**Analysis**

**Investigation**

The problem I am trying to solve is to do with organising bus-based travel in the Guildford area. The issue is that the timetables are too complicated to understand and plan trips around and, unless you do hours of research beforehand, chances are you are not going to get the best price for your journey.

**Sub-Problems:**

The largest problem I am going to have to overcome whilst creating this program is entering the data I require from different websites as I require a range of pricing and timing information. This is an issue because I either have to copy down the data from the individual websites to my database or create a custom web scraper to collect the information for me and format it into the database.

Another problem I am going to have is the storage of data in a place, which it can access via my program. In order to do this, I would either have to set up a web server, which I would keep constantly running at my house or I could have the data downloaded with the application, which would mean it would not need a constant internet connection to run the program.

**Sub-Argument**

One advantage of using a web server is that it would allow me to store all the data required for the website without needing to download it onto any device that would use the application every time I needed to use it on a different device. This means it takes less time to download as well as taking up less storage space, which is especially useful for smaller devices such as mobile phones.

One disadvantage of using a web server is that the application would become unusable unless you are connected to a fast internet connection. This means that if you are in the middle of nowhere the app becomes virtually unusable. However, if I were to download the data onto the device alongside the application, the user would be able to access the app anytime without a required internet connection.

Another advantage of using a web server is that it will allow me to provide updates to my database based on the constantly updating bus timetable system without having to update the application, as the code will not change. This allows for a better experience for users as the app will only be down for a minimum amount of time at any one time.

Another disadvantage of using a web server is that I would have to set up a server at home, which would take a lot of time to set up as well as being expensive in terms of power and resources to set up and keep running so I would have to make sure the hardware is capable of running constantly. Whereas if the data came with the app when it was downloaded, it would cost me nothing in expenses.

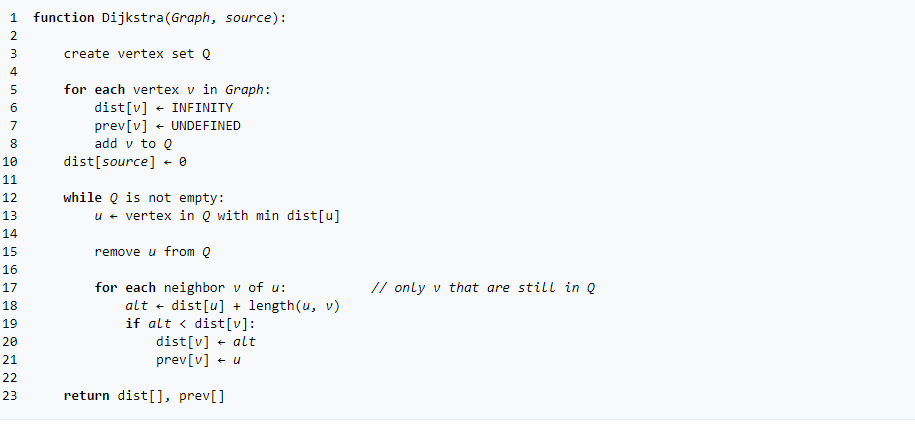
An additional sub-problem is the fact that I have to learn both kotlin and swift to create applications in both of those languages without any consequential runtime or syntax errors that are caused by automated translation, which often comes with using an automatic translator.

Another sub-problem is calculating the lowest price and lowest travel time from a collection of data within a time range. Theoretically, this is a simple case of taking two lowest values and searching through the data to get the best of both. This is easy to do by eye but less easy to do via an algorithm. My plan is to order all of the prices and times from lowest to highest, assign a weight to each value based on this order then assign them back to their label and total the weights. This will allow me to get the best of both values as well as the best individual price and time.

A further problem is that I am going to have to create a user interface that looks nice and makes the functions easy to use, does not put strain on the eyes and attracts new users. I plan to do this by creating a simplistic looking user interface with a minimum number of buttons and external links so that I do not have to teach the user how to use the app beforehand.

One other problem I am going to have to tackle is going to be how I organise the data in the database. Theoretically, I do not need to calculate the actual distance between the bus stops as I am going to have the bus timetables as a reference of how long the busses take to get between bus stops. I am however going to need create some sort of order to connect the bus stops so that I can actually create journeys. I plan to do this by defining each connection between stops as an edge with a weight of the time it takes to get from one stop to the other, as this should be uniform. Each edge will have another value associated with it, which will be the time the bus arrives at the first node so that I can select a time range based on the first node in a path. Then in order to find the shortest path between nodes I plan to implement dijkstra’s algorithm.

