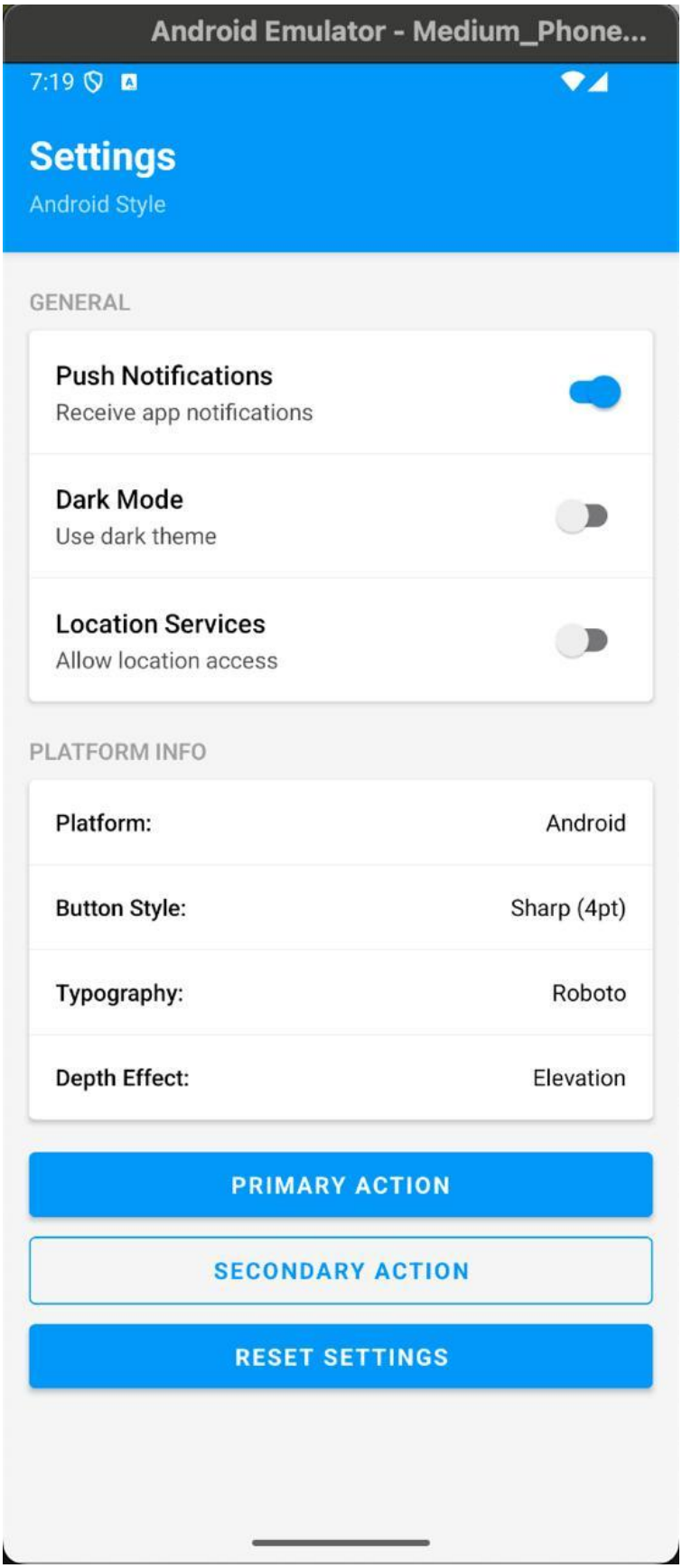
Lab 5

Aishwarya Chhablani

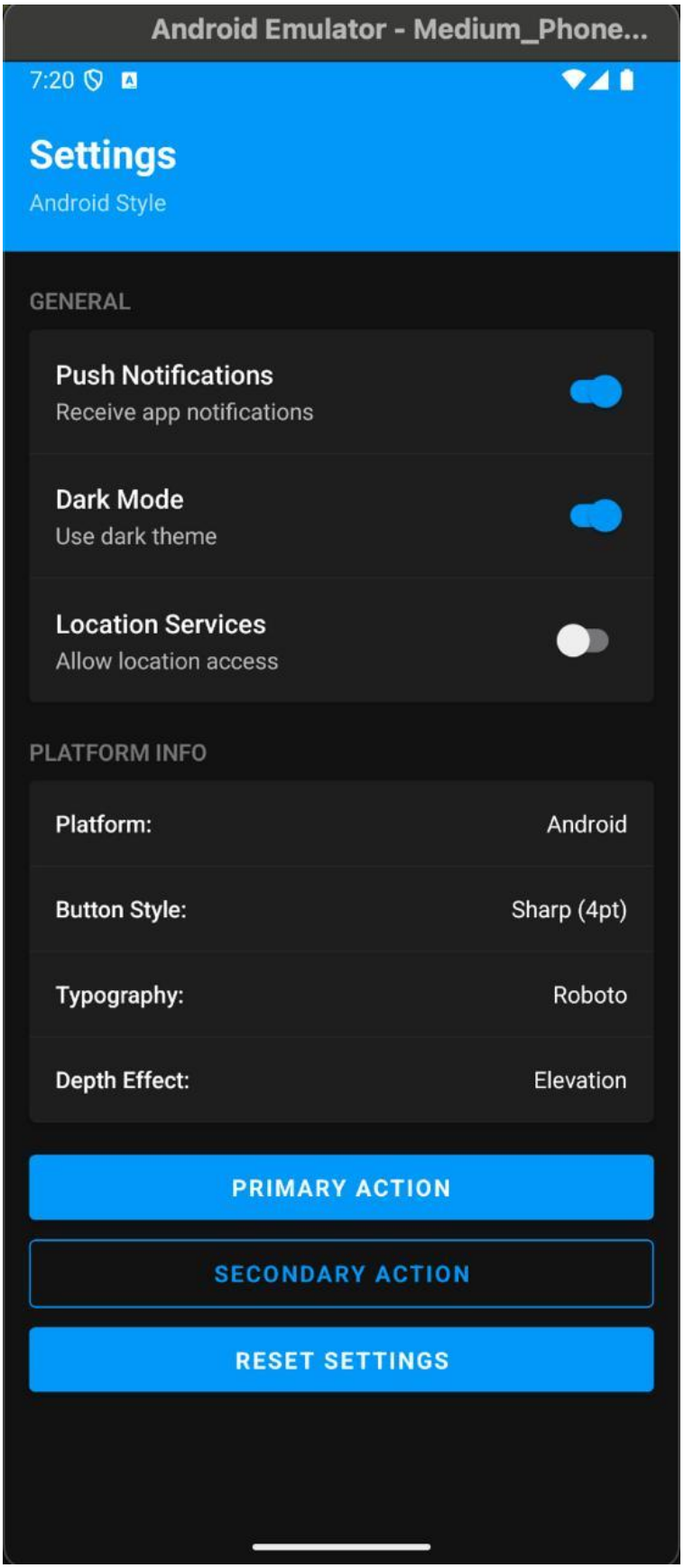
N01711300

PLATFORM SETTINGS APP

1.Screenshot of Android Main Setting Screen:



2. Screenshot of DarkMode


Android Emulator - Medium_Phone...
7:20
Settings
Android Style
GENERAL
Push Notifications
Receive app notifications
Dark Mode
Use dark theme
Location Services
Allow location access
PLATFORM INFO
Platform:
Android
Button Style:
Sharp (4pt)
Typography:
Roboto
Depth Effect:
Elevation
PRIMARY ACTION
SECONDARY ACTION
RESET SETTINGS

B.

Platform Difference (in 250-200 words):
Android and iPhone buttons have noticeably different visual styles. On android, buttons appear raised due to a soft shadow, creating a 3D effect that feels tactile and pressable. In contrast, iPhone buttons ae flatter and simpler, following Apple's clean and minimal design approach with very light or no shadows.
The text style also varies: Android commonly uses ALL CAPS like SAVE, while iPhone buttons display normal Captalization sucj as "Save".
When designing my buttons, I followed each platform's design guidelines . The android button was created with boldtext and deeper shadow, while iPhone button used softer tones and smooth finish. The colors and font weights were adjusted to match the look each system is known for.

By adapting the visuals to match the both Android and iPhone environments, I ensured each button blended naturally into its platform. This made the designs look authentic and consistent instead of identical copies.

C. Implementaiton Approach (150 words)
To make the app display correctly on both Android and iPhone, I developed two platform-specific files.
React Native automatically recognizes which one to use depending on the device.
I also added a short condition in the code to check the system type—"if Android, apply these styles; otherwise, use the iPhone layout."
The toughest part was balancing uniqueness with harmony. Android needed bold shadows and stronger color contrast, while iPhone demanded lighter, more delicate styling.
I tested both versions on real devices to verify that the appearance stayed consistent and functional across light and dark themes. The buttons looked distinct but behaved identically, giving users the same smooth experience on both phones.

D. Code Quality (100 words)

I structured my project to stay organized and readable.
All files were arranged into folders such as "components" and "screens," so navigation was quick and logical.

Each platform's button had its own file with a clear, descriptive name, reducing confusion during development.

I avoided repeating code by reusing shared functions and focusing on simple, well-formatted logic.

Through this project, I learned how important clean code and proper structure are when working across multiple platforms. Even with stylistic differences, keeping a tidy project helped maintain efficiency, readability, and a professional coding workflow.