Group Project Report
Optimisation for Transport for London Barclays Cycle Hire

COMP3001: Technology Management and Professional Issues
Department of Computer Science, University College London

Authored by
Team 5
Marcin Cuber- marcincuber@hotmail.com
Jonny Manfield, Nicola Greco, Ran Gutin, Edward James, Toshiyuki Nishino (NII), Richard Isaac, Keqin Feng, Christodoulos Demetriades, Navid Hallajan, Mohan Dai, Michael Detmold, Ragavan Guneshalingham

Client
Internal client - Graham Collins
External client - Dr. Stephen Pryke (s.pryke@ucl.ac.uk)

Links
GitHub Repository (Main) - https://github.com/jonnymanf/3001
GitHub Repository (TubeMaps library) - https://github.com/nicola/tubemaps
Extracted data API for future use - http://178.62.32.221:5000/

Abstract: The TfL Barclays Cycle Hire Scheme has a redistribution problem which results in a loss of an estimated £25 million per year. Users have total autonomy over docks they take from and return to. This leaves the network in an uneven state, creating an undesirable situation. A multitude of factors affect this autonomous distribution of bikes. As a result, TfL has the vast expense of re-distributing the bikes by vehicle with the goal of restoring bike availability throughout the network by removing bikes from full docking stations and placing them at undersupplied docking stations.

We built an analytics application to normalise and process various TfL data sources and provide interactive data visualisations to demonstrate the network disruptions. As a case study, we investigated the correlation between tube engineering works/closures/delays and availability of bikes by providing methods to detect anomalous network behaviour, i.e. abnormally high or low bike demand/supply. We focused on how docking stations within a given radius of a tube station are affected by tube engineering works or delays/closures. We found the investigated correlation to be causal and demonstrated the Return On Investment (ROI) TfL would benefit from by making redistribution decisions informed by our data visualisations, saving in the order of £1 million a year.

Can we optimise re-distribution of the bikes by providing data visualisations showing anomalies within the network and identifying potential causes?

Keywords: Network Optimisation, Visualisation, Analytics, Project Management

Contents

Group Project Report 5
1 Introduction 5
2 Pre-Study and Related Work 5
3 Analysis 5
3.1 Requirements 5
3.2 Architecture 6
3.3 Data Sets 6
3.4 Data Science 6
3.5 Data Visualisation ..... 8_Toc405538745
4 Results 10
4.1 Description 10
4.2 Visualisation Results 10
4.3 Specification 13
5 Conclusions and Future Work 14
5.1 Potential Return On Investment (ROI) 14
Appendix A - Research and Requirements Gathering 16
A.1 Research 16
A.1.1 Research Phase 16
A.2 Client and Customer Interviews 18
A.2.1 Client Interview 18
A.2.2 Customer Interview 20
A.3 Lightweight Requirements Shell 21
A.4 Personas and scenarios 22
A.4.1 Persona & Scenario 1 - Daniel 22
A.4.2 Persona & Scenario 2 - Brian 23
A.5 Use case 24
A.6 Research Findings 24
Appendix B - Development Plan 25
B.1 Overview 25
B.2 Software Requirements Specification 25
B.2.1 Functional Requirements 25
B.2.2 Non-Functional Requirements 26

B.3 Need for project 26
B.4 Challenges 27
B.5 Opportunities 27
B.6 Initial analysis 27
B.6.1 Recognition of a problem- 5 whys approach 27
B.7 Team division 29
B.7.1 Explanation of Team Roles 29
B.8 Project Objectives 30
B.8.1 Project Constraints 30
B.8.2 Risk Management 30
B.9 Architecture structure 32
B.9.1 Architecture 32
B.10 Testing 34
B.11 Project Management Approach 34
B.11.1 Development Model 34
B.11.2 Configuration Management 36
B.11.3 Communication Management 36
B.12 Feedback and Iterative Design Process 38
B.12.1 Feedback and Iterative Design Process 38
B.12.2 Changes to our Hypothesis/Goals 38
B.12.3 Changes to visualisations 39
B.12.4 Changes to formula 40
B.13 - Evaluation about the re-development 40
B.13.1 How We Would Do It If We Were To Do It Again 40
Appendix C - Architectural Diagrams and Data Sets 43
C.1 Data Sets 43
C.1.1 TfL Bikes Open Data 43
C.1.2 TfL tube data 43
C.1.3 Maps Data 44
C.2 Extracted data for future use 44
C.2.1 API Documentation 44
C3 Architectural Diagrams 45
C.3.1 Application Overview Diagram 45

C.3.2 Network Architecture Diagram 46
C.3.3 Detailed Application Diagram 47
C.4 UML Diagram 48
C.5 Sequence Diagram 48
Appendix D - Earned Value and Costs 49
D.1 Value of Product 49
D.2 Earned Value 49
D.3 Ease of Use 49
D.4 Development Costs 50
D.5 Commercial Viability Estimate 50
D.6 Return On investment (ROI) 50
Appendix E - Conduct Policy 53
E.1 Ethical and Privacy Policy Considerations 53
Appendix F - Project Resources 55
F.1 Primary Source 55
F.2 Secondary Sources 56
Appendix G - Agile Approach 57
G.1 Overview 57
G.1.1 System Over Documentation 57
G.1.2 Principles 58
G.2 Agile and Interactive Solution 59
G.2.1 Iterative Solution 59
G.2.2 Very short feedback loop and adaptation cycle 59
G.2.3 Quality Focus 59
G.2.4 Agile Approach Conclusion 60
Appendix H - Peer Assessment ..... **Error! Bookmark not defined.**
H.1 Peer Assessment ..... **Error! Bookmark not defined.**

Group Project Report

Word count: 1499 (Excluding abstract, figures, tables, captions, references and (see Appendix X))

1 Introduction

Re-distribution problem. The Transport for London (TfL) Barclays Cycle Hire (BCH) scheme started in 2010; it now operates 10,000 bikes across ≈750 docking stations. Users can rent a bike from a docking station and return it to any other docking station. The re-distribution of the bikes within the network is outsourced to a company called Serco Group, who uses vehicles to move bikes to different docking stations.

TfL doesn't have an automated dynamic re-distribution system and incurs a heavy cost during re-distribution which prevents the BCH scheme from being profitable. Currently, TfL doesn't perform any optimisation on the re-distribution process; this was established in the client interview with Dr. Stephen Pryke. TfL monitors the public transport network and collects usage data which is released as Open Data.

Project hypothesis. The project hypothesis is as follows: Can we optimise re-distribution of the bikes by providing data visualisations showing anomalies within the network and identifying potential causes? In our case study, anomalies are docking stations which experience abnormally high or low demand/supply of bikes. The potential causes we are examining are tube closures or engineering works and delays.

Project goal. We aim to develop an application that can analyse usage patterns in the network and detect these anomalies. The results will be presented as interactive data visualisations. To directly address the project hypothesis, the visualisations will show areas of the tube network which are affected by closures or engineering works/delays and indicate how nearby docking stations are affected.

This report consists of five main sections. Section 1 is the introduction. Section 2 presents key findings from pre-study and related work. Section 3 details the analyses we have performed. Section 4 outlines our results. Section 5 concludes our results and presents scope for future work as well as improvements.

2 Pre-Study and Related Work

To research the problem, we conducted an interview with Dr. Stephen Pryke (see Appendix A.2.1) and customer interviews with users of the BCH (see Appendix A.2.2), from this we found the following:

- Disruptions cause users problems (Customer interviews)
- TfL doesn't currently optimise the re-distribution of bikes (Client interview - Dr. Stephen Pryke)
- Lack of communication between TfL and BCH (Client interview - Dr. Stephen Pryke)

Our team completed extensive reading on research papers and journals related to the subject area. More information on the research (see Appendix A), resources used throughout the project (see Appendix F).

3 Analysis

3.1 Requirements

To determine requirements, interviews were carried out to effectively gather information from our client and users of the TfL BCH scheme. From this information, a lightweight requirements shell was formed (see appendix A.1.3).

This was converted into a comprehensive set of requirements (see appendix B.1.1) in our development plan so there is a clear idea of how the system should function.

3.2 Architecture

The development of our application was completed with the following architecture:

- Web Scrapers
  - Python web-scraper for engineering works data
  - Python web-scraper for BCH journey information
- Back-end Development
  - Python script to parse XML data
  - PostgreSQL database to store data
- Data Visualisation
  - D3.js Data Visualisation

3.3 Data Sets

We used the following data sets:

- **TfL Open Data:** Data for bike availability
- **Engineering works data:** Scraped information
- **Google Maps API:** Map data

More information (see Appendix C).

3.4 Data Science

The Data Science team analysed the correlation between engineering works and bike availability at nearby docking stations.

Fig 1 illustrates the number of bikes available at a docking station close to Embankment station on a day without engineering works disruptions.

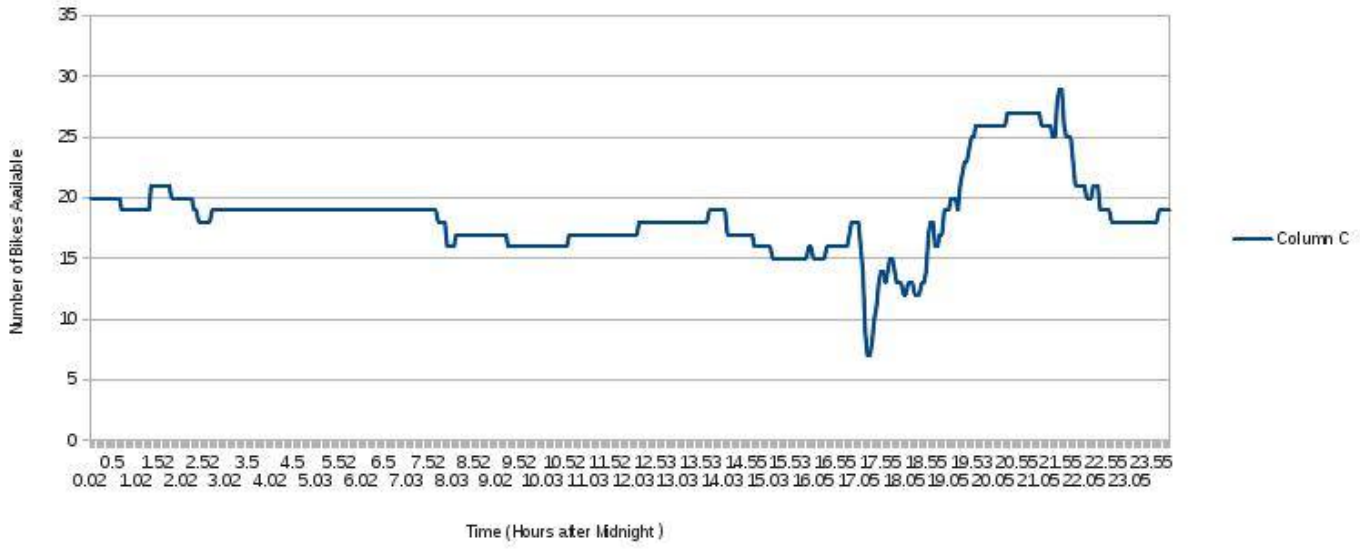


Fig 1 - Data Visualisation Iteration #1

Fig 2 illustrates how the usage pattern is affected between 07:30 and 12:00 (when there were severe delays). We observe that the supply of bikes to the station increases, this is because no passengers are there to rent them, while some commuters are taking bikes to Embankment to get on a connecting (unaffected) line.

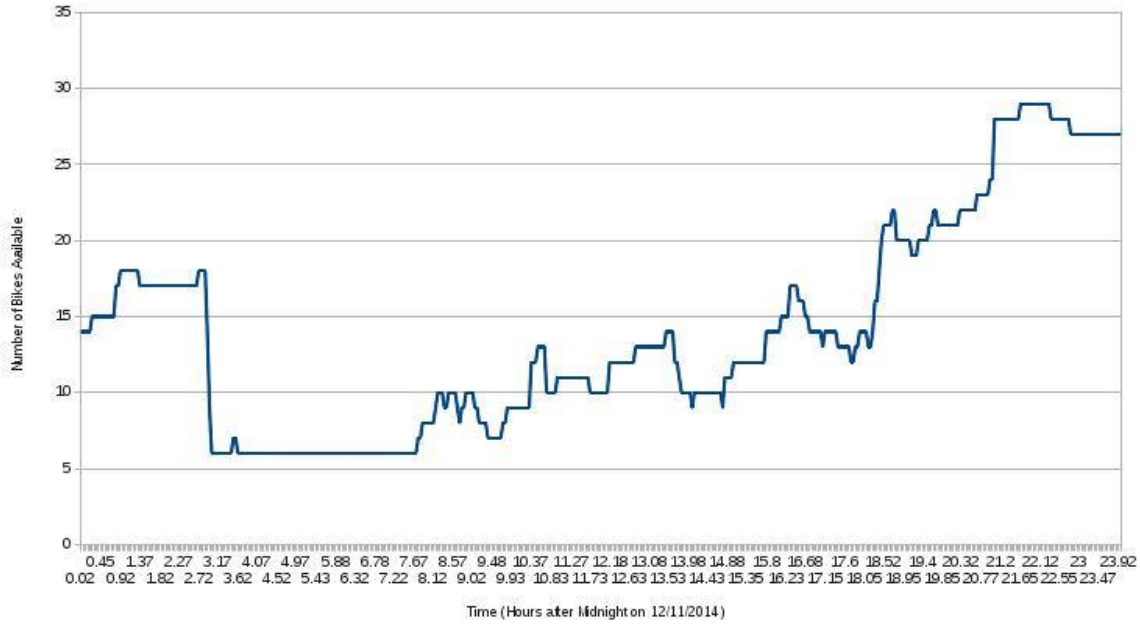


Fig 2 - Data Visualisation Iteration #1

The above pattern was reinforced by other examples. However, in some situations, nearby docking stations had an increased demand. The variation of results highlights the importance of developing data visualisations to display this information. To achieve this, we created a Python library run on the command line that queries our API for data about docking stations, generating a plot of bike availability over time.

Figure 3 illustrates an example run of the Python library. The terminal shows a request for the station with id ‘123’, the resulting visualisation on the right.

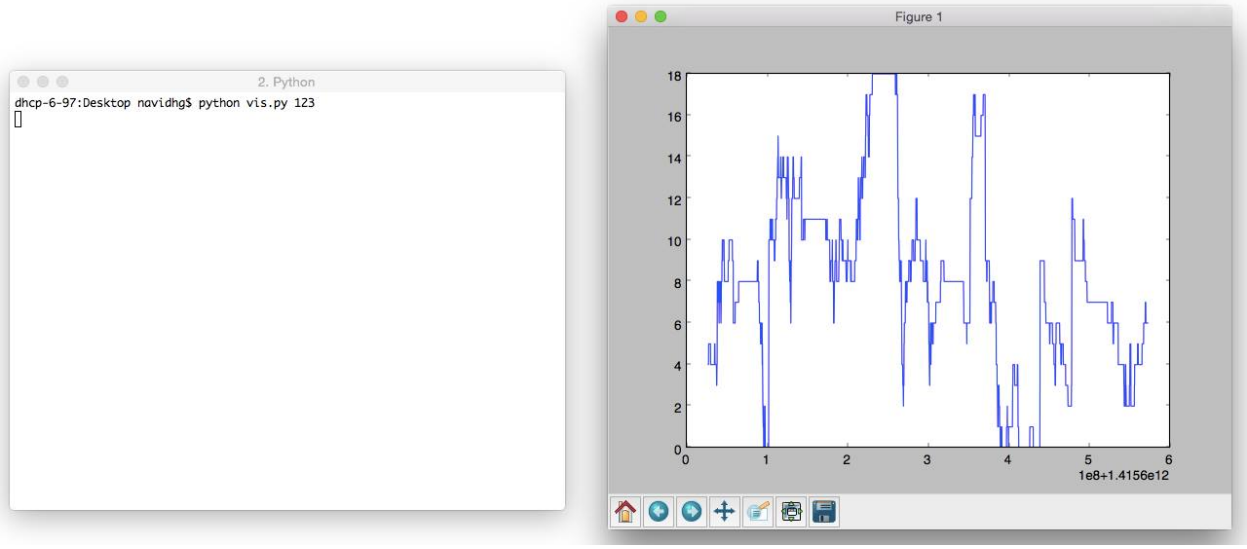


Fig 3 - Python library for availability graphs

3.5 Data Visualisation

Fig 4 visualisation illustrates an overlay of tube stations and bike docks over Google Maps. Fig 5, uses the TubeMaps library and illustrates the location of tube lines overlaid with bike rack data. Fig 6 illustrates our custom framework in use identifying line closures.

Fig 7 illustrates a Tube station Voronoi map of London, for each polygon on the map, there is only single tube station within it, which is the closest station to any point in that polygon).