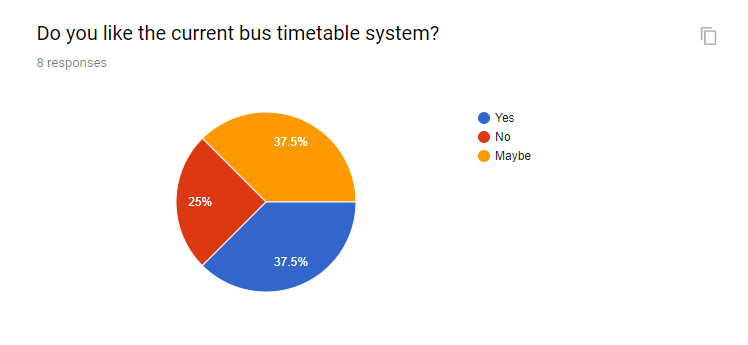
**Dijkstra’s Algorithm**

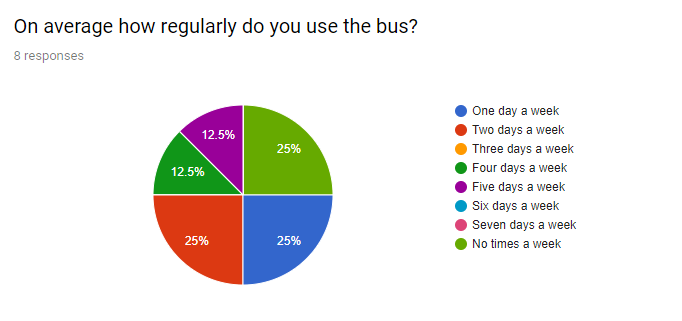


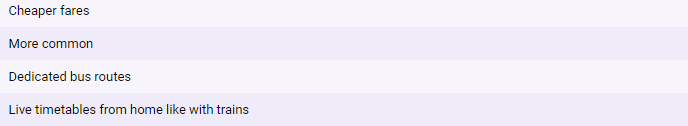
**Survey**<https://docs.google.com/forms/d/1H6H_xFWHw7GR6orgCDeaRuRkNQ84psBmXeIS4X2_Ds/edit>

**Initial data**

When I started gathering data these were my initial results from my poll.



From this data, I learned that the vast majority of people take the bus 0-2 days a week, which makes me less likely to recommend people buy weekly tickets, as most people do not take it very often.

This revelation made me consider having a questionnaire as a user enters the app for the first time as to configure the recommendation bus tickets based on how often they take the bus and whether they are a student or a person over 80.

However, this would mean that I would have to have a separate webserver to allow people to make and sign into their accounts. As this would take a lot more time, I am only going to create it as a feature if I have enough time, as it is not a priority.

**Client**My client is my grandma who often has issues planning bus travel in the local area due to small text and the lack of pricing information on the timetable. She is under the age where she gets a free bus pass but over the age where she is able to use the internet effectively. She also has sight problems, which gives her trouble reading the physical bus timetables. I am able to have weekly meetings with her to discuss the layout and possible improvements to help her and her friends organise their travel.

**How will it be built?**

I will build my program using python and convert it to swift so that it can be used as an app.

**Research into how to make an app**

**IOS Application**

I am planning to use rubicon-objc to allow my python code to work for and iOS app as by definition you can only make iOS apps using objective-c or swift. This would mean that I would have to use the interface created by BeeWare ([*https://github.com/beeware/rubicon-objc*](https://github.com/beeware/rubicon-objc)). They describe it simply as a bridge between python and objective-c. Once set it properly it gives the user to any method or any class in any library, in the entire macOS or iOS ecosystem. If you can invoke something in Objective-C, you can invoke it in Python. This is done by loading the library with ctypes, registering the classes you want to use, and then using those classes as if they were written in Python.

Furthermore, I plan to learn enough swift to convert the python code myself. This is hard to do and will take a lot more time but also reduces the risk of the software incorrectly translating my code or any other potential problems such as how the code works being critically altered. It would also allow me to understand my translated code as well as making alterations significantly easier. The point of this is so that I can proof read the translation myself one it has been completed.

Then I will use Xcode so that it can easily be transferred to App Store Connect which will streamline the process as apple expect programs to be sent to them in this format as it is the coding platform they provide for all users to use. Then all I’ll have to do is submit the code for review in the apple app store and wait for a response.

**Learning swift**

Yes.

**Android Application**

In order to program and application for android, I am going to need to learn kotlin. I have chosen kotlin as it is much easier to learn than the alternative, which is java. This means I will have more time to work on the actual program than having to convert it to java.

