

Software Requirement

Specification

for

<UniQuest>

Version 1.0

Prepared by <Gabriel John Q Lambojon>

<organization>

<December 2025>

Pls. pattern the title page with Project Plan title page

TABLE OF CONTENTS

1. Introduction

1.1 Purpose.....
1.2 Scope.....
1.3 Intended Audience.....

2. Overall Description

2.1 Product Perspective.....
2.2 Product Functions.....
2.3 User Classes and Characteristics.....
2.4 Design and Implementation Constraints.....
2.5 User Documentation.....
2.6 Assumptions and Dependencies.....

3. Operating Environment

3.1 User Interfaces.....
3.2 Hardware Interfaces.....
3.3 Software Interfaces.....
3.4 Communication Interfaces.....

4. Functional Requirements

4.1 System Features.....

5. Non-functional Requirements

5.1 Performance Requirements.....
5.2 Safety Requirements.....
5.3 Security Requirements.....
5.4 Software Quality.....
5.5 Business Rules.....

6. Other Requirements

Appendix A: Glossary (Definitions, Acronyms, Abbreviations)

Appendix B: Analysis Model

- Use Case Diagram
- ERD
- Activity and
- Process ModelReference

1. Introduction

1.1. Purpose

This Software Requirements Specification (SRS) document provides a comprehensive description of **UniQuest**, a gamified mobile application designed to transform the university experience into an engaging adventure. The application aims to make academic and social participation more interactive by incorporating game-like elements into students' daily university activities. It is intended to enhance motivation, streamline university processes, and promote student involvement through an immersive and rewarding experience.

The document details the functional and non-functional requirements of UniQuest, outlining its core features such as quest-based task completion, point accumulation, achievement unlocking, and in-game currency systems. Students will be able to complete real-life university tasks including enrollment steps, academic requirements, and social or community challenges as if they were quests within a game. By doing so, players earn rewards that can be used for in-app customization and progression.

This SRS will serve as the foundation for the subsequent system design and development phases, ensuring all stakeholders have a clear understanding of what the system will do and how it will operate. It will also guide the technical and design specifications necessary for implementing UniQuest, establishing a structured framework for development and evaluation throughout the project lifecycle.

1.2. Project Scope

The **UniQuest** project aims to develop a gamified mobile application that enhances student engagement by transforming daily academic and campus activities into interactive quests and challenges. The system promotes motivation, collaboration, and healthy competition among students while helping them track academic progress in a fun and organized way. It will also provide university administrators with tools to manage quests, monitor engagement, and encourage student participation. The project covers the complete software development lifecycle from analysis and design to implementation, testing, and deployment—and ensures scalability for future feature enhancements and system integrations.

Key Components:

- **Gamified Student Life** – Converts everyday academic and campus tasks into quests and interactive challenges.
- **Achievement & Rewards System** – Allows students to earn badges, XP that can be used for unlocking in-game items and customization.
- **Interactive Campus Map** – Integrates an API-based university map to help students navigate key campus locations and quest areas.

- **Social & Competitive Features** – Includes leaderboards, daily challenges, and batchmate competitions to encourage interaction and friendly rivalry.
- **Customization with Cosmetics** – Lets players spend earned in-game currency to personalize avatars, themes, and quest trackers for a unique experience.

Out of Scope

The following items are excluded from the current scope of the **UniQuest** project:

- **Third-Party System Integration** – Integration with academic or administrative systems outside the university's existing infrastructure will not be included.
- **Web-Based Version** – The project will focus solely on the mobile application; no web version will be developed in this phase.
- **Hardware-Dependent Features** – Functions requiring physical devices, such as NFC based attendance or ID scanning, are excluded.
- **Server Hosting and Maintenance** – Ongoing hosting, maintenance, and technical support beyond initial deployment are not part of this phase.
- **Long-Term Content Management** – Continuous quest creation and in-app content updates after launch will not be covered.
- **Third-Party API Licensing** – Paid or external API services not originally included in the project plan are excluded.
- **Cross-Institution Expansion** – Multi-university or inter-campus integration will not be implemented during the initial release.

1.3. Intended Audience

University Administration	Officials responsible for reviewing and approving system requirements, managing academic quests, and serving as the primary system owners.
---------------------------	--

Project manager	Oversees the entire development process, ensures milestones are met, and coordinates communication among all stakeholders.
System Analyst	Analyzes user and institutional needs, translates them into technical requirements, and ensures alignment with project objectives.

Developer	Designs, codes, and implements the UniQuest application based on the defined functional and non-functional requirements.
Tester	Conduct testing to ensure system reliability, usability, and compliance with specifications.

Integration partners	Include potential third-party systems or databases that may connect with UniQuest in future phases.
Students (End users)	Primary users who will interact with the application to complete quests, earn rewards, and enhance their academic experience.
Faculty Members	May create or recommend academic quests, monitor student participation, and encourage engagement.