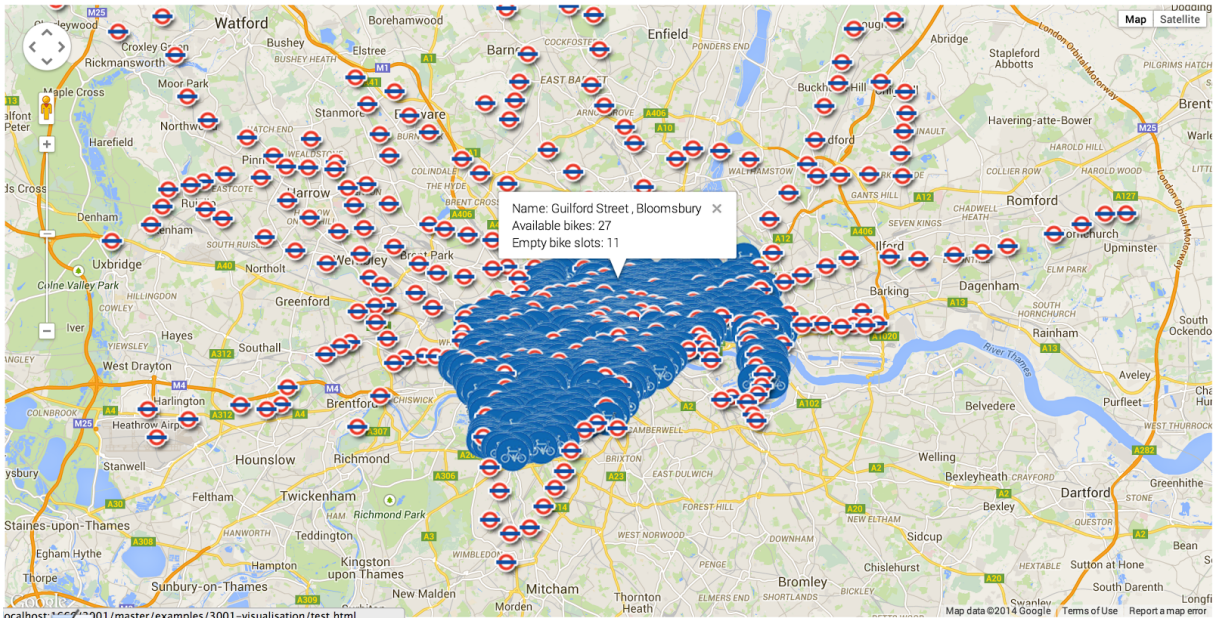


Fig 4 - Initial visualisation

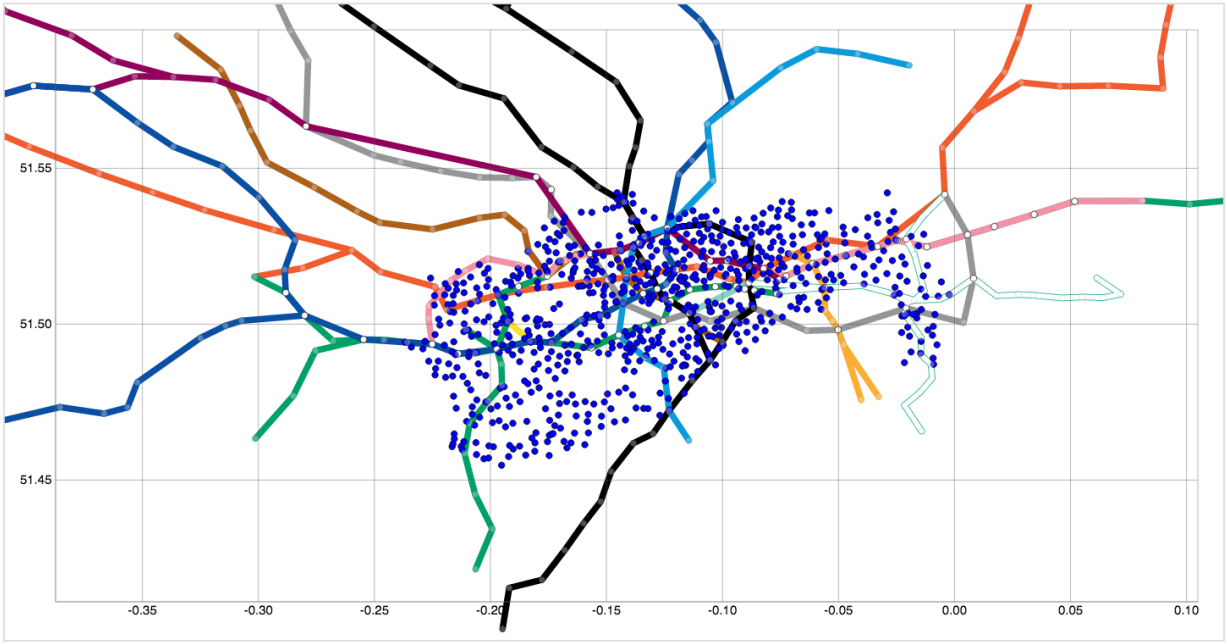


Fig 5 - TubeMaps Library

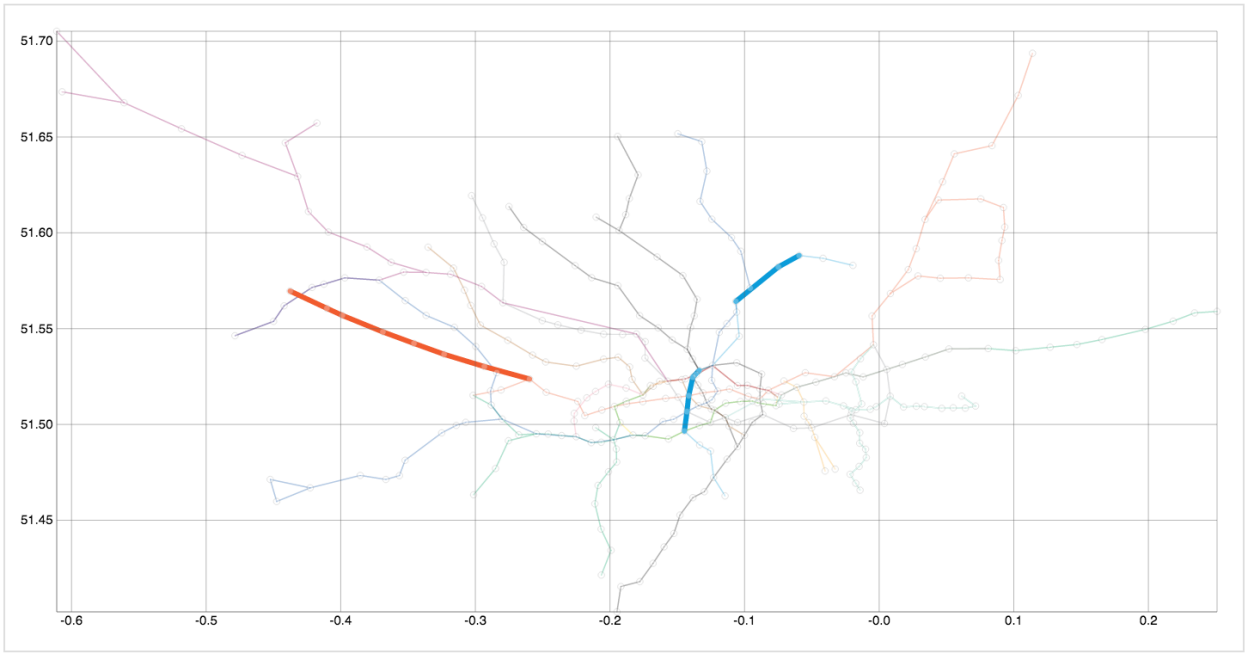


Fig 6 - Line Closure Framework

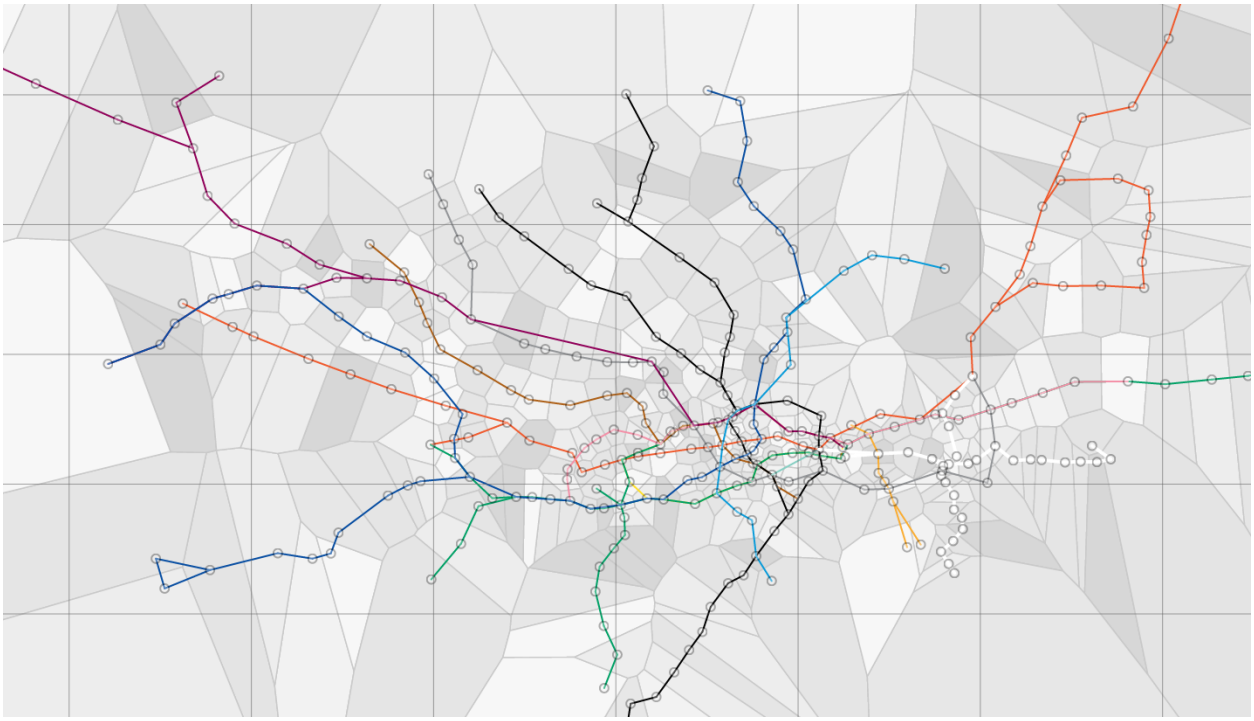


Fig 7 - Voronoi Map of Tube stations

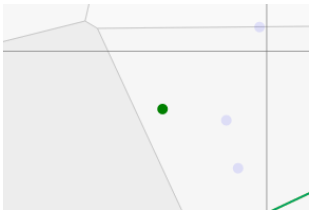
Our analytics allowed us to generate usage patterns for docking stations in the network, and showed that they are affected by engineering works, delays and closures.

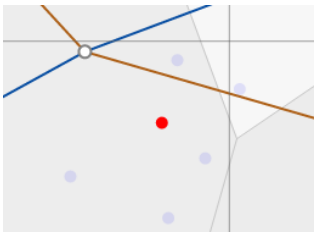
4 Results

4.1 Description

Our optimisation is provided via informative data visualisations. Our analytics application allows TfL to visualise the engineering works and infer how it affects the flow of bikes in the affected areas.

4.2 Visualisation Results

In all our visualisations, anomalies are represented with the following **key:**	
	2 or fewer bikes remaining

	2 or fewer spaces remaining

Visualisation anomaly key

Due to the flexibility of our libraries, we can adjust this threshold for anomalies.

Fig 8 illustrates bike availability across the entire network. Using this visualisation, TfL can detect anomalies at docking stations.

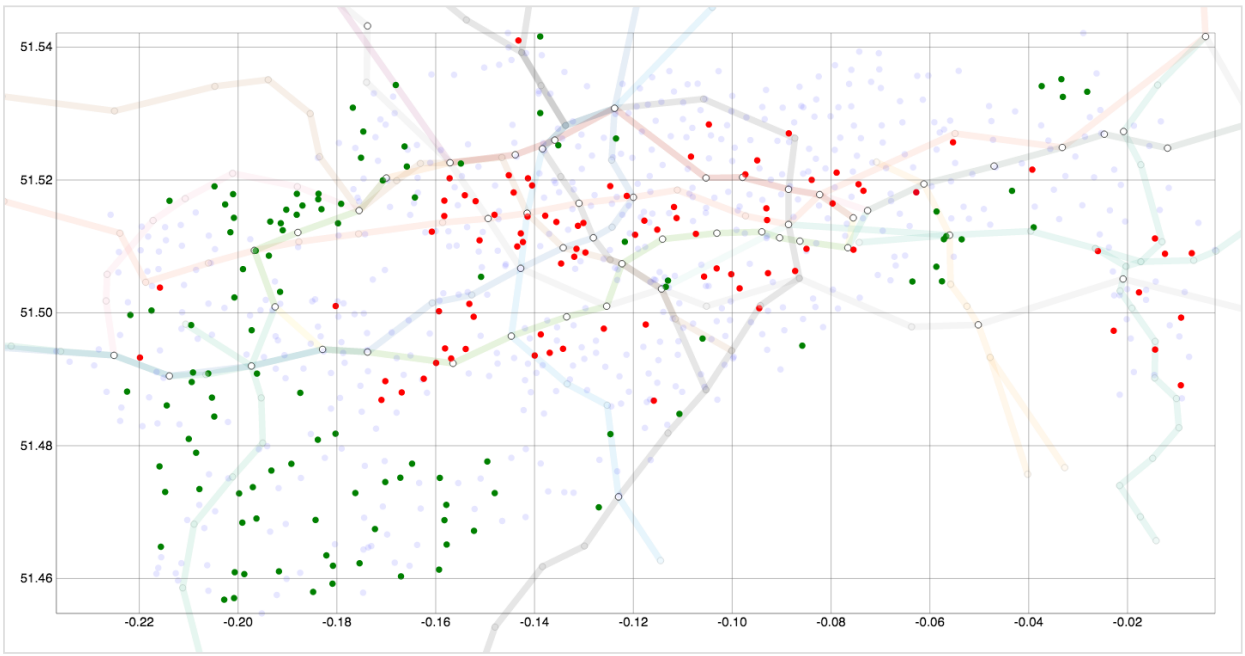


Fig 8 - Data Visualisation Iteration #2

Fig 9 illustrates areas of London affected by closures on Bakerloo and Jubilee lines.

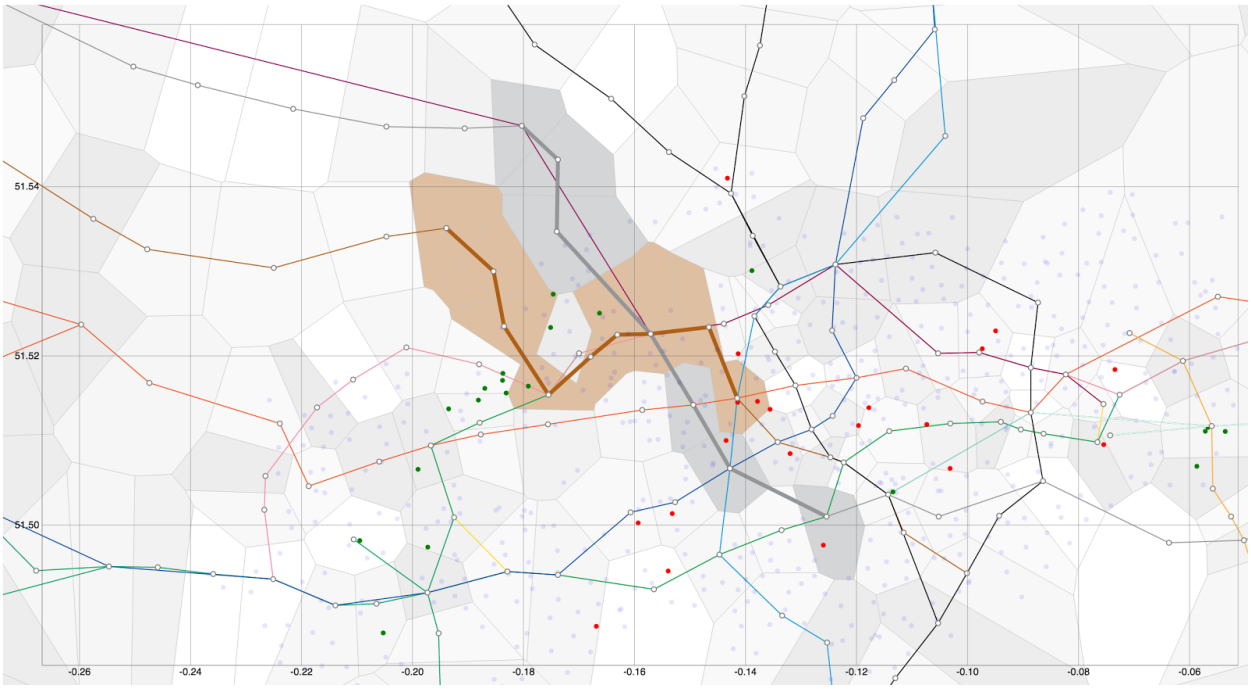


Fig 9 - Bakerloo/Jubilee line closures

Fig 10 illustrates areas of London affected by closures on Central and Victoria lines.

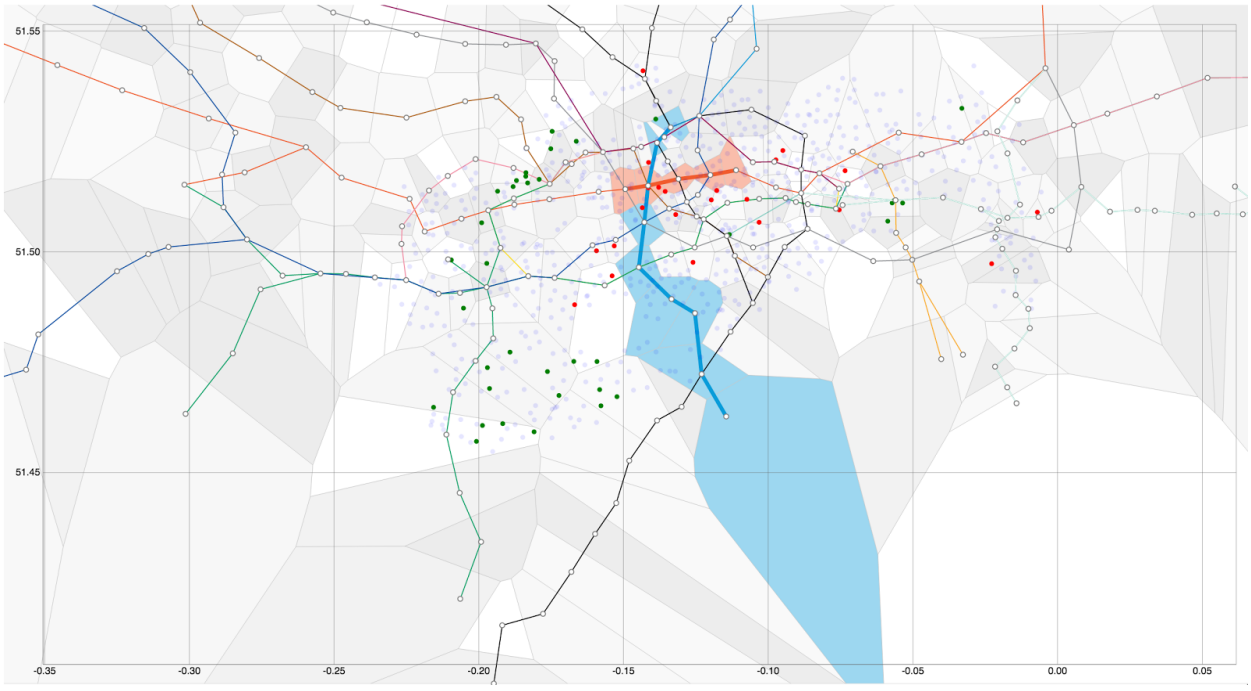


Fig 10 - Central/Victoria line closures

Fig 11 illustrates areas of London affected by closures on District and Northern lines.

