Tarikh Chouhan – w1481217

Draft of Project Initiation Document

Author : Tarikh Chouhan

Supervisor: Dimitris Drocopolous

Summary

This project aims to ease users frustration when parking their car by providing real time update to car parking spots.

Purpose

Identification and discussion for the scope of the project

Aims + Objectives/ Goals

Identification and justification of project aims and objectives that are related to the project

Identification of activities and sources required to satisfy initial investigation relating to related works and literature review.

Identification and reasoning of initial list of requirements from initial project investigation.

Project Deliverables

Supporting Project Plan indicating and detailing activities, timescales and deliverables.

Risks

Identification and discussion of the project problem

Ethics

Form

--

Agile, Git

Using IoT to simulate a car parking bay and provide real time updates via an app based on external factors.

# Project Initiation Document

## Author: Tarikh Chouhan - w1481217

## Supervisor: Dimitris Dracopoulos

**General Scope of Project**

Finding a car parking bay can be very frustrating to drivers. As a result of this frustration, drivers tend to park illegally and end up having to pay a penalty/fine. Local councils are generating massive amounts of revenue by handing out parking fines. The following statistics paint a picture on how significant the car parking industry is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Surplus in £(millions) made in parking fines per local council [[1]](#footnote-1) | | | | | |
| Local authority | **2010/11** | **2011/12** | **2012/13** | **2013/14** | **2014/15** |
| Westminster | 38.2 | 41.6 | 39.7 | 51 | 46.4 |
| Kensington & Chelsea | 21.1 | 28.1 | 30.4 | 33.5 | 33 |
| Camden | 21.1 | 25 | 23.5 | 24.9 | 24.5 |
| Hammersmith & Fulham | 16.6 | 19.5 | 19.4 | 23 | 23.8 |
| Wandsworth | 14.4 | 16.1 | 15.9 | 19.7 | 20.4 |
| Brighton & Hove UA | 12.7 | 14.4 | 16.3 | 18.1 | 18.6 |
| Haringey | 3.3 | 5.3 | 5.2 | 5.7 | 16.1 |
| Islington | 5.6 | 10.9 | 8.2 | 10.4 | 13.7 |
| Nottingham City UA | 3.7 | 3.3 | 11.8 | 12.1 | 13.3 |
| Hackney | 4.7 | 5.9 | 7.8 | 8.2 | 10.8 |
| Brent | 3.9 | 2.7 | 2.7 | 8.3 | 10.5 |
| Tower Hamlets | 6 | 5.8 | 7 | 8.3 | 10 |
| Birmingham | 5.1 | 5.5 | 6.9 | 7.8 | 9.7 |
| Lambeth | 7 | 5.8 | 12 | 7.2 | 9.7 |
| Milton Keynes UA | 6 | 6.6 | 6.7 | 8.2 | 9 |
| Cornwall UA | 8.2 | 7.9 | 8.1 | 8 | 8.7 |
| Manchester | 1.9 | 6.3 | 8.8 | 8 | 7.9 |
| Hounslow | 6 | 7.3 | 6.4 | 7.8 | 7.7 |
| Newham | 3.9 | 7.3 | 8.2 | 7.2 | 7.3 |
| Merton | 4.4 | 5.7 | 6.9 | 7 | 7.2 |

The aim of this project is to help solve an on-going problem most car drivers face daily; finding an available bay to park in. This project will aid and help car drivers find a car parking bay that they’ll be able to park in without driving further than necessary, searching for an available space to park in. This will be achieved by creating a sensor that will record the data from an ultrasonic sensor and thermistor module. Once the modules acknowledge a drastic change in these two external factors, the results will be shown to the user via an app hence indicating whether or not the bay is vacant or occupied. As well as that, the app will learn how the data correlates to the bay being vacant or occupied, essentially machine learning. This will be achieved by implementing the concept, neural network. By implementing this feature, the app will be able to predict whether or not the bay is being used and will come in handy if for some reason the sensor cannot fulfil its purpose (e.g. if there are networking issues, wiring issues, external factors such as rain etc.).

Upon preliminary research, it shows that this type of concept does exist but is still in its infancy. An example of this is Google Maps. In their latest update to the app (August 2017), they are using historic data with machine learning to predict the availability of car parking bays[[2]](#footnote-2). Whilst this is a step in the right direction to solving the problem described, it is not as accurate as having a physical sensor embedded to the parking bays. As the world progresses further in technology, more and more devices are being connected to the internet, the concept of IoT (internet of things) will be greatly beneficial here and will outperform the use of machine learning. With this in mind, I believe my project will perform greatly as it has the concept of IoT and will be able to serve its purpose with accurate results whilst fulfilling its requirements.

