provide the details on the related function, focus on mentioning below information*

* *Function trigger: how this function is triggered (navigation path, a timing frequency, etc.*
* *Function description: actors/roles, purpose, interface, data processing, etc.*
* *Screen layout: mock-up prototype of the screen, sample below is for Manage Products screen*

**

* *Function Details: provide explanation for the data, validation, business rules, functionalities (for both normal cases and abnormal cases), etc. of the function so that the reader can image how it work.*

*]*

#### 3.2.2 <<Function Name 2>>

…

### 3.3 <<Feature Name 2>>

…

## 4. Non-Functional Requirements

### 4.1 External Interfaces

*[This section provides information to ensure that the system will communicate properly with users and with external hardware or software/system elements.]*

### 4.2 Quality Attributes

*[List all the required system characteristics (quality attributes) specification. Some of the possible attributes are provided with the guide/descriptions are mentioned here]*

#### 4.2.1 Usability

*[This section includes all those requirements that affect usability. For example, specify the required training time for a normal users and a power user to become productive at particular operations specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards]*

#### 4.2.2 Reliability

*[Requirements for reliability of the system should be specified here. Some suggestions follow:*

*Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, and so on.*

*Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.*

*Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?*

*Accuracy—specifies precision (resolution) and accuracy (by some known standard) that is required in the system’s output.*

*Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).*

*Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]*

#### 4.2.3 Performance

*[The system’s performance characteristics are outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.*

*Response time for a transaction (average, maximum)*

*Throughput, for example, transactions per second*

*Capacity, for example, the number of customers or transactions the system can accommodate*

*Resource utilization, such as memory, disk, communications, and so forth.]*

#### 4.2.4 …

## 5. Requirement Appendix

*[Provide business rules, common requirements, or other extra requirements information here]*

### 5.1 Business Rules

*[Provide common business rules that you must follow. The information can be provided in the table format as the sample below]*

|  |  |
| --- | --- |
| ID | Rule Definition |
| BR-01 | Character Creation (3-step) is mandatory before unlocking the dashboard: 1) Route (Curriculum), 2) Class (Roadmap.sh specialization), 3) Skill-based roadmap generation. |
| BR-02 | Academic document upload is optional and influences skill tree visualization and Arsenal personalization; verified FPTU documents impact quest generation and calendar integration. |
| BR-03 | System maintains predefined curriculum/syllabus database for supported routes/subjects. |
| BR-04 | Primary quest line is generated directly from curriculum analysis and organized by semester; integrates the FPTU academic calendar. |
| BR-05 | AI gap analysis generates supplementary quests to bridge curriculum with selected career path (roadmap.sh). |
| BR-06 | Skill tree is populated and leveled from academic data; shows prerequisites/dependencies; supports knowledge decay and cross-skill synergies. |
| BR-07 | Arsenal is a Notion-like workspace; notes link to skill nodes and display contribution indicators. |
| BR-08 | Boss Fights are Unity WebGL exams with difficulty tiers; scores convert to skill XP using defined distribution rules. |
| BR-09 | Leaderboards include global/class/major/guild/event categories; real-time updates for active events, batch for quests/skills; seasonal resets. |
| BR-10 | Main quest line is dynamically adjustable based on performance, schedule, preferences, and academic updates, with 24-hour rollback. |
| BR-11 | Browser extension organizes academic/web content into Arsenal and suggests context-aware actions; integrates with FPTU portal for real-time sync. |
| BR-12 | Co-op Boss Fight sessions follow allowed configuration ranges and server-authoritative rules; React Unity WebGL bridge enables session control and reporting. |
| BR-13 | Guilds can be created by Players or Verified Lecturers; Verified Lecturers get enhanced analytics/quest tools; Admin retains publishing/governance. |
| BR-14 | Competitive event platform includes Code Arena and Guild Events with event wizard, room assignment, scoring, and analytics. |
| BR-15 | Objective types map to workspace UIs; completion criteria vary by type; automated project verification; knowledge graph drives micro-objectives; reward cascade and contextual unlocks. |
| BR-16 | Quest memory preserves continuity across semesters; failed courses trigger adaptive recovery quests. |
| BR-17 | Core platform capabilities: notifications, structured logging, real-time sync/data versioning, optimized content delivery. |

### 5.2 Common Requirements

*[Fill all the common requirements here..]*

### 5.3 Application Messages List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Message code** | **Message Type** | **Context** | **Content** |
| 1 | MSG01 | In line | There is not any search result | *No search results.* |
| 2 | MSG02 | In red, under the text box | Input-required fields are empty | *The \* field is required.* |
| 3 | MSG03 | Toast message | Updating asset(s) information successfully | *Update asset(s) successfully.* |
| 4 | MSG04 | Toast message | Adding new asset successfully | *Add asset successfully.* |
| 5 | MSG05 | Toast message | Confirming email of asset hand-over is sent successfully | *A confirmation email has been sent to {email\_address}.* |
| 6 | MSG06 | Toast message | Resetting asset information successfully | *Return asset(s) successfully.* |
| 7 | MSG07 | Toast message | Deleting asset information successfully | *Delete asset(s) successfully.* |
| 8 | MSG08 | In red, under the text box | Input value length > max length | *Exceed max length of {max\_length}.* |
| 9 | MSG09 | In line | Username or password is not correct when clicking sign-in | *Incorrrect user name or password. Please check again.* |

### 5.4 Other Requirements…