CMPS 312 Project Phase 1 – UI Design and Classes Implementation (15% of the course grade).

The project phase 1 submission is due by 9am Sunday 11st October 2020. Demos will be organized during office hours in the same week.

1. Requirements

You are requested to design and implement *LingoSnacks* to aid language learning for both the teachers and learners. Each team should select to work **either** on the ***Learning Package Editor App*** to be used by the teachers to create learning packages or the ***LingoSnacks App*** to be used by the students for game-like learning activities.

- *Learning Package Editor* App should allow teacher to provide a list of words and associated word definitions and sentence examples. Words and/or sentences in the package can have associated photos, videos, or web links.
- *LingoSnacks App* should allow the learner to list / search the available learning packages and get the desired one from an online repository. Then the app will use the downloaded package to provide content to interactive learning activities to study and practice the content of the learning package.

The main of the app is to facilitate the authoring of the learning material for the teacher; meanwhile providing an entertaining learning experience for the students.

Note that storing data in a local database and reading/writing data from remote Cloud Firestore will be done in phase 2 of the project. For phase 1, data should be kept in memory and if needed json files could be used.

The main LingoSnacks use cases are described Table 1.

Table 1. Use cases description

Use case	Brief description
Learning Package Editor App	
T1 - Login	Allows the user to login. Login is prerequisite for all the teacher use cases shown below.
T2 - Sign Up	Allows the user to create an account to be able to sign in.
T3 - List and search packages	Get a list of learning packages with the ability to search by keyword or level. From the list of packages displayed, the user can select to edit/delete a package (only if the package was created by them).
T4 - Delete Learning Package	Teacher can delete a learning package (they have created earlier. They should NOT be able to delete packages not belonging to them. The App should ask for the user confirmation before deleting.
T5 - Add/Update Learning	Teacher can add/update the package details as shown the package

Package. This use case includes:	entity.
T5.1 – Add/Update/Delete Words	Add/Update/Delete the words to be included in the learning package.
T5.2 - Add/Update/Delete Definitions	Add/update/delete one or many definitions for each word.
T5.3 - Add/Update/Delete Sentences	Add/update/delete one or many sentences for each word.
T5.4 - Attach Multimedia	Teacher can attach multimedia content (photos, videos or web links) to a word or a sentence. The photos and videos should be either recorded using the phone's camera or selected from the image gallery.
T5.5 – Save and Publish the Learning Package	Publish the package to the server so that the students can get it. In phase 1 saving the package locally is enough.
LingoSnacks App	
S1 - Login	Allows the user to login. Login is optional. It is only needed if the user would like to evaluate a particular Learning Package.
S2 - Sign Up	Allows the user to create an account to be able to login. This is optional and only needed if the user needs to login to enter the evaluation of a learning package.
S3 - List and search learning packages	Get a list of learning packages with the ability to search by keyword or level. Once the list is displayed the user can select the desired one to download and use as content for the learning games.
S4 - Play Flash Cards	Play Flash Cards based on the package content with the ability to loop through them and play/display multimedia content associated with words or sentences (e.g., image, video, or web link) within the app using Android VideoView and WebView.
S5 - Unscramble Sentences	The learner needs to reorder the words of a sentence to form a meaningful sentence. The user can loop through and play this game for all sentences in the package. The app should validate the user attempt and indicate whether it was successful or not.
S6 - Match Word & Definition	Given a pool of words and their definitions, the learner should match a word and its corresponding definition. The user can loop through and play this game based on the words and the definitions in the package. The app should validate the user attempt and indicate whether it was successful or not.
S7 - View Scores	The app should keep track of the user score as they play the game. The user should be able to view their scores for each game type (Unscramble Sentences, Match Word & Definition)
S8 - Rate Learning Package	Enter rating and comments for a learning package. Login is required to do so.

The entities class diagram is shown in Figure 1. These are the base entities provided as a guide you may update them or enhance them as needed.

