Software Requirements Specification for QuestNest

Lab 2 Section 3 Version 1

Prepared by Natalie Avila, Ejara Dinku,

Kanaan Graham, Caitlyn Heath,

Junsen Huang, Colin McGowan, Uzziel Vea-Linares

Old Dominion University

October 31, 2025

Table of Contents

3 Specific Requirements

This section defines the software requirements for the QuestNest application. It outlines the external interfaces, system features, performance requirements, design constraints, and software system attributes. Each requirement is organized by feature to provide a clear view of the system's functionality.

3.1 System Features

This subsection describes the key features that make up QuestNest. Each feature includes its purpose, expected stimulus and response, and functional requirements for operation.

3.1.1 Account Management

The Account Management feature handles the creation, authentication, and maintenance of user profiles within QuestNest.

3.1.1.1 Introduction/Purpose of Feature

This feature enables caregivers and children to create and manage user profiles within QuestNest by providing an email, username, and password. It provides a simple and secure sign-in process using avatars and a 4-digit personal identification number (PIN).

3.1.1.2 Stimulus/Response

Stimulus A: Caregivers initiate the process of creating a new account by submitting the required information: username, email, and password.

Response A: The system registers the account in the database, stores credentials securely using hashing, and confirms successful creation.

Stimulus B: Caregivers create a profile for themselves or a child, selecting an avatar and creating a 4-digit PIN.

Response B: The system links the child profile to the caregiver's family account, initializes XP to zero, and securely stores the avatar and hashed PIN.

3.1.1.3 Associated Functional Requirements

The following functional requirements specify how the system manages account creation and profile setup.

3.1.1.3.1 Account Creation *(O: Graham)*

The system shall allow new users (caregivers) to register with a username, email, and password.

3.1.1.3.2 Input Validation *(O: Graham, O: Dinku)*

The system shall validate that all required fields (username, email, password, PIN) are provided and meet defined format requirements during account and profile creation.

3.1.1.3.3 Email Validation *(O: Graham)*

The system shall validate that each email address is unique before account creation.

3.1.1.3.4 Descriptive Error Messages *(O: Graham)*

The system shall display descriptive error messages when credential entries are invalid.

3.1.1.3.5 Password Storage *(O: Graham)*

The system shall store passwords as hashed values and not in plain text.

3.1.1.3.6 Avatar Selection *(O: Dinku)*

The system shall provide a gallery of predefined avatars and allow each profile to select exactly one avatar image during setup.

3.1.1.3.7 PIN Creation (*O: Dinku*)

The system shall require a 4-digit numeric PIN during profile creation for authentication.

3.1.1.3.8 PIN Storage *(O: Dinku)*

The system shall store all PINs as hashed values and not in plain text.

3.1.1.3.9 XP Initialization *(O: Graham)*

The system shall automatically assign a default XP value of 0 to new accounts.

3.1.1.3.10 Linked Child Accounts *(O: Graham)*

The system shall allow caregivers to create and manage multiple child profiles under their family account.

3.1.2 User Login

The User Login feature authenticates users to ensure secure access to their personalized QuestNest dashboards.

3.1.2.1 Introduction/Purpose of Feature

This feature enables caregivers and children to access their personalized QuestNest dashboards. It verifies user identity through credentials (email, password, and 4-digit PIN).

3.1.2.2 Stimulus/Response

Stimulus A: Caregivers enter valid credentials (email and password) on the login page.

Response A: The system validates credentials against stored records in the database.

Stimulus B: Children select their avatar profile and enter their 4-digit PIN.

Response B: The system verifies the entered PIN against stored records in the database.

3.1.2.3 Associated Functional Requirements

The following functional requirements specify how the system manages login authentication.

3.1.2.3.1 Email and Password Authentication *(O: Graham)*

The system shall allow caregivers to log in using an email and password.

3.1.2.3.2 PIN Authentication *(O: Dinku)*

The system shall verify entered PINs against stored hash values when logging in.

3.1.2.3.3 Credential Verification *(O: Graham)*

The system shall validate login credentials against database records.

3.1.3 Chore Management

The Chore Management feature enables caregivers to create, assign, and monitor household chores for each family member. Children can view their assigned chores, mark them as complete, and submit proof for caregiver review.

3.1.3.1 Introduction/Purpose of Feature

This feature ensures that tasks are clearly organized, responsibilities are distributed fairly, and progress can be easily tracked. The purpose of this feature is to provide an intuitive and interactive system for families to manage daily tasks, reinforce accountability, and streamline communication between caregivers and children.

