National College of Ireland

BSc Hons in Computing

2015/2016

Andrzej Stozek

12118397

andrzej.stozek@student.ncirl.ie

Software Project

Technical Report

Multi-platform

Hybrid mobile application

“Dublin Commuting - Live Routing”

Supervisor:

Dr Ralf Bierig

(Mikhail Timofeev)

Table of Contents

1. Executive Summary

This document is a Technical Report of the software project which subject is multi-platform hybrid mobile application called “Dublin Commuting - Live Routing”.

The main focus in this document is on the description of the project problem, requirements specification, analysis and design of the proposed solution as well as description of the implementation, testing and evaluation.

The project subject is the multi-platform hybrid mobile application called “Dublin Commuting - Live Routing”.

The problem which the application will solve is to keep the user, who is commuting in the Greater Dublin Area using public transport (Dublin Bus, LUAS, Dart, Commuter Train, Bus Eireann, Dublin Bikes), updated with live information on the estimated travel times for the route from user’s current location (or other selected location) to selected destination point. The user will be able to take informed decision if the selected route of transport is the best currently available option.

The application being developed is a mobile application created using PhoneGap which will let the system to be converted into mobile application for multiple mobile platforms (Android, iOS, Windows Phone).

Technologies used to implement are web technologies:

- HTML5,
- CSS3,
- JavaScript,
- Bootstrap,
- JavaScript frameworks: jQuery, jQueryMobile.

Application uses external APIs:

- Real time Passenger information (RTPI – Dublin Bus, LUAS, Bus Eireann),
- Irish Rail Real Time API (Dart, Commuter Trains),
- Dublin Bikes API (Dublin Bikes),
- Google Map API,
- Google Directions API,
- Google Places API.

The intended users of the application are the commuters who use public transport within Greater Dublin Area.

2. Introduction

2.1. Background

Greater Dublin Area offers many public transport services (bus – Dublin Bus, LUAS, train – Dart and Commuter trains, bicycle – Dublin Bike) which lead to multiple possible routes the commuter could take to get to the required destination. Also changing traffic conditions might have an impact on the travel times. The mobile application will constantly keep checking for possible routes so the user will be able to take informed decision which route to take, stay on the current route or switch to different one.

There are no existing solutions which would include Dublin Bikes into routing options and also provide automatic updates of live arrival times on the user route.

There are many application which provide information only for single means of transport e.g. only Dublin Bus or Irish Rail. Google maps provides directions functionality for public transport but it doesn’t include Dublin Bikes in routing options. Also there is no live updates with departure/arrival times.

The application will be user friendly and intuitive and will deliver relevant information to the user in the shortest time possible. In few clicks/taps information of all possible route options will be displayed and user will be able to take informed decision on the next step on his/her way to the destination point.

2.2. Aims

The scope of the project is to develop multi-platform hybrid mobile application called “Dublin Commuting - Live Routing”. The application will use RTPI API, Irish Rail API, Dublin Bikes API, Google Directions API, Google Map API and Google Places API.

The constraint on the project is the 11th of May 2016 - final project documentation and code submission.

The product is a mobile application which will be available for multiple mobile platforms – Android, iOS and Windows Phone.

The objective of the project is to develop multi-platform hybrid mobile application which will solve the problem of keeping the user, who is commuting in the Greater Dublin Area using public transport (Dublin Bus, LUAS, Dart, Commuter Train, Bus Eireann, Dublin Bikes), updated with live information on the estimated travel times for the route from user's current location (or other selected location) to selected destination point. The user will be able to take informed decision if the selected route of transport is the best currently available option.

2.3. Technologies

The application will be a hybrid mobile application created using PhoneGap which is the framework which lets the system developed using web technologies to be converted into mobile application for multiple mobile platforms (Android, iOS, Windows Phone)

Technologies to be used:

- HTML5,
- CSS3,
- JavaScript,
- jQuery,
- jQueryMobile,
- Bootstrap,
- Some JavaScript frameworks for the backend development if necessary: Angular.js, Ember.js, Backbone.js, Node.js (not used in the current implementation).

The application uses multiple external APIs in order to provide all relevant information to the user.

External API used are:

- Real time Passenger information (RTPI – Dublin Bus, LUAS, Bus Eireann)
- Irish Rail Real Time API (Dart, Commuter Trains)
- Dublin Bikes API (Dublin Bikes)
- Google Directions API
- Google Map API
- Google Places API

2.4. Definitions, Acronyms, and Abbreviations

Application – is a computer program designed for a specific task or use

Application Programming Interface - (API) specifies how software components should interact with each other

API – Application programming interface

Global Positioning System - an electronic system that uses satellites to determine the position of a vehicle, person, etc.:

GPS – Global Positioning System

Graphical User Interface - is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators

GUI – Graphical User Interface

Hybrid mobile application – mobile application built using HTML5 and JavaScript, wrapped inside a thin native container that provides access to native platform features

Mashup – is a web or mobile application that uses data from two or more sources to create something new.

Mobile application – is an application designed to be run on smart phones and tablets.

RTPI - Real Time Passenger Information – real time information for Dublin Bus / Bus Eireann / LUAS arrival/departure times, timetables and stops

RTPI – Real Time Passenger Information

REST - Representational state transfer (REST) is an architectural style consisting of a coordinated set of constraints applied to components, connectors, and data elements, within a distributed hypermedia system. REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements

3. System

3.1. Requirements specification

3.1.1. Functional requirements

The application will provide the user with the functionality listed below:

1. Search for possible public transport route options from the selected starting point to the selected destination point, with checks for actual currently available arrival times. Public transport included in search will be: Dublin Bus, LUAS, Dart, Commuter Train, Bus Eireann, and Dublin Bikes.
2. Set starting point using Google Map.

2a. Adjust the starting point location by drag and drop of the marker.

3. Set destination point using Google Map.
   3a. Adjust the destination point location by drag and drop of the marker.
4. Set starting point using Device Geolocation.

4a. Adjust the starting Geolocation point location by drag and drop of the marker.

5. Set starting point using address search.
   5a. Adjust the starting from point using address search by drag and drop of the marker.
6. Set destination point using address search.
   6a. Adjust the destination from point using address search by drag and drop of the marker.
7. Browse found route options.
8. Automatically refresh the list of possible route options for selected start and end point every 60 seconds.
9. On Demand refresh the list of possible route options for selected start and end point.
10. View route option details.
11. Automatically refresh details of the selected route option for selected start and end point every 60 seconds.

12. On demand refresh details of the selected route option for selected start and end point every 60seconds.
13. Indicate currently used means of transport ("On the bus", "On the bike", "On the train", "On the LUAS") for the selected route.
14. If means of transport is selected – provide real time details of the arrival time to the final stop, and updates on any delays on the route.
15. If "On the bike" selected keep the user updated with the current situation at the destination bike station – if free bike stands are available, and if not alert the user and provide closest alternative bike station.
16. Automatically save selected destination point to the history list of recent destination points.
17. Automatically save selected start point to the history list of recent start points.
18. Save selected start point to the favourite start points.
19. Save selected destination point to the favourite destination points.
20. Select start point from history.
21. Select destination point from history.
22. Select start point from favourites.
23. Select destination point from favourites.
24. Clear favourites.
25. Clear history.
26. Delete start point from favourites.
27. Delete destination point from favourites.
28. Adjust application settings (MAX_WALK_DISTANCE, don't use Dublin Bikes, Days to keep history).
29. Reset application settings to the default values.

The functional requirements of the system are described in use cases along with description of each.

This section lists the functional requirements with their priority. Functional requirements describe the possible effects of a software system, in other words, *what* the system must accomplish.

Requirement ID	Long Name	Description	Priority	Input	Output
01-FindRoutes	Find Possible Route Options	Search for possible route options from the selected start to the selected end point	1	Start GPS coordinates, End GPS Coordinates	List of available route options displayed on the screen
02-StartFromMap	Get Start Point From Map	Enable selection of the Start point directly from the Google Map by tap on the map, additionally enable adjustment of the selected point by drag and drop of the map marker (requirement 02a)	1	Map Position	GPS coordinates of map position passed to Start point
02a-AdjustStartFromMAp	Adjust Start Point Location on The Map	See 02	1		
03-EndFromMap	Get End Point From Map	Enable selection of the End point directly from the Google Map by tap on the map. Additionally enable adjustment of the selected point by drag and drop of the map marker (requirement 03a)	1	Map Position	GPS coordinates of map position passed to End point

Requirement ID	Long Name	Description	Priority	Input	Output
03a-AdjustEndFromMap	Adjust End Point Location on The Map	See 03	1		
04-StartFromLocation	Get Start Point From Device Location	Acquire GPS start point from device position. Additionally enable adjustment of the GPS location by drag and drop of the map marker (requirement 04a)	1	Device location service enabled	GPS coordinates of device location passed to Start point
04a-AdjustStatrFromLocation	Adjust Strat GPS Location on the Map	See 04	1		
05-StartFromSearch	Get Start Point From Search	Enable start point GPS coordinates to be acquired by address search with aid of the Google Places API. Additionally enable adjustment of the selected point by drag and drop of the map marker (requirement 05a)	1	address of the start point	GPS coordinates from the search passed to Start point

Requirement ID	Long Name	Description	Priority	Input	Output
05a-AdjustStartFromSearch	Adjust Start Point Location on The Search Map	See 05	1		
06-EndFromSearch	Get End Point From Search	Enable and point GPS coordinates to be acquired by address search with aid of the Google Places API. Additionally enable adjustment of the selected point by drag and drop of the map marker (requirement 06a)	1	address of the end point	GPS coordinates from the search passed to End point
06a-AdjustEndFromSearch	Adjust End Point Location on The Search Map	See 06	1		
07-ListRoutes	List Route Options	Enable user to view list of found available route options	1	relevant content from the Find possible route options	relevant content displayed

Requirement ID	Long Name	Description	Priority	Input	Output
08-ViewRoute	View Route Details	Enable user to view details of route option	2	ID of selected route	Details screen for selected route displayed
09-AutoRefreshRoutes	Auto Refresh Route Options	Search for possible route options from the start to end point every 60 seconds	1	Start GPS coordinates, End GPS Coordinates	List of available route options displayed on the screen
10-OndemandRefreshRoutes	On Demand Refresh Route Options	Search for possible route options from the start to end point on user demand	1	Start GPS coordinates, End GPS Coordinates	List of available route options displayed on the screen
11-AutoRefreshSelect	Auto Refresh Selected Route	Search for possible route options from the start to end point every 60 seconds and display details of selected option	2	Start GPS coordinates, End GPS Coordinates	Details screen for selected route displayed

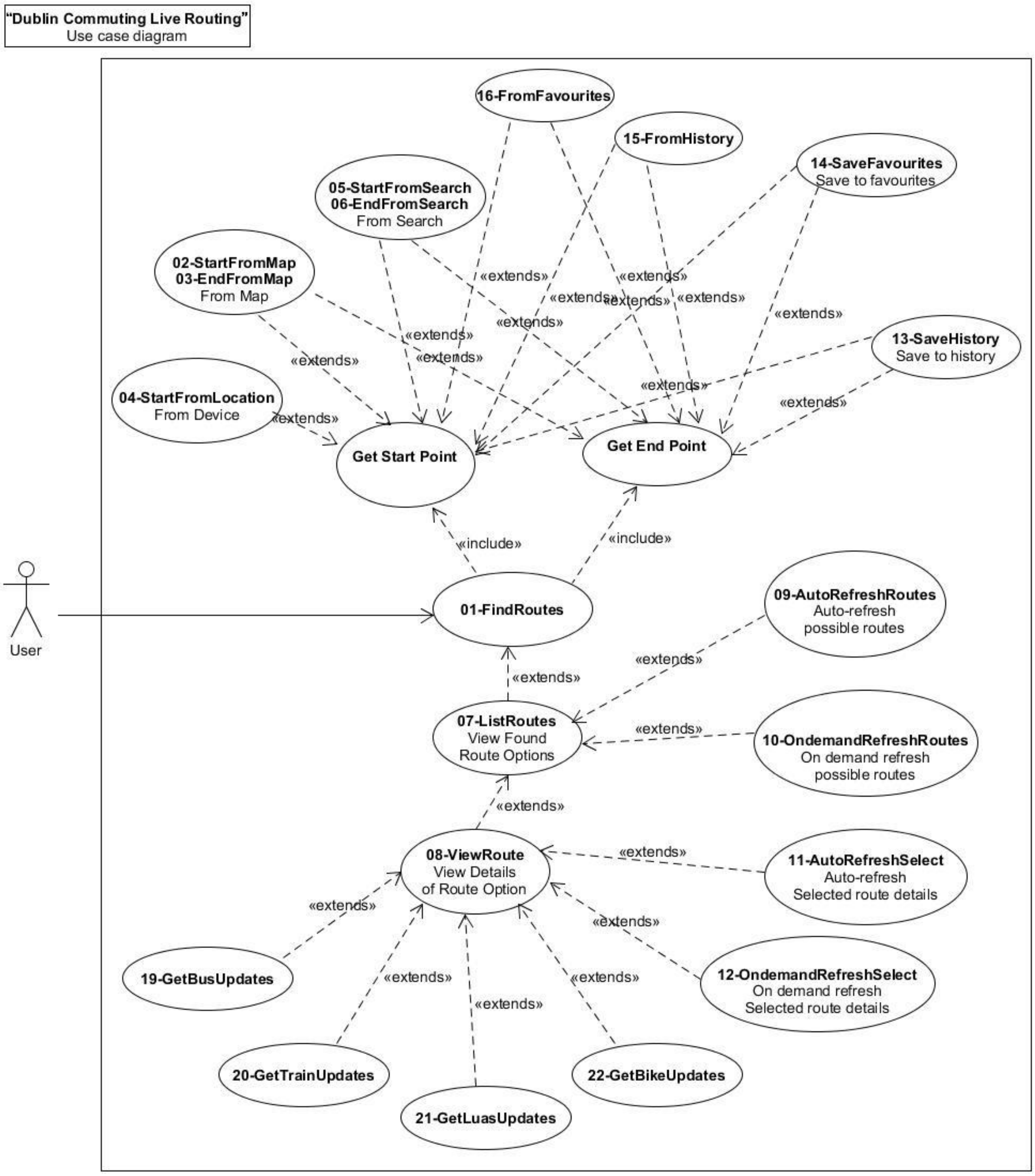
Requirement ID	Long Name	Description	Priority	Input	Output
12-OndemandRefreshSelect	On Demand Refresh selected Route	Search for possible route options from the start to end point on user demand and display details of selected option	2	Start GPS coordinates, End GPS Coordinates	Details screen for selected route displayed
13-SaveHistory	Save history	Save selected start/end point to history	2	Start/End point GPS Coordinates	Add start/end point GPS Coordinates and description to the history
14-SaveFavourites	Save Favourites	Save selected start/end point to favourites	2	Start/End point GPS Coordinates	Add start/end point GPS Coordinates and description to the favourites
15-FromHistory	Select Start/End From History	Select start/end point from history	2	Selected Start/End point ID	GPS coordinates of selected record passed to Start/End point

Requirement ID	Long Name	Description	Priority	Input	Output
16-FromFavourites	Select Start/End From Favourites	Select start/end point from favourites	2	Selected Start/End point ID	GPS coordinates of selected record passed to Start/End point
17-ClearHistory	Clear history	Delete all Start/end points from history	2	-	delete all records from history
18-DeleteFavourites	Delete Favourites	Delete selected start/end points from favourites	2	Selected Start/End point ID	delete selected records from favourites
19-GetBusUpdates	Get Selected Dublin Bus updates	Provide live updates about actual arrival time to the last stop on the route for selected bus number	2	Selected bus number	Display update information
20-GetTrainUpdates	Get Selected Irish Rail Train updates	Provide live updates about actual arrival time to the last stop on the route for selected train	2	Selected Train	Display update information
21-GetLuasUpdates	Get Selected Luas updates	Provide live updates about actual arrival time to the last stop on the route for selected LUAS	2	Selected LUAS	Display update information

Requirement ID	Long Name	Description	Priority	Input	Output
22-GetBikeUpdates	Get Selected Dublin Bikes updates	Provide live updates about available free bike stands at the destination bike station, if no free stands available, search for the other possible bike station and provide information	2	On the bike flag	Display update information
23-DisplayBikestations	Display Bike Stations on the map	Display all Dublin Bikes Bike Station on the map	1		Dublin Bikes stations displayed on the map
24-GetBikeStationUpdates	Get Selected Dublin Bikes station updates	Get and display updated live information of available bikes and available bike stands at selected bike station	1	Bike station number	Display updated information of available bikes and bike stands

3.1.2. Use Case Diagram

The Use Case Diagram provides an overview of all functional requirements.



3.1.3. Requirement 1: Find Possible Route Options

01-FindRoutes	Find Possible Route Options	Search for possible route options from the selected start to the selected end point	1	Start GPS coordinates, End GPS Coordinates	List of available route options displayed on the screen

Identifier: 01-FindRoutes

3.1.3.1. Description & Priority

Priority 1 (highest)

This requirement ("Find Possible Route Options") is the most important requirement of the application. It represents the core functionality of the application and has the highest priority.

This function of the systems provides to the user list of possible route options using public transport (Dublin Bus, Irish Rail, LUAS, Bus Eireann and Dublin Bikes) within Greater Dublin Area, from selected start point to selected destination, end point.

3.1.3.2. Use Case

Descriptive Use case Scenario

user (Actor)	"Dublin Commuting Live Routing" App
1. Actor turns on the App	
	2. App turns on. Welcome screen is briefly displayed and then Home screen is displayed
3. Actor enters start point	
4. Actor enters end point	

user (Actor)	“Dublin Commuting Live Routing” App
5. Actor presses “Search” button	
	6. App Saves Start and End points to history.
	7. App disables all widgets on the Home screen and displays refresh icon
	8. App gets required information from external APIs
	9. App performs required computation
	10. App displays list of available routes and on the “Route Options” screen.
	11. App goes into Auto refresh possible routes mode

Alternate flow: No Alternative flows available.

Exceptional flow(s): No network connection; external API information is not available

6. Application displays an error

Application continues from step 3 of main flow

Termination: App goes to “Auto refresh route options” use case

Post condition: The system goes into a wait state

3.1.4. Requirement 2: Get Start Point From Map

02-StartFromMap	Get Start Point From Map	Enable selection of the Start point directly from the Google Map by tap on the map, additionally enable adjustment of the selected point by drag and drop of the map marker	1	Map Position	GPS coordinates of map position passed to Start point

Identifier: 02-StartFromMap

3.1.4.1. Description & Priority

Priority 1 (highest)

This requirement ("Get Start Point From Map") is the very important requirement of the application. It provides the information about start location to the "Find Possible Routes" function, it has the highest priority.

This function of the systems provides the user availability to select starting point for the routing function by tapping on the displayed map. Once start point selected the address is retrieved using its GPS coordinated and Google's reverse GeoCoding, and relevant information (address GPS coordinates) are displayed to the user. In addition the drag and drop of the selected point marker is available to the user, which provides with functionality of quickly adjusting the start location.

3.1.4.2. Use Case

Descriptive Use case Scenario

user (Actor)	“Dublin Commuting Live Routing” App
1. Actor turns on the App	
	2. App turns on. Welcome screen is briefly displayed and then Home screen is displayed
3. Actor presses the “start from Map” button	
	4. App displays the Map
5. Actor clicks/taps on the required point on the map	
	6. App passes GPS coordinates of the selected point to Google reverse GeoCoding function in order to retrieve address
	7. App gets required information from external APIs
	8. App performs required computation
	9. App displays Address ad GPS coordinates of the selected point to the user
10 Actor accepts selected point be pressing “OK”	
	11. App saves selected point as the start point for routing function, displays the start point details on the home and goes to the “Home” screen
Alternate flow: .User cancels selection of the start point by closing the map. App display the information “Start point not saved” and displays “Home” screen	

user (Actor)	“Dublin Commuting Live Routing” App
Exceptional flow(s): No network connection; external API information is not available 6. Application displays an error Application continues from step 5 of main flow **Termination:** App goes to “Home” screen **Post condition:** The system goes into a wait state	

3.1.5. Requirement 3: Get End Point From Map

03-EndFromMap	Get End Point From Map	Enable selection of the End point directly from the Google Map by tap on the map, additionally enable adjustment of the selected point by drag and drop of the map marker	1	Map Position	GPS coordinates of map position passed to End point

Identifier: 03- EndFromMap

3.1.5.1. Description & Priority

Priority 1 (highest

)

This requirement (“Get End Point From Map”) is the very important requirement of the application. It provides the information about end location to the “Find Possible Routes” function, it has the highest priority.

This function of the systems provides the user availability to select end point for the routing function by tapping on the displayed map. Once end point selected the address is retrieved using its GPS coordinated and Google’s reverse GeoCoding, and relevant information (address GPS coordinates) are displayed to the user. In addition the drag and drop of the selected point marker is available to the user, which provides with functionality of quickly adjusting the end location.

3.1.5.2. Use Case

Descriptive Use case Scenario

user (Actor)	“Dublin Commuting Live Routing” App
1. Actor turns on the App	
	2. App turns on. Welcome screen is briefly displayed and then Home screen is displayed
3. Actor presses the “end from Map” button	
	4. App displays the Map
5. Actor clicks/taps on the required point on the map	
	6. App passes GPS coordinates of the selected point to Google reverse GeoCoding function in order to retrieve address

user (Actor)	“Dublin Commuting Live Routing” App
	7. App gets required information from external APIs
	8. App performs required computation
	9. App displays Address ad GPS coordinates of the selected point to the user
10 Actor accepts selected point be pressing “OK”	
	11. App saves selected point as the end point for routing function, displays the end point details on the home and goes to the “Home” screen

Alternate flow: .User cancels selection of the end point by closing the map. App display the information “End point not saved” and displays “Home” screen

Exceptional flow(s): No network connection; external API information is not available

6. Application displays an error

Application continues from step 5 of main flow

Termination: App goes to “Home” screen

Post condition: The system goes into a wait state

3.1.6. Requirement 4: Get Start Point From Device Location

04-StartFromLocation	Get Start Point From Device Location	Acquire GPS start point from device position	1	Device location service enabled	GPS coordinates of device location passed to Start point

Identifier: 04- StartFromLocation

3.1.6.1. Description & Priority

Priority 1 (highest)

This requirement ("Get Start Point From Device Location") is the very important requirement of the application. It provides the information about start location to the "Find Possible Routes" function, it has the highest priority.

This function of the systems provides the user availability pass device GPS location as the start point for the routing function. Once GPS coordinates of the device are obtained the address of the device location is retrieved using its GPS coordinates and Google's reverse GeoCoding, and relevant information (address GPS coordinates) are displayed to the user. In addition the drag and drop of the selected point marker is available to the user, which provides with functionality of quickly adjusting the start location.

