**KiliTrekker Software Requirements Specification**

Prepared by: Kyle Krueger

September 24th, 2021

**1. Introduction and Overview**

**1.1 Purpose**

The purpose of this KiliTrekker Software Requirements Specification is to describe the requirements set by the stakeholder of the software. The intention of the program will be to provide various information to users in the Mt. Kilimanjaro area. This document will give an overview of the systems within the program.

**1.2 Intended Audience and Reading Suggestions**

This document should be conversed between the development team and the stakeholder to ensure requirements are clear between the two parties. The document will use more common verbatim for the stakeholder, and use more technical terms for the development team. The reader should begin with reading the Introduction and Overview section to generate an understanding of the general system. The rest of the document will contain user requirements, system requirements, and other information.

**1.3 Project Scope and Risk**

The KiliTrekker system is meant to provide information to hikers, tourists, park rangers, park staff, and tour guide leads information about 12 of the trails leading up to Mt. Kilimanjaro. This system will provide information on how to lead a group of untrained hikers through potentially dangerous territory and provide accurate information on the local Kenyan rebel forces. Information on the rebels is high risk, because inaccurate information could lead to potential harm to human life.

**2. User Requirements**

The system will be a website that users and administrators can visit and find any information on the local park area and trails including local events, celebrations, park news, and emergency situations. The users can access the website from their smartphones or computers, while admins can access the administration page from the park computer to update safety information, local information, or information on tours. Users will be able to book tours and see a map of the trail with the distances of each of the 12 available trails. Weather will be displayed on the home page, and if there is any rebel activity, there will be information displayed on the home page as well. We will have an option to download an offline version of the information displayed on the website which will include the map of the trails, and the most recent security information.

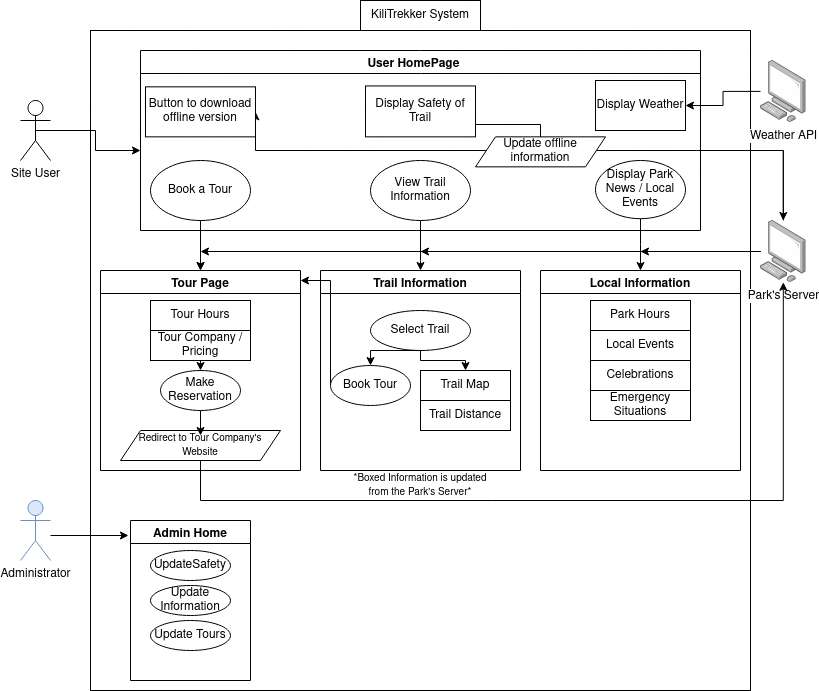
**2.1 User Interface**

The user interface will not be very crisp looking, and will be much more functional, as we are going to be focusing more on the reliability of the website rather than the appearance. If the user chooses to download the offline version of the application, we will provide some polish to the website including some images on the home page of the website of the local area and some cleaner buttons.

**3. System Requirements**

**3.1 Functional Requirements**

The functions of the system will be fairly simple, we should have a function that gathers information that was previously entered from an admin on the park server computer. The website should only require around 750kb of data or less in order to be uploaded and downloaded, for the server may run on older hardware, and may not be powerful enough to transfer information quickly.

**Figure 3.1.1 Use Case Diagram**

In figure 3.1.1, we see the use case for our users and administrators. Interactive buttons are displayed as circles, Server communication based on user activity is displayed in parallelograms, and rectangles represent information displayed based on what the admin sets on the park server.

The user will start on the homepage of the KiliTrekker website and will be shown information on the weather that is gathered from an external weather api, important safety information such as rebel activity and emergency notifications from the park server, and a button to download the offline version of the website which will then display the time it was last updated by the park server. There are 3 pages the user can go from the homepage, book a tour, view trail information, and display park news and local events.

If the user selects book a tour, information based on available tour hours will be shown to them, along with pricing of each available tour and the company that hosts the tour. When a user makes a selection of their tour, they will be redirected to the website of the tour company to complete any payment and find additional information on the exact tour.

If the user selects view trail information, they will be greeted with an interactive map where they are able to view the routes on all 12 trails that our system supports. These should show how long each trail is on the map, as well as a button to book a tour on that trail, which will redirect them to the book a tour page.

Finally, if the user selects display park news and local events, they will be redirected to a basic information page that displayed information on park hours, local events, celebrations, and any emergency situations. All of this information will be updated from the park server.

If an admin logs in, they will have the option to instead go to the admin home screen, which will add functionality to update any safety information which could include rebel activity in the area or emergency weather situations that could mark some of the trails off as closed trails, in which any tours for the trail will be marked unavailable. The admin will also update the information that is on the local information section, including adding events and celebrations or adjusting hours. The admin may also update the book a tour page, by adding or removing tour companies, inputting or removing upcoming tours, changing the links that the users are redirected to when they click on the make reservation buttons, and edit the information that is displayed on the home screen, as well as include anything for the offline versions of the app that may include images, or higher resolution maps.

**3.2 Non-functional Requirements**

The system will be hosted on the park’s server which is located at the entrance of the park. This server is fairly weak, so it is important for us to consider this when it comes to our file management and information size. We are looking for the system to be reliable enough given the restraints of the server strength, we are looking for our files to download with a 70% success rate within 4 attempts of communicating with the server. Information on the safety of the trails should be the first to be uploaded to the user, as they are the highest priority, and be updated to the offline version

**4. Definitions**

*User –* The person or person(s) interacting directly with the program. Users include hikers and tourists

*Administrator / Admin –* The person or person(s) who have access to updating information on the website