2. Overall Description

2.1. Product Perspective

The **UniQuest** mobile application is a new standalone system designed to enhance student engagement and campus participation through gamification. It transforms academic, social, and extracurricular tasks into interactive quests, making the university experience more engaging and rewarding. Initially, UniQuest will operate as an independent mobile platform accessible to students, faculty, and staff across all university departments and year levels.

The system includes an **administrative dashboard** for university personnel to manage quests, monitor engagement, and track overall participation. It also integrates an **interactive campus map** powered by a location-based API to help users navigate key university areas and complete location-specific quests.

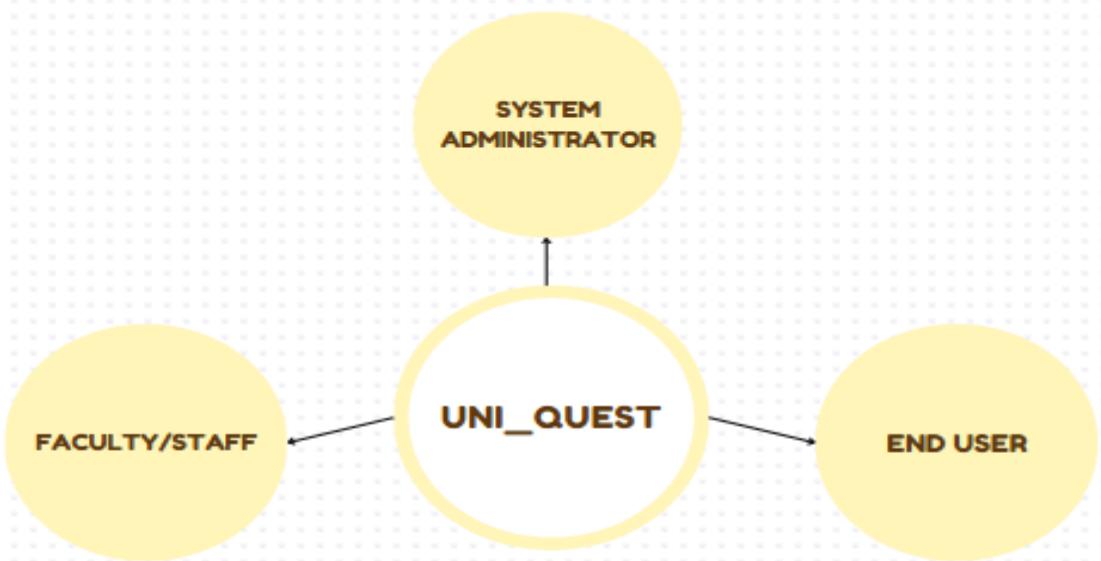
While UniQuest is designed as a standalone application for its initial phase, it is structured with scalability in mind for future integration with institutional systems such as learning management platforms, student information systems, and academic record databases. Planned enhancements for future phases include **event scheduling**, **academic performance tracking**, and **cross campus expansion** to support a wider university network.

The platform will interface with:

- **University Map API** – Provides navigation and location-based quest functionality within the campus.

- **Administrative Dashboard (Client/Faculty/Staff)** – Used to create, manage, and evaluate quests
- and student engagement (*Future development*). Complete quests, earn rewards, and interact with gamified academic and social challenges.
- **System Administrators (Developers/IT)** – Manage system performance, maintenance, and security.
- **Students (End Users)** – Complete quests, earn rewards, and interact with gamified academic and social challenges.

Diagram of entity interaction - UniQuest



2.2. Product Functions

The **UniQuest** mobile application provides a gamified platform that transforms academic, social, and extracurricular activities into engaging quests. It is designed to enhance student motivation, participation, and overall campus involvement. While faculty and staff currently share the same app functionalities as students, future updates will include administrative features for quest creation and performance monitoring through the UniQuest Admin Dashboard.

Key Functional Areas:

1. Students (End Users)

- **Quest Participation** – Complete academic, social, and campus-related quests to earn experience points (XP), badges.
- **Achievements and Rewards** – Unlock badges and cosmetic items as recognition for task completion and consistent participation.
- **Profile Customization** – Use earned in-game currency to personalize avatars, themes, and quest trackers.
- **Interactive Campus Map** – Navigate the university using a location-based map showing quest areas and key locations.
- **Leaderboards and Progress Tracking** – View personal progress, rankings, and compare achievements with peers.
- **Notifications and Updates** – Receive alerts for new quests, events, and achievements.
- **User Authentication** – Secure login and data protection for personalized access.
- **In-App Notifications:** The system provides immediate visual feedback (toasts or pop-ups) to the user upon the successful completion of actions, such as finishing a quest or equipping an item.
- **Interactive Campus Map:** The system utilizes an external Map API (e.g., Google Maps/Mapbox) to render the university layout and overlays quest pins fetched from the database.