3.1.6.2. Use Case

Descriptive Use case Scenario

user (Actor)	"Dublin Commuting Live Routing" App
1. Actor turns on the App	

user (Actor)	“Dublin Commuting Live Routing” App
	2. App turns on. Welcome screen is briefly displayed and then Home screen is displayed
3. Actor presses the “GeoLocation” button	
	4. App displays the Map and asks for permission to obtain GeoLocation
5. Actor provides permission to obtain device location	
	6. App used GeoLocation service to obtain GPS coordinates passes GPS coordinates of the selected point to Google reverse GeoCoding function in order to retrieve address
	7. App passes GPS coordinates to Google reverse GeoCoding function in order to retrieve address
	8. App gets required information from external API
	9. App performs required computation
	10. App displays Address and GPS coordinates of device location to the user
11 Actor accepts by pressing “OK”	
	12. App saves device location as the start point for routing function, displays the start point details on the home and goes to the “Home” screen

user (Actor)	“Dublin Commuting Live Routing” App
Alternate flow: .User cancels saving the device location by closing the map. App display the information “start point not saved” and displays “Home” screen **Exceptional flow(s):** GeoLocation service fails or No network connection; external API information is not available 6. Application displays an error Application continues from step 5 of main flow **Termination:** App goes to “Home” screen **Post condition:** The system goes into a wait state	

3.1.7. Requirement 5: Get Start Point From Search

05-StartFromSearch	Get Start Point From Search	Enable start point GPS coordinates to be acquired by address search with aid of the Google Places API	1	address of the start point	GPS coordinates from the search passed to Start point

Identifier: 05- StartFromSearch

3.1.7.1. Description & Priority

Priority 1 (highest)

This requirement (“Get Start Point From Search”) is the very important requirement of the application. It provides alternative way of passing the information about start location to the “Find Possible Routes” function, it has the highest priority.

This function of the systems provides the user availability to search for the start location using the address and place name. This function uses Google Places API to aid the user in the search for the address of the start point. Once start point address is selected the address additional relevant information are displayed to the user. In addition the drag and drop of the point marker is available to the user, which provides with functionality of quickly adjusting the start location.

3.1.7.2. Use Case

Descriptive Use case Scenario

user (Actor)	“Dublin Commuting Live Routing” App
1. Actor turns on the App	
	2. App turns on. Welcome screen is briefly displayed and then Home screen is displayed
3. Actor presses the “start from search” button	
	4. App displays the Map with the text field for entering the address
5. Actor enters the address	
	6. App provides hints with possible address matches
7. Actor selects one of the provided address options	
	8. App displays Address and relevant information of the selected address to the user
9 Actor accepts selected address by pressing “OK”	

user (Actor)	“Dublin Commuting Live Routing” App
	11. App saves selected address point as the start point for routing function, displays the start point details on the home and goes to the “Home” screen

Alternate flow: .User cancels selection of the start address by closing the map. App display the information “Start point not saved” and displays “Home” screen

Exceptional flow(s): No network connection; external API information is not available

6. Application displays an error

Application continues from step 5 of main flow

Termination: App goes to “Home” screen

Post condition: The system goes into a wait state

3.1.8. Requirement 6: Get End Point From Search

06-EndFromSearch	Get End Point From Search	Enable and point GPS coordinates to be acquired by address search with aid of the Google Places API	1	address of the end point	GPS coordinates from the search passed to End point

Identifier: 06- EndFromSearch

3.1.8.1. Description & Priority

Priority 1 (highest)

This requirement (“Get End Point From Search”) is the very important requirement of the application. It provides alternative way of passing the information about end location to the “Find Possible Routes” function, it has the highest priority.

This function of the systems provides the user availability to search for the end location using the address and place name. This function uses Google Places API to aid the user in the search for the address of the end point. Once end point address is selected the address and additional relevant information are displayed to the user. In addition the drag and drop of the point marker is available to the user, which provides with functionality of quickly adjusting the end location.

3.1.8.2. Use Case

Descriptive Use case Scenario

user (Actor)	“Dublin Commuting Live Routing” App
1. Actor turns on the App	
	2. App turns on. Welcome screen is briefly displayed and then Home screen is displayed
3. Actor presses the “end from search” button	
	4. App displays the Map with the text field for entering the address
5. Actor enters the address	
	6. App provides hints with possible address matches
7. Actor selects one of the provided address options	
	8. App displays Address and relevant information of the selected address to the user

user (Actor)	“Dublin Commuting Live Routing” App
9 Actor accepts selected address by pressing “OK”	
	11. App saves selected address point as the end point for routing function, displays the end point details on the home and goes to the “Home” screen

<u>Alternate flow:</u> .User cancels selection of the end address by closing the map. App display the information “End point not saved” and displays “Home” screen

<u>Exceptional flow(s):</u> No network connection; external API information is not available

6. Application displays an error

Application continues from step 5 of main flow

<u>Termination:</u> App goes to “Home” screen

<u>Post condition:</u> The system goes into a wait state

3.1.9. Non-Functional Requirements

3.1.9.1. Performance/Response time requirement

Speed of the application is the most important requirement. The system is the mobile application specifically aimed for the user on the move who require timely and accurate information in order to be able to take informed decision.

Code optimisation, refactoring and minimization should be used to achieve the best system performance. Performance testing of the application should be very thorough.

As long as all required APIs are available, the application must respond to 99% of user requests within 10 seconds of the request.

3.1.9.2. Usability

Usability is very important requirement of the application. It is essential that the system is user friendly and intuitive. That new user is should be comfortably operating the application after not more than an hour of learning to use the system.

3.1.9.3. Availability requirement

The system is designed as a mobile application which will be installed and run on user’s device and therefore it will be available for the user all the time. Nevertheless availability to provide prompt response will be dependent on the availability of all external APIs required – namely RTPI API, Dublin Bike API, Irish Rail API, Google Directions API, Google Maps API and Google Places API.

3.1.9.4. Robustness requirement

The application must be able to respond to errors, and continue execution despite abnormalities in data and calculation

Users must be kept informed on the state of the application.

Good coding standards and thorough testing of the system is required to ensure robustness.

3.1.9.5. Security requirement

As the mobile application which runs locally on user's device and doesn't required any sensitive data. The system doesn't have any specific security requirements. If some user specific data (e.g. favourite destinations) data will be required, it will be stored in the local database on the user's device.

3.1.9.6. Reliability requirement

The system should consistently perform according to the specification. If all required external APIs are available system should reliably respond with accurate information. System must be comprehensively tested in order to maximize its reliability.

3.1.9.7. Maintainability requirement

Maintainability of the system is important but performance of the application takes the priority. So in the case where good, system maintainability would have a negative impact on response times some maintainability practices might need to be sacrificed.

3.1.9.8. Portability requirement

The application will be developed using PhoneGap framework and will be available for main mobile platforms – Android, iOS, Windows Phone.

3.1.9.9. Extendibility requirement

Future growth of the system should be taken into consideration during design and implementation of the application. Good coding practice and code modularisation should be used.

3.1.10. User requirements

Users who use public transport within Greater Dublin area, want live, real time, frequently updated information on the arrival time to the selected destination. Users want Dublin Bikes to be included in the routing options.

Users want mobile application which is responsive has good graphical user interface design which is intuitive and easy to follow.

3.2. Design and Architecture

The system is implemented using web technologies: HTML5, CSS3, JavaScript and frameworks which aid in web development e.g. jQuery, jQueryMobile Bootstrap. All of the computation and data processing will be performed on the client side.

The mobile application will be created using the PhoneGap framework.

System will use static data stored in xml and json files as well as the live data provided by the external APIs. User specific data (e.g. favourite destinations,) will be stored in the local storage and kept in json format.

3.2.1. Application logic algorithm:

Below is a brief description of the process implemented by the application in order to provide additional information to the user based on the information available from the eternal APIs (RTPI API, Dublin Bikes API, Irish Rail API and Google APIs – Map, Direction and Places).

1. Get user current location (location of the smartphone) (or start point from the map)
2. Get destination location (search or map)
3. Button “Find Routes” – start and end point will be passed to the Google Directions API to get available public transport routes.
4. Data returned by Google Directions API will be used to check current departure times for trains, buses included in the routes. Possible routes will be updated with the actual departure times.
5. Possible routes will be checked for availability of Dublin Bikes stations along the way. If Dublin Bikes stations are available within reasonable distance from the stops or user’s current location, bikes will be included in the routing as additional route options. If bikes are available at the start bike station or free bike stands are not available – route option will be highlighted orange.
6. This whole process described above will be repeated every 60 seconds or on the user request, until stopped by the user. So the user will always have actual best arrival time to the destination location, and will be able to undertake informed decision whether to change the route or to stay on the chosen one.

Use of Dublin Bikes on the route.

User will be able to inform the App that he/she is currently using the Dublin Bike (button “On Dublin Bike”). If the user is cycling the App will keep checking the bike stations for empty stands close to the end of the current route section. If the closest station does not have free stands user will be alerted that he/she needs to cycle to different station and an alternative bike station will be suggested.

3.3. Implementation

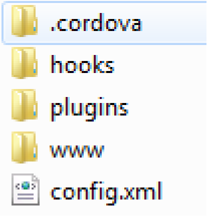
The application being developed is a hybrid mobile application implemented using:

- HTML5,
- CSS3,
- JavaScript,
- Bootstrap,
- JavaScript frameworks: jQuery, jQueryMobile.

The application installation file is created using PhoneGap which lets the system to be converted into mobile application for multiple mobile platforms (Android, iOS, Windows Phone).

Android .apk file is included with the source code files of this project submission.

The directory structure of the application is presented below:



.cordova
hooks
plugins
www
config.xml

Config.xml file contains the configuration necessary to enable PhoneGap application build.

Main application implementation files are in the www folder.

This folder contains:

- ***index.html*** file – this is the HTML5 with the user interface of the application
- css folder contains the application CSS3 styling. Application specific styling is in the ***main.css*** file. Other .css files in this folder are components of the frameworks used in the implementation - Bootstrap, jQuery, jQueryMobile. Reset.css file resets browser specific styling in order to remove any differences in the GUI rendering on different platforms.
- js folder contains the JavaScript files where the application functionality algorithms and logic are implemented, and also Bootstrap, jQuery and jQueryMobile scripts.
- ***app.js , ajax.js*** and ***map-functions.js*** JavaScript files which contain Application implementation.

- data folder contains json and xml files with static data used in the implementation,
- fonts folder contains glyphicons files used in the implementation
- images folder contains images for the application.

Local storage is used for persisting application settings and storing favourite and recent Start and Destination locations.

All application functionality is implemented using JavaScript which is event driven programming language.

As the application depends heavily on the data from external sources – e.g. Google Directions, RPTI API Dublin Bikes API – which are accessed thru the network and can take undetermined amount of time to respond, and successful response is not always possible – implementation relies quite often on the call-backs.

In JavaScript private variables can be made possible with closures.

Pieces of code which use closures are quite difficult to follow, but closures are very powerful feature of JavaScript.

I have used closures in few places in the application, particularly when registering on click events for bicycle markers info-windows data refresh.

3.4. Testing

Application was implemented with thorough debugging using Chrome Developer Tools and Mozilla Firefox Developer Tools.

Console logging was used quite often to follow application flow and verify variables' values. This can be seen in the source code files.

This development approach allowed to eliminate syntax errors and fix application logic errors encountered during debugging.

Set of User Acceptance Tests was prepared to verify application functionality and behaviour from the user point of view.

The results of these test are listed in tables below.

Screen Name:	Home

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
H	1	H1	Tap on the Get Geolocation button	Application opens progress screen and after few seconds once geolocation has been identified displays the Geolocation screen with the map and marker on the map with the info-window showing the address and GPS location	yes	-	as expected		
H	2	H2	Tap on the Get Start location from the map button	Application opens the Start From the Map screen	yes	-	as expected		
H	3	H3	Tap on the Get Start location from Address search button	Application opens the Start From Address search screen	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
H	4	H4	Tap on the Get Destination location from the map button	Application opens the End From the Map screen	yes	-	as expected		
H	5	H5	Tap on the Get Destination location from Address search button	Application opens the End From Address search screen	yes	-	as expected		
H	6	H6	Tap on the Find Available routes button - both start and destination not selected	Application displays popup alert with message "Start and destination not selected"	yes	-	as expected		
H	7	H7	Tap on the Find Available routes button - destination not selected	Application displays popup alert with message "Destination not selected"	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
H	8	H8	Tap on the Find Available routes button - start not selected	Application displays popup alert with message "Start not selected"	yes	-	as expected		
H	9	H9	Tap on the Find Available routes button - both start and destination selected	Application opens progress screen and starts searching for available Commuting, Cycling and Walking routes. Once all routes have been found screen with list of available routes is displayed, "updated" time is displayed and auto refresh counter starts counting down from 60	yes	-	as expected		
H	10	H10	Tap on the "Menu" button	dropdown menu is displayed	yes	-	as expected		
H	11	H11	Tap on the "Settings" menu option	Settings screen is opened	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
H	12	H12	Tap on the "Favourites" menu option	list of favourite locations opened	yes	-	as expected		
H	13	H13	Tap on the "History" menu option	list of recent locations is opened	yes	-	as expected		
H	14	H14	Tap on the "Favourites" button in the Start section	list of favourites start locations is opened	yes	-	as expected		
H	15	H15	Tap on the "History" button in the Start section	list of recent start locations is opened	yes	-	as expected		
H	16	H16	Tap on the "Favourites" button in the Destination section	list of favourites end locations is opened	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
H	17	H17	Tap on the "History" button in the Destination section	list of recent end locations is opened	yes	-	as expected		
H	18	H18	press device back button	Display popup alert box with question "Do you want to exit the App?"	yes	-	as expected		

Screen Name:	Geolocation

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
G	1	G1	Tap On the refresh button	Application opens progress screen and after few seconds once geolocation has been identified displays the Geolocation screen with the map and marker on the map with the info-window showing the address and GPS location	yes	-	as expected		
G	2	G2	drag and drop the location marker	Marker is positioned at the new location and info-window with updated Address and GPS coordinated is opened	yes	-	as expected		
G	3	G3	tap on the X close button	Geolocation screen is closed	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
G	4	G4	tap on the Favourites checkbox	Favourites checkbox changes state (from checked to unchecked or from unchecked to checked)	yes	-	as expected		
G	5	G5	Tap on the OK button - geolocation found and favourites checkbox unchecked	Geolocation screen is closed, and location is added to history of start locations; On the Home screen "History" button is enabled or remains enabled, address and GPS coordinates of start location are displayed in the Start textarea	yes	-	as expected		
G	6	G6	Tap on the OK button - geolocation found and favourites checkbox checked	Geolocation screen is closed, and location is added for favourite start locations; On the Home screen "Favourites" button is enabled or remains enabled, address and GPS coordinates of start location are displayed in the Start textarea	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
G	7	G7	Tap on the OK button - geolocation not D16found and favourites checkbox unchecked	Popup alert is displayed with message "No Goelocation..."	yes	-	as expected		
G	8	G8	Tap on the OK button - geolocation found and favourites checkbox checked	Popup alert is displayed with message "No Goelocation..."	yes	-	as expected		
G	9	G9	press device back button	Geolocation screen is closed	yes	-	as expected		

Screen Name:	Tap Start

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
TS	1	TS1	tap on the map to select start location	Marker is positioned at the selected location and info-window with address and GPS coordinated is opened	yes	-	as expected		
TS	2	TS2	drag and drop the location marker	Marker is positioned at the new location and info-window with updated Address and GPS coordinated is opened	yes	-	as expected		
TS	3	TS3	tap on the X close button	Tap Start screen is closed and App goes back to Home screen	yes	-	as expected		
TS	4	TS4	tap on the Favourites checkbox	Favourites checkbox changes state (from checked to unchecked or from unchecked to checked)	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
TS	5	TS5	Tap on the OK button - start location selected and favourites checkbox unchecked	Tap Start screen is closed, and location is added for history of start locations; On the Home screen "History" button is enabled or remains enabled, address and GPS coordinates of start location are displayed in the Start textarea	yes	-	as expected		
TS	6	TS6	Tap on the OK button - start location selected and favourites checkbox checked	Tap start screen is closed, and location is added for favourite start locations; On the Home screen "Favourites" button is enabled or remains enabled, address and GPS coordinates of start location are displayed in the Start textarea	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
TS	7	TS7	Tap on the OK button - start location not selected and favourites checkbox unchecked	Popup alert is displayed with message "Please select start location ..."	yes	-	as expected		
TS	8	TS8	Tap on the OK button - start location not selected and favourites checkbox checked	Popup alert is displayed with message "Please select start location ..."	yes	-	as expected		
TS	9	TS9	press device back button	Tap start screen is closed	yes	-	as expected		

Screen Name:	Address Start

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
AS	1	AS1	start entering the address of start location in the text box	list of suggested results appear	yes	-	as expected		Once address was already selected and user wants to search for new address, when user taps in the text box, previously found address remains in the text box and needs to be deleted by

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
									the user before new address can be entered. It would be a good idea to clear the text box when the user swipes on the text box
AS	2	AS2	select one of the suggested results	Marker is positioned on the map and info-window with location name and the address is opened	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
AS	3	AS3	drag and drop the location marker	Marker is positioned at the new location and info-window with updated Address and GPS coordinated is opened	yes	-	as expected		
AS	4	AS4	tap on the X close button	Address Start screen is closed and App goes back to Home screen	yes	-	as expected		
AS	5	AS5	tap on the Favourites checkbox	Favourites checkbox changes state (from checked to unchecked or from unchecked to checked)	yes	-	as expected		
AS	6	AS6	Tap on the OK button - start location found and favourites checkbox unchecked	Address Start screen is closed, and location is added to history of start locations, and new list is saved to local storage. On the Home screen "History" button is enabled or remains enabled, address and GPS coordinates of start location are displayed in the Start textarea	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
AS	7	AS7	Tap on the OK button - start location found and favourites checkbox checked	Address start screen is closed, and location is added to favourite start locations, and new list is saved to local storage. On the Home screen "Favourites" button is enabled or remains enabled, address and GPS coordinates of start location are displayed in the Start textarea	yes	-	as expected		
AS	8	AS8	Tap on the OK button - start location not found and favourites checkbox unchecked	Popup alert is displayed with message "Please select start location ..."	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
AS	9	AS9	Tap on the OK button - start location not found and favourites checkbox checked	Popup alert is displayed with message "Please select start location ..."	yes	-	as expected		
AS	10	AS10	press device back button	Address start screen is closed	yes	-	as expected		

Screen Name:	Tap End

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
TE	1	TE1	tap on the map to select end location	Marker is positioned at the selected location and info-window with address and GPS coordinated is opened	yes	-	as expected		
TE	2	TE2	drag and drop the location marker	Marker is positioned at the new location and info-window with updated Address and GPS coordinated is opened	yes	-	as expected		
TE	3	TE3	tap on the X close button	Tap End screen is closed and App goes back to Home screen	yes	-	as expected		
TE	4	TE4	tap on the Favourites checkbox	Favourites checkbox changes state (from checked to unchecked or from unchecked to checked)	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
TE	5	TE5	Tap on the OK button - end location selected and favourites checkbox unchecked	Tap End screen is closed, and location is added to history of start locations, and new list is saved to local storage. On the Home screen "History" button is enabled or remains enabled, address and GPS coordinates of end location are displayed in the Destination textarea	yes	-	as expected		
TE	6	TE6	Tap on the OK button - end location selected and favourites checkbox checked	Tap End screen is closed, and location is added to favourite end locations, and new list is saved to local storage. On the Home screen "Favourites" button is enabled or remains enabled, address and GPS coordinates of end location are displayed in the Destination textarea	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
TE	7	TE7	Tap on the OK button - end location not selected and favourites checkbox unchecked	Popup alert is displayed with message "Please select end location ..."	yes	-	as expected		
TE	8	TE8	Tap on the OK button - end location not selected and favourites checkbox checked	Popup alert is displayed with message "Please end start location ..."	yes	-	as expected		
TE	9	TE9	press device back button	Tap End screen is closed	yes	-	as expected		

Screen Name:	Address End

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
AE	1	AE1	start entering the address of end location in the text box	list of suggested results appears	yes	-	as expected		
AE	2	AE2	select one of the suggested results	Marker is positioned on the map and info-window with location name and the address is opened	yes	-	as expected		
AE	3	AE3	drag and drop the location marker	Marker is positioned at the new location and info-window with updated Address and GPS coordinated is opened	yes	-	as expected		
AE	4	AE4	tap on the X close button	Address Start screen is closed and App goes back to Home screen	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
AE	5	AE5	tap on the Favourites checkbox	Favourites checkbox changes state (from checked to unchecked or from unchecked to checked)	yes	-	as expected		
AE	6	AE6	Tap on the OK button - end location found and favourites checkbox unchecked	Address End screen is closed, and location is added to history of end locations, and new list is saved to local storage. On the Home screen "History" button is enabled or remains enabled, address and GPS coordinates of end location are displayed in the Destination textarea;	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
AE	7	AE7	Tap on the OK button - end location found and favourites checkbox checked	Address End screen is closed, and location is added to favourite end locations, and new list is saved to local storage. On the Home screen "Favourites" button is enabled or remains enabled, address and GPS coordinates of end location are displayed in the Destination textarea	yes	-	as expected		
AE	8	AE8	Tap on the OK button - end location not found and favourites checkbox unchecked	Popup alert is displayed with message "Please select end location ..."	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
AE	9	AE9	Tap on the OK button - end location not found and favourites checkbox checked	Popup alert is displayed with message "Please select end location ..."	yes	-	as expected		
AE	10	AE10	press device back button	Address end screen is closed	yes	-	as expected		

Screen Name:	Available Routes

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
R	1	R1	tap on the Back button	Available Routes screen closes and App goes back to the Home screen	yes	-	as expected		
R	2	R2	tap on the Stop button	Auto refresh counter stops counting down and Auto refresh is disabled	yes	-	as expected		
R	3	R3	tap on the Start button	Auto refresh counter is reset to 60 and starts counting down; Auto refresh is enabled	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
R	4	R4	tap on the Refresh button	Request for available routes is sent, results are displayed on the screen. Updated time is updated to new value. Auto refresh counter is reset to 60 and starts counting down; Auto refresh is enabled.	yes	-	as expected		Maybe auto update should only be enabled if it was enable prior of tapping on the Refresh button. If Auto update was disabled and user taps on the Refresh auto update is currently enabled and user needs to stop it again.

