Neelofar Karimi
C196
Reflections and APK Screenshots

1. **Explain how your application would be different if it were developed for a tablet rather than a phone, including a discussion of fragments and layouts.**

The biggest difference in developing an application for a tablet as opposed to a phone involves utilizing the larger screen size to display more content simultaneously. With a phone layout, fragments are used in a stack, where one fragment replaces another. With a tablet layout, fragments are used side by side, so things are visible at the same time. This improves usability and leverages the extended screen real estate to provide a more accessible user interface.

2. **Identify the minimum and target operating system your application was developed under and is compatible with.**

The application was developed with a minimum SDK of 26 (Android 8.0 Oreo) and targeted SDK 34 (Android 14.0).

3. **Describe *(suggested length of 1–2 paragraphs)* the challenges you faced during the development of the mobile application.**

Creating a user-friendly interface that logically associates assessments with courses was initially confusing. User friendly features usually involve having more than one way to do something without creating confusion and clutter. I tried to take that into consideration, but I also wanted to ensure that users understood the associations in a streamlined way. I had to change the application's flow a few times until I landed on the current design, even though it could still use improvements.

I had a lot of trouble implementing the notification system, particularly with scheduling alarms accurately due to discrepancies between system time and app time. At first, I attempted using "schedule exact alarm" instead of the solution presented in the course instructor's video, due to wanting to try a more robust method. It failed because the app couldn't set notifications for the same day due to the date/time discrepancies. I added logging statements to validate that the issue was due to the time difference. I did not want to set a fixed date/time for the application because it could not be a long-term solution.

Initially, I wanted to use a selection model for actions with recycler views in list screens. I believed that having three buttons on the bottom of a recycler view was sleeker than having the same buttons on every row. This method proved to be technically challenging and unintuitive for users, even though it would have simplified the UI substantially. This led to a redesign of my recycler view rows.

4. **Describe *(suggested length of 1–2 paragraphs)* how you overcame *each* challenge discussed in part F3.**

To solve the application flow issue, I constrained functionality to detail screens. This simplified user flows and clarified the process of adding associated elements, like assessments to courses. Detail screens contain a single existing course, assessment, or term. Since the data for that element is detailed there already, I thought that it would be best to keep details, associations, and the functions of that association in one screen.

After struggling with direct scheduling using an "exact alarm" method, I followed the course instructor's approach but abstracted it into a utility class for reuse across different parts of the application. It worked as expected but then I discovered that I had to find a way to display the specific element's details on the notification, as the notification could not be a general "Course is Starting on XX-XX-XXXX". I already had a way to get this information due to my associated elements functionality, so it was a straightforward fix.

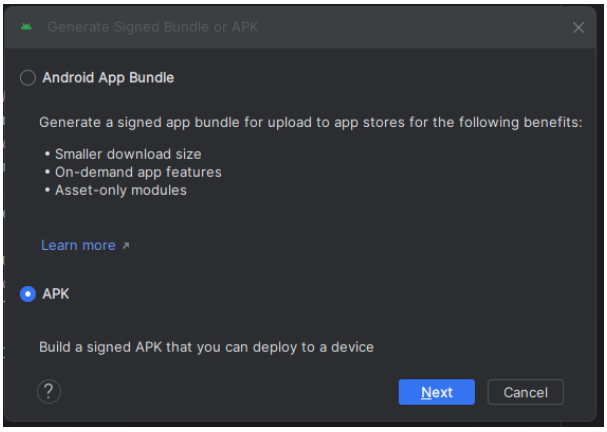
The implementation of direct-action buttons on recycler view items instead of a separate control set resolved interaction issues. In addition, I was able to use these redesigned items for other screens, such as the associated elements list, and to display and manage notes.

