SDPVocabQuiz: An app for sharing and practicing vocabulary quizzes (Deliverable 1–Preliminary Work)

Background

One of your professors is looking for a way to help students study vocabulary terms for their classes and challenge one another. Knowing that you and your team members are expert developers, she asked you to develop a program that students can use on a locally shared (Android) tablet to (1) create and share vocabulary quizzes and (2) compare their progress. As a first step, she would like your team to provide a possible design for the app, expressed in UML. Luckily, the members of your team already have some experience with this, so the team should be able to hit the ground running and produce a good design quickly.

Requirementshttps://docs.google.com/document/d/114w-W-szpFhtKF9HbFLOef-ztOUuqDhrFkgZqNwCLjE/edit#

See the requirements in Assignment 5.

Your professor has also supplied some additional clarification for some of the requirements, based on discussions with the students in your team and in other teams:

1. Requirement 7 (quiz score statistics):
   a. The quiz score statistics for a student S should list *all* quizzes, whether they were played by S or not, and including the quizzes created by S.
   b. The quizzes not played by S can be displayed in any order (after the ones played).
   c. For quizzes not played by S, only the names of the first three students to score 100% on the quiz should be displayed.
   d. The names displayed (and used to sort) in the statistics for the first three students to score 100% on the quiz can be either their usernames or their real names.
2. Requirement 6 (practicing a quiz):
   a. Every word in a quiz should be shown once and only once.
   b. Incorrect definitions, conversely, may repeat.
3. General clarification: all relevant data (scores, statistics, quizzes, student logins) should persist between uses of the application.

Instructions

1. **Before starting, make sure that you have completed and submitted Assignment 5.**
2. Create a directory called "`GroupProject`" in the **team repo** we assigned to you. Hereafter, we will refer to this directory as `<dir>`.
3. Create two directories, `Design-Individual` and `Design-Team`, under `<dir>`.
4. Each team member must copy both the UML design and the design information document he or she created for Assignment 5 in a directory called `<student's GT username>` under `<dir>/Design-Individual`.
5. Discuss and critique the different designs within the team.
   Important: Avoid being confrontational or defensive; keep in mind that **the goal is not to judge each other's work, but rather to come up with a good design for the system**.
6. Based on the results of the discussion, create a design on which the whole team agrees. The team design can be one of the team members' designs, possibly updated, a combination of two or more of the team members' designs, or a completely new design based on what the team members learned while doing Assignment 5 and during the design discussion.
7. Save the team design in directory `<dir>/Design-Team` as a PDF file named `design-team.pdf`.
8. Create, also in directory `<dir>/Design-Team`, a document in MD format called `design-discussion.md` with the following content:
   a. One section for each of the individual designs, called "*Design 1*", "*Design 2*", and so on, that shows that design (as an embedded figure) and discusses its main pros and cons as they emerged during the team discussion.
   b. One section for the team design, called "*Team Design*", that shows the team design (as an embedded figure), discusses the main commonalities and differences between this design and the individual ones, and concisely justifies the main design decisions.
   c. One final section, called "*Summary*", that concisely summarizes the lessons learnt in the process of discussing the designs, in terms of design, team work, and any other aspect that the team members consider relevant.
9. Commit and push `<dir>` to your remote repository and submit the corresponding commit ID on Canvas. Only the current project manager (selected by the team) should submit the commit ID. **There is one submission for the entire team.** Please note that anyone in the team will be able to check the commit ID in Canvas after it has been submitted by the project manager.

Important:

- **Team members who have not submitted their individual design yet (i.e., late submitters) must not be involved in the discussion until they are done with their individual submissions.**

- We will **not** use this deliverable to grade Assignment 5; in fact, we will not even look at it before we return your Assignment 5 grades. In other words, please be candid in your discussion about the individual designs and in reporting such discussion.
- **Although you may want to get feedback on your individual designs before producing the team design, this would completely defeat the purpose of this deliverable, as our comments would clearly and heavily influence the discussion within the team.**
- The fact that the system will be implemented on the Android platform should not affect your design, which should not contain Android specific elements (e.g., activities).
- The project manager, who will submit the commit ID, is selected by the team. You may change project manager for subsequent deliverables and do not need to notify us. **The team as a whole is responsible for the single commit ID submitted.** The whole team can see the submission, and should ensure that a correct ID is submitted on time.