Kilimanjaro Trekker System

03.01.2023

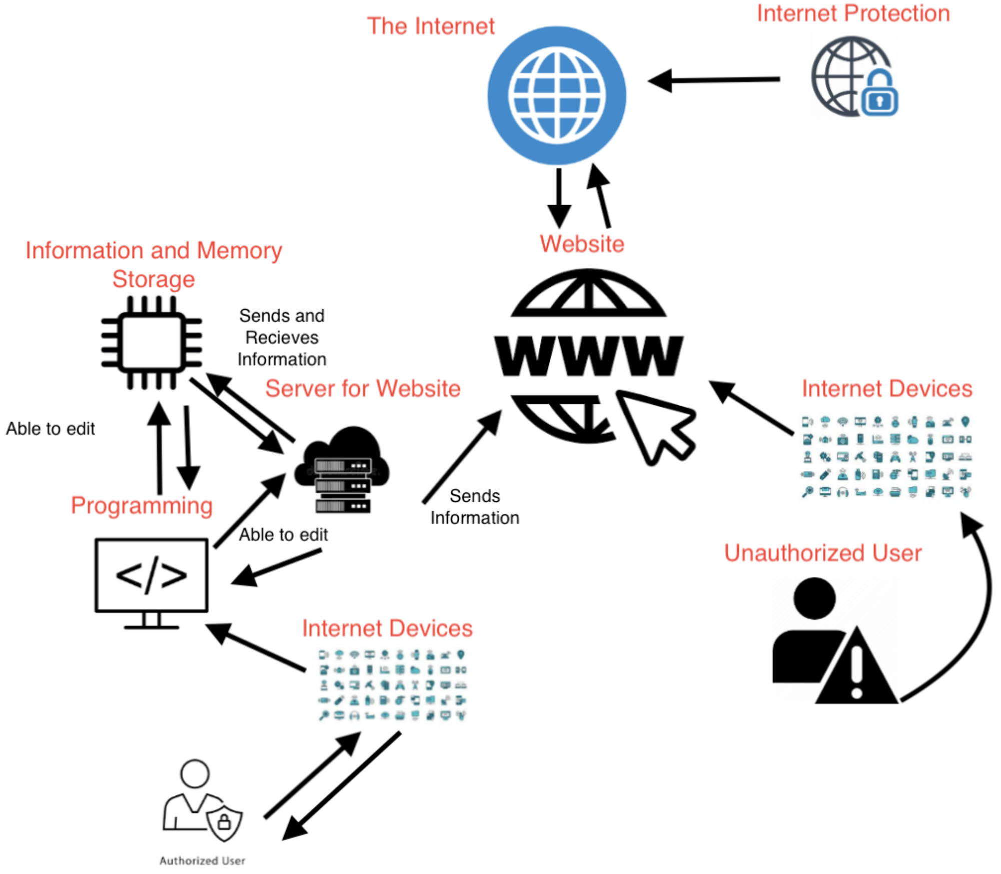
Steven Gervacio, Jason Lam, Tri Pham
CS 250
Dr. Hanna

