

National Park Map ++ Design Summary

Group 22: Omar Salas, Brian Goldenberg, Harsh Gupta, Emmanuel Utomi

The National Park Map ++ Mobile Application will focus on the Model-View-Controller as the overall system design. The frontend will incorporate software, such as Flutter for multi-platform development. The backend will host a cluster of Azure servers to handle requests and subsystem communication. Lastly, the MS SQL database will be used for high volume and fast quiering of data the user can access through the mobile application.

The application could be developed using various softwares, one of the suggested tools is Flutter. It is important for developers to provide users with beautiful, consistent UI with smooth animations across all types of devices. Flutter enables frontend developers to achieve such design goals, it is a free and open source Google mobile UI framework that provides a fast and expressive way for developers to build native apps on both IOS and Android. Flutter allows fast development, has expressive and flexible UI and provides a vast library of predefined widgets. Flutter apps follow platform conventions and interface details such as scrolling, navigation, icons, fonts, etc. The design is also focused on providing users with a seamless experience throughout the app while they try to navigate the national park.

As previously mentioned, Microsoft Azure will be providing our cloud-hosting solution. Microsoft Azure is a reliable solution as their design choices provide a stable framework for large scale applications. A significant feature of our backend is the utilization of load balancing across the network. The IO load balancing system allows our cloud servers to operate at low-latency speeds, in spite of heavy usage and activity on the network. This is accomplished as when one particular endpoint accumulates traffic, data can be routed to an alternate location in order to evenly distribute the work across the network.

In terms of the database for the application, it will be using a MS SQL Server and Database. This database adds support for easy management through the Azure Portal. One major advantage for this database is the fact that it's easily scalable to the needs of the system. When the database needs more space, it scales automatically and seamlessly. Likewise, the database has the ability to scale according to the volume of queries from the backend. This gives the overall system the best database since it handles the needs of the system, small or large. Finally, the big advantage of using Azure's MS SQL Server and Database has a 99.995% uptime. So, the system can always rely on a database with nearly 0 downtime.

This is our final system design which implements the Model-View-Controller design pattern

