# Introduction

Background

The Tourist Information Centre in Cambridge provides information about local accommodation, public transport, conference services, events, UK holiday information, walking tours of Cambridge, maps & guidebooks and much more. While they have appropriate website and software for local events and information they do not have a computerised version for information about public transport. Currently the Tourist Information Centre uses printed timetables provided by Stagecoach to store information about buses. In the past this system was sufficient as Cambridge had very few buses and it was easy find a way around. Over time Cambridge has grown and expanded and now there are over 30 buses coming from and to Cambridge. A few problems have arisen with the current system and Joe Johnson, who works at the information centre, has asked me to make a computerised solution to this problem.

# Investigation

Summary of the current system based on a 1st interview

The problems with the current system lie in finding out the optimal bus journey between bus stops. The tourist office in Cambridge uses printed timetables of buses provided by Stagecoach in Cambridge for the tourists to find how to get from place to place. This requires a lot of time as people have to look up a table. In most cases the bus journey involves at least one bus change. It is not easy to spot which bus stops share bus stops and where it is best to get on and/or off. The tourists often ask for help at the desk, unless the worker has memorised all the timetables the workers usually result to looking at the printed timetables themselves. This takes up a lot of time which could otherwise be spent answering other questions. The printed timetables are often given to tourists for their benefit but this is often unnecessary as only specific bus journeys are required. This system requires a lot of printing and is expensive. The printed timetables only show the name of the bus stops and not the address and although the bus stop name often corresponds to a nearby street name or facility, it is difficult to locate its actual location.

The printed timetables are purchased from by Stagecoach Cambridge for a reasonably small financial cost. The timetables are printed using the information from their database which gets updated at least once a year, so Tourist Information Centre has to request a reprint of their timetables to be kept up to date.