2. Faculty and Staff (Future Development)

- **Quest Management** – Used to create, manage, and evaluate quests and student engagement.
- **Performance Monitoring** – Review participation data and identify trends in student engagement.
- **Event and Activity Oversight** – Post and manage campus events linked to quests or achievements.
- **Analytics Dashboard** – Access summarized reports of student progress and participation
 - Rates.

3. System Administrators (IT / Development Team)

- **System Maintenance** – Ensure reliable operation, software updates, and performance optimization.
- **Database Management** – Manage user accounts, quest data, and in-app currency records.
- **Security and Access Control** – Protect user information through secure authentication and authorization systems.
- **Integration Management** – Configure and manage API connections (e.g., University Map API).
- **Backup and Recovery** – Maintain regular data backups and system restore options.

4. Supporting System Functions

- **User Registration and Login** – Allows creation, authentication, and management of user accounts.
- **Notification System** – Sends in-app and push notifications for quests, rewards, and events.
- **Analytics and Reporting** – Collects participation metrics and engagement statistics for analysis.
- **Scalability Support** – Enables addition of new modules such as event scheduling, academic tracking, and expanded admin tools in future phases.

2.3. User Classes

1. Students (End Users)

Characteristics:

- Primarily undergraduate and graduate students from Cavite State University.
- Varying levels of technical familiarity and mobile app usage experience.
- Regular access to smartphones and campus internet, though some may have intermittent connectivity.

Primary Needs:

- Engage with academic and extracurricular tasks in an interactive and rewarding way.
- Track progress through quests, achievements, and leaderboards.
- Access campus maps, events, and personalized notifications easily.

2. Client, Faculty and Staff

(Future Development)

Characteristics:

- Academic instructors, department heads, and administrative personnel.
- Moderate to high technical literacy, depending on role and department.
- Responsible for managing events, monitoring engagement, and promoting campus activities.

Primary Needs:

- Tools to create, manage, and assess student quests and engagement levels.
- Access to dashboards showing student progress and participation trends.
- A simplified system to coordinate with other faculty or departments for event management

3. System Administrators

Characteristics:

- Members of the UniQuest technical team or IT department of the university.
- Possess strong technical skills in database management, system maintenance, and network security.
- Responsible for ensuring the reliability, scalability, and security of the platform.

Primary Needs:

- Manage user accounts, data integrity, and system updates.

- Monitor performance metrics and resolve technical issues promptly.
- Oversee system backups, integration, and overall platform stability.

3. University Administrators

Characteristics:

- University officials, program directors, and organizational units overseeing student activities.
- Involved in decision-making and system performance evaluation.
- Require summarized, high-level insights rather than operational details.

Primary Needs:

- Access to analytics and reports on student engagement and campus activity participation.
- Ability to evaluate the impact of the gamified system on academic and extracurricular involvement.
- Data-driven insights to guide institutional policies and engagement strategies.

2.4 Design and Implementation Constraints

The following constraints will impact the design and implementation of the UniQuest platform:

Technical Constraints

- Must function as a **standalone mobile application** (Android and iOS) with a web-based admin dashboard.
- Must **integrate with the official University Map API** and allow for future linkage with academic or event management systems.
- Must be **accessible through standard mobile and web browsers** without requiring additional plugins.
- Must operate reliably even under **limited or unstable internet connectivity**, utilizing local caching or offline mode where possible.
- Must ensure a **user-friendly and accessible interface**, supporting students, faculty, and staff, including **users with disabilities**.

Regulatory Constraints

- Must comply with the **Data Privacy Act of 2012 (Republic Act No. 10173)** of the Philippines.
- Must adhere to **university data security policies** and maintain confidentiality of student and staff records.

- Must follow **international data protection best practices** (e.g., GDPR principles) to safeguard user information.

Business Constraints

- Must be **implemented in phases**, prioritizing the gamification and task management features for the initial release.
- Must be **scalable** to accommodate additional modules such as event scheduling, academic tracking, and faculty management tools in later phases.
- **System maintenance and updates** will be exclusively performed by the UniQuest development team. The university or its departments will coordinate with the team for any support requests, updates, or feature expansions.
- Must be **cost-efficient** and adaptable for use across different university departments without requiring major infrastructure changes.

User Constraints

- Must accommodate users with varying levels of technical proficiency, ensuring ease of navigation for both students and staff.
- Must conform to accessibility standards to ensure inclusivity across all user groups.

2.5. User Documentation

All user documentation will be developed and maintained by the **UniQuest development team** to ensure consistency with each system update.

1) Online Help System

- In-app help and FAQs for students and staff.
- Step-by-step guides and troubleshooting tips.
- Content updated regularly by the development team

2) User Manual

- **Student Guide** – Account setup, quests, rewards, and customization.
- **Faculty & Staff Guide** – (*Future*) Managing quests and viewing progress.
- **Admin Guide** – User management and system monitoring.

