Name: Dang Khoa Le
Student ID: 103844421

Unit Learning Reflection

Introduction: In this report, I reflect on my learning journey in COS30017, Mobile Application Development, where I have explored various facets of Android app development and mobile system design. This reflection highlights how I have applied the Unit Learning Outcomes (ULOs) and discusses key experiences, challenges, and insights gained throughout the semester.

Approach to the Unit:

With two years of web development (HTML, Angular.js, Vue.js) and one year of Android app development (Java and Spring API), I approached this unit confidently, expecting an easy transition. My experience with Kotlin projects, such as a Roster Management app and a Guitar Tuner app, provided a foundation. However, I soon realized that there was much to learn about complex app development concepts, theories, and building truly mobile-friendly applications.

Meeting the Unit Learning Outcomes (ULOs):

ULO 1: System Differences in Mobile vs. Desktop Development

In Assignment 1, I adapted app design to mobile constraints, utilizing **ConstraintLayout** for efficient UI, **Locale** for multi-languages support and optimizing data storage with **Room** Database in Assignment 3 to handle memory and processing limits.

ULO 2: Effective Mobile Application Design

For the **Rent with Intent** app (**Assignment 2**), I implemented chip-based filtering for better user interaction and efficient data sharing between activities using **Intent** and **Parcelable**, ensuring performance on limited mobile resources.

ULO 3: Building, Testing, and Debugging

In **Assignment 3**, I developed a multi-fragment CRUD app with **Room**, **ViewModel**, and **LiveData**, optimizing performance via **Android Profiler** and unit testing with **Espresso** (also **Assignment 2**), ensuring data persistence and UI responsiveness across assignments.

Challenges in Mobile Development:

Despite my confidence, I found optimizing **CPU** and **Memory** usage in mobile development challenging. Mobile apps require careful memory management due to limited resources, as seen in the **Exercise Activity Tracker** app in **Assignment 3**, where I balanced animations and performance constraints. Transitioning from cloud databases (MySQL, MariaDB) to **Room** for local data storage was another adjustment, highlighting the need for efficient, localized data handling in mobile apps.

Additionally, designing an effective UI to provide a user-friendly experience required careful consideration of screen space, accessibility, and intuitive navigation. This involved prioritizing

essential information, simplifying interactions, and choosing appropriate UI components to make the app easy to navigate and visually appealing.

Ensuring thorough UI-test coverage meant developing effective and comprehensive test cases that would capture a wide range of user interactions, including edge cases. It's fundamental to validate not only functionality but also the app's responsiveness and stability under different conditions.

Exploring Beyond the Unit:

This unit deepened my understanding of the MVVM architecture and helped me explore more advanced techniques for data handling and UI updates in Android apps. I gained a stronger grasp of asynchronous programming with Kotlin coroutines and how to incorporate these into mobile apps to ensure smooth performance and UI responsiveness. I also would like to continue exploring more about UI testing with Espresso, especially for larger applications, and automated testing approaches that streamline the app development lifecycle.

Key Takeaways:

1. **Developing for Mobile Platforms**:
    - Covered mobile-specific constraints (screen size, memory, processor) and setup with Android Studio and GitHub.
2. **Kotlin**:
    - Focused on Kotlin's syntax and algorithms, safety features, and its compatibility with Java for Android development.
3. **Core Components and Activity Lifecycle**:
    - Learned Android's activity lifecycle and state persistence, including ViewModel, SavedStateHandle, onView, onCreate etc for data retention.
4. **User Experience (UX)**:
    - Applied UX principles, using ConstraintLayout and Material Design for intuitive, scalable interfaces on smaller screens.
5. **Communicating Between Components**:
    - Mastered Intents, Bundles, and SafeArgs for secure data passing between activities and fragments.
6. **Working with Lists and RecyclerView**:
    - Managed dynamic data with RecyclerView, optimizing through adapters, ViewHolders.
7. **Concurrency and Persistent Data**:
    - Implemented Room for local storage, used coroutines for concurrency, and LiveData for real-time UI updates.
8. **Advanced Topics**:
    - Explored networking, secure data storage (Room, Firebase), and CRUD operations with error handling for robust apps.
9. **Putting Everything Together**:
    - Integrated learned concepts in a capstone project, utilizing MVVM architecture, Espresso testing, and responsive UI design.

During the semester, key events included exploring the differences between mobile development and other platforms. Examining mobile-specific design patterns such as MVI and MVVM. User experience (UX) was a major focus, ensuring applications are optimized for mobile interactions. We also explored the importance of testing through tools for UI validation. These activities emphasized efficient resource management, intuitive UI design, and building scalable, maintainable mobile applications.

Conclusion:

In conclusion, this unit has significantly enhanced my understanding of Android mobile development, from recognizing the system differences between mobile and desktop apps to applying practical solutions in real-world mobile applications. The combination of practical assignments and theoretical knowledge has helped me achieve the ULOs and prepared me for future mobile development projects.