1. COMP47360 Home . . . 3
1.1 Product Specification . . . 3
1.2 Product Requirements . . . 3
1.3 Product Roadmap . . . 7
1.4 Project Management . . . 8
1.4.1 Team Member Contact List . . . 8
1.4.2 Requirements - Discovery Phase . . . 8
1.4.3 Sprints . . . 8
1.4.3.1 Sprint 1 . . . 8
1.4.3.1.1 Sprint 1 - Pre-Planning . . . 8
1.4.3.1.2 Sprint 1 - Burndown Chart . . . 11
1.4.3.1.3 Sprint 1 - Retrospective . . . 11
1.4.3.2 Sprint 2 . . . 11
1.4.3.2.1 Sprint 2 - Pre-Planning . . . 11
1.4.3.2.2 Sprint 2 - Burndown Chart . . . 13
1.4.3.2.3 Sprint 2 - Retrospective . . . 13
1.4.3.3 Sprint 3 . . . 14
1.4.3.3.1 Sprint 3 - Pre-Planning . . . 14
1.4.3.3.2 Sprint 3 - Burndown Chart . . . 15
1.4.3.3.3 Sprint 3 - Retrospective . . . 15
1.4.3.4 Sprint 4 . . . 16
1.4.3.4.1 Sprint 4 - Pre-Planning . . . 16
1.4.3.4.2 Sprint 4 - Burndown Chart . . . 17
1.4.3.4.3 Sprint 4 - Retrospective . . . 17
1.4.3.5 Sprint 5 . . . 17
1.4.3.5.1 Sprint 5 - Pre-Planning . . . 17
1.4.3.5.2 Sprint 5 - Burndown Chart . . . 18
1.4.3.5.3 Sprint 5 - Retrospective . . . 18
1.4.4 Meetings . . . 19
1.4.4.1 Daily Standups . . . 19
1.4.4.1.1 Daily Standups - WK01 . . . 19
1.4.4.1.2 Daily Standups - WK02 . . . 20
1.4.4.1.3 Daily Standups - WK03 . . . 21
1.4.4.1.4 Daily Standups - WK04 . . . 23
1.4.4.1.5 Daily Standups - WK05 . . . 24
1.4.4.1.6 Daily Standups - WK07 . . . 26
1.4.4.1.7 Daily Standups - WK08 . . . 27
1.4.4.1.8 Daily Standups - WK09 . . . 29
1.4.4.1.9 Daily Standups - WK10 . . . 30
1.4.4.1.10 Daily Standups - WK11 . . . 31
1.4.4.1.11 Daily Standups - WK12 . . . 33
1.4.4.2 Workshops . . . 34
1.4.4.2.1 Workshop 1 - WK01 . . . 34
1.4.4.2.2 Workshop 2 - WK02 . . . 36
1.4.4.2.3 Workshop 3 - WK03 . . . 37
1.4.4.2.4 Workshop 4 - WK04 . . . 38
1.4.4.2.5 Workshop 5 - WK05 . . . 39
1.4.4.2.6 Workshop 6 - WK06 . . . 41
1.4.4.2.7 Workshop 7 - WK07 . . . 42
1.4.4.2.8 Workshop 8 - WK08 . . . 43
1.4.4.2.9 Workshop 9 - WK09 . . . 45
1.4.4.2.10 Workshop 10 - WK10 . . . 46
1.4.4.2.11 Workshop 11 - WK11 . . . 47
1.4.4.2.12 Workshop 12 - WK12 . . . 49
1.4.4.3 Sprint Pre-Planning . . . 50
1.4.5 Presentations . . . 50
1.4.5.1 Presentation Templates . . . 50
1.4.5.2 Presentation 1 - Initial Big Idea . . . 50
1.4.5.3 Presentation 2 - Detailed Requirements and Architecture . . . 55
1.4.5.4 Presentation 3 - Prototype and Data Analytics . . . 58
1.4.5.5 Presentation 4 - Pre-final Prototype . . . 61
1.4.5.6 Presentation 5 - Final Presentation . . . 64
1.4.6 Final Report . . . 67
1.4.6.1 Reviewed Literature . . . 67
1.4.6.2 Reviewed Apps . . . 67
1.4.6.2.1 Dublin Bus App . . . 69
1.4.6.2.2 Dublin Bus website . . . 84
1.4.6.2.3 Google Travel Planner App . . . 86
1.4.6.2.4 Moovit . . . 102
1.4.6.2.5 TFI Journey Plan App . . . 125
1.4.6.2.6 Page Title Here . . . 159
1.5 System Design . . . 159

