CS344 Discrete Mathematics Project

Using Mapping APIs and Public Transport Data to Optimise Meet-up Destinations in Greater London

Progress Report

The University of Warwick

Zahra Kanji [1403922]

Contents

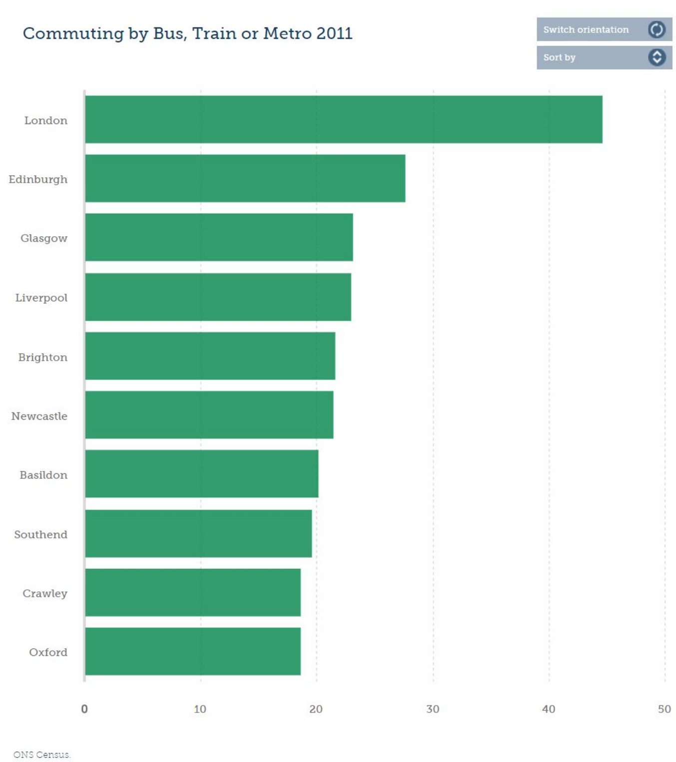
1 Introduction

At 1,572 km^2, [1] with a population of 8,173,941, [2] Greater London is the largest city in the UK, both in terms of land size and population. Unsurprisingly, therefore, it comes with a wide variety of transport options, people travelling from significant distances apart to meet up, and an abundance of locations for their meetings to take place. The aim of this project is to produce a web-application which, whilst taking into consideration the modes of transport being used by each individual in a group of people and the type of place they want to meet up at, finds optimal locations based on criteria which will be discussed later on in this report.

2 Background Research

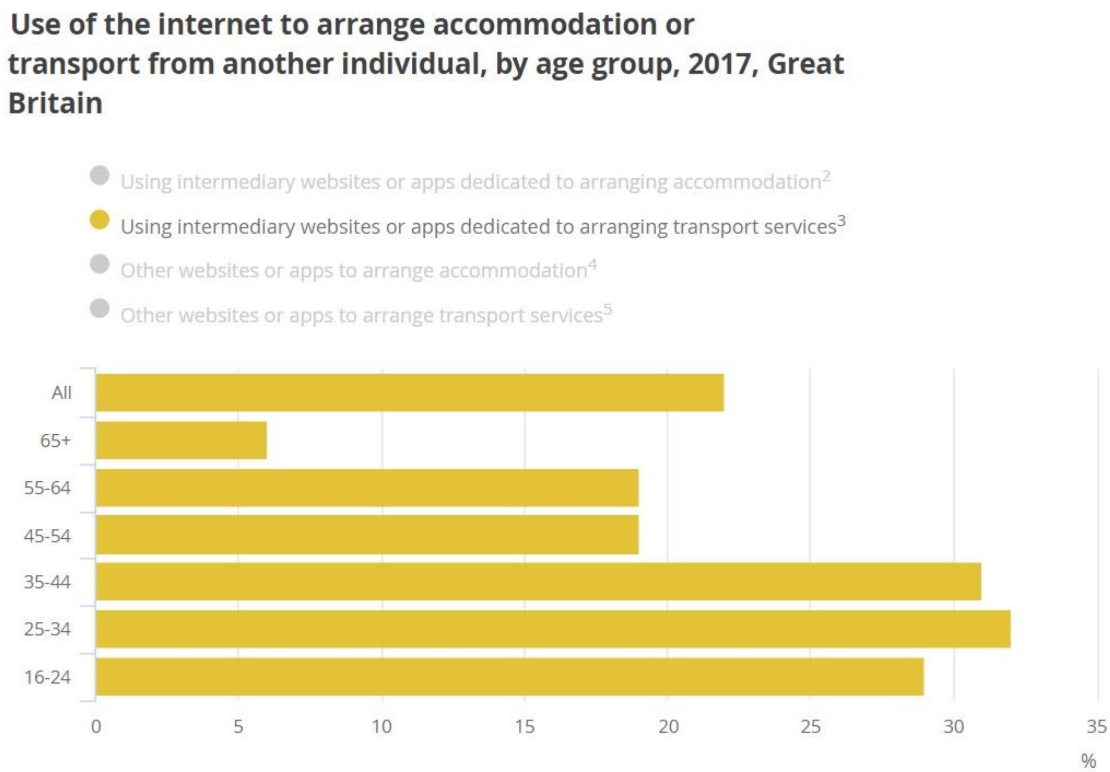
London is well-known for its extensive transport network, including, but not limited to, trams, tubes, and buses. [3] Furthermore, data indicates that public transport is much more widely used in London than in other cities in the UK. Figure 1 below [4] shows the percentage of commuters per city for some modes of public transport. It is evident here that the percentage of Londoners who use public transport for commuting (44.63%) is significantly higher than the percentage in other cities, and considerably higher than in the next city, Edinburgh (27.61%).

Figure 1



It can also be seen from available statistics in Figure 2 [5] that the internet is a medium which many people in Great Britain use to arrange their transport. Although the statistics do not explicitly mention the websites or apps included, this is still a good indication that the internet is used for planning journeys and transport. The popularity of public transport in London means that it is probable that these percentages are even higher in London.

Figure 2



London's public transport system is also continuously improving. The bus network, for example, has been heavily invested in over recent years. Bus routes have been extended and improved; weekend-only night bus services have been running more frequently since the introduction of the Night Tube services; [6] and by 2020, around 3,250 Zero Emission and Ultra Low Emission buses will be on the streets of London.