Data flow of the current system

Printed timetables stacks

P1

Bus journey

The worker from the tourist office/ the tourist looks at the printed timetables

1

Timetable Printing

The timetables are printed using data provided by Stagecoach’s database

2

Destination and source bus stop

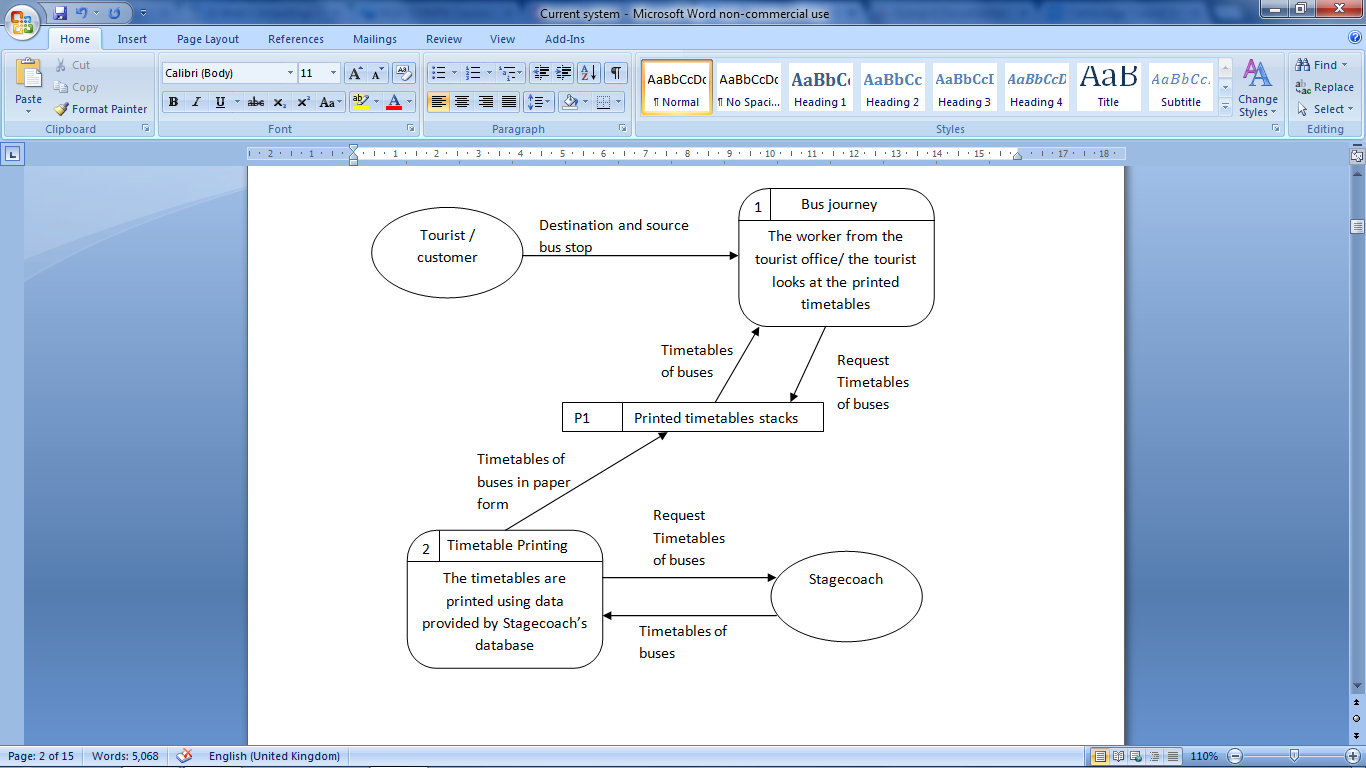
Request Timetables of buses

Timetables of buses

Timetables of buses in paper form

Request Timetables of buses

Timetables of buses

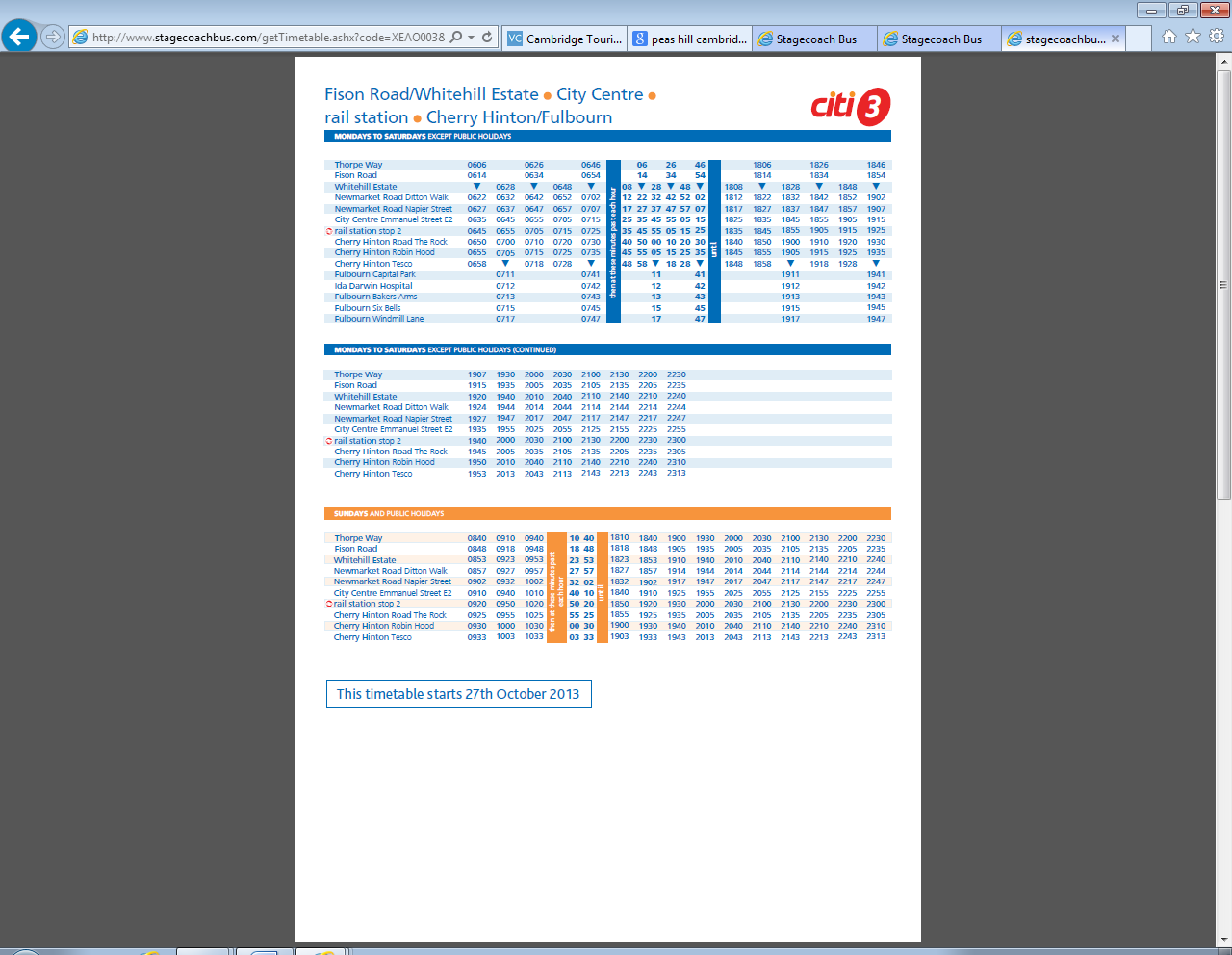


Problems with the current system

Here are the gathered problems with the current system:

* Printed timetables – needed for every bus coming from and to Cambridge, given to each tourist if they would like a copy of a timetable for themselves, costs money, uses a lot of paper and ink, hard to search through the timetable, hard to seek bus stops, time costly, only key bus stops are include in the timetable, reprinting required each time a timetable changes, only bus routes are given and one has to look through several bus routes to plan an appropriate bus journey, hard to find alternative bus journeys
* Finding the appropriate bus stop – not all bus stops are included in the timetable, no data about the location of the bus stop apart from its name, no address, hard to locate on a map

Input, output, forms and report formats from existing system

Here is an example of a timetable for a bus used by the current system:

# Objectives of the new system:

General Objectives:

1. Easy to follow and learn to use within a few seconds of working with the program
2. Easy to update information about timetables

Specific Objectives:

1. The program should accept the start and destination bus stop, the desired time of departure and date.
2. Upon entering an address of a location, the program should output a list of nearest bus stops to that location on a map.
3. The program should show the shortest bus journey depended on the time and day of the journey.
4. The bus journeys should be calculated based on Stagecoach schedule in Cambridge.
5. The system should be compatible of handling about 1.2Mbytes of data at a time and the bus journeys should be displayed within 2 seconds.
6. At least three bus journeys should be displayed showing the time of start of the journey and the time of arrival at the destination, and number of bus changes, total time of journey.
7. Upon selecting a route more details should be shown including: the time and place of the bus changes, the bus name of the buses, time of the journey on each bus and waiting time between bus changes and the total waiting time.
8. The program should produce a document that could be printed or stored if desired. The information should include: the time and place of the bus changes, the bus name of the buses, time of the journey on each bus and waiting time.
9. When selected, the route will be displayed on a map, showing all bus stops and bus changes.
10. The map should have basic zoom in and out feature and span.

Extension Objectives:

1. When clicked on a bus stop on the map the following information could be displayed: the buses going through the bus stop, geographical location.

# Entity relationship diagram and entity description for new system

Data flow diagram

Update

Requesting an update for the database

2

Destination and source bus stop

Bus stops

Timetable and day category

Request

Bus Routes

Database bus routes

D1

Bus route information

General information about the bus

1

Find bus journey

The tourist office worked looks at the printed timetables

2

Find Bus stop

Find appropriate bus stop given a location

3

A postcode/ address of a location to display nearest bus stops

Bus name

Database bus stops

D3

A copy of the database bus information

Database timetable and day category

D2

Request

Request

Request

Bus stop info

timetable info

Bus stop location

Maps interface

bus info

Entity relation diagram

Timetable

BusRoutes

Busstops

DayCategory

Data Volumes:

**Background**

After doing some preliminary research about the timetables I calculated roughly how much data is needed.