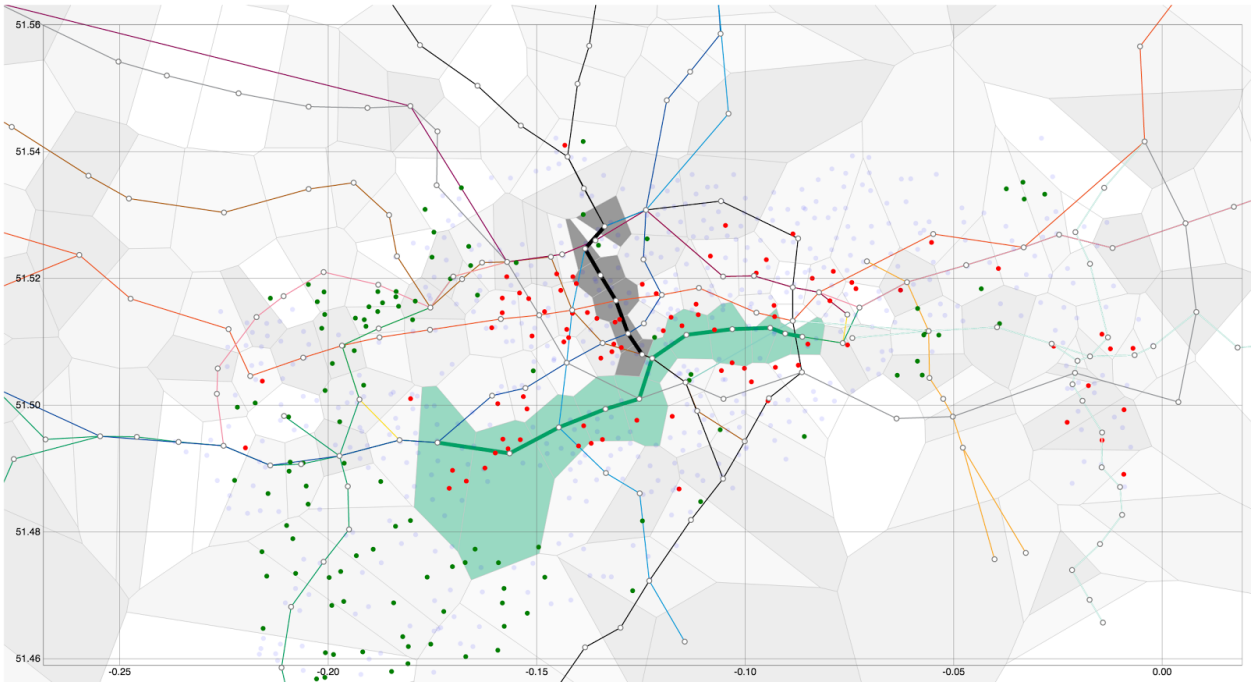


Fig 11 - District/Northern line closures

Fig 12 views the District line. Red colour of many nearby docking stations indicate that docks are becoming full as no passengers are renting bikes in these areas, number of full stations in close proximity could cause a large network effect.

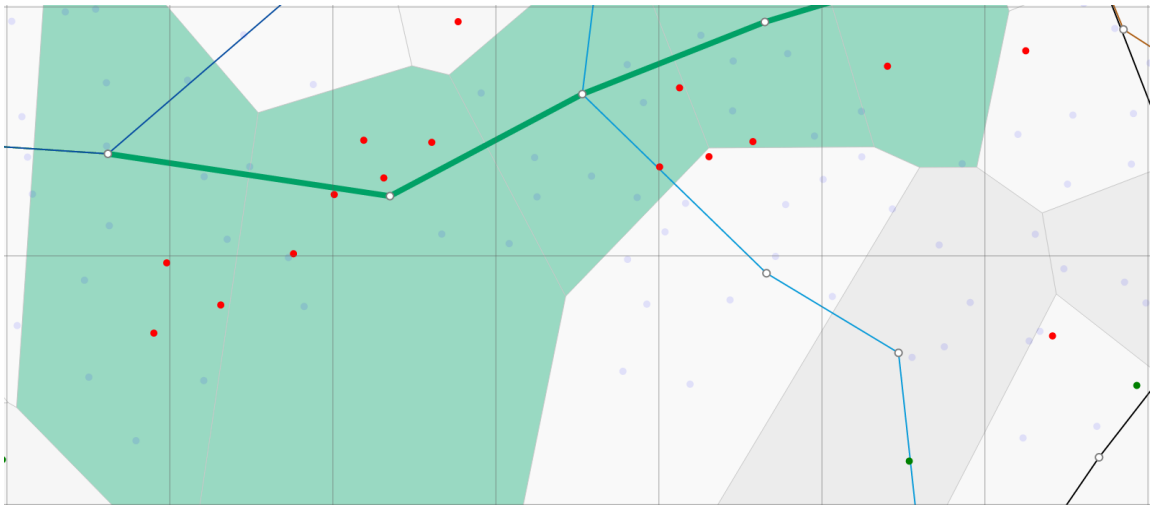


Fig 12 - District line zoom

4.3 Specification

Specification of the optimisation has been discussed in sections:

- B.9.1.1 Functional
- B.9.1.2 Technical

- B.9.1.3 Security

5 Conclusions and Future Work

In conclusion, we have produced an analytics platform which provides a multitude of visualisations. We collected our datasets, scraped the necessary data and normalised it before placing it all in a database. We then iterated through several versions of our system and produced visualisations beginning with a rudimentary graph based on Google Maps, before moving onto creating our custom framework and queryable API in an open source project called TubeMaps, granting us a more powerful tool to create our visualisations. The final iteration of the visualisation uses Voronoi areas around tube stations experiencing closures/delays to show the areas where bike docks will be affected.

We found a correlation between engineering works/delays and the BCH data, this lead us to a means of causation, as the engineering works/delays strongly affect the demand/supply of bikes to docks. The correlation and causation was reinforced by simulations with and without engineering works showing significantly different supply/demand of bikes from docks in the affected area compared to an average day.

Conclusion we can draw from this is that we can optimise the re-distribution of the bikes by providing data visualisations showing the network anomalies.

5.1 Potential Return On Investment (ROI)

We estimated cost of our software to ≈£10,000, with running costs of £30,000 per year (see Appendix D for full details). TfL BCH currently makes a loss of £20,000,000 - £30,000,000 per year.

$$E = \frac{\sum_{i=1}^{S} T_i P_i}{12S}$$

Developed formula analyses the efficiency of the BCH network by modelling the popularity of each docking station as a ratio of outgoing to incoming bikes for each hour of availability in the day. Please see section D.6 Return on investment (ROI) for a very detailed explanation.

Using this mathematical model, it is estimated that TfL can save £1,300,000 per year, a noticeable reduction of their current losses.

5.2 Contributions

The open-source contributions made to the field have been recognised by developers globally, with our public library for tube maps gathering 107+ stars on GitHub. Additionally, a developer in Toronto is currently extending our visualisation for their own transport network.

We also created an API that returns bike data in a normalised manner scraped from the TFL feeds.

Furthermore, engineering works data wasn't available as a public data source before we embarked on creating a custom library to scrape this data from an engineering works calendar web site. We have now released this information as open data on GitHub.

In fact, we have illustrated how easily our open source library can be extended (see Fig 13); we have created a Voronoi diagram for all the tube stations in Tokyo. This has been well received by Toshiyuki Nishino from 国立情報学研究所/**National Institute of Informatics.**

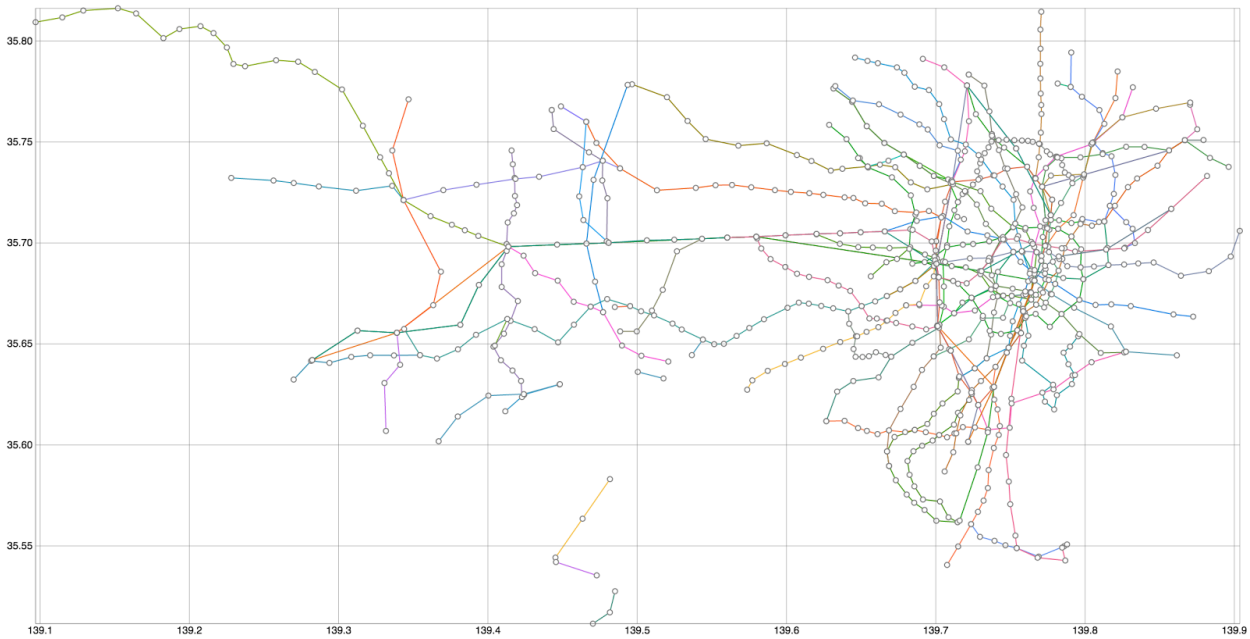


Fig 13 - Tokyo Tube Stations

5.3 Future Work

Going forward, there are a few desirable steps we can take. We would like to expand the platform so it can find correlations with any bike data and any other dataset (see Appendix B.12, 4th iteration). In addition, while we found a correlation there were some outliers that did not conform to the correlation. There is further, more sophisticated analysis required to find the cause of these outliers.

If this project were to be run again, we would change our analytics method to something more concrete (see Appendix B.13).

Appendix A
Research and Requirements Gathering

Appendix A - Research and Requirements Gathering

A.1 Research

The first thing that was carried out in the project was preliminary research. By doing so, we were able to get a better understanding of the problem and also identify meaningful ways to solve it. Moreover, this allowed us to build on top of the work of others and also helped us to determine the resources that were needed. Thus, we established a realistic goal.

In the following subsection we will present the work that was carried out during the research phase as well as a summary on how it helped us to shape the project.

A.1.1 Research Phase

Identifying the need to research into our domain, we first explored the TfL BCH in general so as to get an overview and a broad understanding of how the system worked [1]. Then, after getting a clear picture of the workings of BCH, we moved on to identify potential problems. The problems were initially highlighted when we looked into the financial analysis of the scheme [2,3] and then subsequently when we researched our client, where he confirmed our hypothesis and helped us narrow down to the root of the problem.

Having identified the problem, we moved on to examine the work conducted by others in order to learn from them and build on top of their work, but more importantly in order to contribute in a meaningful way and not repeat what has already been done. We read a thesis by Ryszard T. Kaleta [4] and then a follow up by Zhanzhan He [5] and on top of getting a lot of useful insights, we have realised that work on building an integrated journey planner had already been done before. Then, we investigated an article by the New Scientist [6] where they presented some impressive visualisations of the BCH journeys and concluded that people use the scheme as an interim form of transport to get to their work. More specifically, they use the tube to get to the stations that are close to business hubs and then they use nearby bikes that gets them to work.

By keeping the aforementioned information in mind, we have hypothesised that since users use tube stations before using the bike scheme, we have hypothesised that disruptions in the service of

---

[1] "Barclays Cycle Hire - Wikipedia, the free encyclopedia." 2010. 1 Dec. 2014 <http://en.wikipedia.org/wiki/Barclays_Cycle_Hire>

[2] "Barclays Cycle Hire - Wikipedia, the free encyclopedia." 2010. 1 Dec. 2014 <http://en.wikipedia.org/wiki/Barclays_Cycle_Hire#Finances>

[3] "TfL reveals how much Barclays has paid for Cycle Hire ..." 2014. 1 Dec. 2014 <http://www.mayorwatch.co.uk/exclusive-TfL-reveals-how-much-barclays-has-paid-for-cycle-hire-scheme/>

[4] Kaleta, RT. "An Integrated London Journey Planner - Department of ..." 2012. <http://www.doc.ic.ac.uk/teaching/distinguished-projects/2012/r.kaleta.pdf>

[5] "An Integrated London Journey Planner - Department of ..." 2013. 1 Dec. 2014 <http://www.doc.ic.ac.uk/teaching/distinguished-projects/2013/z.he.pdf>

[6] "Short Sharp Science: Tron-like map of bike journeys reveals ..." 2012. 1 Dec. 2014 <http://www.newscientist.com/blogs/shortsharpscience/2012/09/map-of-bicycle-journeys-reveal.html>

underground stations must inevitably affect the network flow of BCH. Nevertheless, after continuing to research, we have realised that to the best of our knowledge there has not been any research conducted to account for this scenario and hence we moved on to examine it further.

Going ahead, we looked into available datasets that would allow us to examine our hypothesis and then build a visualisation tool for TfL. We came upon the Google Maps Geocoding API [7], the BCH open data [8], the TfL tube data [9], the list of tube stations from OpenStreetMap [10] and the data from bike-stats [11]. Having identified our datasets, we selected the most useful ones and moved on to research development tools and frameworks.

In that regard, we realized that Beautiful Soup would be a good data scraping framework for our purposes [12] and then looked into database implementations. Amongst others, we researched and considered MySQL [13], MongoDB [14] and Hadoop [15].

After that, we investigated possible visualisation tools and considered using LeafletJS [16], the Google Maps API [17] and d3.js [18]. We have produced a lot of early iterations of the application using all the aforementioned visualisations frameworks but as you will see from other sections, our final product uses d3.js.

So, to summarise, we have looked into TfL BCH from a more general perspective and then we dived into examining the drawbacks of the system. Having identified some problems, we formed a hypothesis from the research that was carried out and confirmed it by interviewing a client which also helped us to narrow down the scope of the project. Then, we audited previous work that was carried out and besides gathering a lot of knowledge in the process; we have identified a domain of the problem that was not addressed properly. After that, we investigated potential datasets to use as well as data scraping and database technologies. Lastly, we explored tools that would allow us to produce meaningful visualisations of the data.

---

[7] "The Google Geocoding API - Google Developers." 2012. 1 Dec. 2014 <https://developers.google.com/maps/documentation/geocoding/>

[8] "Our feeds - Transport for London." 2014. 1 Dec. 2014 <https://www.TfL.gov.uk/info-for/open-data-users/our-feeds>

[9] "Our feeds - Transport for London." 2014. 1 Dec. 2014 <https://www.TfL.gov.uk/info-for/open-data-users/our-feeds>

[10] "List of London Underground stations - OpenStreetMap Wiki." 2010. 1 Dec. 2014 <http://wiki.openstreetmap.org/wiki/List_of_London_Underground_stations>

[11] "TfL Cycle Hire API | BikeStats." 2010. 1 Dec. 2014 <http://api.bike-stats.co.uk/service/rest/bikestats?format=json>

[12] "Beautiful Soup: We called him Tortoise because ... - Crummy." 2004. 1 Dec. 2014 <http://www.crummy.com/software/BeautifulSoup/>

[13] "MySQL :: The world's most popular open source database." 1 Dec. 2014 <http://www.mysql.com/>

[14] "MongoDB." 2008. 1 Dec. 2014 <http://www.mongodb.org/>

[15] "Welcome to Apache™ Hadoop®!." 2007. 1 Dec. 2014 <http://hadoop.apache.org/>

[16] "Leaflet - a JavaScript library for mobile-friendly maps." 2012. 1 Dec. 2014 <http://leafletjs.com/>

[17] "Google Maps API - Google Developers." 2012. 1 Dec. 2014 <https://developers.google.com/maps/>

[18] "D3.js - Data-Driven Documents." 2010. 1 Dec. 2014 <http://d3js.org/>

A.2 Client and Customer Interviews

A.2.1 Client Interview

TfL optimisation requirements interview - Optimisation of TfL Barclays Cycle Hire bike distribution during planned engineering disruptions.

Our team, as previously mentioned, had set the goal of researching the impact that planned engineering disruptions have on the distribution of the TfL BCH bikes.

So, to confirm that our goal is meaning, useful and realistic, we sought out to meet with an external client who is involved with TfL and is thus more knowledgeable and experienced in this domain. Hence, the client that we have chosen is Dr. Stephen Pryke of The Bartlett School of Architecture [19]. Dr. Stephen Pryke is involved in a project to improve TfL's competence in the design, implementation and evaluation of collaborative teams, through Organisational Network Analysis. Thus, we have deduction that he is an appropriate client to interview for our needs.

The following is the interview that was conducted on the 21st of October 2014 at the Bartlett School of Architecture:

Questions and Answers provided by Dr. Stephen Pryke
Length of the interview - 40min
Interview conducted on 21 Oct. 2014 at 16:00

Q1. What is the current main optimisation problem that the Barclays Cycle Hire network faces?

A1. Currently there are no problems with Bikes in my opinion; however there are problems with the distribution of them especially in central areas. I would like to point out that the places where racks are full are not near station, these places are more random and this would be a good thing for you to analyse.

Q2. How is data currently aggregated and analysed by TfL to model the network?

A2. I don't know exactly but I can get in touch with people who are working with TfL's data and I will email you their contacts.

Q3. What successful optimisations have been implemented by TfL in the past?

A3. There wasn't any optimisation lately but what is happening is that TfL started analysing data based on the number of bikes that have been damaged. As an outcome they found that there are bike stations which are too close to the road and these stations are producing the damaged bikes. So far, they tried to move the bike racks further away from the road. TfL is losing a lot of money mainly on the expensive costs of repairing these bikes.

Another one of TfL's problems is the communication channels about faults whether for the bikes or tube. There isn't any real system which controls that and even staff working on the stations most of the times they are not aware of any problems at the station. Most of the data online which shows the statuses of the tube stations are not real-time as there is a great deal of delay until the correct data is displayed.

[19] "Dr Stephen Pryke - Iris View Profile - University College ..." 2012. 1 Dec. 2014 <https://iris.ucl.ac.uk/iris/browse/profile?upi=SDPRY73>

As already outlined, the problem of re-distribution of the bikes has not been optimised yet because the system is copied from the Montreal network and it's not easy to change.

Q4. Is our goal of the Project suited to be a real problem TfL might face when optimising the bike hire network?

A4. It is a great proposal and great idea to work on especially because TfL has never worked on it before and there isn't anything that can be found online.
I believe there can be a very interesting outcome once these two sets of data are analysed and visualised and I would definitely like to see it. The use of engineering works data is very interesting and you should be able to find correlations and come up with a good analysis.

Q5. Can we get more data from TfL directly? The current status updates calendar only shows current and future events, is there another way to access historical engineering works data?

Q6. In addition to that, if the historical data exist, how far back in the archives will we have to look to get a good amount of past data on how planned engineering works affect the hiring of bikes?

