GPS Trail Project Requirements Summary

Group 1: Fade Abdeljaber, Chris Lenell, Deonvell Shed, Luke Wittenkeller

The requirements of the systems are all made for the purpose to enhance the experience of the user end of the application. The product use cases vary for different types of users and subsystems inside the application. The main users such as visitors, park employees and admins will each have varying accessibility to certain functions in the system. The visitors will be able to use the main application’s features such as planning a route, sharing routes, creating and viewing user profiles, receiving system warnings and etc. While the park employees and admin include the visitor’s features and others that will handle the back end of the system to make sure information and system processes are working efficiently.

Within the main system there will be subsystems such as the gps communication, database and the warning systems. While most of these systems will be running independently of each other, in some cases the subsystems will have to work together to get certain tasks done. These tasks include but are not limited to issuing warnings to local police departments, creating backups of gps information if the system shuts down and retrieval of important gps information in case of emergency situations.

For each of the important aspects of the application such as design, dependability, security and accessibility there are individual requirements that should be met. Each of these requirements are meant to ensure that the end product is made as properly as it can for the client. While each requirement has a description and reason why it is needed for the final product. The tests that are done for each of the requirements may involve one or multiple system features. It is important that these tests make sure that the requirement is being met.

For this program, we also want to ensure privacy. This includes requirements specifying what data system administrators can access along with what our target users can see. These administrators may be able to see aggregate data concerning usage of the system, but they will not be able to see specific data about any one client. Similarly, our main users, namely park visitors, should not be able to see any data that does not directly concern them. Additionally, the system should resist malware from third-parties. Tests will be performed to check that privacy concerns are adequately addressed.

The system must be maintainable for its whole life cycle. We estimate this to be around 5 years. For this life cycle we want a support system for problems users encounter. This support would be through online and in person in national parks. The system should also be able to be scaled for an increase in usage during this life cycle. This increase can be expected to be up to 30% from its first year to its second.

Another set of requirements that were developed were under the operational and environmental category. The only environmental requirement that we needed to have deals with the possible connectivity issues. Being deep in mountains or forests does not bode well for keeping a stable internet connection. We address this issue by utilizing the user’s device storage until internet connection is re-established. The operational requirements we have for this application are linking to the Google API, how the software is going to be distributed, and how we plan to update the software. We can test the linkage of the Google API by calling it’s functions in our tester. We can assure the distribution by searching the app on The App Store on iOS and Google Play Store on Android. And finally we can do a simple test that checks that the current version of the app is compatible with the current OS version.

When developing software there are cultural, political, and legal requirements that need to be made. In our case there are only a few things to take into account. The cultural requirement is that the app be in U.S. English. The app is meant to be used for National Parks, which are only in the United States, so it is obvious that the common user will expect it to be in English. The political requirement is to give credit to Google for using their API. The main functionality of the app is to be a GPS system which is solely dependent on them. In the FAQ section of the app there will be an explanation about the partnership we have with Google and what they bring to the app. The legal requirements that need to be made are related to information privacy. The California Consumer Privacy Act (CCPA) states that users must be informed about what data we are collecting about them. This also gives the user the choice to opt-out of data collection. When creating an account the user will be given the disclaimer and can choose whether they want to be involved in the statistics or not. This can be tested by running a test with artificial data of users where some users have their data collected and others do not.

Another set of requirements that needed to be implemented was the registration system. Users needed to be given the option to register for a new account within the app in order to save their data. Additionally, the ability to create a new account needed to be simple in order to avoid turning off the user from using the app. The system was required to load the registration form within 5 seconds. Having this requirement ensures that the app isn’t held up by constant loading times.

It was important that a tutorial phase be shown to the user the first time they open the app. Enforcing the user to go through a tutorial ensures that the user is able to utilize the app most efficiently and cut off the learning curve of the app significantly. Furthermore, the user will have the option to customize the app to fit whatever theme they would like. Providing that creative control to the user provides an easier experience while using the app.

Another requirement that needed to be implemented was the voting system. The system must be able to provide the user the ability to vote for their favorite route. By implementing a voting system into the app, it allows the user to filter through the routes starting with the most popular. The user will use a polling system that is provided in each route and allowed to rate it 1-5 starts. From there, any user who wishes to filter through the popular routes are free to do so. The polling system can be tested simply by voting for a particular route and seeing the results change due to the new votes.

Additional requirements that need to be implemented is the ability to view pictures of a park. The system should only load five images at a time and the system should give the user the ability to load five more images at a time. By only loading five images at a time, the strain on the system would be a lot less resulting in faster loading times. These set of requirements can easily be tested by simply uploading images and checking if only five are loaded at a time. There will be a button that will allow the user to load five more images should the user wish to view more pictures of the park.