2019 will see two more important improvements to the London Underground services; the Elizabeth Line will be opening, [7] and commuters using the tube network will have access to 4G mobile coverage. [8] There are no signs of public transport services in London slowing down, and in the future, public transport will only continue to increase in popularity amongst Londoners. Despite all this, many of the few existing applications for finding meet-up destinations do not take public transport into consideration.

2.1 Examples

In the specification for the project, Table 1 below was created showing existing applications which solve similar problems of finding "halfway" points for people to meet up at.

Table 1: Features in different websites

	www.roudle.com	www.whatshalfway.com	www.meetways.com	a.placebetween.us	www.wheretomeetup.com	www.shallwemeetinthemiddle.com	www.geomidpoint.com/meet/
aesthetically pleasing	x	x	xx		xx	xx	
different modes of transport	x		x			x	
equal travel time		xx					
equal travel distance	x	xx	xx			xx	xx
meetup time							
destination ratings	x	xx	xx		xx		xx
includes directions		x	xx	x		x	x
can be used in London		xx	xx	xx	x	xx	xx
works for more than 2 people	xx	xx		xx			xx
display responsive	x		xx	x	x	xx	
destination type	x	xx	xx	xx	xx		xx
passed my algorithm test*		xx			x		x

*a test to determine how well the algorithm performs in an unusual situation - to find an optimal cinema for two people where the optimum cinema is not directly between their two addresses

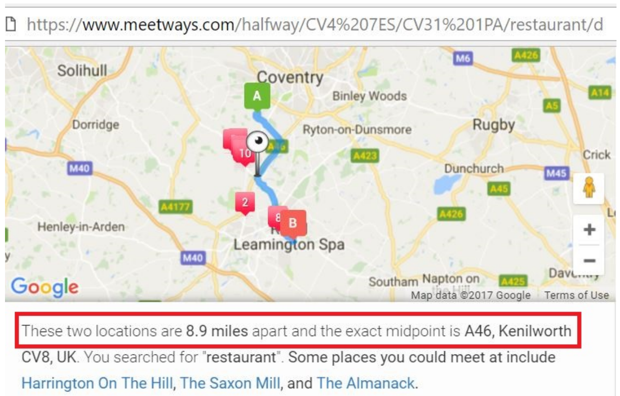
Table key:
x - feature implemented
xx - feature implemented very well

Two of the applications from this list were created as direct results of frustration. Meetways.com was created after a member of their team considered "how many hours she had wasted in the car driving more than her fair share of miles to meet up with clients". [9]

One of the founders of Roudle.com had similar things to say, although his vision was more about saving travel distance overall. When discussing the outcome of choosing where to meet up based on intuition, he said "...when I actually calculated the various distances everyone would have to travel to Utrecht, it turned out we could save about 20% of travel distance by meeting up in a different place.” A Roudle developer went on to say, “Apps or websites that can help solve this problem are virtually non-existing, even in today’s world. In the few cases they have been developed, these tools simply refer to the middle point on the map, not taking into account the complexity of the roads and real travel times.” [10]

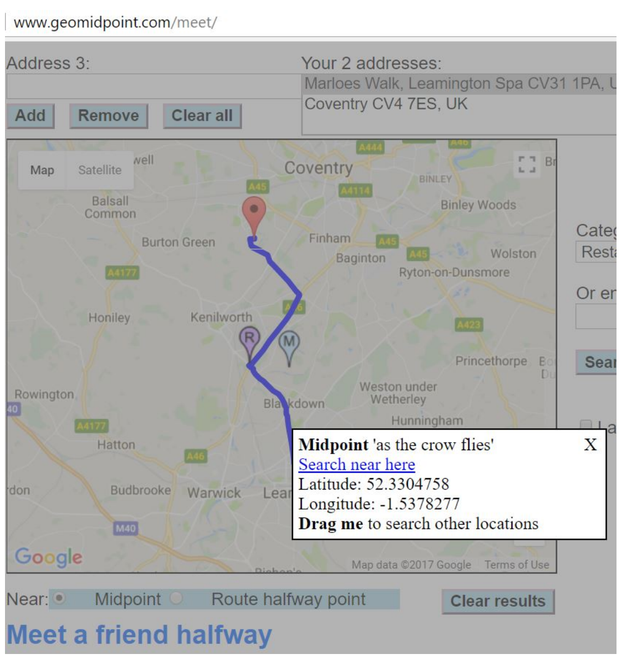
While researching web-applications of this nature in depth, the truth behind these comments became increasingly clear. Many of the web-applications researched followed a similar algorithm. On Meetways.com, for example, what appears to be the midpoint of the route between two addresses is located, and destinations to meet up around this area are found and suggested to the users. [11]

Figure 3



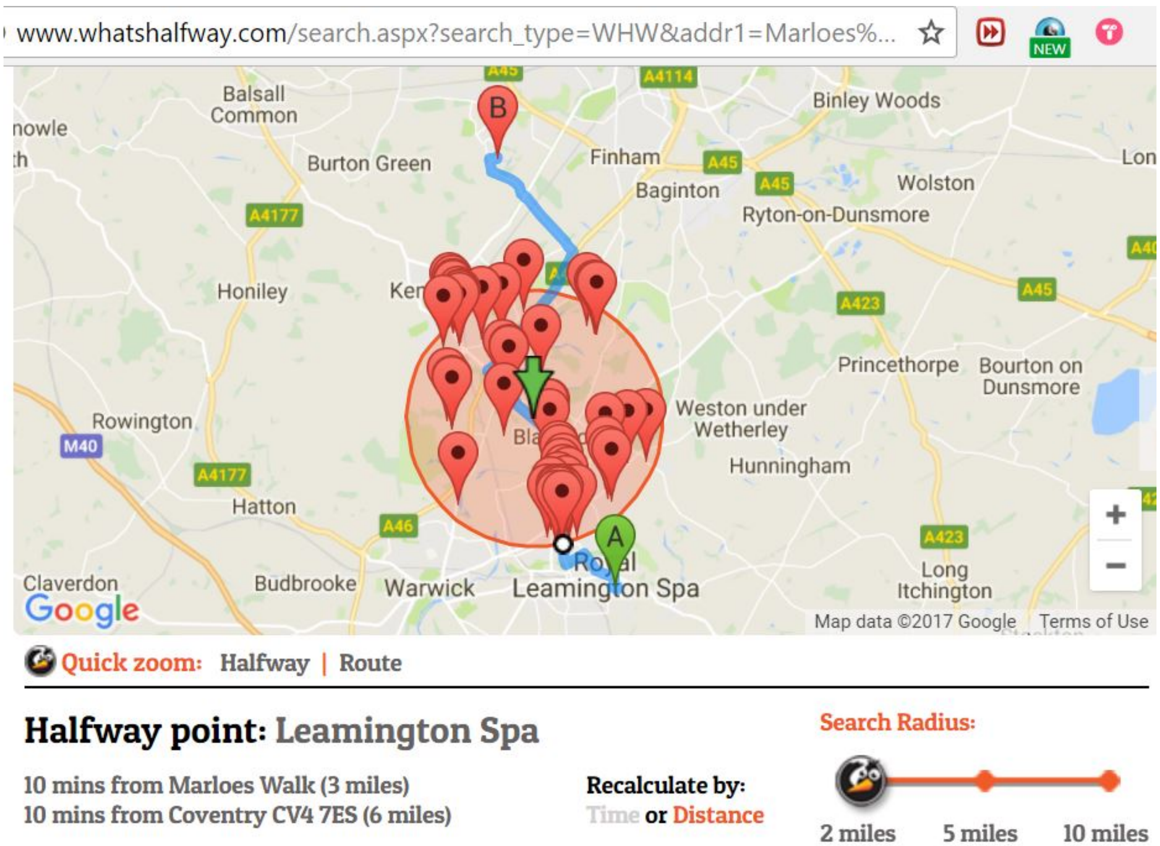
Geomidpoint.com follows a similar principle, but it allows the user to select either the geographical midpoint (as the crow flies), or, when there are only two users, the midpoint of the route between them. [12]

