

## **E-beacon: a web-based survey control finder application**

Ojima I. Apeh, Okwuchukwu C. Nwaka, Raphael I. Ndukwu, Ikechukwu Maduako, Olatomiwa Arilewola

Department of Geoinformatics & Surveying, University of Nigeria, Enugu Campus, Nigeria  
Tel:+234-803 040 3992, ojima.apah@unn.edu.ng, apehisaac@yahoo.com

DOI: <http://dx.doi.org/10.4314/sajg.v8i2.5>

### **Abstract**

*The interplay between Geographical Information System (GIS) and Computer Science has continued to yield improved methods of carrying out many surveying-related activities. In the past, survey control points were stored in file systems and at the best in Database Management applications thereby leading to the limited usage of the survey control points since they are difficult to locate in the field. This study however, suggests another approach for the storage of these survey control points which makes them to be easily accessible and gives room for faster update and geo-visualization of the survey control points. This was achieved by means of web programming applications such as Node-JS, Leaflet Javascript Mapping API, MONGODB, HTML and CSS, integrating GIS into web technologies. The end product is an interactive web application that can be accessed using any smart device with the control points rendered on the user interface. The Survey Control Finder application (E-Beacon) is a WebGIS platform for the geovisualisation of survey control stations in the University of Nigeria, Enugu campus (UNEC), Nigeria. This study gives insight into what the latest trend in technology has in store for surveying and the need to arouse the interest of learners in web application programming in order to develop more surveying-related applications.*

### **1. Introduction**

Technological advancement has greatly influenced so many activities of man, ranging from the usual selling and buying of products, sending of information from one end to another as well as the way assets are kept and managed. It is true that many older ways of doing things have been improved in this era in that most of those activities can now be carried out electronically. The improvements caused by the trend in technology have therefore led to the change in the names of several human activities by the addition of the letter “E”. Therefore, as of today, we now have E-Banking, E-commerce, E-business and so on.

The 21<sup>st</sup> century can actually be regarded as the E-era because virtually everything can now be done electronically and this has resulted in the change of modes of operations of many organisations to what can be referred to as the E-format which entails electronic, digital and internet-based activities. This study is focused on the application of the “E” to enhance Surveying operations such as storage, location,

accessibility, management, analysis and usage of survey control points. In surveying, new surveys are required to be connected to previously established control points (points of relatively precise measurements established by triangulation, trilateration, traversing, intersection, resection, levelling used to determine the position of features beneath, on or above the earth surface) or establishing new control points before carrying out survey operations. This is to maintain a framework and a network of control points that will enhance the practice of the profession and delivery of quality services to clients.

In developing this application, GIS (which entails hardware, software, spatial data, people and methods) was integrated with Web technologies (which entail markup languages, programming languages, web servers, databases and others) (Alphadevx, 2018). These (GIS and web technologies) are the basis for Location-Based Services (LBS) as discussed by several authors (IAMAI, 2008; Amit & Vineet, 2011; Manar & Aruipam, 2012; Pankti *et al.*, 2014; Omar *et al.*, 2014; Hu & Lee, 2004; Jegadeesnari & Parameswaran, 2014). Location-based services (LBS), according to IAMAI (2008) refer to a set of applications that uses the knowledge of the geographical location of a mobile device that has location aware capability and then provides services based on the information retrieved from the mobile device.

Similar to this research are the City of Orlando Control Network application (City of Orlando Control Network, 2018), the Google Store Locator application (Google Maps APIs, 2018) and Survey Control Finder (Survey Control Finder, 2018). The City of Orlando Control Network application was developed using the ArcGIS web app builder and also uses ArcGIS Javascript Map API (Application Programming Interface) as the backbone for interactivity, HTML 5 (HyperText Markup Language 5) for the user interface design and the ESRI world street base map on which all the control points are displayed. The Google Store Locator was developed using the Google JavaScript map API as the backbone. It has several functionalities which include giving the user direction to a particular store on the map, the shortest route to take and the user can also filter results to ensure that they find what they need. Survey Control Finder is a mobile-friendly web application that provides a central point of access to over 28,000 control points and over 200,000 Public Land Survey System records in Wisconsin. It uses Leaflet API (Application Programming Interface) as the backbone for interactivity.

This study presents a web-based GIS finder application (E-Beacon) for locating passive survey control points. The application enables the storage, location, accessibility, management, analysis and search of survey control points in an area based on the location of the user.