All manuals are digital and revised alongside system updates.

3) Training Materials

- Quick reference sheets, tutorial videos, and presentation slides.
- Used for onboarding new users and university personnel.
- Managed and updated by the UniQuest team.

2.6. Assumptions and Dependencies Assumptions

- Students, faculty, and staff have access to smartphones or compatible devices capable of running the UniQuest mobile application.
- The university provides stable internet connectivity within campus premises for app access and synchronization.
- Users will regularly engage with the system to maintain active participation and data relevance.
- The university administration will support the integration of UniQuest in student and campus activities.
- Feedback and suggestions from users will be communicated through official channels to the UniQuest development team for system improvement.

Dependencies:

- **External Map Service Availability:** The functioning of the Interactive Campus Map is strictly dependent on the availability and valid licensing of the third-party Map Provider API (e.g., Google Maps Platform or Mapbox). Any downtime from the provider will result in the map layer failing to render.
- **Supabase Connectivity:** The map and quest features depend on a stable connection to the Supabase Database to retrieve coordinate data (**map_markers**) and synchronize user progress.
- **Device Permissions:** The In-App Notification system assumes that the user has not disabled OS-level restrictions (e.g., "Do Not Disturb") that would block the application from displaying toast messages or pop-up alerts.
- **Campus Network Infrastructure:** The campus network and servers must support UniQuest data traffic and security protocols to ensure reliable access for students on site.
- **Future Integration Permissions:** Future features such as event scheduling and academic tracking depend on successful integration with university systems and granting of necessary data access permissions.
- **Operational Support:** Continuous collaboration with university departments is required for updates, testing, and feature validation. Long-term operation depends on ongoing maintenance provided by the UniQuest development team.

3.) Operating environment

The system shall provide user interfaces for the following user types:

Students (End Users) -

- **Registration and Profile Interface** – Allows students to create accounts, customize avatars, and manage their in-app profiles.
- **Quest and Task Interface** – Displays available, ongoing, and completed quests; shows rewards, XP, and progress tracking.
- **Dashboard Interface** – Provides an overview of achievements, quest history, leaderboards, and current event participation.
- **Notification and Messaging Interface** – Sends alerts for new quests, events,

deadlines, and inter-user communications.

- **Campus Map Interface** – Offers an interactive map with building details, event markers, and navigation features.

Faculty and Staff (TBD)

- **Quest Management Interface** – Enables creation, modification, and publishing of academic or extracurricular quests.
 - **Student Progress Dashboard** – Displays participation data, performance statistics, and completion rates.
 - **Event and Activity Management Interface** – Allows scheduling and tracking of university events or student engagements.
 - **System Administrators (Developers / IT Team)**
- **Content Management Interface** – Manage event listings, news updates, and featured activities.
 - **User, Roles, and Permissions Interface** – Manage student, staff, and administrator Accounts.
 - **System Configuration Interface** – Control feature availability, maintenance schedules, and API integrations.
 - **Reporting and Analytics Interface** – Generate summaries on app engagement, user activity, and quest participation.

University Administrators (Management Team)

- **Monitoring and Analytics Interface** – View overall student engagement trends and usage statistics.
- **Data Visualization Dashboard** – Access summarized charts and reports for institutional review.
- **Feedback and Coordination Interface** – Communicate directly with the UniQuest development team for updates or issue reports.
- **Guest Users**
 - The system **SHALL** ensure clear display of **privacy policies** and consent banners in compliance with the **Data Privacy Act of 2012**, providing transparency in data collection and use.
 - Guest users **SHALL** be prompted to register or log in when attempting to participate in quests, earn rewards, or track progress.

All user interfaces shall adhere to the following requirements:

- The system **SHALL** have a **user-friendly and dynamic UX/UI**, displaying featured quests, achievements, and events in a visually engaging layout.
- **Responsive design** ensuring compatibility across mobile devices and desktop browsers.
- **Multilingual support** (English, with future expansion for local dialects).
- Compliance with **Web Content Accessibility Guidelines (WCAG 2.1 or later)** for inclusivity and accessibility.

- **Consistent navigation and design patterns** for all user types.
- **Context-sensitive help and guidance** within each major module.

3.2 Hardware Interfaces

The system shall interface with the following hardware components:

Cloud Infrastructure (Supabase)

- Supabase Authentication Services
- Supabase PostgreSQL Database
- Supabase Storage Buckets for file and media assets
- Supabase Edge Functions for backend logic
- Supabase Realtime Services for live updates and data synchronization

Network Infrastructure

- Internet connectivity for real-time synchronization and cloud access
- Secure HTTPS communication via Supabase-managed servers
- Network firewalls and routers provided by the university or mobile carriers

Client Devices

- Smartphones (Android 10 / iOS 13 or higher)
- Tablets(***Future development***)
- Laptops or desktop computers (for admin dashboard access via web interface)
(Future development)

The system shall rely on Supabase's cloud infrastructure to minimize local hardware dependencies and ensure scalability, reliability, and secure data access across all devices used by students, faculty, and staff.