Figure 4



Whatshalfway.com is the only application I could find which takes into consideration the time travelled by users and this feature works very well for many example cases that were tested. The application appears to use a heuristic algorithm and it allows the user to improve search results by increasing the area being searched on the map. The only shortcomings of this web-application is that it does not consider public transport and is not mobile-friendly. [13]

Figure 5



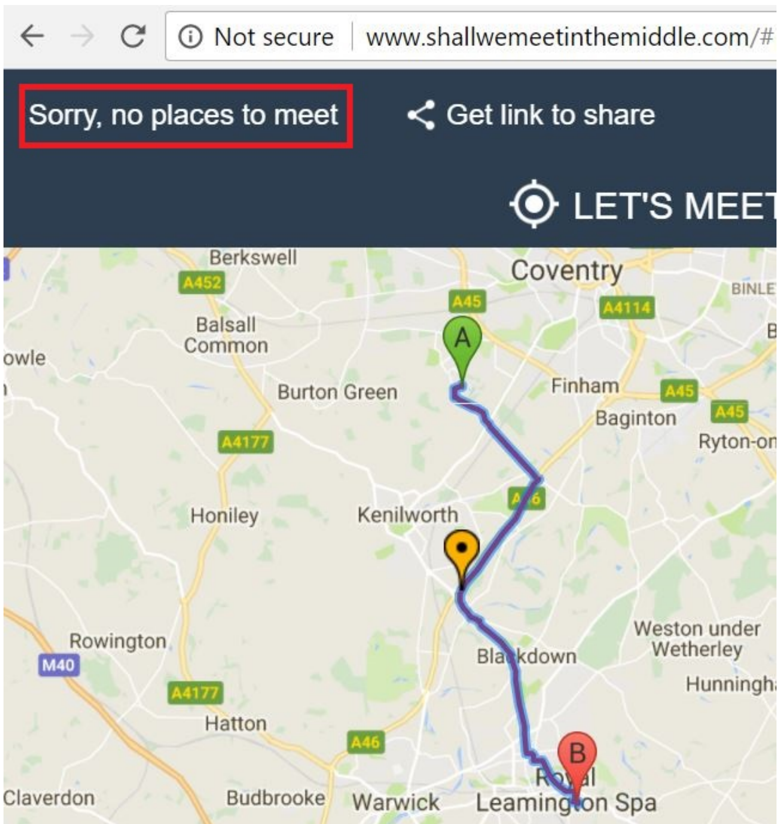
Roudle.com, mentioned at the start of this section, gives the optimum meeting points based on actual travel times, and also considers public transport in the Netherlands. [14] Alongside Whatshalfway.com, it is the closest to incorporating the features that this project aims to.

2.2 The Gap

As mentioned in the previous section, the people who produced the applications did so out of frustration, and this reflects similar, and evidently common, frustrations expressed by Londoners who have been spoken to about the idea behind the project.

Only two of the applications that work in the UK consider different modes of transport and one of them, seen in Figure 6, does not even show results [15] for the same starting locations shown in Figures 3, 4, and 5.

Figure 6



Given how important public transport is in London, these facts together provide a good justification for this project. Moreover, having never completed a web development project before, this project will be appropriately challenging. There is a development aspect to the project, as well as research and an algorithm that will require Mathematics, making it a good and balanced project for a Discrete Mathematics student.

2.3 Objectives

Ideally, the system should include all the features listed in Table 1, however not all of these are required to meet the overall aim of the project. For this reason, I have split them into primary objectives, those which are required, and secondary objectives, those which improve the user experience.

2.3.1 Primary Objectives

1. Create an intuitive User Interface
   (a) Allow up to n* starting addresses to be inputted
   *this number will be finalised after carefully considering the appropriate number of API calls which is explained later on in the report
   (b) Allow mode of transport for each traveller to be selected
   (c) Allow type of destination to be chosen
   (d) Allow either arrival or departure time to be selected
2. Produce an algorithm to recommend optimal meet-up destinations
   The three choices for the user will be:
   (a) Minimising the total time travelled
   (b) Minimising the total distance travelled
   (c) Finding the fairest* location with regards to the length of time spent travelling
   *this concept of "fairness" is discussed and elaborated upon later on in the report
3. Show the suggested destinations in relation to the starting addresses on a map to visualise the journey for each person

2.3.2 Secondary Objectives

1. Improve the UI to make it aesthetically pleasing and responsive to different screen resolutions
2. Include ratings of the suggested meet-up locations
3. Rank the suggested locations
4. Extend the UI to show directions for each person travelling

2.3.3 Possible Extensions

1. Allow a link to be created for each traveller's directions
2. Provide a feature for midpoints and starting addresses to be saved for later use

3. Consider prices
    (a) Display the price of each journey using Transport for London data and petrol prices in the UK
    (b) Extend the algorithm to minimise total cost and find the fairest cost

