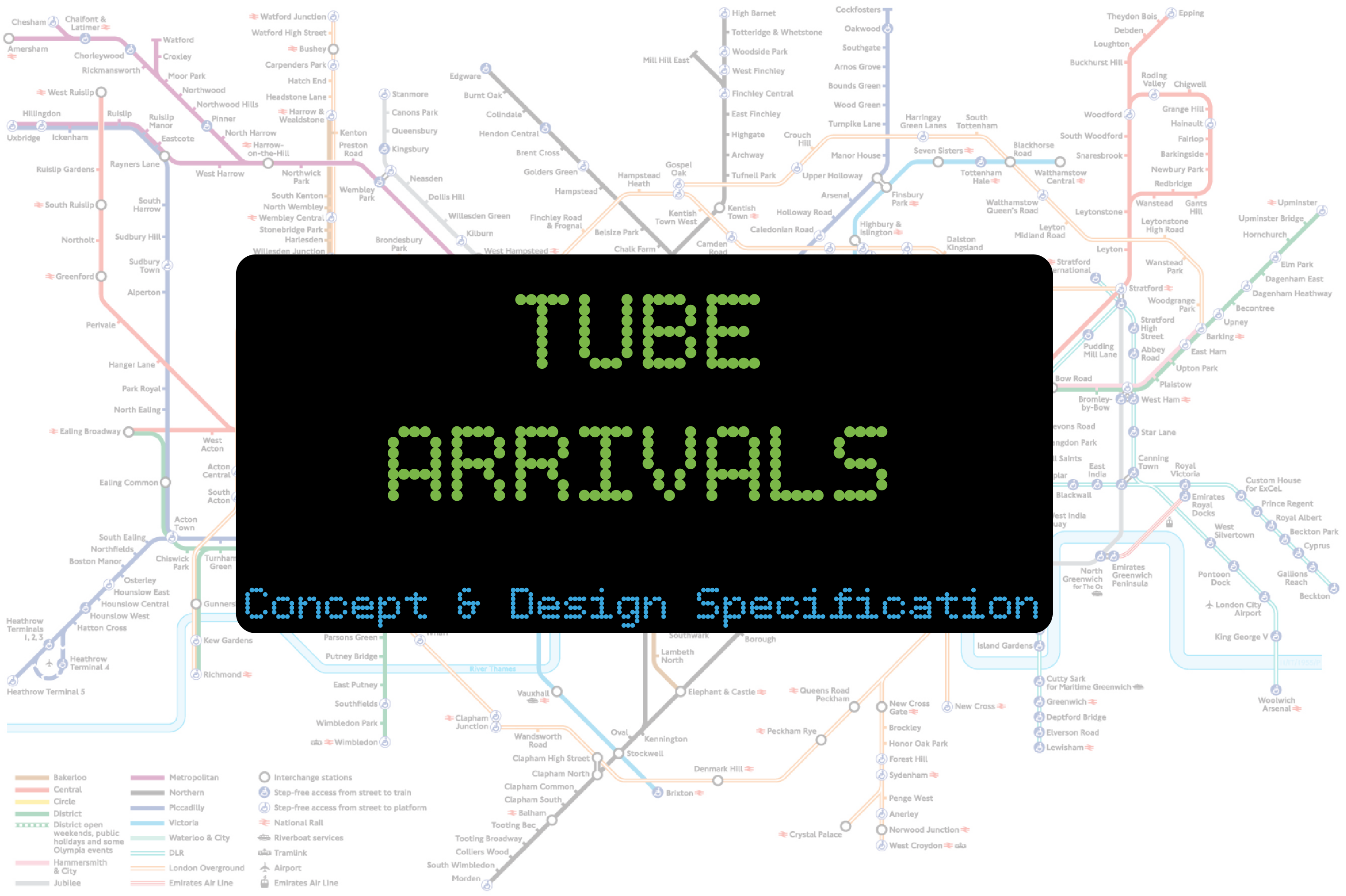


TUBE ARRIVALS

Concept & Design Specification

This specification explains the thinking process which results in an idea transforming from a concept into a working machine.