Once I have learned kotlin I can use android studio to create my code in kotlin and once completed easily turn it into a mobile application. This is done by submitting the code in a zipped folder to the android store so that they can look over it and made sure it is under their guidelines. Once they have gone over it, it will go onto the app store and be downloadable.

**Learning Kotlin**

I will use kotlin koans *(*[*https://play.kotlinlang.org/koans/overview*](https://play.kotlinlang.org/koans/overview)*)* to learn how to code in kotlin

***Update once learned kotlin***

**Current systems**

**Arriva journey planner:**

The problem with this system is that it only includes Arriva bus services for planning bus journeys and therefore cannot provide information about other, possibly cheaper, option and you may not be taking the most cost effective or time efficient options.

In addition, it does not have a mobile application and therefore is useless unless you have an internet connection. Furthermore, having to connect to a website often takes up more mobile data than having the information already downloaded onto your phone which makes my application more useful for someone who either can’t afford excessive mobile data or someone who lives in a more remote area.

Another issue with the current model links to my first point wherein due to Arriva only displaying their own bus services it makes their software useless to places where Arriva doesn’t have busses which is a large proportion of the UK as Arriva tend to operate in areas around the middle of UK as they don’t have any services in the south or north.

**Transport for London**

The TfL is a travel app for all transport in London. It is used in order to reduce traffic and consequential travel time in the capital city of the UK. This means that is suffers from similar restrictions to the Arriva app wherein it only includes bus services in London and is therefore useless to anyone outside of that area.

Furthermore, the travel in London is a much simpler system in terms of busses as there is only 1 Bus Company that has a few busses constantly doing the same routes at mostly regular intervals. This means that their software is practically self-written as London already has a set system that is not going to change as well as extremely limited options when it comes to bus travel. London’s travel system has been designed specifically to avoid confusion and maximise movement throughout the city. Other areas however do not have such luxuries and as a result, their bus systems are a lot more complicated. This with the added problem of overlapping routes and traffic means that especially in Guildford you end up with busses coming at seemingly random intervals. That causes confusion for the average user who does not regularly use the busses.

**Train line**

Train line is a train-based ticket booking system that allows you to buy and plan any train ticket from any country between any two locations in the UK and some outside of the UK. It is also great for planning a journey as it allows you to sort options by time taken and price. However, what is lacks it the ability to display the best price and time as it mainly focusses on pricing information. This is well as the fact that it is for trains are why my product is necessary.

**Database layout**

My plan is to define each Bus as a different Bus\_ID in a table called tblBus. Then I will create a second table called tblStop which has a Stop ID and a sun table within StopID that contains an array of composite keys of 1 Time and 1 Bus ID

RouteD

**tblStop**

StopID

[Time: BusID, …]

**Tbl/Route**

RouteLength

**tblTimes**

Bus#

**\*Pricing information yet to come**

**Performance criteria**

|  |  |  |
| --- | --- | --- |
| **Order of importance** | **Criteria** | **Criteria reached. (y/n)** |
| 1. | Is the system able to calculate the shortest route, cheapest route and the best of both for any route between a given area? |  |
| 1.1 | Is the system able to interact with a database? |  |
| 1.2 | Is the system able to select a specific type of data based on the time restrictions? |  |
| 1.3 | Is the system able to separate between the pricing, time and ID information to apply different processes and then regroup them afterwards? |  |
| 1.4 | Is the system able to efficiently sort a group of values and/or recognise the meaning of the values of ordered numbers and/or their order given values? |  |
| 1.5 | Is the system able to reach 3 finalised values based on the cheapest, quickest and best of both of said values? |  |
| 2. | Is the system expandable and able to deal with a larger amount of data? |  |
| 2.1 | Is the database able to expand each of its tables in order to add or remove values over time? |  |
| 2.2 | Does the program work with a larger amount of data? |  |
| 2.3 | Is the program able to re-order the database every time a new value is added or one is removed in order to stay accurate? |  |
| 3. | Are the mobile applications functional on both apple and android phones? |  |
| 3.1 | Is the system capable of being accessed and used by mobile users? |  |
| 3.2 | Has the code been translated properly between the different programming languages? |  |
| 4. | Does the system have a nice user interface which can be used easily by someone with little understanding of technology? |  |
| 4.1 | Has it been user tested with people from different technology backgrounds? |  |
| 4.20 | Is it appealing to look at? |  |
| 5. | Is the system able to self-update based on changes in bus schedules over time? |  |
| 5.1 | Is there a functional web-scrapper which is able to fetch data from the different bus websites and updates as their timetables update? |  |
| 6. | Is the system able to adapt based on the requirements of the end user at any given time? |  |
| 6.1 | Is there a questionnaire to acquire the preferences of the user? |  |