

Software Requirements Specification - Kili Trekker System

Created by: Dante Viscuso

September 23, 2021

1 Introduction and Overview

This software system is going to manage all relevant and real-time information about the 12 available trails located at the Kilimanjaro National Park. Relevant trail information includes things like weather reports, trail conditions, ongoing park events, and any rebellious or malicious activity within the park. This system will work in tandem with the already in place cellphone towers scattered across the park in order to connect all of the ranger stations to the main server. Users of this system should be able to interact with and view all information as needed throughout their time in the park. It will be the job of the park rangers and guides to keep the system's information up to date via ranger station or web-based cell phones. This system is being implemented with visitors, hikers, and tourists in mind in order to provide a smooth experience through the trail information available as well as a safe experience through the various alerts whenever there might be false information or rebel activity. This document is meant to outline the various requirements that must be considered in order to create this system. It is laid out in various sections such as the user requirements, systems requirements, and other relevant information and it will contain information about how this system works, the features it will include, examples of how different users might interact with this system as well as an overview on how this system will be created.

2 User Requirements

There are quite a bit of different user requirements that are necessary to make his system complete. The reason being is that first off, it needs to be an interactive system, and this leads to many different options and requirements in terms of your average user. Having the information being up to date is the backbone of this system, so one of the requirements includes that rangers and guides have access to go in and change any information as they see fit. In terms of the information itself that the system holds, it must be able to support all 12 trails within the park, there must be webpage support to view his information to plan trips, and all of it must be in English. Furthermore, while viewing the information, the user must have in-depth access to each category of information. For example, when the user wants to view the weather report, they must also have access to the camera view on top of the mountain, or if they wish to view the ongoing events, they must have access to things like trail conditions or emergencies going on in the park. Lastly, anyone and everyone within the park will have access to viewing this system, but it is crucial that only the rangers and guide have the ability to change said info.