Author Sukhbinder Singh Nijjar a.k.a Sukh
Year 2018

CONCEPT

Transport for London (TfL) have got a helpful website which shows the status of the various tube lines. It is also possible to see real-time train arrivals for any station on the network, although this is a bit hidden away.

TfL also provide an API known as the **Transport for London Unified API** which allows the retrieval of static and real-time data. One of the web serivices offered gets the predicted arrivals for given line and station.

This application ("Tube Arrivals") shines the spotlight on these real-time train arrivals by providing a simple station search and uses results obtained via the API to responsively present this data styled in the form of a digital display board.

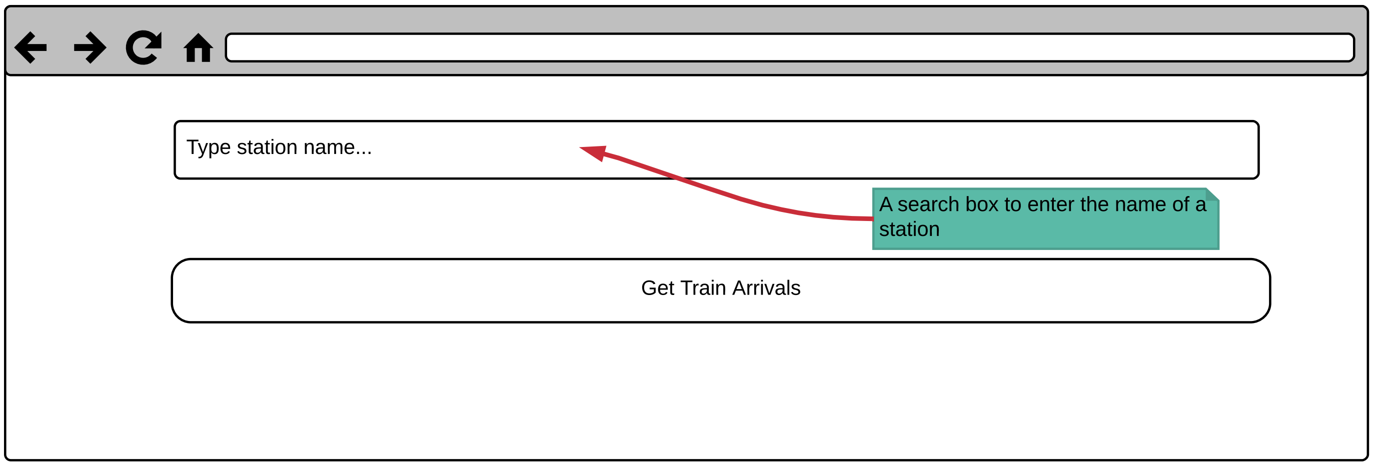
APPROACH

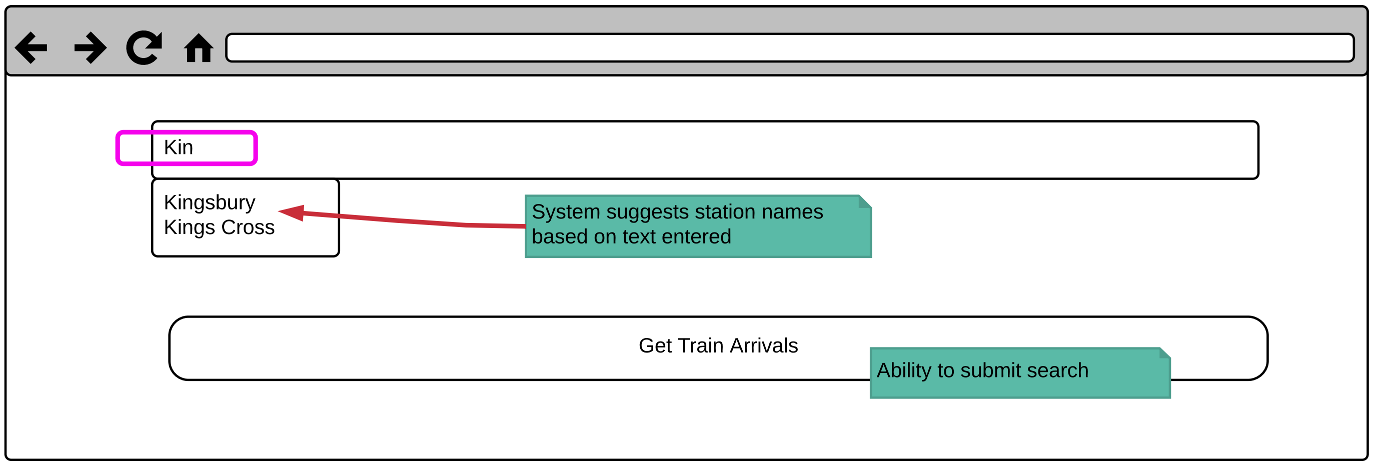