2.3.4 Optimisation and Fairness

Originally in the specification, the options that would have been given to the user were:

(a) Geographical midpoint

(b) Travelling midpoint

(c) Timed midpoint

To translate these into something more tangible, the following options mentioned in the objectives above are more appropriate:

(a) Minimising the total time travelled

(b) Minimising the total distance travelled

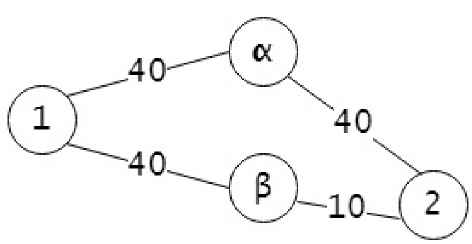
(c) Finding the fairest location with regards to the length of time spent travelling

A location will be considered "fair" if it is not too far away from everyone in terms of distance, and also if everyone is travelling as close as possible to similar lengths of time. This will be explored further and defined explicitly when the mathematical functions are considered, but some of the things that will hopefully be avoided include:

- Choosing a place unnecessarily far away simply because, for all the people travelling, their journeys would be the same length of time

- If person 1's journey is the same length of time regardless of whether person 2 travels for a longer or shorter length of time, it does not make sense for person 2 to have to travel for longer unnecessarily simply for the sake of "fairness", and this should be reflected in the fairness function.

Figure 7



In Figure 7 above, 1 and 2 are people (or starting locations); α and β are possible destinations; and the edge weights are the length of the journeys between them in minutes. In this case, α should not be chosen over β because person 2 does not make person 1's journey shorter by travelling for longer.

3 Current Progress

Because of the nature of the project, it can be split into some clear parts: the algorithm, the implementation of this algorithm, the front-end of the web-application, the back-end, and connecting these.

3.1 Algorithm

3.1.1 Original Approach

Originally, the plan was to start looking for locations matching the user's description around the geographic midpoint of the starting addresses, much like the websites mentioned earlier in the report. Then, if none of the places in that area are fair enough, increase the search radius and see what else is available.

This method would work quite well for people who live close to each other because there would not be too large a number of places to look up. However, this approach becomes more problematic under different circumstances, such as when travel time to the destination is significantly affected. For example, when people are either travelling from far apart as there are many nuances to public transport in London, or when users are travelling by different modes of transport; if one user walks whilst the other drives, their travel times to reach the geographic midpoint would be significantly different.

It could be possible to move this radius closer to the people travelling longer, but it was not initially clear whether this would be a simple task or whether it would require some complicated considerations. Lastly, since all options are not being considered, there is a chance that the optimal solutions are being missed.

3.1.2 New Approach

Consequently, after some research, a new method was suggested. This method relies on linear algebra and requires "fairness" and "optimal" to be defined very clearly from the start. The way this method works is by finding all the possible locations of a given type in London, and then composing vectors using data that is relevant to these definitions, such as distance, for example. Once this function is applied, each vector is reduced to a number, making ranking and selecting options very easy.

For example, if a function were required to rank locations based on lowest overall travel distance between the users, we could have:

Let

$$L := \text{the set of all possible meet-up locations}$$

and

$$P := \text{the set of starting positions}$$

Then

$$\forall\ l \in L, \text{ we create the vectors } V_l$$

where

$$V_l \coloneqq \text{a vector such that, } \forall\ p \in P, \text{ each row of } V_l \text{ is } d_{l,p}$$

where

$$d_{a,b} \coloneqq \text{the distance between a and b.}$$

If this set of vectors is

$$V \coloneqq \{V_l \mid l \in L\},$$

a simple function f can be applied to each of the vectors $v \in V$, reducing them to a single number, for example the sum of every element of the vector:

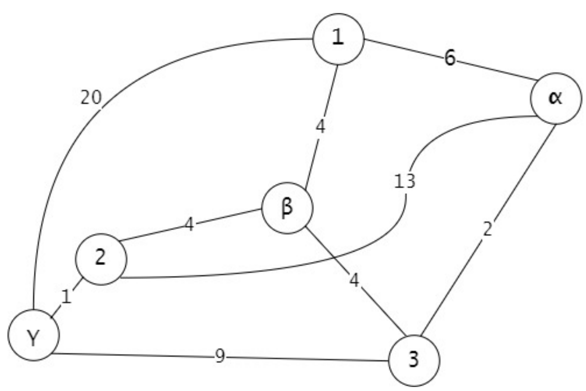
$$f(v) = \sum_{i=1}^{n} v_i$$

$$\text{where n} \coloneqq dim(V_\alpha).$$

The results of applying this function could then return the locations in order of lowest overall travel distance to highest.

For example, the graph in Figure 8 shows the distances between starting locations, $P = \{1, 2, 3\}$, and possible destinations, $L = \{\alpha, \beta, \gamma\}$:

Figure 8



Let

$$V = \{V_\alpha, V_\beta, V_\gamma\}$$

where

$$V_\alpha = \begin{pmatrix} d_{\alpha,1} \\ d_{\alpha,2} \\ d_{\alpha,3} \end{pmatrix} = \begin{pmatrix} 6 \\ 13 \\ 2 \end{pmatrix},$$

$$V_\beta = \begin{pmatrix} d_{\beta,1} \\ d_{\beta,2} \\ d_{\beta,3} \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix},$$

$$V_\gamma = \begin{pmatrix} d_{\gamma,1} \\ d_{\gamma,2} \\ d_{\gamma,3} \end{pmatrix} = \begin{pmatrix} 20 \\ 1 \\ 9 \end{pmatrix}$$

The function f is applied to all $v \in V$, giving

$$f(V_\alpha) = 6 + 13 + 2 = 21,$$

$$f(V_\beta) = 4 + 4 + 4 = 12,$$

$$f(V_\gamma) = 20 + 1 + 9 = 30$$

When these are ordered from lowest to highest, we have

$$12 \leq 21 \leq 30$$

$$\implies f(V_\beta) \leq f(V_\alpha) \leq f(V_\gamma)$$

$$\implies \text{the order of locations should be } \beta \text{, then } \alpha \text{, then } \gamma$$

This seemed like a much better approach as every possible destination option would be considered and, if the optimisation and fairness functions are defined carefully, the algorithm has the potential to display very good results. However, it soon became clear that there are several thousand establishments in London of places to meet and socialise; there are at least 3,000 pubs, bars and nightclubs, [16] and over 65,000 restaurants, takeaways or food shops. [17]

This number of destinations is colossal, especially when considering API usage limits, and being able to access all of these locations would most certainly exceed these usage limits for the APIs I had chosen.

