

Design Document

To utilize what we have learnt in the Mini 1, we implemented our project as shown below. For the structure, we have separated our project into two parts, an internal part and an external part, as shown in Figure 1, so we can hide away the internal code. Then, in order for the two parts of the project to interact, we have designed several interfaces, such as CreateAlarm and DeleteAlarm, so that we can connect the external part of the project to the internal part.

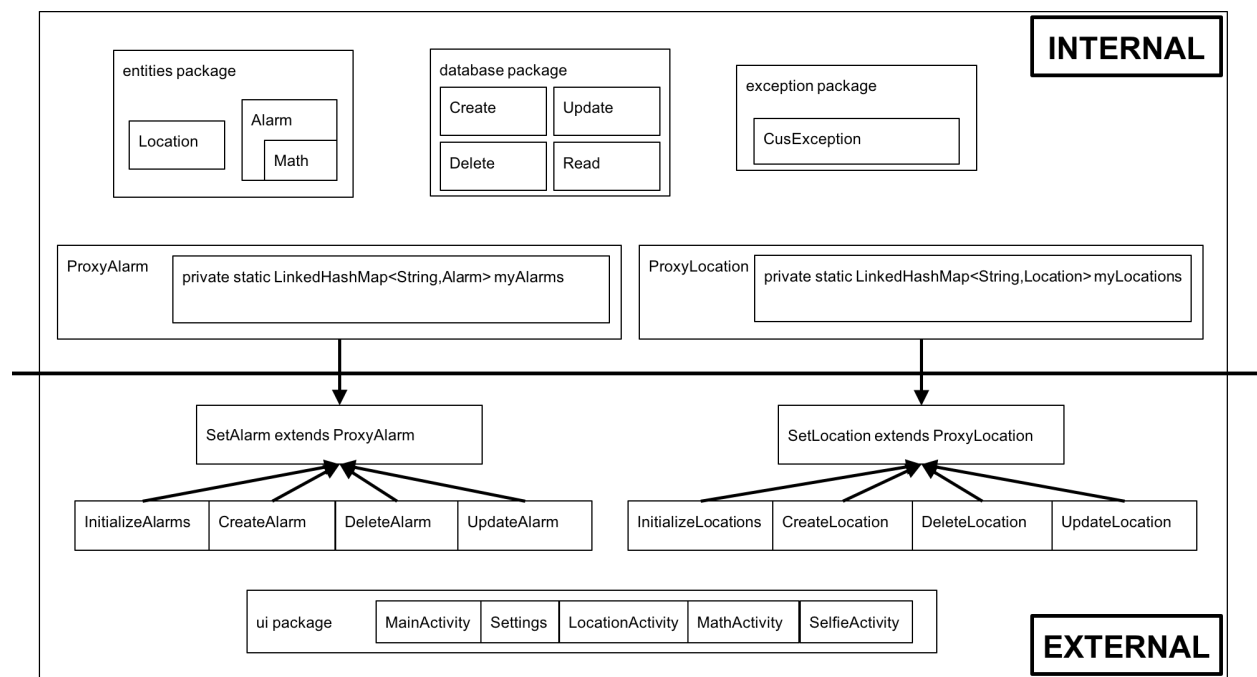


Figure 1 – Structure of our project.

In designing our project, we followed the structure and constructed a total of four tiers, the presentation tier, the content provider tier, the application tier, and the integration tier as shown in Figure 2.