- Create user stories with wireframes to determine **Information Requirements**
- Build skeleton application passing in mocked (hard-coded) arrivals data into views
- Call TfL web services and analyse responses
- Create Python logic to identify and process those elements from the API response that fulfill the identified *information requirements*
- Replace mocked arrivals data with real processed data and pass into views

REQUIREMENTS

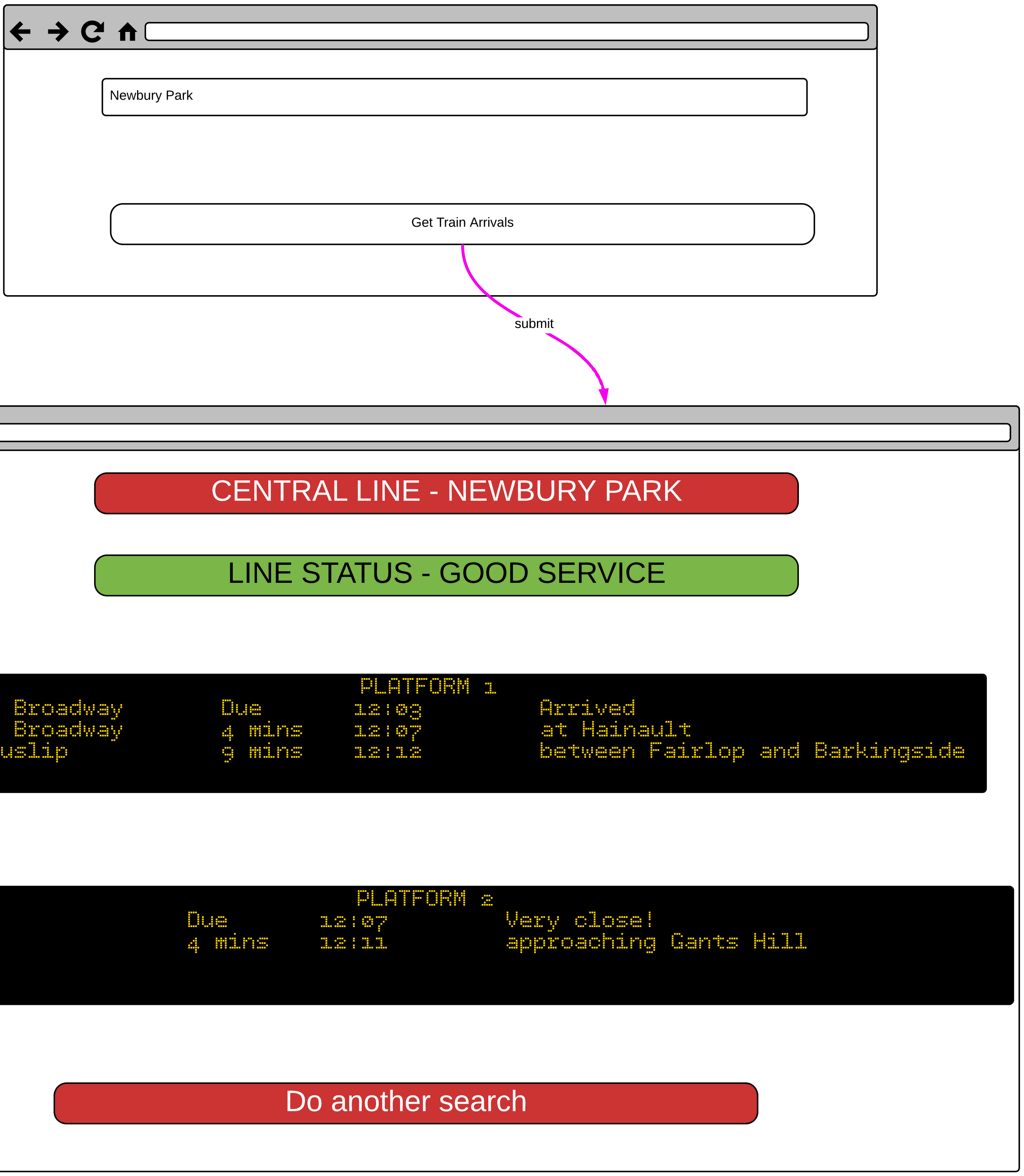
Story	Acceptance Criteria	Required Features/Technical Requirements
As a passenger **I want** to know when the next trains are arriving at my local station **so that** I can work out when I need to leave my location in order to get the train I need and become aware of any issues	**GIVEN** • A passenger wants to know train arrivals for a local station **WHEN** • They search for arrival information **THEN** • They must specify the station name they are interested	• A search box to enter the name of a station • System suggests station names based on text entered • Ability to submit search
	GIVEN • A station is submitted for viewing it's arrivals information **WHEN** • Arrivals information is retrieved **THEN** the following information is shown: • Platform number • Train sequence (e.g. 1,2,3) • Destination station • When due (in minutes) • Expected arrival time • Current location of train	• Receive data submitted and construct URL required for calling TfL web service • Extract and shape data from response returned by TfL API to create arrivals information
	GIVEN • There is no arrivals information available for a station **WHEN** • Arrivals information is retrieved **THEN** • A message is displayed informing no arrivals information is available and line status is displayed	

