Activity 1 - Software Engineering for Mobile Applications

Mobile Apps Development - LIS4012-1

Itzel Xhunaxhi Flores Ordaz - 174271
Iker Isla Torre - 176847
Daniel Martínez Maimone - 173825
Juan Carlos Sánchez Bronca - 173944

Universidad de las Américas Puebla

Dr. Julio Noé Hernández Torres

February 4th, 2025

Step	Software Engineering Steps	Software Engineering Steps for mobile Applications	Main Differences
Requirement Analysis	• Focus on college students to facilitate money management, finances and understand where their money is going. Help users keep their data updated	• We can take advantage of the fact that a mobile application could remind users not to go overboard with their expenses and update data in the application as soon as possible.	A mobile application offers the advantage of real-time notifications and reminders for users to manage their expenses efficiently, unlike traditional software, which may rely on manual updates.
Design	• The use of the architecture MVVM and clean architecture for a scalable and maintainable code. • Defined a well-structured and normalized database. • Define an intuitive interface taking advantage of the areas of the screen highlighting important places without saturating the user with too much information	• Since the screen space on mobile devices is reduced compared to a PC, we must find a way to present enough information without losing ease of navigation in the application.	Mobile applications require a more compact and intuitive interface due to limited screen space, ensuring information is presented efficiently without compromising usability, unlike desktop applications that have more display area.

Step	Software Engineering Steps	Software Engineering Steps for mobile Applications	Main Differences
Development	• Set up the development environment. • Develop functionalities as designed (e.g. user registration, data synchronization). • Integrate APIs and external services (e.g. Firebase, banking API). • Write clean, modular and well-documente d code.	• We are coding the application using the programming language Kotlin, for Android. • Integrate APIs and external services (e.g. Firebase, banking API). • Write clean, modular and well-documente d code.	Mobile applications require coding in specific programming languages (e.g., Kotlin for Android) and integrating mobile-specific APIs, whereas general software engineering may involve a wider range of technologies and platforms.
Testing	• Find a group of people to make testing and to get feedback about the software • See what kind of information we are look for these being quantitative or qualitative research	• Find a group of people to make testing and to get feedback about the software • See what kind of information we are look for these being quantitative or qualitative research	While both involve gathering feedback, mobile applications must consider factors like different screen sizes, touch interactions, and operating system variations, which are less relevant in traditional software testing
Deployment	Releasing the app to app stores (Google Play, App Store) after completing testing and approval processes. Ensuring the app is compatible with various devices and operating systems.	Publishing the app on app stores following store-specific approval guidelines. Conducting final testing for performance, security, and compatibility before release.	Mobile applications must comply with app store guidelines (Google Play, App Store) before release, whereas traditional software can often be deployed directly without such restrictions.
Maintenance & Updates	Regular updates to fix bugs, improve security, and introduce new	Adapting the app to OS updates, optimizing performance for new	Mobile applications require frequent updates to adapt to

Step	Software Engineering Steps	Software Engineering Steps for mobile Applications	Main Differences
	features in the app. Monitoring user feedback to enhance functionality and performance.	devices, and addressing user-reported issues. Implementing frequent updates to ensure smooth financial management for users.	OS changes, new devices, and app store policies, while traditional software maintenance focuses more on bug fixes and performance improvements.