3.1.3.2 Stimulus/Response

Stimulus A: A caregiver creates a new chore and assigns it to one or more children, setting its title, description, type, due date, and XP value.

Response A: The system stores the chore in the database, associates it with the assigned child's account, and displays it on both the caregiver and child dashboards.

Stimulus B: A child marks a chore as complete and submits proof (photo or video).

Response B: The system updates the chore's status to "Pending" and notifies the caregiver for validation. Upon caregiver approval, the system awards XP and updates the progress bar.

3.1.3.3 Associated Functional Requirements

The following functional requirements specify how the system manages chore creation, tracking, and storing data

3.1.3.3.1 Create Chore (*O: Huang*)

The system shall allow caregivers to create new chores by specifying a title, description, type, due date, and XP value.

3.1.3.3.2 Assign Chore (*O: Huang*)

The system shall allow caregivers to assign a created chore to one or more family members.

3.1.3.3.3 Edit/Delete Chore *(O: Huang)*

The system shall allow caregivers to edit chore details or delete a chore prior to completion.

3.1.3.3.4 View Assigned Chores (*O: Huang*)

The system shall allow each user to view their assigned chores on their dashboard and within the shared calendar.

3.1.3.3.5 Chore Status Tracking (*O: Huang*)

The system shall display the status of each chore, including “View”, “Pending”, “Accepted”, or “Re-do”.

3.1.3.3.6 Submission of Proof (*O: Huang*)

The system shall allow children to upload photo or video proof when marking a chore as complete.

3.1.3.3.7 Automatic XP Allocation (*O: Huang*)

Upon caregiver approval, the system shall automatically award the designated XP value to the child.

3.1.4 Experience Points (XP) and Progress Tracking

Experience Points (XP) are used to track a user's progress and as currency to redeem rewards. XP is awarded once a completed chore is verified to motivate users to complete their assigned chores. Progress and assigned chores will be more accessible for caregivers to track.

3.1.4.1 Introduction/Purpose of Feature

Experience Points (XP) are a numeric point system that children will earn after a complete chore is verified. A child's progress is determined and tracked by the total amount of XP earned, amount of XP needed to level up, and current level. As the child levels up, they will have access to new rewards. Each reward will cost a certain amount of XP, which the child must accumulate to redeem. The purpose of this feature is to motivate children to complete assigned chores. Caregivers can view their child's progress such as their accumulated XP and the status of their assigned chores. This will allow for a stress-free and more manageable way to track their child's progress and responsibilities.

3.1.4.2 Stimulus/Response

Stimulus: Children view their XP on the dashboard.

Response: The system retrieves the child's accumulated XP and XP goal from the database and displays it as a progress bar.

3.1.4.3 Associated Functional Requirements

The following functional requirements specify how the system manages Experience Points (XP), user progression, and dashboard interaction.

3.1.4.3.1 Parent Progress View (*O: Avila*)

The system shall provide a parent dashboard that allows the caregiver to view the status of their child's assigned chores (e.g., Accepted, Re-Do, or View).

3.1.4.3.2 System Dashboard (*O: Avila*)

The system shall provide a dashboard that displays chores assigned to the user.

3.1.4.3.3 Experience Points (XP) Progress Bar (*O: Avila*)

The system dashboard shall display a progress bar containing the current accumulated XP and the goal to level up.

3.1.4.3.4 Collaborative Progress Bar (*O: Avila*)

The system dashboard shall display a progress bar containing the current accumulated XP and the goal for the collaborative family reward.

3.1.4.3.5 Navigation *(O: Avila)*

The system shall provide a home button to provide users with access to the dashboard graphical user interface.

3.1.4.3.6 Set Chore XP Amount *(O: Avila)*

The system shall allow caregivers to create chores and designate the amount of XP earned after completion.

3.1.5 Tiered Leveling System

The Tiered Leveling System provides structured progression that rewards users as they consistently complete chores. It promotes continued participation by linking household task completion with advancement and new reward opportunities.

3.1.5.1 Introduction/Purpose of Feature

This feature defines a level-based structure that tracks user progress through accumulated Experience Points (XP). Its purpose is to motivate and engage by rewarding consistent task completion with advancement and access to higher-value rewards.

3.1.5.2 Stimulus/Response

Stimulus: A user completes a verified chore.