WIREFRAMES - STATION SEARCH



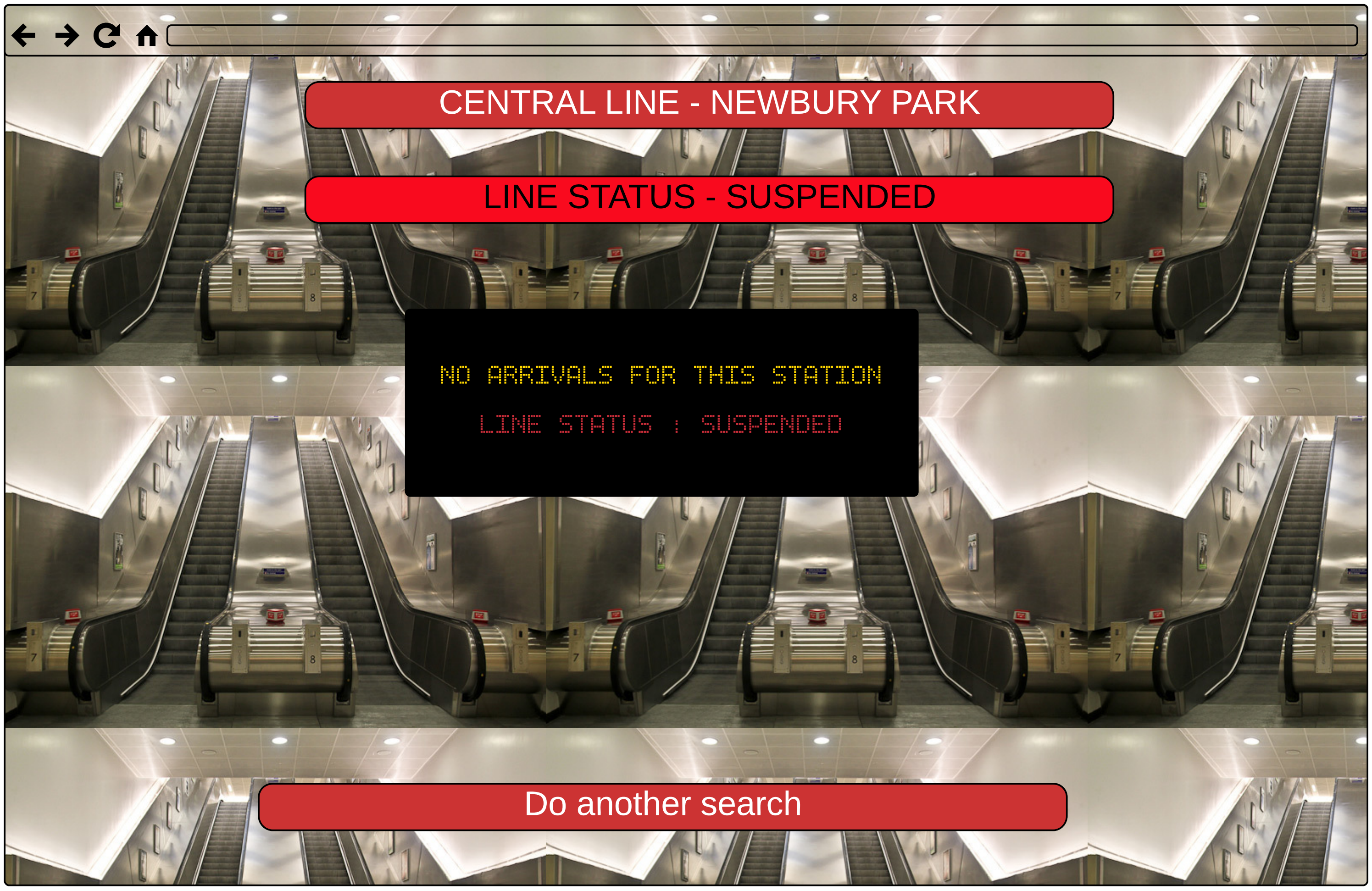


WIREFRAMES - SHOW ARRIVALS


Newbury Park
Get Train Arrivals
submit
CENTRAL LINE - NEWBURY PARK
LINE STATUS - GOOD SERVICE
PLATFORM 1
1 Ealing Broadway Due 12:03 Arrived
2 Ealing Broadway 4 mins 12:07 at Hainault
3 West Ruslip 9 mins 12:12 between Fairlop and Barkingside
PLATFORM 2
1 Woodford Due 12:07 Very close!
2 Hainult 4 mins 12:11 approaching Gants Hill
Do another search

WIREFRAMES - NO ARRIVALS FOUND

Design view when no arrivals information found plus exploring ideas for background imagery