1.5.1 Data Exploration . . . 159
1.5.2 Machine Learning Models . . . 159
1.5.2.1 Model 1 : time Series Analysis . . . 159
1.5.2.2 Model 2 : Random Forest Regression . . . 160
1.5.2.3 Other models . . . 162
1.5.3 Application Structure . . . 162
1.5.3.1 App Folder Structure . . . 162
1.5.3.2 Import Structure . . . 163
1.5.3.3 Routes . . . 163
1.5.3.4 Tests . . . 163
1.5.4 DevOps . . . 163
1.5.5 Integrations . . . 164
1.5.5.1 Dublin Bus Data . . . 164
1.5.5.1.1 RT_Trips Data . . . 164
1.5.5.1.2 RT_LeaveTimes Data . . . 165
1.5.5.1.3 GTFS-R Data . . . 166
1.5.5.2 Weather Data . . . 167
1.5.5.2.1 Historic Data . . . 167
1.5.5.2.2 Current and Forecast Data . . . 168
1.5.5.3 Other Integration 4 . . . 210
1.5.5.4 UCD Systems . . . 211
1.5.5.4.1 CS UCD Virtual Machine . . . 211
1.5.5.4.2 CS UCD GIT . . . 211
1.5.5.4.3 CS UCD High Performance Server . . . 212
1.5.5.5 Azure MS SQL . . . 212
1.5.6 Data Model . . . 213
1.5.6.1 Overall Data Model . . . 213
1.5.6.2 Extract Data Store . . . 215
1.5.6.2.1 RDS_HistoricBus_RT_Trips . . . 215
1.5.6.2.2 RDS_HistoricBus_RT_LeaveTimes . . . 219
1.5.6.2.3 RDS_HistoricalWeather_Data . . . 223
1.5.6.2.4 RDS_CurrentWeather . . . 228
1.5.6.2.5 RDS_WeatherForecast . . . 229
1.5.6.2.6 rds_EventsData . . . 230
1.5.6.2.7 rds_RealTimeBusData . . . 230
1.5.6.2.8 rds_Calendar . . . 230
1.5.6.2.9 rds_CalendarDates . . . 230
1.5.6.2.10 rds_Shapes . . . 230
1.5.6.2.11 rds_Stops . . . 230
1.5.6.2.12 rds_Transfers . . . 230
1.5.6.2.13 rds_Trips . . . 230
1.5.6.2.14 rds_Agency . . . 230
1.5.6.2.15 rds_StopTimes . . . 230
1.5.6.2.16 rds_Routes . . . 230
1.5.6.3 Transform Data Store . . . 231
1.5.6.3.1 Machine Learning Store . . . 231
1.5.6.3.2 Frontend Data Store . . . 231
1.5.6.4 Loading Data Store . . . 232
1.5.6.4.1 full_routes.json . . . 232
1.5.6.4.2 route_Names.json . . . 232
1.5.6.4.3 stops_with_routes.json . . . 232
1.5.7 Design Decisions . . . 233
1.5.8 Docker and containers . . . 233
1.6 UAT, PREPROD, PROD Plan . . . 233
1.6.1 Deployment Guide . . . 233
1.7 Retrospectives . . . 234

COMP47360 Home

Hey Turlough, Daniel, Danning,

This is the homepage of our Confluence.

Confluence is primarily used for the Research Practicum project documentation and comes with many cool features and automations which can be run. There are some default tip sections present and I've added some to various pages to help you navigate and learn how to use it.

The Jira board and Confluence pages can be quickly accessed via the navbar on the left-hand side.

For reference, please see tips below on how to get the most of Jira and Confluence:

Welcome to your new space

Use it to create something wonderful.

To start, you might want to:

- **Customise this overview** using the **edit icon** at the top right of this page.
- **Create a new page** by clicking the **+** in the space sidebar, then go ahead and fill it with plans, ideas, or anything else your heart desires.

Need inspiration?

- Get a quick intro into what spaces are, and how to best use them at Confluence 101: organize your work in spaces.
- Check out our guide for ideas on how to set up your space overview.
- If starting from a blank space is daunting, try using one of the space templates instead.

Product Specification

Bus companies produce schedules which contain generic travel times. For example, in the Dublin Bus Schedule, the estimated travel time from Dun Laoghaire to the Phoenix Park is 61 minutes (http://dublinbus.ie/Your-Journey1/Timetables/All-Timetables/46a-1/).

Of course, there are many variables which determine how long the actual journey will take. Traffic conditions which are affected by the time of day, the day of the week, the month of the year and the weather play an important role in determining how long the journey will take.

These factors along with the dynamic nature of the events on the road network make it difficult to efficiently plan trips on public transport modes which interact with other traffic.

This project involves analysing historic Dublin Bus data and weather data in order to create dynamic travel time estimates.

Based on data analysis of historic Dublin Bus data, a system which when presented with any bus route, departure time, the day of the week, current weather condition, produces an accurate estimate of travel time for the complete route and sections of the route.

Users should be able to interact with the system via a web-based interface which is optimised for mobile devices. When presented with any bus route, an origin stop and a destination stop, a time, a day of the week, current weather, the system should produce and display via the interface an accurate estimate of travel time for the selected journey

Product Requirements

Target release	20 Aug 2021
MVP Release Date	13 Aug 2021
Document status	BEGIN
Confluence owner	@ Adam Ryan
Product Owner	@GavinMcCardle

Purpose

The purpose of this page is to define the requirements for the creation of a Dublin Bus web application which allows users to choose journeys and receive predictions based on 2018 journey data.

This document is for the use of the development team and the product sponsors (the module coordinators of COMP47360).

Scope

The scope of this project will entail

1. The production and deployment of a mobile web application by the development team.

This application will:

- Allow users to input a start point, end point, day, and time, and return a bus route and the estimated travel time for that journey.
    - This journey should be visualised on a map with information on the stops during the route.
- Be optimised for mobile web browsers but compatible with desktop browsers.
- Contain a user account system which will allow the user to save their favourite journeys.
- All users to share journeys via social media.
- Incorporate RTI for vehicles.
- Display the current weather and predicted weather for upcoming journeys to allow the user to plan their journey.
- Allow the user to provide feedback on journeys taken and the application for business intelligence tools.