A solution to searching for all these locations in real-time would be to do some pre-processing and have them stored in JSON form or in a database along with all the information about them that would be needed later for the algorithm to produce results.

Another problem was that for each user, data is required about the journey between them and each location in question. With k number of users, and m number of locations, we see that to find distance or travel time, the number of calls that needs to be made to the API reaches a very high number, and subsequently either requires an unreasonable wait time for a result, or exceeds the usage limits once again.

One solution to this problem could have been to make a heuristic algorithm by implementing some rules to reduce the number of searches that have to be made – for example reducing the number of relevant locations by ignoring others within a 5 minute walk from each other, or discretising a map of London and implementing some pre-processing with the intention of showing average travel times between places in London. However it is not guaranteed to reduce the number of locations to an appropriately low quantity.

The best solution to this problem would be the ability to find the travel times required offline, however this is not necessarily achievable without downloading a great deal of raw data. This data would then need to be processed to produce an algorithm which returns travel-times, but Google Maps, OpenStreetMaps, the TFL Journey Planner, and many other services already do this. This is therefore outside the scope of the project as it would not be sensible to prioritise this and reinvent the (already impressive) wheel given the time constraints.

Whilst searching for a solution, an application called Here Maps came to my attention. [18] It allows users to download a local area map and, without connection to the internet, also tells users the length of their journey using any form of transport. This is something that very few apps have achieved – Interrail is one other [19]. I contacted Here Maps to see whether their service would be able to solve the problem, and was told that this would cost money. I also contacted Roudle to see if they might be able to guide me towards a solution, as they use public transport data for a part of their web-application.

This shows that the aims of the project are certainly achievable, however it is not worth the time to solve this specific problem of being unable to access data offline. My supervisor agreed that a more appropriate solution would be to show that the algorithm produced will work for a specific type of location of which there are a smaller number of options, such as cinemas, which can be extended to incorporate other venue types in the future.

This new approach to the algorithm is also preferable because the objectives of the project include allowing users to choose between a different options and definitions of “optimal”. This method will make it easy to create different functions, each corresponding to an option available on the web-application.

Some limitations that will be included to reduce the number of API calls are:

- Only one mode of transport per user
- Some maximum number of starting locations (however the algorithm will be easily modifiable to include more)
- A limited number of location types (for now, to show that my solution works)

3.2 Technical

There is a lot of data required for this project: time taken to travel between places, places matching a given description, public transport data, directions between places, maps, ratings, and there are also many different APIs available which I have been researching.

Now that I have a better understanding of which features I will need because there is a clearer understanding of the algorithm, selecting APIs will not be too difficult. Where possible, the web-app will stay as close as possible to the familiarity of Google Maps, however some other options that have been considered include:

- Places: Foursquare, Yelp, TripAdvisor [20]
- Maps: OpenStreetMap, MapBox [21]
- Public Transport: TransportAPI, CityMapper, Transport for London Unified API [22]

3.2.1 Back-End

The Back-End of the program will either be written in Node.JS or Python, depending on which has better documentation for the chosen API. I have already started to use Node.JS to make calls to the Google Maps Directions API, and have found that it is time consuming to learn Node alongside the API, so I may use Python instead as I am more familiar with this language.

3.2.2 Front-End

For the front-end, I will be using HTML, CSS, JavaScript, Bootstrap, and either Flask or Express depending on whether I use Python or Node.JS respectively. I have started to learn how to use HTML, CSS, and JavaScript, and have already considered the elements that will be needed on the front-end.

4 Future Steps

4.1 Appraisal and Reflection

Against the initial criteria and considering the problems that have been encountered, the project is relatively on track against its timeline. Keeping on track with the tasks set out in the initial specification, the specification has been written, and web development technologies, as well as mapping and public transport APIs have been researched.

The only task yet to be achieved is the basic user interface, however progress has been made in this area. Progress has also been made on creating the algorithm, and, with more clarity about what needs to be done, there is no need to create an algorithm just to work for two people now. Removing this from the timeline should allow time to complete the primary objectives and then continue with the secondary ones.

5 Ethical Considerations

The ethical considerations at this stage have not changed since the specification, with the exception of one addition. In correspondence with Roudle, they have made it clear that the commercial nature of their product limits how much they are willing to discuss their algorithm, and this must be kept in mind.

- Whichever APIs end up being used for the project, it is important that the terms and conditions are read and understood carefully to ensure the project abides by them.
- Any user location information used and stored by the system will not be redistributed for the safety of the users.
- For the safety of users, it may be necessary to consider vetting the locations suggested by the system. This can potentially be done by using Yelp, Google, or TripAdvisor reviews.
- When user acceptance testing takes place, the participants will be from my social circle, so formal consent will not be required.
- Ensure that the privacy of any external people, such as Roudle, is respected.

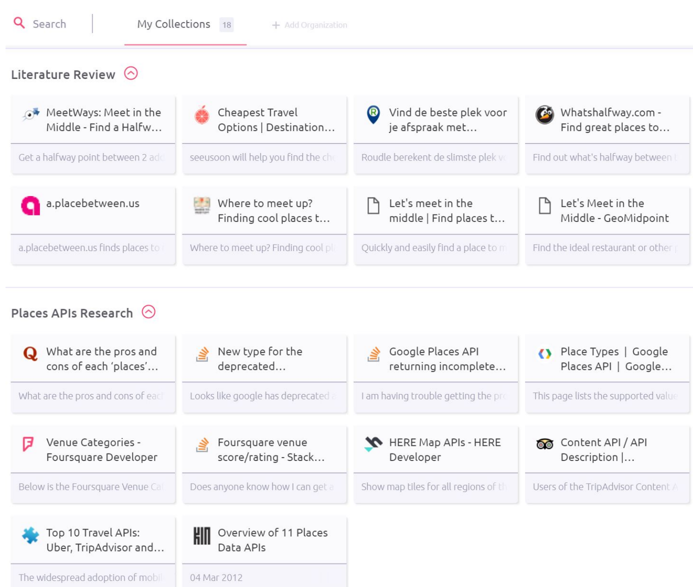
6 Project Management

6.1 Resources

Alongside the resources mentioned in my specification, I have also found two other useful tools for project management. The first is Cloud9, which allows online development from a web browser so work can be accessed from any device. This has helped and will help a great deal with working between lectures with minimal disruption, and when showing progress to my supervisor.