CENTRAL LINE - NEWBURY PARK

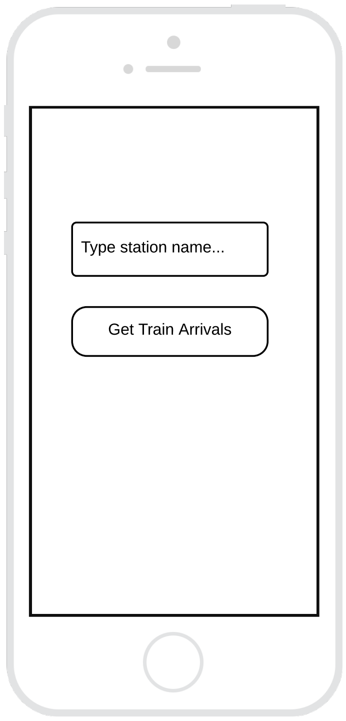
LINE STATUS - SUSPENDED

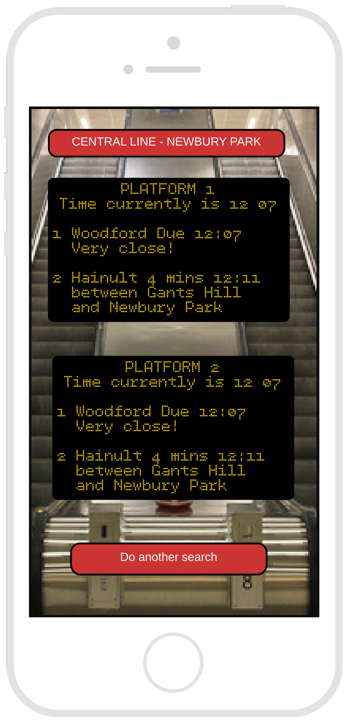
NO ARRIVALS FOR THIS STATION

LINE STATUS : SUSPENDED

WIREFRAMES - RESPONSIVE

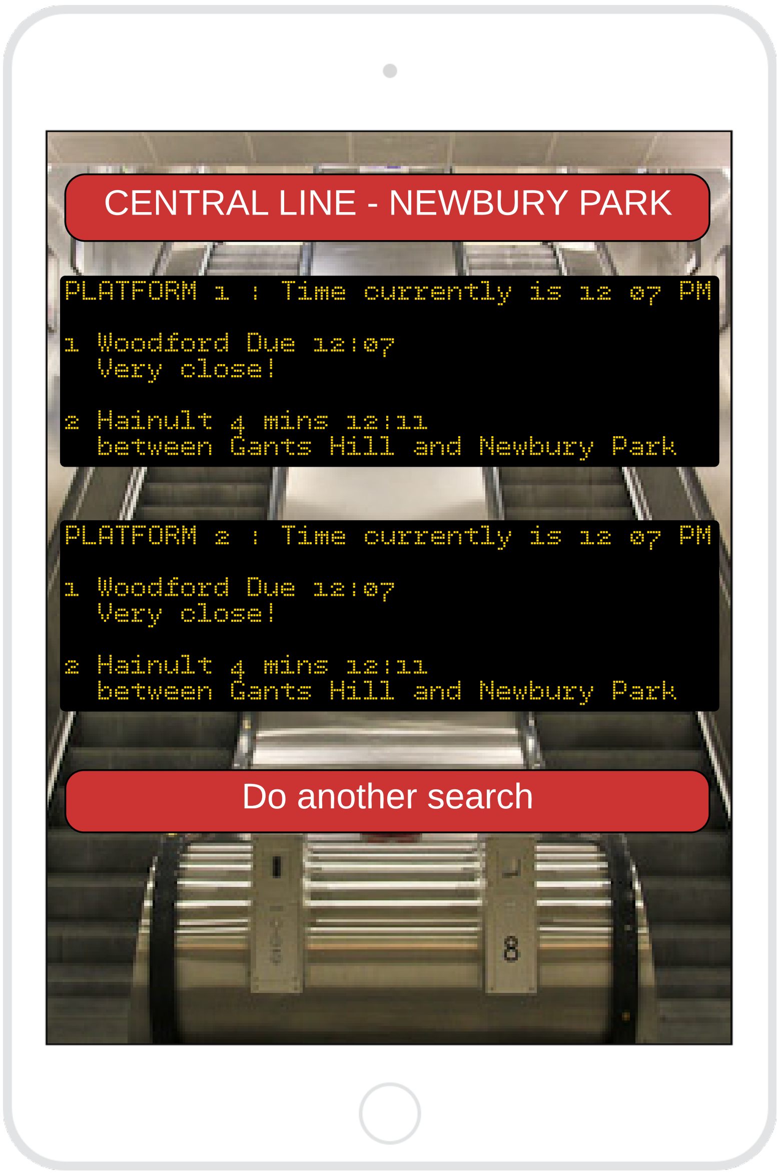
This is some analysis I did to explore how to render the site responsively ; I used Bootstrap 4 for this aspect





```
PLATFORM 1
Time currently is 12 07

1 Woodford Due 12:07
  Very close!

2 Hainult 4 mins 12:11
  between Gants Hill
  and Newbury Park
```

```
PLATFORM 2
Time currently is 12 07

1 Woodford Due 12:07
  Very close!

2 Hainult 4 mins 12:11
  between Gants Hill
  and Newbury Park
```



```
PLATFORM 1 : Time currently is 12 07 PM

1 Woodford Due 12:07
  Very close!

2 Hainult 4 mins 12:11
  between Gants Hill and Newbury Park
```

```
PLATFORM 2 : Time currently is 12 07 PM

1 Woodford Due 12:07
  Very close!

2 Hainult 4 mins 12:11
  between Gants Hill and Newbury Park
```