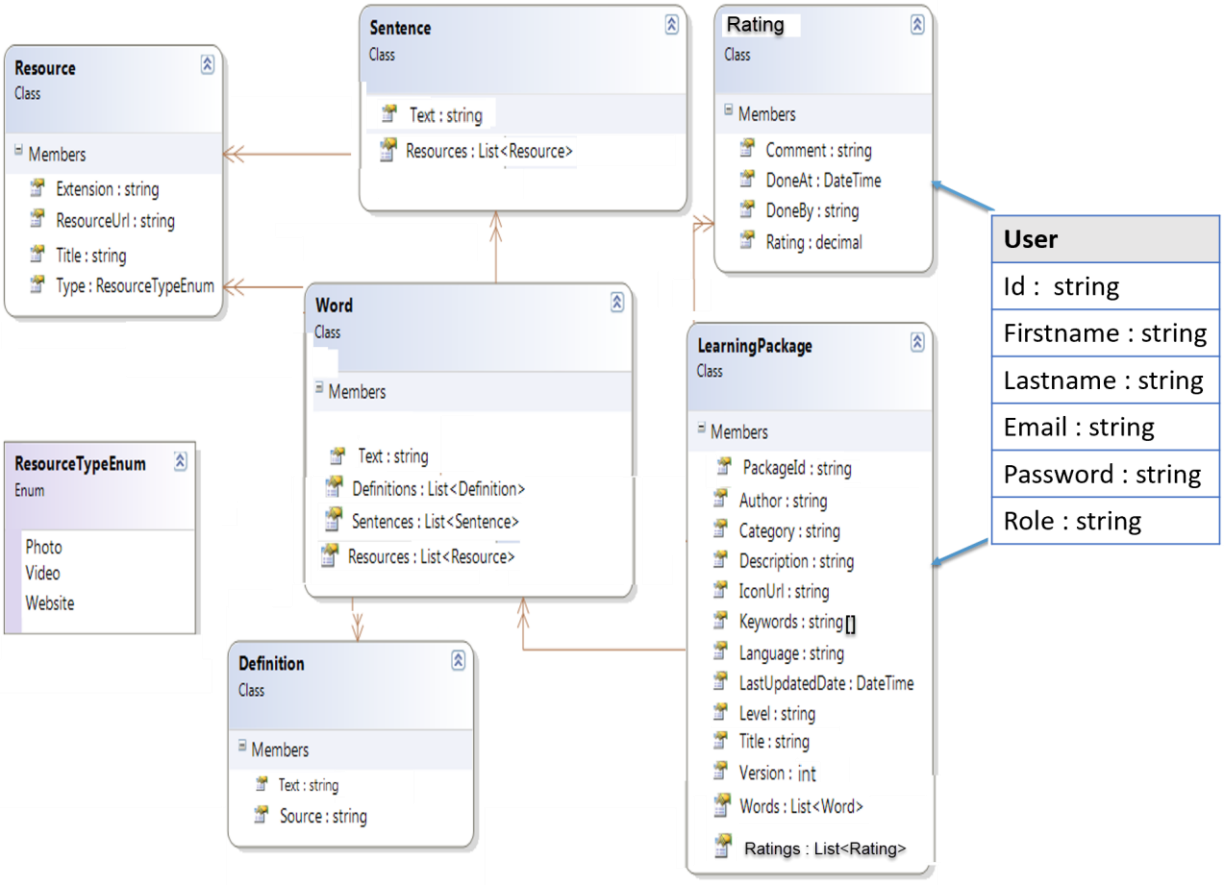


Figure 1. LingoSnacks entities class diagram

Important notes about the entities class diagram:

- The class properties should be lowercase as per Android naming convention. They are shown in uppercase in the diagram by the tool used to create this diagram.
- LearningPackage.packageId should be auto assigned and not entered by the user
- LearningPackage.version should be auto-incremented every time the package is updated.
- The learning package should be read/written as an aggregation that could be serialized as single json document.

2. Deliverables

Seek further clarification about the requirements/deliverables during the initial progress meeting with the instructor. Note that further important clarifications maybe modified/added to the project requirements.

1) Application design documentation that includes the Repositories Class Diagram.
   During the weekly project meetings with the instructor, you are required to present and discuss your design with the instructor and get feedback. You should only start the implementation after addressing the feedback received about your design.

2) Implement UI for each use case following design best practices. The UI should be fully working using learning package data loaded from a json file. Remember that 'there is elegance in simplicity'!
3) Design and implement the app navigation. It should be fully working, and the user can navigate from one activity to another in intuitive and user-friendly way.
4) Implement the entities and repositories using Kotlin. They should be fully working. Create some test Learning Package data to ease testing. First test them using a main function that displays the results to the console before using them in the UI.
5) Document the testing of UI and repositories using screen shots illustrating the testing results.
6) Every team member should submit a description of their project contribution. Every team member should demo their work and answer questions during the demo.