There are 33 stagecoach buses coming to and from Cambridge. Each bus has a ‘to’ and ‘back’ route and each route has about 70 bus stops and at least 2 day categories. About 6 buses are every 10 min, 10 buses are every 20 min, and the rest 17 on average are every 45 min in a day. Roughly a day lasts from 6:00 until 21:00. This means that there are 1330 bus courses every day.

**Data Volumes:**

Buses (33 buses but 66 bus routes):  
- (primary) BusID (unsigned integer value between 1 and 33): 6 bits  
- BusName (variable number of chars (up to 50)): 8 bits for every char: 400 bits  
- Description (variable number of chars (up to 200)): 8 bits for every char : 1600 bits

Total storage of Buses = 130000 bits

Bus stops (rough estimate of 2000 bus stops):  
- (primary) BusstopID(unsigned integer value between 1 and 2000): 11 bits  
-BusstopName(variable number of chars (up to 80)): 8 bits for every char: 640  
-Latitude location (signed integer value ): 50 bits  
-Longitude location (signed integer value ): 50 bits

Total storage of Bus stops = 1500000 bits

Timetable (altogether about 200000 entries):  
- (combined primary)BusID (unsigned integer value between 1 and 33): 6 bits  
-(combined primary)BusCourseID (unsigned integer value between 1 - 99): 7 bits  
-(combined primary)BustopID (unsigned integer value between 1 and 2000): 11 bits  
-Time (in format of HH:MM (HH being from 00 to 24, MM being 00 to 59)): 11 bits  
-DateCategoryID (unsigned integer value between 1 and 7): 3 bits

Total storage of Timetable = 7500000 bits

All together ≈ 8000000bits which means roughly 1 Mbyte of data

# Constraints

Software and Hardware constraints

The devices must have access to the internet.

Time constraints

The final outcome should be completed and ready by 27th February.

Users and Clients

The users are going to tourists that not necessarily understand English the software should be in simple English for foreign tourists to understand

Not included in the computerisation

This solution would require software to manage the database and sync it with the Stagecoach database.

Future computerisation

...

# Possible solutions

***A desktop application***- has to be install on every computer, can only be accessed from the tourist office, could use both a local database or from a server, can be operating system specific in certain programming languages, slightly easier to develop  
  
***A phone application*** - nearly everyone if not all have a phone, but operating system specific android, OS and Windows Phone, assumes every tourist has a smart phone, could use a database but limited processing speed and storage space, a server based database may be required, with this solution printing could be avoided because a phones is portable

*My limitations*: I would need to discover a lot more about making phone apps, I would develop it the apps in C# which would only work on Windows Phones and I do not have access to a Windows Phone, also it cannot be assumes that the tourist have windows phones and access to internet

There this solution is ruled out  
  
***A web-based app*** - can be accessed from anywhere with internet connection, anyone would be able to access the application, cross platform, requires a server based database, accessible from both phone and desktop, calculation would be done on a server

*My limitations*: I need to learn a bit more about making websites in C#, discover how ASP.NET works  
  
Data storage

# *Selected Solution*

I choose to make a web based solution with a database because...

Analysis

# Introduction

### Project Definition

Client: The Cambridge Tourist Information Centre

Contact: Joe Johnson

Peas Hill

Cambridge,

CB2 3AD

### Background

The Tourist Information Centre in Cambridge provides information about local accommodation, public transport, conference services, events, UK holiday information, walking tours of Cambridge, maps & guidebooks and much more. They already have a website and software that run in their office that provide general information about local events and information. However they do not have a computerised version for information about public transport. Currently the Tourist Information Centre uses printed timetables provided by Stagecoach to store information about buses. In the past this system was sufficient as Cambridge had very few buses and it was easy find a way around. Over time Cambridge has grown and expanded and now there are over 30 buses coming from and to Cambridge. A few problems have arisen with the current system and Joe Johnson, who works at the information centre, has asked me to make a computerised solution to this problem.

# Investigation

### Summary of the current system based on interview from 8th Nov 2014

The problems with the current system lie in finding out the optimal bus journey between bus stops. The tourist office in Cambridge uses printed timetables of buses provided by Stagecoach in Cambridge for the tourists to find how to get from place to place. This requires a lot of time as people have to look up a table. In most cases the bus journey involves at least one bus change. It is not easy to spot which bus stops share bus stops and where it is best to get on and/or off. The tourists often ask for help at the desk, unless the worker has memorised all the timetables the workers usually result to looking at the printed timetables themselves. This takes up a lot of time which could otherwise be spent answering other questions. The printed timetables are often given to tourists for their benefit but this is often unnecessary as only specific bus journeys are required. This system requires a lot of printing and is expensive. The printed timetables only show the name of the bus stops and not the address and although the bus stop name often corresponds to a nearby street name or facility, it is difficult to locate its actual location.

The printed timetables are purchased from by Stagecoach Cambridge for a reasonably small financial cost. The timetables are printed using the information from their database which gets updated at least once a year, so Tourist Information Centre has to request a reprint of their timetables to be kept up to date.

### Data Flow Diagram of the current system

Data flow of the current system

Problems with the current system

Here are the gathered problems with the current system:

-          Printed timetables – needed for every bus coming from and to Cambridge, given to each tourist if they would like a copy of a timetable for themselves, costs money, uses a lot of paper and ink, hard to search through the timetable, hard to seek bus stops, time costly, only key bus stops are include in the timetable, reprinting required each time a timetable changes, only bus routes are given and one has to look through several bus routes to plan an appropriate bus journey, hard to find alternative bus journeys

-          Finding the appropriate bus stop – not all bus stops are included in the timetable, no data about the location of the bus stop apart from its name, no address, hard to locate on a map

Input, output, forms and report formats from existing system

Here is an example of a timetable for a bus used by the current system:

# Objectives of the new system:

General Objectives:

1.       Easy to follow and learn to use within a few seconds of working with the program

2.       Easy to update information about timetables

Specific Objectives:

1.       The program should accept the start and destination bus stop, the desired time of departure and date.

2.       Upon entering an address of a location, the program should output a list of nearest bus stops to that location on a map.

