Tarunbir Singh

Martin Maquilan

Jose Ochoa

**Assignment: Architectural Decisions**

**Scenario 2: University Social Network Mobile App**

**Native, web, or hybrid app**

* *Status*

Proposed

* *Context*

The application must be compatible with both iOS and Android platforms, offer a responsive and seamless user experience, and accommodate offline functionality. A diverse array of individuals with differing degrees of technological proficiency and access to a variety of devices comprise the user base.

* *Decision*

We decided to develop a Hybrid App.

* *Consequences*

Easier: Hybrid apps are easier to manage and cost less to develop and maintain. The common codebase lets updates and new features to roll-out across all platforms concurrently, speeding up time to market.

Harder: Performance may be lower than the native apps and third party frameworks are used to access native device functionality, which may add complexity.

**UI Framework**

* *Status*

Proposed

* *Context*

The application needs a user interface that is easy to use and understand, accessible to every user including those with special needs, attractive and interactive. The general theme would be that one of a social network application but taking into account the additional features that relate more to those of an educational-institution application.

* *Decision*

First of all, consideration of accessibility is needed at all times, which includes proper font size, high contrast and other metrics. Also, users will be able to choose between light and dark mode at any time.

Initially for first time users, the login/register page will display, which will be simple and practical for users to be able to easily access the application. From there on, or on application load for existing users, the interface will have a horizontally aligned navigation bar at the bottom with all the application’s navigational options and a main scroll layout above it to view the detailed information and actions of the selected tab. These navigation options will change depending on the user mode which would be toggled with a small button: academic or user mode.

While in academic mode, the bottom navigation bar will display options such as grades, classes, schedule, class management, and others (these will change depending on whether the user is student or professor).

While in user mode, the bottom navigation bar will display options such as social profile, contacts, chats, discussions, and others.

* *Consequences*

With this user interface the users will have a simple yet attractive and full-of-features application which will also allow navigation for users with special needs.

**Backend language**

* *Status*

Proposed

* *Context*

The backend must be able to handle updates in real time, scale and integrate well with other services such as data synch and push notifications. Authentication, responsiveness and user data management all depend on a strong backend framework.

* *Decision*

Use Node.js

* Consequences

Easier: Node.js offers high performance and scalability which are essential for real time updates. Node.js encompasses a vast environment and has strong community support. JavaScript is already a familiar language to many developers which can reduce the learning curve.

Difficult: This decision will of course require expertise in JavaScript. There are potential issues with the management of asynchronous code and ensuring that the security follows best practices.

**Permissions**

* *Status*

Proposed

* *Context*

Using the university’s current Active Directory Infrastructure for user authentication and role administration the app will require secure access control. Ensuring users can only access the features they need and protecting sensitive data will depend on proper permissions and authentication.

* *Decision*

The decision will be to implement Role based Access Control integrated with the Active Directory.

* *Consequences*

Easier: RBAC will streamline user authentication and the authorization process. Utilizing the existing Active Directory for secure and reliable access control.

Difficult: The integration process with the Active Directory could prove to be complex, requiring specialized knowledge. It could prove difficult to ensure a seamless user experience during login and role assignment.

**Data storage**

* *Status*

Proposed

* *Context*

The application will need to a way to store information which will allow both online and offline usage, the latter naturally having less functions. Offline storage will be used for students to access their schedule and grades offline. Online storage will be used to store students and professors account credentials, user social items (such as user contacts, chats, etc...), listed courses (with their information and availability), events and club’s information, student schedules and grades, and other.

* *Decision*

Local data storage will use **SQLite** since it is a reliable local multiplatform storage solution. This will allow users to view certain information offline by creating a local record of these items the first time they register them and keeping them updated every time they have online internet connection again.

Server-side online storage will use **Azure SQL Database** since a cloud solution provides better efficiency, production-time, scalability and other metrics than an on-premises solution. Also, its structured topology makes it easy to organize the application’s information while also being able to be de-structured in SQL Views for performance. Azure SQL Database also has integrated encryption (Transparent Data Encryption a.k.a. TDE) which makes it very secure.

For storage and management of user credentials the application will use **Azure Active Directory** which provides features such as encrypted information, Multifactor Authentication and Single Sign-Ons.

* *Consequences*

The application will communicate with reliable, secure and performant database solutions to access information both online and off-line.

**Any additional frameworks or technology stacks**

* *Status*

Proposed

* *Context*

The application will need a way to deliver notifications to its users across all platforms, and accessibility features to allow the largest number of users to enjoy and be able to navigate it, thus taking into consideration features for users with special needs using the application.

* *Decision*

The application will need a team of designers which will ensure that the user interface complies with all standard accessibility requirements such as high colors contrast or proper font sizes.

For cross-platform notifications, the **Firebase Cloud Messaging** solution will be used, which will allow users across multiple operating systems to get notifications from the application in real time.

For text-to-speech capabilities, the **Expo Speech** solution will be used, which will optionally allow the user to understand the information and available actions inside the different screens of the application through audible input.

For speech recognition, the **React Native Voice** library will be used, which will optionally allow users to navigate around with voice commands.

* *Consequences*

With these solutions, the application will be able to push notifications to its users when needed independently of their operating system, and allow visually impaired.