DESIGN

UI :
Responsive front end - the application must render properly across various device types
Look and feel -

- the site will have a background image familiar to tube users
- the arrivals presentation will be inspired by the digital display boards found at tube stations
- font to replicate dot matrix display is Codystar
- font to display the station name and line status is the elegant Julius Sans One
- lines will be represented by their familar TfL colours (e.g. the central line is red)
- line status will be represented by the familar Red-Amber-Green scheme (Red = severe delays, Amber = minor delays, Green = good service)

DATA - Sourced from the TfL Unified API
Real Time Live
Arrivals for a specific station on a specific line : https://api.tfl.gov.uk/Line**/*line_id***/Arrivals/***station_id***
(e.g. arrivals at Liverpool Street on the Circle line) : https://api.tfl.gov.uk/Line/***circle***/Arrivals/***940GZZLULVT***
The arrivals view needs to refresh every 60 seconds

Status of a specific line : https://api.tfl.gov.uk/Line/***line_id***/Status?detail=true
(e.g. current status (good service, minor delays etc) of the circle line) : https://api.tfl.gov.uk/Line/***circle***/Status?detail=true

Distruption information for a specific line : https://api.tfl.gov.uk/Line/***line_id***/Disruption
(e.g. service can be distrupted for various reasons (engineering works etc) : https://api.tfl.gov.uk/Line/***circle***/Disruption

Static
The list of stations made available on the station search view must be generated once and stored locally to act as the source data for the search input field. Calling the API each time the page loaded is time consuming and wasteful.
Stations on a specific line : https://api.tfl.gov.uk/Line/***line_id***//StopPoints
(e.g. get the list of stations on the circle line) : https://api.tfl.gov.uk/Line/***circle***/StopPoints
This webservice is invoked once for each line to build up the total list of stations on the tube network. The station list is generated as a one-off activity and once generated is stored as a local json file.