3.       The program should show the shortest bus journey depended on the time and day of the journey.

4.       The bus journeys should be calculated based on Stagecoach schedule in Cambridge.

5.       The system should be compatible of handling about 1.2Mbytes of data at a time and the bus journeys should be displayed within 2 seconds.

6.       At least three bus journeys should be displayed showing the time of start of the journey and the time of arrival at the destination, and number of bus changes, total time of journey.

7.       Upon selecting a route more details should be shown including: the time and place of the bus changes, the bus name of the buses, time of the journey on each bus and waiting time between bus changes and the total waiting time.

8.       The program should produce a document that could be printed or stored if desired. The information should include: the time and place of the bus changes, the bus name of the buses, time of the journey on each bus and waiting time.

9.       When selected, the route will be displayed on a map, showing all bus stops and bus changes.

10.   The map should have basic zoom in and out feature and span.

Extension Objectives:

1.       When clicked on a bus stop on the map the following information could be displayed: the buses going through the bus stop, geographical location.

# Entity relationship diagram and entity description for new system

Data flow diagram

Entity relation diagram

|  |  |
| --- | --- |
|  |  |
|  |  |

Data Volumes:

**Background**

After doing some preliminary research about the timetables I calculated roughly how much data is needed.

There are 33 stagecoach buses coming to and from Cambridge. Each bus has a ‘to’ and ‘back’ route and each route has about 70 bus stops and at least 2 day categories. About 6 buses are every 10 min, 10 buses are every 20 min, and the rest 17 on average are every 45 min in a day. Roughly a day lasts from 6:00 until 21:00. This means that there are 1330 bus courses every day.

**Data Volumes:**

Buses (33 buses but 66 bus routes):

- (primary)  BusID (unsigned integer value between 1 and 33): 6 bits

- BusName (variable number of chars (up to 50)): 8 bits for every char: 400 bits

- Description (variable number of chars (up to 200)): 8 bits for every char : 1600 bits

Total storage of Buses = 130000 bits

Bus stops (rough estimate of 2000 bus stops):

- (primary) BusstopID(unsigned  integer value between 1 and 2000): 11 bits

-BusstopName(variable number of chars (up to 80)): 8 bits for every char: 640

-Latitude location (signed  integer value ): 50 bits

-Longitude location (signed integer value ): 50 bits

Total storage of Bus stops = 1500000 bits

Timetable (altogether about 200000 entries):

- (combined primary)BusID (unsigned integer value between 1 and 33): 6 bits

-(combined primary)BusCourseID (unsigned  integer value between 1 - 99): 7 bits

-(combined primary)BustopID (unsigned  integer value between 1 and 2000): 11 bits

-Time (in format of HH:MM (HH being from 00 to 24, MM being 00 to 59)): 11 bits

-DateCategoryID (unsigned  integer value between 1 and 7): 3 bits

Total storage of Timetable = 7500000 bits

All together ≈ 8000000bits which means roughly 1 Mbyte of data

Constraints

Software and Hardware constraints

The devices must have access to the internet.

Time constraints

The final outcome should be completed and ready by 27th February.

Users and Clients

The users are going to tourists that not necessarily understand English the software should be in simple English for foreign tourists to understand

Not included in the computerisation

This solution would require software to manage the database and sync it with the Stagecoach database.

Future computerisation

...

Possible solutions

***A desktop application***- has to be install on every computer, can only be accessed from the tourist office, could use both a local database or from a server, can be operating system specific in certain programming languages, slightly easier to develop

***A phone application*** - nearly everyone if not all have a phone, but operating system specific android, OS and Windows Phone, assumes every tourist has a suitable phone, could use a database but limited processing speed and storage space, a server based database may be required, with this solution printing could be avoided because a phones is portable

*My limitations*: I would need to discover a lot more about making phone apps, I would develop it the apps in C# which would only work on Windows Phones and I do not have access to a Windows Phone, also it cannot be assumes that the tourist have windows phones and access to internet

***A web-based app*** - can be accessed from anywhere with internet connection, anyone would be able to access the application, cross platform, requires a server based database, accessible from both phone and desktop, calculation would be done on a server

*My limitations*: I need to learn a bit more about making websites in C#, discover how ASP.NET works

Introduction

Background

The Tourist Information Centre in Cambridge provides information about local accommodation, public transport, conference services, events, UK holiday information, walking tours of Cambridge, maps & guidebooks and much more. While they have appropriate website and software for local events and information they do not have a computerised version for information about public transport. Currently the Tourist Information Centre uses printed timetables provided by Stagecoach to store information about buses. In the past this system was sufficient as Cambridge had very few buses and it was easy find a way around. Over time Cambridge has grown and expanded and now there are over 30 buses coming from and to Cambridge. A few problems have arisen with the current system and Joe Johnson, who works at the information centre, has asked me to make a computerised solution to this problem.

Investigation

Summary of the current system based on a 1st interview

The problems with the current system lie in finding out the optimal bus journey between bus stops. The tourist office in Cambridge uses printed timetables of buses provided by Stagecoach in Cambridge for the tourists to find how to get from place to place. This requires a lot of time as people have to look up a table. In most cases the bus journey involves at least one bus change. It is not easy to spot which bus stops share bus stops and where it is best to get on and/or off. The tourists often ask for help at the desk, unless the worker has memorised all the timetables the workers usually result to looking at the printed timetables themselves. This takes up a lot of time which could otherwise be spent answering other questions. The printed timetables are often given to tourists for their benefit but this is often unnecessary as only specific bus journeys are required. This system requires a lot of printing and is expensive. The printed timetables only show the name of the bus stops and not the address and although the bus stop name often corresponds to a nearby street name or facility, it is difficult to locate its actual location.

