CMPS 312 Project Phase 2 – Data Management using Cloud Firestore and local SQLite Database (15% of the course grade).

The project phase 2 submission is due by 11 am Wend 11th November 2020. Demos will be organized during office hours on the same day.

1. Deliverables

You are complete the delivery of *LingoSnacks* app to aid language learning for both the teachers and learners by delivering the use cases documented in the project phase 1 requirements. The ***LingoSnacks App*** includes both the ***Learning Package Editor*** used by the teachers to create learning packages and learning activities used by the students for game-like learning activities. In this phase, you will extend an improved version of the app produced in phase 1. The base solution to be provided will have the **full** app. You are require to manage the app data using Cloud Firestore and the local SQLite Database. The app should work both **online and offline**.

Your project phase 2 deliverables include:

Part 1 - Cloud Firestore Database Design and Implementation

1. Design Cloud Firestore database to manage LingoSnacks App data.
2. Implement the repositories to read/write entities using Cloud Firestore as the data source. All **data filtering should be done on the server** using Cloud Firestore capabilities and only the required data should be retrieved.
3. Populate the database with the data from the the provided JSON files.
4. Document your database design in a schema diagram.

Important notes: When adding/updating a Learning Package you should auto-set the:

- LearningPackage.**packageId** should be auto-assigned and not entered by the user ownedBy to the currently logged-in user
- LearningPackage.**version** should be auto-incremented every time the package is updated.
- lastUpdatedDate should be auto-set to the current date and time

Part 2 - Local SQLite Database Implementation

1. Implement the needed entities to manage LingoSnacks data in a local SQLite database to allow offline usage of the app even without internet.

   The keep the database design simpler, the learning package Words (and the assiated definitions, sentences and resources) could stored in a String column as a single JSON document.

2. Implement the Data Access Objects (DAO) and repositories to read/write from SQLite using the Room library.

3. Populate the database with the data fetch from Cloud Firestore.
4. Document your database design in a schema diagram.
5. Sync Local DB with Cloud Firetore when the app starts. More clarificatins will be added

Part 3 – Signup and Login using Firebase Authentication

- Implement signup and authentication using Firebase Authentication.

Design and Testing Documentation

1. Document the MVVM architecture diagram for your overall design.
2. Document 4 technical lessons learned by comparing your submitted project phase 1 with the model solution provided. You need to provide detailed reflections about the new concepts and lessons learnt.
3. Write a testing document including screenshots of conducted tests illustrating a working implementation of both the Repositories and the Services.
4. Every team member should submit a description of their project contribution. Every team member should participate in the solution demo and answer questions during the demo.

Important notes:

- Continue posting your questions to https://piazza.com/qu.edu.qa/fall2020/cmps312/

- Do not forget to submit your design and testing document (in Word format) and fill the *Functionality* column of the grading sheet provided in the phase 2 Word template.

- Push your implementation and documentation to your group GitHub repository as you make progress.

- You need to test as you go!

- Seek further clarification about the requirements/deliverables during the initial progress meeting with the instructor. Note that further important clarifications maybe to this document and you will be notified.

2. Grading rubric

Criteria	%	Functionality*	Quality of the implementation
1) Cloud Firestore Database Design and Implementation Repositories to interact with Firestore	40		
2) Local SQLite Database and Implementation - Design and implementation of the database to manage LingoSnacks data in the SQLite database. (package [7 pts], account [3 pts] entities) - Populate the database with the data from the provided JSON files. - DAO and repositorties implementation to read/write data from/to SQLite. - Sync Local DB with Cloud Firetore when the app starts.	40		
3) Signup and Login using Firebase Authentication	10		
4) Design and Testing Documentation *** Design documentation:** - 4 key lessons learned from Phase 1 - MVVM architecture diagram - Database schema diagram	10		

* **Testing documentation:** with evidence of correct implementation using snapshots illustrating the results of testing of UI and Repositories (you must use the provided template).			
6) **Discussion of the project contribution** of each team member [-10pts if not done]			
Total	100		
Copying and/or plagiarism or not being able to explain or answer questions about the implementation	-100		

* **Possible grading for functionality** - ***Working*** (get 70% of the assigned grade), ***Not working*** (lose 40% of the assigned grade and ***Not done*** (get 0). The remaining grade is assigned to the quality of the implementation.

In case your implementation is not working then 40% of the grade will be lost and the remaining 60% will be determined based on the code quality and how close your solution to the working implementation.

Solution quality also includes meaningful naming of identifiers (according to Android naming conventions), no redundant code, simple and efficient design, clean implementation without unnecessary files/code, use of comments where necessary, proper code formatting and indentation.

Marks will be reduced for code duplication, poor/inefficient coding practices, poor naming of identifiers, unclean/untidy submission, and unnecessary complex/poor user interface design.