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
R	5	R5	Tap on the heading of currently opened route option	Currently opened route option is closed	yes	-	as expected		
R	6	R6	Tap on the heading of currently closed route option	Currently opened route option is closed and tapped route option is opened	yes	-	as expected		
R	7	R7	tap on the bicycle marker	Info-window with station details is opened - station number and location, number of available bikes, number of available bike stands, time when the details were last updated on the API server, and time when the App last checked for the details	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
R	8	R8	tap on the refresh on the bicycle info-window	details are updated - number of available bikes, number of available bike stands, time when the details were last updated on the API server, and time when the App last checked for the details	yes	-	as expected		
R	9	R9	press device back button	Available Routes screen closes and App goes back to the Home screen	yes	-	as expected		

Screen Name:	Settings

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
S	1	S1	Open settings screen, change some values and tap Save button both fields with values (Max Walk Distance, Days to keep history). Open settings again to verify that new values were saved	Settings screen closed, new values saved on GUI and to local storage, subsequent opening of settings screen confirms that new were saved	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
S	2	S2	tap Save button, Max Walk Distance empty	Alert popup displayed with message "One or more empty fields…"	yes	-	as expected		
S	3	S3	tap Save button, Days to keep history empty	Alert popup displayed with message "One or more empty fields…"	yes	-	as expected		
S	4	S4	tap Save button both fields empty (Max Walk Distance, Days to keep history)	Alert popup displayed with message "One or more empty fields…"	yes	-	as expected		
S	5	S5	change one or more values and tap reset button	all field values reset to the default values (Max Walk Distance = 1000; Don’t use Dublin Bikes = unchecked; Days to keep history = 10)	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
S	6	S6	Open settings screen, take note of the setting values; change some values and tap cancel button, open setting again to verify that changes were not saved and setting values are the same as before	Settings screen closed, any changes discarded, subsequent opening of settings screen confirms that values are kept unchanged	yes	-	as expected		
S	7	S7	tap on the X close button	Settings screen is closed and App goes back to Home screen, any changes to settings are discarded	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
S	8	S8	tap Cancel button	Settings screen is closed and App goes back to Home screen, any changes to settings are discarded	yes	-	as expected		
S	9	S9	press device back button	Settings screen is closed and App goes back to Home screen, any changes to settings are discarded	yes	-	as expected		

Screen Name:	Favourites

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
F	1	F1	tap on the X close button	Favourites screen is closed and App goes back to the Home screen	yes	-	as expected		
F	2	F2	tap Cancel button	Favourites screen is closed and App goes back to the Home screen	yes	-	as expected		
F	3	F3	press device back button	Favourites screen is closed and App goes back to the Home screen	yes	-	as expected		
F	4	F4	tap on one of the records on Start list	Favourites screen is closed, and tapped record is used as the start location - address and GPS location are displayed in the Start textarea	yes	-	as expected		
F	5	F5	tap on one of the records on Destination list	Favourites screen is closed, and tapped record is used as the end location - address and GPS location	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
				are displayed in the Destination textarea					
F	6	F6	swipe left or right on the record on Start list	Record is removed from the list. Favourites screen remains open		yes	no action on the phone		works on the Nexus 7 tablet, doesn’t work on the Galaxy Note phone
F	7	F7	swipe left or right on the record on End list	Record is removed from the list. Favourites screen remains open		yes	no action on the phone		works on the Nexus 7 tablet, doesn’t work on the Galaxy Note phone

Screen Name:	History

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
H	1	H1	tap on the X close button	History screen is closed and App goes back to the Home screen	yes	-	as expected		
H	2	H2	tap Cancel button	History screen is closed and App goes back to the Home screen	yes	-	as expected		
H	3	H3	press device back button	History screen is closed and App goes back to the Home screen	yes	-	as expected		
H	4	H4	tap on one of the records on Start list	History screen is closed, and tapped record is used as the start location - address and GPS location are displayed in the Start textarea	yes	-	as expected		
H	5	H5	tap on one of the records on Destination list	History screen is closed, and tapped record is used as the end location -	yes	-	as expected		

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
				address and GPS location are displayed in the Destination textarea					
H	6	H6	swipe left or right on the record on Start list	Record is removed from the list. History screen remains open		yes	no action on the phone		works on the Nexus 7 tablet, doesn't work on the Galaxy Note phone
H	7	H7	swipe left or right on the record on End list	Record is removed from the list. History screen remains open		yes	no action on the phone		works on the Nexus 7 tablet, doesn't work on the Galaxy Note phone

Screen Name:	Start Favourites

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
SF	1	SF1	tap on the X close button	Start Favourites screen is closed and App goes back to the Home screen	yes	-	as expected		
SF	2	SF2	tap Cancel button	Start Favourites screen is closed and App goes back to the Home screen	yes	-	as expected		
SF	3	SF3	press device back button	Start Favourites screen is closed and App goes back to the Home screen	yes	-	as expected		
SF	4	SF4	tap on one of the records on the list	Start Favourites screen is closed, and tapped record is used as the start location - address and GPS location are displayed in the Start textarea	yes	-	as expected		
SF	5	SF5	swipe left or right on the record on the list	Record is removed from the list. Favourites screen remains open		yes	no action on the phone		works on the Nexus 7 tablet, doesn't

									work on the Galaxy Note phone

Screen Name:	End Favourites

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
EF	1	EF1	tap on the X close button	End Favourites screen is closed and App goes back to the Home screen	yes	-	as expected		
EF	2	EF2	tap Cancel button	End Favourites screen is closed and App goes back to the Home screen	yes	-	as expected		
EF	3	EF3	tap device back button	End Favourites screen is closed and App goes back to the Home screen	yes	-	as expected		
EF	4	EF4	tap on one of the records on the list	End Favourites screen is closed, and tapped record is used as the end location - address and GPS location are displayed in the Destination textarea	yes	-	as expected		

EF	5	EF5	swipe left or right on the record on the list	Record is removed from the list. End Favourites screen remains open		yes	no action on the phone		works on the Nexus 7 tablet, doesn’t work on the Galaxy Note phone

Screen Name:	Start History						

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
SF	1	SF1	tap on the X close button	Start History screen is closed and App goes back to the Home screen	yes	-	as expected		
SF	2	SF2	tap Cancel button	Start History screen is closed and App goes back to the Home screen	yes	-	as expected		
SF	3	SF3	press device back button	Start History screen is closed and App goes back to the Home screen	yes	-	as expected		
SF	4	SF4	tap on one of the records on the list	Start History screen is closed, and tapped record is used as the start location - address and GPS location are displayed in the Start textarea	yes	-	as expected		
SF	5	SF5	swipe left or right on the record on the list	Record is removed from the list. Start History screen remains open		yes	no action on the phone		works on the Nexus 7 tablet, doesn’t

									work on the Galaxy Note phone

Screen Name:	End History

Test screen id	Test number	Test id	Test description	Expected results	Pass	Fail	Actual results	Identified issues	Comments
EH	1	EH1	tap on the X close button	End History screen is closed and App goes back to the Home screen	yes	-	as expected		
EH	2	EH2	tap Cancel button	End History screen is closed and App goes back to the Home screen	yes	-	as expected		
EH	3	EH3	press device back button	End History screen is closed and App goes back to the Home screen	yes	-	as expected		
EH	4	EH4	tap on one of the records on Destination list	End History screen is closed, and tapped record is used as the end location - address and GPS location are displayed in the Destination textarea	yes	-	as expected		

EH	5	EH5	swipe left or right on the record on End list	Record is removed from the list. End History screen remains open		yes	no action on the phone		works on the Nexus 7 tablet, doesn’t work on the Galaxy Note phone

3.5. Interface requirements

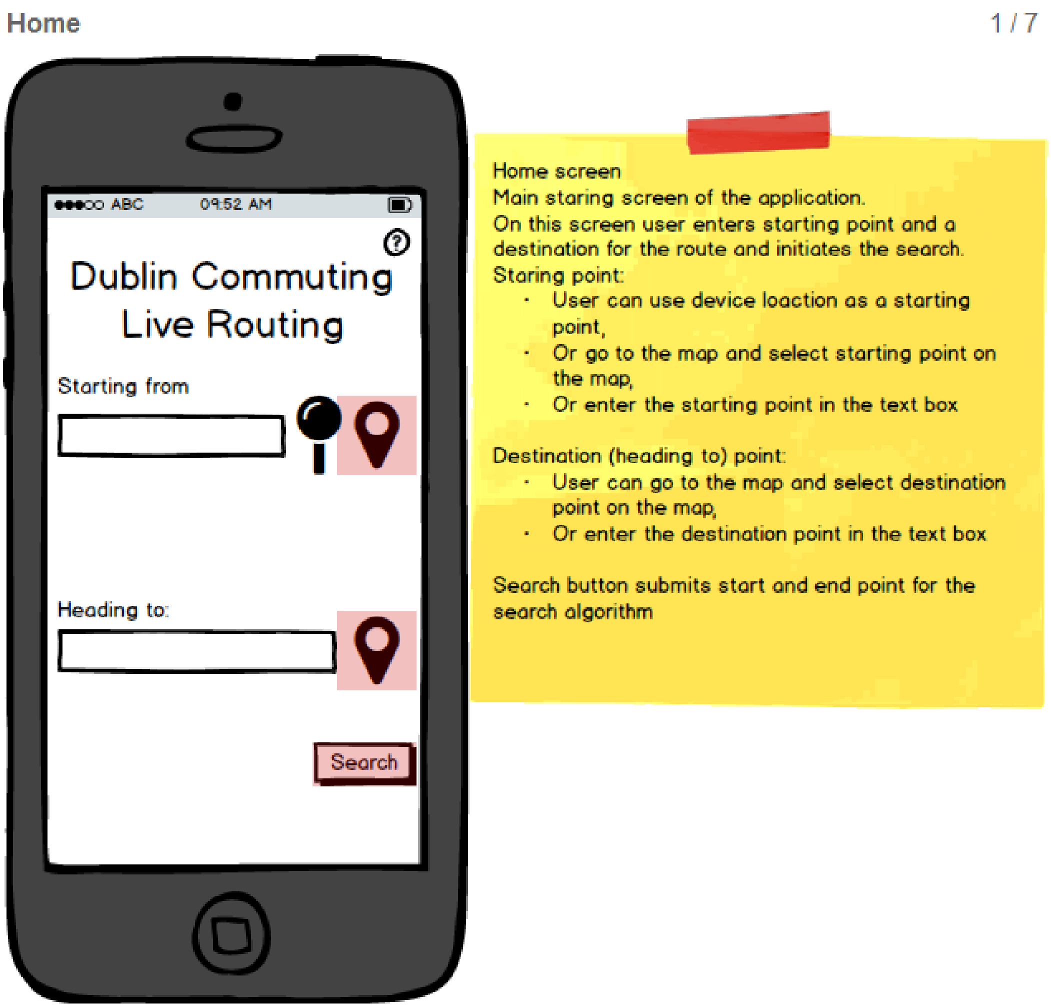
3.5.1. Graphical User Interface (GUI) Layout

Application GUI should be designed and optimised for smart phone and tablet devices.

Application GUI should be user friendly, intuitive and consistent.

3.5.1.1. GUI Mock-ups

GUI mock-up for smart phone devices is included below.



Home 1 / 7

Home screen
Main staring screen of the application.
On this screen user enters starting point and a destination for the route and initiates the search.
Staring point:

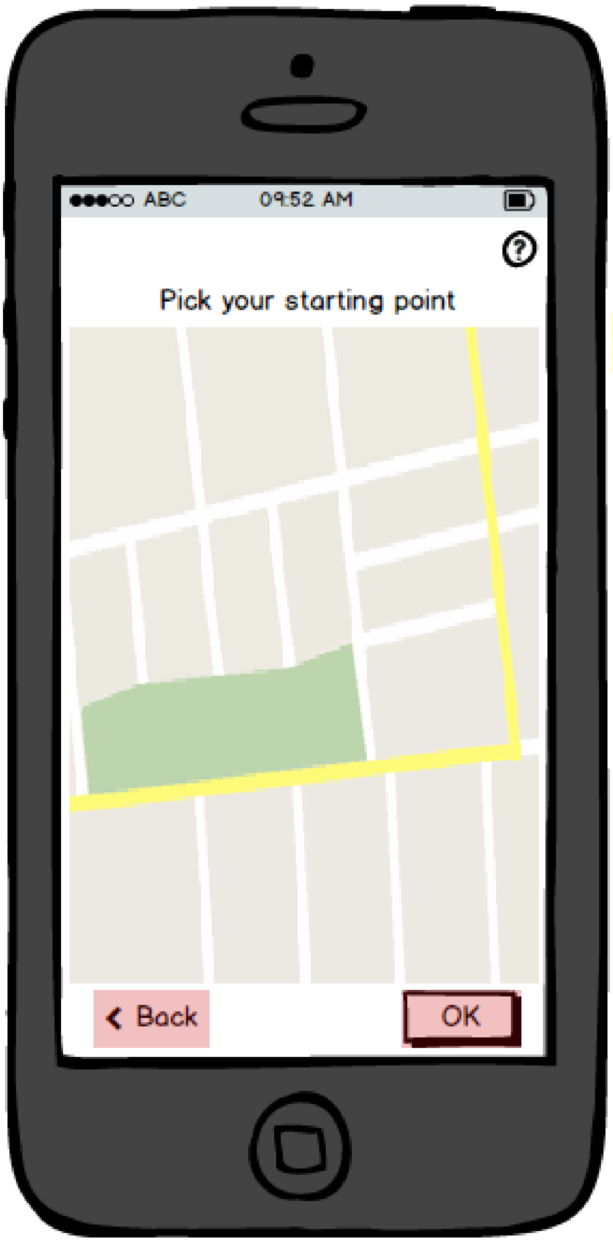
- User can use device loaction as a starting point,
- Or go to the map and select starting point on the map,
- Or enter the starting point in the text box

Destination (heading to) point:

- User can go to the map and select destination point on the map,
- Or enter the destination point in the text box

Search button submits start and end point for the search algorithm

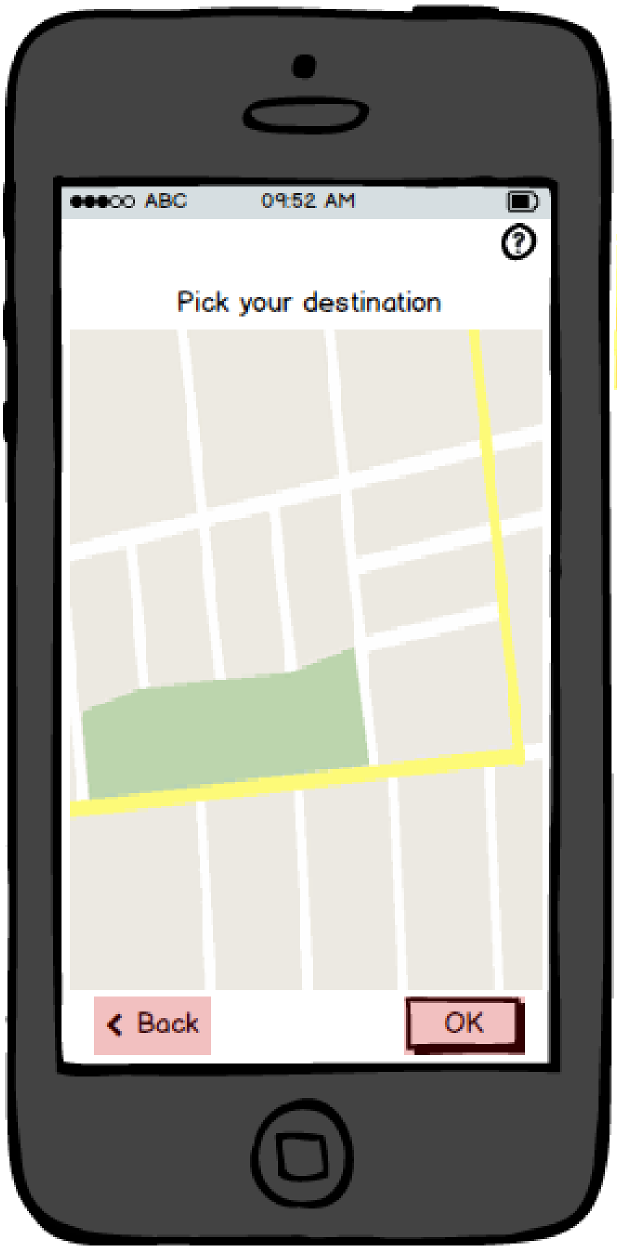
Pick Starting Point



Pick Starting Point screen

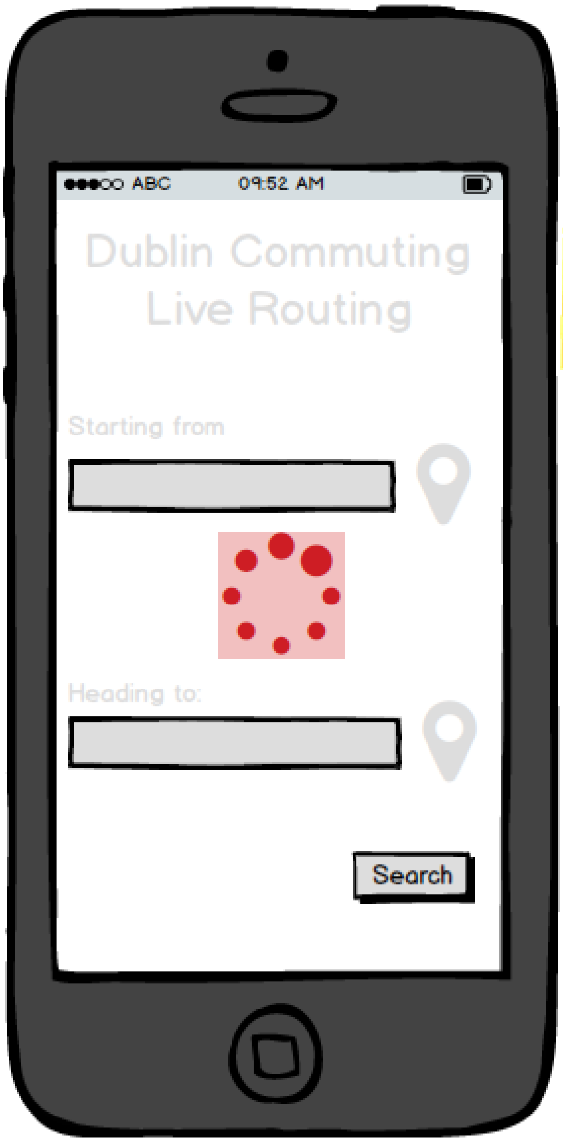
- On this screen User can select starting point from the map and submit by tapping "OK"
- or can return to "Home" screen by tapping "Back"

Pick Destination 3 / 7



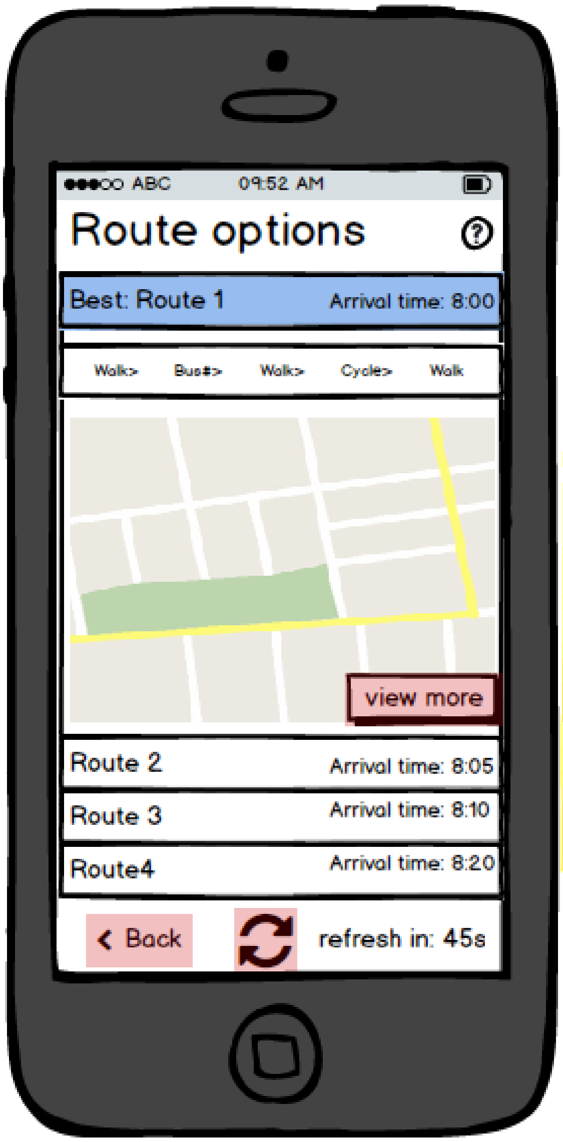
Pick Destination Point screen

- On this screen User can select destination point from the map and submit by tapping "OK"
- or can return to "Home" screen by tapping "Back"



Search screen.
This screen will indicate to the user that the search is in progress.
All widgets are disabled and "resresh" icon is displayed

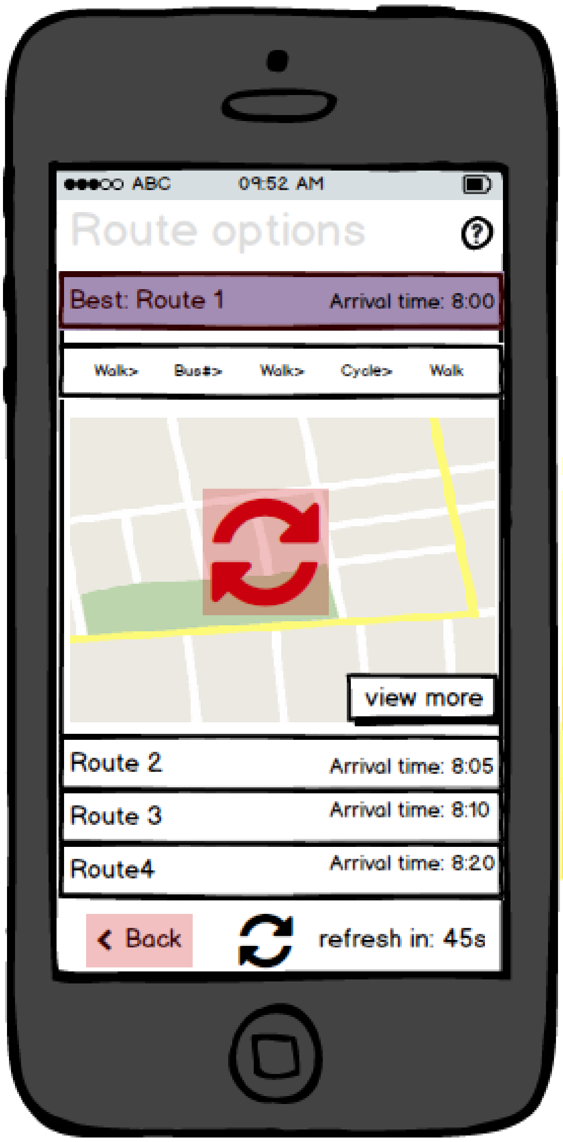
Route Options



Route Options Screen.
On this screen found route options for selected starting point and destination point are displayed in the arrival time order (erliest on top).

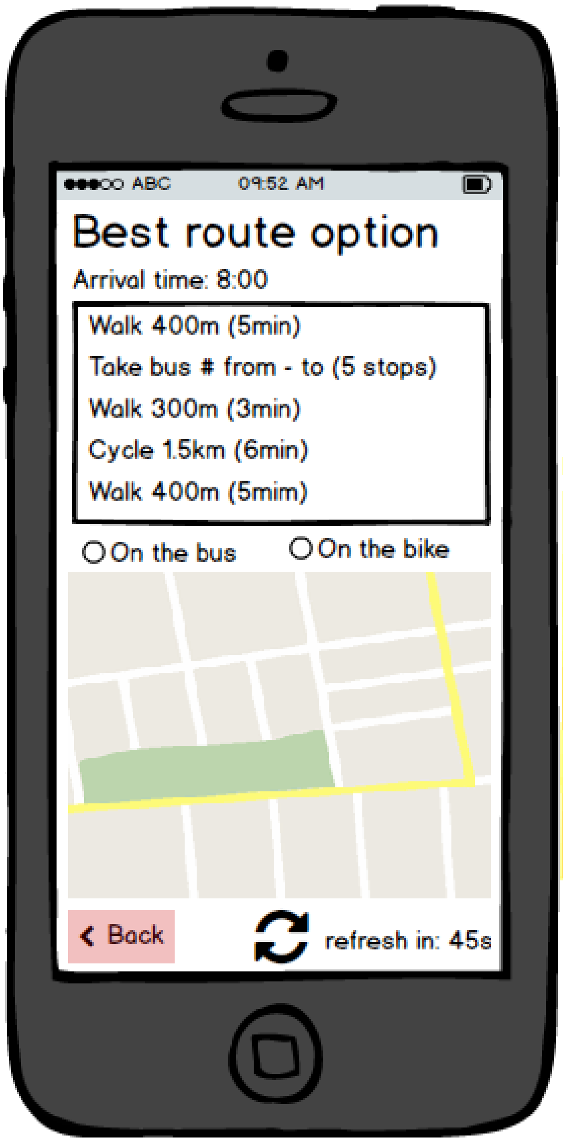
- User can switch between route option to view brief details
- "view more" buttons opens "Route Details" screen
- "Back" button returns to "Home" screen
- "Refresh" button allows user to initiate recalculation of the routes (if user used point on the map as a start point - user will be asked if this start point should be used or the device loaction)
- application will automatically refresh routes recalculation evary 60 seconds (possible item for settings option)

Refresh Options



Refresh Options screen.
This screen will indicate to the user that the route options refresh is in progress.
All widgets are disabled and "resresh" icon is displayed.