The printed timetables are purchased from by Stagecoach Cambridge for a reasonably small financial cost. The timetables are printed using the information from their database which gets updated at least once a year, so Tourist Information Centre has to request a reprint of their timetables to be kept up to date.

Data flow of the current system

Problems with the current system

Here are the gathered problems with the current system:

-          Printed timetables – needed for every bus coming from and to Cambridge, given to each tourist if they would like a copy of a timetable for themselves, costs money, uses a lot of paper and ink, hard to search through the timetable, hard to seek bus stops, time costly, only key bus stops are include in the timetable, reprinting required each time a timetable changes, only bus routes are given and one has to look through several bus routes to plan an appropriate bus journey, hard to find alternative bus journeys

-          Finding the appropriate bus stop – not all bus stops are included in the timetable, no data about the location of the bus stop apart from its name, no address, hard to locate on a map

Input, output, forms and report formats from existing system

Here is an example of a timetable for a bus used by the current system:

# Objectives of the new system:

General Objectives:

1.       Easy to follow and learn to use within a few seconds of working with the program

2.       Easy to update information about timetables

Specific Objectives:

1.       The program should accept the start and destination bus stop, the desired time of departure and date.

2.       Upon entering an address of a location, the program should output a list of nearest bus stops to that location on a map.

3.       The program should show the shortest bus journey depended on the time and day of the journey.

4.       The bus journeys should be calculated based on Stagecoach schedule in Cambridge.

5.       The system should be compatible of handling about 1.2Mbytes of data at a time and the bus journeys should be displayed within 2 seconds.

6.       At least three bus journeys should be displayed showing the time of start of the journey and the time of arrival at the destination, and number of bus changes, total time of journey.

7.       Upon selecting a route more details should be shown including: the time and place of the bus changes, the bus name of the buses, time of the journey on each bus and waiting time between bus changes and the total waiting time.

8.       The program should produce a document that could be printed or stored if desired. The information should include: the time and place of the bus changes, the bus name of the buses, time of the journey on each bus and waiting time.

9.       When selected, the route will be displayed on a map, showing all bus stops and bus changes.

10.   The map should have basic zoom in and out feature and span.

Extension Objectives:

1.       When clicked on a bus stop on the map the following information could be displayed: the buses going through the bus stop, geographical location.

# Entity relationship diagram and entity description for new system

Data flow diagram

Entity relation diagram

|  |  |
| --- | --- |
|  |  |
|  |  |

Data Volumes:

**Background**

After doing some preliminary research about the timetables I calculated roughly how much data is needed.

There are 33 stagecoach buses coming to and from Cambridge. Each bus has a ‘to’ and ‘back’ route and each route has about 70 bus stops and at least 2 day categories. About 6 buses are every 10 min, 10 buses are every 20 min, and the rest 17 on average are every 45 min in a day. Roughly a day lasts from 6:00 until 21:00. This means that there are 1330 bus courses every day.

**Data Volumes:**

Buses (33 buses but 66 bus routes):

- (primary)  BusID (unsigned integer value between 1 and 33): 6 bits

- BusName (variable number of chars (up to 50)): 8 bits for every char: 400 bits

- Description (variable number of chars (up to 200)): 8 bits for every char : 1600 bits

Total storage of Buses = 130000 bits

Bus stops (rough estimate of 2000 bus stops):

- (primary) BusstopID(unsigned  integer value between 1 and 2000): 11 bits

-BusstopName(variable number of chars (up to 80)): 8 bits for every char: 640

-Latitude location (signed  integer value ): 50 bits

-Longitude location (signed integer value ): 50 bits

Total storage of Bus stops = 1500000 bits

Timetable (altogether about 200000 entries):

- (combined primary)BusID (unsigned integer value between 1 and 33): 6 bits

-(combined primary)BusCourseID (unsigned  integer value between 1 - 99): 7 bits

-(combined primary)BustopID (unsigned  integer value between 1 and 2000): 11 bits

-Time (in format of HH:MM (HH being from 00 to 24, MM being 00 to 59)): 11 bits

-DateCategoryID (unsigned  integer value between 1 and 7): 3 bits

Total storage of Timetable = 7500000 bits

All together ≈ 8000000bits which means roughly 1 Mbyte of data

# Constraints

Software and Hardware constraints

The devices must have access to the internet.

Time constraints

The final outcome should be completed and ready by 27th February.

Users and Clients

The users are going to tourists that not necessarily understand English the software should be in simple English for foreign tourists to understand

Not included in the computerisation

This solution would require software to manage the database and sync it with the Stagecoach database.

Future computerisation

...

# Possible solutions

***A desktop application***- has to be install on every computer, can only be accessed from the tourist office, could use both a local database or from a server, can be operating system specific in certain programming languages, slightly easier to develop

***A phone application*** - nearly everyone if not all have a phone, but operating system specific android, OS and Windows Phone, assumes every tourist has a suitable phone, could use a database but limited processing speed and storage space, a server based database may be required, with this solution printing could be avoided because a phones is portable

*My limitations*: I would need to discover a lot more about making phone apps, I would develop it the apps in C# which would only work on Windows Phones and I do not have access to a Windows Phone, also it cannot be assumes that the tourist have windows phones and access to internet

***A web-based app*** - can be accessed from anywhere with internet connection, anyone would be able to access the application, cross platform, requires a server based database, accessible from both phone and desktop, calculation would be done on a server

*My limitations*: I need to learn a bit more about making websites in C#, discover how ASP.NET works

Data storage

Stagecoach does not provide access to their database but is willing to send copies of their timetables in electronic file, which can be easily pulled into a database. Requires someone to maintain and update the database when they change, software would be helpful in a long run

If a desktop solution was going to be developed with a local storage solution a text file would be possible. This solution of storing information would have a high to seek time as it would involve looking over 70000 records. A database has existing seek algorithms which the seek time to be much lower.