Response: The system awards associated XP, updates the user's total, and advances their level if a threshold is reached. Upon leveling up, the user unlocks new rewards.

3.1.5.3 Associated Functional Requirements

The following functional requirements specify how the system manages Experience Points (XP), user progression, and reward unlocking within the tiered leveling system.

3.1.5.3.1 Awarding Experience Points (*O: Heath*)

The system shall assign XP to users upon verification of completed chores.

3.1.5.3.2 Accumulate XP (*O: Heath*)

The system shall maintain a cumulative record of each user's XP total.

3.1.5.3.3 Evaluate Level Thresholds (*O: Heath*)

The system shall measure when a level threshold is reached based on XP totals.

3.1.5.3.4 Update Available Rewards (*O: Heath*)

The system shall update available rewards according to the user's current level.

3.1.6 Reward Management

3.1.6.1 Introduction/Purpose of Feature

The Reward Management feature is a system that includes the Rewards Store and the mechanisms to allow reward creation and exchange. The purpose of this feature is to facilitate the incentives for children to complete their assigned tasks.

3.1.6.2 Stimulus/Response

Stimulus: The parent creates a reward and designates an XP amount to be exchanged for it.

Response: The child can view the reward, regardless of their current level. A reward that costs N XP will be able to be exchanged for N amount of the child's XP, if the child has an XP amount greater than or equal to N. If less than N, the reward will not be able to be redeemed.

3.1.6.3 Associated Functional Requirements

3.1.6.3.1 Navigation *(O: McGowan)*

The system shall provide a Rewards Store button to provide children with access to the Rewards Store graphical user interface.

3.1.6.3.2 Graphical User Interface (GUI) *(O: McGowan)*

The system shall:

- Display all active rewards inside the Rewards Store.
- Not display all redeemed or deleted rewards from inside of the Rewards Store.
- Provide a button for each reward to indicate the reward can still be redeemed.

3.1.6.3.3 Reward Creation Mechanisms *(O: McGowan)*

The system shall allow caretakers to create rewards and designate an XP cost, image, and name.

3.1.6.3.4 Reward Redemption Mechanisms *(O: McGowan)*

The system shall:

- Allow children to redeem rewards for N XP, if their total XP is greater than or equal to N amount needed.
- Subtract N XP from the child's current XP total, if a reward is redeemed.

3.1.7 Collaborative Family Rewards

The Collaborative Family Rewards feature encourages teamwork and cooperation by linking family-wide incentives to the completion of shared household tasks. It allows caregivers to define rewards that benefit all family members, such as a family outing.

3.1.7.1 Introduction/Purpose of Feature

This feature enables caregivers to define rewards that are unlocked only when every family member completes their assigned family tasks. Family tasks are defined within the Chore Management feature and are linked to collaborative rewards. The purpose of this feature is to foster cooperation and share accountability across the household.

3.1.7.2 Stimulus/Response

Stimulus: The caregiver creates a collaborative reward, and labels associated chores as family tasks. When a family task is completed, the earned XP is added to a shared progress pool.
Response: Once all members have contributed their portions and the reward progress bar is filled, the system marks the collaborative reward as unlocked and the caregiver then fulfills the reward.

3.1.7.3 Associated Functional Requirements

The following functional requirements specify how the system manages creation, tracking, and completion of collaborative family rewards.

3.1.7.3.1 Create Collaborative Rewards (*O: Heath*)

The system shall allow caregivers to create rewards that require participation from all family members.

3.1.7.3.2 XP Contribution to Shared Progress (*O: Heath*)

The system shall allocate XP from family tasks to a shared reward pool.

3.1.7.3.3 Update Progress (*O: Heath*)

The system shall update the collaborative reward's progress as XP is added to the shared pool.

3.1.7.3.4 Unlocking/Claiming Collaborative Rewards (*O: Heath*)

The system shall unlock the collaborative reward once all family members have completed their required contributions.

3.1.8 Validation and Verification

3.1.8.1 Introduction/Purpose of Feature

The Validation and Verification feature allows caregivers to confirm whether a submitted chore has been completed satisfactorily before awarding XP. When a child marks a task as complete, the system prompts them to submit proof of a photo or video. The photo or video gets sent to the caregiver for review. The caregiver can approve or deny the chore so the child can get awarded the points or is sent a message as to why the chore was not complete, respectively. The purpose of this feature is to ensure accuracy, accountability, and consistency in the completion of chores before granting the XP.