**Aims and Objectives**

In order to successfully complete this project and have a live working sensor and app, there will be strict aims and objectives that will have to be met set against the listed requirements. Failure to do so would result in the sensor and/or app not performing as intended or worse, not performing at all. In order to prioritise my aims and goals, the MoSCoW principle will be used. The MoSCoW principle is widely used as a prioritization technique and is heavily used in field of business analysis, project management and software development. The capital letters in the term ‘MoSCoW’ is an acronym coming from the first letter of each prioritization groups:

M – Must

S – Should

C – Could

W - Would

Firstly, my main aim is to build the sensor as this project resolves around solving the described problem using the concept of IoT. The sensor must take readings from external factors such as temperature and distance from an object. This will be achieved by acquiring components that will be able to record these factors; thermistor to measure the temperature and an ultrasonic sensor to measure distance. The reason for recording external factors such as temperature and distance is because this will indicate that a car has been driven on top of the sensor; effectively indicating that the bay is now being occupied. The base of the car will be hot since the engine will have been running, and so there will be a drastic temperature change to the sensor as well as the base of the car being close to the sensor, effectively being detected by the ultrasonic sensor.

Moreover, in order for the sensor to be connected to the internet and be a part of the IoT category, it must be able to send and receive data over some form of network protocol. Ideally, this will be in HTTP as this form of protocol is widely used to send and receive data. In order to achieve this, a microcontroller (the sensor) will be used and will be capable of connecting to WiFi by wiring a wifi module to it. The sensor will register external changes every 30 seconds and send the data to a server.

Furthermore, the user must be able to know whether the parking bay is being occupied or not. An effective way of showing this will be from an app. This will be achieved by incorporating the data from the sensor with the Google Maps API. By using Google Maps API, I will be able to show the location of the sensor to the user as well as the status of the parking bay. As of now, the app will be an android app.

Even more, a server must be created so that the sensor and app will be able to communicate together. An effective way of communicating to this server will be over the REST architecture. REST is widely used in servers and web services as it caters a wide array of different formats for the response, i.e. JSON, CSV, XML etc. REST is widely chosen over SOAP as parsing through XML and the friction between Javascript and XML can cause issues[[3]](#footnote-3).

The above were the main aims and objectives of this project, without them, the project would be unable to perform and fulfil its main requirements which is essentially to notify the user if a car parking bay is available or not. The following aims fall under the ‘should’ category:

Firstly, the app should be able to predict when a parking bay will be in use or not depending on the data already collected by the sensor. This will be achieved by implementing neural network. The inputs into the network will be date and time and the output should be 1 or 0, occupied or vacant respectively.

Secondly, the database being used to store the data from the sensor should be in a NoSQL database. The reason behind this is because NoSQL databases are not confined in a set structure once created in contrast to relational databases. By making sure the database is able to cater for structured and not structured data, this will reduce complexity when it comes to the event if I need to include another component to the sensor and record its result. If I chose the traditional tabular relational database method, then editing the table heavily will increase complexity but with a NoSQL approach, it will be with ease.

The table below is an overview of my aims and objectives.

|  |  |  |
| --- | --- | --- |
| **MoSCoW Category** | **Aim** | **How will this be achieved** |
| Must | Building the sensor | Using an Arduino Uno and gathering the write components, wiring them and writing the source code. |
| Must | Sensor must be able to connect to wifi | By a connecting a wifi module to the microcontroller |
| Must | Sensor must be able to connect to server using HTTP | Using the Arduino libraries for HTTP connection |
| Must | App should show results of the sensor | The app will communicate with the senor, gather the data and display them using on a map using Google Maps API |
| Must | Server should communicate using REST | Using a server capable of REST methods (GET, POST, etc) and return response in JSON format. |
| Should | App should predict when a parking bay will be taken. | By implementing neural network. |
| Should | Use a non-tabular relational database | Use a NoSQL database such as MongoDB |
| Would | Implement a feature where user enters address and app will tell if there is a parking bay spot available | To show more than one parking bay availability, I’d have to create a sensor for each parking bay which is not in the scope of this project. |

GPS – write in must category

MENTION ARDUINO

MATLAB

HTTP mode of transporting

research

**Project Deliverables**

The majority of this project is split into 3 major parts; Sensor, Web Server and App. By focusing on each part individually, I’ll be able to split and prioritise my time accordingly and maximise my efforts as I will be solely focused on the part I am currently working on.