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

3. Grading rubric

Criteria	%	Functionality*	Quality of the implementation
1) Application Design - Repositories Class Diagram.	5		
2) Design and implement the **UI**	50		
3) Design and implement the **UI Navigation**	10		
4) Implement the entities and repositories using Kotlin	30		
5) Testing documentation using screen shots illustrating the testing of UI and Repositories.	5		
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 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 of 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.

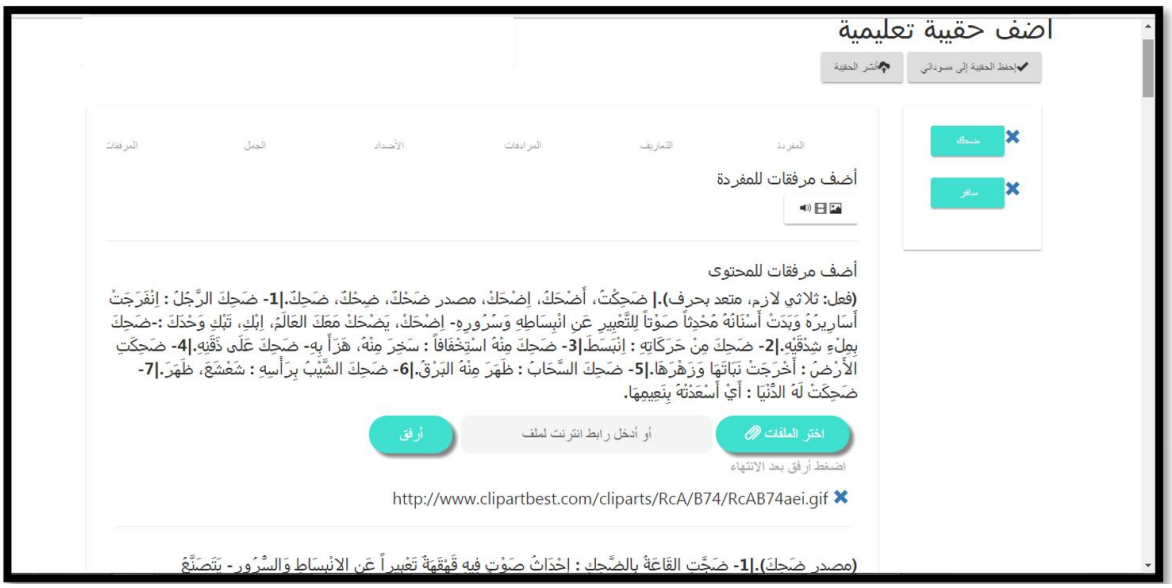
Appendix 1 - Example UI design (These screen shots are provided just to clarify the requirements. Surely, they are NOT the recommended design. Be creative and come-up with your own design better than one below).

1. **Teacher use cases**

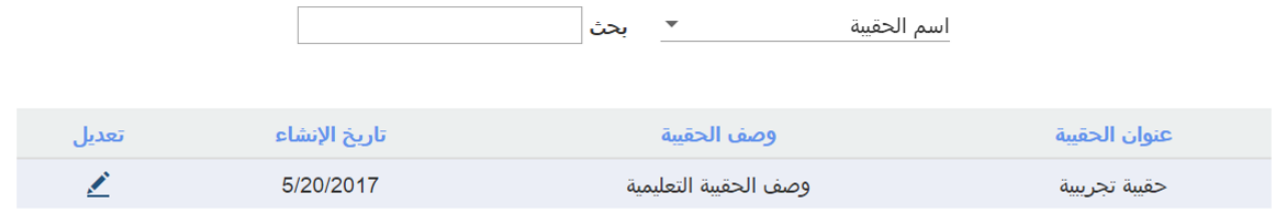
Learning package Editor


معلومات الحقيبة:
عنوان الحقيبة
حقيبة تجريبية
وصف الحقيبة
وصف الحقيبة التعليمية
الفئة
أخرى..
المستوى
مبتدىء
أيقونة الحقيبة
المرفقات
تابع