A5 & A6. I cannot provide you with any data but I can get you in contact with people who have access to TfL's data and should be able to answer this question.

Q7. Do you have any suggestions for secondary datasets to look at? (On top of planned engineering works etc.)

A7. Engineering is a very good area to work on but also look at the weather if possible and see where people are leaving the bikes the most. Are they taking bikes at all or are they leaving them closer to the stations or is there no change?

Q8. Can we get some more information on the temporary racks? (How quickly they can be set up etc.)

A8. Temporary stands have never been used or even considered, therefore the price or the speed they can be brought up is unknown but it's a good idea especially when there are stations with high demand at peak times. There are a number of bike stations which cannot be expanded because of the space or infrastructure limitations, therefore it would be a good idea to consider temporary stations and how they would work.

Q9. What are some key pitfalls that are faced by the current cycle hire system?

A9. In my opinion there is definitely a problem with re-distribution of the bikes. There are safety issues and all users around London should be wearing a helmet preferably provided by TfL (or buying it cheaper) or a new solution which acts like a helmet but it's a scarf around the neck. The scarf reacts to drastic changes and covers the head. People usually are not aware of the fact that they can use a bike to get somewhere quicker therefore it would be good to have an application outlining this stuff.

Additional ideas:

- Develop an application for users and staff which sends short messages informing of any delays or problems for both Tube and Bikes.

- Give people a choice to rent a bike and drive to selected bike racks for free between peak hours. It would partly solve the problem of re-distributing of the bikes but we may have to wait for the response from the users.
- Public health problems should be taken into account because of the lack of helmets.

A.2.2 Customer Interview

Questions and Answers provided by Pejhmon Kamaie
Length of the interview- 30min
Interview conducted on 29 October 2014 at 11:30

Q1. How regularly do you cycle a week?
A1. I cycle for about 2 to 4 hours a week, each trip taking 5-15 minutes.

Q2. How many (return) trips do you make a week?
A2. About 5 to 7 trips, 1 every weekday, as I commute to university.

Q3. Does the 30 minute time limit affect how long you cycle for?
A3. No.

Q4. Does the 30 minute time limit affect how far you cycle?
A4. No.

Q5. At what times do you normally cycle?
A5. 08:30 to 09:30, and then 17:00 to 19:00.

Q6. Do you find it hard to acquire a bike or find an empty spot?
A6. Yes, after rush hour, all bikes are taken. Not finding an empty slot isn't as frequent.

Q7. If you do, or have ever had an issue with this, how do/did you get around this?
A7. I just go to the next nearest stop with either spare bikes or free space.

Q8. Do ever take the tube or bus?
A8. Only really on the weekends, if I am traveling with someone.

Q9. What if you can't cycle, for any reason, maybe the bikes are all gone during rush hour? What do you do in such a case ?
A9. It depends on the distance; I tend to walk over to other forms of transport.

Q10. Have you ever cycled instead of using other public transport due to a closure such as planned engineering works?
A10. Yes, but it doesn't happen frequently.

Q11. If you have, have you considered doing it more frequently?

A11. I would, but my destination tends to be outside of bike route.

Q12. How frequently do you check for travel delays before departing for a journey?

A12. I never really do.

Q13. Do you bike to a certain location (like the tube) and then continue the journey by other means?

A13. Very rarely.

Q14. Why don't you do it more often?

A14. It's normally close enough to cycle.

A.3 Lightweight Requirements Shell

Having extracted requirements from our research as well as the interviews that we conducted, we proceeded to form the lightweight requirements shell that is listed below.

ID	Description	Rationale	Type
LR1	The system should ensure that enough bikes are available at all stations.	Users will not be able to cycle from the station otherwise.	Functional
LR2	The system should ensure that several stations close to each other are not all full.	Users will have trouble trying to find an empty rack to deposit the bike otherwise.	Functional
LR3	The system should alert bike redistributors to fill up stations that are known to empty quickly during specific times.	Users will be able to travel unaffected by stations with high demand.	Functional
LR4	The system should alert bike redistributors to remove bikes from stations that are known to fill up quickly during specific times.	Users will be able to travel unaffected by stations with high demand.	Functional
LR5	The system should predict which stations will have an increased demand for bikes within a time frame due to link closures.	Users will be able to travel unaffected by transport link alterations.	Functional
LR6	The system should predict which stations will have an increased demand for free racks within a time frame due to link closures.	Users will be able to travel unaffected by transport link alterations.	Functional
LR7	The Cycle hire system should not look different from it's current implementation.	Users do not need consider that the system will function any differently for them.	Non-Functional

LR8	The system should have past as a well as future (at least a year) planned engineering work dates.	The system needs to have sufficient amounts of data for predictions.	Functional

A.4 Personas and scenarios

From the gathered data, we created two personas to represent the different archetypal users in the following scenarios. This allowed us to better imagine what our end user will be like and design our application accordingly.

A.4.1 Persona & Scenario 1 - Daniel

Behaviour

- Cycles to and from Uni every day
- Use cycle hire instead of using own bike

Demographic

- Student
- 18 - 30 year old
- lives in flat relatively close to uni

Needs & Goals

- Limited income (student loan)
- Wants to get to university ASAP
- Wants bikes to be available

8:30 am, Daniel wakes up on a Tuesday morning realising that he's going to be late to his 9am lectures. He gets dressed, grabs a piece of toast and rushes out the door without checking if his commute to university has any irregularities today. Being a student, he prefers to take a bike from the cycle hire station next to his local tube station rather than take the tube itself, saving him enough to party on the weekends. Unluckily for Daniel, his local tube station is currently undergoing planned engineering works. Like many others, Daniel didn't know about this planned closure and as a result the bike dock is completely empty. If no bikes were at the station, Daniel would normally take the tube to avoid being late, but the reason there are no bikes is because the station is closed.

Unfortunately, the cycle hire system was not informed of this planned closure (and therefore the sudden increase in demand of bikes). Even as a re-distribution vehicle with fresh bikes arrives there is simply not enough bikes for the growing crowd of angry commuters.

A.4.2 Persona & Scenario 2 - Brian

Behaviour

- Drives to train station, cycles when he arrives in London
- Uses cycle hire every weekday (if they are available) else gets a taxi

Demographic

- Inner city worker
- 30 - 40 year old
- lives with a wife & 2 children

Needs & Goals

- Income £75,000
- Doesn't want to be late for work
- Is always looking out for opportunities to exercise - however, will not compromise being late as a result

After a brief but stressful commute to the train station, Brian parks in a nearby car park and takes a train to Euston. Due to his hectic lifestyle, Brian tries to keep healthy by taking a bike to work from the station. However, one of the underground lines that goes from Euston station is currently undergoing engineering works, and Euston is the last stop on this line for the time being.

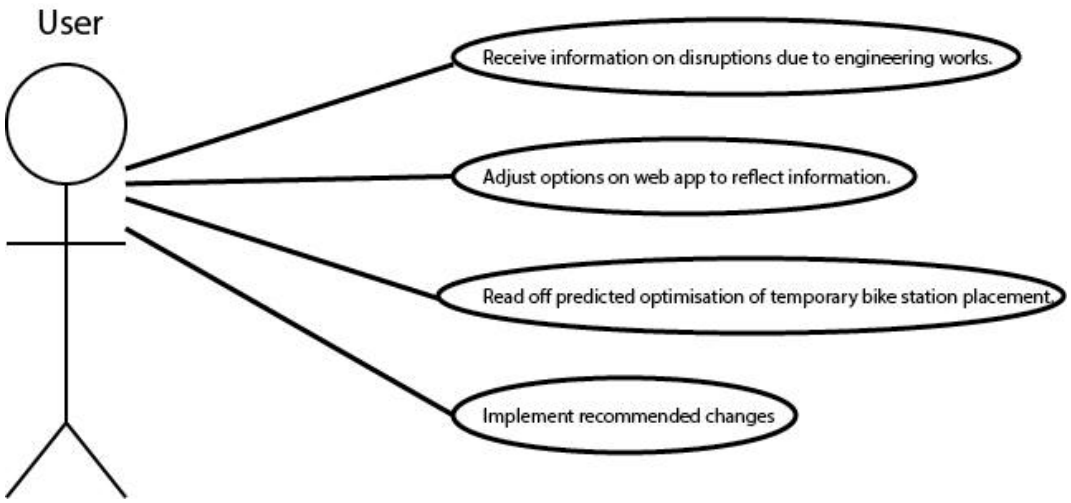
As a result, there is now an unusually high demand for bikes, as other commuters have to finish their journey by cycling. Brian is well organised, but as he doesn't take the underground, wasn't aware of the engineering works until arriving at the station.

Barclays bikes are not available for hire from the docks near the station as the operators of the cycle hire system didn't foresee this situation, meaning that Brian can't continue his commute to work by bike.

Upon hearing the news, Brian is forced to get a taxi from the local rank as he can't be late for work.

A.5 Use case

The following use case depicts four possible uses of our application by our end-users.



A.6 Research Findings

- Short term disruptions cause a lot of grief for users of the system due to a lack of bikes to cover demand produced by a closed transport link.
- There is a lack of communication between the Cycle hire system and TfL in regards to the planned engineering works for the underground.
- It can be assumed that users of the system are not aware of its current state. As such, they are caught unprepared.

Further evaluation of requirements we gathered can be found in Appendix B.2

Appendix B
Development Plan

Appendix B - Development Plan

B.1 Overview

In the overview, we will briefly restate what the project is. We were asked to produce a system which optimises TfL BCH. Our team has set the goal of researching on the impact planned engineering disruptions have on the distribution of the TfL Barclays Cycle Hire bikes. It is our plan to collect and process data from multiple sources to investigate this hypothesis.

The TfL open data API provides a comprehensive set of data for the Barclays Cycle Hire scheme allowing us to model the usage of the bikes and track when docking stations have bikes available or not. The status update calendar provides us with some guidance on which lines and stations are affected during works.

Throughout this section, we will cover the following:

- Project deliverables
- Project risks and opportunities
- Estimates
- Project resource information
- Project delivery method
- Configuration and change management

B.2 Software Requirements Specification

In order to keep on track and have an overarching view of our final goal it is important to detail and lay out all the requirements of the software we developed. Once we had a general idea of what we wished to do to tackle the issue at hand, we sat down and decided on the following requirements (taking into account the data we collected from user interviews):

B.2.1 Functional Requirements

- **The system should always ensure that bikes are available at stations.** This is a vital part of optimizing the network, ensuring that there are always bikes available enables the shortest interaction time for the TfL customers and it means that they will have a bike as soon as they want one, increasing the flow of the network.
- **The system should ensure that stations within close proximity of each other have at least a few free spaces.** The rationale behind this is that there are certain stations in London that always have a high chance of being full (for example stations near a busy commerce hub will be very likely to have few free spaces). With this in mind the system should direct bike redistributions so that there are at least some empty docking points for a Barclays bike within a pre-determined radius.
- **The system should direct the redistribution such that priority is given to stations that are known to empty quickly**. This is important because no two stations are the same and the flow of bikes to and from them can vary vastly depending on its location and what else is nearby. The

system should have indicators for stations which are very popular and have a large flow of bikes away from it, and make sure that these are paid attention to and stocked up again as and when necessary.

- **The system should alert bike redistributors to remove bikes from stations that are known to fill up quickly during specific times**. This is the counterpart to the requirement above. To ensure optimum network flow we will need to add bikes to places where they are in high demand. These bikes will have to come from somewhere and the most logical solution would be to take them from stations that are popular places to drop off bikes. The result being that at locations were spaces are in high demand; they are available more often, with these places being taken to locations in which the bikes themselves are in high demand.
- **The system should predict which stations will have increased demand during pre-defined events, such as engineering works.** This can be broken down into two sections:
    - **The system should predict which stations will have a higher demand for free racks, and the scale of this demand**. Our focus is on how engineering works affect the TfL bike hire network and so we need to be able to see how the closure of a tube station will affect the bike network flow around it. If the system predicts that people will redirect themselves to drop off their bikes at a new location, it should notify the users of this so a more appropriate redistribution strategy can be undertaken.
    - **The system should predict which stations will have a higher demand for bikes.** Again, the behaviour of people will change due to the tube line closures and in being able to predict for this and redistribute the bikes in such a manner the cycle hire network can be kept running as optimally as possible, we will be fulfilling our project directive.
- **The system should have data on past engineering works as well as ones that are planned in the future**. To be able to predict upcoming changes to the cycle hire network data on its efficiency and running is needed, so that this can be correlated with engineering works allowing the system to have a clearer understanding of how the cycle hire scheme and tube network affect each other. The more past data is available, the more accurately the system can predict future changes in the cycle hire network.
- **The system should provide a number to use as a metric to represent the flow of the network.** While we can optimize the network as much as we wish we need a way to be able to see how optimally the network is running. The system should produce a coefficient which can be compared to previous coefficients (a number ranging from 0 to 1) which provides an easy way to gauge how the network is faring compared to the average.

B.2.2 Non-Functional Requirements

- **The user facing part of the TfL cycle hire system should remain unchanged.** People are already very used to the cycle hire network as it is and are generally resistant to change. We aim to keep the process of hiring the bikes exactly the same as before, and optimizing instead where the bikes are available, to ensure a smooth transition from the previous system to our new proposed one.

B.3 Need for project

This necessity of the project is to verify what can be changed or improved inside the bike's network to optimise it for both user's and TfL's staff. Our external client Dr. Stephen Pryke has verified the need of

this project. He stated that this can be a very important analysis and the outcome can be interesting. If the analyses are successful and we reach definite conclusions, our client will pass it onto TfL. We have undertaken this project and system to develop, keeping in mind important facts such as:

- Cost savings
- Benefit for users and company (TfL)
- Improving communication channels inside TfL
- Improvement of the system

B.4 Challenges

Here is a list of challenges that can impact our project planning and execution:

- Short amount of time until the end date for the report
- Unknown lead/delivery time frame for a key project component which are analysis (this would directly impact the scheduling critical path)
- Unknown amount of data needed for good conclusion
- Technological limitations (lack of knowledge)
- Bugs in the system (testing shall help detect them)
- Group's communication, group has 12 members and we need to produce a lot of work
- If we going to produce an application or website for the visualization we have to try eliminating possibility of application crash (large flows of data may cause app crashing)
- Satisfying all the requirements and working with close deadlines
- Getting correct data sets for specific dates or range of dates

B.5 Opportunities

By implementing the project, we will definitely provide a good piece of analysis and hopefully the application will be helping users and TfL. Using our visualisations, which will show which bike stations are overloaded, we will indicate that TfL staff needs to move them to different station. Here are the main opportunities:

- The system will be uniquely analysing data based on two or more data sets such as scheduled engineering works and bike usage.
- The resulting product may be deployable and become live once we prove the concept of our work.
- If our clients decide to make it a real product and distribute it further there is definitely a market for it and it will optimise entire network.
- Analysis can be used for both bike users and TfL.
- Analysed data with clear conclusions should be able to improve TfL's network and bring some profits or at the very least reduce losses.

B.6 Initial analysis

B.6.1 Recognition of a problem- 5 whys approach

The BCH scheme has some problems about renting bikes. For example, "users never return the bike to the initial docking station" and "disorderly distribution of bikes at the docking station within the network". Especially, disorderly distribution of bikes is big trouble, because this requires "re-distribution" by someone. Therefore TfL need to pick up "used" or "dropped-off" bikes and move them to places where

they are most likely to be hired again. This "re-distribution" problem causes an increase in cost for the bike-provider and the inconveniences of hiring for bike-users. So we think it's worth fixing this problem.

B.6.1.2 Analysing the Problem

For understanding this problem well, we applied the 5 whys method. The 5 whys is a method to get at the root cause of a problem by an iterative question-asking technique. By applying the 5 whys method for this problem, we will find the root cause of this problem and take measures to relieve the situation. There may be a lot of things that caused the "redistribution" problem. So we need to pick up the effective and realistic one or two things which we can treat.

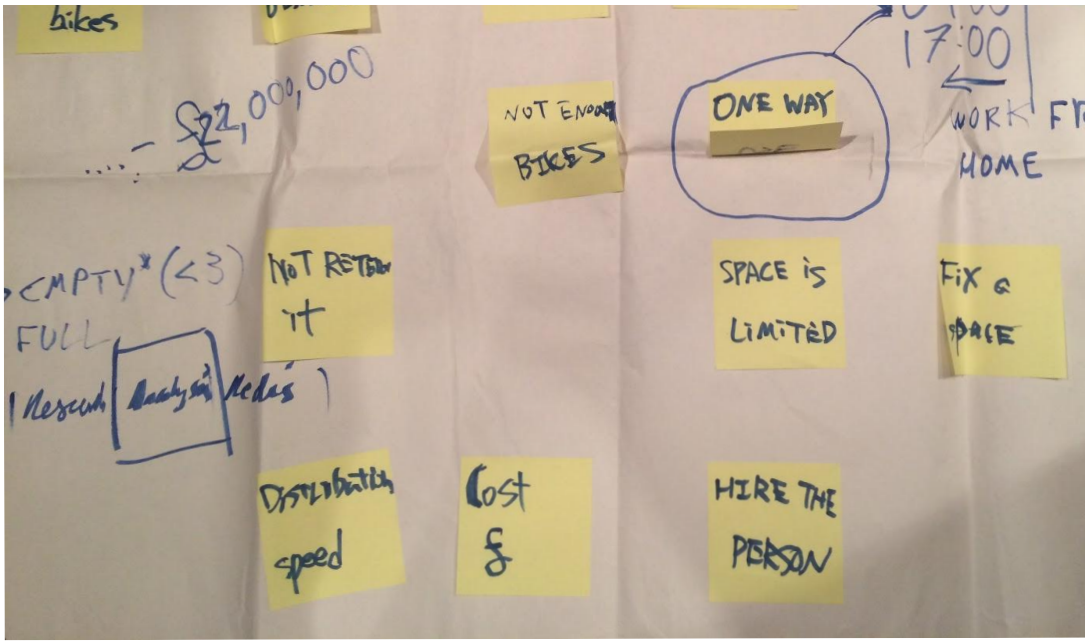
At first, on applying the 5 whys, we presume that "the lack of bikes at specific stations" is the problem. Then we tried to apply the “first why” and arrived at some causes. The results of the “first why" are the following:

- too much demand
- users don't return the bike to the original station
- users leave the bike anywhere
- users don't come from another station with a bike
- less frequent replacements by provider
- bike is stolen
- bike is broken

The results of “second why” for “too much demand” are the following:

- too many users exist (a usual situation)
- the number of bikes at the station are not too many
- users can't use other transport facilities (an unusual situation)

After we have repeated "why" for analysing these causes, we have found that the major root cause of the "re-distribution" problem is "one way use".



In order to get a better understanding of the situation of "disorderly distribution of bikes", we developed an analytics application that will detect anomalies and identify causes by displaying informative data visualisations. This application is also useful for Serco Group which is the company doing manual re-distribution of the bikes.