Route Details



Best route option

Arrival time: 8:00

Walk 400m (5min)
Take bus # from - to (5 stops)
Walk 300m (3min)
Cycle 1.5km (6min)
Walk 400m (5mim)

○ On the bus ○ On the bike

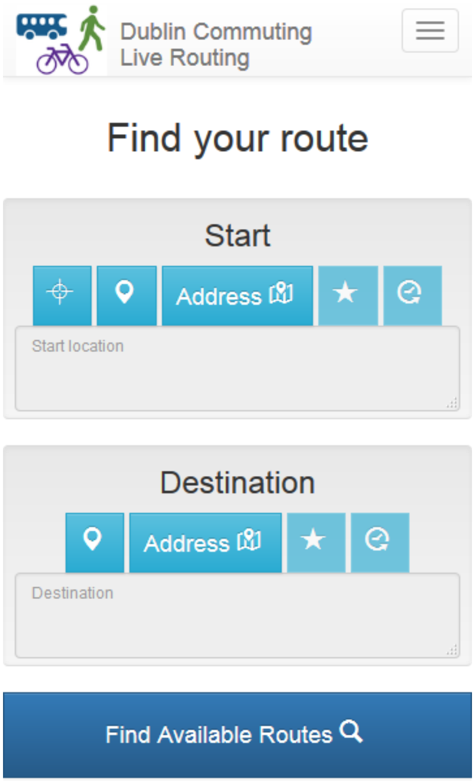
Route Details Screen.
On this screen detail for selected route option for selected starting point and destination point are displayed.

- User can indicate if he/she is currently "on the bus", "on the bike", "on the train" or "on the LUAS" (these options will dbe desplayed if they are part of the selected route)
- "Back" button takes the user back to "Route Options" screen
- "Refresh" button allows user to initiate recalculation of the selected route (if the user selected one of the means of transport ("on the bus", "on the bike", "on the train" or "on the LUAS" - this will be tacen into accout in the route recalculation)
- application will automatically refresh routes recalculation evary 60 seconds (possible item for settings option)

3.5.1.1. Implemented Application GUI.

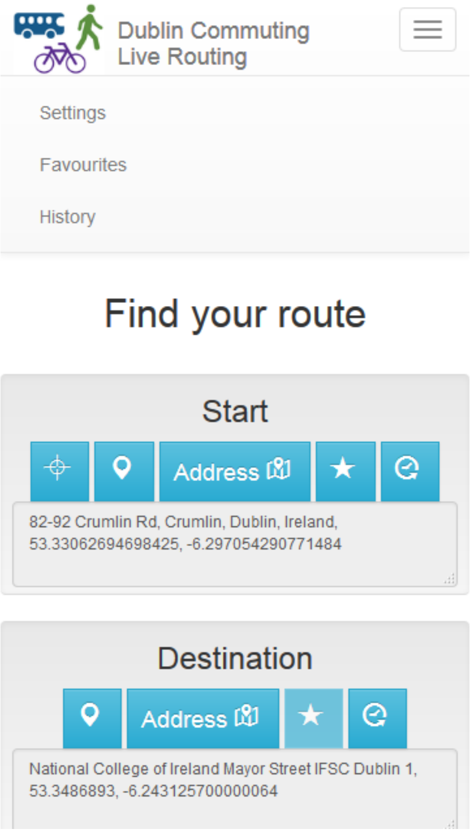
Home Screen:

This is the start screen of the application. On this screen user has access to application setting and all functional buttons.



Access to the Application settings, Favourites and history:

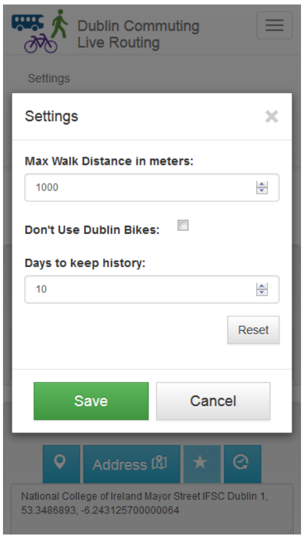
By tapping on the button ☰ in the top-right corner drop down menu opens where user can access Settings, Favourite locations and History of recent locations



Start: 82-92 Crumlin Rd, Crumlin, Dublin, Ireland, 53.33062694698425, -6.297054290771484

Destination: National College of Ireland Mayor Street IFSC Dublin 1, 53.3486893, -6.243125700000064

Settings Screen:



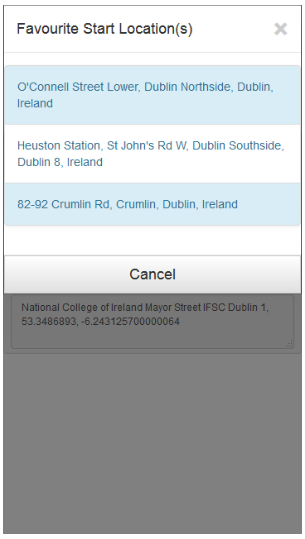
Max Walk Distance in meters: 1000

Don't Use Dublin Bikes: □

Days to keep history: 10

National College of Ireland Mayor Street IFSC Dublin 1, 53.3486893, -6.243125700000064

Favourites Screen:



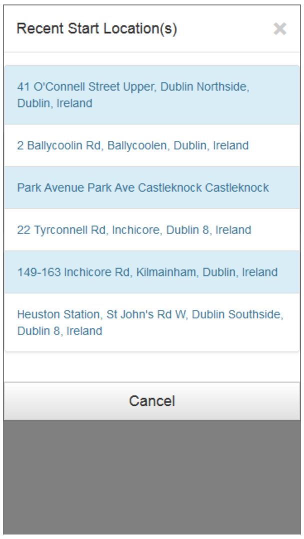
O'Connell Street Lower, Dublin Northside, Dublin, Ireland

Heuston Station, St John's Rd W, Dublin Southside, Dublin 8, Ireland

82-92 Crumlin Rd, Crumlin, Dublin, Ireland

National College of Ireland Mayor Street IFSC Dublin 1, 53.3486893, -6.243125700000064

History Screen:



41 O'Connell Street Upper, Dublin Northside, Dublin, Ireland

2 Ballycoolin Rd, Ballycoolen, Dublin, Ireland

Park Avenue Park Ave Castleknock Castleknock

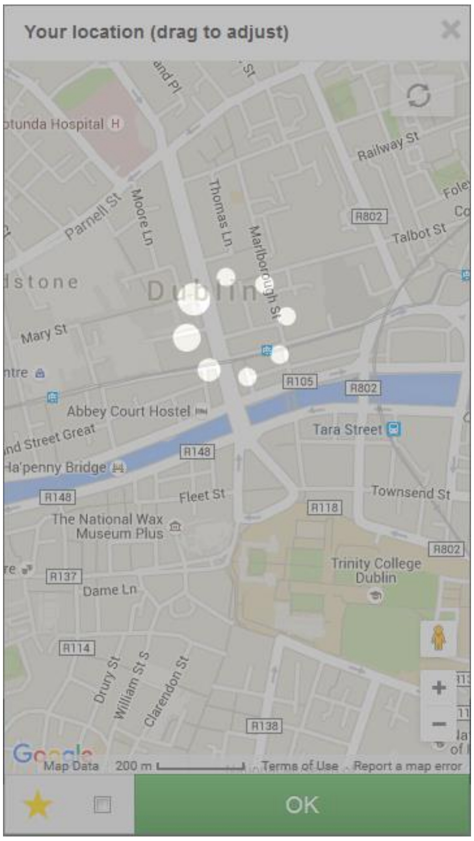
22 Tyrconnell Rd, Inchicore, Dublin 8, Ireland

149-163 Inchicore Rd, Kilmainham, Dublin, Ireland

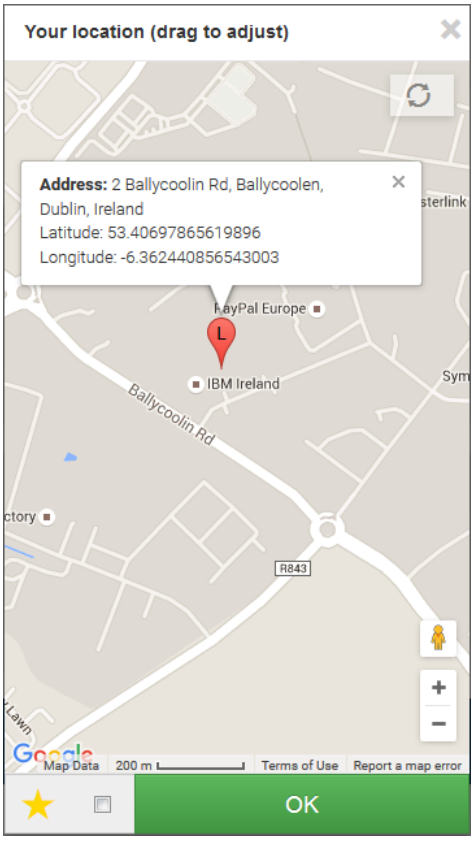
Heuston Station, St John's Rd W, Dublin Southside, Dublin 8, Ireland

Device Geolocation Screen:

By tapping this button ⌖ on the home screen. Application opens the screen below and tries to obtain device geolocation, if successful location will be displayed as a marker on the map with the info window with address and geolocation. User can adjust location by drag and drop. OK button confirms start location. And X button closes the screen.



Your location (drag to adjust)

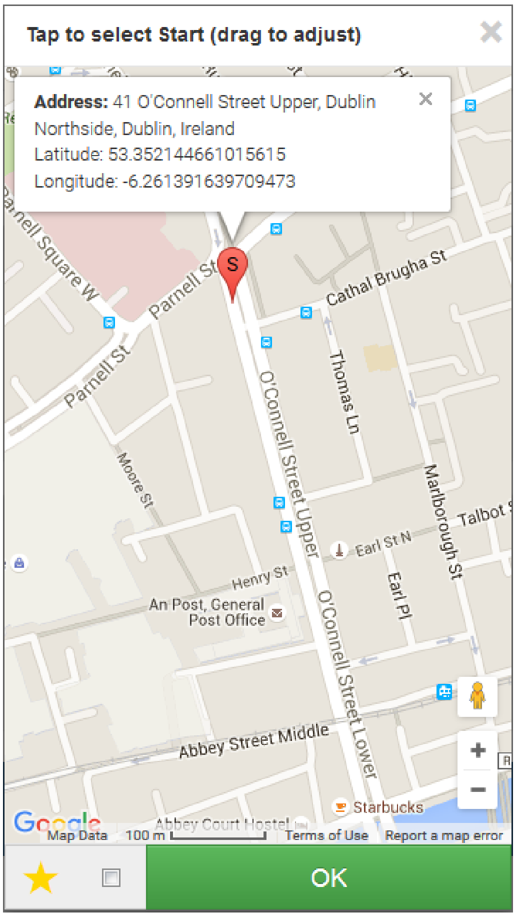


Your location (drag to adjust)

Address: 2 Ballycoolin Rd, Ballycoolen, Dublin, Ireland
Latitude: 53.40697865619896
Longitude: -6.362440856543003

Start Location from the Map screen:

By tapping this button in the “Start” section on the home screen application opens the screen below where user can select Start location by tapping on the map. Marker with the info window with address and GPS location. User can adjust location by drag and drop. OK button confirms start location. And X button closes the screen.

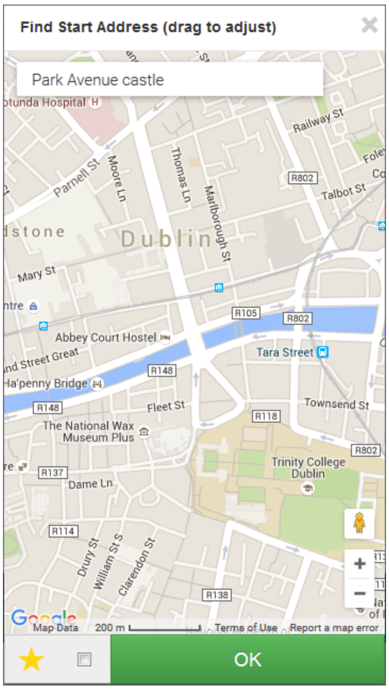


Tap to select Start (drag to adjust)

Address: 41 O'Connell Street Upper, Dublin Northside, Dublin, Ireland
Latitude: 53.352144661015615
Longitude: -6.261391639709473

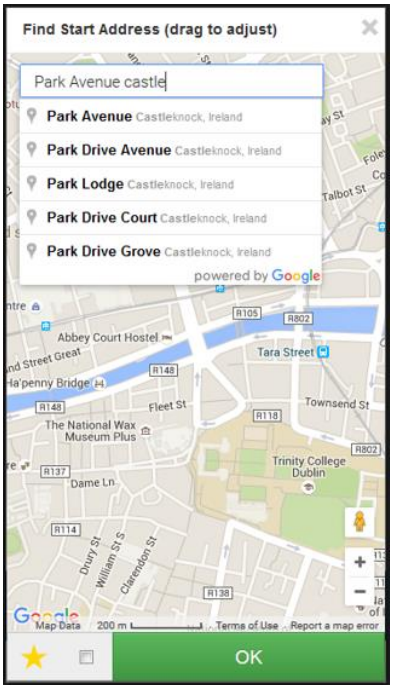
Start Location from Address search screen:

By tapping this button Address in the “Start” section on the home screen, application opens the screen below where user can search for the address. Goole Places API is used to provide suggestions. Once address is selected, marker with the info window with address and GPS location will be added to the map. User can adjust location by drag and drop. OK button confirms start location. And X button closes the screen.



Find Start Address (drag to adjust) ✕

Park Avenue castle

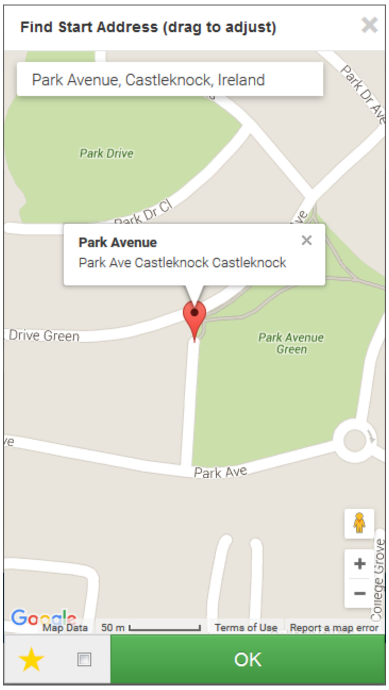


Find Start Address (drag to adjust) ✕

Park Avenue castle

- Park Avenue Castleknock, Ireland
- Park Drive Avenue Castleknock, Ireland
- Park Lodge Castleknock, Ireland
- Park Drive Court Castleknock, Ireland
- Park Drive Grove Castleknock, Ireland

powered by Google



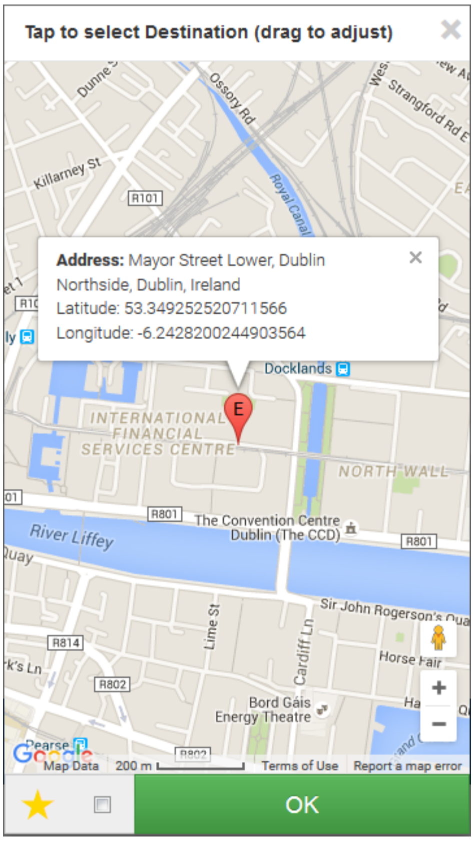
Find Start Address (drag to adjust) ✕

Park Avenue, Castleknock, Ireland

Park Avenue
Park Ave Castleknock Castleknock ✕

End Location from the Map screen:

By tapping this button in the “Destination” section on the home screen application, opens the screen below where user can select Start location by tapping on the map. Marker with the info window with address and GPS location. User can adjust location by drag and drop. OK button confirms start location. And X button closes the screen.

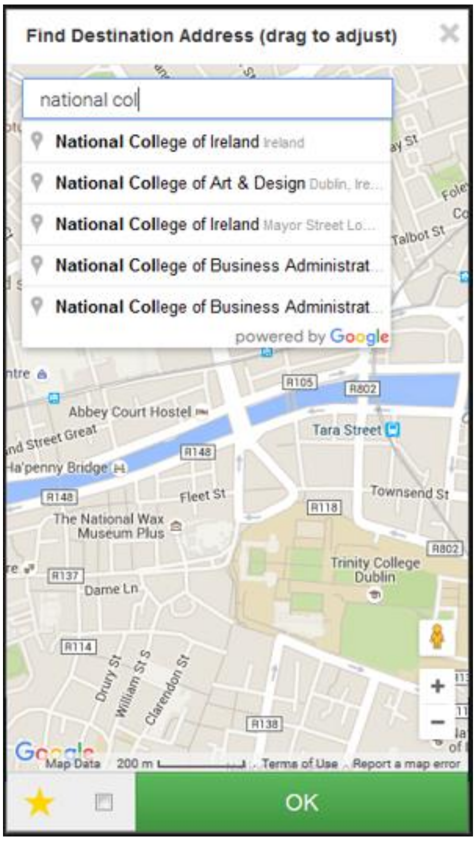


Tap to select Destination (drag to adjust)

Address: Mayor Street Lower, Dublin Northside, Dublin, Ireland
Latitude: 53.349252520711566
Longitude: -6.2428200244903564

End Location from Address search screen:

By tapping this button Address in the “Destination” section on the home screen, application opens the screen below where user can search for the address. Goole Places API is used to provide suggestions. Once address is selected, marker with the info window with address and GPS location will be added to the map. User can adjust location by drag and drop. OK button confirms start location. And X button closes the screen.

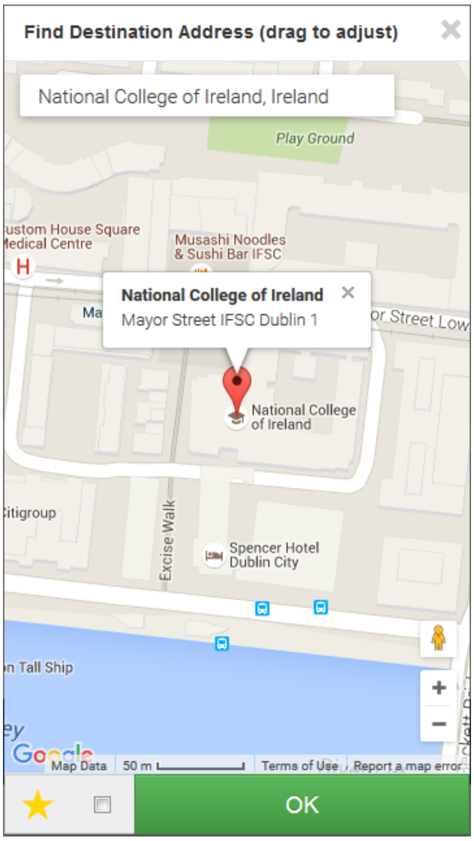


Find Destination Address (drag to adjust)

national col

- National College of Ireland Ireland
- National College of Art & Design Dublin, Ire...
- National College of Ireland Mayor Street Lo...
- National College of Business Administrat...
- National College of Business Administrat...

powered by Google



Find Destination Address (drag to adjust)

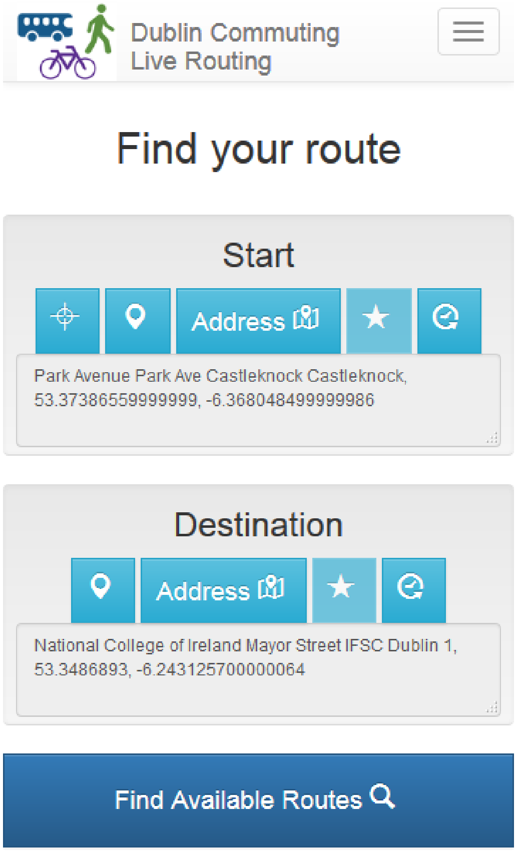
National College of Ireland, Ireland

National College of Ireland

Mayor Street IFSC Dublin 1

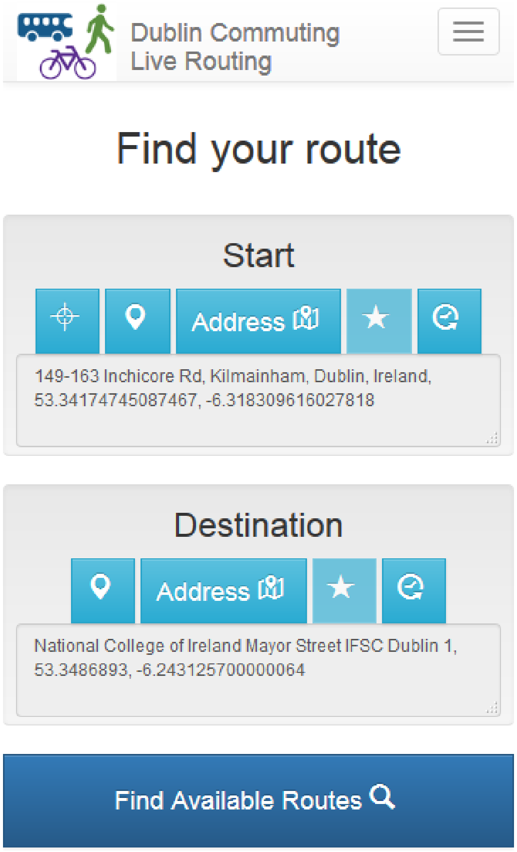
✕
Play Ground
Custom House Square Medical Centre
Musashi Noodles & Sushi Bar IFSC
National College of Ireland
Citigroup
Excise Walk
Spencer Hotel Dublin City
Tall Ship
Google
Map Data 50 m
Terms of Use
Report a map error
★
□
OK

Home Screen with selected Start and Destination locations:



Park Avenue Park Ave Castleknock Castleknock, 53.37386559999999, -6.368048499999986

National College of Ireland Mayor Street IFSC Dublin 1, 53.3486893, -6.243125700000064



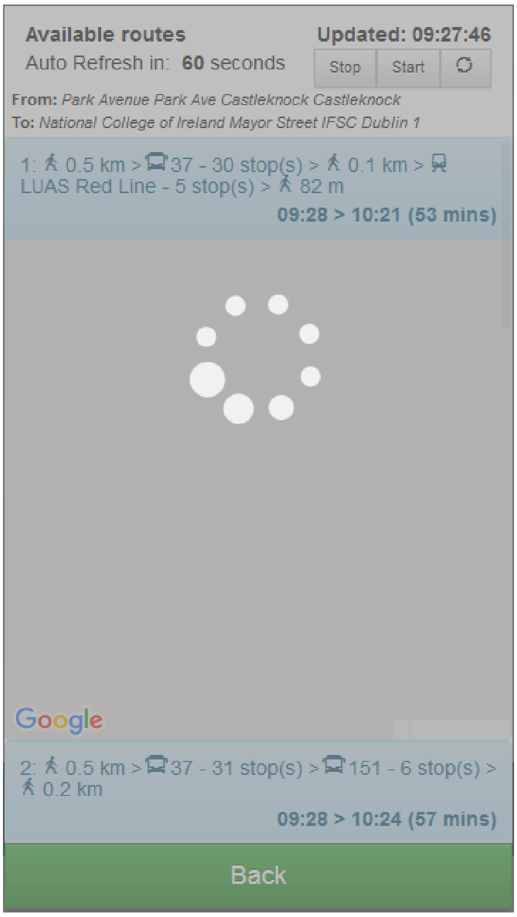
149-163 Inchicore Rd, Kilmainham, Dublin, Ireland, 53.34174745087467, -6.318309616027818

National College of Ireland Mayor Street IFSC Dublin 1, 53.3486893, -6.243125700000064

Once start and destination locations are selected user can tap the “Find available Routes” button

, to start searching for available commuting, Cycling (Dublin Bikes) and walking routes.

