
Squad Goals

Software Architecture

2026/03/24

Project Description

SquadGoals is a gamified social media app that allows you and your friends to set goals and track your progress together by sharing photos and updates. It lets your friends and family help you stay on track so you can crush your goals together.

Project Deliverables

An MVP will require the following functionalities:

1. Secure account creation allows users to register and sign in.
2. Social connectivity features that enable users to search for and add friends and organise them into collaborative groups.
3. Goal-setting functionality where users define habits shared with a group, defaulting to a daily frequency for the MVP.
4. An evidence-based logging system requiring a photo upload to cloud storage as proof of habit completion.
5. An automated verification step utilising an LLM to perform a preliminary check of the uploaded "proof of completion."
6. A social validation mechanism allowing group members to approve or reject the evidence shared by their friends to finalize completion status.
7. A simple notification system to send notification reminders to complete a habit, and to notify users about progress updates from friends.
8. A visualisation dashboard featuring a streak counter and progress view to show how many days in a row the group has completed their goals.

Quality Attributes

Below are the main quality attributes the application will focus on. In addition to these, implementing strong privacy and security measures is crucial since users will be uploading personal data, including photos.

1. **Extensibility:** The architecture must be modular to allow for future feature expansion without disrupting the core habit-tracking engine, keeping up with the latest trends and user demands. The components should be independent of each other and implemented in isolation; evaluation will involve demonstrating how a new feature (such as a "Rewards Service") can be implemented into the existing API without requiring a rewrite of the core authentication or storage logic.

2. Scalability: The application must be designed to handle variable workloads, specifically handling "bursty" traffic during peak habit-logging hours. This should not be measured simply by the presence of a scaling service, but by the system's efficiency in provisioning and de-provisioning resources.
3. Availability: Users should be able to view and interact with the application at all times. The system will be provided with basic tests to monitor and ensure it performs under different environments and load/stress levels. Evaluation will be based on the system's ability to maintain a responsive state and display consistent data during these stress tests.