The latter is Toby for Chrome, the purpose of which is to organise browser tabs and research. [23]

Figure 9



6.2 Specification Changes

The main changes to the specification throughout this document have been more robust definitions of each aspect of the project, as these were quite vague beforehand. This includes the justification for the project; the objectives for the project; the definition of "optimal"; and the approach to the algorithm. The research done over this term has clarified most of these things, and this should encourage a smoother journey to the required solution.

6.3 Unforeseen Circumstances

There have been some unforeseen circumstances limiting progress this term, such as the problem of not being able to access data quickly, but the agreement with my supervisor to limit the problem but still show that the solution can work should avoid any more problems.

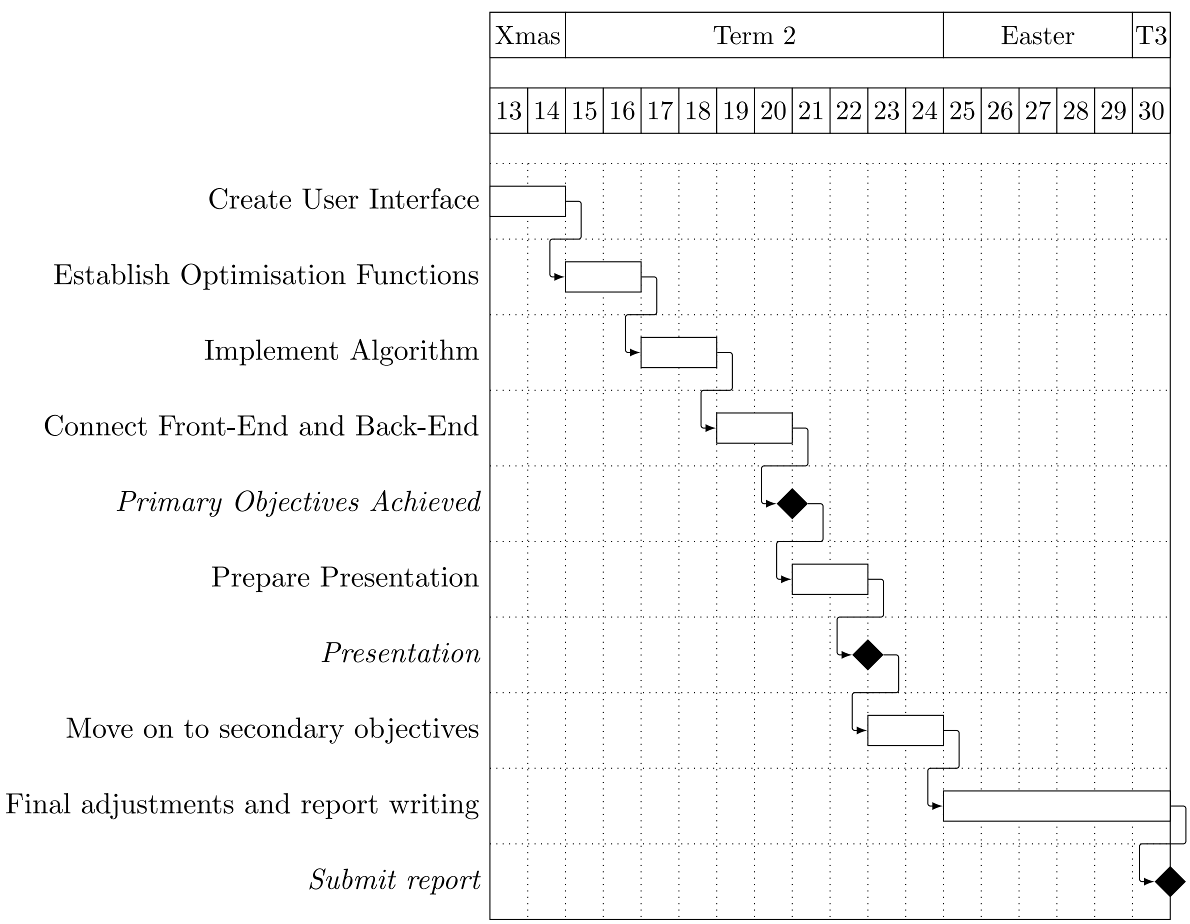
I have also had some personal situations affecting the amount of progress I could make this term, but I hope to rectify this and stick to the new Gantt chart below, outlining the steps that will be taken during term 2.

6.4 Gantt Chart

It became apparent that trying to schedule every part of the project was not working very well because of the problems encountered. So for the next stage of the project, the agile methodology will be used properly. All the steps that will take place have been split into 2 week sprints which should allow for any problems to be fixed, as well as for catching up if there have been any problems, and will also allow for other commitments such as lectures and coursework for other modules.

It was also clear this term that I need a more structured approach to the project for the best results, so I intend to schedule specific slots into my timetable next term to ensure greater focus and quicker progress.

Table 2



7 References

[1] Wikipedia. *Greater London.* URL: https://en.wikipedia.org/wiki/Greater_London.

[2] New Statesman. *Where are the largest cities in Britain?* URL: http://www.citymetric.com/skylines/where-are-largest-cities-britain-1404.

[3] Transport for London. *Stations, stops & piers.* URL: https://tfl.gov.uk/travel-information/stations-stops-and-piers/.

[4] Office for National Statistics. *Commuting by Bus, Train or Metro 2011.* URL: http://www.centreforcities.org/data-tool/#graph=bar&city=show-all&indicator=commuting-by-bus-train-or-metro%5C%5Csingle%5C%5C2011&sortOrder=high&tableOrder=tableOrder%5C%5C1%7B%7D,1.

[5] Office for National Statistics. *Use of the internet to arrange transport from another individual, by age group, 2017, Great Britain.* URL: https://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/bulletins/internetaccesshouseholdsandindividuals/2017.

[6] Transport for London. *Improving Buses.* URL: https://tfl.gov.uk/modes/buses/improving-buses.

[7] Transport for London. *Countdown to Launch.* URL: https://tfl.gov.uk/travel-information/improvements-and-projects/countdown-to-launch.

[8] The Londonist. *4G Mobile Coverage on the Tube from 2019.* URL: https://londonist.com/london/transport/4g-mobile-coverage-on-the-tube-from-2019.

[9] Meetways.com Kristen McCaffrey. *Company Overview.* URL: https://www.meetways.com/about.