3.3 Software Interfaces

The system shall interface with the following software systems:

1) Supabase Services - The app shall communicate with Supabase Authentication, PostgreSQL Database, Supabase Storage, and Supabase Realtime/Edge Functions through secured API calls. All data exchanged with Supabase shall use HTTPS with TLS encryption and comply with privacy and data protection standards .

2) Map Service API - The application shall interface with an external mapping provider (e.g., Google

Maps Platform or Mapbox SDK) to render the visual map layer. The system shall communicate with this API over HTTPS to retrieve map tiles and geospatial data.

3) Analytics and Reporting Tools - Integration with Supabase Analytics Dashboard and third-party data visualization tools. This includes export capabilities for performance reports and engagement summaries.

All software interfaces shall include detailed documentation covering data formats, communication protocols, authentication methods, and security requirements to ensure reliable and secure data exchange between the app and external systems.

3.4 Communication Interfaces

The UniQuest system shall rely on secure and reliable communication interfaces to ensure smooth data exchange between users, the mobile application, and the Supabase backend.

1) Network Communication

- The system shall use **HTTPS protocol** for all client–server communications to ensure data security and privacy.
- The app shall function over **Wi-Fi** and **mobile data (3G, 4G, 5G)** connections.
- Communication shall be optimized to handle limited or unstable internet connections, allowing temporary offline access with automatic data synchronization once reconnected.

2) Cloud Communication (Supabase Services)

- The app shall communicate with **SupabaseAuthentication**, **Cloud Storage**, and **Cloud Messaging** through secured API calls.
- All data exchanged with Supabase shall use **TLS encryption** and comply with privacy and data protection standards.

4.0 Functional Requirements

1.1 System Features

Feature 1: Gamified Student Life

Description and Priority: Transforms campus and academic tasks into quests and challenges to enhance engagement. (*Priority: High*)

Stimulus/Response Sequences: User selects a quest → System displays objectives → User completes the task → System grants XP and rewards.

Functional Requirements:

- **REQ-1:** The system shall allow students to view and accept available quests.
- **REQ-2:** The system shall update quest status upon completion.
- **REQ-3:** The system shall award XP for completed quests.

Feature 2: Achievement and Rewards System

Description and Priority: Awards points and badges to motivate continued participation.
(Priority: High)

Stimulus/Response Sequences: User completes a quest → System validates progress → System updates leaderboard and awards rewards.

Functional Requirements:

- **REQ-1:** The system shall record earned XP, badges.
- **REQ-2:** The system shall display a leaderboard of top users.

Feature 3: Interactive Campus Map

Description and Priority: Displays a map of the university with marked quest areas and locations.
(Priority: Medium)

Stimulus/Response Sequences: User opens the map → System loads campus layout → User selects a building → System shows details.

Functional Requirements:

- **REQ-1:** The system shall render a dynamic map view using the integrated Map API.
- **REQ-2:** The system shall query the map_markers table in Supabase to display active quest locations as pins on the map layer.
- **REQ-3:** The system shall allow users to tap on map markers to view quest details.

Feature 4: Social and Competitive Features

Description and Priority: Promotes collaboration through leaderboards, daily challenges, and competitions. *(Priority: Medium)*

Stimulus/Response Sequences: User joins a challenge → System tracks progress → System updates rankings in real time.

Functional Requirements:

- **REQ-1:** The system shall generate leaderboards for individuals and groups.
- **REQ-2:** The system shall support batch or department-based competitions.

Feature 5: Customization with Cosmetics

Description and Priority: Allows users to personalize avatars, themes, and quest

trackers using in-game currency. (*Priority: Low*)

Stimulus/Response Sequences: User opens the customization menu → System displays available items → User purchases an item → System updates user profile.

Functional Requirements:

- **REQ-1:** The system shall store user-selected cosmetics and themes.
- **REQ-2:** The system shall deduct Quest EXP upon purchase.

Feature 6: Administrative Dashboard

Description and Priority: Provides monitoring tools for activity tracking and content updates. (*Priority: High*)

Stimulus/Response Sequences: Admin logs in → System displays dashboard → Admin views reports or manages content.

Functional Requirements:

- **REQ-1:** The system shall allow admins to view user engagement statistics.
- **REQ-2:** The system shall support quest creation and content management.

Feature 7: In-App Notification System

Description and Priority: Provides immediate visual confirmation of user actions within the application interface. (*Priority: Low*)

Stimulus/Response Sequences: User completes an action (e.g., "Mark as Complete") → System verifies local logic → System displays temporary success message.