The key objective for the creation of this application is:

1. To provide an application which creates accurate dynamic travel time estimates factoring in the start and end point of the journey, the departure time, the day of the week, and the weather conditions, improving upon a model which solely relies on bus schedules. This application should be accessible on mobile with a modern interface which allows the user to intuitively use the app.

This objective will be assessed by the following key KPIs:

1. The applications should have an NPS exceeding 70% by the final sprint.
2. An A- grade or higher should be received by the end of the project.
3. The project should be released meeting the requirements.

System Interfaces

User Interface

This section details the interfaces between the user and the application.

- A user will access and interact with the mobile application through a graphical user interface.
- A user should be presented with a default homepage.
- All pages should be able to navigate to the Journey Planner, the Account Menu, and the Route view.
- All pages should be intuitively navigate able to the user through the usage of consistent design language.
- All pages should be responsive to the user.
- All pages must be available in English.
- All pages should be structured in a consistent manner.
- All pages should implement consistent font, styling, visual cues, and colouring.
- In-Inbox messaging should be available.
- All pages should have colour scheme customisation.
- Copy should be written with minimal complexity.
- All interactions with opt-ins must be preserves.
- All user interactions should be captured.
- Forms should capture data in a secure manner.

Hardware Interface

This section details the hardware interfaces.

- A modern device (mobile or desktop) is expected to access the application.

- Web browsers should be optimised for mobile, with an emphasis on Safari, Chrome, Firefox, and Windows Edge.

Software Interface

There are no explicit software interfaces required for the running of the application outside of the web browser. Interfaces will be required to the various data sources.

Communication Interfaces

The app is required to be able to communicate with users in the following way.

- Users should be able to push routes to social media.
- In-inbox messaging should be supported.
- Data capture forms for user feedback should be implemented.

Memory Constraints

The application should be built to fit within the UCD Virtual Machine's memory.

Operational Constraints

There are no explicit operational constraints.

Product Functions

Journey Planner

The user should be able to enter their a day, time, start location, and end location, and show the nearest stop to the starting and ending location, and routes that can be taken to get there.

Map View

There should be a map view of a route.

Route Sharing

The user should be able to share their route on social media and save it to their calendar.

Fare Info

The application should display fare info to the user.

Business Intelligence Info

The application should display info on the user interactions with the application and customer base.

User Accounts

The user should be able to create and login to a user account.

Route Favouriting

The user should be able to favourite and store routes.

User Feedback

The user should be able to provide feedback on routes taken and the application.

Journey Model

The application should contain models for journey travel.

Transport Systems

The application may contain real time information on other transportation systems.

Traffic Info

The application should display traffic info to the user in Real Time.

In-App Messaging

The application should allow users to see in-inbox messages in the app.

User Journeys

	User Journey	User Role	Release	Acceptance Metric
1	As a user, I would like to be able to set a start date, end date, location, and time, and be able to see the route to take in my journey	End User	MVP	A journey can be planned in the application.
2	As a user, I want to see my route on the map.	End User	MVP	The user can see their journey on the map.
3	As a user, I want the design to be responsive on both mobile and desktop, to have a seamless interface.	End User	MVP	The UX adapts to user interactions on both mobile and desktop based on screen size, with application features scaling appropriately.
4	As a user, I want to be able to share my journey with others on Facebook or Twitter, to allow others to travel with me.	End User	Stretch	The end user can create a post on Social Media with a link to their journey.
5	As a user, I want to be able to favourite routes, in order to quickly return to my favourite journeys.	End User	Stretch	The user can favourite a route, logout, login, and see their favourite route.
6	As a user, I want to be able to save my planned journeys to my calendar, in order to view it on my mobile device to plan my journeys	End User	Stretch.	The user should be able to save their journey to their digital calendar.
7	As a user, I want my data to be held in a secure way, in order to be confident in using the app	End User	MVP	The application should be GDPR compliant, with appropriate opt ins.
8	As a user, I want to be able to request a deletion of all of my data held in the application, to feel secure in the app.	End User	MVP	The application should be GDPR compliant with RTBF requests.
9	As a user, I want to be able to see other agencies outside of just Dublin Bus, to plan better journeys.	End User	Stretch	The application should include routes and trips for other agencies, and should show real time info on these (modelling not in scope).
10	As a user, I want to be able to see the weather at stops at the moment, and I also want to be able to see the forecasted weather in the next five days, to allow me to plan my journey better.	End User	MVP	The application should incorporate current weather and future weather.
11	As a user, I want accurate travel times using a basic model, to get a sense of the accuracy of my journeys.	End User	MVP	The application should possess a basic Modelling system which incorporates weather and historic bus data.
12	As a user, I want very accurate travel times using a sophisticated model, to get accurate journey predictions	End User	Stretch	The application should feature an improved (non-basic model).
13	As a user, I want to earn points for journeys which I take and get an appropriate tier, to be more engaged in the application	End User	Stretch	The application should contain a tier system based on the number of journeys travelled to aid in user retention.
14	As a user, I want to see what the fare will be for a journey, to allow for more accurate journey planning.	End User	Stretch	The application should display the cost of fares to users.
15	As a user, I want to see real time traffic info in the application, to allow for a sense of timing.	End User	Stretch	The application should display RT traffic info.
16	As a user, I want to be able to give feedback on specific journeys, to help the app improve route info.	End User	Stretch	The application should prompt the user for feedback on journeys taken.

