NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

SS22053 – Architecting a Backend System using Modern Methodologies and Frameworks

Presented by Leow Ken Hing Bryan

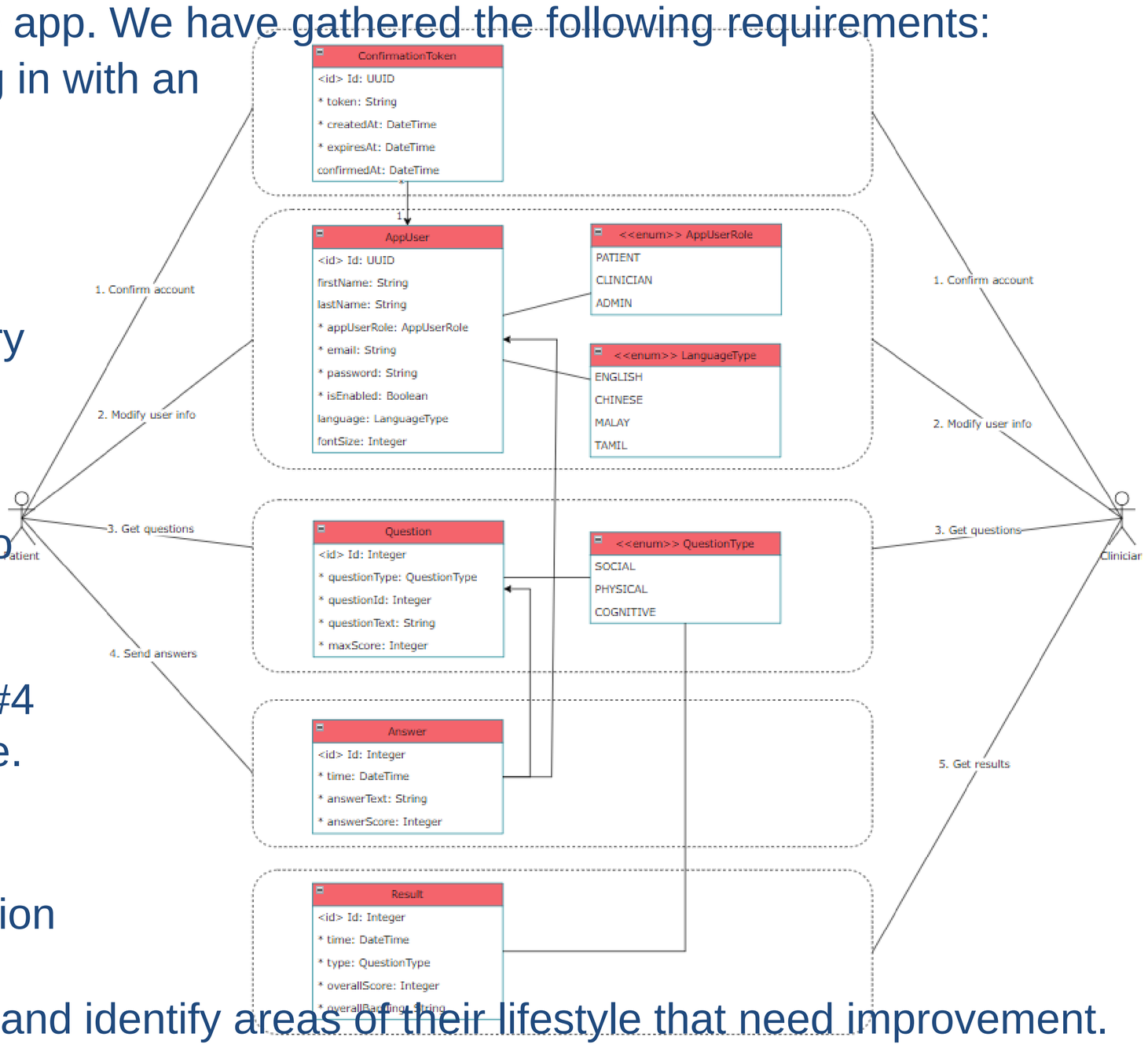
Supervised by Assoc Prof Ringo Ho, Co-supervisor Mr Pham Tan Phat

Motivation

- In this research project, we were tasked to develop a multidimensional frailty screening tool prototype.
  - Frailty has three dimensions: physical, cognitive, and social.
  - This prototype will help the elderly perform self-assessments and screening for frailty.
- The prototype consists of both a client mobile application (for elderly to see and do their assessments) and a server (for the app to calculate and save results).
  - The client application is written in Flutter and the server is written in Java Spring Boot.
  - In this research project, we will delve into the development of the Java Spring Boot server. This is also known as the backend of the mobile application.

Methodology

1. In developing any application, the first step is to gather business requirements for the user, i.e., what functionality the elderly need in the app. We have gathered the following requirements:
   1. Registering, confirming, and logging in with an account
   2. Modifying user information
   3. Retrieving questions
   4. Sending question answers
   5. Retrieving results and answer history
2. Next, we prioritized the most important features, which were #3 and #4.
3. In developing #3, we allow users to choose which questionnaire they want to do and return them the questions.
4. In return, they will choose their answers and send their responses back to us in #4 once an entire questionnaire is complete. Our data includes the following:
   1. Date of questionnaire completion
   2. Exact answers given for each question
   3. Calculated score attained by user



These allow the user to track their progress and identify areas of their lifestyle that need improvement.

Findings

We found difficulty in storing information about the questionnaire. Each question had their own set of options and scoring methods which initially made it hard to capture and program for. Through research on design patterns, we found the “strategy design pattern” to be a good fit for this problem. By reducing the questions to a few scoring strategies (e.g., Yes/No, multiple choice, multi-part) and applying the appropriate strategy, we kept our code simple and general.

Future Work

Doing data collection and analytics on user behaviour, like time spent per question, etc.