A server based database would require hosting. There a server is required

# *Selected Solution*

I choose to make a web based solution with a database because...

Data storage

Stagecoach does not provide access to their database but is willing to send copies of their timetables in electronic file, which can be easily pulled into a database. Requires someone to maintain and update the database when they change, software would be helpful in a long run

If a desktop solution was going to be developed with a local storage solution a text file would be possible. This solution of storing information would have a high to seek time as it would involve looking over 70000 records. A database has existing seek algorithms which the seek time to be much lower.

A server based database would require hosting. There a server is required

# *Selected Solution*

I choose to make a web based solution with a database because...

Analysis

# Introduction

### Project Definition

Client: The Cambridge Tourist Information Centre

Contact: Joe Johnson

Peas Hill

Cambridge,

CB2 3AD

### Background

The Tourist Information Centre in Cambridge provides information about local accommodation, public transport, conference services, events, UK holiday information, walking tours of Cambridge, maps & guidebooks and much more. They already have a website and software that run in their office that provide general information about local events and information. However they do not have a computerised version for information about public transport. Currently the Tourist Information Centre uses printed timetables provided by Stagecoach to store information about buses. In the past this system was sufficient as Cambridge had very few buses and it was easy find a way around. Over time Cambridge has grown and expanded and now there are over 30 buses coming from and to Cambridge. A few problems have arisen with the current system and Joe Johnson, who works at the information centre, has asked me to make a computerised solution to this problem.

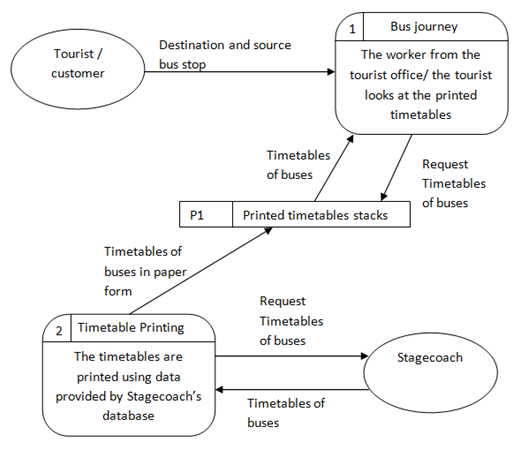
# Investigation

### Summary of the current system based on interview from 8th Nov 2014

The problems with the current system lie in finding out the optimal bus journey between bus stops. The tourist office in Cambridge uses printed timetables of buses provided by Stagecoach in Cambridge for the tourists to find how to get from place to place. This requires a lot of time as people have to look up a table. In most cases the bus journey involves at least one bus change. It is not easy to spot which bus stops share bus stops and where it is best to get on and/or off. The tourists often ask for help at the desk, unless the worker has memorised all the timetables the workers usually result to looking at the printed timetables themselves. This takes up a lot of time which could otherwise be spent answering other questions. The printed timetables are often given to tourists for their benefit but this is often unnecessary as only specific bus journeys are required. This system requires a lot of printing and is expensive. The printed timetables only show the name of the bus stops and not the address and although the bus stop name often corresponds to a nearby street name or facility, it is difficult to locate its actual location.

The printed timetables are purchased from by Stagecoach Cambridge for a reasonably small financial cost. The timetables are printed using the information from their database which gets updated at least once a year, so Tourist Information Centre has to request a reprint of their timetables to be kept up to date.

### Data Flow Diagram of the current system



### Problems with the current system

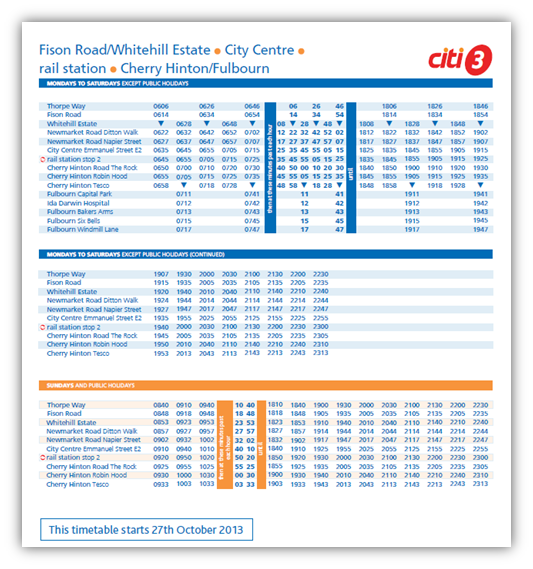
Here are the gathered problems with the current system:

Printed timetables – needed for every bus coming from and to Cambridge, given to each tourist if they would like a copy of a timetable for themselves, costs money, uses a lot of paper and ink, hard to search through the timetable, hard to seek bus stops, time costly, only key bus stops are include in the timetable, reprinting required each time a timetable changes, only bus routes are given and one has to look through several bus routes to plan an appropriate bus journey, hard to find alternative bus journey.

Finding the appropriate bus stop – not all bus stops are included in the timetable, no data about the location of the bus stop apart from its name, no address, hard to locate on a map

Input, output, forms and report formats from existing system

### Example of a timetable for a bus used by the current system:



# Objectives of the new system:

### General Objectives:

1. Easy to follow and learn to use within a few seconds of working with the program
2. Easy to update information about timetables

### Specific Objectives:

1. The program should accept the start and destination bus stop, the desired time of departure and date.
2. Upon entering an address of a location, the program should output a list of nearest bus stops to that location on a map.
3. The program should show the shortest bus journey depended on the time and day of the journey.
4. The bus journeys should be calculated based on Stagecoach schedule in Cambridge.
5. The system should be compatible of handling about 1.2Mbytes of data at a time and the bus journeys should be displayed within 2 seconds.
6. At least three bus journeys should be displayed showing the time of start of the journey and the time of arrival at the destination, and number of bus changes, total time of journey.
7. Upon selecting a route more details should be shown including: the time and place of the bus changes, the bus name of the buses, time of the journey on each bus and waiting time between bus changes and the total waiting time.
8. The program should produce a document that could be printed or stored if desired. The information should include: the time and place of the bus changes, the bus name of the buses, time of the journey on each bus and waiting time.
9. When selected, the route will be displayed on a map, showing all bus stops and bus changes.
10. The map should have basic zoom in and out feature and span.