## 2. Materials and Method

### 2.1. Data

Data is one of the most important components of any Web-based GIS application. For this study, Survey Ground control data were obtained from the Department of Geoinformatics and Surveying, University of Nigeria, Enugu campus (UNEC). The data contains many survey/geodetic control points situated within UNEC. The survey/geodetic control points (as shown in Table 1) are in geographic coordinate system (WGS84) and in the UTM (Universal Transverse Mercator) projection. They are in UTM zone 32 and not localized to Nigerian Local grid. In the process of using Quantum GIS software to convert these survey controls points to Geojson format for use with the LEAFLET JAVASCRIPT API, they were automatically converted to longitudes and latitudes in decimal degrees. See Table 1 for an excerpt.

Table 1. An excerpt from control points database

Station ID	Eastings (m)	Northings (m)	Heights (m)
NIO5	334781.511	710783.718	225.829
NIO4	334739.746	710703.924	228.4
KIN01	334778.438	710789.915	225.348
KIN02	334974.631	710711.61	224.467
KIN03	335197.309	710621.292	218.851
KIN04	335104.571	710470.083	225.27
KIN05	335015.115	710326.92	232.06
KIN06	334986.136	710192.22	234.459
DPR772/773_3	334898.858	710121.708	232.931
KIN07	334775.455	710238.925	233.203
KIN08	334564.051	710242.093	228.703
KIN09	334487.648	710277.506	232.459
KIN10	334519.362	710385.742	238.843
KIN11	334551.022	710485.641	235.955
KIN12	334655.723	710629.803	230.812

### 2.2. Application Design and Architecture

To achieve the main goal of the study as earlier stated, system analysis and requirements specification were performed according to user's everyday needs. This was achieved with interviews and assessment, in order to specify the application in terms of its strategic objective and required functionalities, design and usability.

The application architecture is presented in Fig.1. It is a three-tier architecture entirely based on Open source technology and uses a set of powerful JavaScript framework - Model View Control (MVC)

framework and libraries. This Open source solution allows the development and deployment of truly interoperable and fully functional Web GIS application at no annual license cost, which is an enormous advantage. On the MVC framework, there is a complete separation between the server-side and the client-side logic. The data model is implemented in the server side, and the views and the controller are implemented in the client side to build more organized and maintainable code.