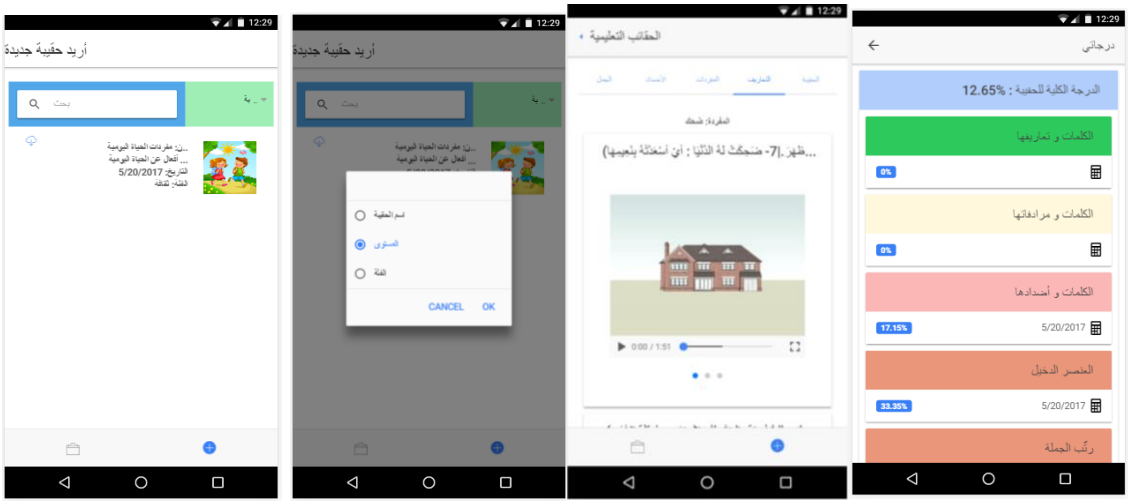

أضف حقيبة تعليمية
إنشاء المحتوى
قاموس المعاني
(فعل: ثلاثي لازم، متعد بحرف).| ضَحِكْتُ، أَضْحَكُ، إِضْحَكْ، مصدر ضَحْكٌ، ضِحْكٌ، ضَحِكٌ.|1- ضَحِكَ الرَّجُلُ : إِنْفَرَ
قاموس المعاني
(مصدر ضَحِكَ).|1- ضَحِكَ الغَاعَةُ بِالضَّحِكِ : إِحْدَاثُ صَوْتٍ فِيهِ قَهْقَهَةٌ تَعْبِيرًا عَنِ الانْبِسَاطِ وَالسُّرُورِ- يَتَصَنَّعُ الضَّحِ
تعريف
أضف تعريفا جديدا


اضف حقيبة تعليمية
أضف مرفقات للمفردة
أضف مرفقات للمحتوى
(فعل: ثلاثي لازم، متعد بحرف).| ضَحِكْتُ، أَضْحَكُ، إِضْحَكْ، مصدر ضَحْكٌ، ضِحْكٌ، ضَحِكٌ.|1- ضَحِكَ الرَّجُلُ : إِنْفَرَجَتْ أَسَارِيرُهُ وَبَدَتْ أَسْنَانُهُ مُحْدِثًا صَوْتاً لِلتَّعْبِيرِ عَنِ انْبِسَاطِهِ وَسُرُورِهِ- إِضْحَكْ، يَضْحَكُ مَعَكَ العَالَمُ، إِبْكِ، تَبْكِ وَحْدَكَ :-ضَحِكَ بِمِلْءِ شِدْقَيْهِ.|2- ضَحِكَ مِنْ حَرَكَاتِهِ : إِنْبَسَطَ.|3- ضَحِكَ مِنْهُ اسْتِخْفَافاً : سَخِرَ مِنْهُ، هَزَأَ بِهِ- ضَحِكَ عَلَى ذَقَنِهِ.|4- ضَحِكَتِ الأَرْضُ : أَخْرَجَتْ نَبَاتَهَا وَزَهْرَهَا.|5- ضَحِكَ السَّحَابُ : ظَهَرَ مِنْهُ البَرْقُ.|6- ضَحِكَ الشَّيْبُ بِرَأْسِهِ : شَمْشَعَ، ظَهَرَ.|7- ضَحِكَتْ لَهُ الدُّنْيَا : أَيْ أَسْعَدَتْهُ بِنَعِيمِهَا.
اختر الملفات
أو أدخل رابط انترنت لملف
أرفق
اضغط أرفق بعد الانتهاء
http://www.clipartbest.com/cliparts/RcA/B74/RcAB74aei.gif
(مصدر ضَحِكَ).|1- ضَحِّتِ القَاعَةُ بِالضَّحِكِ : إِحْدَاثُ صَوْتٍ فيه قَهْقَهَةٌ تَعْبِيراً عَنِ الانْبِسَاطِ وَالسُّرُورِ- يَتَصَنَّعُ

1. **Student use cases**



(Note that only teachers can edit a package)



View Scores