System Description

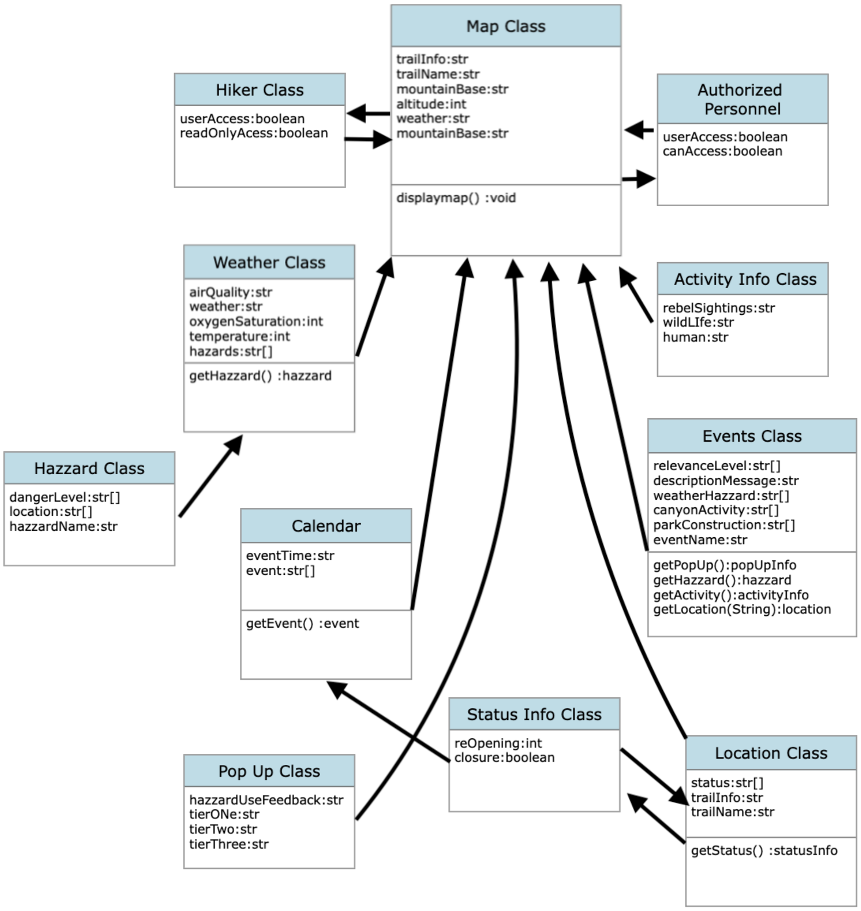
The Kilimanjaro Trekker system is a tracking software that is used to help with Tanzanian tourist business in Kilimanjaro. This system includes every information within the Kilimanjaro area including weather and upcoming events. It is designed to help planning trips to Kilimanjaro from Tanzania easier. This is a system to monitor the 12 trails that are located in the National Park. This will help keep the tourist safe and allow the park rangers to take precautions on the park. This system will provide details on the architecture and how the plan will be executed.

Software Architecture Overview

Architectural Diagram



UML Class Diagram



Description of Diagrams:

Architecture Diagram:

This architecture diagram that is provided is what we made for the Kilimanjaro Trekker System. In this diagram we provided what it will look like from a front end and back end perspective which follows the standard website architecture. The main part of this system is the website in which all of the information is displayed. We protected the website by adding a firewall. The diagram then splits into two boxes in which one is accessed to edit and other to be viewed only. Following the editable access there will be functions that include where the database is stored, where information is transferred, and the updating aspect. All of these functions can be accessed when a user is on the website. This type of access can only be accessed by an authorized user while an unauthorized user can only have access to being able to view the information.

UML Class Diagram:

This UML class diagram is what we made for the Kilimanjaro Trekker System. In the diagram we display all of the classes that will be used in the software. We provide a map of the class in what each box represents and is used for. The diagram shows the functions, attributes, and operations. It provides functions that we label as strings and return types. The classes show the purpose of the class and what it is made for. Classes like the calendar, we store the event as a string and then return it by getting the event. All of the classes listed are what we use for the information to make the website running.

Description of Classes

The map class provides details on the information regarding the Kilimanjaro area. The calendar class has the date for upcoming events and storing data. The location class has the place for the events and provides information about the trails. The events class has information about the events and information about things like weather. The weather class describes the current weather and trail conditions. The hazard class describes if the current weather and trails conditions are in danger level or not to provide safety. The status info class is used for the time for the event to open or not depending on the date and situation. The activity info class contains information about any sightings of rebels and Kenyan trails to deal with emergencies. The authorized personnel class provides information on the staff and guides have full access to the systems including write and delete. The hiker class has customers that can only use the read function. All of these classes have their own value that they add to the system to make it a complete system.

Description of Attributes and Operation

The map class will contain different attributes such that all classes will lead back to the main class. Such classes include the calendar one which will contain values such as the times and dates which will be stored as a string and communicated through the map class. The operation of the system will include the communication between the internet service, authorized users, and internet devices in which will utilize each of the classes to determine what output is needed for the map and the trekking system. Functions like temperature and air quality will be used through the program to determine what is outputted through the system. There will be functional requirements such as maintenance and web accessibility that will determine what will be outputted as well as non functioning ones such as unauthorized access security measures. Throughout our software we will be implementing strings, booleans, popUps and other data types that will be stored in the various classes.

Development Plan and Timeline

I. Partitioning of Tasks

Steven Gervacio- Github repository, Architectural Diagram, Description

Jason Lam- UML Class Diagram, Description

Tri Pham- Overview of System, Description

II. Team Member Responsibilities

Steven- Layout of System, Description of Attributes and Operations

Jason Lam- Description of Classes, Diagrams

Tri Pham- Description of Overview