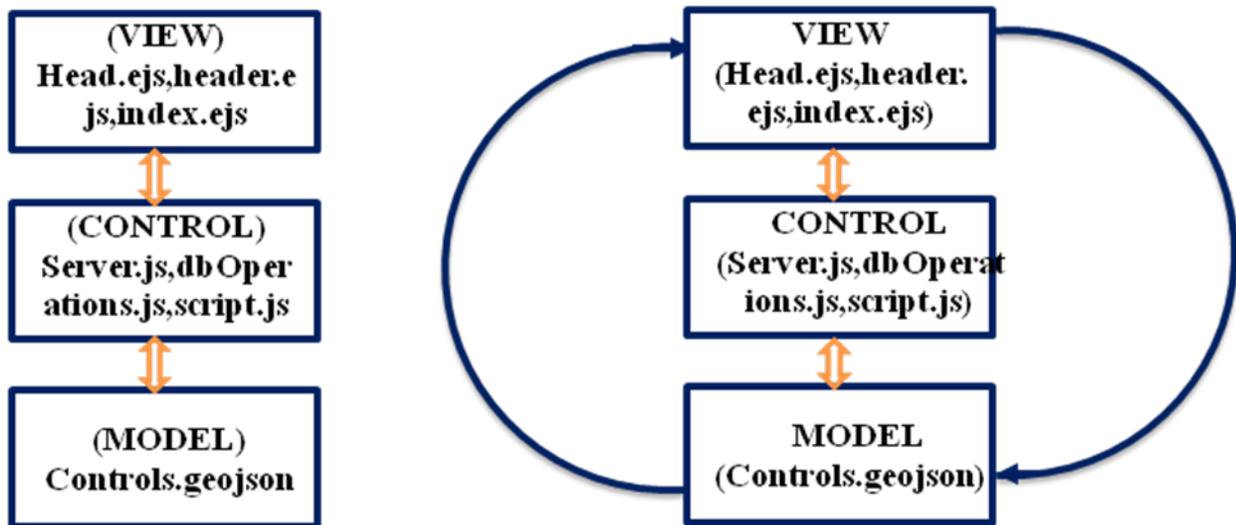


Figure 1. Application Architecture

## 2.3. Implementation

### 2.3.1. Development Environment Setup

To implement E-Beacon application or the components of the application, a development environment was first setup by installing Atom development environment and Node.js Server with all the required dependencies especially Express and EJS template engine. Node.js is a software platform which helps to build asynchronous and event - driven network applications. It contains http server libraries which allow developers to create their own web server and build highly scalable web applications on top of it. Node.js introduces the power of JavaScript on the Server side allowing the developers to create both Server-Side and Client-Side logic in one single language "JavaScript". Node.js Server uses the Express framework which serves as a solid starting point, provides a robust set of features, middleware and methods. It is shipped with a directory generator which provides a basic directory structure for a web application. For this application (E-Beacon), the directory structure of the server is shown or illustrated in Fig. 2.



Figure 2. Directory Structure of the Application

The Main logic happens in package.json which defines the whole server, includes important dependences and connects routes. The route folder describes different endpoints. The endpoints in route folder use helpers from utility folder. The Model folder consists of mongoose models. Mongoose is an ORM (Object Relational Mapping) wrapper for talking to MongoDB. MongoDB is very flexible and does not enforce the developer to have a database scheme defined. MongoDB allows changing the database schemer on the fly, by adding new keys without any over head.

1. The */assets* folder is the folder containing the front-end framework, i.e. the SEMANTIC UI (SEMANTIC UI, 2018), files used in this research and it contains some sub-folders as well as other files within each sub-folder.
2. The */lib* folder is the folder containing the mapping API which is the Leaflet.js and it contains all the files needed to ensure that Leaflet runs on the desktop.
3. The */view* folder contain two sub-folders which are the pages and the partials sub-folders. This view folder takes care of the structuring of the front-end display to the user i.e. the HTML pages. It is the templating engine folder.
4. The *dbOperations.js* is the file that housed the requests to the MongoDB database and it is the file containing the codes handling the database operations.
5. The *server.js* file within the project directory contains the server setup and the database configurations together with the end-points and handlers.

### 2.3.2. Database tier

One of the most important parts of any application is the database component. It is essential to understand how data will be handled, saved and organized before starting to build an application. Following the user requirements and in accordance with the non-functional requirements of the client,

E-Beacon application database was implemented using MongoDB Server version 5.1 and Mongoose. MongoDB is an open-source, which stores its data in documents. If well designed, it provides swift way of retrieving data. The data in MongoDB is stored in Json format, therefore, once retrieved, it is ready to use with no need of conversion and mapping between formats and structures. The format in which the MongoDB stores the data is called BSON, which stands for binary JSON. Since JSON is the JavaScript way of storing data, MongoDB works perfectly with the applications built with JavaScript framework. MongoDB, like MySQL uses unique identifier (primary key) for each document so that it is easy to query and find the data. MongoDB supports insert, query, update, and delete operations like any other databases.

One of the features that makes MongoDB stand out against the traditional databases is the inclusion of dynamic schema. Collections in MongoDB have different schema and the documents within the same collection can have as many different schema and shapes as required. This feature enables developers to start storing data in the database without any consideration of the database structural design. The database schema of E-Beacon application is presented in Fig. 3.

```
5 //mongodb://127.0.0.1/new-database
6 mongoose.connect("mongodb://127.0.0.1/new-database", function(err) {
7   if(err){
8     console.log(err);
9   }else {
10    console.log("connection successfull");
11  };
12 });
13
14 //create the schema
15 var dataSchema = mongoose.Schema({
16   type:{type:String},
17   "geometry":{
18     type:{type:String},
19     "coordinates":{
20       "type":[Number],
21       "index":"2dsphere",
22       "required":true
23     }
24   },
25   "properties":{"type":"object"}
26 },{minimize:false});
27
28 //create the model
29 var controlPoint = mongoose.model("controls", dataSchema, "controls");
30 module.exports = {
31   controlPoint
32 }
```

Figure 3. Database schema of E-Beacon application

In order to populate and host E-beacon database online, GEOJSON (Geographic Javascript Object Notation), an open standard format designed for representing simple geographical features along with their non-spatial attributes, was used to convert control data to Geojson format in QGIS and subsequently migrated to the online Mongo database version; MLAB website (mLab, 2018) .