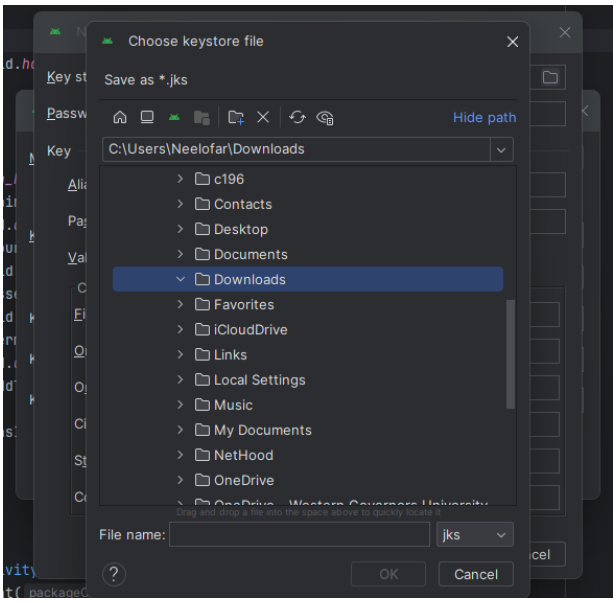
5. **Discuss *(suggested length of 1–2 paragraphs)* what you would do differently if you did the project again.**

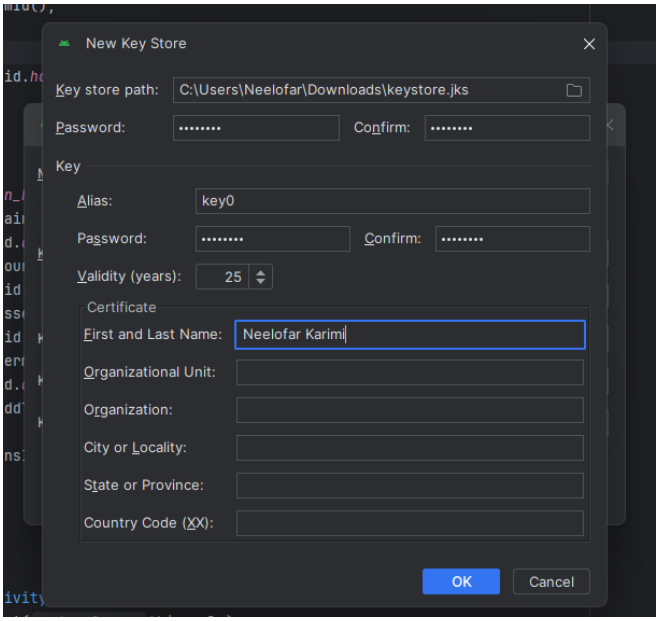
A more thorough initial design phase could have prevented the usability and application flow issues I ran into. Providing multiple pathways for common actions (like adding assessments) could enhance the user experience. Introducing interactive elements like floating action buttons for adding terms could make the application easier to navigate.

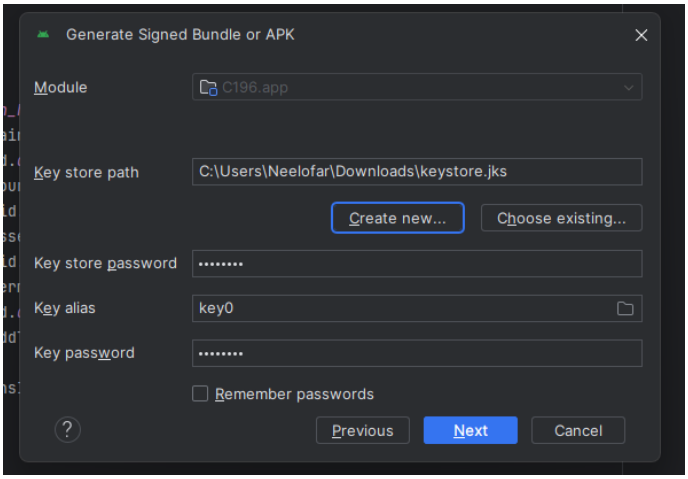
6. **Describe how emulators are used and the pros and cons of using an emulator versus using a development device.**

