Software Requirements Specification

for

Maths Club SAT

Version 1.0.2 approved

Prepared by Garv Shah

Software Development 3/4

31/5/2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Link Update	15/6/22	Outdated Links in References	1.0.1
Use Case Explanation	16/6/22	Use Case Diagram Could Be Clarified	1.0.2

1. Introduction

1.1 Purpose

The purpose of this SRS is to create an app for the CGS Maths Club that will be able to make it easier to access questions, easier to create questions, and more fun to answer them. These three main objectives are what each feature should connect to, allowing for better community interaction and making the process of running and being part of the Maths Club more streamlined.

1.2 Intended Audience

The intended audience for this solution would be members of the CGS Maths Club, that allows for an archive of posts and problems which students can comment on, collaborate on, and discuss with each other. Namely, the app has two target clients, the general users who submit and view quizzes/posts, and the content creators/administrators who manage users and content on the app. The experience for both these types of users should be streamlined.

1.3 Product Scope

1.3.1 Will Do

- Search Engine
  - Search for certain sections or individual quizzes/posts
  - Be able to open and enter different parts of the app from the search engine
  - Should be relatively fast with increasing size
- Quiz System (Answer Online)
  - Will have posts that can remain as quizzes for a certain period of time and then get archived to be a post
  - While it is an active quiz, members will be able to do the quiz and get points for questions answered
  - Solutions will only be released after the quiz period
  - Members should be able to answer questions within the quiz interface
  - Should be able to differentiate between different math being equal or the same
- Divided Section for Junior/Senior etc
  - The GUI should be able to provide easy separation for sections such as Junior and Senior, or different filters
  - There will be a "group" system, where you can create different groups and assign users to them, which gives them certain access to different parts of the app. These may include "admin", "junior" or "senior"
- Leaderboards
  - There should be a leaderboards page, where people can view their points from each of the quizzes
  - People should be able to click "view my position" and automatically scroll to where they are
  - Badge System
    - Members should be able to get badges that will be manually assigned
- Optional Hints
  - Question curators should be able to optionally provide hints for questions that users can look at without actually looking at the solution
- Forums/Ability to Comment on Posts
  - Members should be able to comment on posts, which should facilitate community interaction
  - Members who are not approved (i.e. people not part of the CGS community) should not be able to post messages
    - Dependant on "group" system

1.3.2 Won't Do

- Be able to search for other users
- Be able to message users
- Be able to get notifications for posts (getting into extra territory, probably not necessary at all for a working system)
- Be able to scan paper answers and have them be processed by the app
- Basically, anything that isn't listed in the scope above

1.4 References

This document refers to https://github.com/garv-shah/software-dev extensively, as this address is where a variety of work on the project exists, including interviews, ideas, and documentation. https://garv-shah.vercel.app is also where project updates live, under the "Software Development 3&4" tag.

2. Overall Description

2.1 Constraints

- **Time:** has to have the design finished by the end of Term 2; product must be done by the end of Term 3
- **Cost:** I won't be able to spend money on the app, so there should be no fees with the amount of expected users
- **Hardware:** Ideally, I won't be running my own server for this, my server's can be unreliable at best and would significantly decrease the usability of the project especially once I'm not in school anymore, everything should run on other **free** servers
- **Software:** everything must be created (and possible) in Flutter, for at least MacOS, iOS, Android and the Web. Windows and Linux are not priorities, though it would be nice if they were supported.
- **Social:** the app has to be can't be wayyy to big and complicated, because then it would be overwhelming for the average maths club member. The design and UX has to stay simple, and will not be able to go overboard
- **Legal:** privacy laws must be respected, namely in a school environment when this app is available to the public, so it may be an issue if we are endorsing this platform as a school and anyone can contact school students
- **Ability:** the app is constrained by my ability to program, so I will not be able to do anything *crazy*, especially stuff that I have no clue how to make right now