### *2.3.3. Logic/Business Tier*

On the logic tier, Node.js (Node.js, 2018), a server and server-side scripting web development component was used as a server side scripting language for development of dynamic web pages. The server-side function of Node.js was augmented with Express, mongoose, Geojson, Ejs and MapBox JavaScript libraries dependencies. Node.js is a free and Open Source JavaScript library and provides the necessary server functionality for complex query execution on the application database and the development of routing, search and update function of E-Beacon application. Once a request or requests arrive, Node.js server connects to Node.js Server script files, which with the help of other dependencies/libraries) then decides whether to start a database process, a search process or routing process using MapBox or any other service present on the server. The geospatial data or model of E-Beacon application is stored in MongoDB and managed by mongoose and Geojson.

### *2.3.4. Client-Side Implementation*

HTML handled the arrangement of the various contents in designing the user interface of the E-Beacon application. CSS 3 (Cascading Style Sheet) handled the aspect of making the user interface attractive and other front-end frameworks were used to ensure that the best result is obtained. HTML5 Geolocation API was used for the geo-location features since it can work in any browser and can thereby work with any smart device.

Looking at the various web mapping API's available for use and also considering their functionalities, Leaflet JavaScript mapping API 1.2 (Leaflet, 2018) was used as the backbone for this application since the API is an open source mapping API and it is a very light API compared to other web mapping APIs. Leaflet can be effectively incorporated with other JavaScript libraries and it can also be hosted locally on a desktop computer for testing. Leaflet handled the various map visualization, symbolization as well as interactivities with the map interface that were implemented in the design of E-Beacon application.

The remote database provider used for this research was MLAB. It is an online version of MONGODB and it allows one to connect to any mongo database from any server. Since E-Beacon is a server-side application, we made use of Node-JS 8.9.4 as the server side scripting language.

Open Street Map ((<https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png>)) and satellite imagery ([http://server.arcgisonline.com/ArcGIS/rest/services/World\\_Imagery/MapServer/tile/{z}/{y}/{x}](http://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x})) were used as base maps to give the user a realistic view of the environment where the control points are located. HEROKU Cloud (HEROKU, 2018) was used for hosting the application.

### *2.3.5. Application Administration: Updating of Database*

While developing the application, the task of updating the coordinates of the control points within the database was put into consideration because as new control points are being established in the field, they have to be made available for users. This necessitated the development of another application called E-

Beacon-UPDATER. This application can only be accessed by the administrator and it is linked directly to the E-Beacon application page. The administrator prepares the data in a Microsoft Excel file and then updates the control points by logging in through the ***admin section*** of the E-Beacon application.

### *2.3.6. Creation of Views*

Views in web development are the HTML pages. This aspect of the application coding involves the use of a templating engine to create the various views within the application. In order to achieve this, two additional directories, namely, the ***pages*** and ***partials*** were created within the views directory located in the E-Beacon project directory. The page directory contains some HTML files with the .ejs extension. These files are the pages which were rendered to the users. The partials directory also contains some additional HTML, CSS and JAVASCRIPT files with the .ejs extension. These files are *Index.ejs*, *script.ejs*, *head.ejs*, *header.ejs*. We wrote the codes for the views.

1. **Index.ejs:** This file contains some HTML tags as well as the necessary .ejs tags that make the rendering of the page possible by combining all other files with the .ejs extension
2. **Script.ejs:** This file contains the Javascript codes responsible for the following:
  - i. Displaying of base map to the user interface
  - ii. Plotting of control points on the base map
  - iii. Addition of some user interface widget for interactivity
  - iv. Styling of the control points
  - v. Generation of the shortest distance to a particular control point
  - vi. Displaying of additional information about each control points
3. **Head.ejs:** This file contains the major CSS code for styling the various HTML elements within the application. It is responsible for the colours of elements their sizes positioning and others. It also contains the various HTML tags needed in the head section of an HTML file. These tags include the META tag, LINK tags to mention but a few.
4. **Header.ejs:** This file contains the various HTML tags that made up the sidebar within the application.

### *2.3.7. Application Deployment*

This was the final stage involved in carrying out research. It entails sending the already developed E-Beacon application to the cloud for use. The application was deployed to the HEROKU cloud which is a free hosting site for testing web-based applications. In order to do this, an account was created with HEROKU before the application was then launched to the cloud.

### 3. Results and Discussions

#### 3.1. Results

The end product of this research is an interactive web-based survey control finder application called “E-Beacon”. A preview of the application is shown in Fig.4.