Application first displays “search in progress”. And after few seconds list of available routes is displayed:



Available routes **Updated: 09:27:46**

Auto Refresh in: **60** seconds

From: *Park Avenue Park Ave Castleknock Castleknock*

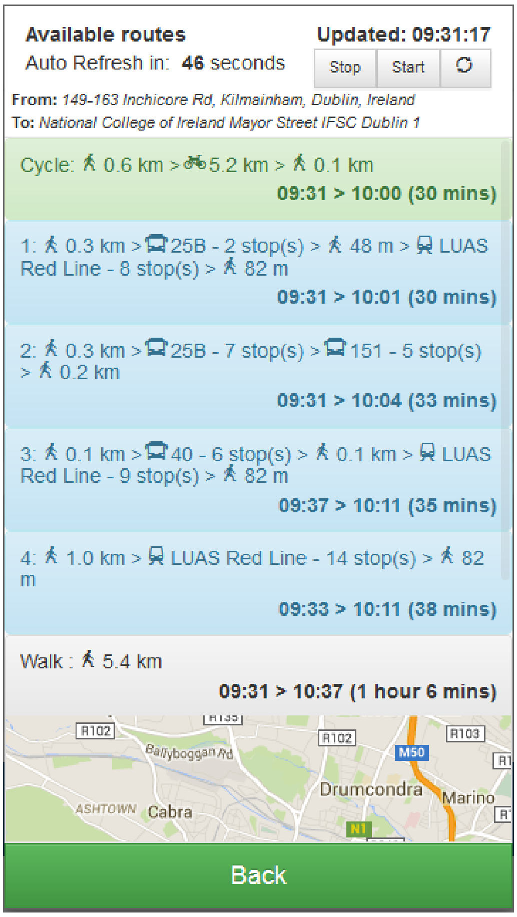
To: *National College of Ireland Mayor Street IFSC Dublin 1*

1: 🚶 0.5 km > 🚌37 - 30 stop(s) > 🚶 0.1 km > 🚋 LUAS Red Line - 5 stop(s) > 🚶 82 m

09:28 > 10:21 (53 mins)

2: 🚶 0.5 km > 🚌37 - 31 stop(s) > 🚌151 - 6 stop(s) > 🚶 0.2 km

09:28 > 10:24 (57 mins)



Available routes **Updated: 09:31:17**

Auto Refresh in: **46** seconds

From: *149-163 Inchicore Rd, Kilmainham, Dublin, Ireland*

To: *National College of Ireland Mayor Street IFSC Dublin 1*

Cycle: 🚶 0.6 km > 🚲5.2 km > 🚶 0.1 km

09:31 > 10:00 (30 mins)

1: 🚶 0.3 km > 🚌25B - 2 stop(s) > 🚶 48 m > 🚋 LUAS Red Line - 8 stop(s) > 🚶 82 m

09:31 > 10:01 (30 mins)

2: 🚶 0.3 km > 🚌25B - 7 stop(s) > 🚌151 - 5 stop(s) > 🚶 0.2 km

09:31 > 10:04 (33 mins)

3: 🚶 0.1 km > 🚌40 - 6 stop(s) > 🚶 0.1 km > 🚋 LUAS Red Line - 9 stop(s) > 🚶 82 m

09:37 > 10:11 (35 mins)

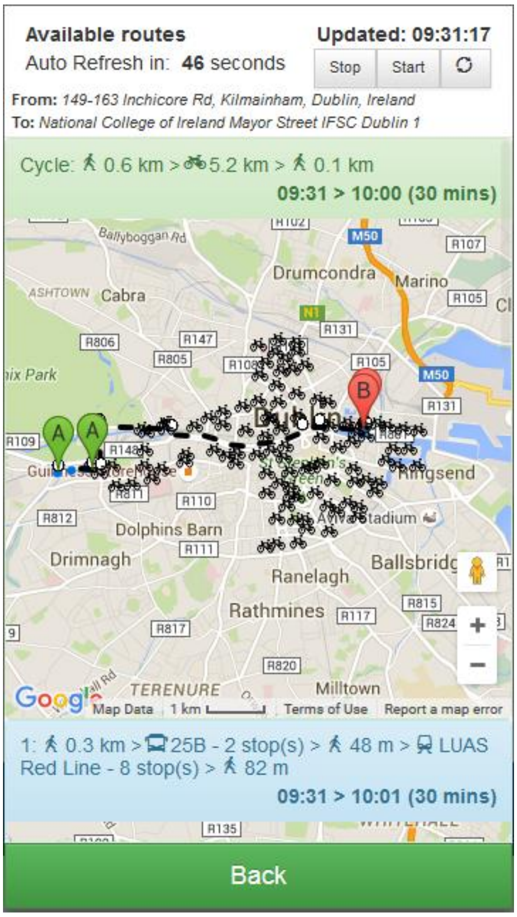
4: 🚶 1.0 km > 🚋 LUAS Red Line - 14 stop(s) > 🚶 82 m

09:33 > 10:11 (38 mins)

Walk : 🚶 5.4 km

09:31 > 10:37 (1 hour 6 mins)

User can switch between available routes, zoom in and out, and tap on the markers for direction details and tap on the bicycle markers to display current situation at the bike station (available bikes, free stands) and request new updated information.



Available routes **Updated: 09:31:17**

Auto Refresh in: **46** seconds

From: *149-163 Inchicore Rd, Kilmainham, Dublin, Ireland*

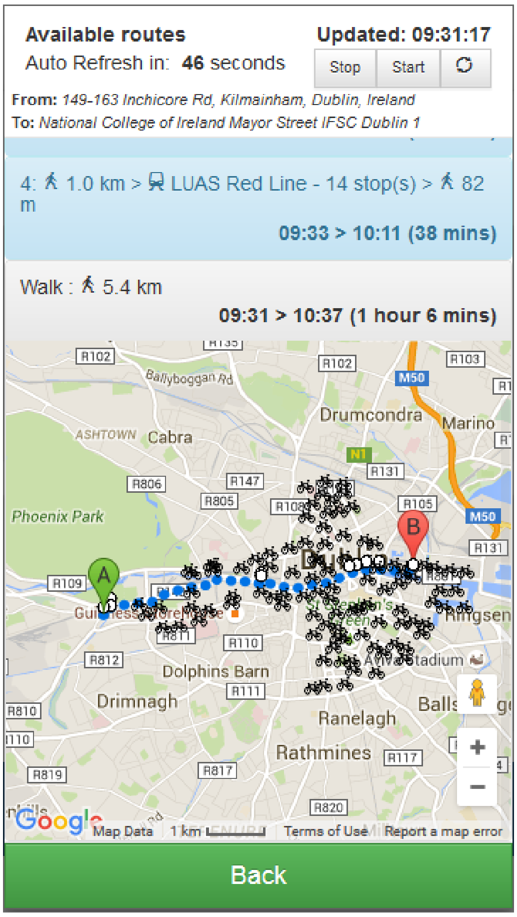
To: *National College of Ireland Mayor Street IFSC Dublin 1*

Cycle: 🚶 0.6 km > 🚲 5.2 km > 🚶 0.1 km

09:31 > 10:00 (30 mins)

1: 🚶 0.3 km > 🚌 25B - 2 stop(s) > 🚶 48 m > 🚊 LUAS Red Line - 8 stop(s) > 🚶 82 m

09:31 > 10:01 (30 mins)



Available routes **Updated: 09:31:17**

Auto Refresh in: **46** seconds

From: *149-163 Inchicore Rd, Kilmainham, Dublin, Ireland*

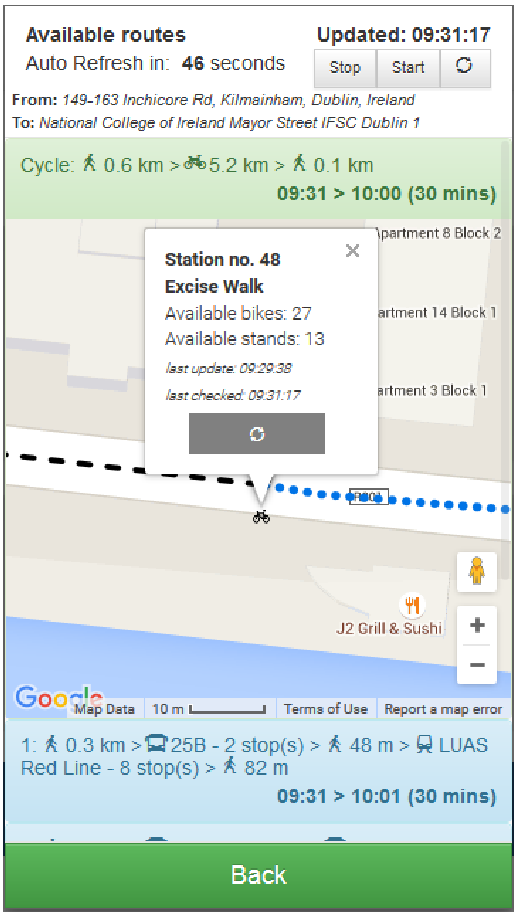
To: *National College of Ireland Mayor Street IFSC Dublin 1*

4: 🚶 1.0 km > 🚊 LUAS Red Line - 14 stop(s) > 🚶 82 m

09:33 > 10:11 (38 mins)

Walk : 🚶 5.4 km

09:31 > 10:37 (1 hour 6 mins)



Available routes **Updated: 09:31:17**

Auto Refresh in: **46** seconds

From: *149-163 Inchicore Rd, Kilmainham, Dublin, Ireland*

To: *National College of Ireland Mayor Street IFSC Dublin 1*

Cycle: 🚶 0.6 km > 🚲5.2 km > 🚶 0.1 km

09:31 > 10:00 (30 mins)

Station no. 48

Excise Walk

Available bikes: 27

Available stands: 13

last update: 09:29:38

last checked: 09:31:17

1: 🚶 0.3 km > 🚌25B - 2 stop(s) > 🚶 48 m > 🚊 LUAS Red Line - 8 stop(s) > 🚶 82 m

09:31 > 10:01 (30 mins)

3.5.2. Application Programming Interfaces (API)

The application make uses of the APIs listed below:

- Real Time Passanger Information API - RTPI REST Web Services API (RRWS API) provides REST interface to retrieve information on real time bus information, timetables and bus stops.,
- Dublin Bikes API - Dublin Bikes real time and static data,
- Irish Rail Real-time API,
- Google Maps API,
- Google Direction API,
- Google Places API.

3.6. Customer testing

Customer testing was performed with the execution of set of User Acceptance Test.

The results of the UAT tests are included in section 3.4 of this document.

3.7. Evaluation

Application was installed on the devices of few potential users who used the application for few days.

Comments received from the users are listed below:

- Application is a very good idea.
- Responsiveness of the application could be improved.
- Popup notification of current arrival/departure time of busses trains could be added
- Notification of current situation at the starting and ending Dublin bike stations is something which would added value to the application.

4. Conclusions

Challenges of event driven programming and multiple asynchronous requests to the external services and APIs (Google Directions, Dublin Bikes, RTPI,) has proven to be very time consuming and difficult to debug.

Also developing hybrid mobile application which performs much like a native application is quite problematic. I have encountered many situations were functionality tested in the browser on the PC was working fine, but when tested on the mobile device behaviour was different and not as expected. One example of the problem could be the use of modal windows, which on the mobile devices cause to transmit on click (touch) evens to the background screen.

Also on touch events perform differently on different devices, e.g. on swipe removal of records worked correctly on the tablet but did not work on the phone.

I think the Application idea is quite good and has a potential of being very useful.

There are a lot of additional features, which could be added to the application in order to increase its usability.

5. System Evolution, further development or research

Future evolution of the application could include additional transport options once they become available, also use of taxis as the form of transport, in the circumstances where the user is in a hurry, could be implemented.

Further development of the system could include implementation of the features planned during the analysis and design phase, which were not implemented due to the time constraints. Also additional update of the algorithm including Dublin Bikes on the routes and how the information is presented to the user could be the area of further development. Popup notification of changing conditions could be included. For example delays, bikes and free stands availability.

JQuery mobile version of the application with the main focus on the optimisation of speed and enhancement of user experience could be another area of future development.

6. References

Alan Dennis, B. H. W. D. T., (2009). *Systems Analysis and Design with UML.* 3 ed. s.l.:Wiley.

Bootstrap, n.d. *Bootstrap.* [Online]
Available at: http://getbootstrap.com/
[Accessed 26 September 2015].

Code School, n.d. *Code School.* [Online]
Available at: https://www.codeschool.com/
[Accessed 2015/2016].

dublinked, n.d. *http://dublinked.com/datastore/datasets/dataset-326.php.* [Online]
Available at: http://dublinked.com/datastore/datasets/dataset-326.php
[Accessed September 2015].

Dublinked, n.d. *Real-time Passenger Information (RTPI) for Dublin Bus, Bus Eireann, Luas and Irish rail.* [Online]
Available at: http://dublinked.com/datastore/datasets/dataset-300.php
[Accessed October 2015].

Google, n.d. *Google Maps Directions API.* [Online]
Available at: https://developers.google.com/maps/documentation/directions/
[Accessed October 2015].

Google, n.d. *Google Maps JavaScript API.* [Online]
Available at: https://developers.google.com/maps/documentation/javascript/
[Accessed October 2015].

Google, n.d. *Google Places API.* [Online]
Available at: https://developers.google.com/places/
[Accessed October 2015].

JCDecaux developer, n.d. *JCDecaux developer.* [Online]
Available at: https://developer.jcdecaux.com/#/opendata/vls?page=getstarted
[Accessed September 2015].

jQuery mobile, n.d. *jQuery mobile.* [Online]
Available at: https://jquerymobile.com/
[Accessed 26 September 2015].

jQuery, n.d. *jQuery - write less do more.* [Online]
Available at: https://jquery.com/
[Accessed 26 September 2015].

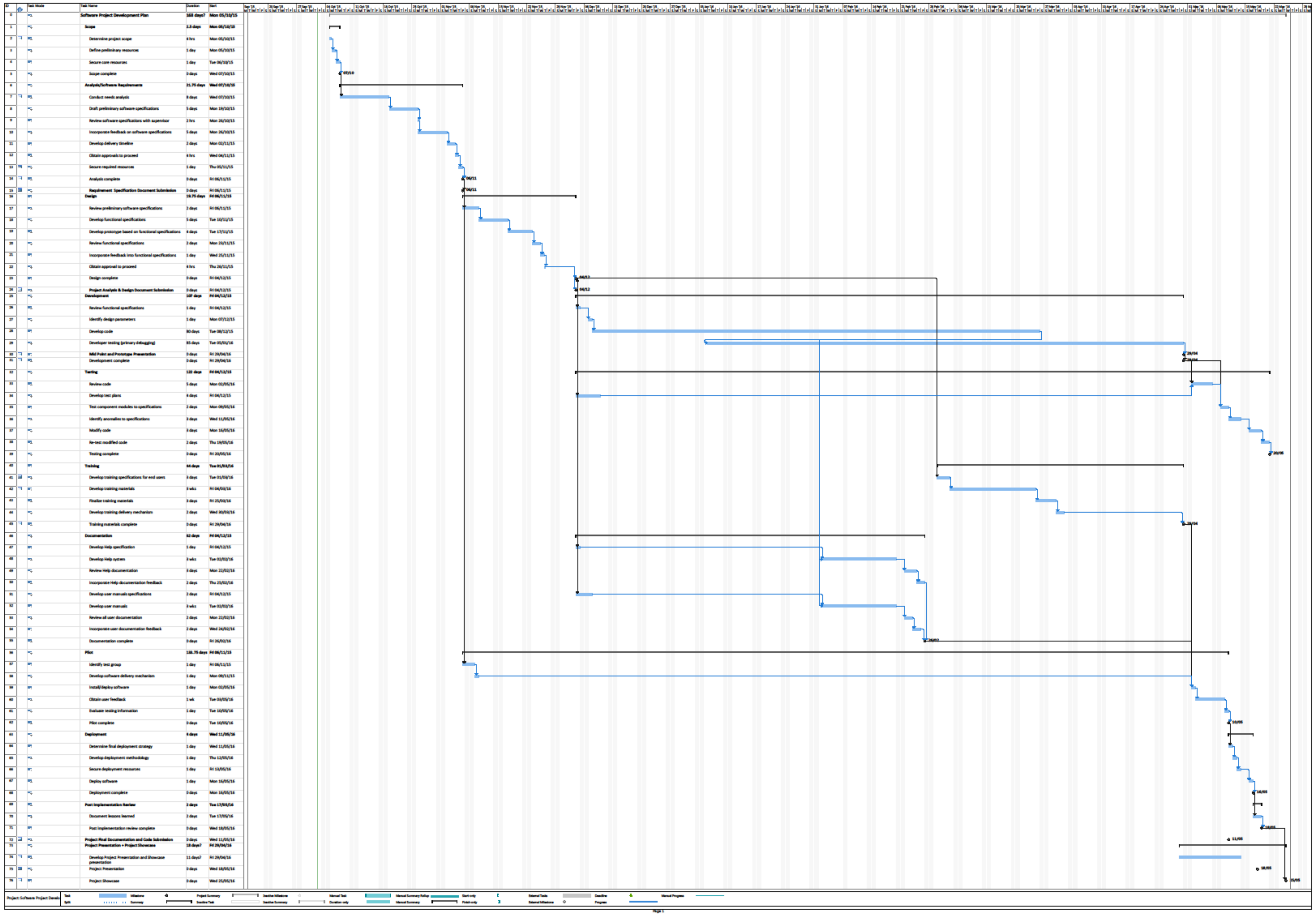
PhoneGap, n.d. *PhoneGap.* [Online]
Available at: http://phonegap.com/
[Accessed 26 September 2015].

Pressman, R. S., 2010. *Software Engineering: A Practitioner's Approach.* 7 ed. s.l.:R. S. Pressman & Associates, Inc..

7. Appendix

7.1. Project Plan

Gantt chart



ID		Task Mode	Task Name	Duration	Start
0			**Software Project Development Plan**	**168 days?**	**Mon 05/10/15**
1			**Scope**	**2.5 days**	**Mon 05/10/15**
2			Determine project scope	4 hrs	Mon 05/10/15
3			Define preliminary resources	1 day	Mon 05/10/15
4			Secure core resources	1 day	Tue 06/10/15
5			Scope complete	0 days	Wed 07/10/15
6			**Analysis/Software Requirements**	**21.75 days**	**Wed 07/10/15**
7			Conduct needs analysis	8 days	Wed 07/10/15
8			Draft preliminary software specifications	5 days	Mon 19/10/15
9			Review software specifications with supervisor	2 hrs	Mon 26/10/15
10			Incorporate feedback on software specifications	5 days	Mon 26/10/15
11			Develop delivery timeline	2 days	Mon 02/11/15
12			Obtain approvals to proceed	4 hrs	Wed 04/11/15
13			Secure required resources	1 day	Thu 05/11/15
14			Analysis complete	0 days	Fri 06/11/15
15			**Requirement Specification Document Submission**	0 days	Fri 06/11/15
16			**Design**	**18.75 days**	**Fri 06/11/15**
17			Review preliminary software specifications	2 days	Fri 06/11/15
18			Develop functional specifications	5 days	Tue 10/11/15
19			Develop prototype based on functional specifications	4 days	Tue 17/11/15
20			Review functional specifications	2 days	Mon 23/11/15
21			Incorporate feedback into functional specifications	1 day	Wed 25/11/15
22			Obtain approval to proceed	4 hrs	Thu 26/11/15
23			Design complete	0 days	Fri 04/12/15
24			**Project Analysis & Design Document Submission**	0 days	Fri 04/12/15
25			**Development**	**107 days**	**Fri 04/12/15**
26			Review functional specifications	1 day	Fri 04/12/15
27			Identify design parameters	1 day	Mon 07/12/15
28			Develop code	80 days	Tue 08/12/15
29			Developer testing (primary debugging)	85 days	Tue 05/01/16
30			**Mid Point and Prototype Presentation**	0 days	Fri 29/04/16
31			Development complete	0 days	Fri 29/04/16
32			**Testing**	**122 days**	**Fri 04/12/15**
33			Review code	5 days	Mon 02/05/16
34			Develop test plans	4 days	Fri 04/12/15
35			Test component modules to specifications	2 days	Mon 09/05/16
36			Identify anomalies to specifications	3 days	Wed 11/05/16
37			Modify code	3 days	Mon 16/05/16
38			Re-test modified code	2 days	Thu 19/05/16
39			Testing complete	0 days	Fri 20/05/16
40			**Training**	**44 days**	**Tue 01/03/16**
41			Develop training specifications for end users	3 days	Tue 01/03/16
42			Develop training materials	3 wks	Fri 04/03/16
43			Finalize training materials	3 days	Fri 25/03/16
44			Develop training delivery mechanism	2 days	Wed 30/03/16
45			Training materials complete	0 days	Fri 29/04/16
46			**Documentation**	**62 days**	**Fri 04/12/15**
47			Develop Help specification	1 day	Fri 04/12/15
48			Develop Help system	3 wks	Tue 02/02/16
49			Review Help documentation	3 days	Mon 22/02/16
50			Incorporate Help documentation feedback	2 days	Thu 25/02/16
51			Develop user manuals specifications	2 days	Fri 04/12/15
52			Develop user manuals	3 wks	Tue 02/02/16
53			Review all user documentation	2 days	Mon 22/02/16
54			Incorporate user documentation feedback	2 days	Wed 24/02/16
55			Documentation complete	0 days	Fri 26/02/16
56			**Pilot**	**133.75 days**	**Fri 06/11/15**
57			Identify test group	1 day	Fri 06/11/15
58			Develop software delivery mechanism	1 day	Mon 09/11/15
59			Install/deploy software	1 day	Mon 02/05/16
60			Obtain user feedback	1 wk	Tue 03/05/16
61			Evaluate testing information	1 day	Tue 10/05/16
62			Pilot complete	0 days	Tue 10/05/16
63			**Deployment**	**4 days**	**Wed 11/05/16**
64			Determine final deployment strategy	1 day	Wed 11/05/16
65			Develop deployment methodology	1 day	Thu 12/05/16
66			Secure deployment resources	1 day	Fri 13/05/16
67			Deploy software	1 day	Mon 16/05/16
68			Deployment complete	0 days	Mon 16/05/16
69			**Post Implementation Review**	**2 days**	**Tue 17/05/16**
70			Document lessons learned	2 days	Tue 17/05/16
71			Post implementation review complete	0 days	Wed 18/05/16
72			**Project Final Documentation and Code Submission**	0 days	Wed 11/05/16
73			**Project Presentation + Project Showcase**	**18 days?**	**Fri 29/04/16**
74			Develop Project Presentation and Showcase presentation	11 days?	Fri 29/04/16
75			Project Presentation	0 days	Wed 18/05/16
76			Project Showcase	0 days	Wed 25/05/16

7.2. Monthly Journals

Appendix A. Reflective Journal September

Andrzej Stozek

X12118397

BScHons in Computing, part time

Month: September

My Achievements

This month, I worked on deciding my project idea.