2.2 Product Functions

2.2.1 Functional Requirements

- Search Engine
  - Search for certain sections or individual quizzes/posts
  - Be able to open and enter different parts of the app from the search engine
- Quiz System (Answer Online)
  - Will have posts that can remain as quizzes for a certain period of time and then get archived to be a post
  - While it is an active quiz, members will be able to do the quiz and get points for questions answered
  - Solutions will only be released after the quiz period
  - Members should be able to answer questions within the quiz interface
  - Should be able to differentiate between different math being equal or the same
  - The quizzes or posts should not have a file size limit, we should be able to create as many posts as we would like without limitations
- Divided Section for Junior/Senior etc
  - There will be a "group" system, where you can create different groups and assign users to them, which gives them certain access to different parts of the app. These may include "admin", "junior" or "senior"
- Leaderboards
  - There should be a leaderboards page, where people can view their points from each of the quizzes
  - People should be able to click "view my position" and automatically scroll to where they are
  - Badge System
    - Members should be able to get badges that will be manually assigned
- Optional Hints
  - Question curators should be able to optionally provide hints for questions that users can look at without actually looking at the solution
- Forums/Ability to Comment on Posts
  - Members should be able to comment on posts, which should facilitate community interaction
  - Members who are not approved (i.e. people not part of the CGS community) should not be able to post messages
    - Dependant on "group" system

2.2.2 Non-Functional Requirements

- Search Engine
  - Should be relatively fast with increasing size
  - Should make it easier to navigate the app and get to where you want to
- Quiz System (Answer Online)
  - The quiz system should feel fair, and it should promote a healthy competition
  - It should be easy to use and fast
- Divided Section for Junior/Senior etc
  - The GUI should be able to provide easy separation for sections such as Junior and Senior, which should feel intuitive
  - This should provide easier navigation of the app
- Leaderboards
  - The leaderboards must include a sense of competition that encourages people to partake in Maths Club more often
- Forums/Ability to Comment on Posts
  - The forums should protect and consider the privacy of school students

2.3 Stack

For my app, I intend to use the F^3 stack: Firebase, Flutter and Fastlane.

Along with this, I plan to use Notion as a CMS for my content, combined with Firestore. This combination should give me the scalability I need (with unlimited file storage and a manageable content backend) combined with the security and flexibility of Firebase.

Flutter will be my frontend, a cross-platform framework that I can use to develop iOS, MacOS, Web and Android apps, along with many more. Also being made by Google, the platform has a pretty good integration with Firebase, and as such, has a very good interop with the BaaS.

Firebase will be my backend, along with my auth and analytics. Firebase is basically a Swiss Army knife, so most of what I need from my backend should be able to be done by Firebase, with the exception of file storage, since this would get very expensive very fast with the service. Otherwise, I'll use it for live updates, an authentication workflow and profile pictures.

Finally, Fastlane is an app deployment platform which will allow me to implement CI/CD into my project, which is ideal for my scrum project plan, with continuous iterations based on client feedback. Fastlane should automate a few mundane jobs and make things a lot faster so my clients can get updates faster, speeding up everyone's workflow.