### Extension Objectives:

1. When clicked on a bus stop on the map the following information could be displayed: the buses going through the bus stop, geographical location.

**Entity relationship diagram and entity description for new system**

Data flow diagram

|  |  |
| --- | --- |
|  |  |
|  |  |

Entity relation diagram

|  |  |
| --- | --- |
|  |  |
|  |  |

Data Volumes:

**Background**

After doing some preliminary research about the timetables I calculated roughly how much data is needed.

There are 33 stagecoach buses coming to and from Cambridge. Each bus has a ‘to’ and ‘back’ route and each route has about 70 bus stops and at least 2 day categories. About 6 buses are every 10 min, 10 buses are every 20 min, and the rest 17 on average are every 45 min in a day. Roughly a day lasts from 6:00 until 21:00. This means that there are 1330 bus courses every day.

**Data Volumes:**

Buses (33 buses but 66 bus routes):

- (primary)  BusID (unsigned integer value between 1 and 33): 6 bits

- BusName (variable number of chars (up to 50)): 8 bits for every char: 400 bits

- Description (variable number of chars (up to 200)): 8 bits for every char : 1600 bits

Total storage of Buses = 130000 bits

Bus stops (rough estimate of 2000 bus stops):

- (primary) BusstopID(unsigned  integer value between 1 and 2000): 11 bits

-BusstopName(variable number of chars (up to 80)): 8 bits for every char: 640

-Latitude location (signed  integer value ): 50 bits

-Longitude location (signed integer value ): 50 bits

Total storage of Bus stops = 1500000 bits

Timetable (altogether about 200000 entries):

- (combined primary)BusID (unsigned integer value between 1 and 33): 6 bits

-(combined primary)BusCourseID (unsigned  integer value between 1 - 99): 7 bits

-(combined primary)BustopID (unsigned  integer value between 1 and 2000): 11 bits

-Time (in format of HH:MM (HH being from 00 to 24, MM being 00 to 59)): 11 bits

-DateCategoryID (unsigned  integer value between 1 and 7): 3 bits

Total storage of Timetable = 7500000 bits

All together ≈ 8000000bits which means roughly 1 Mbyte of data

Constraints

Software and Hardware constraints

The devices must have access to the internet.

Time constraints

The final outcome should be completed and ready by 27th February.

Users and Clients

The users are going to tourists that not necessarily understand English the software should be in simple English for foreign tourists to understand

Not included in the computerisation

This solution would require software to manage the database and sync it with the Stagecoach database.

Future computerisation

...

Possible solutions

***A desktop application***- has to be install on every computer, can only be accessed from the tourist office, could use both a local database or from a server, can be operating system specific in certain programming languages, slightly easier to develop

***A phone application*** - nearly everyone if not all have a phone, but operating system specific android, OS and Windows Phone, assumes every tourist has a suitable phone, could use a database but limited processing speed and storage space, a server based database may be required, with this solution printing could be avoided because a phones is portable

*My limitations*: I would need to discover a lot more about making phone apps, I would develop it the apps in C# which would only work on Windows Phones and I do not have access to a Windows Phone, also it cannot be assumes that the tourist have windows phones and access to internet

***A web-based app*** - can be accessed from anywhere with internet connection, anyone would be able to access the application, cross platform, requires a server based database, accessible from both phone and desktop, calculation would be done on a server

*My limitations*: I need to learn a bit more about making websites in C#, discover how ASP.NET works

Introduction

Background

The Tourist Information Centre in Cambridge provides information about local accommodation, public transport, conference services, events, UK holiday information, walking tours of Cambridge, maps & guidebooks and much more. While they have appropriate website and software for local events and information they do not have a computerised version for information about public transport. Currently the Tourist Information Centre uses printed timetables provided by Stagecoach to store information about buses. In the past this system was sufficient as Cambridge had very few buses and it was easy find a way around. Over time Cambridge has grown and expanded and now there are over 30 buses coming from and to Cambridge. A few problems have arisen with the current system and Joe Johnson, who works at the information centre, has asked me to make a computerised solution to this problem.

Investigation

Summary of the current system based on a 1st interview

The problems with the current system lie in finding out the optimal bus journey between bus stops. The tourist office in Cambridge uses printed timetables of buses provided by Stagecoach in Cambridge for the tourists to find how to get from place to place. This requires a lot of time as people have to look up a table. In most cases the bus journey involves at least one bus change. It is not easy to spot which bus stops share bus stops and where it is best to get on and/or off. The tourists often ask for help at the desk, unless the worker has memorised all the timetables the workers usually result to looking at the printed timetables themselves. This takes up a lot of time which could otherwise be spent answering other questions. The printed timetables are often given to tourists for their benefit but this is often unnecessary as only specific bus journeys are required. This system requires a lot of printing and is expensive. The printed timetables only show the name of the bus stops and not the address and although the bus stop name often corresponds to a nearby street name or facility, it is difficult to locate its actual location.

The printed timetables are purchased from by Stagecoach Cambridge for a reasonably small financial cost. The timetables are printed using the information from their database which gets updated at least once a year, so Tourist Information Centre has to request a reprint of their timetables to be kept up to date.