I worked on project proposal

I have researched the resources required: API, programming and deploying platforms I may be using for delivering my project.

Intended Changes

Next month, I will work on the modifications of the project proposal, following my meeting with Mikhail Timofeev. Once finalised I will work on requirements specification and design analysis for my application.

I will research the APIs which I will use for my project. Need to do detail analysis of data returned by APIs and how I will use that data.

Supervisor Meetings

I was in contact with Mikhail Timofeev via email regarding my proposal.

I had a meeting with Mikhail Timofeev on Tuesday 6th of October to discuss the project idea. During the discussion few modifications were suggested. I am currently working on the specification of the “new” idea.

Appendix B. Reflective Journal-October

Andrzej Stozek

X12118397

BScHons in Computing, part time

Month: October

My Achievements

This month, I worked on gathering functionality of the application which will be developed.

I worked on the first version of the requirement specification document

I have gathered development tools which I will be using for the project.

I have setup GIT version control system which I will use during development process

I have started looking advanced programing with JavaScript, JQuery

Intended Changes

Next month, I will work on project Design and Analysis document. I will continue working on the requirements specification.

Still need to research the APIs which I will use for my project. Need to do detail analysis of data returned by APIs and how I will use that data.

I will continue working on improving my JavaScript programming skills

I will start coding a framework for the application, and communication with external APS.

Supervisor Meetings

I had a meeting with my supervisor, Mikhail Timofeev on Thursday 22nd of October to discuss the new project proposal and intended functionality of the application.

Appendix C. Reflective Journal-November

Andrzej Stozek

X12118397

BSc in Computing, part time

Month: November

My Achievements

This month, I researched APIs which are used as the core data source for the application:

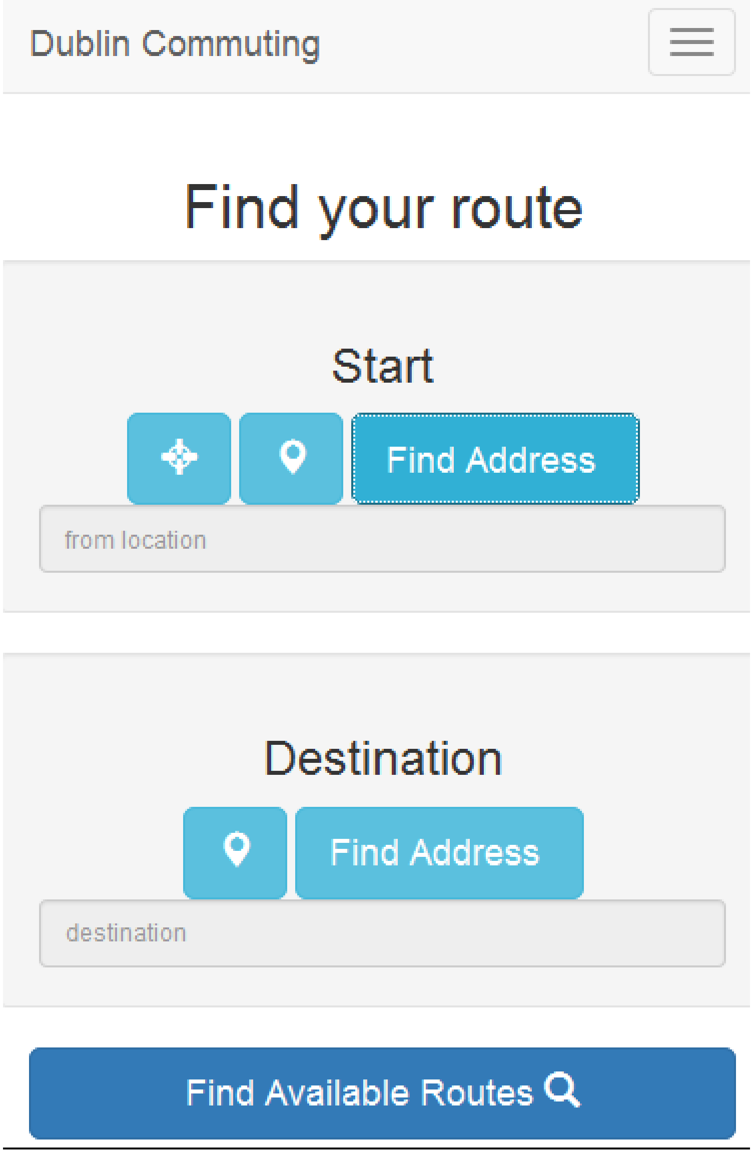
- RTPI API – real time passenger information API
- Dublin Bikes API,
- Irish Rail API,
- Google maps API,
- Google geocoding API,
- Google directions API.

I have now good understanding of the data that can be provided by the APIs. And I am positive that I can utilise the data in order to deliver timely information to the application user.

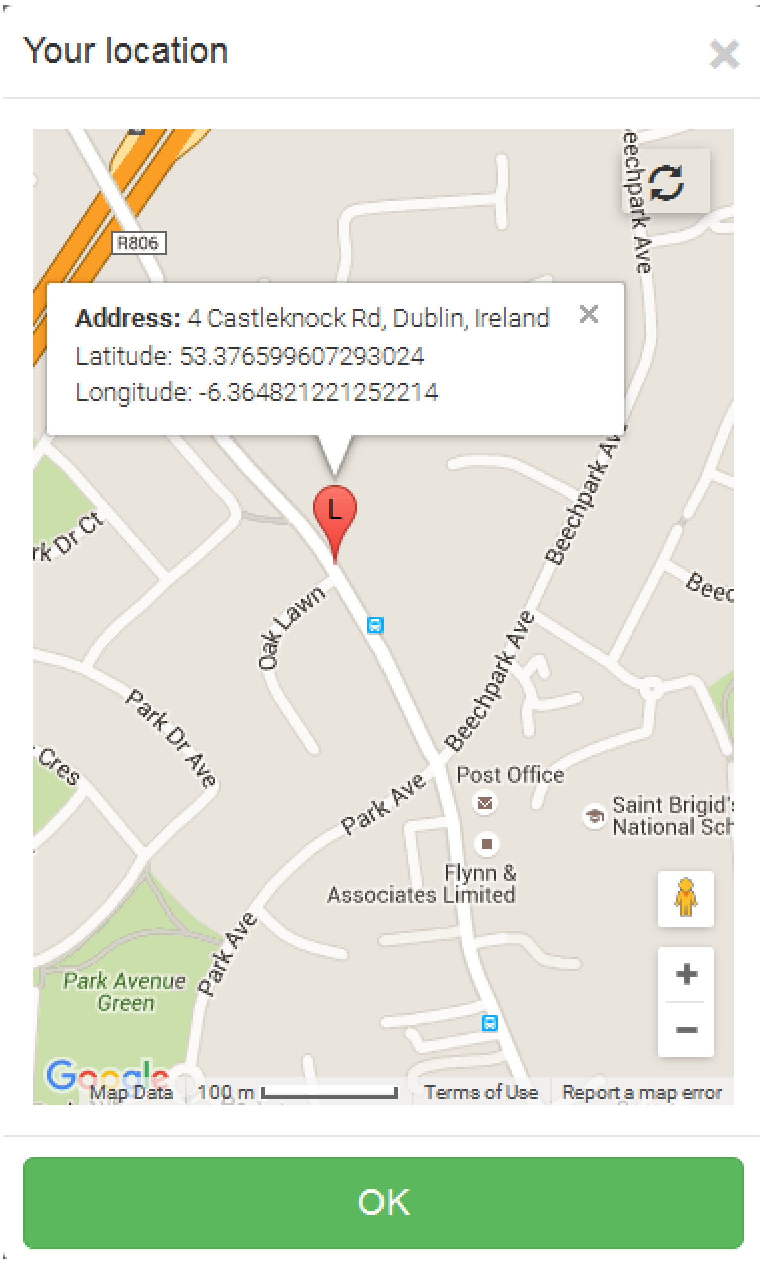
I have improved my JavaScript programming skills. And also gained more understanding of jQuery. I have also researched more on Bootstrap.

I have started coding the framework for the application, and communication with external APIs.

For now I have just built the main screen and the screens with google maps which are used for setting the start and end point of the route.



I have built the functionality to get the user's geolocation data and display on the map.

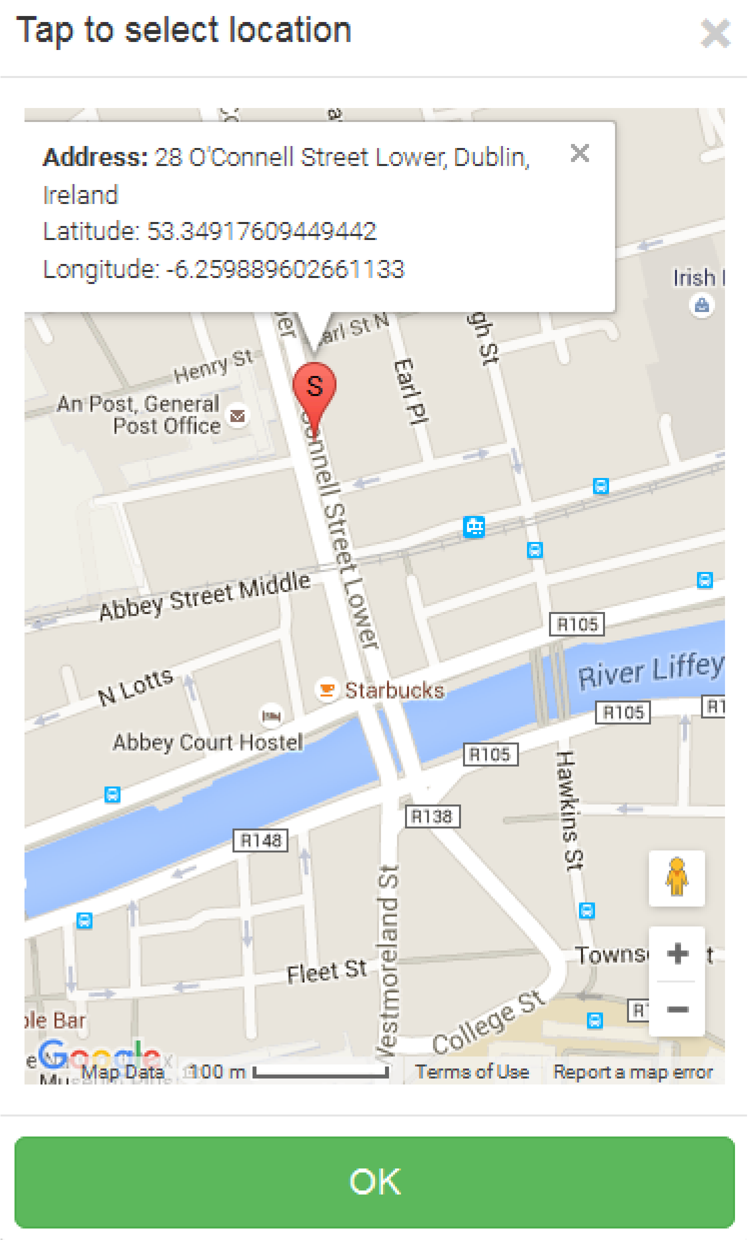


Your location

Address: 4 Castleknock Rd, Dublin, Ireland
Latitude: 53.376599607293024
Longitude: -6.364821221252214

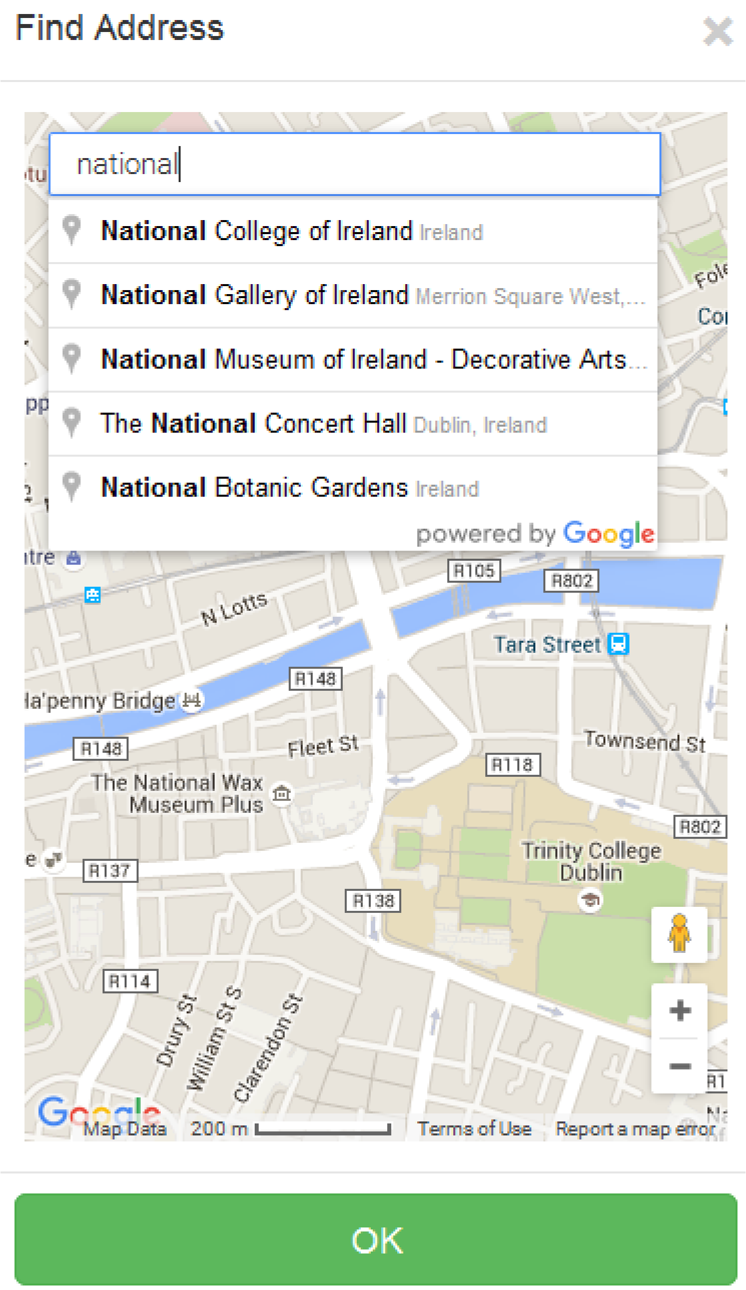
I have also added refresh button which user can use to request a refresh of details of the current geolocation.

Also I built the screen for selecting the point from the map by tap/click which also display the address and GPS coordinates.



Address: 28 O'Connell Street Lower, Dublin, Ireland
Latitude: 53.34917609449442
Longitude: -6.259889602661133

I have also built the screen where user can search for the destination/start point by providing address. This search functionality uses Google’s autocomplete service which give the user list of possible answers.

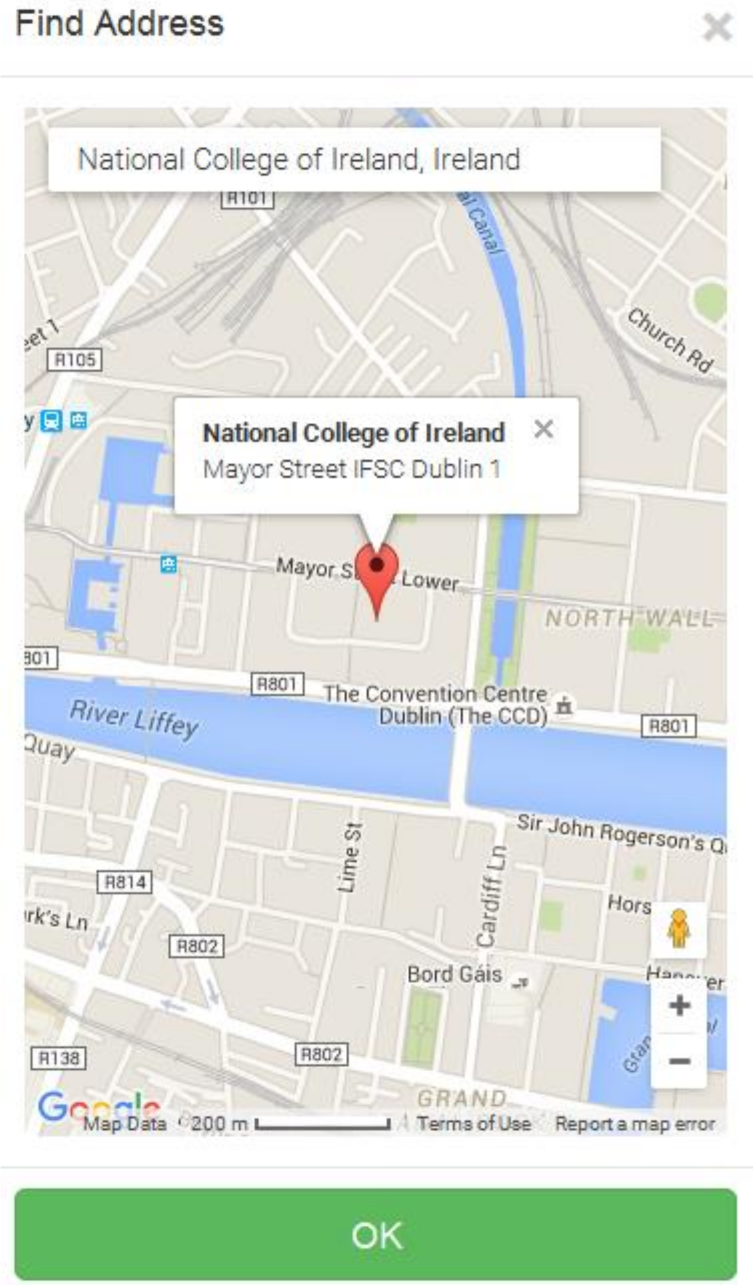


Find Address

national

- **National** College of Ireland Ireland
- **National** Gallery of Ireland Merrion Square West,...
- **National** Museum of Ireland - Decorative Arts...
- The **National** Concert Hall Dublin, Ireland
- **National** Botanic Gardens Ireland

powered by Google



Find Address

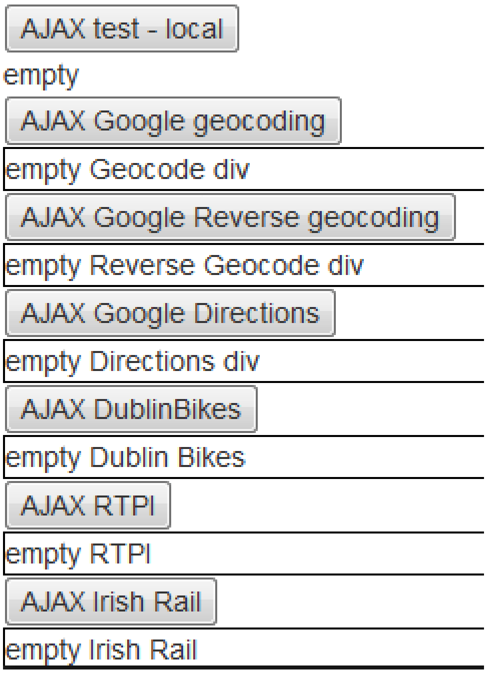
National College of Ireland, Ireland

National College of Ireland
Mayor Street IFSC Dublin 1

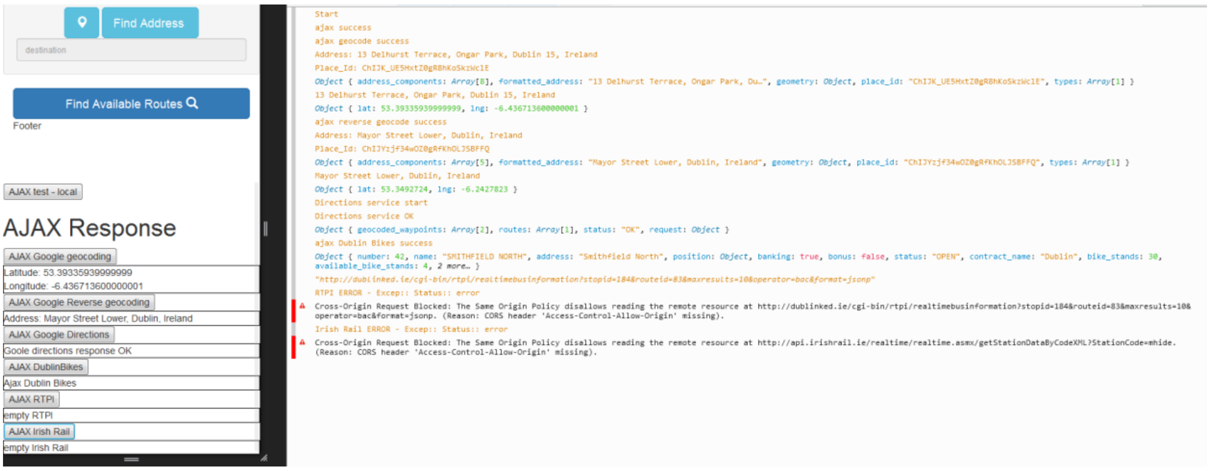
I added a drag and drop functionality to the map so user can easily adjust start/end point of the route.

I must say programing the google maps is quite challenging. I got some problems with multiple click events being triggered on one click, and markers/info-windows not being cleared. It took me a while to figure out solutions to these issues, but I have found the answers and now I am quite confident that I know how to add the functionality necessary to display route details on the map in the user friendly and intuitive way.

I have also implemented AJAX communication with all the APIs which are used by the application.



Below is the screenshot with AJAX/service responses.