In summary, my stack will be F^3 + Notion

2.4 Operating Environment

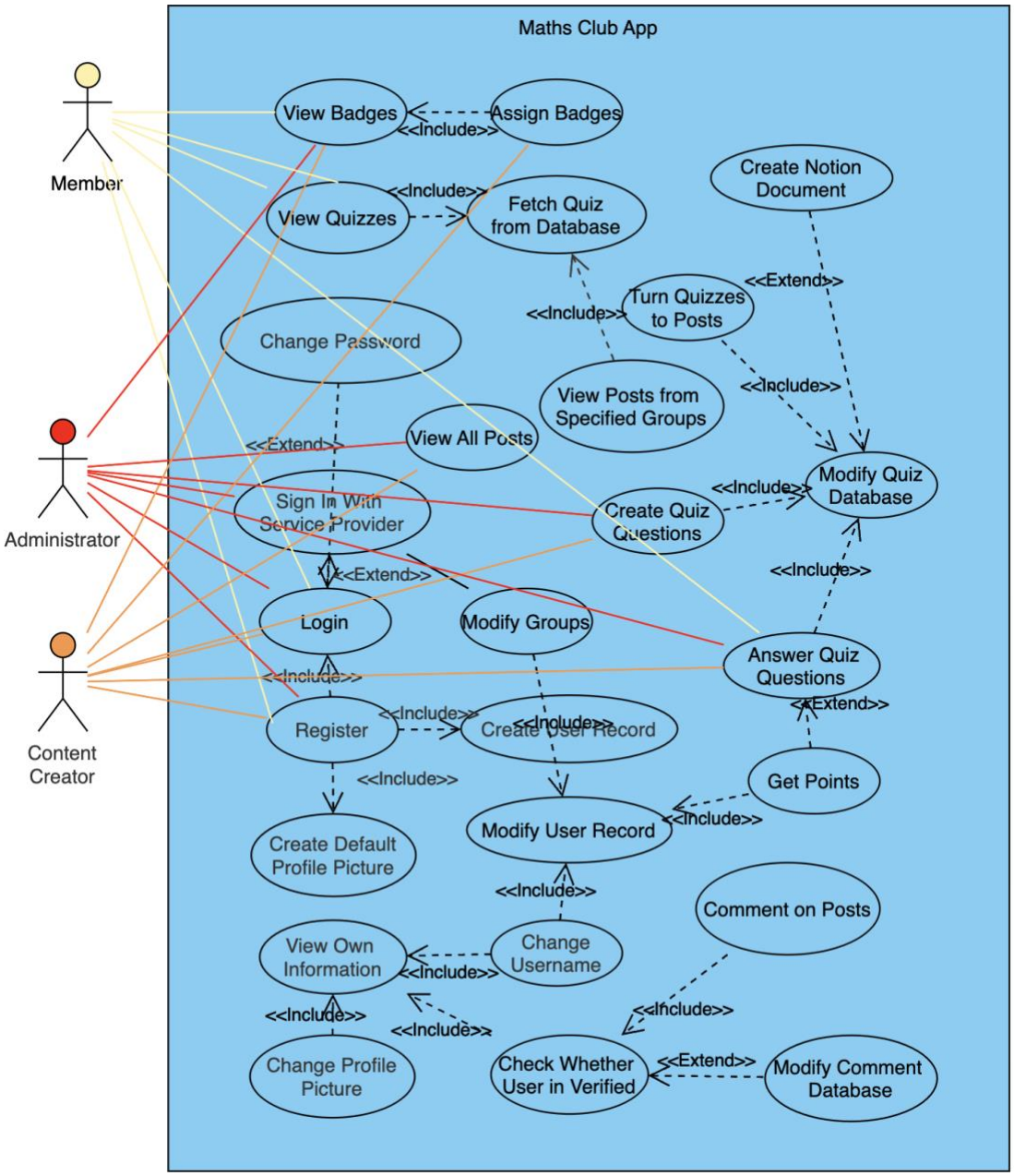
The solution will be implemented in the following environments:

