Winter 2016 Contest Software Requirements

Software Requirements Introduction

You may notice that this list may contain more or less requirements then your team submitted as part of your contest entry. This was done for two significant reasons.

The first is to somewhat level the playing field. We are taking into account your questions and your proposed requirements, and those teams who have asked the best questions should have gotten a head start on implementing their solution. However, we want all teams involved to be oriented to the same end goal, which is why we have created a single set of requirements for each team to work through.

A Quick side note from a learning standpoint: In the “real world” the team or company who asks the best questions, and figures out what the customer really wants first will usually have a head start on its competitors.

The second reason was to bound scope. We know from our own experience that an unbounded software solution can quickly overrun both time and money. We have narrowed down the required components so that each team can complete a solution in the time allowed. This isn’t to say that features not on our list will not add value, but they will be evaluated as a differentiator as compared to the must have requirements.

Software Requirements

1. The solution shall handle multiple simultaneous GPS tracked packages sending updates.
2. The solution shall be easily accessible from a Windows 7 computer.
3. The solution shall support an admin mode that shows all package location updates on a map.
4. The solution shall support a user mode that shows a subset of package location updates on a map.
5. The solution shall accept a list of UUIDs in user mode to control the subset of package location updates displayed on the map.
6. The solution shall accept name, destination, and GPS unit UUID information as HTTP query parameters on a HTTP GET of the URL path "/tracknewpackage". An example follows:
   GET
   http://127.0.0.1:8080/tracknewpackage?name=Some+Name+Here&destinationLat=42.4877185
   &destinationLon=-71.8249125&uuid=b0f9bb21-160f-4089-ad1c-56ae8b2d5c93

7. The solution shall respond with a JSON encoded body which includes the registered uuid on an HTTP GET of the URL path "/tracknewpackage". An example follows:
   GET Response Body: { "ackUUID":"[b0f9bb21-160f-4089-ad1c-56ae8b2d5c93]" }

8. The solution shall accept a JSON encoded body which includes location, elevation, and time on a HTTP POST to the URL path "/packagetrackupdate/". An example follows:
   POST http://127.0.0.1:8080/packagetrackupdate/b0f9bb21-160f-4089-ad1c-56ae8b2d5c93
   POST Body: {"lat":"42.4879714","lon":"-71.8250924","ele":"195.9","time":"2015-12-08T08:42:33.188-05:00"}

9. The solution shall accept a JSON encoded body which includes a delivered flag on a HTTP POST to the URL path "/packagetrackupdate/". An example follows:
   POST http://127.0.0.1:8080/packagetrackupdate/b0f9bb21-160f-4089-ad1c-56ae8b2d5c93
   POST Body: {"delivered":"true"}

10. The solution shall calculate and display distance to destination.

11. The solution shall calculate and display estimated arrival time.