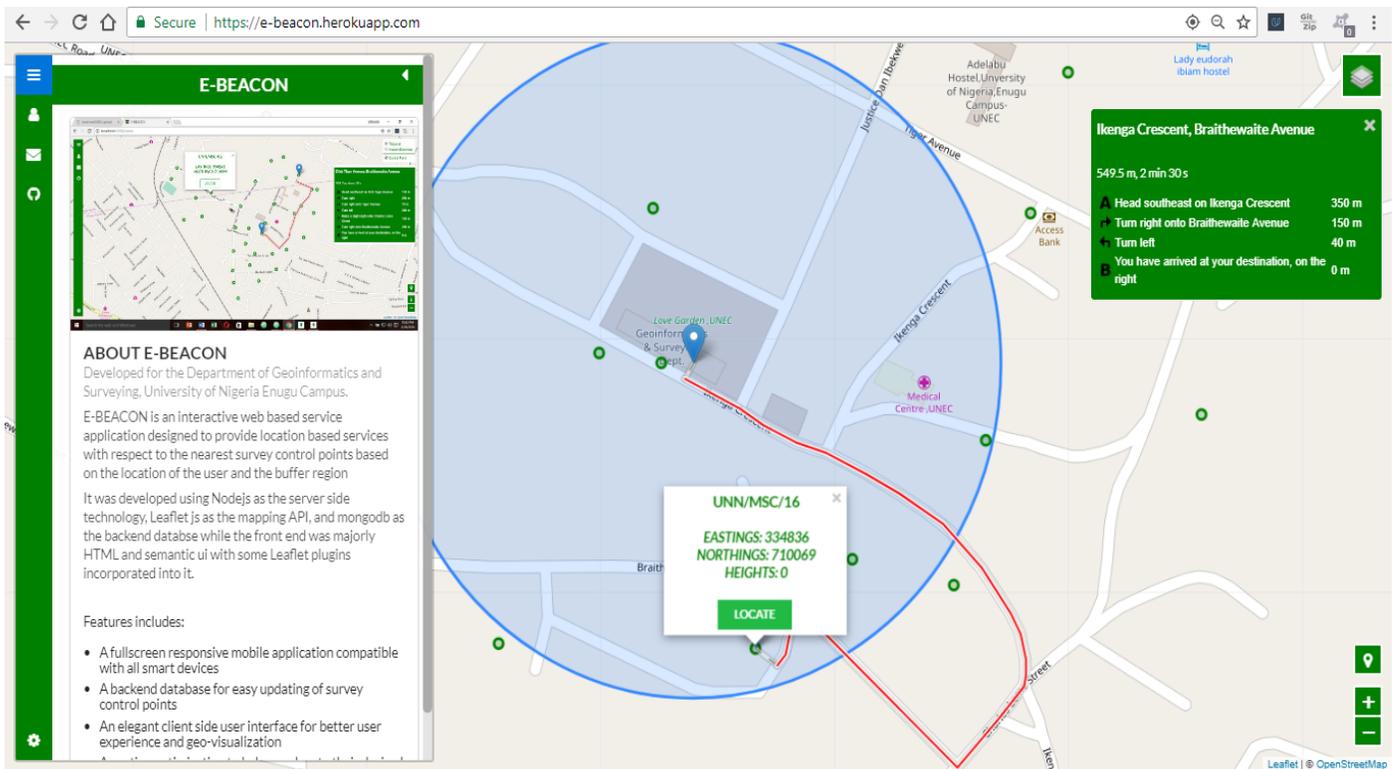


Figure 4. E-Beacon application

#### 3.2. Discussions

E-Beacon was designed to provide location-based services with respect to the nearest survey control point based on the location of any user within the University of Nigeria, Enugu Campus.

Furthermore, the application supports addition and updating of new survey control points into the on-line database since it is a server-side application. The application can be accessed online using the following link <https://e-beacon.herokuapp.com>. E-Beacon can perform the following basic functions:

1. Tracking the current location of the user (this attribute makes it usable globally but the current version has no information on survey control points of other places except the test area which is UNEC). This application can be used by all so long as they have access rights to include the control points of their area or country into the system. It is not localized to Nigerian geodetic datum
2. Responding to users click on any of the user interface widgets

3. Generating a route from the user's position to any control point of choice
4. Displaying attribute information about any control point
5. Displaying the control points within a particular buffer radius

E-Beacon also has the following features:

- A full screen responsive mobile application compatible with all smart devices
- A backend database for addition and easy updating of control points
- An elegant client side user interface for better user experience
- A routing optimization to help user locate control points nearest to them
- Possibility of updating the control points within the database which can only be done by the administrator as shown in Fig. 5. The administrator can login into the E-Beacon UPDATER application by inputting his or her username and password in the *admin section* of the E-Beacon app as shown in Figure 5(A). After logging into the application, he or she navigates to the file containing the control points that is to be uploaded. He or she then clicks the *upload* button (as shown in Figure 5(B)) and a page (as shown in Figure 5(C)) will come up, notifying the administrator that the upload was successful. We wrote the various codes for the UPDATER application.