- Website
- MacOS
- iOS/iPadOS
- Android

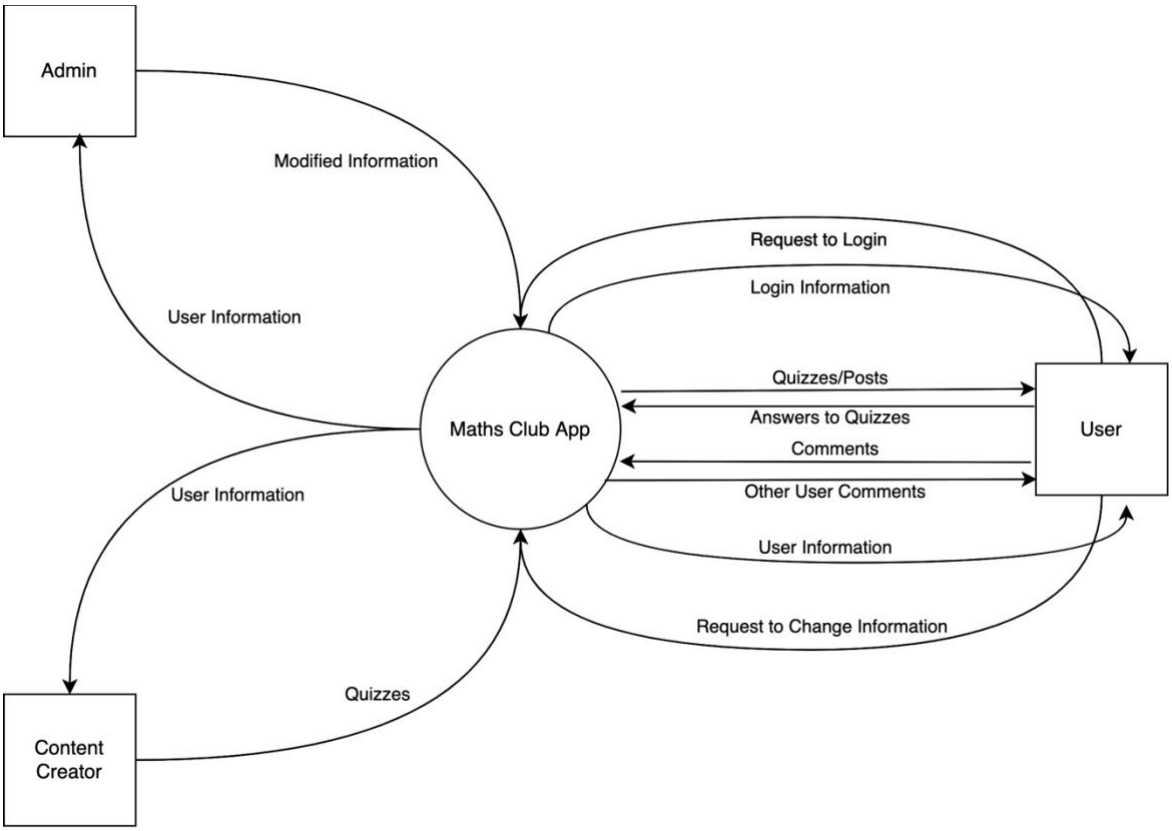
3. Analysis

3.1 Use Case Diagram

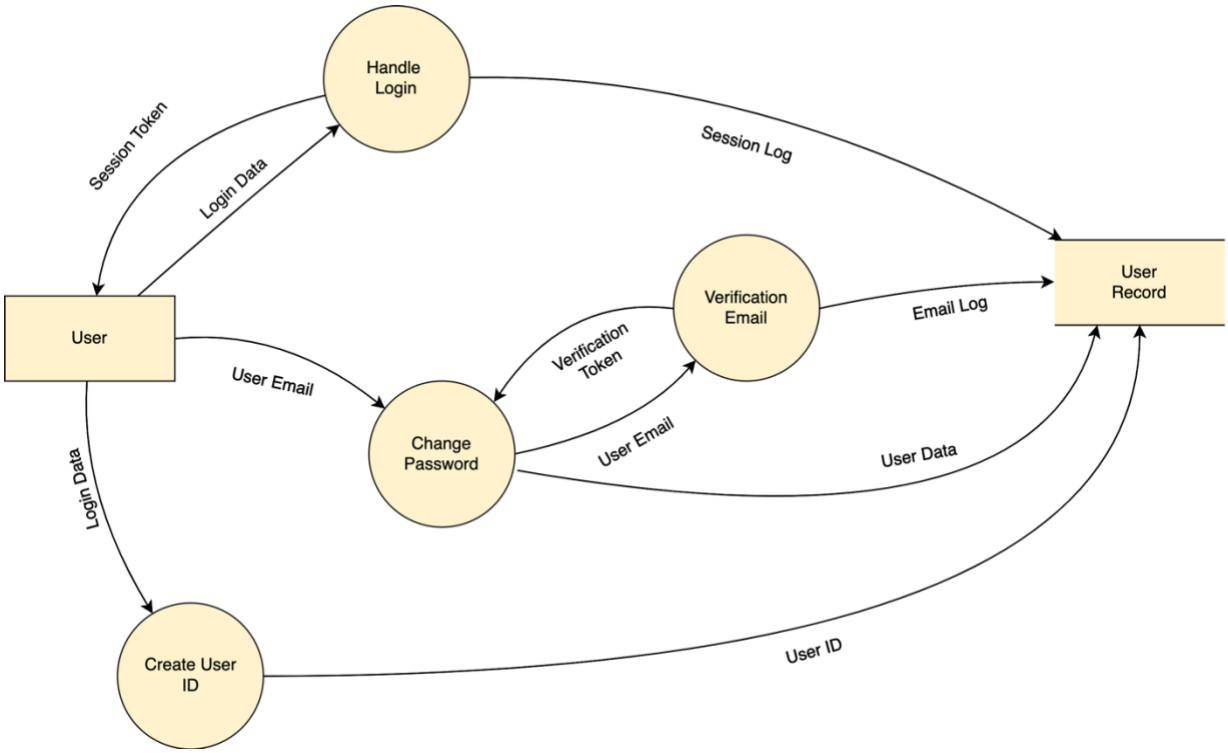
The Maths Club system is a relatively complicated system, combining various smaller subsystems into one larger application. As such, the Use Case Diagram below is quite complicated, as it includes the interactions of 3 different systems into one. The DFDs have proven to be more useful, as in essence, they provide zoomed in views of each of the systems, showing isolated data flow in comparison to use cases. Due to the nature of the app, it has been deemed that though the below Use Case Diagram is complicated and doesn't provide a view of individual functions, it provides a broad overarching view of the application and mostly serves as the basis for interactions between systems, that the more detailed views of DFDs do not provide. Each system is separate but unified, as the diagram shows.