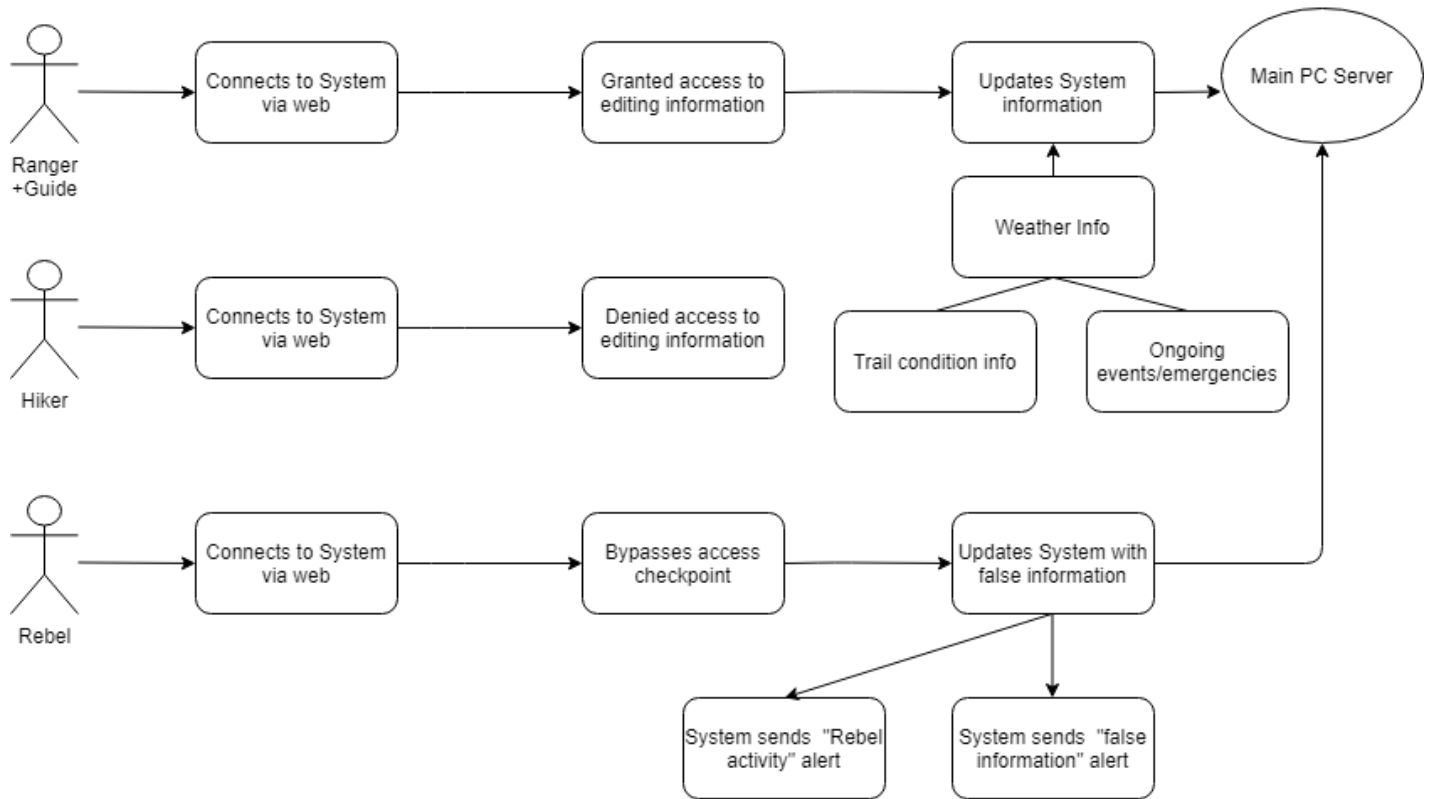
3 System Requirements

3.1 Functional Requirements

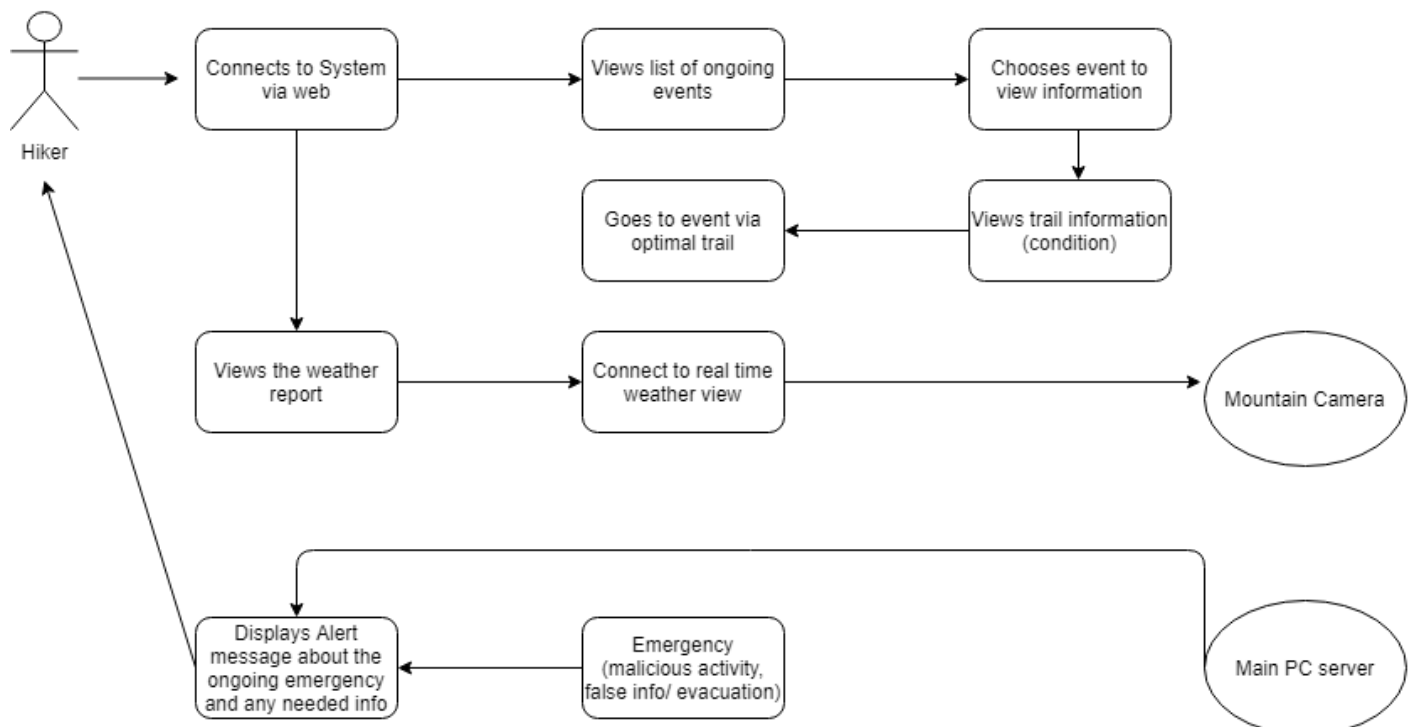
In essence, this system will function in 2 types of manners. It displays the information of the trails such as weather, events in the park, or emergencies which is consumed and used by anyone and everyone within the park. It must also require that the information is able to be updated, which is done solely by the park ranger and guide. The following diagram will outline the possible ways in which these functions play out depending on the person and the function.

Kili Trekker System: Use Cases

Case 1: Updating the Systems Information



Case 2: Viewing the Systems Information



Use case 1 describes the various ways that you would interact with the systems in terms of updating the information. First, you have the ranger and guide who are the intended actors in this case, and they connect to the webpage of the system, are granted access to editing the information and, then go on to update the system which sends said information to the main server. Next, you have your average park-goer/hiker who may try and connect to the webpage and edit the information but are denied access since only the ranger and guides have this permission. Lastly, we have the expected malicious route in which they connect to the webpage of the system, but this time bypass the access checkpoint and go on to update the system with false information which gets sent to the main server. This will result in the system sending out alerts about false information as well as rebellious activity wherever this occurred at.