Class Diagram

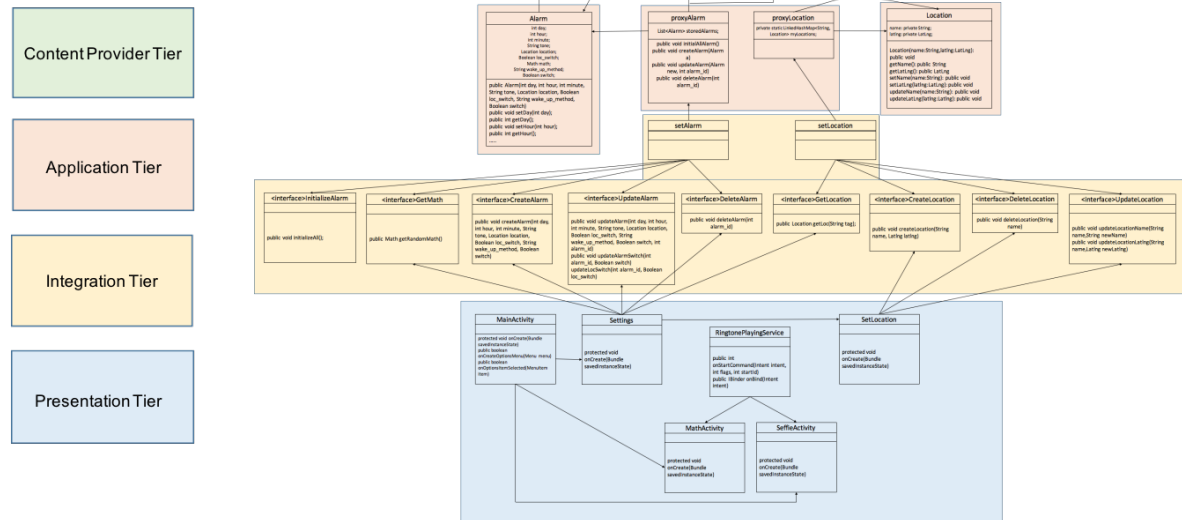


Figure 2 – Class diagram of our project.

For the application tier, we have created the entities package where our Alarm, Location, and Math objects live. The Alarm object will contain information about the alarm, such as the time it is supposed to go off, the location, the math question/answer, etc. For the Location object, it will have information such as the latitude and longitude. The Math object will contain information such as the actual question and the answer to that question.

Then for the content provider tier, we have one local SQLite database on the Android device and another remote MySQL database on a server. We separate into two different databases because some information that we want to store is individualized while some other information is globally the same for all user. For Alarm and Location, we store these in the local SQLite database since each user will have different alarms and locations. Also, it is faster and you don't require a connection if all these information is stored on user's own device. The Math objects we have are stored in the remote MySQL database since we have pre-determined question-answer pairs that all user can use. The database schemas for the two databases are show below in Figure 3.

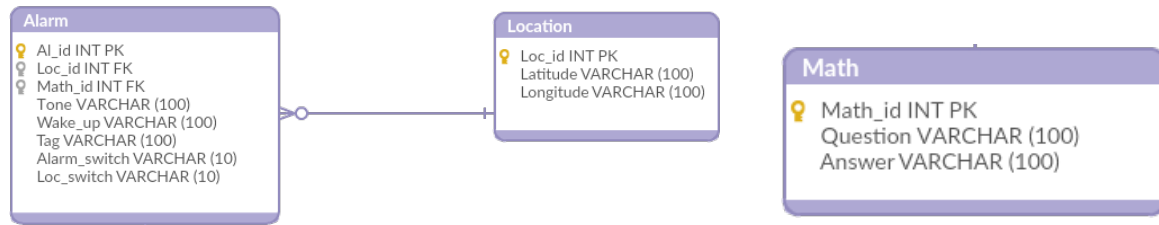


Figure 3 – Database schemas of the local and remote databases.

For the integration tier, we have the the two interfaces SetAlarm and SetLocation that are the APIs for accessing the internal part of our project. Within each interface, we also separated all the API function calls into different classes according to their usage. For example, we have function calls that create an alarm separated from the calls that can delete an alarm. By using these APIs, the presentation tier of our project will be able to interact with the application and content provider tiers.

Lastly, for the presentation tier, we have the package ui where it contains all the Activities for the Android application. We have a total of five Activities: MainActivity, Setting, LocationActivity, MathActivity, and SelfieActivity. We have the MainActivity as the main screen where it will show all the alarms and a button for the user to add new ones. Then we have the Setting where the user can customize a particular alarm. Then in the LocationActivity, it allows the user to search for a particular location and add it to the alarm. Finally, the MathActivity and SelifeActivity are for when the alarm sounds and the asks that the user has to complete before shutting off the alarm.