B.6.1.3 How we can use the 5 whys to talk about commercial viability

The 5 whys is the method which divides a problem into some small causes. This division is very useful for seeking the root cause or the core of the problem, because the things become smaller and hence things become simpler.

Commercial viability is a big and complicated thing. So, for talking about commercial viability, we need to clarify and define the most important things.

For applying the 5 whys method, there are some important points we need to consider. We need to:

- Define the theme that we are analysing before starting to analyse
- Confirm the constraints before starting to analyse
- Express a phenomenon in a simple word as much as possible
- Keep the logical connection between the hierarchy of “why”
- Check the actual things; actual place; actuality. (to avoid assumptions and logic traps)
- Continue to repeat "why" until you can take measures to meet the situation
- Don’t guess the cause or the core of the problem at first (“When you have eliminated all which is impossible, then whatever remains, however improbable, must be the truth.” - Sherlock Holmes)
- Don’t use “if” (the cause without proof is no more than a guess)

B.7 Team division

In order to optimise the productivity of the team we divided it into 3 sub-teams which are:

Team Leader	**Jonny Manfield**			
Engineering Lead	**Nicola Greco**			
Project Management	**Marcin Cuber**	Ran Gutin	Edward James	Toshiyuki Nishino(NII)
Data Science	**Christodoulos Demetriades**	Richard Isaac	Keqin Feng	
Development	**Navid Hallajian**	Mohan Dai	Michael Detmold	Ragavan Guneshalingham

Key: **Sub-team leader**, (NII) - □□□□□□□□ National Institute of Informatics

B.7.1 Explanation of Team Roles

Team Leader: The team leader will be in charge of leading the project management team, in addition to communicating with the engineering lead and other team members to ensure that the project remains on track and moving forward.

Engineering Lead: The person placed in this role will be organising and planning with the others in engineering roles. He will be the main point of contact between the management team and the

engineering team and will delegate tasks and roles so that everyone's specific talents are put to the best use.

Project Management: This is the project management team. They will deal with keeping the project heading forward in a smooth manner, and with deal with matters such as planning the project, outlining clear goals and aims and reshuffling the project when inevitable problems in the development occur. They will have a very dynamic and reactive role to ensure that the project is delivered on time.

Data Science team: This team will be analysing the data that we have available to us for this project, as well as searching for new datasets. They will be the key to finding correlations within the data and to pinpoint factors which significantly affect the efficiency of the cycle hire network. This team will have to perform a lot of data analytics and aggregation.

Development team: This will be the team in charge of developing the system. They will communicate with the data science team and the project management team, combining the output of the two teams into a final deliverable product.

B.8 Project Objectives

These are the criteria which will be used to measure project success:

- Complete analysis of data and conclude the project with report and video and additionally a software
- Establish standards, implementation and management guidelines
- Provide clear justification of conclusions
- Provide clear visualisation of analysed data
- Provide a system which is easy to use
- Provide a clear documentation about the analysis
- Develop a website for visualisation or an application
- Produce diagrams which clearly explain the architecture of potential application or website

B.8.1 Project Constraints

Our project is related to quality, scope, and timeframe and in small parts with budget (especially minimising it). We have attempted to minimise each of those constraints and identify them as soon as possible. Knowing these specific constraints, we kept in mind that it will be challenging for us to work with tight deadlines. However, with a good plan, great organisational skills and a team leader we should be able to overcome any problems. We were aware of these constraints and we had to make sure our client is also aware of them as they may pose an adverse risk to successful project completion.

Additional constraint can be the fact that we are producing it for project managers, therefore for good understanding we need to produce a clear documentation, which can be taken further into development and maybe implementation.

B.8.2 Risk Management

In this part, we will identify and qualify all project risks, which are:

1. Technical knowledge for both understanding the system and understanding how to program hardware or software
2. System will be using two data sets and therefore joining them could present a risk of failure
3. Software and package selection for production of the solution
4. Extremely short timeframe, risk of not finishing on time
5. Large teams don't usually work efficiently

6. Applications/website in real-time needs to be working fast without crashing therefore algorithm needs to be optimised
7. Lack of communication and organisation

Risk 1:

This risk relates to everyone in the team. Each of us has a different skill set, therefore clear selection of tasks and management should minimize the risk of failures. Failure to discuss and manage responsibilities, which match our skills, may lead to great delays. A delay in this particular project may lead to complete failure because of the short timeframe. To minimise this risk we already spent more time at the meetings discussing who is doing what and making sure we are all doing things we can do. Each of the twelve members of the group has been assigned with a task that they can succeed with and ideally we shall not be wasting our time doing wrong designs.

Risk 2:

Directly links with our possible solution. We are going to use different types of data sets such as weather, maps engineering work etc. We are constrained with the short timescale and it is difficult to tell how many we are going to use. However, the risk in this case can be the fact that we won't get any correlations between data. Another potential risk was that we would not be able to find good data sets but we have in fact already minimised this risk as we have found sound datasets from our research phase.

Risk 3:

Directly links with the idea of using specific software and making right decision of who is working with what. Throughout group meetings we discussed this problem and we came to the agreement of which software is going to be used. The careful consideration of the use cases gave us an advantage and we minimised the risk of failing with software choice.

Risk 4:

It is a very serious problem for which our team had to meet a number of times to establish a careful plan of action. We established two meetings a week to keep on track with our development and this surely should help us in meeting the deadline. Additionally, weekly reports will help us judge what has been achieved and what are the plans for upcoming weeks in order to start each week with a sense of purpose. This risk is also directly aiming at individual group members who have to have a defined plan of action for each week; this should help them keep on top of their work.

Risk 5:

It belongs to team construction. For this project we have a team of 12 members. This can be problematic when working on a project like this, since after all it has been proven that smaller teams are more efficient [[20]]. To minimise the risk of having communication issues or any other in-team problems, we divided the team into 3 sub-teams and each sub-team has their own leader. This has been very important and very well oriented step and currently we are working much more efficiently and faster. We may encounter problems such one of the sub-teams has to wait for other sub-group to complete their work. However, this problem has been largely minimised through the regular meetings we are holding each week (at least once a week).

Risk 6:

It is the risk that is being faced by everyone working on designing websites/apps. To minimise the risk of failing this one we are using an iterative approach, therefore each feature or function is being

[20] "Team Size Can Be the Key to a Successful Software ... - Qsm." 2011. 1 Dec. 2014 <http://www.qsm.com/process_improvement_01.html>

carefully tested and bugs are being detected. However, testing functionality which is being implemented is the most important part of the system; therefore we must make a good effort when testing parts of the system at implementation stage.

Risk 7:

Communication and organisation in large groups can play a crucial part. The system we are working with may become very complex and one person will definitely not manage it. To minimise the impact of this risk we are having regular meetings and division of the team. Regular meetings will help us recognise each other's strengths and weaknesses. Grouping people into selected sub-groups can be very effective because they will be more productive and also provide skills for the rest of the team that other members don't have.

B.9 Architecture structure

This section will cover all the technical details of how the system is running.

B.9.1 Architecture

B.9.1.1 Functional Specifications

The back-end team was tasked with providing data to the visualisation team. At a high level, the backend team was set to gather and normalise data, making it available to the visualisation team through an API.

The primary data sets were tube closures, static data on where cycle racks are positioned, live cycle data on the number of available bike spots and bike journey data.

There was no direct way of collecting the tube closure data. A website had this data on a web page, and the pages were scraped to gather a year's worth of closure data. This was imported statically by the visualisation team for their visualisations.

The static and live data was provided by the TfL cycle hire API. This API returns data about various static properties of cycle racks such as longitude and latitude and install date. Live data was simultaneously supplied, indicating how many free stops there are for a cycle rack at a given point in time. The feed updates every 3 minutes. No historical data was available in this format, so the feed had to be scraped on a regular basis. This was achieved through a Python script [21] that polled the API and stored the raw XML data. A cron [22] job was set to run this script every 3 minutes. One week's worth of data was collected in this manner, resulting in over 2 million rows of data. This data is provided through the backend API to the visualisation team.

Journey data was available in a CSV format for a small subset of bike journeys over the span of two years. This data is again imported statically into the visualisation tools.

The static and live data was stored in a remote PostgreS [23] server. Postgres is a DBMS that is queryable through SQL. Schemas were made for both the static and live data. A Python script was written to go through the scraped data retrieved through the TfL API to extract the data as defined by the schema.

Once the data was imported into the remote database, a middleware layer was written in node.js [24] that provided our custom API to the visualisation team. This made use of the node-pg [25] library that

---

21 "jonnymanf/3001 · GitHub." 2014. 4 Dec. 2014 <https://github.com/jonnymanf/3001/blob/62ea4ad63f7734b059f615e942f61ca7b11d2c9e/live-scraper/liveDataExtractor.py>

22 "Cron - Wikipedia, the free encyclopedia." 2004. 4 Dec. 2014 <http://en.wikipedia.org/wiki/Cron>

23 "PostgreSQL: Servers." 2011. 4 Dec. 2014 <http://www.postgresql.org/about/servers/>

24 "Node.js - Wikipedia, the free encyclopedia." 2010. 4 Dec. 2014 <http://en.wikipedia.org/wiki/Node.js>

interfaces with Postgres servers, and other standard Node libraries to provide a RESTful interface to the data. This allows users to pass the parameters of their queries into a URL and run a get request to retrieve the data. Data is returned in a JSON format given that the visualisation team has opted to use Javascript for their front-end code and Javascript has rich methods to interact with JSON data. Other languages also have mature libraries to deal with JSON.

This method was chosen because of its versatility - it allows any programming language to make use of our API that supports running get requests against servers. This was preferable to providing specific libraries for programming languages, since the code base is contained to one server application, and support only needs to be given for the API as opposed to a host of different libraries.

B.9.1.2 Technical Specifications

We used D3.js which is a powerful tool for creating data visualisations. It has been selected out of many tools mainly because it works on the web. We have considered Manyeyes library [26], however it was lacking graphical flexibility. Other possibilities were Prefuse [27], Flare [28] and Quadrigram [29] which are nice but none of them can run in a browser without a plugin. D3.js is making use of JavaScript without using any plugins and its one of the reasons we selected it. Another advantage of D3.js is its flexibility [30]. Since the library works seamlessly with existing web technologies and can manipulate any part of the document object model, therefore it is flexible and can be used as the client side web technology alongside HTML and CSS. Giving us this power, we were not limited to working on small regions of a webpage like in Processing.js [31], Raphael.js [32] or SVG based libraries [33]. The D3.js library also takes advantage in the region of built in functionalities that the browser has, therefore it has greatly simplified our work, especially for mouse interaction.

By selecting this library it was important to acknowledge its disadvantages and thus its limitations. So, we found that Document Object Model (DOM) manipulation can be extremely slow for large numbers of entries. Nevertheless, SVG also has performance limitations when dealing with large data sets. However, with D3.js we took the idea that good data visualisation rarely requires drawing these quantities of elements on the screen. More importantly we have selected D3.js because it has tools that make the connection between data and graphics easy without using pre-built charts that limit creativity.

Also, we have found that the data visualisation company Datameer [34] officially uses D3.js as its core technology and additionally The New York Times uses it for rich graphs. Essentially for us D3.js has been extensively used for GIS map making [35], managing both GeoJSON [36] and TopoJSON [37] files. In

---

[25] "brianc/node-postgres · GitHub." 2010. 4 Dec. 2014 <https://github.com/brianc/node-postgres>
[26] "Many Eyes - IBM." 2013. 1 Dec. 2014 <http://www.ibm.com/software/analytics/manyeyes/>
[27] "prefuse | interactive information visualization toolkit." 2006. 1 Dec. 2014 <http://prefuse.org/>
[28] "Flare | Data Visualization for the Web - Prefuse." 2007. 1 Dec. 2014 <http://flare.prefuse.org/>
[29] "Quadrigram." 2011. 1 Dec. 2014 <http://www.quadrigram.com/>
[30] "Why D3.js is So Great for Data Visualization | Visually Blog." 2013. 1 Dec. 2014 <http://blog.visual.ly/why-d3-js-is-so-great-for-data-visualization/>
[31] "Processing.js." 2009. 1 Dec. 2014 <http://processingjs.org/>
[32] "Raphaël—JavaScript Library." 2008. 1 Dec. 2014 <http://raphaeljs.com/>
[33] "javascript - Svg charting library - Stack Overflow." 2009. 1 Dec. 2014 <http://stackoverflow.com/questions/793808/svg-charting-library>
[34] "Datameer - Wikipedia, the free encyclopedia." 2010. 1 Dec. 2014 <http://en.wikipedia.org/wiki/Datameer>
[35] "Geographic information system - Wikipedia, the free ..." 2003. 4 Dec. 2014 <http://en.wikipedia.org/wiki/Geographic_information_system>

fact, our visualisation is making use of GeoJSON and therefore we have selected a tool that can work very well with it.

B.9.1.3 Security Specifications

Security is an important consideration for a project of this scale, due to the value of the datasets we are using in our visualisations. As it is to be used by TfL to make decisions regarding how they re-distribute bikes if the system is subject to malicious attacks, this could have a serious effect.

To mitigate against this risk, we can isolate certain sections of the architecture so they are not accessible.

- Queries that aggregate and analyse the data are made via an API, this helps to constrain the type of query that can be run.
- The actual simulations and data visualisations can be run in a Virtual Machine once they are provided to the client; this prevents any edits to the visualisation once it has been generated.
- The application will be developed in a separate development environment and will also have a separate testing environment. This ensures that the most robust version of the application is on our deployment server at any time.
- The server will be regularly monitored and the logs will be regularly checked for abnormalities.
- Firewalls will be used on all endpoints, including servers and desktops, to prevent illegal access as much as possible.

B.10 Testing

The testing process undertaken in our project did not incorporate any special tools or methodologies such as unit testing. Due to the agile approach we took, as detailed in appendix G, we used multiple iterations of our project and testing while developing each individual module. By having multiple iterations allowed us to identify bugs and weaknesses in our code within iteration and between iterations. This meant that unit testing would be too resource intensive and provide little more aid in our project that wouldn't be provided by our other testing, and so we found it to be unnecessary.

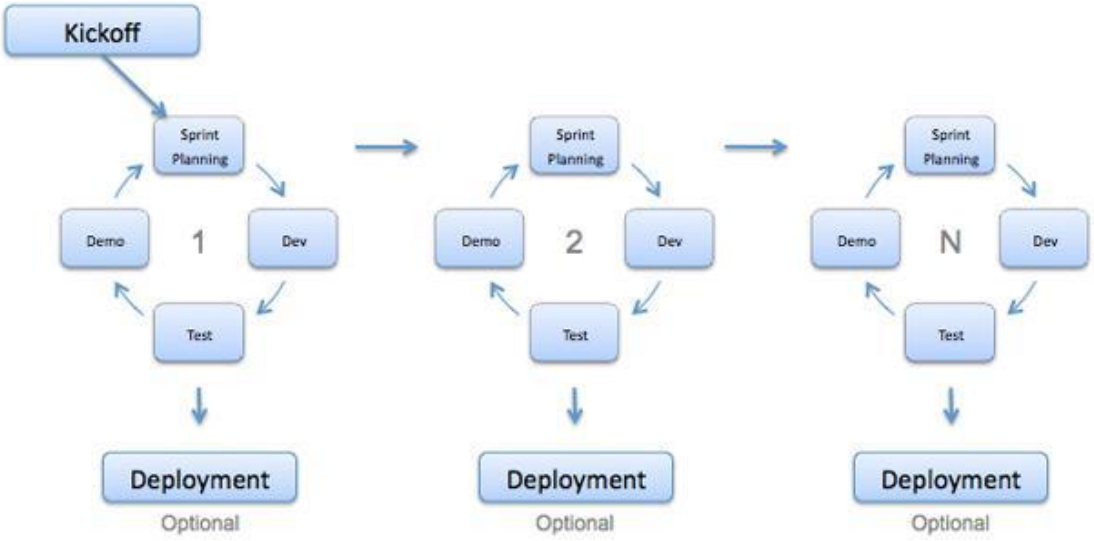
B.11 Project Management Approach

Section B.13.1 contains information on the development model we used. Additionally, we have included configuration and communication management which indicate the ways we managed both the group and development of the system. The configuration section is containing settings that will be necessary before deployment. However, the full configuration or implementation can be found in Appendix B.13.2. The communication management section explains all the communication links we had inside the team.

B.11.1 Development Model

In this section we will introduce a simple methodology that has been used. It will be based on Agile approach which we followed and details of it can be found in Appendix G. The diagram is a simple illustration of how the development cycle was managed and it's simplified phases.

[36] "GeoJSON - Wikipedia, the free encyclopedia." 2008. 1 Dec. 2014 <http://en.wikipedia.org/wiki/GeoJSON>
[37] "GeoJSON - Wikipedia, the free encyclopedia." 2008. 1 Dec. 2014 <http://en.wikipedia.org/wiki/GeoJSON>



The cycles above are very well presenting the actual working cycle we had in a team. This is clearly an Agile approach in which we could find clear instructions of what to follow. We didn't verify the exact number of cycles but the diagram illustrates the idea. We can surely say that the Agile approach worked very well, however we had to consider both advantages and disadvantages of it. The freedom that Agile gave us was the adaptation to changes. New changes could have been implemented at very little cost because of the frequency of new increments that were produced. To implement a new feature or function we needed to lose only few hours of work to roll back and implement it correctly. Comparing the development model we used to the Waterfall approach [[38]], very limited amount of planning was required to get started with the project. The assumption that the end users' needs are ever changing in a dynamic fashion has greatly helped us. Changes and new features could have been introduced in a short timeframe or removed based on feedback. This effectively helped us to deliver a finished system that our clients needed. In our model systems, developers (Team 5) and stakeholders alike, find they also get more freedom of time and options than if the software was developed in a different sequential way. By having different options it gave us the ability to leave important decisions until more or better data or even better software became available; meaning the project could continue to move forward and be developed without the fear of reaching a sudden standstill.

Further evaluating this model, we to consider the downsides of it and make sure we manage them appropriately so that they won't work against us. The very important one is the excess of the effort required at the beginning of the software development cycle. We had to select suitable software deliverables in order to satisfy the requirements. The clear disadvantage is the lack of emphasis on necessary designing and documentation, but this issue has been elaborated in Appendix G and in fact it worked in our advantage. With the Agile approach we followed it could have gone easily off the track if the client was not clear what the final outcome is. However, this problem has been completely minimised by the fact that we could come up with our own idea of how to improve TfL's network and therefore we had full control of the system we were developing. The last problem we had to face was the selection of engineering leader who can act as a senior programmer who is capable of taking the kind of decisions required during the development process. Hence this was no place for newbie programmers, unless

[38] "Waterfall model - Wikipedia, the free encyclopedia." 2004. 1 Dec. 2014 <http://en.wikipedia.org/wiki/Waterfall_model>

combined with experienced resources which we had considering large group with members who have experience in variety areas.