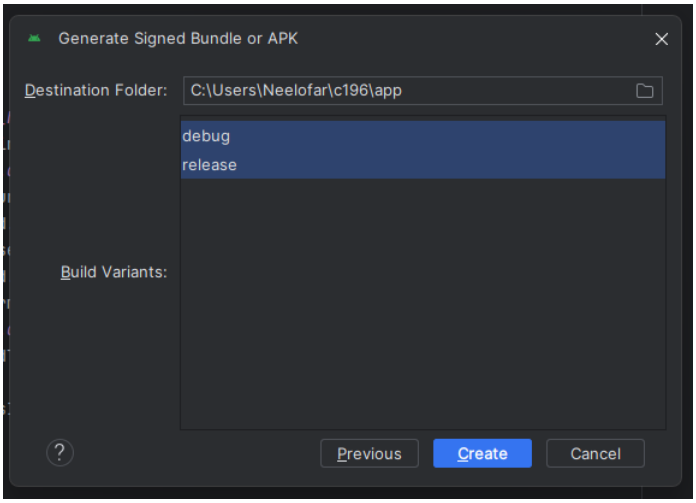
Emulators allow testing across different devices and configurations without physical hardware, which is very important in the development/testing process. However, they are resource-intensive and may not accurately replicate real device performance, especially on older/less powerful machines. My experience highlighted the limitations of emulators, as they were extremely sluggish on my system with 8gb of RAM. Using an emulator in development and a development device during testing could be a way to harness the advantages of both methods, if the developer's system has enough memory.

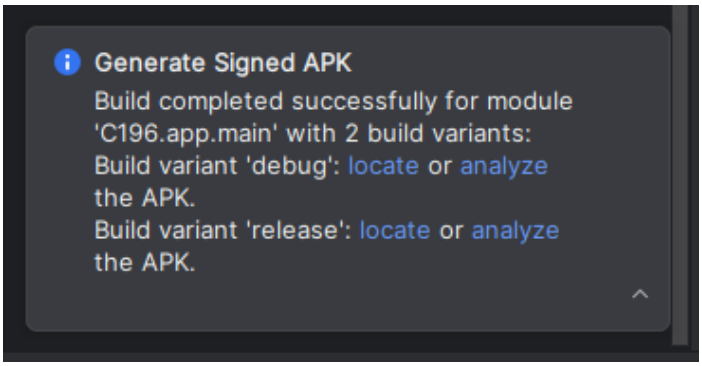

Generate Signed Bundle or APK
Android App Bundle
Generate a signed app bundle for upload to app stores for the following benefits:
• Smaller download size
• On-demand app features
• Asset-only modules
Learn more
APK
Build a signed APK that you can deploy to a device
Next
Cancel


Choose keystore file
Save as *.jks
Hide path
C:\Users\Neelofar\Downloads
c196
Contacts
Desktop
Documents
Downloads
Favorites
iCloudDrive
Links
Local Settings
Music
My Documents
NetHood
OneDrive
File name:
jks
OK
Cancel


New Key Store
Key store path: C:\Users\Neelofar\Downloads\keystore.jks
Password:
Confirm:
Key
Alias: key0
Password:
Confirm:
Validity (years): 25
Certificate
First and Last Name: Neelofar Karimi
Organizational Unit:
Organization:
City or Locality:
State or Province:
Country Code (XX):
OK
Cancel


Generate Signed Bundle or APK
Module
C196.app
Key store path
C:\Users\Neelofar\Downloads\keystore.jks
Create new...
Choose existing...
Key store password
Key alias
key0
Key password
Remember passwords
Previous
Next
Cancel


Generate Signed Bundle or APK
Destination Folder:
C:\Users\Neelofar\c196\app
debug
release
Build Variants:
Previous
Create
Cancel


Generate Signed APK
Build completed successfully for module 'C196.app.main' with 2 build variants:
Build variant 'debug': locate or analyze the APK.
Build variant 'release': locate or analyze the APK.