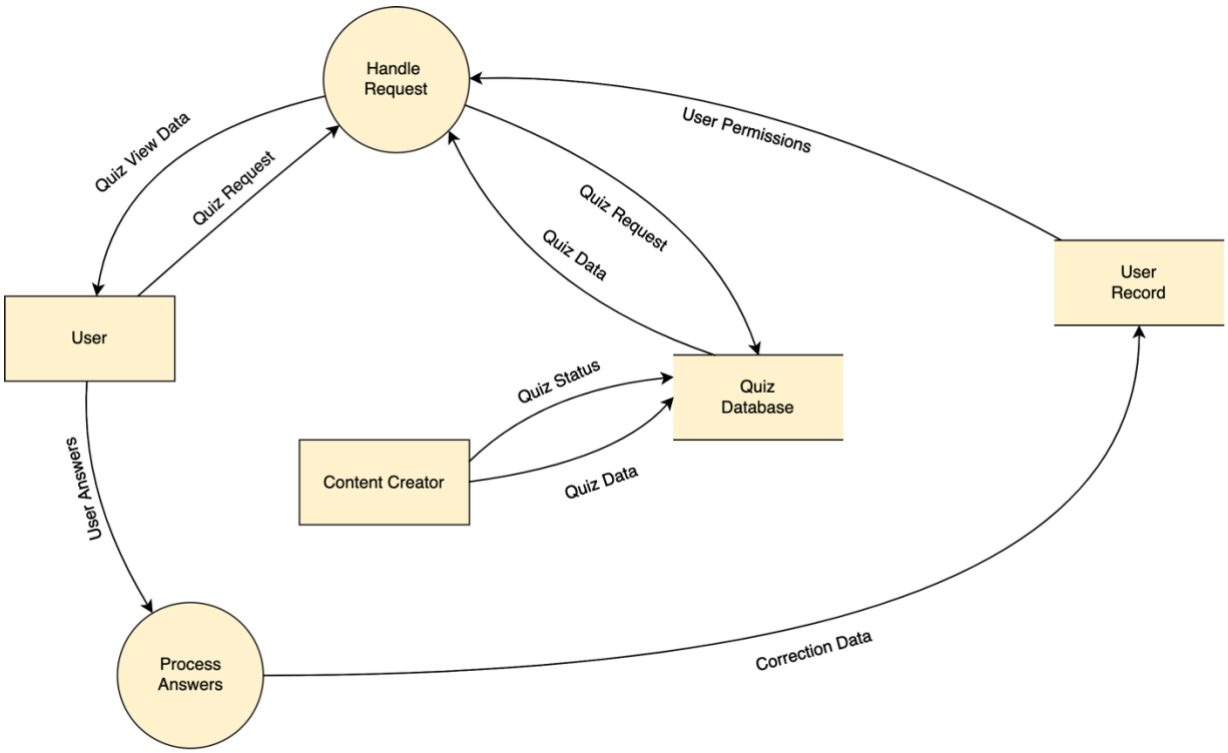
3.2 Context Diagram



3.3 Login System DFD



3.4 Quiz System DFD



3.5 Misc. Systems DFD