B.11.2 Configuration Management

This project had a combination of local and server side development. Almost all of the development on visualisation happened locally and involved small subsets of data. It was initially agreed that the visualisation team would work on a framework that would process and display data, and efforts were made to completely abstract the team from needing actual data. However for the purposes of validating their work, they needed a small subset of the data to check that the visualisations yielded valid results.

The bulk of the data is held server side and accessed through an API for the visualisation team. The server is a VPS hosted by DigitalOcean. Using a VPS allows us to not think about the problems of building servers and making them achieve high availability. Ubuntu 14.04 was the chosen operating system. Its package management system was simple and versatile enough to install all the software that we required, the most important pieces of software being nodejs and Postgres that Ubuntu made simple to install.

The server hardware we chose was basic. It has 4GB of storage space, and 512MB or RAM. RAM is the key limitation in this system, given that the API may return large results that are buffered in RAM. This means that some of the larger queries cannot be run on the server (e.g. a query asking to return all data stored by the system), although even with the existing hardware it suits our average use cases.

The size of the raw data set the server is providing is around 1GB, and slightly reduces in size when parsed into the database. This corresponds to approximately 2.3 million rows, with four queries run against the tables that comprise the API. The visualisation team then do further processing based on the results of these queries, and in future iterations of the project this could be done server-side.

B.11.3 Communication Management

In order to facilitate communication effectively across teams members, both the ones located in London as well as the one in Japan, we have used a variety of methods. First, we have set up an email thread and we have also created a Facebook group which were both used to arrange meetings. Then, we have created a Github Issue Management page [[39]] that was used to create an environment of collaboration between team members, regarding the technical aspects of the project. Usually, the team would decide on what the milestone was for the week during the weekly meeting on Monday and then we would advance into breaking the milestone down into small manageable tasks. After that, the engineering lead would post them on the Github Issues Management page and everyone would volunteer to complete a task. If someone needed help, he could attach a "Help Wanted" tag and thus request help from other team members. Moreover, it was signified when a task was completed and there was also a comments section to provide feedback on each other's work.

It is also important to note that we have taken into account the time difference between the team members located in Japan and the rest of the team and have thus chosen communication tools that would allow us to work past this issue.

[39] "Issues 2.0: The Next Generation · GitHub." 2011. 1 Dec. 2014 <https://github.com/blog/831-issues-2-0-the-next-generation>

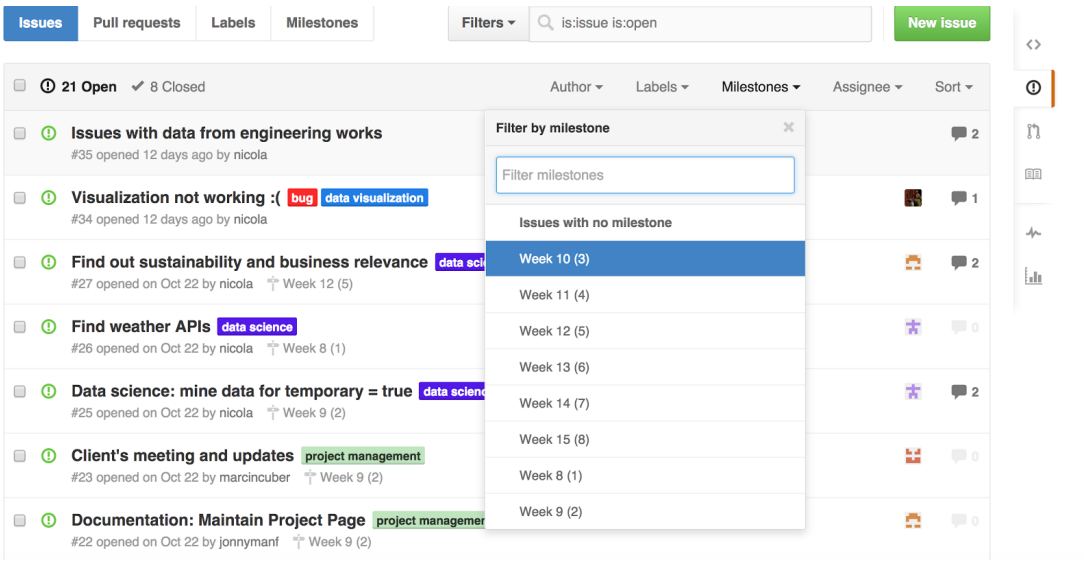


Fig. 4.1: The Github Issue Management Page

Wherever possible, we have opted to use collaborative communication tools and working environments to create an environment where each team member can instantly check and update information.

B.11.3.1 Physical Meetings

Throughout the project, physical meetings have been an essential tool for communication management. During each meeting, there was an opportunity for each team member to discuss their progress, issues and ask direct questions to any of the other members of the team. Also we had a dedicated person who had meetings with the client and based on his feedback we could modify our work and make it better.

These physical meetings were also a very useful way to manage group decisions; each team member could cast votes in person and express opinions in an easier way than online.

Weeks 1-4: 1 hourly meeting per week (Monday – 13:00)

Weeks 5-10: 2 hourly meetings per week (Monday – 13:00, Wednesday – 13:00)

B.11.3.2 General Communication and Messaging

Early in the project, when initial communications and introductions were taking place, we made use of group emails and used this channel to circulate information about the online communication tools we were using. It was through this group email, we opted to use the following messaging tools:

- Group email thread for important exchange of information
- Facebook Group and instant messaging channel to post updates and discuss quick queries

B.11.3.3 GitHub Issues

As discussed earlier, we are using GitHub's services to host our code online and for version control purposes, however GitHub also became an important communication tool.

Following our pattern of weekly meetings, we made use of the milestone functionality offered within GitHub which allowed us to have weekly milestones and assign tasks to each week. This was useful to help us make sure we were meeting weekly goals. In situations where goals were not met, issues can be switched to have a different milestone. Furthermore, each issue can have tags added to it as such:

- Data visualization

- Data science
- Project management
- Bug
- Blocking – Blocking progress of other issues

B.12 Feedback and Iterative Design Process

B.12.1 Feedback and Iterative Design Process

Our project underwent many changes throughout the development process, largely driven by physical meetings with our external client. We sought out an external client in order to gain advice and an external view for our project. He provided us with valuable information about the workings of TfL as well as encouraging us to consider stakeholders in our project, so that we could build something that would be of direct use to our stakeholders. We identified the primary stakeholder as TfL themselves (given their financial losses, they would massively benefit from anything that could make them, or save them, a significant amount of money). A secondary stakeholder we identified would be the customers of the bike hire scheme themselves. A more efficient system would benefit the customers as well as TfL, and so obviously customers would have an interest in the outcome of this project too. All of this was born out of our meetings with our external client, and so he was an extremely valuable resource to us.

Our project changed as a result of these meetings in three sections which are our hypothesis, the visualisations that we produced, and how we evaluated the efficiency of the TfL network (in other words, the formula that we calculated to provide a rough estimate of the flow of the network). Detailed below are the iterations undergone by our project:

B.12.2 Changes to our Hypothesis/Goals

Goal Iteration 1:

Our initial hypothesis, or goal, was that we would optimise the TfL bike hire network in relation to a particular dataset. What this meant was that we would identify a factor that correlated well with the bike hire data and analyse datasets relevant to this factor in an effort to reduce impact to the bike hire network, or even to optimise the network accounting for this external factor.

Goal Iteration 2:

After disagreements over which factor we would choose, we took our options to our external client. We considered many factors, such as engineering works, weather considerations, the effect of the rise or fall in tourism levels on the bikes and the effect of special events (football matches at Wembley stadium, NFL matches etc.). Our external client had a look over our multiple proposals and advised that if we wished to go down this route, our best bet would be the engineering works. He explained that the communication between the different departments within TfL was nearly non-existent, and that the teams running the bike hire scheme would not be talking to the teams in charge of the tube system, and that a tool that would collate this information and combine it would be very useful.

Goal Iteration 3:

Following this we carried out analytics on the data we gathered, we came along to a bump in the road. While there was a correlation detailing an impact on the bike hires numbers by the engineering works, this correlation was not as strong as we have hoped. The decision was now whether we push on in hopes of further analytics providing a stronger correlation, or rethinking our project goal. In our second meeting with our external client, he advised that whereas there could or could not be a strong correlation between the bike hire data and the engineering works, it might be worthwhile to consider other datasets as

well, since we had an analytics framework in place. Following further discussion we decided that it would be possible to not only consider other datasets, but build a more general platform to allow us to find correlations between the TfL bikes hire scheme and *any* dataset. We felt that this would take our project in a strong direction. In addition, we decided to create an API to allow us to incorporate multiple datasets. The API gathers information and converts it into standard data as well as allowing for more sophisticated communication with the database. This in tandem means that we can generalise the datasets we are trying to find a correlations with.

Goal Iteration 4:

For the fourth iteration, ambition took hold. We reasoned that since we were able to create an analytics framework that could find correlations between the TfL bike data and specific datasets, why not further generalise the framework, allowing for analysis of any bike data with these datasets? The idea was to be able to incorporate data from Parisian and Canadian bike hire systems, and in fact any other bike hire system, and allow it to find key correlations (since these correlations could vary from city to city). This is our vision for this project and while this is not yet implemented, we hope that in the future we will be able to implement this and provide a genuinely useful tool for public transport networks all over the world.

B.12.3 Changes to visualisations

Visualisations Iteration 1:

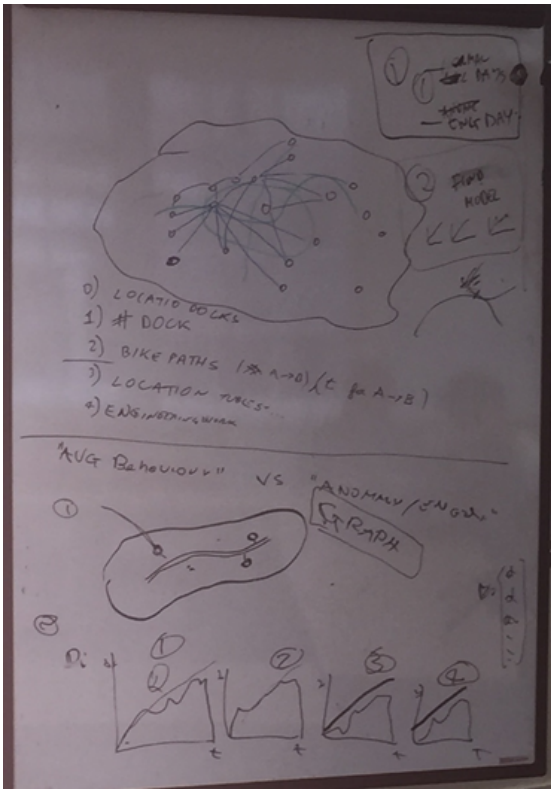
For goal iterations one and two, we were using Google Maps to power our visualisations. There was no coordinate system so we hard-coded absolute positions of tube and bike stations to display where they were in relation to each other on the map, and showing various desirable pieces of information.

Visualisations Iteration 2:

When we moved to goal iteration 3, we realised Google Maps was nowhere near powerful enough for what we wanted to do. Firstly, the fact that we had to hardcode in data that we wanted meant that incorporating more datasets could not be automated and would be incredibly time consuming. It was also saturated with a lot of unnecessary information which would only disrupt those people looking at the visualisations and attempting to make decisions based off of them.

We decided to move to D3.js to fix these issues. First of all, it gave us a custom scale, by providing the minimum and maximum longitude and latitude values for the area we were looking at and it lets us have relative latitude and longitude values which would be placed on the map (and so each object on the map had its own coordinates). We could also create custom vectors and segment lines on the map, with each object having the ability to be individually edited. This meant that we could display specific visual properties on the object, such as an area of impact, or the number of available bikes etc. This means we can automatically generalise data in a much more powerful way.

All the above helped in choosing d3.js in conjunction with the advantages that were stated in section B.9.1.2.



[Considering changes to the visualisation between Visualisation Iteration 1 and Visualisation Iteration 2]

B.12.4 Changes to formula

The original iteration of our efficiency formula was of a similar form to our final version (or second iteration). In the main iteration the formula would produce a score between 0 and 1 for the network flow efficiency in a given time period, however, after some meetings, our external client notified us of some shortcomings within the formula. Firstly, the original one simply defined "inefficient time" as when a bike station has either less than 1/3 of bikes available or less than 1/3 of docking points available. This was not sufficient as we realised that some stations are busier than others, and some have certain expected loads on each station making this metric insufficient. In addition, the final number that would represent the network would not represent the state of the network, i.e. we would not be able to tell whether there was, on a whole, more bikes than necessary available, or less bikes than needed available. With this feedback, a usage type coefficient and a popularity metric were introduced into the equation, resulting in a number from -1 to 1 for any day, with 0 being optimum network flow, 1 representing more bikes than necessary available at maximum inefficiency and -1 representing not enough bikes at maximum inefficiency. The details for the formula can be found in section D.

B.13 - Evaluation about the re-development

B.13.1 How We Would Do It If We Were To Do It Again

When we serialized the data for tube delays, part of the information was in natural language form. We immediately attempted to convert the natural-language into something else more basic. We believe now that in the first stage, the natural language data should be serialized as-is without any processing. This would be a simple conversion from HTML to SQL which represents a form of abstraction. The

information about which stations are affected is in the natural-language field. So in the next stage, the rows should be classified. Those rows which can be interpreted, in the sense that we can totally interpret the natural language and know which interval of stations was affected by a delay, or which can be interpreted because they say 'good service', are put into one class, and the rest put into a class saying 'Not understood'. Then, we would produce a new table associating lines and times to service status (including ''Severe Delay', 'Minor Delay') and interval affected if applicable with a possible value of 'Not understood'. This is a more principled approach than the one we took, in which we threw away the 'good service' rows immediately because we didn't have enough storage on the machines we were working on (a solved problem now because we know how to use the cloud), and we had a bit of bad data (imperfectly processed from the natural language) along with the good data. We took the approach we did because we didn't have a lot of time and a lot of storage. The next stage would be to produce a table associating stations, lines and times to statuses.

From this stage, we would try to find some way of relating this data to the data about demand in bike docking stations. The data we got for the bikes only goes back three weeks and is therefore an insufficient sample.

At this stage, the tables would all be publishable. We would include a document saying what the data says. This can be picked up by other people who can complete the natural language processing.

Further processing of the tube station status data is needed. It's scientifically rigorous to minimize the number of variables which can affect a measured outcome. It might therefore seem a bit funny that we would try to do the following: given a tube station and time, give a status that is somehow a summary over all lines going into the station. How to do the summary is a major unknown. One approach is to simply say whether or not a delay happened over *some* line that prevented trains coming in and out of the station. We already have evidence that this has an effect. We would keep the results from the previous stage along with the current one.

Finally, we would estimate a physical distance threshold. For each bike docking station, we would produce two histograms. One is for when there are delays on stations within the distance threshold to the docking station, and one for when everything is OK. The y-axis would be frequency and the x-axis would be the number of bikes taken out over the previous 15 minutes. It should be said that what goes on the x-axis depends on what data is available and what processing can be done on that data. This needs a lot of thinking. On each time period in which there is a delay, we would add the information about bike demand to the histogram for bike delays. Whenever there aren't delays, we would likewise introduce the data into the appropriate histogram.

At this point you can produce a map. If you mouse over a bike docking station, you can see the two histograms. This might help in coming up with new ideas.

The visualisations we've done are good and prove that there is a trend.

You can then produce, for every histogram, statistics to summarize the data. Good statistics might be median and median deviation from the median. The median is good because it's a stable statistic, which means it's not, affected by outliers, whom we imagine should be present in the data.

After that, you can produce a new map. The map has circles in it. The circles are over bike docking stations. Each circle has a colour. The colour says whether or not there's a big difference (above a certain threshold) between the median values for demand when there are delays and the median demand for when there aren't.

It would be a good idea to discuss this with statisticians, data scientists and people who take out bikes.

In summary, we could've been more rigorous, and gathered more data. We could then use that to compute aggregate statistics over time like histograms and medians.

We could not do the above because we didn't have enough time to implement it and the samples we collected were not large enough. Also, since it's so complicated it needs to be discussed with experts and actual users of the Barclays service.

Appendix C
Architectural Diagrams and Data Sets

Appendix C - Architectural Diagrams and Data Sets

The following architectural diagrams were designed in order to give the development teams a more clear view on what needed to be developed. In essence, it was a bridge between the project management team and the development team so as to translate the knowledge that the management team gathered, using various ways such as interviews. So, in the following section we will present some architectural, UML and sequence diagrams.

In addition to that, we will also present that datasets that we have used and provide an overview of them.

C.1 Data Sets

For the visualisation we used three data sets, they are:

- TfL bikes open data[40]
- TfL tube data- engineering works
- Maps data

C.1.1 TfL Bikes Open Data

This set that we have used contained information about 6,000 bikes available from over 400 docking stations across central London. At any point, we could verify whether docking stations are full or empty, working or not working, and the number of bikes that are available to rent. The feeds we could collect contain date and time stamp which should be used to check that the information is up-to-date and be displayed when publishing the information or in our case when data is visualised against the time.
The information we had is summarised below:

- Each docking station contains location details (longitude and latitude) and so visualisation produced by us can display the data and can be filtered for geographic area of interest, on a map
- The information has been used to provide a complete picture of available docking points or bikes along a route or in an area
- The data has been fed into an online system which we used

An initial issue was the form of the legacy data. This only contained information about journeys undertaken with the bikes rather than the status of docking stations. Given this setback we began polling data from the live TfL bike station feed and resulted with 2.2 million rows of records which we could use for analysis.

C.1.2 TfL tube data

The TfL tube dataset that we had available to us was merely the data on journeys undertaken via the underground tube system. This wasn't very helpful for our analysis since the information we needed was just the information about which engineering works had been undertaken on which days over the previous

[40] "Our feeds - Transport for London." 2014. 4 Dec. 2014 <https://www.tfl.gov.uk/info-for/open-data-users/our-feeds>

few years. TfL provided data for engineering works but there was no legacy data available, and so we had to search for legacy engineering works data. This was found in the form of a website [41]. This website had polled data for the last several years on engineering works that had been taking place, detailing between which stations and the exact times as well. This data was then scraped and was made ready to use.

C.1.3 Maps Data

The map data that we used was from OpenStreetMap [42]. This was chosen because having decided that we would use D3.js, we found that OpenStreetMap worked best with D3.js, as well as GeoJSON[43] (another technology that we thought would best fit our project). OpenStreetMap has very similar capabilities to Google Maps but as I mentioned provides better integration with D3 and GeoJSON. It is also an open data set allowing for many interesting tools that have been built off of it.