Latitude: 53.39335939999999
Longitude: -6.436713600000001

Address: Mayor Street Lower, Dublin, Ireland

Goole directions response OK

Ajax Dublin Bikes

empty RTPI

empty Irish Rail

```
Start
ajax success
ajax geocode success
Address: 13 Delhurst Terrace, Ongar Park, Dublin 15, Ireland
Place_Id: ChIJK_UE5HxtZ0gR8hKoSkzWc1E
Object { address_components: Array[8], formatted_address: "13 Delhurst Terrace, Ongar Park, Du…", geometry: Object, place_id: "ChIJK_UE5HxtZ0gR8hKoSkzWc1E", types: Array[1] }
13 Delhurst Terrace, Ongar Park, Dublin 15, Ireland
Object { lat: 53.39335939999999, lng: -6.436713600000001 }
ajax reverse geocode success
Address: Mayor Street Lower, Dublin, Ireland
Place_Id: ChIJYzjf34wOZ0gRfKhOLJSBFFQ
Object { address_components: Array[5], formatted_address: "Mayor Street Lower, Dublin, Ireland", geometry: Object, place_id: "ChIJYzjf34wOZ0gRfKhOLJSBFFQ", types: Array[1] }
Mayor Street Lower, Dublin, Ireland
Object { lat: 53.3492724, lng: -6.2427823 }
Directions service start
Directions service OK
Object { geocoded_waypoints: Array[2], routes: Array[1], status: "OK", request: Object }
ajax Dublin Bikes success
Object { number: 42, name: "SMITHFIELD NORTH", address: "Smithfield North", position: Object, banking: true, bonus: false, status: "OPEN", contract_name: "Dublin", bike_stands: 30,
available_bike_stands: 4, 2 more… }
"http://dublinked.ie/cgi-bin/rtpi/realtimebusinformation?stopid=184&routeid=83&maxresults=10&operator=bac&format=jsonp"
RTPI ERROR - Excep:: Status:: error
Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at http://dublinked.ie/cgi-bin/rtpi/realtimebusinformation?stopid=184&routeid=83&maxresults=10&
operator=bac&format=jsonp. (Reason: CORS header 'Access-Control-Allow-Origin' missing).
Irish Rail ERROR - Excep:: Status:: error
Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at http://api.irishrail.ie/realtime/realtime.asmx/getStationDataByCodeXML?StationCode=mhide.
(Reason: CORS header 'Access-Control-Allow-Origin' missing).
```

I came across number of challenges and problems with accessing the APIs data due to the way the application is intended to be designed (client side).

Because all AJAX calls are made by the client side, the issue of CORS (cross origin resource sharing) has to be addressed.

I was able to make successful AJAX calls from the client side to the Dublin Bikes API. Google Geocoding and Google directions don't allow CORS requests but fortunately Google Maps API comes with the services which provide the same functionality, so I can use Google Geocoding service and Google Directions service in order to get all necessary data.

I was not able to make a successful CORS request to the RTPI API. I have research quite a lot possible solutions, and tested many implementations, but with no success due to missing header in server response:

Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at http://dublinked.ie/cgi-bin/rtpi/realtimebusinformation?stopid=184&routeid=83&maxresults=10&operator=bac&format=jsonp. (Reason: CORS header 'Access-Control-Allow-Origin' missing).

I have also tired json-p requests to RTPI server, but server doesn't respond with "json with padding" so this didn't work as well.

I have encountered similar problems with accessing Irish Rail API.

After additional research of CORS and possible solutions, the conclusion is that it may not be possible to successfully make CORS AJAX calls from the client side if the destination server is not configured to respond with required headers.

I will build a PhoneGap mobile App to test if CORS AJAX request are possible and accepted by mobile devices. If this works then I can continue with implementing just "client side application". If not will have to introduce a back end server which will make necessary AJAX calls.

Intended Changes

I will need to work more on updating project Requirements Specification and on developing the Design and Analysis document, on which I didn't get a chance to do much work during last month.

I will build a PhoneGap mobile App to test CORS AJAX on the mobile device.

I will add the backend AJAX request scripts if I will not be successful with making all CORS AJAX calls from the client side.

I will continue coding the application prototype:

- Add directions request,
- Display possible results on the UI,
- Add algorithm to introduce “Dublin Bikes” into possible route options.

Supervisor Meetings

I have contacted Mikhail Timofeev via email on the 24th of November to report current development progress and the challenges with the CORS (cross origin resource sharing) AJAX communication with the APIs.

I had a meeting with my supervisor, Mikhail Timofeev on Tuesday 01st of December to discuss the CORS AJAX request issues.

I have mentioned a possibility of introduction of a backend server for the application.

I have presented already implemented functionality of the application and the plan for next few weeks.

Appendix D. Reflective Journal - December

Andrzej Stozek

X12118397

BScHons in Computing, part time

Month: December

My Achievements

Due to family, work and other college commitments I have not made much progress on the application development.

I have done some testing of the already implemented functionality and fixed some small issues.

Intended Changes

I will build a PhoneGap mobile App to test CORS AJAX on the mobile device.

I will add the backend AJAX request scripts if I will not be successful with making all CORS AJAX calls from the client side.

I will continue coding the application prototype:

- Add directions request,
- Display possible results on the UI,
- Add algorithm to introduce “Dublin Bikes” into possible route options.

Supervisor Meetings

Appendix E. Reflective Journal - January

Andrzej Stozek

X12118397

BScHons in Computing, part time

Month: January

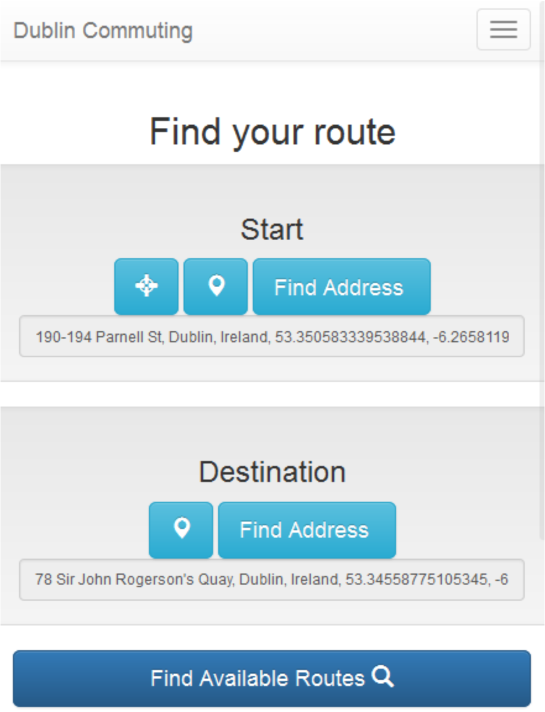
My Achievements

This month I worked on application code in order to implement as much as possible functionality for the mid-point presentation.

I have also worked on preparation of the presentation and the mid-point technical report.

I have Implemented “Find Routes” functionality for selected start / end points which when successful displays the results on the google maps, each route option is displayed on separate map for ease of read.

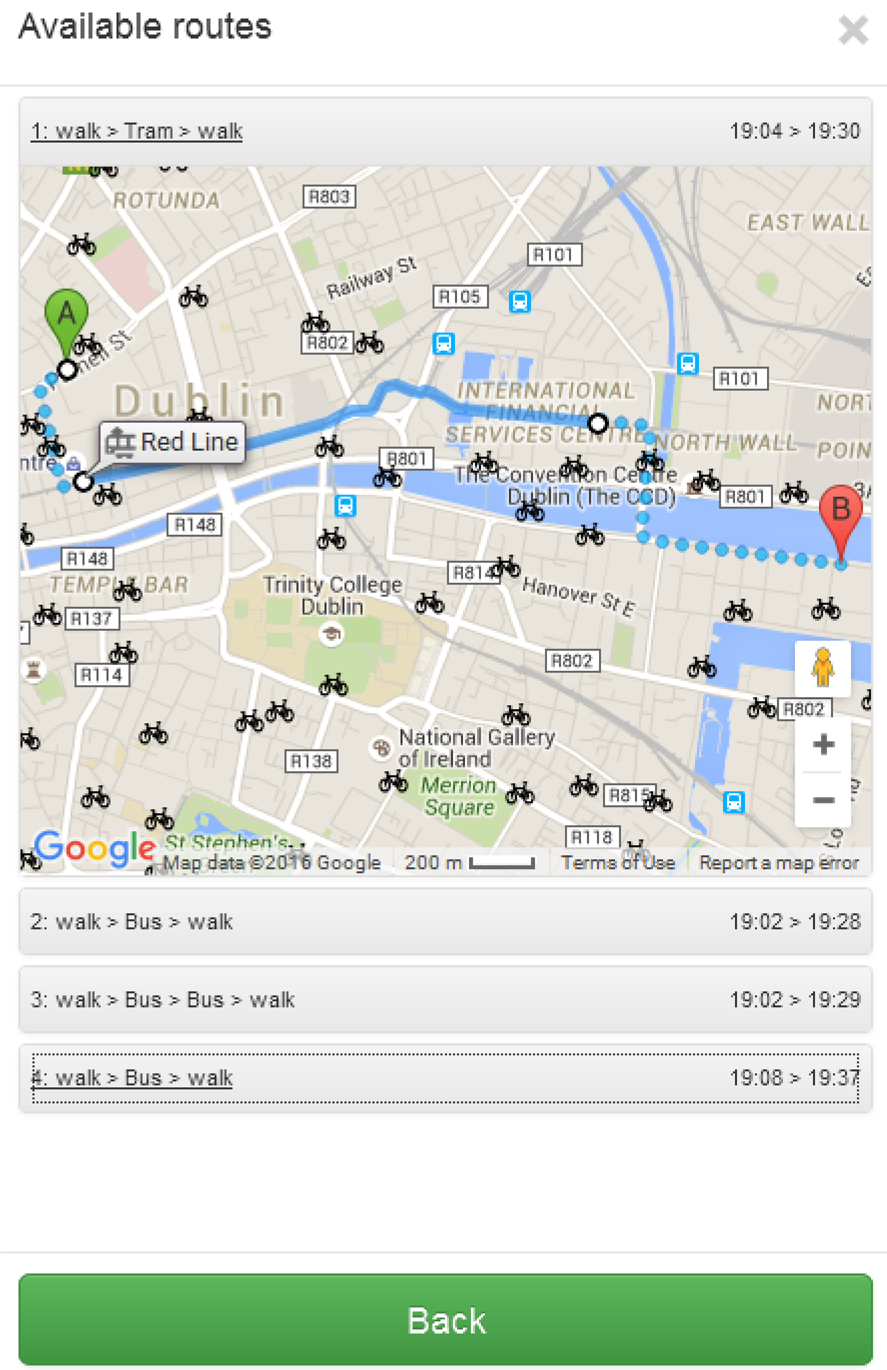
Start and end point successfully selected:



190-194 Parnell St, Dublin, Ireland, 53.350583339538844, -6.2658119

78 Sir John Rogerson's Quay, Dublin, Ireland, 53.34558775105345, -6

Once found – routes are displayed on the map



Available routes

1: walk > Tram > walk 19:04 > 19:30

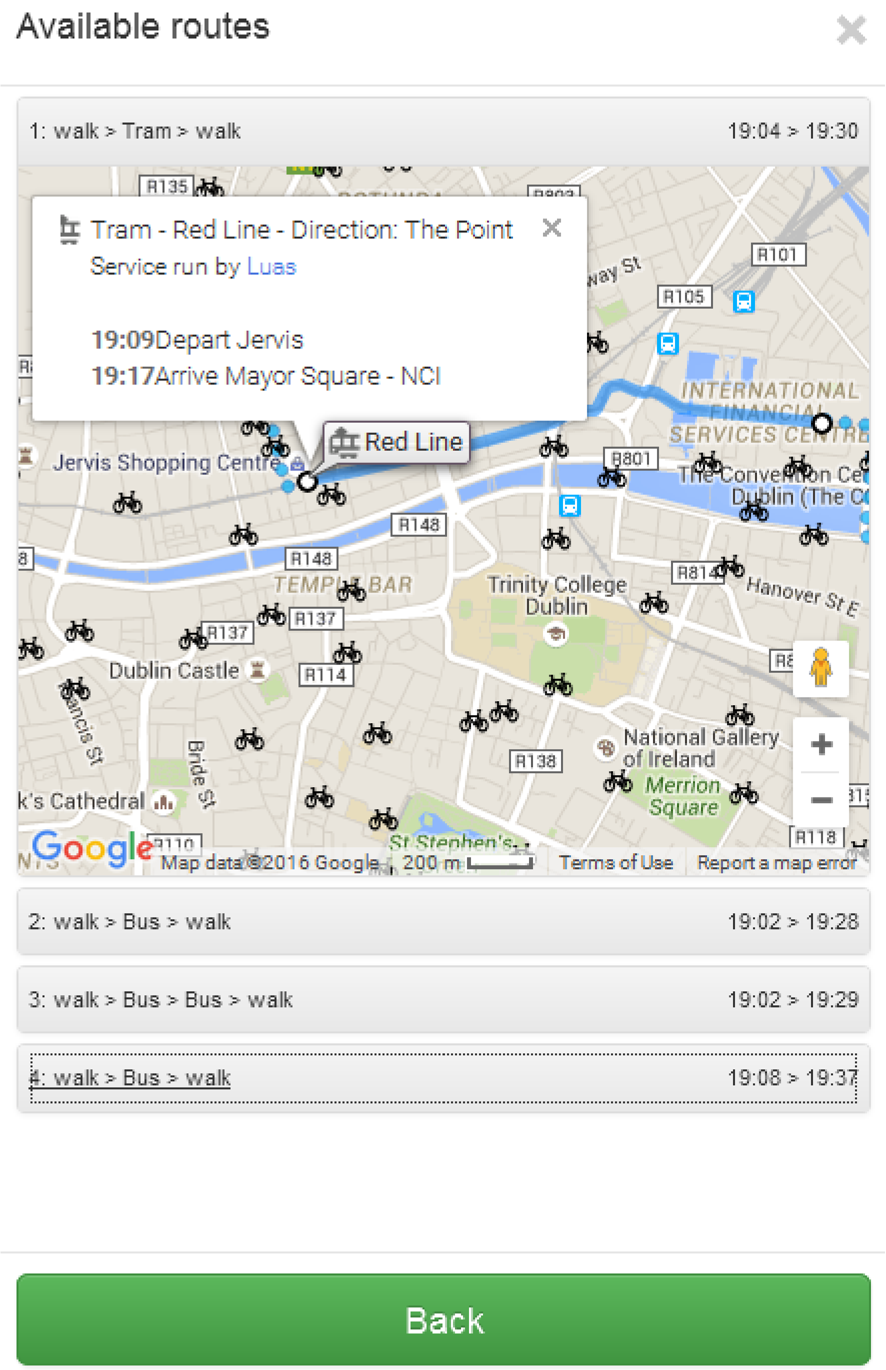
2: walk > Bus > walk 19:02 > 19:28

3: walk > Bus > Bus > walk 19:02 > 19:29

4: walk > Bus > walk 19:08 > 19:37

Back

Click on the points on the route provides additional routing information



Available routes

1: walk > Tram > walk 19:04 > 19:30

Tram - Red Line - Direction: The Point

Service run by Luas

19:09Depart Jervis

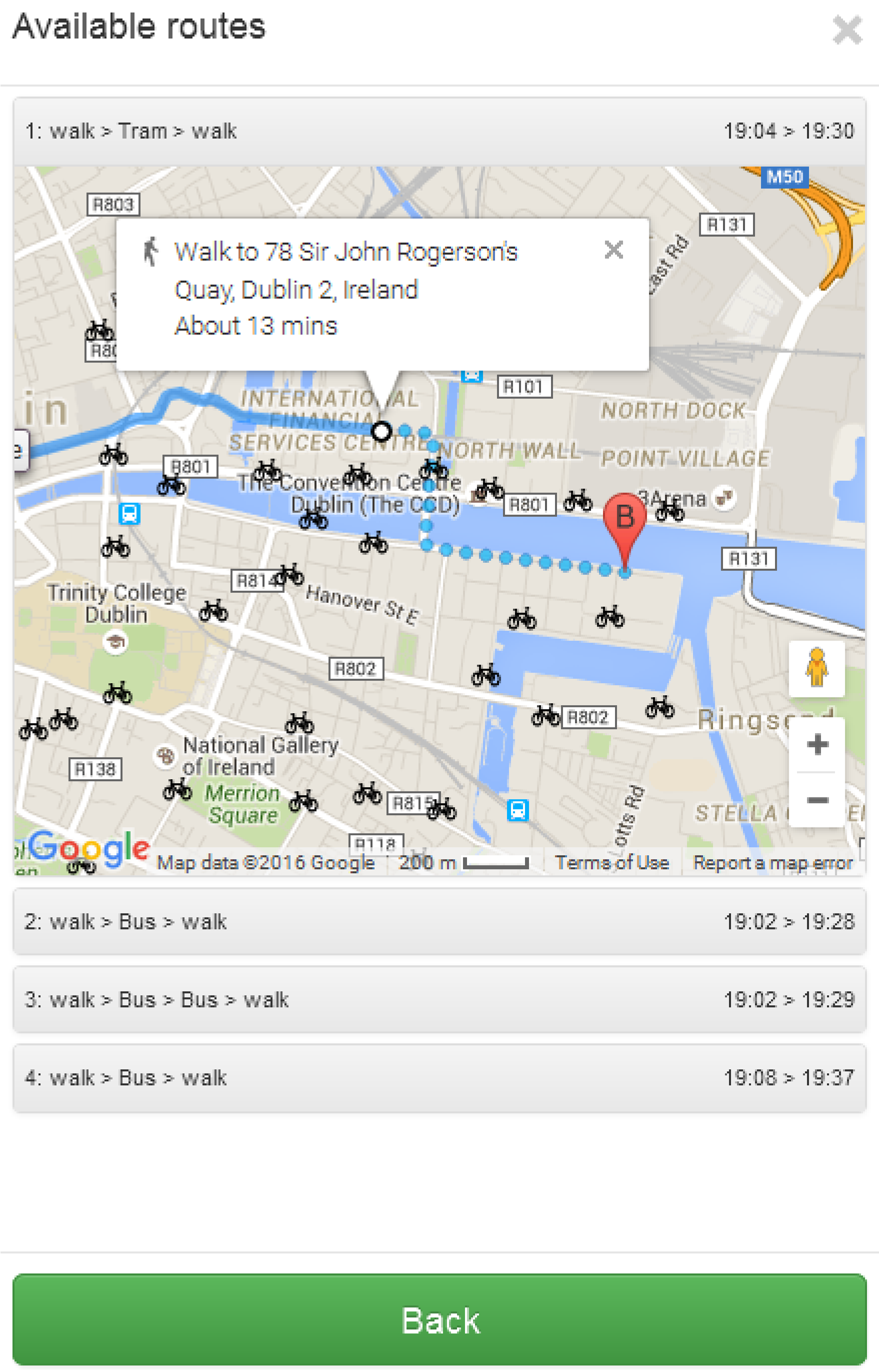
19:17Arrive Mayor Square - NCI

2: walk > Bus > walk 19:02 > 19:28

3: walk > Bus > Bus > walk 19:02 > 19:29

4: walk > Bus > walk 19:08 > 19:37

Back



Available routes

1: walk > Tram > walk 19:04 > 19:30

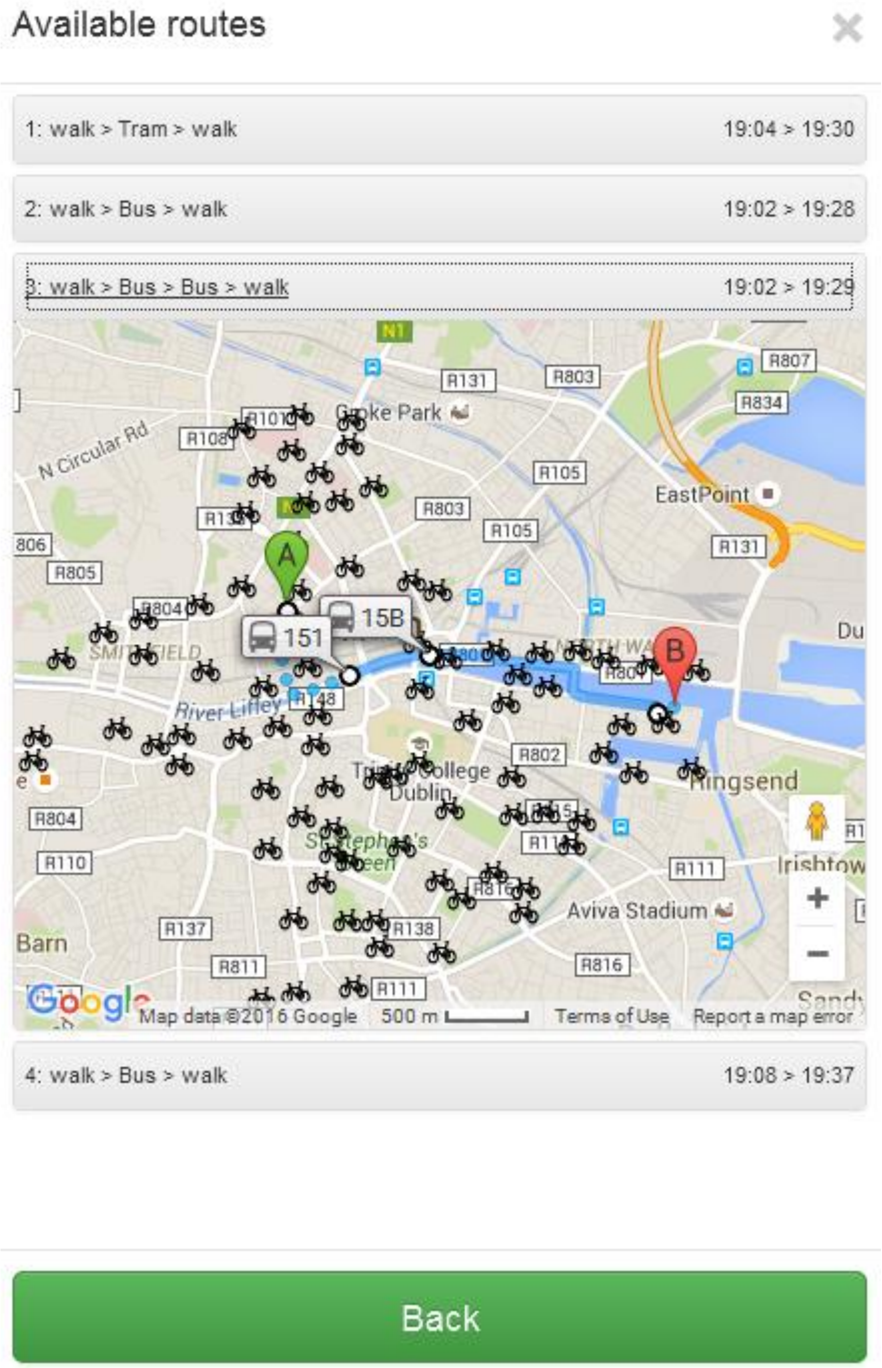
Walk to 78 Sir John Rogerson's Quay, Dublin 2, Ireland
About 13 mins