17	As a user, I want to be able to give feedback on the app design, to help the app fine-tune the development.	End User	Stretch	The application should have a form for the user to provide numeric/categorical feedback on the app design.
18	As a user, I want to be able to choose between light mode and dark mode, to support my design.	End User	Stretch	The application should have a minimum of two colour schemes.
19	As a product owner, I want to understand how many accounts are using the application	PO	Stretch	The application should display an admin screen with info on accounts in the applicaiton.
20	As a product owner, I want to understand the demographic of my application, to understand my base	PO	Stretch	The application should display info on the metrics of the base.
21	As a product owner, I want to understand where my users found the application, to support marketing efforts	PO	Stretch	The application should incorporate Google Analytics and UTM tracking.
22	As a product owner, I want to understand the user feedback submitted in forms, to help develop the application and address feedback including an NPS commentary	PO	Stretch	The application should have a display of the user feedback provided with a timeframe filter,

Detailed Requirements

Goal	Metric
Fill in this table	Other

Milestones and User Stories

The Roadmap contains information on the product roadmap and various Milestones.

Epics in this roadmap correspond to the list of User Stories.

[Insert Jira Links]

Product Roadmap

> **Team mission**
>
> The goal of this project is to develop a web application to predict travel times for Dublin Bus, optimised for Mobile.

📋 Roadmap overview

🔍 Detailed sprint roadmap

Sprint 1 - SM:

Sprint 2 - SM:

Feature	Initiative	Dates	Priority	Effort	Status	Notes
			HIGH / MEDIUM / LOW	HIGH / MEDIUM / LOW	IN PROGRESS / NOT STARTED / SHIPPED	

Sprint 3 - SM:

Feature	Initiative	Dates	Priority	Effort	Status	Notes
			HIGH / MEDIUM / LOW	HIGH / MEDIUM / LOW	IN PROGRESS / NOT STARTED / SHIPPED	

Sprint 4 - SM:

Feature	Initiative	Dates	Priority	Effort	Status	Notes
			HIGH / MEDIUM / LOW	HIGH / MEDIUM / LOW	IN PROGRESS / NOT STARTED / SHIPPED	

Sprint 5 - SM:

Feature	Initiative	Dates	Priority	Effort	Status	Notes
			HIGH / MEDIUM / LOW	HIGH / MEDIUM / LOW	IN PROGRESS / NOT STARTED / SHIPPED	

Project Management

Team Member Contact List

	FirstName	LastName	Internal /External	Role	Email	PhoneNumber	Availability	Jira @
1	Adam	Ryan	Internal	Customer Lead	adam.ryan@ucdconnect.ie	0834240698	Tuesdays, Thursdays, Saturdays, Sundays	@ Adam Ryan
2	Daniel	Danev	Internal	Developer	daniel.danev@ucdconnect.ie		All	@ Daniel Danev
3	Danning	Zhan	Internal	Developer	danning.zhan@ucdconnect.ie		All	@ Danning Zhan
4	Turlough	Hannon	Internal	Product Owner	turlough.hannon@ucdconnect.ie		All	@ Turlough Hannon

Requirements - Discovery Phase

This page is to discuss specific requirements to be converted into epics, stories, tasks.

Sprints

This page is designed to track our sprints

Project timeline and sprint plan visualisation: click here

Sprint 1

This Page is designed to track pages related to the first sprint (i.e. Change Management, Preplanning, or Retrospective).

Sprint 1 - Pre-Planning

- Sprint planning checklist
- Sprint team members
- Sprint planning meeting items
  - Agenda
  - Discussion / Minutes
- Sprint planning resources
  - Sprint boards and retrospectives

✅ *Sprint planning checklist*

Meeting	Follow up
☐ Determine Sprint 1 Objective	☐ Create Jira Epics for Sprint
☐ Close Out Previous Sprint	☐ Create Jira Stories for Sprint
☐ Determine User Journeys	☐ Update the User Journey Section
☐ Any other business	☐ Complete Confluence Pages

👥 *Sprint team members*

Name	Role
@ Adam Ryan	
@ Daniel Danev	
@ Turlough Hannon	
@ Danning Zhan	

✏️ *Sprint planning meeting items*

Agenda

1. The following questions / points of discussion were placed on the agenda prior to the meeting:
   - Sprint planning:
     - What is the end goal of this sprint? What do we want to have achieved by Friday / next Monday?
     - From a logistical perspective, when will we start planning our next sprint / move on?
   - Data management:
     - General look at the Dublin Bus data: how we should approach it / start working with it?
     - How should we handle the question of real-time versus historical data?
     - What might we look to use a database for? Which database approach would meet these needs best?
   - Frontend:
     - Do we want to use a frontend framework, and if so which one?