Use case 2 describes how someone would go about viewing the data that is provided by this system. This use case is much smaller only because this function is very straightforward and allowed to be performed by anyone and everyone. It starts with connecting to the system itself via the web and then some examples include viewing all the ongoing events and choosing one to go to, which will then display said events information and everything needed to get there including trail condition and this person will go ahead and go to the event the optimal way. Another scenario includes connecting to the system in order to view the weather report information and even connecting to the real-time view which links with the camera atop the mountain. The last event outlined in the viewing portion is the case in which an emergency is going on. Whether that be malicious activity, false information, or an evacuation the system then sends an alert to be viewed by the hiker/ranger/guide.

3.2 Non-functional Requirements

Seeing as how rebel activity is extremely prevalent within the area that this system is being implemented, the majority of the nonfunctional requirements revolve around security/safety and establishing the most secure system possible. For example, one of the requirements is to enforce security against rebel and malicious attacks on the system. This is to prevent rebels from not only accessing the information itself within the system but also shutting the entire system down from the inside. Another requirement that goes hand in hand with the previous is allowing access only to park rangers and guides so that these types of attacks are far less likely, although not impossible. A requirement that falls out of the scope of security but is just as important is about usability. The environment in which this system is established can be very harsh, so it is required that this system works under extreme weather conditions. This would allow for a much better uptime as well as more coverage on weather-based emergency situations.

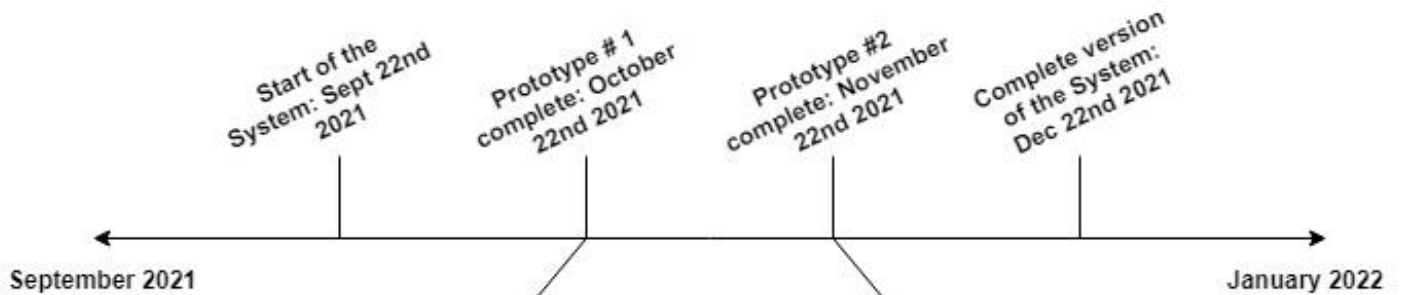
4 Other

There is one main requirement that falls into the other category and it has to deal with everything that revolves around the timeline of this system. Things like how much is the budget, team size, dates and, how the prototypes will look and act. First and foremost, the budget of this project is going to be 40\$, this should cover things like prototyping, any and all meetings, and pay for designers. For the team size, we wanted to keep it small in order to prevent butting heads and clashing ideas, so the team size is 2. This should be perfect as you would have one person design the prototypes/system and one person build it. As for deadlines, the first one will be one month from now in which you will need the first basic prototype. This prototype will only have basic functions that will be controlled by the main computer. The second deadline will be two months

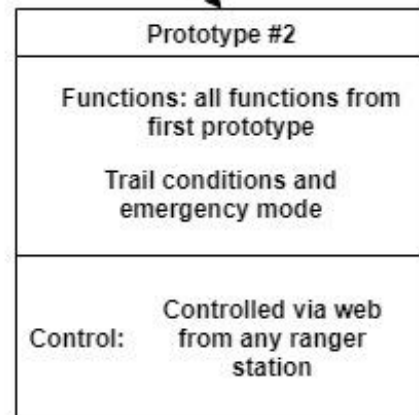
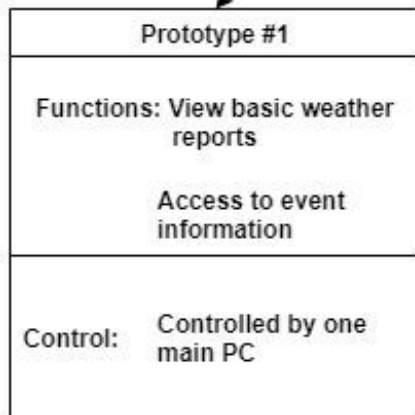
from now, where you will need a prototype with all of the functions along with web access from each and every ranger station.

Creating the System: Overview

1. Timeline

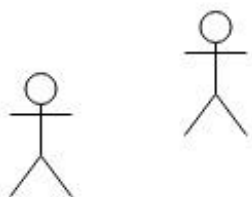


2. Prototype desc.



3. Other info

Team size: 2



Budget



Approx. time

