Terms used in this document are described on the final page

Analysis process

The team was given a specification to develop a railway management system. This in itself can be one of endless assumption, so firstly we had to decide what exactly we wanted to do and how we would approach this. As all members had used rail travel significantly, we were in agreement that we considered ourselves domain experts. Experts in that we all had initial thoughts and ideas of what we wanted from railway software and had all dealt with railway software in some shape or form previously. Therefore following from his we engaged in brainstorming as well as the forming of semantic nets, to give initial ideas as to what the domain would involve, who the users and stakeholders would be.

From this start we then began to look at other systems and how they operated, specifically those of the National Rail Network[[1]](#footnote-1) and the Eastcoast Mainline[[2]](#footnote-2). These systems were close to what we were initially thinking of how our system would look and act like as these systems allowed for users to plan, book and find trains for their journeys.

Summary

The client has requested development of a web based system which calculates the most efficient train route dependent on the users starting and destination points, and date and time of travel. It is essential that the website be useable by the majority of the population, and the instructions be clear and easily understood. The design must allow for future changes in the train network as new stations and timetables are implemented.

Environment

The train network we our to develop on has 50 stations, with an expected of 25% increase in the next 10 years. There are 10 crossover stations where a traveller may change to another train to complete their journey. Some lines from the crossover stations intersect, meaning at some times of the day staying on the train is faster and at others transferring is the best option.

The majority of users will either search a website to find the best route, or ask at the station itself. Websites in this sector are very well developed, for example on [www.thetrainline.co.uk](http://www.thetrainline.co.uk) a journey can be searched with less than 10 keystrokes and clicks.

Stakeholder summary

The end user - The person who is travelling. Every aspect of the design has to be tailored to fit their requirements and expectations so they become repeat users.

* Rival websites – They have because the norm for users over the past 5 years and to attract people to our own website we need to exceed the design and functionality of these
* Train companies – A low level requirement is to provide users with a list of train facilities, so links may have to be established to obtain this data
* Train network operaters - Needed to ensure the client has updated lists of timetables and station

Project scope is the process of ensuring that the project deliverables and project boundaries are known to those about to start the project[[3]](#footnote-3). For the purposes of this project the deliverables of the project, essentially that which is understood to be within the scope, are as follows:

In Scope

-The development of a web based system for the client, essentially the purpose of the project

-To ensure the website has the ability to calculate the routes depending on various users’ inputs, such as dates, times and starting journeys.

-Usability for a wide range of consumers with clear instructions.

- Making sure the project is done within the timeframe that the client has set, although there is a possibility of going over the time, correct milestones and priorities will ensure this happens.

These are targets which we want to remain within, however it is important to documents what the boundaries of the project are and what is not expected. These can be identified as follows.

Out of scope

-The website is for train routing only. It is not concerned with train booking or train ticket purchasing. Therefore this acts as a constraint on the software, as users will need to be notified from the outset that they cannot purchase tickets through our system.

- Following on from this, the software is designed for a website only. The team does not have to construct a hardware based version for ticketing booths or ticket machines on various train platforms. Therefore the team doesn’t have to consider the writing of staff training manuals or factor in potential staff and or ticket machine shortages at various stations.

-Furthermore the software has no interaction with the trains themselves, the software team is given a set of times and routes and updates the web interface accordingly. The team doesn’t have to implement any sort of real time adjustment software and doesn’t correspond with potential delays that could happen and the train times that are show at each station.

-The software is for 50 stations initially. Of course the client does seek to add stations over a ten year period, but this will be a gradual process. The system is not being designed for a significantly large number trains, or is it expected to be adaptable to other systems outside of the clients rail network.

Hardware Platforms

The business requirements are very specific regarding the hardware available. The hardware to be used is chosen because that is what is available and the business requires us to use. The website will be hosted on a MySQL database accessible via AJAX (jQuery) which is ran on an Intel PowerEdge 2850 Server. The system has two Intel Xeon processors (dual-core). The CPU has a clock rate of 3.06GHz and the system has 2 GB single rank DDR2 memory (4x512MB). This runs a multikernal operating system which treats the multicore machine as a network of independent cores.