Discussion / Minutes

- Preferences
  - Daniel: Frontend, not analytics.
  - Adam: Backend, analytics, not frontend
  - Danning: Analytics, not frontend
  - Turlough: not frontend
- Analytics Model
  - What sort of data do we have?
    - Have to get started on analytics ASAP (deadline of late sprint 2 / early sprint 3)

        - This gives historic data greater immediate priority, as that is what we'll use for modelling
        - We have to ensure compatibility between our historic and RT data
        - Store everything + Save it
        - NTA Data is pretty extensive so we need a way to handle its scale
            - SQLite
            - dask
        - Road / Traffic data is definitely a dead end
        - Must look into Event API + Holiday Season API
        - Customer Data
            - Drive model Refreshes
        - Overview of our data sources
            - Historic modelling data
                - NTA
                - Weather (MET Éireann or Open Weather Map)
                - (possibly) Events
            - Real-Time data
                - GTFS
                - Weather
                - (possibly) Events
            - User account / login data
            - User interaction data
    - Finally:
        - Historic Route Data
        - Historic Weather Data
            - MET Éireann (advantage: data is free)
            - Open Weather Map (advantages: familiarity, code reuse)
        - Future Data:
            - API real-time
- Database
    - While the NTA data must be stored on UCD servers, that data is not for use in the running app (only for modelling) and can afterwards be considered decommissioned. There may be advantages to storing application data separate from modelling data
    - Azure:
        - Advantages
            - Data Sizes
            - Cloud Hosting is ubiquitous
            - Could connect locally.
        - Disadvantages
            - Latency if hosted distantly on free/student tier (e.g. in US)
    - MySQL
        - Advantages
            - Local
            - Low latency
        - Disadvantages
            - Size issue???
            - Installation Size
    - AWS Free Tier
        - Check this piece
        - Changes in Amazon took effect last month - details?
    - SQLite
        - Local hosting
        - Analysis - not for use in app architecture
- Frontend
    - Happy to use Vue as base
    - Examine options such as Bootstrap-Vue / Quasar
- **Sprint 1 - Broad Outcome**
    - **Backend framework selected**
    - **Frontend framework selected**
    - **DBMS selected (local vs remote)**
    - **User stories prepared**
    - **Initial/Preliminary UI design**

Sprint planning resources

Sprint boards and retrospectives

- Sprint 1: Discovery + Exploratory Analytics + Schema
  - Exploratation
  - User requirements and User Journeys
    - Wireframe
  - Settling on
- Sprint 2: MVP/Prototype
- Sprint 3: Advanced Features + ML with Analytics
- Sprint 4: Advanced
- Sprint 5: Bug Testing

Sprint 1 - Burndown Chart

This page should be updated with the Burndown Chart of the Sprint

Sprint 1 - Retrospective

Date	12 Jun 2021
Team	COMP47630
Participants	@ Adam Ryan @ Turlough Hannon @ Danning Zhan @ Daniel Danev

Background

This page is designed to provide a retrospective on Sprint 1

Retrospective

> Input retrospective details into the table below.

Start doing	Stop doing	Keep doing
• Daily Meetings (5/10 minutes)	• Key Pointing meetings to stop out-of-handedness	• Good communication in Discord (statuses).
• Asking about features a bit more / tagging / etc.	• N/A	• Teamwork went well.
• Work on LaTeX template.	• N/A	• N/A

Action items

The below are the actions based on the retrospective:

- [ ] @ Turlough Hannon - Schedule a daily standup in calendars (10 minutes max)

Sprint 2

This Page is designed to track pages related to the second sprint (i.e. Change Management, Preplanning, or Retrospective).

Sprint 2 - Pre-Planning

- Sprint planning checklist
- Sprint team members

- Sprint planning meeting items
  - Agenda
- Sprint planning resources
  - Sprint boards and retrospectives
- Actions

✅ *Sprint planning checklist*

Meeting	Follow up
☐ Determine Sprint 1 Objective	☐ Create Jira Epics for Sprint
☐ Close Out Previous Sprint	☐ Create Jira Stories for Sprint
☐ Determine User Journeys	☐ Update the User Journey Section
☐ Any other business	☐ Complete Confluence Pages

👥 *Sprint team members*

Name	Role
@ Adam Ryan	Developer
@ Daniel Danev	Developer
@ Danning Zhan	Developer
@ Turlough Hannon	Developer

✏️ *Sprint planning meeting items*

Agenda

1. Close previous sprint.
2. Recap project to date.
    a. Backend / Frontend Decided
        i. Vue + Flask
    b. Data Exploration Started
    c. Initial ETL Diagram
    d. Setup on:
        i. Github
        ii. Azure
        iii. UCD servers
    e. Data for:
        i. Historic Bus Data
        ii. Historic Web Data
        iii. APIs for OpenWeatherMap - Weather
        iv. Calendarific - Events
        v. GTFS-R - Bus
3. Set sprint objective.
    a. MVP Application
        i. Map View
        ii. Basic ML Model
        iii. Backend Tables Setup (incl. data flows/extraction)
        iv. Route Planner (Start Bus Stop, End Bus Stop, Show on Map, timing as stretch)
        v. Real Time Data Integration (incl. Traffic View as Stretch)
        vi. Backend of User Login (Frontend as Stretch)

b. Surveys + Ethical Approval Application
    i. Create survey
    ii. Create analytics
    iii. Creating Ethical Approval/Sheet/Submitting
c. LaTeX Template for Report.
d. Backlog.

4. Any other business.
    a. N/A

Sprint planning resources

Sprint boards and retrospectives

- https://disappster.atlassian.net/jira/software/projects/COMP47360/boards/2

Actions

- [x] @ Turlough Hannon - Convert Plans above into Epics and Tasks
- [ ] @ Turlough Hannon - Schedule Daily Meeting in evening (10 minutes at max) (if later than 5pm this would be good - In the Daily Standup Folder I've some templates here to help schedule and keep it in control; a suggested way of approaching it is:
  "Hey everybody, yesterday I was working on looking at documenting the ETL Diagram and I've put it in the Confluence. Today I'm going to be working on analysing the MET data. I've no blockers [alt: I need a basic model incorporated to continue]. Turlough, I'll catch up with you after this on Ticket: COMP47360-32 - Getting issue details... STATUS ".
  This means everybody is forced to keep it as one to three sentences. Any deviations get handled strictly after the meeting.