Functional Requirements:

- **REQ-1:** The system shall display a temporary pop-up (toast message) at the bottom of the screen when a user successfully accepts or completes a quest.
- **REQ-2:** The system shall show a visual error alert if an action cannot be completed (e.g., "GPS location not found").
- **REQ-3:** The system shall update the UI counters (XP/Coin balance) immediately after a notification is acknowledged.

5. Non functional Requirements

Performance Requirements

- **REQ-PERF-01:** The system shall provide real-time synchronization of data between users and the Supabase backend.
- **REQ-PERF-02:** The mobile application shall load primary pages within 3 seconds on a stable connection.
- **REQ-PERF-03:** The backend (Supabase) shall support at least 1,000 concurrent users without degradation in response times.
- **REQ-PERF-04:** Supabase database shall efficiently handle CRUD operations and real-time updates.

5.2 Safety Requirements

- **REQ-SAFE-01:** Regular data backups shall be maintained in Supabase's secure storage to prevent data loss.
- **REQ-SAFE-02:** The system shall prevent accidental data deletion by requiring confirmation dialogs for critical actions.
- **REQ-SAFE-03:** Fail-safe mechanisms shall ensure that unsaved data is not lost during network interruptions.

5.3 Security Requirements

- **REQ-SEC-01:** The system shall use Supabase Authentication for secure user login and session management.
- **REQ-SEC-02:** All data in transit shall be encrypted using HTTPS (TLS 1.2 or later).
- **REQ-SEC-03:** Access control rules shall restrict sensitive data to authorized users only (e.g., admins, faculty).
- **REQ-SEC-04:** The system shall comply with relevant data-protection regulations (e.g., GDPR principles).

5.4 Software Quality Usability

- **REQ-QUAL-01:** The user interface shall be intuitive and accessible, meeting WCAG 2.1 standards.
- **Reliability & Availability**
- **REQ-QUAL-02:** The application shall maintain at least 99% uptime, leveraging Supabase's managed infrastructure.
- **Maintainability**
- **REQ-QUAL-03:** The system shall be maintainable and upgradable by the UniQuest development team without external dependency.
- **REQ-QUAL-04:** Code quality shall follow modular design principles for easier debugging and feature updates.

5.4 Business Rules

- **REQ-BR-01 (Ownership & Maintenance):** Only the UniQuest development team shall maintain and update the application; the school organization will contact them for feature changes.
- **REQ-BR-02 (Feature Approval):** New features and modules shall undergo review and approval before deployment.
- **REQ-BR-03 (Accessibility & Scalability):** The application shall prioritize accessibility

- and scalability using Supabase's open-source platform.
- **REQ-BR-04 (Standards Compliance):** All development shall adhere to IEEE 830-1998 standards for documentation and requirement traceability.

6. Other Requirements

Appendix A:

Glossary (Definitions, Acronyms, Abbreviations)

Glossary

Accessibility – The practice of designing the app so it can be used by individuals with disabilities, including support for screen readers, proper color contrast, and keyboard navigation.

API (Application Programming Interface) – A set of rules and methods that allows different software systems to communicate and exchange data.

Authentication – The process of verifying a user's identity (e.g., logging in with email and password).

Database – A structured system used to store, manage, and retrieve application data.

GDPR (General Data Protection Regulation) – A European Union regulation governing the protection, privacy, and handling of personal data.

HTTPS (Hypertext Transfer Protocol Secure) – A secure communication protocol that encrypts data exchanged between the client and server using TLS.

TLS (Transport Layer Security) – A cryptographic protocol used to secure communication over networks. TLS provides the encryption used in HTTPS.

IEEE 830-1998 – An international standard that defines the structure and best practices for writing Software Requirements Specifications (SRS).

Integration – The process of connecting the system with external platforms, services, or APIs to exchange data or functionality.

Load Balancer – A network component that distributes incoming traffic across multiple servers to improve reliability and performance.

Multilingual Support – A system capability that allows the interface and content to be displayed in more than one language.

Navigation (Map) – The feature that allows users to view a campus map and explore locations. In UniQuest's current version, this is limited to static or online viewing (no GPS-based navigation).

Notifications – Alerts sent to users, including reminders, updates, announcements, or system messages.

Online Platform – A system that operates entirely over the internet and requires an active connection.

Real-Time Data – Information that updates immediately without delay when changes occur in the system.

Responsive Design – A design approach ensuring the user interface adapts smoothly to different screen sizes and orientations.

SRS (Software Requirements Specification) – A document that describes the system's features, requirements, and constraints in detail.

