Scenario Two

This document offers the ADR decisions for the first scenario.

ADR 1 - Native/Web/Hybrid App

Hybrid App

To ensure our app reaches as many students and professors as possible, we're going with a hybrid approach using ReactNative. This lets us support both iOS and Android platforms with a single codebase, saving time and resources while maintaining a high-quality user experience. ReactNative's ability to provide a native-like experience is crucial for our app’s functionality, especially when it comes to accessing schedules, announcements, and assignments smoothly. ReactNative also accelerates our development process by allowing us to see changes in real-time and fix issues quickly. This efficiency is vital in an educational setting where reliability and responsiveness are key. By sharing code across platforms, we not only streamline development but also ensure consistency and performance, making the app accessible and enjoyable for all users, regardless of their device.

ADR 2 - UI Framework

Decision - ReactNative

ReactNative is the ideal choice for our UI framework because it offers a seamless, native-like experience on both iOS and Android devices. This is essential in ensuring the app is accessible and user-friendly for all students and professors. ReactNative’s hot-reloading feature significantly reduces the time needed for testing and debugging, allowing us to quickly identify and resolve any issues.

ADR 3 - Backend Language

Decision - Node.js

Node.js is chosen for our backend because of its ability to handle high traffic and large amounts of data efficiently. Its event-driven, non-blocking I/O model is perfect for real-time applications like ours, ensuring that the app remains responsive even during peak usage times. This is crucial for a social networking app where students and professors are constantly interacting. Using Node.js also means we can leverage the widespread knowledge of JavaScript among our developers, speeding up the development process. Its rich ecosystem, with numerous libraries available through npm, supports rapid development and deployment, allowing us to build and scale our app effectively.

ADR 4 - Permissions

Decision - Fine Grained Permissions Management

To build trust and ensure security, we're implementing fine-grained permissions management. This approach ensures that the app only requests necessary permissions at appropriate times, protecting sensitive information like grades and personal profiles. By integrating with the university's existing Active Directory system, we can leverage established authentication mechanisms, providing secure and streamlined access control based on user roles (students, professors).

ADR 5 - Data Storage

Decision - Local/Cloud Storage

To provide a seamless user experience, we're using a combination of local and cloud-based storage. Local storage allows users to access essential features and data even when offline, ensuring the app remains functional in various network conditions. This is particularly important for students and professors who may not always have a stable internet connection. Cloud storage offers scalability and security for larger datasets, ensuring that as our user base grows, our app can handle increasing amounts of data efficiently. This hybrid approach balances the advantages of both storage types, maintaining high performance and data availability, and supporting our goal of creating a reliable, efficient, and user-friendly app.

ADR 6 - Additional Frameworks/Technology Stacks

Push Notifications - Firebase Cloud Messaging
Accessibility - ReactNative Accessibility API
Analytics - Google Analytics for Firebase

For our push notification service, we've chosen Firebase Cloud Messaging (FCM) because it reliably delivers updates across both iOS and Android platforms, ensuring that students and professors stay informed about important announcements, assignments, and events. This is essential for maintaining engagement and smooth communication within our university community. To gain insights into user behavior, we selected Google Analytics for Firebase. This tool helps us understand how users interact with the app, enabling us to make data-driven decisions to enhance performance and user experience continually. Finally, to ensure our app is accessible to everyone, including those who rely on assistive technologies, we've integrated the React Native Accessibility API. Features like text-to-speech and high contrast modes make our app user-friendly for individuals with diverse needs, reinforcing our commitment to inclusivity and creating an equitable digital environment for all students and professors. By combining these robust tools, we aim to build a comprehensive and effective social networking app that meets the diverse needs of our university community.