Sprint 2 - Burndown Chart

This page should be updated with the Burndown Chart of the Sprint

Sprint 2 - Retrospective

Date	12 Jun 2021
Team	COMP47630
Participants	@ Adam Ryan @ Turlough Hannon @ Daniel Danev @ Danning Zhan

Background

This page is designed to provide a retrospective on Sprint 2

Retrospective

Start doing	Stop doing
• Clear Direction on what's needed	• Presentation should not be done on the day before
• Define more tasks in advance	• Cut down meeting time on presentation prep
• Formally working on the report	• Keep project limit down
• Implement Production/Dev Branches in VueDev and Frontend	

Action items

Sprint 3

This Page is designed to track pages related to the third sprint (i.e. Change Management, Preplanning, or Retrospective).

Sprint 3 - Pre-Planning

- Sprint planning checklist
- Sprint team members
- Sprint planning meeting items
    - Agenda
    - Discussion
- Sprint planning resources
    - Sprint boards and retrospectives

✅ *Sprint planning checklist*

Meeting	Follow up
☑ Determine Sprint 3 Objective	☐ Create Jira Epics for Sprint
☑ Close Out Previous Sprint	☐ Create Jira Stories for Sprint
☐ Determine User Journeys	☐ Update the User Journey Section
☑ Any other business	☐ Complete Confluence Pages

👥 *Sprint team members*

Name	Role
@ Adam Ryan	

✏️ *Sprint planning meeting items*

Agenda

1. Close previous sprint.
2. Recap project to date.
3. Set sprint objective.
4. Determine User Journeys.
5. Any other business.

Discussion

List key discussion points below:

1. Close previous sprint.
    a. Cleared Board and Started Sprint
2. Recap project to date.
3. Set sprint objective.
    a. Twitter / Facebook trip sharing
    b. Twitter embed API (Dublin Bus feed)
    c. Favourite routes/trips
        i. Routes i.e. Journey Segments
        ii. Prior History
            1. search history

2. route history

iii. Locations
1. Redirection of Home Page

d. Save journeys to calendar (Google Calendar API?)
i. Log journeys taken to database based on journeys posted to the user's calendar

e. Deletion of user data

f. Dublin Bus ONLY Weather is the Problem for areas outside of Dublin.

g. Weather should be included.

h. Model Improvements
i. Other types of Models
ii. Other feature combinations
iii. Creation of GTFS data frame for evaluation on current.

i. Advanced Features
i. Visualisations Delay by Day | Average Delay by Weather Type | Most
ii. Gamification Stretch
iii. Admin Panel
1. User Counts Include in App
2. Demographic Cut
3. GA Stretch/Cut
4. Form Include in App

j. Add Cost Feature.
i. Based on Data from Table Static
1. Stretch: Based on Calendar with Users

k. User Feedback Very Basic Form Categorical View

l. Light and Dark Theme.

m. General UX Improvements

4. Any other business.

Sprint planning resources

Sprint boards and retrospectives

Sprint 3 - Burndown Chart

This page should be updated with the Burndown Chart of the Sprint

Sprint 3 - Retrospective

Date	12 Jun 2021
Team	COMP47630
Participants	@ Adam Ryan

Background

This page is designed to provide a retrospective on Sprint 1

Retrospective

Start doing	Stop doing	Keep doing

Action items

Sprint 4

This Page is designed to track pages related to the fourth sprint (i.e. Change Management, Preplanning, or Retrospective).

Sprint 4 - Pre-Planning

- Sprint planning checklist
- Sprint team members
- Sprint planning meeting items
  - Agenda
  - Discussion
- Sprint planning resources
  - Sprint boards and retrospectives

✅ *Sprint planning checklist*

Meeting	Follow up
☐ Determine Sprint 1 Objective	☐ Create Jira Epics for Sprint
☐ Close Out Previous Sprint	☐ Create Jira Stories for Sprint
☐ Determine User Journeys	☐ Update the User Journey Section
☐ Any other business	☐ Complete Confluence Pages

👥 *Sprint team members*

Name	Role
@ Adam Ryan	

✏️ *Sprint planning meeting items*

Agenda

1. Close previous sprint.
2. Recap project to date.
3. Set sprint objective.
4. Determine User Journeys.
5. Any other business.

Discussion

List key discussion points below:

- Sample Point 1
- Sample Point 2
- Sample Point 3

📚 *Sprint planning resources*

Sprint boards and retrospectives

Sprint 4 - Burndown Chart

This page should be updated with the Burndown Chart of the Sprint

Sprint 4 - Retrospective

Date	12 Jun 2021
Team	COMP47630
Participants	@ Adam Ryan

Background

This page is designed to provide a retrospective on Print 1

Retrospective

Start doing	Stop doing	Keep doing

Action items

- [ ]

Sprint 5

This Page is designed to track pages related to the fifth sprint (i.e. Change Management, Preplanning, or Retrospective).

