Mobile Application Development – C196

Bao Quoc Tran – SID 001683938

Western Governors University

Table of Contents

1. Tablet V.S Phone Development

When developing an application for a tablet, compared to a smartphone, there are several factors in layout, user experience, and screen real estate that need to be considered:

- Fragments and Layouts:
    - By taking advantage of Android reusable components, it is possible to build flexible layouts and create a responsive UI to adapt to different device screen sizes.
    - Compared to a phone, which mostly uses a single-pane layout for navigation, the fragments used in tablet design could have different sections for a single activity.
- UI/UX Optimization and Screen Real Estate:
    - More screen real estate which offers more space to display content, such as additional columns in a data grid, larger images, or additional navigation options.
    - Enable drag-and-drop features or split-screen multitasking to enhance user experience.
    - The user interface must be able to adapt to the continuous changes in the landscape and portrait modes, especially separate layouts, or different fragments while in landscape mode, which tablet users often view and interact in this mode.

2. Operating System & Min/Target SDK

Operating System: Mac OS Sonoma V14.0

MinSDK V26

TargetSDK V30

3. The Challenges

I was approached to the project as a front-end development method, which building the GUI first then starts to add entities and adapters later down the road. When it comes to SQL Lite sections, I realized that this approach was way off since I either don't have or set incorrect entities and adapters, which causes the information to not display, or display incorrectly due to the wrong entity and wrong primary/foreign key of SQL Lite.

When the issues were fixed, another issue came up as not being able to delete the term, even though all the courses had been deleted. Also, the alarms were confusing me at first and it took me a while to set up correctly, including the note share function.

4. The Solutions

By having the conversation with the Course Instructor, I understand the bottom-up approach is a better way to start this scheduling project. I created all entities, adapters needed first, then declared all the Term ID, and Course ID appropriately with the setup of primary key/ foreign key for SQL Lite. Also, thanks to the Room Frameworks, all the databases have been clearer and easier to access the data.

When the value/ database structures have been set up, it is now time to implement them into each UI class without any confusion. The same thing with terms/course validation to make sure the backend is updated each time the application creates/updates/deletes values to properly display appropriate information.

5. The Different Approach

Another piece of advice from the conversation with the instructor is to have the plan visually created before starting to code, in this case, is to create a storyboard with all the foundation needed, such as term id, course id, term list, course list and so on with the declaration to easy navigate while setting up database.

When the storyboard is finished with all the requirements needed, then it is time to create each of the entities, adapters, and classes needed to run the application well-structured from the beginning, instead of using the top-down method, which creates the GUI first and got stuck from the first place.

6. The Emulators V.S Development Device (Pros/ Cons)

Emulators are a tool that simulates a certain hardware/ software environment, allowing developers to test and run without buying actual physical devices, such as tablets, phones, and wearable devices…

The pros of using emulators are cost-efficiency, mostly built-in with the software development tools, and free-to-use. Platform flexibility is also a key point to using the emulators since developers don't have to buy all devices that just for testing. Since emulators provide a virtual environment to run and test the application, the risk of data loss or device damage is greatly reduced as well.

The cons of emulators would be the limitation of hardware features, such as fingerprint, face unlock, GPS, and cameras, which limit the testing of certain features. Also, resource-intensive is always a topic to discuss whenever developers looking to upgrade or buy a new computer just to make sure it can run well when using emulators for their apps. Finally, user experience testing is a valid point to argue when it comes to comparing between emulators and physical devices. With a lack of real experience on an actual device, it is difficult to give accurate feedback as a UI tester. The different behavior on physical devices when running certain tasks of the application is also a big drawback for emulators since the battery usage, and mobile network signal don't apply in these cases.