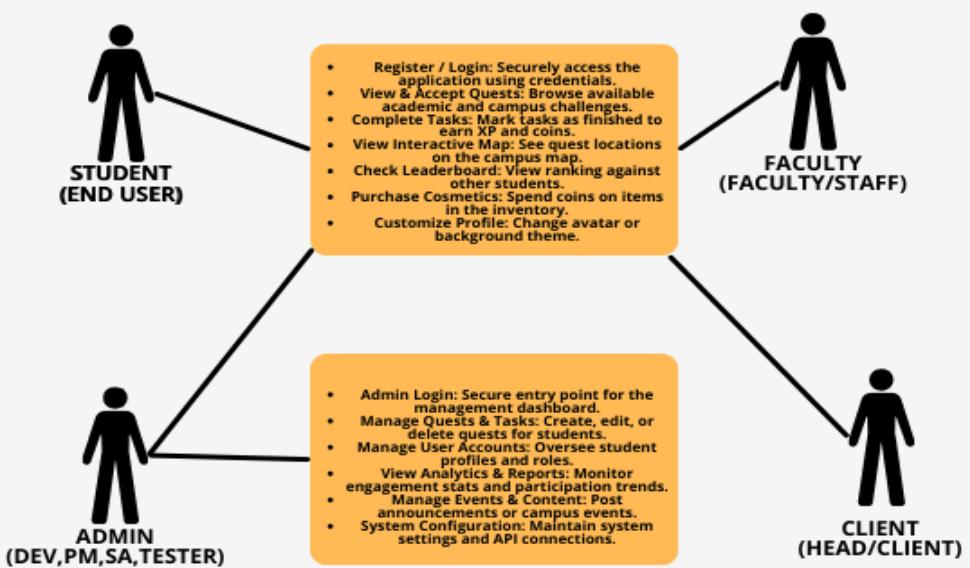
Supabase – A backend-as-a-service platform offering authentication, PostgreSQL database, storage, and API services used as the system backend.

System Administrator – A user responsible for managing system settings, maintaining data, overseeing user accounts, and performing technical administration tasks.

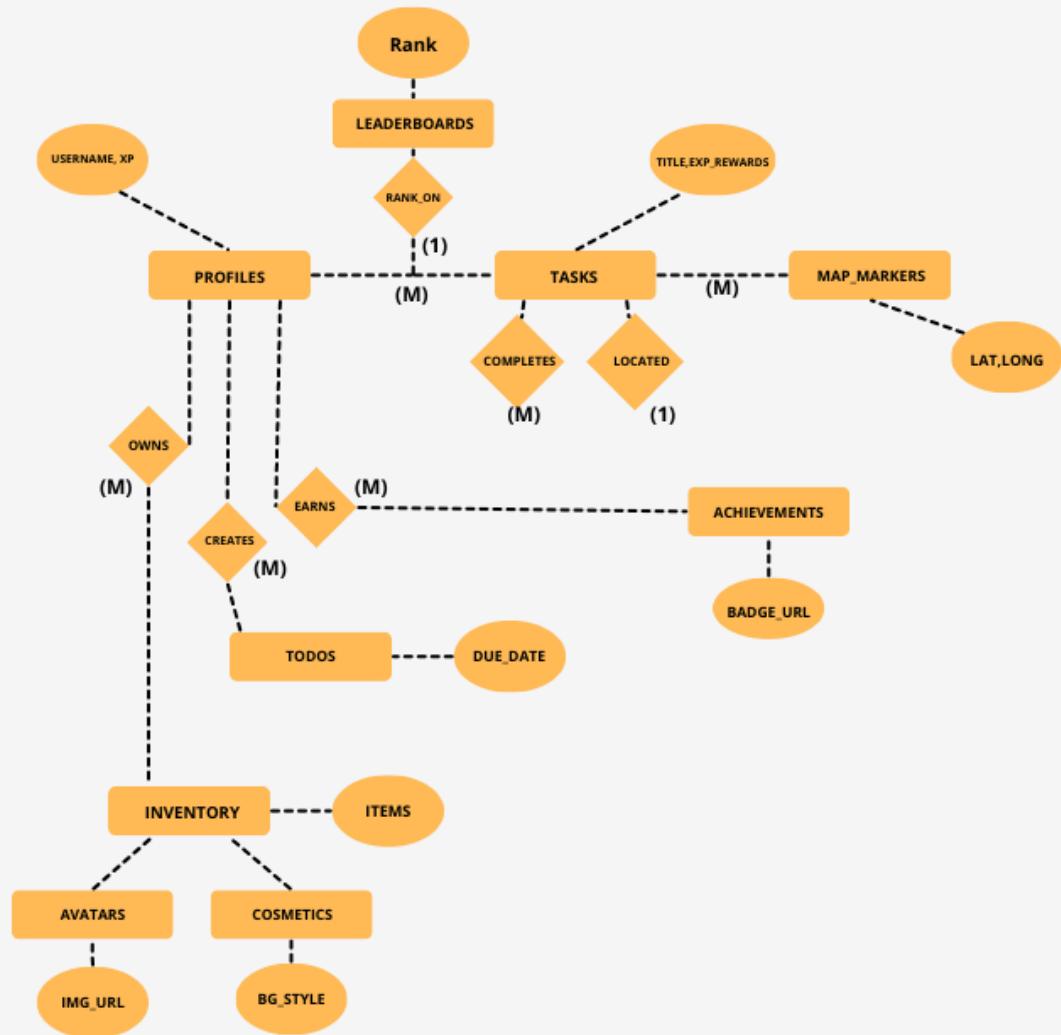
User Roles – Categories of users (e.g., students, faculty, staff, system administrators) defined by different permissions and capabilities.

