Solution Proposition

Visio Team 1 (EiBacon)
Author: Thomas Blank, Geoffrey van Driessel, Youri Dudock, Gerbrecht Ghijsels
Kostis Thanos, Gabriël Wiegel

Version: 1.0
Date: 20-09-2016

Index

Introduction

Visio has an idea for a mobile app for the visually impaired using beacons. The app scans the area for beacons within range and should show text, images and/or a video on their mobile device. The mobile device may also play audio. The mobile application must also be able to present a route from point to point for a user. Visio needs a back-end configuration application and REST API from which the data will be retrieved from by the mobile devices. In this document we will describe our solution to Visio's problem.

Features

This chapter describes the features we will develop. Each feature will have a number, this way it will be easier to reference a feature.

1. Web portal

We will create a web portal where administrators can login and manage the beacons that are located at certain places in cities. The administrator may edit beacon information such as the display text that's being sent to the mobile application when the mobile application requests beacon data. There will also be a section where the administrator can view and manage registered users of the application.

The web portal will also contain functions for administrator tasks such as the managing of administrator accounts and managing certain app & portal configuration. This part of the web portal will be visible for so called ‘super’ admins that have more control than regular admins.

2. API

The API will create a bridge between the mobile application and our server. The API will perform actions such as inserting user data from the mobile application into our database. The API will also read data from our database and display this information to our mobile application.

Our API will make use of a text-based format called JSON. Using JSON we can send objects between the API and the mobile application. The mobile application will visit a link that is being hosted by our server and the API will return JSON output. A example of such a link would be ‘www.visio.com/api/users?user=1’. This link would return all the data of the registered user with id 1.

3. Beacon danger levels

Beacons can be assigned danger levels (for example 1 to 5, higher levels are more dangerous). These levels may be used when planning a route for a user by avoiding beacons with higher danger levels. (See feature 5. Route planning, solution 1)

3.1 Dangerous beacons in radius

The danger level will also be used by the API for a special feature. The API can return a list of dangerous beacons in a certain radius, for example: return all beacons with danger levels 4 and 5 in a 10 km radius from a certain gps location. This way the mobile application can send its gps location every ~15 minutes(depending on the radius that will be used) to the API, then the API will return the list with dangerous beacons, the mobile application will store this list in its cache. This list contains the beacon identification and its data. This way the user will be alerted instantly by the mobile application when he/she encounters a dangerous beacon, this has the advantage that the beacon data won’t have to be retrieved from the API. This will guarantee a fast response time when it’s needed.

4. Beacon Identification

The information we gathered from the presentation has confused us a bit about the identification(major, minor and id together) of the beacons. One interpretation was that a beacon could have the same identification as a different beacon. A different interpretation was that a beacon could have the same major as a different beacon but not the same identification: a beacon has an unique identification. Because we are not sure which way the beacon works we have two solution propositions.

Solution one

The first solution assumes that the major, minor and the id are unique together. This means that each beacon has unique identification: each beacon can be identificated on its own. This solution has the advantage that a gps location can be attached to a beacon.

Solution two

The second solution assumes that a beacon can have the same major, minor and id. This way beacons can have the same identification, this has the advantage that multiple beacons can be attached to one beacon-data entity. Example: 100 beacons with the same identification, is one entry in the database with one identification attached to one beacon-data entity.

5. Routes

We also have two solution propositions regarding routes.

Solution one:

The first solution is dynamic routing, this means that the route will be generated using specific user information and the danger levels of the beacons (See feature 3.1 Dangerous beacons in radius). The route will be generated with the following information: Origin, destination, rush level and Vision impaired user type.

The route will be from origin to destination, google maps will generate this route. But we will filter the route according to the Vision impaired user type and rush level. The user will have to enter the origin, destination and the rush level. The rush level will be on a scale from 1 to 5. According by the entered rush level the route will be filtered. Example: the user selects level 5 on the rush scale, then the api will generate the fastest route and it will pass by dangerous beacons. If the user selects level 1 on the rush scale the API will generate a safe route by avoiding dangerous beacons, thus generating a slower route but safer route.

The Vision impaired user type will be obtained through the API, the database records this information for each user, the user will enter this information with first use of the app.

The route will also be filtered by Vision impaired user type, example: Blind people won’t pass by a beacon with a ”NOTFORBLINDPEOPLE” tag.

Solution two:

Administrators can make static routes through the web panel. He can select and add the desired beacons for the route using the google maps API. He can also add specific text (important instructions or information) to every beacon. After this the administrator can publish the route or assign it to a specific user.