C.2 Extracted data for future use

The backend was built in such a way that adding new functionality based on our existing data set has been simplified. Access to the data is controlled by SQL, and each API call maps to a certain SQL statement. Adding these mappings to the server side application is a simple process, and thus extending the API to meet our future needs has also been simplified.

C.2.1 API Documentation

The following is our API as it stands (accessible at **http://178.62.32.221:5000/**):

/station	Static information for all stations
/station/<id>	Static information for supplied bike station ID
/station/<id>//data	All available data for station
/station/<id>/data/<start>/<end>	Available data for supplied bike station ID during time period denoted by start and end. Times are encoded as UNIX epoch time in milliseconds
/station/data/<start>/<end>	All available data across all stations for the specified time period (time encoded as above)

The API was designed with a RESTful [44] approach. The idea is that the resources provided by the API can be thought of as a file and folder like structure. Each resource that is represented as a file has a file path, and in the API this is represented by the URL. Going deeper into the file structure will yield more detailed and specific results. Having this structure means that adding extensions to the API would not change the way existing resources are accessed through the API. This is due to any extension either providing more detailed data in which case its 'filepath' is based on an existing resource, or it would

[41] "tubestatus.net - live London Underground (TfL) tube status ..." 2012. 2 Dec. 2014 <http://tubestatus.net/>
[42] "OpenStreetMap." 2004. 2 Dec. 2014 <http://www.openstreetmap.org/>
[43] "GeoJSON - Wikipedia, the free encyclopedia." 2008. 2 Dec. 2014 <http://en.wikipedia.org/wiki/GeoJSON>
[44] "RESTful - Wikipedia." 2006. 4 Dec. 2014 <http://en.wikipedia.org/wiki/Representational_state_transfer>

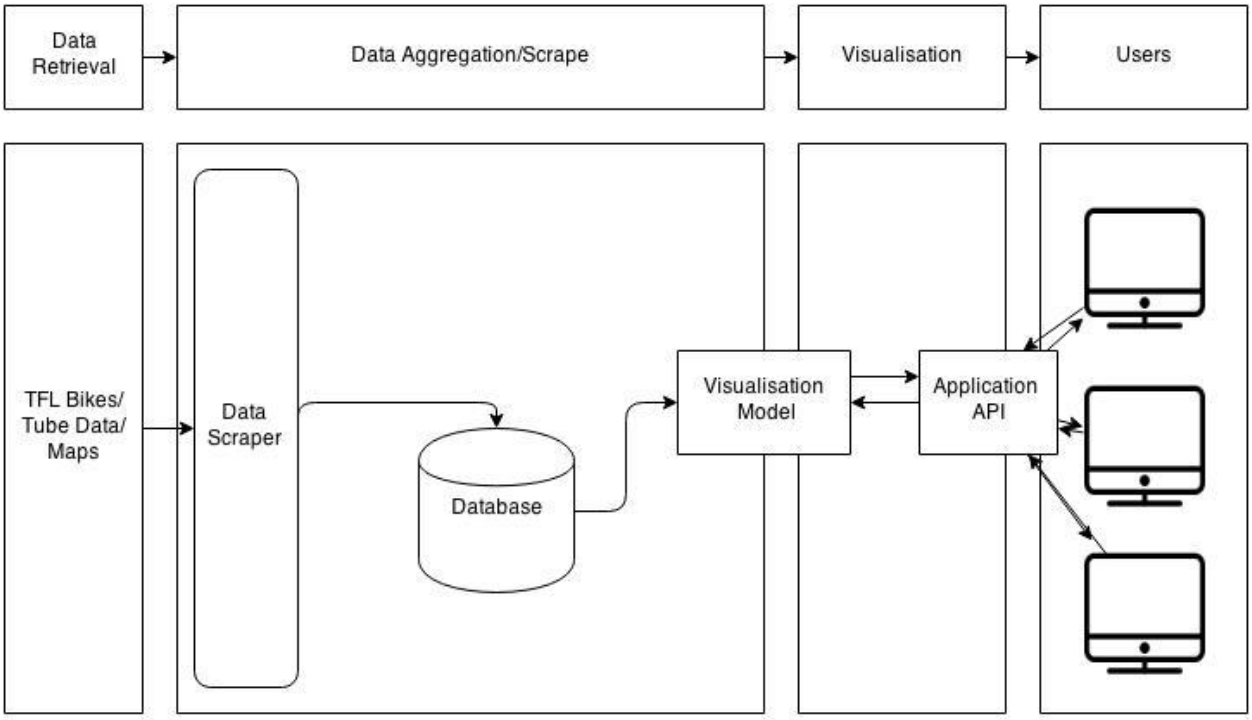
provide completely fresh data and its 'filepath' would be different at the root, thus not affecting any other resource paths.

Consider the following example of how the API can be extended. Weather data can be added at the root of the API, returning weather data based on varying levels of granularity on /weather (e.g. summary for day, hourly forecast). Extensions could be made to /station to then involve this weather data and return how the weather correlates with a particular bike stop.

This provides a more convenient method of accessing the data compared to TfL's API, both in reducing a developer's cognitive load and for efficiency. For future iterations of this system, more intensive tasks could be handled by the server and served to the visualisation team. Exposing the visualisation team to just the API allows for the backend team to develop more sophisticated hardware architectures to handle more complex queries without disruptions to front-end services.

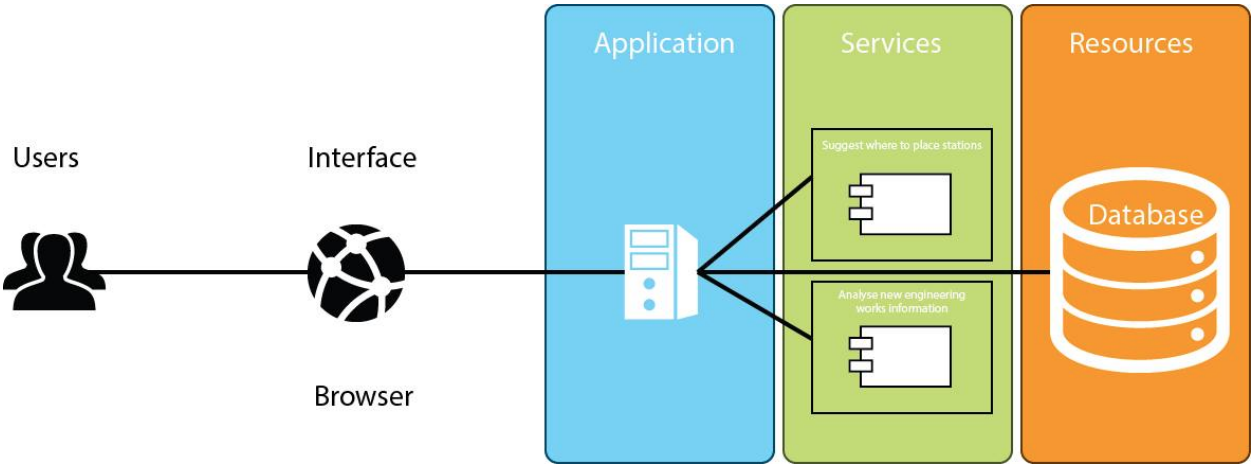
C3 Architectural Diagrams

C.3.1 Application Overview Diagram



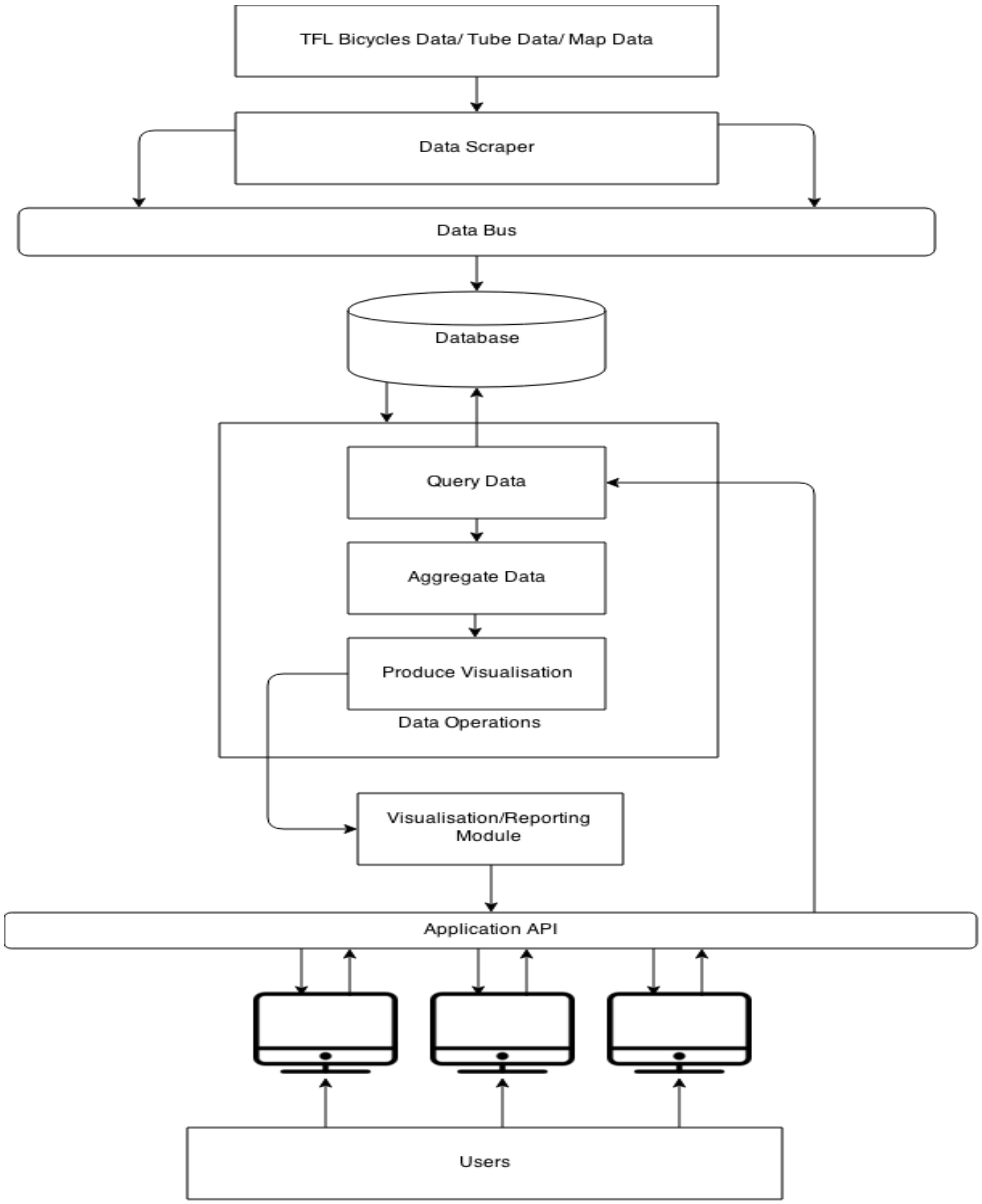
The diagram above shows an overview of the application. We retrieve our data from the sources that TfL provides and we aggregate them. Then we scrape them in order to identify useful data from the other data that hold no value for our system. Then, we store them in a database. After that, the users can interact with our application API on the web and produce visualisation using the visualisation model that will be returned back to the user once the computation is done.

C.3.2 Network Architecture Diagram



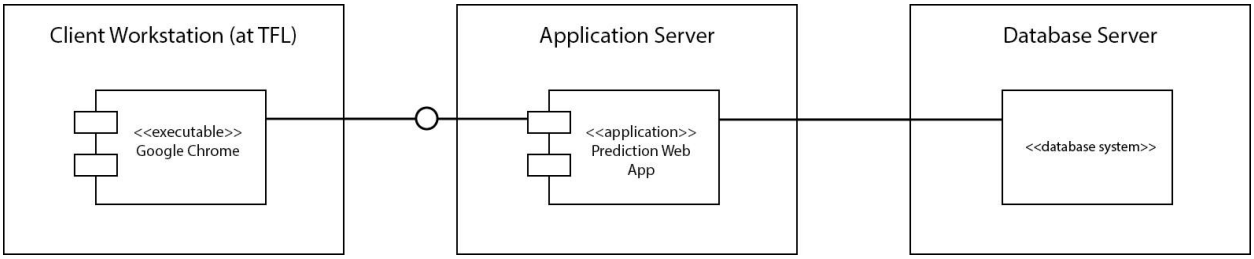
The above diagram illustrates the network architecture that we have opted to follow. To connect to the analytics system, the user would connect to the application via their browser (which acts as an interface). The application provides two services, the first being a suggestion on how to optimise the flow of the network, with the second service being the analysis of new engineering works information. This diagram was created for the second iteration of our goal, and has since been appended to reflect the new services that should be provided. These services should include the ability to request analysis on new data sets, analysis of existing datasets, an output of current and previous efficiency ratings, as well as future predicted efficiency ratings and the ability to select which bike system to optimise for (a service that will be included in the 4th goal iteration onwards). To provide these services the interface will communicate with the database server to retrieve information via queries.

C.3.3 Detailed Application Diagram



The diagram above provides a more detailed aspect on the implementation of the application that was presented in section C.3.1. The diagram illustrates that we begin with the data sources and we scrape the relevant data from these datasets. From here the data is put onto a data bus before it can be parsed into the database in a standardised format. From here, we have multiple links between the data operations we perform and the database. In one instance of a data operation, we would query the relevant data from the database, and then aggregate this data together, entirely depending on what kind of data operation we are producing (this would determine the type of visualisation we produce and the specific pieces of data we need from the database). A visualisation would then be produced from the aggregated data and this would be passed to the visualisation/reporting module. This is finally fed into the application API and is made available to view by the users requesting the visualisation.

C.4 UML Diagram



The above diagram illustrates the relation between the servers in our system, and the entities that exist within them. The client workstation would simply contain the executable program to interface with the internet (of course it contains other entities but these are not relevant to the system). This communicates to the application server which contains the prediction web application, and this application communicates to the database system which exists on the database server.

C.5 Sequence Diagram

This is the sequence diagram which illustrates the sequence of data requests that will pass through our system. The prediction service interface passes along an engineering works information input (information about upcoming engineering works, however after our third goal iteration this is now just any general relevant information) to the prediction engine. This engine then queries relevant data from the database which will return the relevant data to the engine. The engine then makes its predictions and produces the visualisations which are passed to the interface for viewing by the user.

Appendix D
Earned Value and Costs

Appendix D - Earned Value and Costs

D.1 Value of Product

Our product will be of value to our potential users in many different ways. First, it will be of financial value of course. The system is designed to optimise the Barclays Cycle Hire network, and inherent to this is the idea of reducing costs for TfL. By predicting where bikes will need to be moved during engineering works as well as in general, TfL can reduce money wasted in moving bikes to places where they will not be used. If they are taken to the optimum place, not only do you make more money from someone using the bike because it is available (where previously it would not be available) but you also save money from the fuel for the distribution van as well as saving time. Perhaps fewer employees that are involved with the redistribution will be needed as well. Another value for our product is quite possibly an environmental impact. By making vans have to deliver fewer bikes, you save fuel and are therefore greener, a great way to gain popularity with the public. In addition, if it became the everyday norm that a bike is available a lot more often than not, people will hopefully begin using the bikes more (due to the convenience) which increases money intake by TfL as well as reducing their economic impact. This is just one example of the optimisation that can be done with the data provided by our product, and it will be possible for many more to be done.

D.2 Earned Value

In terms of the earned value of our system we must consider the stage the product is at. Our product can analyse the TfL network as a whole and has the capability to detect anomalies in usage patterns, particularly within the TfL BCH. After this it will attempt to identify causes for these anomalies by aggregating numerous data sets. While not producing perfect results it has produced very promising results and is on track to achieve its full value, both in terms of the financial and environmental aspect. Therefore the use of budget and time that has already been put into this product is justified and there is no need to have spent any more at this stage.

D.3 Ease of Use

The product was developed with ease of use in mind, taking a clean approach to its design. The visualisations have been designed to be easy to read and understand, and the tool gives a value for the performance of the network and will compare it to the normal usage. The output is a simple number which tells the user how the network is faring in comparison to the networks average performance. By providing a number to compare to another number, the user can see very easily whether the network is performing optimally or far from optimal. See footnotes [45] and [39], for charts relating to the cycle hire scheme usage and financial impact.

[45] "BCH Transparency to end of Sept 2014." 2014. 2 Dec. 2014 <https://www.TfL.gov.uk/cdn/static/cms/documents/bch-transparency-end-september-2014.pdf>
[39] "Boris Bike Usage 2012 vs 2013" 2014. 2 Dec. 2014

D.4 Development Costs

The development cost of this product lies solely in two places, the cost of buying and hosting a server and the cost of the team members developing the product. Of course given that this is a University module the salary of team members is a moot point. However, for sake of completeness, the cost will be calculated. Our team consists of 12 people, with an average software developer salary being £30,000, which breaks down into **£17.98** per man hour (the average worker in the UK worked 1669 hours in 2013). As a group we undertook 2 group meetings a week, for 11 weeks, resulting in 22 man hours. There was then the development work itself, which would average at another 36 man hours per week, resulting in 396 more man hours over the project course. This totals to **418 man hours**, resulting in a cost of **£7515.64** for the work performed by the team.

D.5 Commercial Viability Estimate

The commercial viability then works out as follows. The cost of developing this product lays at **£7515.64** currently. However, this product does need some further work, as outlined here:

1. Extensibility of API 50 Man Hours
2. Upgrade Analytics and Visualisation Engine- 50 Man Hours
3. Documentation 10 Man Hours
4. Load Testing 05 Man Hours

This totals to another further 115 man hours, bringing the development cost up to **£9583.34**. We will also need to carry out user testing, which totals another 25 hours. People who come in for the user testing will have to be compensated. So, the average rate of compensation is £15 an hour, resulting in a further cost of **£375**.

Finally, will be the cost of running Hadoop on a server. In order to run Hadoop on an Amazon web server (given 16 CPUs, 60 GB of memory and 11 TB of server storage, all of which should be more than enough) would cost **£3.80 an hour**.

So, the final cost of this product is an upfront cost of **£9583.34** and a further cost of **£3.80** an hour (which is equivalent to £91.20 of running costs a day).

The scheme supposedly pulls in £12 million pounds per year, and costs nearly £39 million a year. Obviously the TfL BCH is running at a great loss at the moment, which is only slightly offset by their sponsorship by Barclays. Over a year, our product would cost £33,228 to run. This is a drop in the ocean given how much TfL are losing through the scheme, and quite simply they can't afford not to consider our product.

Of course, the implementation of our analytics application is the first step for TfL to reduce the costs of running the BCH year on year.

Given the extensibility of our software, they can first tackle disruptions to the network during engineering works and then easily use the same analytical framework to assess how they can improve the re-distribution in the network during adverse weather conditions.

D.6 Return On investment (ROI)

<http://road.cc/sites/default/files/imagecache/galleria_900_nocrop/images/News/Boris%20bike%20hires.png>

The bottom line savings were calculated from our efficiency formula, which shall be detailed here.