The hardware to be used will meet our requirements as it meets the business expectations and was submitted to a series of performance tests including CPU, Disk and Memory tests which measure allocating and accessing memory speed and efficiency.

Software Platforms

To choose the software we decided to aim for flexibility. The user would not have to download additional plugins for their web browser e.g. Flash and would still be viewable on a smaller mobile device.

The software we plan to use relies upon the functionality of JavaScript and the different JavaScript libraries available. To develop the train timetable the scripting language JavaScript will be used to develop the web pages. Depending upon the size of the webpages, JavaScript will be used either directly as a JavaScript file or embedded in a HTML file. JavaScript and CSS create a Graphical User Interface (GUI) that improves usability while being executed by all modern web browsers. This ensures the timetable program is portable and reduces the recompilation needed. The JavaScript Library, jQuery, can then (using AJAX) call databases to load data from the server without having a browser page refresh.   
  
The IDE used to develop JavaScript is Aptana Studio. This IDE is Eclipse with plugins for web development. This will ensure maximum use of the developer’s time due to familiarity and experience with Eclipse.   
  
The user requirements are not demanding. The JavaScript code if small can be placed directly into a HTML page or a separate JavaScript file. If the users operating system can use a web browser (typically one of IE, Safari, Chrome or Firefox) then the train timetable program will be used.

The software was chosen to make the program as widely available as possible, to be relevant and current to this environment and business requirements and to allow the developers to use previous experience with IDE’s to develop the program as efficiently as possible.

Functional Requirements

The functional requirements have been detailed in the table below. For each requirement it has been assessed if it is a high, medium or low priority for inclusion in the first version of the website.

|  |  |
| --- | --- |
| Timetable of all train times | High |
| Timetable searches for journeys in the future | High |
| Display minimum transfer time between trains at a specific station | High |
| Display the fastest route between two stations and return route | High |
| Display the complete journey time | High |
| Help function for users | High |
| Record all train stations | High |
| Provide a web based interface to the user | High |
| User can print instructions for train journey | Medium |
| Display a selection of routes between two stations | Medium |
| Register user details and personal information | Medium |
| System allows for users with visual impairment/disabilities | Medium |
| Allows customer feedback on system | Low |
| Display station facilities | Low |
| Display train facilities and rules | Low |
| Display direct train routes by station | Low |
| Users can print train map | Low |
| Display a selection of return journeys | Low |
| Display and search for train stations via postcode | Low |
| Save previous searches and journeys | Low |

Non functional requirements

The non functional requirements have been detailed in the table below. All non functional requirements are assessed as high priority.

|  |
| --- |
| System demonstrates a high level of availability |
| System displays functional correctness |
| System displays consistent results |
| The system is stable |
| The system is usable |
| The system is available on the most commonly used browsers (Chrome, IE, Firefox, Safari) |
| The system has complete real-time information |
| The system is private and encrypts customer profiles |
| The system complies with government/local authority regulations |

Considerations

It is expected that the train network we are providing for will operate without unexpected occurances. A summary of the assumptions we have made of the network are as follows

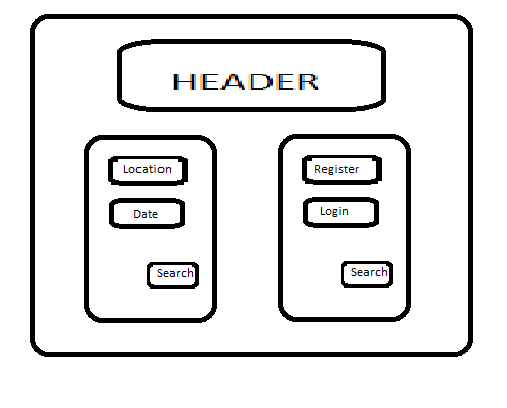
* All lines are bidirectional
* Trains stop at every station
* Trains always take the same time to run between two given stations
* Trains are never late
* Trains always stop for 2 minutes at every station
* There is an infinite number of tracks available
* The website is designed to provide information only

Where possible the design of the website will allow for future development to change the programming to engineer solutions when these assumptions happen. If it is determined during coding that any of the above can be accomodated for in the first version of the website then it will be included.

