**Application Documentation**

Isam Elder

Western Governors University

Bill Jing

Mobile Application Development - C196

1/28/2023

**Differences if this application was developed for an android tablet.**

The layout design of this application mostly utilized relative layouts whose contents are set to wrap around the device width and content. Few constants in terms of the width and the height of the layout were used. Even so, this application will not change on a tablet. Some of the layouts might be smaller on a tablet but overall their positions in the screen would not change.

**Compatibility of this application**

The code that schedules the alarm using the AlarmManager will work on devices running Android Oreo (API level 26) and higher. This is because, starting from Android Oreo, Android introduced new background processing restrictions to improve battery life and overall device performance. One of these restrictions is that background services are limited to a certain time window, after which they are considered to be idle and may be killed by the system. To work around these restrictions, Android provides new APIs for scheduling tasks, such as the AlarmManager API.

The AlarmManager API allows you to schedule alarms to perform actions at a later time, even when the app is not running. It is an effective way to work around the restrictions on background processing, and it is fully compatible with devices running Android Oreo and higher.

**Challenges faced while developing the application.**

Integrating the alarm notification proved to be challenging since it relies on the data from the last activity. If a user triggers the alarm in the last activity and continues to switch activities it throws an error since the intent has changed but if the user only triggers the alarm and remains in the same activity the alarm remains.

**The solution to the problem**

The solution to the problem above was simply to push the notification immediately after it is set and wait for the activity to finish. This way when the activity is destroyed, the alarm notification is already set. However, this also depends on whether the user provided access to the alarm.

**Things to change if the project was to be developed again.**

Using git version control while developing the application was a good decision since a breaking feature could be rolled back to a working one. However, large commits could mean rolling back a lot more than required. In the future, reducing the size of the commit would be the best approach to developing this application faster.

**Pros and cons of using an android emulator**

**Reasons to Use Android Emulators**

1. To imitate your Android tool for software testing, use a distinguished display.
2. Run any applications without being constrained by the Android tool.
3. Android emulators are used to check the execution of intentional test cases (Guerra-Manzanares et al., 2019).
4. Using simulated browsers to carry out cross-browser testing.

**Problems with Android Emulator**

1. The performance is subpar overall and the pace is slow when compared to the genuine tool.
2. It caused issues with software program loss and hardware optimization
3. For testing teams, the absence of reliable Android emulators presents a challenge (Guerra-Manzanares et al., 2019).
4. Disk control is rather limited in Android emulators.
5. Overall performance and hardware performance cannot be simulated.

References

Guerra-Manzanares, A., Bahsi, H., & Nomm, S. (2019). Differences in Android behavior between real device and emulator: A malware detection perspective. *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*. https://doi.org/10.1109/iotsms48152.2019.8939268