[10] Roudle.com Christiaan Rasch & Thijs Brentjes. *Press Release Launch Roudle.* URL: https://mobilitylabel.nl/press-release-launch-roudle/.

[11] *MeetWays: Meet in the Middle - Find a Halfway Point!* URL: https://www.meetways.com/.

[12] *Geo Midpoint, Let's Meet in the Middle.* URL: http://www.geomidpoint.com/meet/.

[13] *What's Halfway? Find great places to meet or stop halfway between...* URL: http://www.whatshalfway.com/.

[14] *Vind de beste plek voor je afspraak met locatieplanner Roudle.* URL: https://www.roudle.com/.

[15] *Let's Meet in the Middle.* URL: http://www.shallwemeetinthemiddle.com/.

[16] Food Standards Agency. *Food Hygiene Ratings - pub/bar/nightclub.* URL: http://ratings.food.gov.uk/enhanced-search/en-GB/%5E/London/Relevance/7843/%5E/%5E/0/1/10.

[17] Food Standards Agency. *Food Hygiene Ratings - restaurants, takeaways or food shops.* URL: http://ratings.food.gov.uk/search-a-local-authority-area/en-GB/London.

[18] HERE Maps Pino Bonetti. *Comparing Google Maps and HERE Maps offline*. URL: https://360.here.com/2015/11/17/comparing-google-maps-and-here-maps-offline/.

[19] Interrail. *Rail Planner App*. URL: https://www.interrail.eu/en/plan-your-trip/rail-planner-app.

[20] API Evangelist. *Overview of 11 Places Data APIs*. URL: https://apievangelist.com/2012/03/04/overview-of-11-places-data-apis/.

[21] ProgrammableWeb Staff Janet Wagner. *Top 10 Mapping APIs*. URL: https://www.programmableweb.com/news/top-10-mapping-apis-google-maps-microsoft-bing-maps-and-mapquest/analysis/2015/02/23?utm_source=self&utm_medium=paralinks&utm_campaign=related-content.

[22] James Peckham. *TfL will never make the ultimate travel app - but there is a very good reason*. URL: http://www.techradar.com/news/phone-and-communications/mobile-phones/tfl-will-never-make-the-ultimate-travel-app-but-there-is-a-very-good-reason-1304247.

[23] *Toby for Chrome*. URL: https://chrome.google.com/webstore/detail/toby-for-chrome/hddnkoipeenegfoeaoibdmnaalmgkpip?hl=en.

Using Mapping APIs and Public Transport Data to Optimise Meetup Destinations in Greater London

Zahra Kanji [1403922]

October 12, 2017

Contents

1 Problem Statement

Greater London is the biggest city in the UK, both in terms of land size and population[1]. As such, it comes with a wide variety of transport options, many places to meet up, and people travelling from significant distances apart. The aim of this project is to produce a web-application which takes into consideration the modes of transport being used by a group of people, and the type of place they want to meet at, to find a location which takes the same length of time for everyone to travel to.

2 Background Research

Currently, one of the most popular navigation tools is Google Maps[2], and more specifically, Citymapper is one of the most popular tools specifically for Londoners[3]. Looking further at these will be useful to establish intuitive and popular layouts for the user interface, especially when making the web-app responsive for smaller screens.

There are already websites which attempt to optimise meetup destinations, but none of then incorporate all the features in Table 1.

Statistics also show us that it is common for people to use the internet to plan their transport.[4] The aim for this project, therefore, is to produce a web application which implements all the features listed in Table 1.

In order to achieve this, there are many resources available. For example, the websites listed in Table 1 all use Google Maps APIs, with the exception of Roudle and wheretomeetup which use Leaflet and MapBox respectively, both of which rely on OpenStreetMap data. Furthermore, there are many different Google Maps APIs which all serve different purposes and are relevant for different features. For example, the Google Maps Distance Matrix API calculates the travel distance and travel time for multiple origins and destinations and the Google Maps Directions API gets directions data. Some of these features are also available in the Google Maps Javascript API. On top of this, there is the Transport for London unified API, so a big part of the project will be researching these in detail to find the most appropriate choices to achieve the objectives of the project which are listed below.

Table 1: Features in different websites

	www.roudle.com	**www.whatshalfway.com**	**www.meetways.com**	**a.placebetween.us**	**www.wheretomeetup.com**	**www.shallwemeetinthemiddle.com**	**www.geomidpoint.com/meet/**
aesthetically pleasing	x	x	xx		xx	xx	
different modes of transport	x		x			x	
equal travel time		xx					
equal travel distance	x	xx	xx			xx	xx
meetup time							
destination ratings	x	xx	xx		xx		xx
includes directions		x	xx	x		x	x
can be used in London		xx	xx	xx	x	xx	xx
works for more than 2 people	xx	xx		xx			xx
display responsive	x		xx	x	x	xx	
destination type	x	xx	xx	xx	xx		xx
passed my algorithm test*		xx			x		x

*a test to determine how well the algorithm performs in an unusual situation - to find an optimal cinema for two people where the optimal cinema is not directly between their two addresses

Table key:
x - feature implemented
xx - feature implemented very well

3 Objectives

Ideally, the system should be able to include all the features listed in the table above. However, not all of these are inherently required to meet the overall aim of the project. Therefore I have split them into primary objectives - those which are required - and secondary objectives - those which improve the user experience.

3.1 Primary Objectives

1. Create a basic but intuitive User Interface
   (a) Allow two starting addresses to be inputted
   (b) Allow mode of transport to be selected
   (c) Allow type of destination to be chosen
2. Produce an algorithm to optimise meetup destination for two users considering
   (a) Geographical midpoint
   (b) Travelling midpoint
   (c) Timed midpoint
3. Show the suggested destinations in relation to the starting addresses on a map to visualise the journey for each user

3.2 Secondary Objectives

1. Extend the algorithm to work for larger groups of people
2. Improve the UI to make it aesthetically pleasing and responsive to different screen resolutions
3. Include ratings of the suggested meetup locations
4. Extend the UI to show directions for each person travelling
5. Provide a save feature for midpoints and starting addresses to be saved for later use

3.3 Possible Extensions

1. Consider prices
   (a) Display the price of each journey using petrol prices in the UK and TFL data
   (b) Extend the algorithm to work for finding a destination that costs the same to travel to for each user
2. Include the choice to reduce overall length, distance, or price travelled by the users together

4 Project Management

4.1 Methodology

The most appropriate methodology for this project is an agile approach as it will help to prioritise a functioning system and can adapt to changes that need to be made as the project progresses.