WCAG (Web Content Accessibility Guidelines) – A set of international standards ensuring that digital content is accessible to users with disabilities.

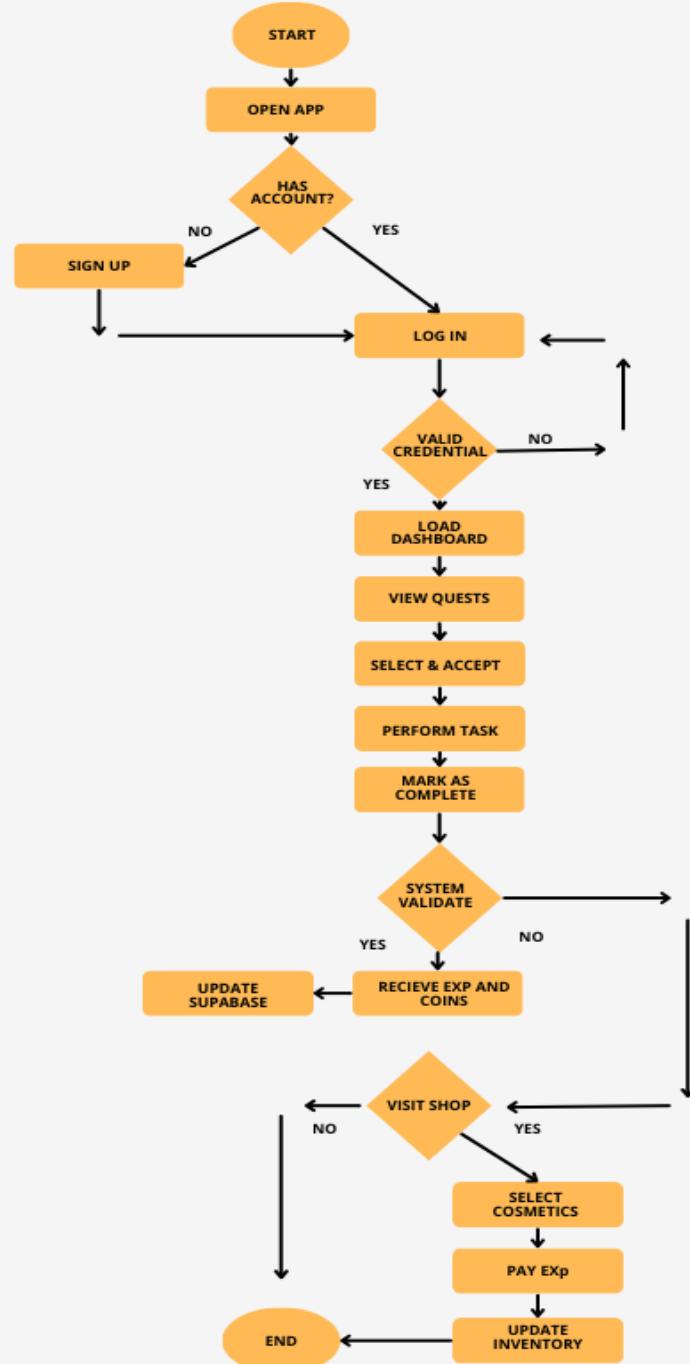
Appendix B:
Analysis Model:
Use Case Diagram



ERD



Activity and Process Model



References:

- ISO/IEC/IEEE. (2018). ISO/IEC/IEEE 29148:2018 — Systems and software engineering: Life cycle processes — Requirements engineering. ISO/IEC/IEEE.*
- Sommerville, I. (2016). Software engineering (10th ed.). Pearson Education Limited.*
- Pressman, R. S., & Maxim, B. R. (2020). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education.*
- Republic Act No. 10173. (2012). Data Privacy Act of 2012. Official Gazette of the Republic of the Philippines.*
- National Privacy Commission. (2017). Data protection guidelines for schools and universities. National Privacy Commission.*
- World Wide Web Consortium. (2018). Web content accessibility guidelines (WCAG) 2.1. W3C.*
- ISO/IEC. (2011). ISO/IEC 25010:2011 — Systems and software engineering: Systems and software quality requirements and evaluation (SQuaRE). ISO/IEC.*