

# **ECE 413/513 Project - UVFit**

**Checking and Demo Due::** Tuesday, November 13, 11:59 PM

**Final Project Due::** Wednesday, December 05, 11:59 PM

## **UVFit Project Overview**

- The UVFit application is an IoT enabled web application for monitoring outdoor fitness activities and the amount of sun exposure a user receives during those activities.
- Users will carry a device with a GPS and a UV sensor to monitor user-defined activities and sun exposure. When a user starts an activity, the device will periodically collect the GPS location, speed, and measured UV exposure.
- When the activity is stopped, the collected activity is transmitted to a web server that maintains the history of each activity for the user.
- The UVFit web application will also allow the user to monitor their fitness activities, UV exposure, and calories burned. The web application should use responsive design to allow users to view the web application on mobile devices.

## **Due Dates and Submission Requirements**

### **Project Checkin and Demo (Nov 13, 11:59 PM)**

- Setup a web server that allows users to create accounts, register a UVFit device, and view basic data collected from the device.
- Submit a video (10 minutes maximum) that demonstrates that your IoT device can log multiple GPS location, GPS speed, and UV readings and transmit that data from the device to your web server, store that data in the database, and have a front-end webpage for viewing that data.
- Submit code and documentation for at least one working endpoint on your server that can be accessed by the instructor and TAs.

### **Final Project Submission, Pitch, and Demo (Dec 05, 11:59 PM)**

- All code for the project, including the device, server, front-end web pages, README, etc., must be submitted as a single archive to the D2L assignment drop box.
- The README should include a link to your server, login credentials for an existing user account with recent collected data, and links to the pitch and demonstration videos.
- Create a 5 minute video that pitches the UVFit project to potential investors (e.g., angel investors, VCs, kickstarter, etc.).
  - See <https://kickstarterguide.com/2012/06/13/examples-of-great-pitch-videos/> for examples of good pitch videos.
- Create a 20 minute video demonstrating your project implementation. This video should demonstrate both the user experience and discuss the actual implementation. The suggestion is to split the video into 10 minutes for the user experience and 10 minutes to discuss your code implementations. This video will be used during the grading of your project, and should mention how you addressed each requirement.
- The pitch video and demonstration video should either be hosted on your server or hosted on some other publicly accessible service (e.g., YouTube)
- Submit the URL of your server, which must be accessible until at least December 14. You will need to provide login credential for a user for our testing.
- The project submission is officially due Wednesday, December 05, but the D2L assignment drop box will remain open until December 11, 11:59pm. You may submit up until this date/time with no penalty.

## Requirements and Clarifications

- Some of the project requirements are open-ended allowing individual groups to be creative in how the requirement is implemented. Intentionally open-ended requirements are underlined below.
- If any requirement is not clear, you should post publicly on Piazza to clarify the requirements.
- You are also responsible for monitoring Piazza for any project requirements clarifications.

## Requirements (ECE 413 and 513)

- **Account Creation and Management**
  - A user must be able to create an account, using an email address as the username and a strong password, and register at least one device with their account.
  - A user should be able to update any of their account information.
  - A user should be able to replace a device with a new device in their account.
- **UVFit IoT Device**
  - The user should use a button on the IoT device to start and stop activities.
  - The IoT device should use the onboard LEDs to indicate the activity status.
  - During an activity, the IoT device should periodically record the GPS location, speed, and UV exposure, at least at a rate of once per second.
  - If the UV exposure during an activity exceeds a user defined threshold (the threshold will be set in the web application), an alert should be provided on the IoT device.
  - After an activity is complete, the IoT device should transmit the activity data to the server.
  - If the WiFi connection is not available, the IoT device should locally store the data for up to 24 hours and submit the data when later connected.
  - The server should require an APIKEY from the IoT device for posting activity data.
- **Web Application**
  - The web application should support the following views (each of which should be visually appealing):
    - A summary view showing the user's total activity duration, total calories burned, and total UV exposure in the past 7 days.
    - An activities list view that lists all fitness activities with data summarizing each activity (e.g., date of activity, duration of activity, calories burned, UV exposure).
    - An activity view for a selected activity that will display the activity date, duration, UV exposure, activity type, calories burned, and plot the user's route for the activity on a map.
  - In the activity view, the web application should allow a user to change the activity type. Activity types should minimally support walking, running, and biking.
  - The calculation of calories burned should be based on the activity type for each activity.
  - The web application should allow the user to define a UV threshold that defines the total UV exposure beyond which the IoT device should alert the user during an activity.
  - The web application should have a navigation menu.
  - The web application should use responsive design to be viewable on desktops, tablets, and smartphones.
- **Server**
  - When an activity is sent to the server by the IoT device, the server should assign a default activity type based on the speed data.

- Your server must be implemented using Node.js, Express, and MongoDB.
- Your server's endpoints must use RESTful APIs. Each endpoint must have accompanying documentation that describes the behavior, the expected parameters, and responses.
- Access to the web application should be controlled using token based authentication.
- **Third Party APIs and Libraries**
  - You may use additional third party APIs.
  - You may use open source JavaScript libraries.
  - You may use open source CSS libraries.

### **Requirements (ECE 513 Only)**

- The IoT device should have an auto-pause feature: During an activity if the user's speed is 0 for more than 2 seconds, the device should pause the activity data recording until the user starts moving again.
- The web application should have two additional views:
  - A local data view showing the average activity data of all *geographically* local users in the past 7 days. Data should minimally include number of activities, average total distance, average total calories burned, and average UV exposure. How geographically local is defined will be left to your group to determine.
  - An all-users views showing the average activity data across *all* UVFit users in the past 7 days. Data should minimally include number of activities, average total distance, average total calories burned, and average UV exposure.
  - Note: You will need to create additional users accounts and simulate data of their activities in other locations to test and demonstrate the geographically local and all users summary views.
- The web application should include a weather forecast including UV index for at least the next three days. Your server should use a third-party API to acquire this information.
- Your server must use HTTPS.
- The web application must use email verification when a user creates an account. If not verified within 1 hour, the account should be disabled.

### **Extra Credit (ECE 413 and 513)**

- Support account creation and authentication using a Google account and OAuth