Unit testing will be implemented at the end of each stage of development to ensure that the system works accurately and correctly. I will try to test specific cases with unusual answers to test the algorithm too.

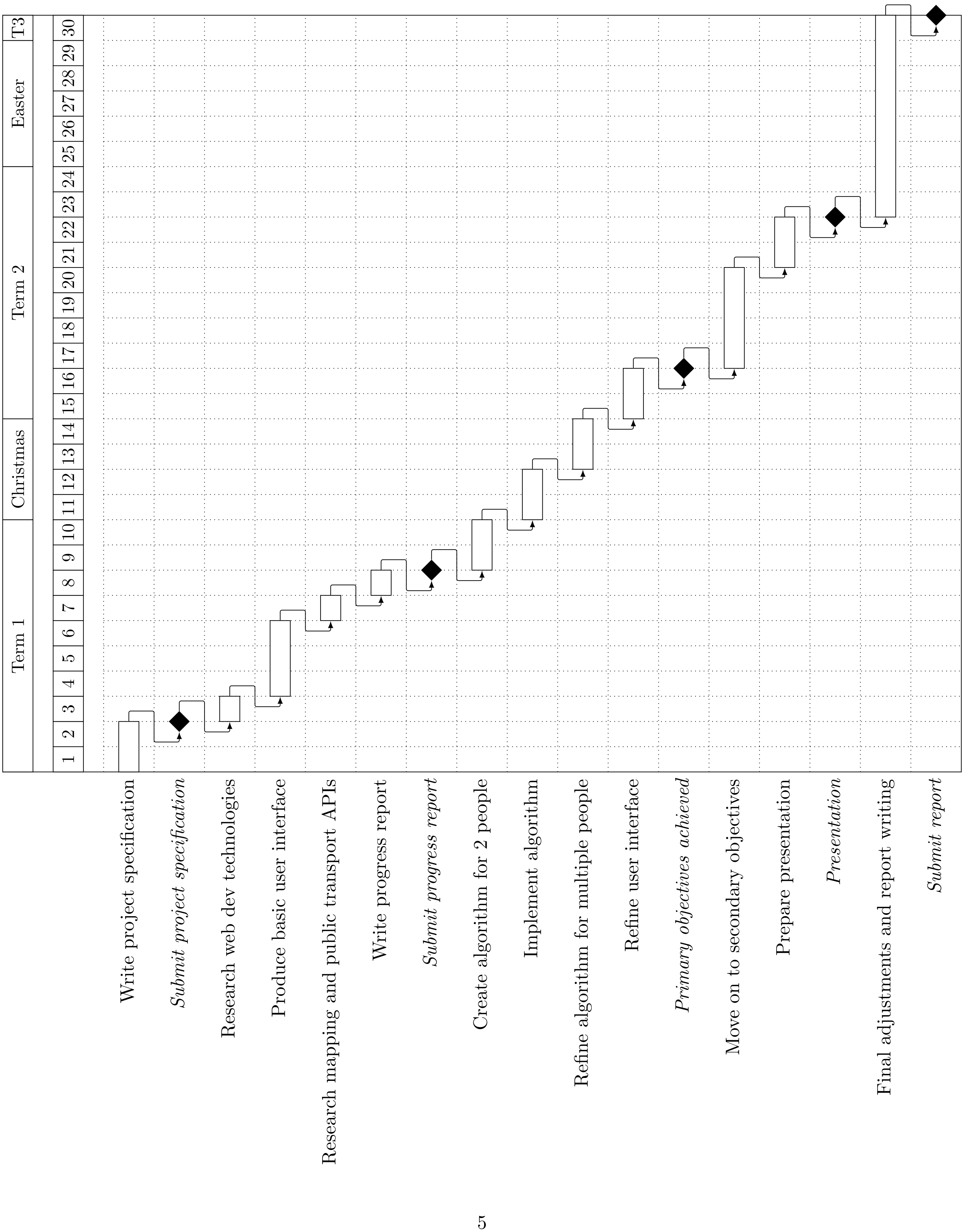
User acceptance testing will also take place to ensure the system is beneficial for the people who will be using it.

4.2 Timeline

In order to meet the 300 hours of work for this project, I plan to spend an average of 10 hours a week on the project. I will review my project weekly. I also intend to meet my supervisor every fortnight to review progress.

The timeline of the project can be seen in the Gantt Chart in Table 2.

Table 2: Gantt Chart



4.3 Resources

- Git
  for version control

- GitHub
  for documenting and updating changes

- Trello
  for keeping track of tasks

- Javascript
  for the User Interface

- Mapping and public transport APIs
  to be researched, but for data essential to the system

4.4 Risks

1. One potential risk of this project is that I have not used almost all the technology before, which has the potential to slow down progress. To avoid this, I will endeavour to spend some extra hours a week using available resources to understand them, and will also choose the APIs easiest for beginners where appropriate.

2. Other risks which could slow down progress of the project include technical difficulties, illness, and other coursework. Where possible, I will take note of busy times in advance and ensure I am still reaching my average of 10 hours per week on the project.

4.5 Ethical considerations

- Whichever APIs end up being used for the project, it is important that the terms and conditions are read and understood carefully to ensure the project abides by them.

- Any user location information used and stored by the system will not be redistributed for the safety of the users.

- For the safety of users, it may be necessary to consider vet the locations suggested by the system. This can potentially be done by using Yelp, Google, or TripAdvisor reviews.

- When user acceptance testing takes place, the participants will be from my social circle, so formal consent will not be required.

References

[1] New Statesman. *City Metric.* URL: http://www.citymetric.com/skylines/where-are-largest-cities-britain-1404.

[2] The Nielsen Company. *Top Smartphone Apps of 2016.* URL: https://cnet1.cbsistatic.com/img/d2LMVbJphEJnjqtke0UBWxMS9rM=/2016/12/28/1058c62c-271d-402d-ac89-64b49aeec2f1/tops-digital-chart-1.png.

[3] Growth Business. *Citymapper.* URL: http://www.growthbusiness.co.uk/navigation-app-citymapper-looks-to-new-cities-after-securing-balderton-capital-and-index-ventures-as-investors-2461427/.

[4] The Office for National Statistics. *Internet access – households and individuals: 2017.* URL: https://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/bulletins/internetaccesshouseholdsandindividuals/2017.