There are many different styles and approaches when it comes to planning. There is the ‘Waterfall’ method which is a sequential approach where you focus one phase, get it done, and move onto the next phase. A general waterfall model will have 4-6 phases which include; requirements, design, implementation and maintenance.

Another method is the SCRUM method which is popular amongst the tech industry. SCRUM is a subset of the Agile methodology which is based on iterative development. SCRUMs core concept is the use of development cycles called Sprints which allows one to dynamically adapt to changes whether it be from issues in code or requirements changing. Furthermore, issues/tasks are placed in Sprints and within the duration of the Sprint, one aims to get everything finished. Upon completion of the Sprint, there is a Sprint planning in which issues and tasks are taken from a backlog and placed into the current Sprint. By having this ability, this is what makes SCRUM more dynamic and adaptable to change.

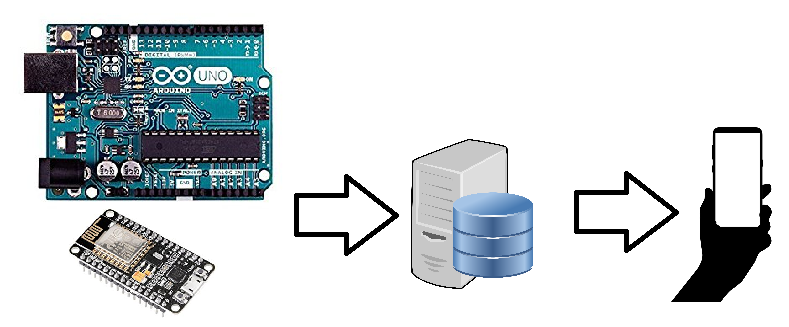
Based on these methodologies, the most sensible approach would be to split the project into 3 parts; sensor, web server and App, and focus on them individually as well as each part having their own methodology. Since the sensor won’t be subject to a lot of changes, it would be advisable to go to the waterfall method. The reason for this is because the waterfall method is useful on clear and concise projects. When a phase is completed, there is necessarily no need to back track on the phases prior. Because of this concept, this makes it advisable to use the waterfall method when constructing and coding the sensor as once it is built and working, there will be no need to go back. However, because the web server and app is more active and subject to changes, tackling these parts of the project will be done in an agile manner. Mainly in a SCRUM manner as it will give more time to adapt to any changes.

Furthermore, Git will be used throughout the project as it is a widely used version control system and is very powerful even when using the basic features of it. Git will aid me as I’ll be committing small segments of working code at a time to ensure nothing goes wrong. If something does go awry, I can always revert to a previous commit.

Even more, I will need to spend time researching, designing, implementing and testing each of the 3 parts to the project. This will be done within the relevant timeframes as I split my project into 3 parts; the sensor, web server and the app. Below is an estimate on how I aim to achieve this:

|  |  |
| --- | --- |
| Week | Type of work |
| 1-3 | Research into Arduino Uno |
| 1-3 | Research into databases |
| 1-3 | Research into Android development |
| 3 | Choose an appropriate tech stack to use |
| 4-5 | Buy components for the sensor |
| 5-6 | UML design for app |
| 5-6 | Construct the foundations of the routing in web server with unit tests |
| 6-8 | Research into Machine Learning and understand whole concept |
| 9-12 | Finish wiring and coding the sensor.  Finish coding the server with unit tests.  Make sure sensor is able to communicate with server by sending via HTTP |
| Christmas |  |
| 13-16 | Construct app following TDD principles |
| 17-19 | Finish app and make sure all requirements are met |
| 20-24 | Start and finish final year report. |

Risks



1. RAC Foundation, December 2015, Council parking 'profits' up again in England, Available at: http://www.racfoundation.org/media-centre/council-parking-profits-up-again-in-england-2014-15 [↑](#footnote-ref-1)
2. Google Blogs, August 2017, Put it in park with new features in Google Maps, Available at:https://www.blog.google/products/maps/put-it-park-new-features-google-maps/

   [↑](#footnote-ref-2)
3. John Mueller, January 2013, Understanding SOAP and REST Basics And Difference, Available at: https://blog.smartbear.com/apis/understanding-soap-and-rest-basics/ [↑](#footnote-ref-3)