3.1.8.2 Stimulus/Response

Stimulus: The child marks a chore as completed and submits a photo or video for caregiver review.

Response: The caregiver gets a notification and reviews the proof. If the caregiver approves the submission, the system awards the XP to the child and updates the task status to "Completed." If the caregiver denies the submission, the system notifies the child that the chore must be redone, and no XP is awarded until a resubmission, and the caregiver can approve.

3.1.8.3 Associated Functional Requirements

The following functional requirement outlines validation/verification logic for completing a chore.

3.1.8.3.1 Submit Completion Proof *(O: Vea-Linares)*

The system shall allow children to submit proof of chore completion in the form of photos or videos.

3.1.9 Shared Calendar

3.1.9.1 Introduction/Purpose of Feature

The Shared Calendar feature allows children to view their upcoming tasks in a centralized area of the application, so they can visualize how they need to create or adjust their schedule. The purpose of this feature is to promote self-accountability for children.

3.1.9.2 Stimulus/Response

Stimulus: The caretaker creates a task for the child. This task will be added to the Shared Calendar for the child to view.

Response: The task is posted to the calendar and will remain visible until completion. Upon completion, it will disappear from the calendar.

3.1.9.3 Associated Functional Requirements

3.1.9.3.1 Navigation *(O: McGowan)*

The system shall provide a Calendar button to provide children with access to the Calendar graphical user interface.

3.1.9.3.2 Graphical User Interface (GUI) *(O: McGowan)*

The system shall provide a graphical user interface which:

- Displays the current month’s schedule.
- Displays uncompleted tasks, which are to be located on the day of the calendar the caretaker has assigned to be the due date.
- Does not display any tasks completed by the child.

3.2 Performance Requirements

The following requirements define how the system manages performance.

3.2.1 Speed of Response *(O: Vea-Linares)*

3.2.1.1 Homepage Loading Time (*O: Vea-Linares*)

The average response time for loading the caregiver and child dashboards shall not exceed 2 seconds to ensure smooth navigation and main user engagement.

3.2.1.2 Account Creation Processing (*O: Vea-Linares*)

The account creation process shall be completed within 5 seconds from the caregiver's submission of registration to receiving confirmation.

3.2.1.3 Task and Reward Page Loading (*O: Vea-Linares*)

Task lists, reward menus, and progress pages shall load within 3 seconds to maintain engagement and have a positive user experience.

3.2.2 Execution Time *(O: Vea-Linares)*

3.2.2.1 User Interaction Response (*O: Vea-Linares*)

The system shall allow user interactions such as completing chores, claiming rewards, or checking progress bar; must respond within 100 milliseconds to ensure a smooth experience.

3.2.2.2 Database Query Execution (*O: Vea-Linares*)

Database queries for user accounts, chore data, and reward information shall be executed within 30 milliseconds to minimize latency, ensure smoothness, and support real-time updates across all connected family devices.

3.3 Design Constraints

This section identifies system constraints and limitations in design, implementation, and performance.

3.3.1 Platform Compatibility (*O: Heath*)

The software shall be a cross-platform mobile application, compatible with both Android and iOS devices.

3.3.2 Development Environment (*O: Heath*)

The system shall be implemented using Python 3.13 and shall follow the Model-View-Controller (MVC) architectural pattern to enforce separation of interface, control logic, and data management.

3.3.3 Network Requirements *(O: Heath)*

The system shall require an active internet connection to synchronize user data with the Google Firestore cloud-based database.

3.4 Software System Attributes

3.4.1 Data Collection *(O: Avila)*

Data usage policies must clearly state and require parent consent for the application to be used by users under the age of 18. Parents must have access to and delete children's data.

3.4.2 Data Protection *(O: Avila)*

To prevent data loss, an offline-first architecture must be implemented with sync queue and auto-backups every 60 seconds. Retry queues are used after failed syncs or offline connections.

3.4.3 Security *(O: Avila)*

Strong password requirements must be enforced with the option of Multi-Factor Authentication (MFA). The application must time out sessions for inactive users after five minutes. End-to-End encryption must be implemented for personal data in transit and at rest.

3.4.4 Storage Maintainability *(O: Avila)*

Media files must be compressed before uploading, have a set size limit, and auto deleted after reward approval (with user notice).

3.5 Other Requirements

The following requirements are miscellaneous.

3.5.1 Compliance *(O. Huang)*

Compliance with COPPA (Children's Online Privacy Protection Act) must be enforced, as the system collects data from users under 18 years of age.