Software Requirements Specification

Old Dominion University

CS 411W - Workforce Development

Prof: Thomas J. Kennedy

Due Date: 24 November 2025

Uzziel Isai Vea-Linares

1. Introduction

This section provides an overview of the QuestNest Software Requirements Specification (SRS). It outlines the purpose, scope, definitions, references, and organization of the Software Requirements Specification (SRS), establishing the foundation for the detailed requirements presented in later sections.

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the requirements for QuestNest, a mobile application. This document describes the expected functionality and behavior of the system from the user's perspective and provides a foundation for future development, validation, and maintenance activities. This document is intended for both the stakeholders and the developers of the system and will be proposed to Professor Kennedy for its approval.

1.2 Scope

This SRS outlines specifications for QuestNest. The system is designed for caregivers to assign chores, track their completion, and approve submissions, while children can earn XP and unlock rewards through a tiered leveling system. Families need a positive and structured system to balance responsibilities and build better habits in an engaging way. QuestNest has a goal to gamify chores and turn them into quests with experience points, levels, collaborative rewards, and caregiver validation. A functional prototype will demonstrate its engaging design, real-time data handling, and potential to improve family cooperation through positive reinforcement.

1.3 Definitions, Acronyms, and Abbreviations

Application Program Interface (API) – a set of protocols that allow different software applications to communicate and interact with each other.

Collaborative Family Reward - collaborative custom rewards awarded if all members of the family complete their assigned tasks.

Cross-Platform Application – software designed to operate on multiple types of devices (e.g., iOS and Android) from a single codebase.

Experience Points (XP) - points awarded as a progression indicator for task completion.

FastAPI – a modern web framework for Python that supports the creation of RESTful APIs.

Firestore – a cloud-based NoSQL database provided by Google, designed for real-time data synchronization across devices.

Flet – a Python framework built on Flutter that enables cross-platform development and user interface design.

Model-View-Controller (MVC) – a software architectural pattern that separates data management (Model), user interface (View), and application control logic (Controller).

RESTful API – an interface used to exchange information securely over the internet.

Tiered Leveling System – a progression structure where children advance through levels by earning experience points (XP), unlocking higher value rewards as they progress.

Uvicorn – a web server for Python, used to run FastAPI applications

1.4 References

IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 25 June 1998. Reaffirmed 9 Dec. 2009. IEEE Xplore, https://ieeexplore-ieee-org.proxy.lib.odu.edu/stamp/stamp.jsp?tp=&arnumber=720574&tag=1. Accessed 3 June 2025.

Lam, C. B., Greene, K. M., & McHale, S. M. (2016). Housework Time from Middle Childhood through Adolescence: Links to Parental Work Hours and Youth Adjustment. https://pmc.ncbi.nlm.nih.gov/articles/PMC5125879/

Rende, R. (2015). The developmental significance of chores: Then and now. https://www.researchgate.net/publication/269578645_The_developmental_significance_of_chores_Then_and_now

Society for the Psychology of Women. (2017). *Research on Household Chores*. household-chores. https://www.apadivisions.org/division-35/news-events/news/household-chores

Tepper, D. L., Howell, T. J., & Bennett, P. C. (2022). *Executive functions and Household chores: Does engagement in chores predict children's cognition?* https://pmc.ncbi.nlm.nih.gov/articles/PMC9796572/

1.5 Overview

The remainder of this document will describe the software to be produced, its expected features, inputs/outputs, and any required supporting information for its development. Section 2 will provide a higher-level overview of the software as a whole, user environment, and core capabilities. Section 3 will be organized into different expected features, as well as their expected implementations.

2 Overall Description

This section provides a high-level overview of QuestNest as a software application. It describes how the system works in its environment, summarizes its main functions, and describes the users who will interact with it.

2.1 Product Perspective

QuestNest is a standalone mobile application that is aimed at helping families track chores. It is developed as a cross platform solution for IOS and Android devices, using a single code base to maintain consistency between environments. QuestNest will help families build responsibility, improve communication, and reduce household stress.

From a user perspective, QuestNest will be developed as a cross-platform mobile application using Python 3.13.5, adopting the Model-View-Controller (MVC) architectural pattern to ensure an organized codebase. The app will use Google Firestore for business logic and database interactions, via the Python Admin SDK. The user interface will be developed with Flet, a Python framework built on Flutter. The controller will be FastAPI, which will connect all the components. Last but not least, Uvicorn will serve as the web server for QuestNest. All the following components were chosen with careful consideration to ensure the best user experience.

2.2 Product Functions

QuestNest transforms daily chores into interactive quests using a gamified system. QuestNest enables caregivers to create accounts, assign chores to children, and manage the verification process required for awarding XP. The system transforms regular boring chores into quests that children can complete to accumulate XP and unlock rewards. Once a chore is completed, children are prompted to upload photo or video evidence, which is sent to the caregiver for review. The caregiver can either approve or deny the submission which will give the XP points or have the ability to give feedback for a redo, respectively.

The one major differentiator between QuestNest and any other application is the collaborative rewards. The application is aimed at promoting consistency, accountability, long term participation which is achieved through collaboration between the whole family. All chores scheduling and remainders are integrated within a shared family calendar inside the application to provide visibility into upcoming tasks. The purpose of a shared family calendar, reminders and a collaborative reward system is to encourage teamwork to claim more enticing rewards. The app promotes lifelong habits of responsibility while reducing household stress and improving family collaboration.

2.3 User Characteristics

QuestNest is intended for users who want to track chores and teach responsibility. QuestNest is designed for family use, primarily by caregivers and child or teen participants. The caregivers are responsible for creating and assigning chores, approving or denying the submitted proof of completion, and managing the availability and cost of rewards.

Functionally, users are responsible for only general familiarity with the mobile application and no technical knowledge beyond standard navigation. The children and teenagers interact through a simplified task list and XP dashboard. They are responsible for marking chores as complete and uploading a video or photo for caregiver approval. The only two user roles are the caregiver who assigns the chores and the child or teen who is responsible for completing the chores.

2.4 Constraints

N/A.

2.5 Assumptions and Dependencies

N/A.