Sprint 5 - Pre-Planning

- Sprint planning checklist
- Sprint team members
- Sprint planning meeting items
  - Agenda
  - Discussion
- Sprint planning resources
  - Sprint boards and retrospectives

✅ *Sprint planning checklist*

Meeting	Follow up
- [ ] Determine Sprint 1 Objective	- [ ] Create Jira Epics for Sprint
- [ ] Close Out Previous Sprint	- [ ] Create Jira Stories for Sprint
- [ ] Determine User Journeys	- [ ] Update the User Journey Section
- [ ] Any other business	- [ ] Complete Confluence Pages

👥 *Sprint team members*

Name	Role
@ Adam Ryan	

✏️ *Sprint planning meeting items*

Agenda

1. Close previous sprint.
2. Recap project to date.
3. Set sprint objective.
4. Determine User Journeys.
5. Any other business.

Discussion

List key discussion points below:

- Sample Point 1
- Sample Point 2
- Sample Point 3

🗂️ *Sprint planning resources*

Sprint boards and retrospectives

Sprint 5 - Burndown Chart

This page should be updated with the Burndown Chart of the Sprint

Sprint 5 - Retrospective

Date	12 Jun 2021
Team	COMP47630
Participants	@ Adam Ryan

Background

This page is designed to provide a retrospective on Print 1

Retrospective

Start doing	Stop doing	Keep doing

Action items

Meetings

This page is designed to track meeting notes.

Daily Standups

These pages are designed to track daily stand-ups if applicable.

Daily Standups - WK01

Information	Values
Date:	11 Jun 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan
Daily Standup Time:	Time
Agenda:	WIP
Link:	Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- Determine what team members are working on, development progress, blockers.

Discussion Topics

- Adam’s Work in Progress + Blockers
- Daniel’s Work in Progress + Blockers
- Danning's Work in Progress + Blockers
- Turlough’s Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Daniel	Danning	Turlough
1	**13 Jun 2021**				
2	**14 Jun 2021**				
3	**15 Jun 2021**				
4	**16 Jun 2021**				
5	**17 Jul 2021**				

Work in Progress

	Date	Adam	Daniel	Danning	Turlough
1	**13 Jun 2021**				
2	**14 Jun 2021**				
3	**15 Jun 2021**				
4	**16 Jun 2021**				
5	**17 Jul 2021**				

Blockers

	Date	Adam	Daniel	Danning	Turlough
1	**13 Jun 2021**				
2	**14 Jun 2021**				
3	**15 Jun 2021**				
4	**16 Jun 2021**				
5	**17 Jul 2021**				

Action items

- [x] @ Adam Ryan - Sample
- [x] Sample

Decisions

- May not be relevant

Daily Standups - WK02

Information	Values
Date:	11 Jun 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan
Daily Standup Time:	Time
Agenda:	WIP
Link:	Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- Determine what team members are working on, development progress, blockers.

Discussion Topics

- Adam's Work in Progress + Blockers
- Daniel's Work in Progress + Blockers
- Danning's Work in Progress + Blockers
- Turlough's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Daniel	Danning	Turlough
1	**21 Jun 2021**				
2	**22 Jun 2021**				
3	**23 Jun 2021**				
4	**24 Jun 2021**				
5	**25 Jun 2021**				

Work in Progress

	Date	Adam	Daniel	Danning	Turlough
1	**21 Jun 2021**				
2	**22 Jun 2021**				
3	**23 Jun 2021**				
4	**24 Jun 2021**				
5	**25 Jun 2021**				

Blockers

	Date	Adam	Daniel	Danning	Turlough
1	**21 Jun 2021**				
2	**22 Jun 2021**				
3	**23 Jun 2021**				
4	**24 Jun 2021**				
5	**25 Jun 2021**				

Action items

- [x] @ Adam Ryan - Sample
- [x] Sample

Decisions

- May not be relevant

Daily Standups - WK03

Information	Values
Date:	11 Jun 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan
Daily Standup Time:	Time
Agenda:	WIP
Link:	Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- Determine what team members are working on, development progress, blockers.

Discussion Topics

- Adam's Work in Progress + Blockers
- Daniel's Work in Progress + Blockers
- Danning's Work in Progress + Blockers
- Turlough's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Daniel	Danning	Turlough
1	**28 Jun 2021**				
2	**29 Jun 2021**				
3	**30 Jun 2021**				
4	**01 Jul 2021**	Compiling scraping models.			
5	**02 Jul 2021**				GTFS-R scraper

Work in Progress

	Date	Adam	Daniel	Danning	Turlough
1	**28 Jun 2021**		Login + Register		
2	**29 Jun 2021**		Login + Register		
3	**30 Jun 2021**		Login + Register		
4	**01 Jul 2021**	Scraper - Weather (forecast /current), Events		Splitting Bike Data	Initiate/scrape GTFS-R

5	**02 Jul 2021**				GTFS-R scraper

Blockers

	Date	Adam	Daniel	Danning	Turlough
1	**28 Jun 2021**				
2	**29 Jun 2021**				
3	**30 Jun 2021**	Azure Connection Params			
4	**01 Jul 2021**	N/A			none
5	**02 Jul 2021**				none

Action items

- [x] @ Adam Ryan - Sample
- [x] Sample

Decisions

- May not be relevant

Daily Standups - WK04

Information	Values
Date:	11 Jun 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan
Daily Standup Time:	Time
Agenda:	WIP
Link:	Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- Determine what team members are working on, development progress, blockers.

