Travel Advisor Summary

Group 11: Ali Darawsha, Meet Patel, Joshua Gonzales, Vedant Majmudar

The application Travel Advisor is a national park navigation mobile application developed on the Flutter framework using the language Dart. The application gives users an enhanced traveling experience, by providing them with a hub of useful information regarding national parks. The information includes, but is not limited to, weather information, location information, and information regarding scenic spots within various parks located in the United States of America.

To begin, the application allows for a user to create and login to a personal account and begin using the application. The home page serves as a central hub for the application, providing users with quick access to the various functionalities available to them. The functionalities include being able to select any national park they desire, and being presented with a dialog box that provides various segues to other facets of the application. Firstly, the ability to download an offline map is available, allowing users to download a PDF version of the chosen national park. The application also allows users to navigate to the parks via Google Maps, or view the national park via the in-application map. The in-app map is filled with markers designated Scenic Spots, which are spots of interest that are recommended to the user. The spots are also loaded into the Scenic Spots page, which gives a list of the spots and a weather forecast for the national park. The Save Page allows for the user to save national parks to remember for later. The Settings Page gives the users access to a form, and an emergency button, upon pressing will send the filled information plus the users location to an online database for emergency use.

The first release for the application consisted of laying the groundwork for the application, setting up environments, and becoming familiar with the Flutter framework and Dart. After the environments were set up, the team began thinking about division of labor and the overall design of the application. The division of labor was an easy task, as the application is designed in such a way that each facet of the application can be easily independently developed. The division of labor was then decided to be that Josh would handle the Firebase database, handle the OpenWeather API, authentication features for account creation and login, and the Scenic Spots Page functionality. Meet would create the foundation of the application with the navigation bar, help with the openWeather API, provide the functionality for the Saved Spots Page, and provide the functionality for the Settings Page. Ali would provide the functionality for the Map Page, handle the GoogleMaps API, create custom map markers with custom windows, and acquire and implement Scenic Spots information. Finally, Vedant would provide the functionality for the Home Page, acquire the resources needed to Home Page visuals, and implement the Firebase upload for national parks. Many of these functionalities were not correctly implemented until release two, and will be discussed there. The second release consisted of finishing functionalities that remained from release one. The functionalities included the Favourites feature, Scenics spots feature, download offline feature, Google Markers, and Emergency feature. During this release, the dialog box for the Home Page was implemented, creating a segue to other functionalities that needed to be implemented. Finally, the visuals of the application were updated to give users the best possible experience.

This project's testing was interesting in that Flutter takes a different approach to coding than other languages the group has been used to. It is slightly like Java, but Flutter is built mainly around widgets. This being said, the navigation between Travel Advisor's functions allows for inputs in testing by way of populating the widgets with data and information for the user to see. The testing cases the team found included the population of scenic spots, Google Maps navigation, and collecting data before sending to the database. These are all essential to the application as National Parks are locations which the users will eventually need to interact with, so the application should help solidify their knowledge of how they work and how to get there. Most of these tests have passed our requirements as well as the requirements set forth by the original creators of this application's idea and were tested during and after implementation.

The group was also tasked with code inspections. This was useful as it gave everyone a chance to collaborate on how to make each person's code better in the near future of development. The functionality of each person's particular significant piece of code was mainly working as intended, so these inspections were mostly to analyze the strengths and weaknesses of the style and structure of their programming. The overall consensus was that comments would have been helpful especially as the team was new to programming in Flutter. The flow of everything inevitably made sense as each person had the team help them in some way to develop the code as there were bugs and issues which were difficult.

We have seen many functionalities implemented during the first and second release. Functionalities such as Scenic Spots, Google Markers, Emergency Contacts Form, Authentication, and Favorites. All of these functionalities were done in a limited given time where if we had more time together as a team, there would be many new features we would have liked to add to this app. For example, a search bar at the top of Homepage to quickly search national parks quickly based on their names. This would help make our app more user friendly and efficient, and this feature would benefit users more because they would get less tired. Second, we would add all the national parks that are located in the USA. Currently we only have 12 national parks, thus search bars will come in handy. Lastly, in order for our users safety, we were planning to add a call feature in the settings page where Emergency Form is located. This will be beneficial for users who find themselves in danger. I believe we have many functionalities but these are the three functionalities that fit in the app and have a purpose behind it. grey box method to test. All of the testing methods have an important role to play. White box method allows us to recollect how the function is going to work and blackbox helps us find if or any faults there are in coding.

In the end, we have accomplished our goal as a team. We decided to work on an app with a completely new language that none of us had hands on experience. However with so much time, research, communication and errors every week we have proven that if we dedicate our effort into and focus, we can accomplish any goals. Using Agile Methodology also was a bonus to us, it helped everyone keep in sync of what their responsibility was. There were moments where we would get stuck and feel hopeless, sometimes we find ourselves lost, however, in the end we pull through and fight. We were able to test the code each one of us wrote, found some errors and fixed it.