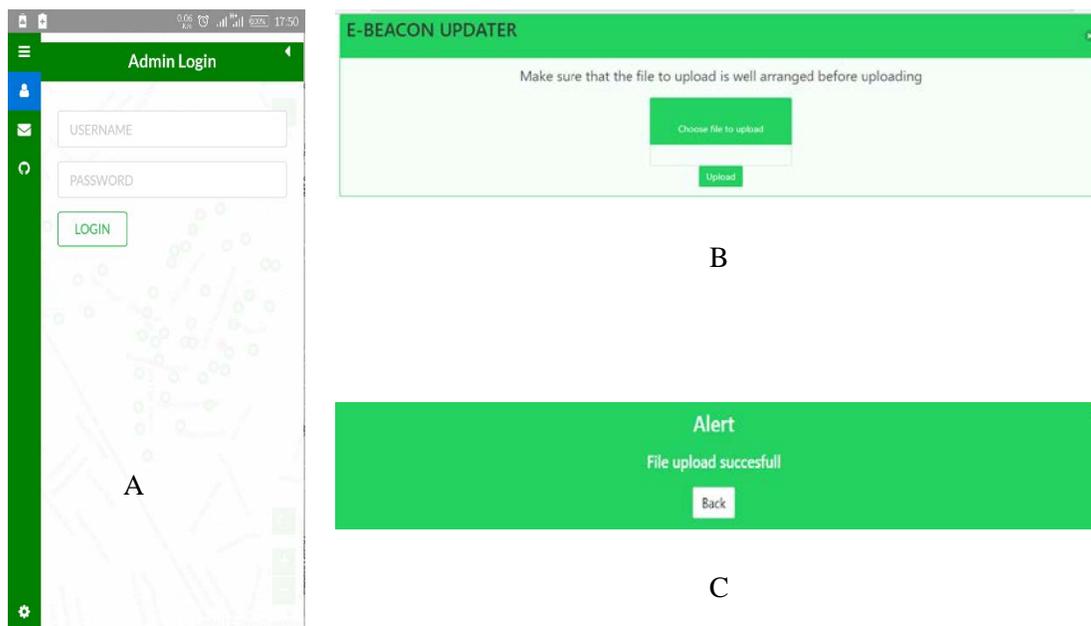


Figure 5. Accessing E-Beacon UPDATER from the E-Beacon application platform

### *3.2.1. User Friendly Interface*

While developing the application the end user was seriously put into consideration and as a result of this, the application was enriched with some important user interface widgets that can give satisfactory user experience. Below are some of the user interface widgets:

**Layer control:** The layer control handles the toggling between different base maps of choice ranging from the street map to the satellite imagery to give the user a realistic view of the environment where the control points are located.

**Zoom control:** The zoom control allows the user to zoom in/out on the desired level of details.

**Locate control:** The locate control allows the user to re-track its location after launching the application and helps the user to track its location when he or she moves to a different location.

**The route control:** The route control gives the user a description of what direction to follow along a particular route to locate any control point of choice.

**The Popup window:** The popup window displays the basic information about a control point whenever the user interacts with them by clicking.

**The Sidebar:** the sidebar was designed to give some pieces of information about the application. The sidebar control houses the contact section, buffer radius settings and the administrator's login section.

### *3.2.2. Responsivity Test*

After the development, the application was tested for responsiveness by previewing the application on various devices of different screen sizes to see how the user interface is rendered. This was done to ensure that the application can work effectively on any smart device and that it would take over the screen size of the device without distortion to the user interface widgets.

### *3.2.3. Location Tracking Comparison*

E-Beacon was also tested to check its user tracking ability. This was done to ensure that the approximate location information provided by the application is accurate and suitable for use. This was carried out by comparing the user coordinates tracked by the application at a point and the actual coordinates of the same point as obtained from the base map of UNEC. The result showed two (2) metres difference in distance between the tracked coordinates and the actual coordinates. This result is considered suitable since any survey control point within 2-metre radius can easily be seen by the user.

