CS31620 Assignment 2021-22 Assignment: Native mobile quiz app

Chris Loftus
Release date: Tuesday, 19th October 2021
Submission date: 1pm Tuesday, 11th January 2022
Type of hand in: Blackboard Assignment
Weighting: 60%

Problem

Our local schools require a prototype quiz app to help their students evaluate their understanding and revise for exams. This prototype must:

- enable staff to build a bank of questions with answers for a specific module,
- enable students to take a quiz associated with a module, where a selection of questions are presented in a random order.

Design of the app

Question bank functional requirements

A teacher manages question banks as follows:

FR1[1]: A teacher can create a new empty question bank. Each question bank has a unique identifier. The format of the identifier is up to the teacher, that is, it can be any text such as a number or a word or even a phrase.

FR2: A teacher can add new questions to an existing question bank. For this prototype these are restricted to single choice questions where a student can select one answer from N where N is between 1 and 10.
For the single-choice question type, the teacher:

- enters the question as text,
- provides each question answer as text, up to N answers,
- indicates which of the answers is correct; there must only be one right answer.

FR3: A teacher can list and then remove questions from an existing bank.

FR4: A teacher can list all the question banks.

FR5: A teacher can delete a question bank.

Quiz functional requirements

FR6: Take a quiz. A student can select quiz mode. When they take the quiz, they can:

FR6a: List the question banks.

[1] FR stands for functional requirement.

FR6b: Take a specific question bank quiz.

A running question bank has the following requirements:

FR7: A score is displayed when the quiz ends. This is the number of correct answers out of the total number of questions in the bank.

FR8: The quiz displays all the questions in a random order and one question at a time.

FR9: The student decides when they want to move to the next question. Skipping a question without answering it is treated as an incorrect answer.

FR10: The student selects the appropriate answer from the list. The answer is stored.

Non-functional requirements

NFR1[2]: The question bank must be persistent and stored in a SQLite relational database on the Android device.

NFR2: You must use Kotlin to develop this app.

For this prototype there are no security requirements.

Construction of your app

Make sure you have a prototype UI design (see below). In addition, create a software design using appropriate UML diagrams. Clearly, you will iterate between implementation and design as you proceed. Module workshop recordings and worksheets covered all of the aspects of app construction that you will need for the implementation - overall app formatters such as tabbed screens, or hierarchical screens, lists of content, interaction with buttons, labels and text boxes, and storing and accessing data within an app.

You should aim to build a working app to match your UI and software designs.

Testing of the app

You should aim for automated unit testing and user interface testing of your app. How this can be set up was covered in workshop recordings and associated worksheets, but you will need to decide what is to be tested, as well as how it is tested. Although not as good, if you cannot achieve automated testing then make sure that you have a manual test table.

Submission of the assignment

By the deadline, you should submit the following via Blackboard:

- A copy of your app (hopefully completely functional, but you should hand it in even if it is not functional as not all the marks are for functionality).

[2] NFR stands for non-functional requirement.

- A non-functional prototype showing your design for your app. This might be paper based (electronically scanned), a set of linked screens in a presentation tool such as Powerpoint, or a set of screens in a prototyping tool such as Adobe XD, FluidUI or Figma. Other tools are possible.
- A report on the project containing the following sections:
    - **UI design section**: A discussion of your UI design. This should discuss the UI prototype and provide screenshots of the design in the report. How was Material Design used?
    - **Software design section**: A discussion of the software design. This should include appropriate UML diagrams and justification for the software structure.
    - **A testing section**: A plan for testing the app, and details on how testing went. Provide screenshots of the running app.
    - **Reflection section**: A reflection on how the assignment went - what went well, what might have gone better and why, what flair was attempted, what you have learned that will help you to do better in your final year project, and a summary of how well you think you did the project and what mark you think it deserves. Have a subsection for each.

Submission details

In submitting your work, you are confirming that you comply with the Department and University statement and policy on Unacceptable Academic Practice [1]. When you submit your report, you will be confirming that you understand the policies and confirm that your work meets those policies.

If you cannot access Blackboard, you need to email your report to Chris Loftus (cwl@aber.ac.uk) by the time of the submission on Blackboard. Your email should also explain the problem that prevented you from uploading to Blackboard. If you send the email after the deadline for submission, it will be treated as a late entry. Please see the Student Handbook [1] for details on how this would be processed.

Breakdown of marks for the project

Marks for the project will be assessed on the following basis:

Criterion	**Value**
Quality of the documented UI design (prototype and its description in the report)	12%
Quality of the documented software design	8%
Fulfilling the functional and non-functional requirements (includes whether it runs correctly and the quality of the code)	40%
Quality of the user interface in the final app	10%
Quality of testing (plan and execution)	10%
Quality and coverage of the report	10%
Flair: going above and beyond expectations, e.g. adding further functions. Ask me if you are not sure.	10%

References

[1] Computer Science Department (2020/21) “Student Handbook” (Online) https://impacs-inter.dcs.aber.ac.uk/en/cs-undergraduate/official-information/student-handbooks (Accessed 13th October 2021)