Group 13 Final Summary

Cecilia Avila, Eugenio Perez, Kaveesha Weerasiri, Adrian Zavala

Teddy's Park Pamphlet aims to provide a consistent, versatile, and reliable pamphlet for all national parks and/or forests in the United States. Simply enter the national park you wish to visit, and the app will download a package to your phone containing an interactive map, park specific rules and/or information, park specific survival tips should you get lost, a record of the weather forecast, and a list of volunteer groups working for the park.

The app is a one stop shop for any park needs. We wanted to make sure that the app can be used without an internet connection. The reasoning is that when one is out in the thick of these national parks, their access to the internet will be spotty at best. One goal of the app is to create something that puts different organizations into one application, reducing the complexity of having lots of different software. The National Park Service may have a thorough database of all the National Parks on their website already, but it's mobile presence leaves something to be desired. Teddy's Brochure aims to fill that niche, and further expand functionality.

The application will use an existing model view controller approach in which the model stores all of the different park information so that it can be supplied to the user, the view shows different screens, and the controller consisting of the different menus and tabs to navigate between screens such as volunteering, map, gallery, and history screens.

Because we want to increase the accessibility of the app, the app will be developed for mobile devices with iOS and Android operating systems. It should be compatible with at least Android 8.0 and iOS 13. This functionality can be achieved through the utilization of Flutter. Reliability will be prioritized over performance as this application will be run on mobile devices. The app should be able to handle 1000 customers at the end of development simultaneously.

No login will be necessary, and there will be a database that stores information about each of the different parks. We believe that in order to maximize accessibility, and appeal to a wider audience, that sidestepping any account creation is the way to go. The database should include at least 85% of the United States national parks at the end of the second release. The database is only to be writable from the server side. Although no personal information will be stored, the application will ask for the users location so that the maps will be accurate. Donations will be handled by third party services. Reviews can be placed on events and reviews. So the ability to fill out a form with some general information will be available and stored. Then employees of the park system can see the comments are legitimate. This can also stop users from posting spam. Employees can delete reviews or reach out to explorers that were not sassified for any reason.

Persistent data will be achieved through caching the user's data when the system fails so that it is there for the user when the system starts again. Sensitive data and broken access control may occur but can be avoided by transfer data by https and regression tests.

Testing will be done on operating systems greater than or equal to Android 8.0 and iOS 11 to cover most user needs. Tests include checking the accuracy of google maps when the user requests a walk through map, checking to see whether the guides and gallery function properly, checking whether the different volunteering organizations show up. There will be frequent regression testing, and test plans will have to change based on the current code base. After each sprint, testing is suggested to keep the quality of the product. There will be a lot of traffic on this system so ensuring it is very important.

The target audience ranges from young to old adults. However, no one should be excluded in this application if possible. Currency and languages should be able to change based on user preferences. The product should not be offensive to any group who may use this application. Before the first release of the product, the product should meet standard regulations in every legal way possible. This can be confirmed by a legal office and/or multiple offices. This ensures that the project will not be held back by any legal issues.

The splash screen of the application should include a list of all (ideally) 62 National Parks, along with an empty text field that the user can use to search. From the Splash screen, the user should be able to either download a cached copy of the "brochure", or see a live version on our cloud. Regardless of what they choose, when they select a national park it should take them to a screen that displays the information in an easy to read format. Whether some portions of the information should be partitioned into different tabs should be left at the discretion of the UI designer. The most important thing to realize for an application like this is that format allows for easy parsing and translation from information from the database. The UI in this scenario should be easy to read and the same for every park. The information can be easily reusable for every park as well.

A lot of the issues that may arise during development will come down to available resources. During development, the Covid-19 pandemic had many parks close down, which raises many questions as to the future of this application. The current administration has also made some cuts to the National Park Administration, so funding will definitely be an issue unless we find a third party source. That being said, we anticipate life to return to the status quo eventually, so hopefully it shouldn't impede too much when it comes time to actually develop the app.

Some other issues might stem as a result of the platform being used. Android development is often infamous for it's finicky bugs. A good solution for a real world application would be flutter. Flutter is a framework that allows functionality for iOS and Android devices. That being said, I do see this project being feasibly made in Android Studio, and I anticipate that is what most students will opt to use should they choose our project due to the popularity of CS478. There are many alternative IDEs and languages the user can use. Whether their frameworks are more stable an/or easy to use will have to be put into consideration when the time comes to start development.