Data flow of the current system

Problems with the current system

Here are the gathered problems with the current system:

-          Printed timetables – needed for every bus coming from and to Cambridge, given to each tourist if they would like a copy of a timetable for themselves, costs money, uses a lot of paper and ink, hard to search through the timetable, hard to seek bus stops, time costly, only key bus stops are include in the timetable, reprinting required each time a timetable changes, only bus routes are given and one has to look through several bus routes to plan an appropriate bus journey, hard to find alternative bus journeys

-          Finding the appropriate bus stop – not all bus stops are included in the timetable, no data about the location of the bus stop apart from its name, no address, hard to locate on a map

Input, output, forms and report formats from existing system

Here is an example of a timetable for a bus used by the current system:

# Objectives of the new system:

General Objectives:

1.       Easy to follow and learn to use within a few seconds of working with the program

2.       Easy to update information about timetables

Specific Objectives:

1.       The program should accept the start and destination bus stop, the desired time of departure and date.

2.       Upon entering an address of a location, the program should output a list of nearest bus stops to that location on a map.

3.       The program should show the shortest bus journey depended on the time and day of the journey.

4.       The bus journeys should be calculated based on Stagecoach schedule in Cambridge.

5.       The system should be compatible of handling about 1.2Mbytes of data at a time and the bus journeys should be displayed within 2 seconds.

6.       At least three bus journeys should be displayed showing the time of start of the journey and the time of arrival at the destination, and number of bus changes, total time of journey.

7.       Upon selecting a route more details should be shown including: the time and place of the bus changes, the bus name of the buses, time of the journey on each bus and waiting time between bus changes and the total waiting time.

8.       The program should produce a document that could be printed or stored if desired. The information should include: the time and place of the bus changes, the bus name of the buses, time of the journey on each bus and waiting time.

9.       When selected, the route will be displayed on a map, showing all bus stops and bus changes.

10.   The map should have basic zoom in and out feature and span.

Extension Objectives:

1.       When clicked on a bus stop on the map the following information could be displayed: the buses going through the bus stop, geographical location.

# Entity relationship diagram and entity description for new system

Data flow diagram

Entity relation diagram

|  |  |
| --- | --- |
|  |  |
|  |  |

Data Volumes:

**Background**

After doing some preliminary research about the timetables I calculated roughly how much data is needed.

There are 33 stagecoach buses coming to and from Cambridge. Each bus has a ‘to’ and ‘back’ route and each route has about 70 bus stops and at least 2 day categories. About 6 buses are every 10 min, 10 buses are every 20 min, and the rest 17 on average are every 45 min in a day. Roughly a day lasts from 6:00 until 21:00. This means that there are 1330 bus courses every day.

**Data Volumes:**

Buses (33 buses but 66 bus routes):

- (primary)  BusID (unsigned integer value between 1 and 33): 6 bits

- BusName (variable number of chars (up to 50)): 8 bits for every char: 400 bits

- Description (variable number of chars (up to 200)): 8 bits for every char : 1600 bits

Total storage of Buses = 130000 bits

Bus stops (rough estimate of 2000 bus stops):

- (primary) BusstopID(unsigned  integer value between 1 and 2000): 11 bits

-BusstopName(variable number of chars (up to 80)): 8 bits for every char: 640

-Latitude location (signed  integer value ): 50 bits

-Longitude location (signed integer value ): 50 bits

Total storage of Bus stops = 1500000 bits

Timetable (altogether about 200000 entries):

- (combined primary)BusID (unsigned integer value between 1 and 33): 6 bits

-(combined primary)BusCourseID (unsigned  integer value between 1 - 99): 7 bits

-(combined primary)BustopID (unsigned  integer value between 1 and 2000): 11 bits

-Time (in format of HH:MM (HH being from 00 to 24, MM being 00 to 59)): 11 bits

-DateCategoryID (unsigned  integer value between 1 and 7): 3 bits

Total storage of Timetable = 7500000 bits

All together ≈ 8000000bits which means roughly 1 Mbyte of data

# Constraints

Software and Hardware constraints

The devices must have access to the internet.

Time constraints

The final outcome should be completed and ready by 27th February.

Users and Clients

The users are going to tourists that not necessarily understand English the software should be in simple English for foreign tourists to understand

Not included in the computerisation

This solution would require software to manage the database and sync it with the Stagecoach database.

Future computerisation

...

# Possible solutions

***A desktop application***- has to be install on every computer, can only be accessed from the tourist office, could use both a local database or from a server, can be operating system specific in certain programming languages, slightly easier to develop

***A phone application*** - nearly everyone if not all have a phone, but operating system specific android, OS and Windows Phone, assumes every tourist has a suitable phone, could use a database but limited processing speed and storage space, a server based database may be required, with this solution printing could be avoided because a phones is portable

*My limitations*: I would need to discover a lot more about making phone apps, I would develop it the apps in C# which would only work on Windows Phones and I do not have access to a Windows Phone, also it cannot be assumes that the tourist have windows phones and access to internet

***A web-based app*** - can be accessed from anywhere with internet connection, anyone would be able to access the application, cross platform, requires a server based database, accessible from both phone and desktop, calculation would be done on a server

*My limitations*: I need to learn a bit more about making websites in C#, discover how ASP.NET works

Data storage

Stagecoach does not provide access to their database but is willing to send copies of their timetables in electronic file, which can be easily pulled into a database. Requires someone to maintain and update the database when they change, software would be helpful in a long run

If a desktop solution was going to be developed with a local storage solution a text file would be possible. This solution of storing information would have a high to seek time as it would involve looking over 70000 records. A database has existing seek algorithms which the seek time to be much lower.

A server based database would require hosting. There a server is required

# *Selected Solution*

I choose to make a web based solution with a database because...

Data storage

Stagecoach does not provide access to their database but is willing to send copies of their timetables in electronic file, which can be easily pulled into a database. Requires someone to maintain and update the database when they change, software would be helpful in a long run

If a desktop solution was going to be developed with a local storage solution a text file would be possible. This solution of storing information would have a high to seek time as it would involve looking over 70000 records. A database has existing seek algorithms which the seek time to be much lower.

A server based database would require hosting. There a server is required

# *Selected Solution*

I choose to make a web based solution with a database because...