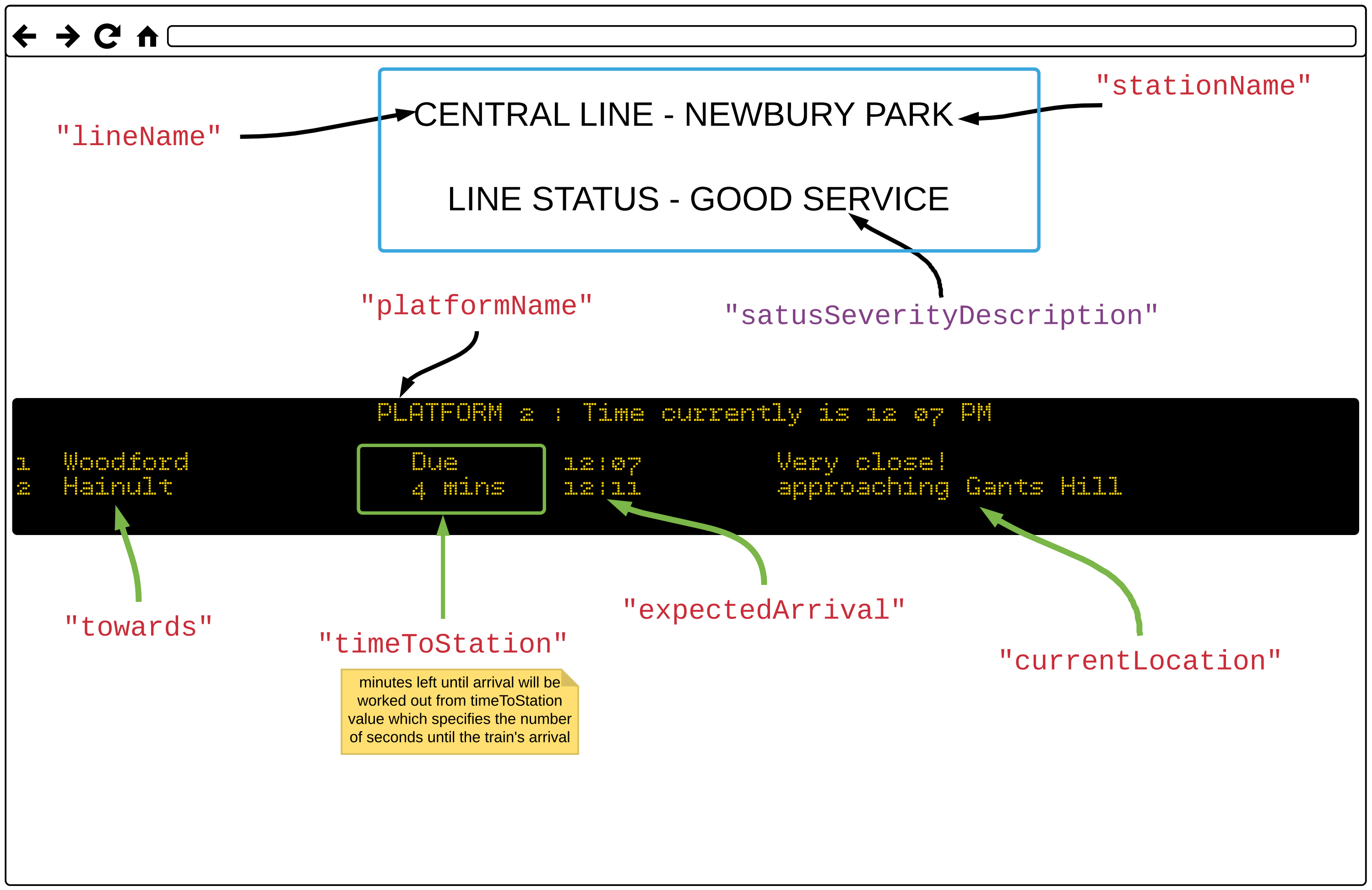
API ANALYSIS

The API responses are verbose and required careful analysis to extract the specific elements relevant to the requirements.
Some things learnt:

- The API does not return arrivals data in time order - the application has responsibility to sort the data according to time requirements
- The list of arrivals can be very long - the application has responsibility for displaying the next 10 arrivals only
- Arrivals data is not split by platform - the application has responsibility for splitting arrivals information by platform
- The API does not always return arrivals data for a station even if there is a good service, for example Becontree never seems to have live arrivals information! There needs to be a check for empty results to deal with it elegantly and not with a crash!

UI - API SOURCE DATA MAPPING

This is some analysis I did to help me understand which API response elements I would consume to feed the UI



"lineName"
CENTRAL LINE - NEWBURY PARK
"stationName"
LINE STATUS - GOOD SERVICE
"platformName"
"satusSeverityDescription"
PLATFORM 2 : Time currently is 12 07 PM
1 Woodford Due 12:07 Very close!
2 Hainult 4 mins 12:11 approaching Gants Hill
"towards"
"timeToStation"
"expectedArrival"
"currentLocation"
minutes left until arrival will be worked out from timeToStation value which specifies the number of seconds until the train's arrival