## **4. Conclusion and Recommendation**

The development of an interactive web-based survey control finder application (APP) which allows easy location of survey controls, while on the field and based on the location of the user, was successfully

carried out and the E-Beacon APP is readily available online (<https://e-beacon.herokuapp.com>) for use. It is a cross-platform application, meaning that it can run on any smart device irrespective of the Operating System (OS). Interestingly, this E-Beacon APP can be improved to meet users' needs and specifications.

This study gives insight into what the latest trend in technology has in store for surveying and how surveyors can maximize the use of some tools to make the surveying profession technologically driven. It also shows how life can be simplified for a particular set of people if what they do is transformed into a well-meaning application which they can easily access from their comfort zones. There is need to arouse the interest of learners in web application programming in order to develop more surveying related applications.

## **5. References**

- Alphadevx 2018, The basics of web technologies, viewed 20 March 2018, <https://www.alphadevx.com/a/7-The-Basics-of-Web-Technology>
- Amit K & Vineet K 2011 'Location Based Services using Android Mobile Operating System', International Journal of Advances in Engineering & Technology, vol.1, no.1, pp. 14-20.
- City of Orlando Control Network 2018, Survey Services Website, viewed 7 January 2018, <http://orl.maps.arcgis.com/apps/webappviewer/index.html?id=cf0cc8e9c3fb4f7f93dcfc27dbdbb37a>
- Google Maps APIs 2018, Store-locator-leaflet.pdf, viewed 7 January 2018, <https://static.googleusercontent.com/media/enterprise.google.com/en/maps/files/store-locator-leaflet.pdf>.
- HEROKU 2018, Heroku Cloud Application Platform, viewed 7 January 2018, <http://www.heroku.com>
- Hu H & Lee D 2004 'Semantic location modeling for location navigation in mobile environment', In Proceedings of the 2004 IEEE international conference on mobile data management (MDM'04).
- IAMAI 2008, Location Based Services on Mobile in India for IAMAI-version, viewed 20 March 2018, [http://www.IAMAI.in/upload/policy/LBS\\_draft\\_indices.pdf](http://www.IAMAI.in/upload/policy/LBS_draft_indices.pdf).
- Jegadeesnari R Parameswaran S 2014, 'Location Based Using Autonomous GPS', International Journal for Research in Applied Science and Engineering Technology, Special Issue 3, pp. 84-91
- Leaflet 2018, Leaflet Javascript Map API, viewed 10 January 2018, <https://www.leafletjs.com>
- Leaflet 2018, Leaflet Sidebar, viewed 10 January 2018, <https://github.com/turbo87/leaflet-sidebar>
- Leaflet 2018, Leaflet Routing Machine, viewed 10 January 2018, <https://github.com/perliedman/leaflet-routing-machine>
- Leaflet 2018, Leaflet Locate Control, viewed 10 January 2018, <https://github.com/domoritz/leaflet-locatecontrol>
- Manar S & Aruipam S 2012, 'Implementation of Location Based Services in Android Using GPS and Web Services', International Journal of Computer Science, vol. 9, no. 2, pp. 237-242.
- mLab 2018, mLab Cloud MongoDB Hosting, viewed 7 January 2018, <http://mlab.com/signup>.
- Node.js 2018, Nodejs Server side Scripting Language, viewed 10 January 2018, <https://www.nodejs.org>
- NPM 2018, Node Package Manager, viewed 10 January 2018, <https://www.npmjs.com>
- Omar A Ibrahim, Khalid J Mohsen 2014, 'Designing and Implementation an Online Location Based Services Using Google Maps for Android', International Journal of Computer Network and Communications Security, vol. 2, no. 3, pp. 113-118

Pankti D, Proja J & Abhishek S 2014, 'Location Based Services in Android', International Journal of Engineering and Computer Science vol. 3, no. 3, pp. 5072-5077.

Semantic UI 2018, Semantic UI Frontend Framework, viewed 7 January 2018, <https://www.semantic-ui.com>.

Survey Control Finder 2018, Survey Control Finder, viewed 20 March 2018,

<https://maps.sco.wisc.edu/surveycontrolfinder/#7/44.735/90.148/NGS,county,USGS,CORS/terrain>