CMPS 312 Project Phase 2 – Data Management using Firestore, Firebase Cloud Store and local SQLite Database (10% of the course grade).

The project phase 2 submission is due by 8am Thursday 14 December 2023. Demos will be organized on the same day.

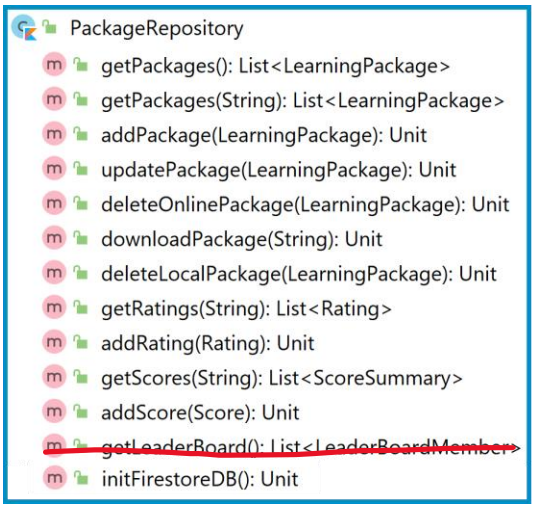
1. Deliverables

You are required to complete the delivery of *LingoSnacks* app to aid language learning for both the teachers and learners. 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 the base solution having the **full** phase 1 app solution. You are required to manage the app data using Firestore, Firebase Cloud Store and local SQLite Database. The app should provide the ability to download learning packages and their associated media so they can be used **offline** without internet connection.

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 repository methods to read/write entities using Cloud Firestore as the data source. All **data filtering should be done on the server** using Cloud Firestore queries and only the required data should be retrieved.
   Firure 1 shows an initial design of the repository classes provided in the base solution. You may **add/modify** the suggested methods as needed. Also note that some **ToDo** notes were added to the repository to provide further tips and clarifications.
   Also, it is important to ensure that the list of displayed packages is properly refereshed after add/update/delete operations.



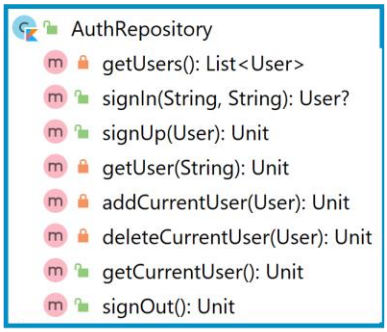


Figure 1: LingoSnacks repositories class diagram

3. Enhance the Learning Package Resource Editor to allow to user to attach an image/video from the galary or take an image/video using the Camera.

   When adding/updating the learning package to Firestore you need to upload the associated media files to Firebase Cloud Storage. Also, when you delete a package you should delete its associated media files from Firebase Cloud Storage.

4. Initialize the Firestore database: when the app starts, check if Firestore database is empty then initialize it with data from *the json files unders the **assets** folder*.

5. Document your database design in a schema diagram.

Important notes: When adding entities (e.g., a Learning Package) to Firestore you should let Firestore auto-set the entity id (e.g., auto-assign the **packageId** when adding a learning package).

6. Download a learning package and their associated media

Allow **download** a learning package and its associated media to enable the user to use the learning activities offline. The package details should be stored in a Local SQLite Database while the Cloud Firestore media files associated with the package should be downloaded to the mobile device.

6.1. Implement the needed entity annotations to manage the package data in a local SQLite database using Room to allow offline usage of learning packages even without internet.

Tip: To keep the database **design simple,** you may use a Type Converter to store the learning package Words (and the associated definitions, sentences and resources) as a single JSON String column in the LearningPackage table).

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

6.3. Populate the database with the learning package data fetched from Cloud Firestore.

6.4. Download Cloud Firestore media files associated with the package to the mobile device. Then update the Urls inside the package to point to local multi-media files (this update should be done in the local DB only NOT the online version of the package).

6.5. If the downloaded package already exists in the local DB and if the online version is greater than the local version then sync the the local package with the online version.

6.6. While the user plays the Unscamble Sentence and Match Definition games record their scores in the local DB if no internet is available then push the recorded scores to Firestore when the internet connection is available.

6.7. Document your database design in a schema diagram.

Part 2 – Signup and Signin using Firebase Authentication

7. Implement signup and authentication using Firebase Authentication. Upon sign-up a user account should be created on Firebase Auth (using the user email and password) and the user details should be stored in a **users** collection in Firestore.

Part 3 - Design and Testing Documentation

8. Document 3 **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 technical lessons learnt.
9. Document the MVVM architecture diagram for your overall solution design.
10. Firestore database schema diagram and SQLite database schema diagram.
11. Write a testing document including screenshots of conducted tests illustrating a working implementation.
12. 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 questions channel on Teams.

- 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 online and during office hours.

2. Grading rubric

Criteria	%	Functionality*	Quality of the implementation
1) Cloud Firestore Database Design and Implementation Repositories to interact with Firestore	50		
2) Download a learning package and their associated media - Populate the database with the learning package data fetched from Cloud Firestore. - Download Cloud Firestore media files associated with the package to the mobile device. - If the downloaded package already exists in the local DB, sync the the local package with the online version. - Record *Unscamble Sentence* and *Match Definition* scores in the local DB if no internet is available then push the recorded scores to Firestore when the internet connection is available.	30		
3) Signup and Signin using Firebase Authentication	10		
4) Design and Testing Documentation *** Design documentation:** - 3 key lessons learned from Phase 1. - MVVM architecture diagram.	10		

- Firestore database schema diagram and SQLite database schema diagram. *** Testing documentation:** with evidence of working implementation using snapshots illustrating the results of your solution testing (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.