2: walk > Bus > walk 19:02 > 19:28

3: walk > Bus > Bus > walk 19:02 > 19:29

4: walk > Bus > walk 19:08 > 19:37

User can switch between route options by tapping on the route title bar



Available routes

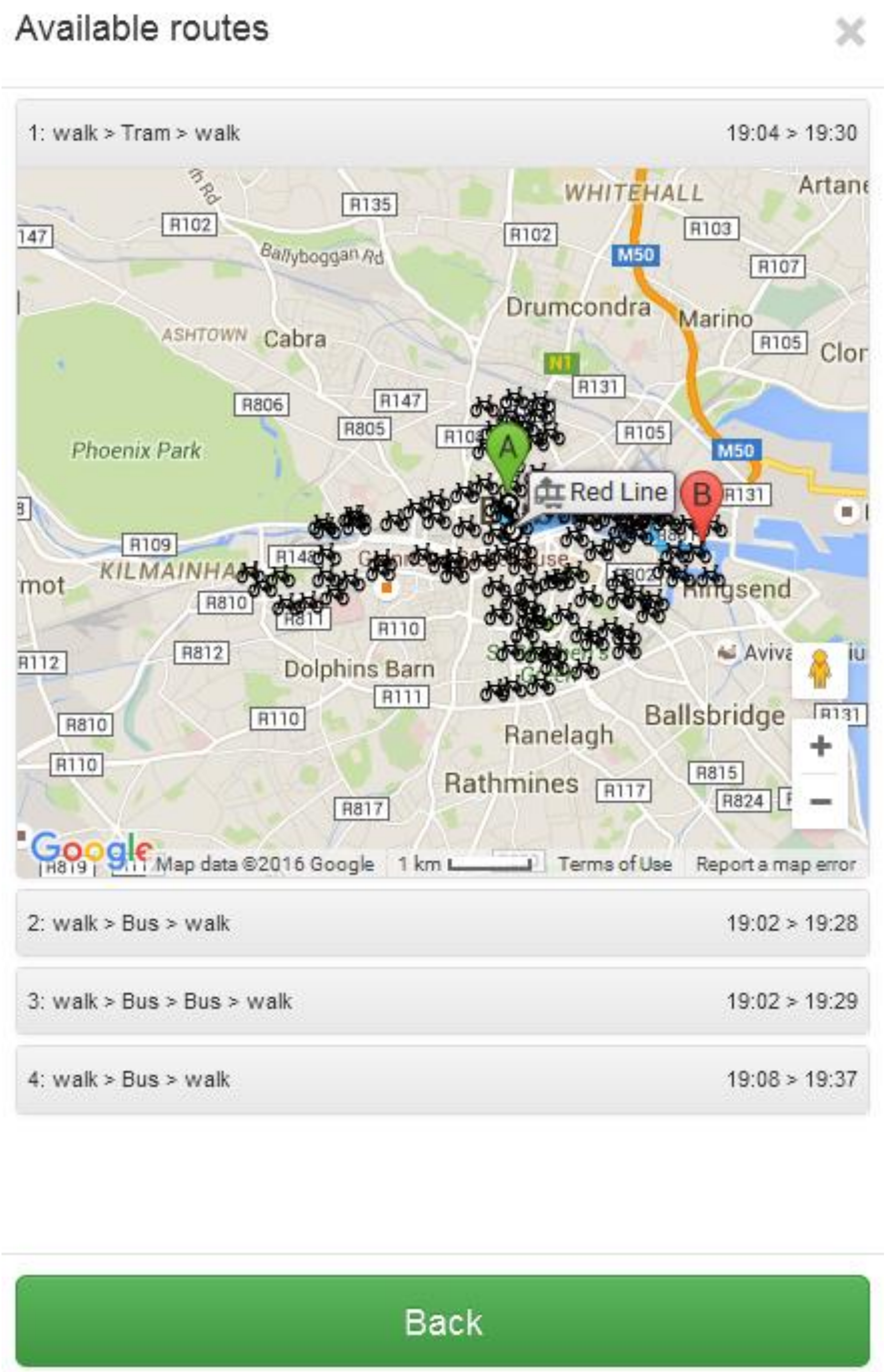
1: walk > Tram > walk 19:04 > 19:30

2: walk > Bus > walk 19:02 > 19:28

3: walk > Bus > Bus > walk 19:02 > 19:29

4: walk > Bus > walk 19:08 > 19:37

I have also implemented display of all Dublin Bike stations on the map



Available routes

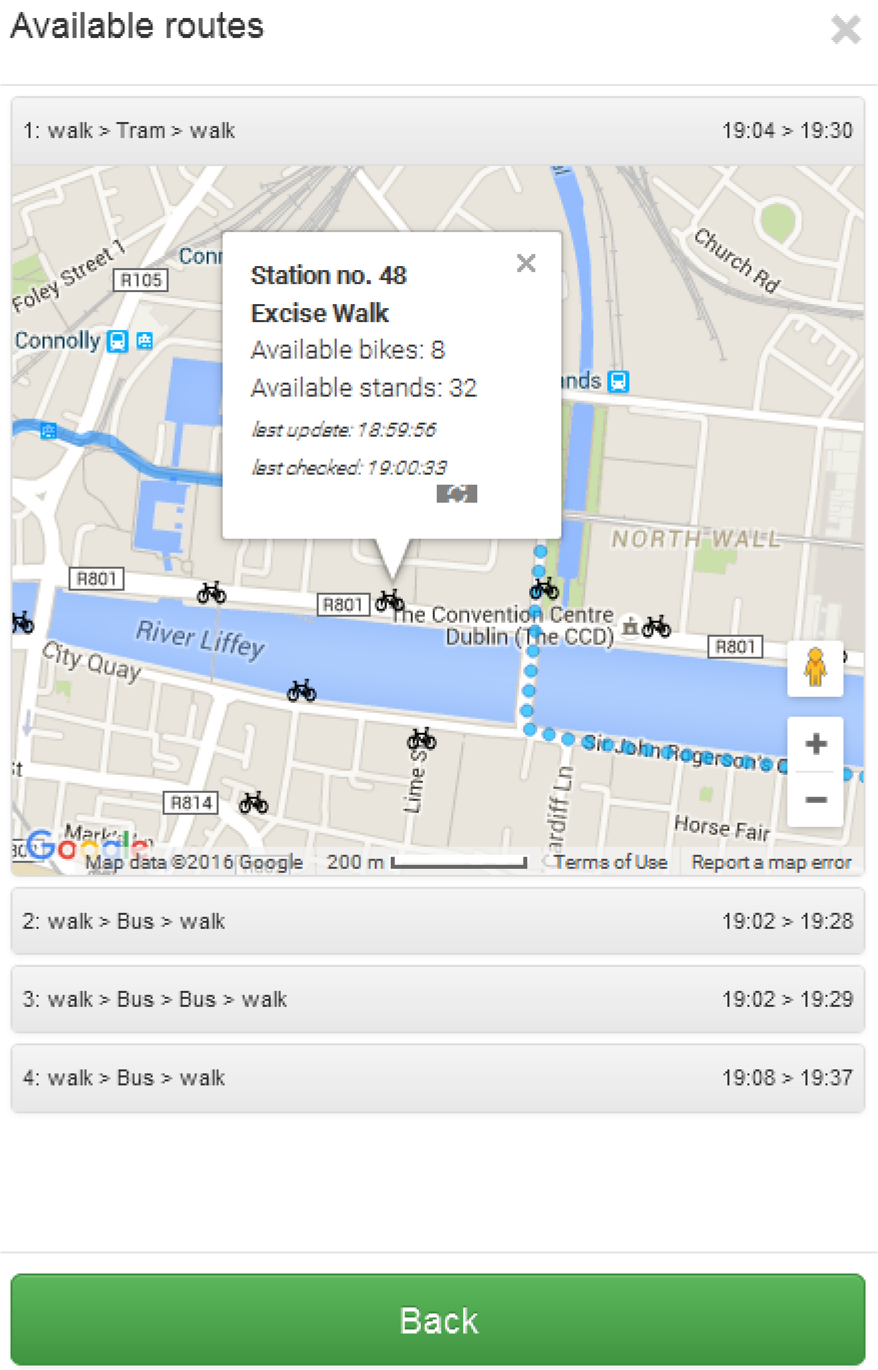
1: walk > Tram > walk 19:04 > 19:30

2: walk > Bus > walk 19:02 > 19:28

3: walk > Bus > Bus > walk 19:02 > 19:29

4: walk > Bus > walk 19:08 > 19:37

User can click on each bicycle representing a bike station to view details and current situation at the bike station (station name and number plus available bikes and available bike stands and well as the last update and last checked time)



Available routes ×

1: walk > Tram > walk 19:04 > 19:30

Station no. 48
Excise Walk
Available bikes: 8
Available stands: 32
last update: 18:59:56
last checked: 19:00:33

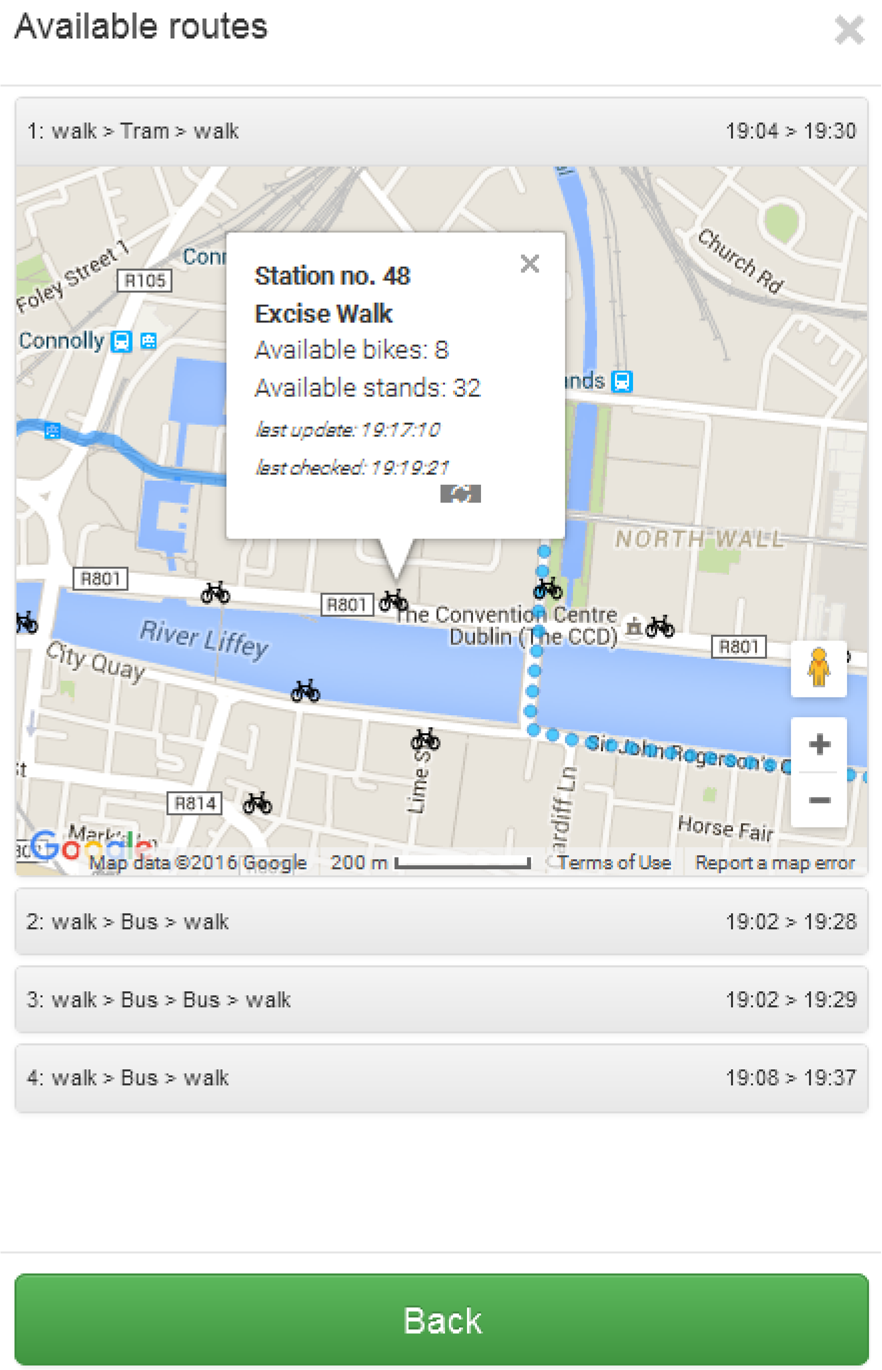
2: walk > Bus > walk 19:02 > 19:28

3: walk > Bus > Bus > walk 19:02 > 19:29

4: walk > Bus > walk 19:08 > 19:37

Back

I also implemented and AJAX call to Dublin Bikes API in order to update station details



Available routes ×

1: walk > Tram > walk 19:04 > 19:30

Station no. 48

Excise Walk

Available bikes: 8

Available stands: 32

last update: 19:17:10

last checked: 19:19:21

2: walk > Bus > walk 19:02 > 19:28

3: walk > Bus > Bus > walk 19:02 > 19:29

4: walk > Bus > walk 19:08 > 19:37

Back

Intended Changes

I will build a PhoneGap mobile App to test CORS AJAX on the mobile device.

I will add the backend AJAX request scripts if I will not be successful with making all CORS AJAX calls from the client side.

I will continue coding the application prototype:

- Develop and Implementing algorithm for including Dublin Bikes on route options,
- Display additional route options with Dublin bikes,

Supervisor Meetings

My supervisor was changed.

I introduced myself to my new supervisor Dr Ralf Bierig at a brief meeting on Tuesday 26th. I was informed that contact plan/schedule will be developed in the next few weeks.

Appendix F. Reflective Journal - February

Reflective Journal

Andrzej Stozek

X12118397

BScHons in Computing, part time

Month: February

My Achievements

Due to the busy semester I didn’t get much work done on the project

Intended Changes

Do some implementation of the application

Supervisor Meetings

I had a meeting with my supervisor on Friday the 4th of March.

I explained that the semester modules are very challenging and time consuming I I didn’t get much work done on the project since the mid-point presentation. And the situation will probably be similar until the time after the module exams.

Appendix G. Reflective Journal - March

Reflective Journal

Andrzej Stozek

X12118397

BScHons in Computing, part time

Month: March

My Achievements

I created the PhoneGap App.

I am currently in the process of testing the App on the phone fixing the encountered errors and optimising the GUI for mobile device.

Intended Changes

Focus all the efforts on the project as soon as possible

Supervisor Meetings

7.3. Other Material Used

7.3.1. Irish Rail Realtime API information

Irish Rail Realtime API's information page

This information is an estimate of train times based on:

- The current location of train services from Iarnród Éireann's central signalling system.
- The scheduled journey times from areas under local signalling control.
- Trains indicated as being late can make up time and arrive on time. Ensure you give yourself plenty of time to catch your train.

Our central signalling system is subject to ongoing work to support this real-time facility. However real-time information has weaker coverage in certain areas, these include:

- Hazelhatch - Heuston Line
- Athlone - Westport/Ballina Line
- Cork Station
- Cork - Cobh/Midleton Line
- Mallow - Tralee Line
- Ballybrophy - Limerick Line
- Limerick - Ennis Line
- Athy - Waterford Line
- Limerick Junction - Waterford Line
- Greystones - Rosslare Line
- Drogheda - Belfast Line
- Drumcondra - Sligo Line

In these cases your query will return the scheduled time only.

The following functions are available

1. Get All Stations - usage http://api.irishrail.ie/realtime/realtime.asmx/getAllStationsXML returns a list of all stations with StationDesc, StaionCode, StationId, StationAlias, StationLatitude and StationLongitude ordered by Latitude, Longitude

2. Get All Stations with Type http://api.irishrail.ie/realtime/realtime.asmx/getAllStationsXML_WithStationType?Statio

nType=D returns a list of all stations with StationDesc, StaionCode, StationId, StationAlias, StationLatitude and StationLongitude ordered by Latitude, Longitude filtered by StationType - takes a single letter with 4 possible values for the StationType parameter (A for All, M for Mainline, S for suburban and D for DART) any other value will be changed to A

3. Get Current Trains -
usage http://api.irishrail.ie/realtime/realtime.asmx/getCurrentTrainsXML returns a listing of 'running trains' ie trains that are between origin and destination or are due to start within 10 minutes of the query time. Returns TrainStatus, TrainLatitude, TrainLongitude, TrainCode, TrainDate, PublicMessage and Direction

TrainStatus = N for not yet running or R for running

TrainCode is Irish Rail's unique code for an individual train service on a date

Direction is either Northbound or Southbound for trains between Dundalk and Rosslare and between Sligo and Dublin. for all other trains the direction is to the destination eg. To Limerick

Public Message is the latest information on the train uses \n for a line break
eg AA509\n11:00 - Waterford to Dublin Heuston (0 mins late)\nDeparted Waterford next stop Thomastown

4. Get Current Trains with Type
usage http://api.irishrail.ie/realtime/realtime.asmx/getCurrentTrainsXML_WithTrainType?TrainType=D returns a listing of 'running trains' ie trains that are between origin and destination or are due to start within 10 minutes of the query time. Returns TrainStatus, TrainLatitude, TrainLongitude, TrainCode, TrainDate, PublicMessage and Direction filtered by traintype, takes a single letter with 4 possible values for the StationType parameter (A for All, M for Mainline, S for suburban and D for DART) any other value will be changed to A

5. Get Station Data By Name
usage http://api.irishrail.ie/realtime/realtime.asmx/getStationDataByNameXML?StationDesc=Bayside returns all trains due to serve the named station in the next 90 minutes

6. Get Station Data By Name with Number of Minutes
usage http://api.irishrail.ie/realtime/realtime.asmx/getStationDataByNameXML?StationDesc=Bayside&NumMins=20 returns all trains due to serve the named station in the next x minutes (x must be between 5 and 90)

7. Get Station Data by StationCode
usage http://api.irishrail.ie/realtime/realtime.asmx/getStationDataByCodeXML?StationCode=mhide returns all trains due to serve the named station in the next 90 minutes

8. Get Station Data by StationCode with number of minutes usage http://api.irishrail.ie/realtime/realtime.asmx/getStationDataByCodeXML_WithNumMins?StationCode=mhide&NumMins=20 returns all trains due to serve the named station in the next x minutes (x must be between 5 and 90)

The above four queries all return the same data as follows

ServerTime - the time on the server

TrainCode - Unique Id for the train

StationFullName - Long version of Station Name (identical in every record)

StationCode - 4 to 5 letter station abbreviation

QueryTime - The time the query was made

TrainDate - The date the service started its journey (some trains run over midnight)

Origin

Destination

OriginTime - The time the train left (or will leave) its origin

DestinationTime - the scheduled time at its destination

Status - Latest information on this service

LastLocation (Arrived|Departed StationName)

DueIn - Num of minutes till the train will arrive here

Late - Minutes late

ExpArrival - the trains expected arrival time at the query station updated as the train progresses (note will show 00:00 for trains starting from query station)

ExpDepart - the trains expected departure time at the query station updated as the train progresses (note will show 00:00 for trains terminating at query station)

SchArrival - the scheduled arrival time (note will show 00:00 for trains starting from query station)

SchDepart - the scheduled depart time (note will show 00:00 for trains terminating at query station)

Direction - Northbound, Southbound or To Destination

Train Type - DART - Intercity etc.

LocationType - O = Origin, D = Destination, S= Stop

9. Get Stations Filter

usage http://api.irishrail.ie/realtime/realtime.asmx/getStationsFilterXML?StationText=br returns a list of station names that contain the StationText

10. Get Train Movements

usage http://api.irishrail.ie/realtime/realtime.asmx/getTrainMovementsXML?TrainId=e109&TrainDate=21 dec 2011 returns all stop information for the given train as follows

TrainCode

TrainDate

LocationCode

LocationFullName

LocationOrder

LocationType O= Origin, S= Stop, T= TimingPoint (non stopping location) D = Destination

TrainOrigin

TrainDestination

ScheduledArrival

ScheduledDeparture

Arrival (actual)

Departure (actual)

AutoArrival (was information automatically generated)

AutoDepart

StopType C= Current N = Next

Please note all these webservice names and parameters are case sensitive

Irish Rail provides this information as is and do not offer any support

7.3.2. Dublin Bikes API data

Dublin Bikes API data

Getting started

Two kinds of data are delivered by the platform:

- Static data provides stable information like station position, number of bike stands, payment terminal availability, etc.
- Dynamic data provides station state, number of available bikes, number of free bike stands, etc.

Static data can be downloaded manually in file format or accessed through the API. Dynamic data are refreshed every minute and can be accessed only through the API.

Usage of the API key

Static data doesn't require an API key. For equal accessibility to dynamic data, you must use a personal API key.

Your key

If you don't have an API key yet, you can sign up for free and request a key from your accountpage.

How to use your key

Once you have a key, you just have to add it as a parameter named « apiKey » to all your requests to the dynamic API.

GET https://api.jcdecaux.com/vls/v1/stations?contract={contract_name}&apiKey={api_key}

If no key is specified, you'll get a `403 Forbidden` error.

My call:

https://api.jcdecaux.com/vls/v1/stations?contract=Dublin&apiKey=437cbd7f9593c3fd981e7634513bef5e66e04572

Data format

In accordance with RFC 4627, results in JSON format are UTF-8 encoded.

Real-time data

Station and contract resources

Stations

Stations are represented as follows:

```
{
  "number": 123,
  "contract_name" : "Paris",
  "name": "stations name",
  "address": "address of the station",
  "position": {
    "lat": 48.862993,
    "lng": 2.344294
  },
  "banking": true,
  "bonus": false,
  "status": "OPEN",
  "bike_stands": 20,
  "available_bike_stands": 15,
  "available_bikes": 5,
  "last_update": <timestamp>
}
```

We can split this representation in two parts: static data and dynamic data.

Static data

- `number` number of the station. This is NOT an id, thus it is unique only inside a contract.
- `contract_name` name of the contract of the station
- `name` name of the station

- `address` address of the station. As it is raw data, sometimes it will be more of a comment than an address.
- `position` position of the station in WGS84 format
- `banking` indicates whether this station has a payment terminal
- `bonus` indicates whether this is a bonus station

Dynamic data

- `status` indicates whether this station is `CLOSED` or `OPEN`
- `bike_stands` the number of operational bike stands at this station
- `available_bike_stands` the number of available bike stands at this station
- `available_bikes` the number of available and operational bikes at this station
- `last_update` timestamp indicating the last update time in milliseconds since Epoch

Contracts

For each client agglomeration of JCDecaux, there's an associated contract:

```
{
  "name" : "Paris",
  "commercial_name" : "Vélib'",
  "country_code" : "FR",
  "cities" : [
    "Paris",
    "Neuilly",
    ...
  ]
}
```

- `name` is the identifier of the contract
- `commercial_name` is the commercial name of the contract, the one users usually know
- `country_code` is the code (ISO 3166) of the country
- `cities` the cities that are concerned by this contract

REST API Description

Each response of the API depends on the request received. We tried to make them the more RESTful possible. If the server encounters an unexcepted error while processing a request, the response will be `HTTP/1.1 500 Internal Server Error`.

Do not forget to add your API key to the following requests (See explanations).

Get the contract list

Request:

GET https://api.jcdecaux.com/vls/v1/contracts HTTP/1.1

Accept: application/json

My call to get contracts:

https://api.jcdecaux.com/vls/v1/contracts?apiKey=437cbd7f9593c3fd981e7634513bef5e66e04572

Response:

HTTP/1.1 200 OK

Content-Type: application/json

[contract_json,contract_json,...]

Get station information

Request:

GET https://api.jcdecaux.com/vls/v1/stations/{station_number}?contract={contract_name} HTTP/1.1

Accept: application/json

My call example:

https://api.jcdecaux.com/vls/v1/stations/42?contract=Dublin&apiKey=437cbd7f9593c3fd981e7634513bef5e66e04572

Response when station exists:

```
HTTP/1.1 200 OK
Content-Type: application/json

station_json
```

Response when not:

```
HTTP/1.1 404 Not Found
```

Get the station list

Request:

```
GET https://api.jcdecaux.com/vls/v1/stations HTTP/1.1
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[station_json, station_json, ...]
```

Get the stations of a contract

Request:

```
GET https://api.jcdecaux.com/vls/v1/stations?contract={contract_name} HTTP/1.1
```

My call:

https://api.jcdecaux.com/vls/v1/stations?contract=Dublin&apiKey=437cbd7f9593c3fd981e7634513bef5e66e04572

```
Accept: application/json
```

Response when contract exists:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
[station_json, station_json, ...]
```

Response when not:

```
HTTP/1.1 400 Bad Request
```

7.3.1.RTPI REST API specification

OPEN SKY
Data Systems

RTPI REST Web Services

Specification