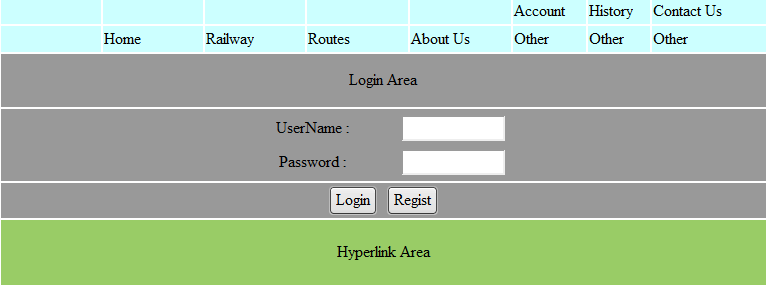
Constraints and Dependencies

* The project is expected to be delivered by May 2013. Every effort will be made to meet all the listed functional and non functional requirements within this timescale though if an unanticipated situation occurs then the plan of the project may need to be adjusted.
* Wide scale testing and feedback by real users will be difficult. A sample audience will test the completed project on different browsers for feedback on the functionality.
* The hardware will not be owned by ourselves and cannot be adjusted if our needs require it to be. However this is not expected to be an issue.

Website design  
Index page:



Log in/out : Each page allows users to log in/out, users' account records their history, management their personal details, information, hobby and so on...

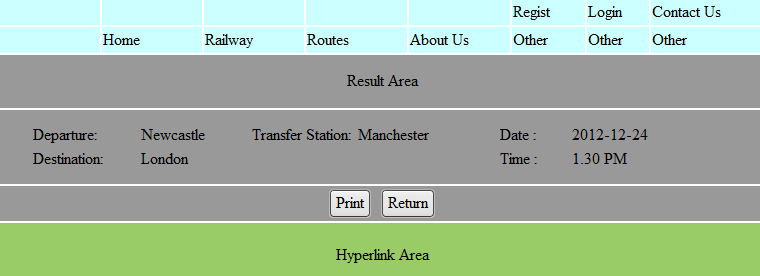


Navigation bars : At the top of page. List some function. Home; Railways; Routes; About us ......

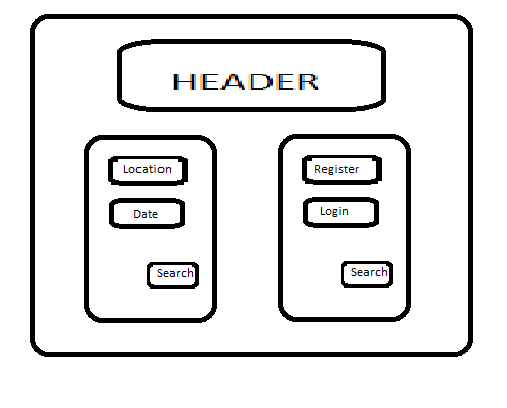
Search module : At the middle of page. The main function -- Search, providing some forms, such as From; To; Time; Return or not; Whether has specified transfer stations .......