First of all, we considered a way to gauge the popularity of a station. This was calculated as follows:

$$P_i = \left(\frac{B_{Outgoing}}{B_{Incoming}}\right)^z$$

Pi represents the popularity of station i.
B outgoing is the numbers of bike leaving a station, B incoming is the converse. Z is the usage type coefficient, and is 1 if a bias towards more bikes exist, -1 if a bias towards empty space exists.

The popularity is the flow from a station, calculated via a ratio of bikes adjusted to always end up as a fraction. A value close to 1 shows a high flow/popularity of a station.

$$F_i = |P_i| \,.\, D_i$$

Fi is the calculation for how many free bikes/slots a station i should have. Pi comes from above and Di is the total number of slots available at station i.

This allows us to calculate the number of available bikes or spaces we would want to have at a stop given its popularity/flow. This allows us to define an inefficient state as the time for which there are less than Fi bikes available or less than Fi spaces available.

$$E = \frac{\sum_{i=1}^{S} T_i P_i}{12S}$$

Here, S stands for the total number of bike stations, and E calculates an unadjusted efficiency value for the bike network. Ti is defined as the time a bike spends in an inefficient state, with the 12 coming from the most active number of hours of use for a Barclays bike.

The Pi is present to allow for a weighting of the most popular stations. Inefficient time has a much larger effect when it occurs at a very popular station and so this needs to be considered in the efficiency rating.

Finally, we have the completed equation:

$$(-1)^Y E$$

Y is defined as 0, if on average there are not enough spaces for demand within the system, and 1 if there are not enough bikes for the demand. The final part negates the value of efficiency of the network if there aren't enough spaces and makes it positive if there are too many bikes. The outcome is that a value close to 0 represents a very efficient network, while a 1 shows that while inefficient, there is a bias towards too many bikes being present without enough spaces. A value close to -1 shows an inefficiency with a bias to not enough bikes being available.

After applying our efficiency formula to the available data, we created a value for the average efficiency of the network, with the value being 0.472. Given our recommended changes, we found that this efficiency value would change to 0.424. What this represents is that the flow of bikes will be

significantly altered, increasing by a factor of 0.472/0.424. The TfL scheme currently earns **£32,876.71** a day (calculated from the TfL website). A better flow will produce a larger amount of earnings with more people being able to use the bikes, and so we calculated the new daily earnings to stand at *(0.472/0.424)x32876.71=***£36,598.60.** This means that with our predictions they can start earning **£3721 a day** more than previously, meaning they would get back the cost of implementing the system within 8.9 days, and within the first year they would make **£1,324,937** more than normal.

Appendix E
Conduct Policy

Appendix E - Conduct Policy

E.1 Ethical and Privacy Policy Considerations

An ethical code of conduct and attitude towards data as well as considerations of data privacy are more important in big data projects than perhaps any other type of computer science project. Given the large quantities of data that we will have analysed it is of paramount importance to have clear guidelines so that we act within the rights of an individual concerning their data and do not break any privacy laws in the course of our project or data gathering. Given the nature of our project, working to optimise the cycle hire network, our only ethical concerns would be how we handled data privacy, since the outcome of this project would have no impact on other ethical issues.

Within the issue of privacy, we can look at 3 distinct problems, or types of privacy. These are freedom from intrusion, control of personal information, and freedom from surveillance, as outlined by Gary Pollice [[46]]. Freedom from intrusion is essentially the right to be left alone, unless explicitly stated (for example you may choose to allow a company to contact you in regards to some deal you may be interested in). This we can ignore since we are using publicly available data sets and the collection of this data has not intruded on the lives of any user.

The control of personal information is a powerful issue. In the modern day where much of our lives are now on the internet and publicly accessible, it is very difficult to control what information about you exists out there on the internet. This comes into play when we consider what could be done with the information that we have. An unscrupulous individual may decide to attempt to access the data we have gathered and use it to impinge on the third privacy issue, that is, freedom from surveillance. Data of any kind is very powerful and can be used for criminal activity, so the data we have collected must be protected from those who would wish to misuse it. Data encryption is one possible manner in which public data can be protected.

However, considering the data we are using is all publicly available, it perhaps makes little sense to protect the data from the TfL streams or Google Maps, however, if we simply encrypt the data from our own analysis, we could reduce the resource use (as compared to encrypting everything) while protecting the vital data. Another idea is data masking. This is the practice of removing uniquely identifying factors from data. This could involve applying unique numbers to specific pieces of data and removing descriptive details from each data entry. The result of this would be that we would know what the numbers meant, but an outside user will not. Apart from this standard data protection procedures will be taken, with firewalls protecting our system from outside attacks.

The final privacy issue is the freedom from surveillance. Given that we have details of bike journeys undertaken for the last 4 years as well as tube line disruptions over this time period as well we could very possibly begin to try and identify a single user and begin tracking his or her movements over the last four years (if we notice that a bike is always taken at the same time from the same station, we could begin tracking the journeys and find out more about this person). This is a very sensitive issue and

[46] "Ethics and software development - IBM." 2006. 2 Dec. 2014 <http://www.ibm.com/developerworks/rational/library/may06/pollice/>

of course every Barclays Bike user has the right not to be tracked or identifiable. We must be careful to not try and estimate popular bike routes from common routes taken since an algorithm we write for this may begin to track very singular journeys to find what it perceives to be a popular route. We must be vigilant in how we analyse data in order not to unconsciously begin surveillance on a specific person.

To best act within the law and interests of the users of the TfL public transport network we will have to comply with the Data Protection Act 1998 [47], and all other governance legislation and the eight Data Protection principles (outlined within the DPA). In addition private data will not be disclosed to third parties without the explicit consent of the data providers and all who are concerned and privacy risks will be of high importance when considering future developments.

[47] "Data Protection Act 1998 - Legislation.gov.uk." 2010. 2 Dec. 2014 <http://www.legislation.gov.uk/ukpga/1998/29/contents>

Appendix F
Project Resources

Appendix F - Project Resources

In the following section we will provide a summary of the resources that we have used in order to build our application. To be more specific, we will list the resources that we have used not only for the technical aspects of the project but for identifying requirements and managing the team.

F.1 Primary Source

Internal Client - Graham Collins
External Client - Dr. Stephen Pryke

The internal client has provided us with general information about the project following with ideas on what should be included and how the problem could be tackled. We were able to verify with him all the ideas we had and how the system should be delivered. The help that was supplied from him gave us an advantage and enabled us to steer our work in the right direction.

The external client has also proven to be a valuable resource to our project. We conducted an interview with him to gather information (see appendix A). Based on the information that he provided we structured the requirements. Additionally, we were able to verify whether our project's hypothesis is a good start point and whether the final product would be of use to TfL. Dr. Stephen Pryke has also provided direct contacts to TfL whom we contacted in order to get more data.

We have looked at information on how the traffic is modelled, prices, future of portable bikes, modernisation, cycle routes but also safety plans. All the aforementioned information was easily accessible online and provided us with general knowledge on how the network is functioning from both TfL's and user's perspective. The following is an aggregated list outlining the resources that we have used:

- Traffic Modelling Guidelines [48]
- Boris Bike Price Hike [49]
- Future of Foldable E-Bikes or Electrical Modular Buses [50]
- Modernisation of TfL Networks [51]
- Bikes Super Highway [52]
- Killed Cyclists and Dangerous Crossings [53]

---

[48] "Traffic Modelling Guidelines - Transport for London." 2012. 2 Dec. 2014 <http://www.TfL.gov.uk/assets/downloads/traffic-modelling-guidelines.pdf>
[49] "Has Boris Bike price hike put off London's cycle hire users ..." 2 Dec. 2014 <http://road.cc/content/news/117423-has-boris-bike-price-hike-put-london%E2%80%99s-cycle-hire-users>
[50] "TfL Bikeshare Network Concept with Foldable E ... - Tuvie." 2009. 2 Dec. 2014 <http://www.tuvie.com/TfL-bikeshare-network-concept-with-foldable-e-bike-and-electrical-modular-buses/>
[51] "Our plan for London's roads - Transport for London." 2 Dec. 2014 <http://www.TfL.gov.uk/campaign/our-plan-for-londons-roads>
[52] "Boris Johnson announces Cycle Superhighway ... - Road.cc." 2013. 2 Dec. 2014 <http://road.cc/content/news/98465-boris-johnson-announces-cycle-superhighway-improvements-he-opens-new-section-cs2>

- Cycle Safety Plan [54]
- Cycle Security Plan [55]
- 25% of Bikes in London [56]
- Cycle Routes and Strategies [57]
- Progress and future challenges [58]
- London Mayor Unveils £913m Plan to Revitalise Travelling by Bicycle in London [59]

F.2 Secondary Sources

In order to build the application we have used a variety of tools. First, we have used Git [60] and Github [61] as version control [62] tools and to facilitate team collaboration.

In addition to that, we have used data from the TfL sources as well as data from Tubemap [63]. In order to scrape the data we have used Beautiful Soup [64]. After that, we have used Javascript and more specifically the D3.JS [65] framework for the visualisations.

Concerning the architecture diagrams, we have used an online tool called draw.io [66].

---

53 "Cyclists in the City: Top 10 dangerous junctions for cycling ..." 2011. 2 Dec. 2014 <http://cyclelondoncity.blogspot.com/2011/10/top-10-killer-junctions-for-cycling-in.html>

54 "London's draft Cycle Safety Plan and Design Guide: pretty ..." 2014. 2 Dec. 2014 <http://www.ctc.org.uk/blog/roger-geffen/londons-draft-cycle-safety-plan-design-guide-really-pretty-good-uk-standards>

55 "Cycle Security Plan - Transport for London." 2014. 2 Dec. 2014 <http://www.TfL.gov.uk/cdn/static/cms/documents/cycle-security-plan.pdf>

56 "Cyclists make up a quarter of London vehicles, says TfL ..." 2013. 2 Dec. 2014 <http://www.theguardian.com/environment/bike-blog/2013/jun/25/cyclists-quarter-london-vehicles>

57 "TfL Consultation on Cycle Routes and Strategy | FitzWest." 2014. 2 Dec. 2014 <http://fitzwest.org/wordpress/have-your-say/consultations-and-external-links/transport-for-london-consultation-on-cycle-routes-and-strategy/>

58 "london's transport: progrEss And fuTurE ... - Siemens." 2013. 2 Dec. 2014 <http://www.siemens.co.uk/pool/news_press/news_archive/2013/world-class-TfL-report.pdf>

59 "Boris unveils billion pound plan to civilise London | Bicycle ..." 2013. 2 Dec. 2014 <http://www.bikebiz.com/news/read/capital-s-cycling-czar-unveils-ambitious-plan-to-get-london-cycling/014484>

60 "Git." 2008. 2 Dec. 2014 <http://git-scm.com/>

61 "GitHub · Build software better, together." 2008. 2 Dec. 2014 <https://github.com/>

62 "Git - About Version Control." 2014. 2 Dec. 2014 <http://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

63 "Tubemap.net: London Underground Map." 2012. 2 Dec. 2014 <http://www.tubemap.net/>

64 "Beautiful Soup: We called him Tortoise because he taught us." 2004. 2 Dec. 2014 <http://www.crummy.com/software/BeautifulSoup/>

65 "D3.js - Data-Driven Documents." 2010. 2 Dec. 2014 <http://d3js.org/>

66 "Draw.io." 2012. 2 Dec. 2014 <https://www.draw.io/>

Appendix G
Agile approach

Appendix G - Agile Approach

G.1 Overview

Our team has been using Agile software development principles [67] which is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. For this project we are using Agile methods to fully develop our prototype system. Agile was the best option and choice for our project because it promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach and encourages rapid and flexible response to change. Because of iterations throughout the development plan this has turned out to be the best approach because it allowed us to work incrementally and not waste a lot of effort and resources when an idea did not turn out to be as useful as it was initially thought to be.

G.1.1 System Over Documentation

Following the Agile approach we have fully covered aspects such as:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over "contract negotiation"
4. Responding to change over following a plan

The project has been very demanding and a lot of things needed to be processed and delivered at during the active development phase.

Number 1: The individuals and interactions; we were organised and motivated which is very important. However, many tasks needed more than that so we had to do some pair programming and pair testing which of course led to better execution of tasks. This was crucial to our project for many reasons, the major one being that it is a huge task to develop an accurate visualisation, which combines two or more data sets. Therefore pair programming/testing has played a very big part in our project and has also reduced the risk of making any errors, which could have been time consuming. It has been a successful decision to take this approach because it helped us be more accurate with our work and also saved us a lot of time during design, development and implantation stages.

Number 2: The working software has been a key part of our project, therefore we did not need to present any documentation to our client as long as the visualisation is done. What we mean by that is that the visualisation had to be working correctly with the result of seeing what is going on with TfL's bike network. We were producing weekly reports, which were sent to our client just to update him on our progress. More importantly though, the client wanted to see results and the system had to be completed first before anything. The visualisation has been presented to Dr. Graham Collins but also to our external client, Dr. Stephen Pryke, where in both cases we received positive feedback.

[67] "Agile software development - Wikipedia, the free encyclopedia." 2004. 2 Dec. 2014 <http://en.wikipedia.org/wiki/Agile_software_development>

Number 3: The customer collaboration is the stage which has been dedicated to requirements gathering. When we began our project we could not collect the full list of requirements because we had to conduct a lot of research to actually start anything. Therefore continuous client involvement was very important and ultimately led to successful decisions. We have been arranging meetings continuously to make sure we are doing the right thing and are on track with all the tasks. Our client has verified each piece of work, which has been accomplished with great communication with him. We had some trouble getting our final idea of what the actual optimisation is going to improve but our main section of the report explains well what we have accomplished. We worked closely to ensure that our solution could potentially have an impact on future expansion of the scheme.

Number 4: The responding to change stage is focused on quick responses to change and continuous development. We completely had to follow this stage to adjust to different problems we encountered. Many problems that have been encountered were identified while extracting data but we managed to solve them very quickly. Correct extraction of data in the format that was needed led to successful implementation of the visualisation.

G.1.2 Principles

The Agile Manifesto [[68]] is based on twelve principles:

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Working software is the principal measure of progress
5. Sustainable development, able to maintain a constant pace
6. Close, daily cooperation between business people and developers
7. Face-to-face conversation is the best form of communication (co-location)
8. Projects are built around motivated individuals, who should be trusted
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work done—is essential
11. Self-organising teams
12. Regular adaptation to changing circumstances

Our team followed all of the principles listed above and it followed that it was a perfect approach we could have selected for this type of the project. So, starting from the first one, we had 3 month to deliver a working system solution along with full documentation. We have definitely managed to do so and satisfy all the requirements. The rapid development and rapid delivery played a crucial part and so we had to make sure we did it well in every stage of the project.

Second point is all about changes in a short period of time. The team had done a great job in both documenting and implementing the system. We have been closely monitoring the progress and the way project is going. It is important to point out that testing has not been identified as one stage; we were testing the system throughout the entire phase of the project. In some cases when something went wrong or the functionality wasn't working as expected we got on with it straight away and solved the problem. We were aware from the beginning that this project could be expanded into any direction.

[68] "Manifesto for Agile Software Development." 2003. 2 Dec. 2014 <http://agilemanifesto.org/>

Our well-organised team has been working very well in terms of time allocation for each task. We kept developing our product at a constant pace.The well-managed team allowed us to achieve a great end-result and each of us has been working on a task. Keeping constant pace with the work allowed us to deliver the project on time, which includes both the visualisation and documentation. Frequent meetings gave us an advantage and updates on the progress were communicated instantly. Nevertheless, it was difficult to arrange face-to-face meetings all the time and therefore we also kept communicating via Facebook, email and Github. We were having two meetings a week in order to update each other and outline future work. All the meetings worked out very well in terms of organisation and communication.

The project was built around motivated individuals. At the beginning of the project we were slightly worried that we may not finish on time, however, careful analysis of the project led us to a complicated prototype of the system (visualisation). An advantage had been gained once we got to know each other and so we had a clear understanding between team members and also we could trust each other in terms of completing tasks.

G.2 Agile and Interactive Solution

G.2.1 Iterative Solution

Agile methods break tasks into small increments with minimal planning and do not directly involve long-term planning. This is something that we needed because everything had to be developed in a specified order and of course we could have on going documentation writing but this was not ideal and we decided to work on the visualisation first. We used the iterations, meaning short time frames that in our case lasted a week each. Each of the iterations involved a cross-functional team working in all functions: planning, requirements analysis, design, coding, testing, and implementation.

Testing has been done throughout the iterations and therefore unit testing has not being used. Each of the iterations has ended with demonstrations to our client of what we have accomplished. The use of iterations in our project has minimised the overall risk for project failure and allowed the project to adapt to changes quickly. We had multiple iterations therefore throughout each of the phases we identified issues, bugs and future development plans.

G.2.2 Very short feedback loop and adaptation cycle

A common characteristic of agile development is daily status meetings. All members of the team are Computer Science students that used actively the communication channels that we have set and therefore updated each other instantly. More importantly, in the final stages of getting the visualisation working, we organised sub-group meetings so that work can be done quicker and so solutions could be reached quicker. The final stage has been dedicated to polishing our system and testing it but not excluding the fact that we were looking for points, which will prove or disprove our hypothesis.

G.2.3 Quality Focus

To improve the quality of our work we used tools and techniques, such as continuous integration [[69]], pair programming [[70]] and code refactoring [[71]]. These techniques enabled us to optimise our visualisation. Some refactoring took place to adhere to coding standards but it did not result in major

[69] "Continuous integration - Wikipedia, the free encyclopedia." 2005. 2 Dec. 2014 <http://en.wikipedia.org/wiki/Continuous_integration>

[70] "Pair programming - Wikipedia, the free encyclopedia." 2004. 2 Dec. 2014 <http://en.wikipedia.org/wiki/Pair_programming>

changes. Moreover, we have used testing in order to ensure that our product would work in any possible scenario, even in ones that seem unlikely to happen in real-life. Even more, we gave each other code reviews [72] to ensure that the work produced by every member met the team standards. Lastly, as was previously mentioned, for each iteration we would present our work to the client to ensure that the work produced met his expectations.

G.2.4 Agile Approach Conclusion

To conclude, the Agile approach that was chosen for our project was followed very well and we believe that it was the right choice to make. Our system has been complex, and needed an approach like Agile to drive the project forward, give us the flexibility and limit the resources wasted. In general, managing the project well and working hard, smart and consistently have been the keys to completing this project successfully.

---

[71] "Code refactoring - Wikipedia, the free encyclopedia." 2007. 2 Dec. 2014 <http://en.wikipedia.org/wiki/Code_refactoring>

[72] "Code review - Wikipedia, the free encyclopedia." 2004. 2 Dec. 2014 <http://en.wikipedia.org/wiki/Code_review>