Hyperlink area : At the bottom of page. Listing several useful hyperlink. A-Z

Results page:



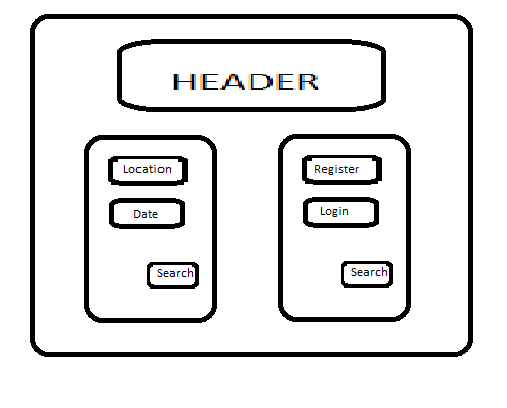
Results module : At the middle of page. Show the results based on the selection of the users, sorting by total price; optimal path; minimum transfer(time/station) ...



Railway page:

……

Railway module : At the middle of page. Display timetable, print it( if could/need ) ...



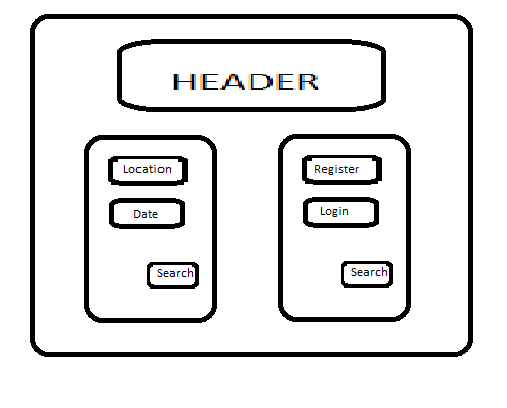
……

Routes page:

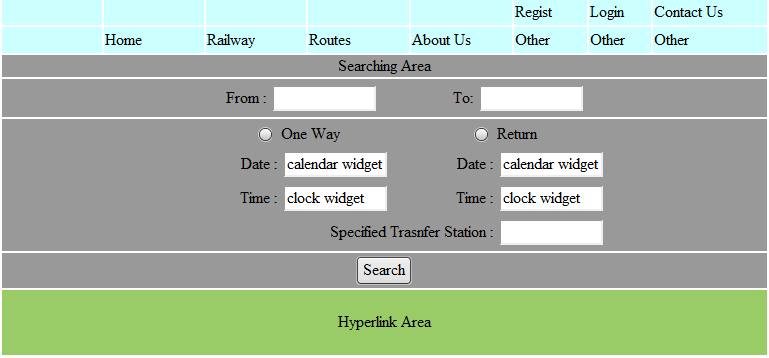
……

Routes module : At the middle of page. Display routes of railways, print it( if could/need) ...

……



Other page:  
List some information and hyperlinks



User Access

Actions described from high level to low level:

Functional:

1) Display the fastest route between two stations and return it

|  |  |
| --- | --- |
| User action  user choose a start route and an end route  user click(or touch) the 'Go' button | System response   1. check train database to find out fastest route display the fastest route |

2) Display the complete journey time

|  |  |
| --- | --- |
| User action  user choose a start route and an end route  user click(or touch) the 'Go' button | S’ystem response   1. check train database to find out journey time 2. display the complete journey time |

3) Timetable of all trains

|  |  |
| --- | --- |
| user action  user click(or touch) the button | system response  check train database for timetable display timetable |

4) Timetable searches for journeys in the future

|  |  |
| --- | --- |
| User action  user choose the start route and end route  user choose a time  user click(or touch) the 'Go' button | system response   1. check train database for timetable 2. display timetable |

5) Display minimum transfer time between trains at a specific station

|  |  |
| --- | --- |
| User Action  user choose a station  user choose to search minimum transfer time  and click 'Go' button | system response   1. check train database 2. display minimum transfer time between trains |

6) Help functions for users

|  |  |
| --- | --- |
| user action  user click(or touch)the 'Help' button  user choose the information he wants for help | system response   1. check database 2. display required information |

7) Record all train stations

|  |  |
| --- | --- |
| user action  staff input all train stations to database | system response  save to database |

8) Provide a web based interface to the user

|  |  |
| --- | --- |
| User Action  user use a web based function | system response   1. execute the web based interface 2. display the output |

9) Display a selection of routes between two stations

|  |  |
| --- | --- |
| User action  choose a start station and end station  user click(or touch) the 'Go' button | System Response   1. check train database 2. display all possible routes |

10) User can print instructions for train journey

|  |  |
| --- | --- |
| User action  user choose on the screen to print instructions | System Response   1. check train database 2. print instructions for journey via a typewriter or on screen |

11) Register user details and personal information

|  |  |
| --- | --- |
| User action  user choose register to input personal details | System Response  save user's details to database |

12) Display direct train routes by station

|  |  |
| --- | --- |
| User action  user choose on the screen | System Response   1. check train database 2. display required routes |

13 )Users can print train map

|  |  |
| --- | --- |
| User action  user choose on the screen to print map | System Response   1. check database 2. print train map via a typewriter |

14) Display a selection of return journeys

|  |  |
| --- | --- |
| User action  user choose a start station and a destination | System Response  check database  display all possible return journeys |

15) Display and search for train stations via postcode

|  |  |
| --- | --- |
| User action  user input a postcode and press 'Go' button | System Response  check database for searching  display the result on the screen |

16) Save previous searches and journeys

|  |  |
| --- | --- |
| User action  user log in | System Response   1. user do searches and journeys 2. save previous searches and journeys to the database automatically |

17) Display station facilities

|  |  |
| --- | --- |
| User action  choose to show station facilities | System Response   1. check database 2. display station facilities |

18) Display train facilities and rules

|  |  |
| --- | --- |
| User action  choose to show train facilities and rules | System Response   1. check database   display train facilities and rules |

19) Allows customer feedback on system

|  |  |
| --- | --- |
| User action  user input feedback information | System Response  save feedback to database |

Definition of Terms

AJAX – A group of technologies to send and receive data from the server asynchronously (in the background) without interfering with the display.

Availability – The percentage of time when the system is operational. For the non-functional requirements over 98% availability is targeted.

Central Processing Unit - The hardware device in a computer that executes all of the instructions from the software.

Clock rate – CPU’s are measured in Hertz to measure performance. The clock rate refers to the speed the electrical voltage changes from low to high and back.

CSS - Cascading Style Sheets. A language used to describe the presentation semantics of a web page or document.

Database – A structured collection of data organized in a way to support accessing this data.

End-user – The person who will be using the site for information and then getting the train

Functional Correctness – Refers to the input-output behaviour of the program e.g. for each input the program produces the correct output.

Graphical User Interface (GUI) – A user interface that allows users to interact with the system using images rather that text commands.

HTML – HyperText Markup Language. The main mark-up language for displaying web pages and other information that can be displayed in a web browser.

IDE –Integrated Development Environment. A software application that provides comprehensive facilities to developers for software development.

JavaScript – A scripting language commonly used as part of the web browser to create enhanced Graphical User Interfaces (GUI).

jQuery – A JavaScript library that simplifies JavaScript programming.

Multi-Platform – Also known as cross platform. The program can be implemented on different computer platforms e.g. operable on Microsoft Windows, Mac OS X and Linux.

MySQL – Open source relational database management system that runs as a server providing access to a number of databases.

Reliability – Ability of the system to perform and maintain its functions in routine circumstances as well as hostile or unexpected circumstances.

Runtime – The period during which a computer program is executing.  
  
Scripting Language – A high level programming language that is interpreted by another program at runtime. Used to add functionality and graphic displays.   
  
Usability – The ease of use of the software application.

1. Eastcoast Mainline website found at http://www.eastcoast.co.uk/ [↑](#footnote-ref-1)
2. National rail website found at http://www.nationalrail.co.uk/ [↑](#footnote-ref-2)
3. ‘http://en.wikipedia.org/wiki/Scope\_%28project\_management%29’ [↑](#footnote-ref-3)