Discussion Topics

- Adam's Work in Progress + Blockers
- Daniel's Work in Progress + Blockers
- Danning's Work in Progress + Blockers

- Turlough's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Daniel	Danning	Turlough
1	**05 Jul 2021**				GTFS scraper
2	**06 Jul 2021**				
3	**07 Jul 2021**				
4	**08 Jul 2021**				
5	**09 Jul 2021**				

Work in Progress

	Date	Adam	Daniel	Danning	Turlough
1	**05 Jul 2021**				
2	**06 Jul 2021**				
3	**07 Jul 2021**				
4	**08 Jul 2021**				
5	**09 Jul 2021**				

Blockers

	Date	Adam	Daniel	Danning	Turlough
1	**05 Jul 2021**				none
2	**06 Jul 2021**				
3	**07 Jul 2021**				
4	**08 Jul 2021**				
5	**09 Jul 2021**				

Action items

- [x] @ Adam Ryan - Sample
- [x] Sample

Decisions

- May not be relevant

Daily Standups - WK05

Information	Values
Date:	11 Jun 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan

People To Attend Daily Standup:	@ Adam Ryan
Daily Standup Time:	Time
Agenda:	WIP
Link:	Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- Determine what team members are working on, development progress, blockers.

Discussion Topics

- Adam's Work in Progress + Blockers
- Daniel's Work in Progress + Blockers
- Danning's Work in Progress + Blockers
- Turlough's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Daniel	Danning	Turlough
1	**12 Jul 2021**				
2	**13 Jul 2021**				
3	**14 Jul 2021**				
4	**15 Jul 2021**				
5	**16 Jul 2021**				

Work in Progress

	Date	Adam	Daniel	Danning	Turlough
1	**12 Jul 2021**				
2	**13 Jul 2021**				
3	**14 Jul 2021**				
4	**15 Jul 2021**				
5	**16 Jul 2021**				

Blockers

	Date	Adam	Daniel	Danning	Turlough
1	**12 Jul 2021**				
2	**13 Jul 2021**				

3	**14 Jul 2021**				
4	**15 Jul 2021**				
5	**16 Jul 2021**				

Action items

- [x] @ Adam Ryan - Sample
- [x] Sample

Decisions

- May not be relevant

Daily Standups - WK07

Information	Values
Date:	11 Jun 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan
Daily Standup Time:	Time
Agenda:	WIP
Link:	Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- Determine what team members are working on, development progress, blockers.

Discussion Topics

- Adam's Work in Progress + Blockers
- Daniel's Work in Progress + Blockers
- Danning's Work in Progress + Blockers
- Turlough's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Daniel	Danning	Turlough
1	**19 Jul 2021**				

2	**20 Jul 2021**				
3	**21 Jul 2021**				
4	**22 Jul 2021**				
5	**23 Jul 2021**				

Work in Progress

	Date	Adam	Daniel	Danning	Turlough
1	**19 Jul 2021**				
2	**20 Jul 2021**				
3	**21 Jul 2021**				
4	**22 Jul 2021**				
5	**23 Jul 2021**				

Blockers

	Date	Adam	Daniel	Danning	Turlough
1	**19 Jul 2021**				
2	**20 Jul 2021**				
3	**21 Jul 2021**				
4	**22 Jul 2021**				
5	**23 Jul 2021**				

Action items

- [x] @ Adam Ryan - Sample
- [x] Sample

Decisions

- May not be relevant

Daily Standups - WK08

Information	Values
Date:	11 Jun 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan
Daily Standup Time:	Time
Agenda:	WIP
Link:	Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- Determine what team members are working on, development progress, blockers.

Discussion Topics

- Adam's Work in Progress + Blockers
- Daniel's Work in Progress + Blockers
- Danning's Work in Progress + Blockers
- Turlough's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Daniel	Danning	Turlough
1	**26 Jul 2021**				
2	**27 Jul 2021**				
3	**28 Jul 2021**				
4	**29 Jul 2021**				
5	**30 Jul 2021**				

Work in Progress

	Date	Adam	Daniel	Danning	Turlough
1	**26 Jul 2021**				
2	**27 Jul 2021**				
3	**28 Jul 2021**				
4	**29 Jul 2021**				
5	**30 Jul 2021**				

Blockers

	Date	Adam	Daniel	Danning	Turlough
1	**26 Jul 2021**				
2	**27 Jul 2021**				
3	**28 Jul 2021**				
4	**29 Jul 2021**				
5	**30 Jul 2021**				

Action items

- [x] @ Adam Ryan - Sample

Sample

Decisions

- May not be relevant

Daily Standups - WK09

Information	Values
Date:	11 Jun 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan
Daily Standup Time:	Time
Agenda:	WIP
Link:	Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- Determine what team members are working on, development progress, blockers.

Discussion Topics

- Adam's Work in Progress + Blockers
- Daniel's Work in Progress + Blockers
- Danning's Work in Progress + Blockers
- Turlough's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Daniel	Danning	Turlough
1	**02 Aug 2021**				
2	**03 Aug 2021**				
3	**04 Aug 2021**				
4	**05 Aug 2021**				
5	**06 Aug 2021**				

Work in Progress

	Date	Adam	Daniel	Danning	Turlough

1	**02 Aug 2021**				
2	**03 Aug 2021**				
3	**04 Aug 2021**				
4	**05 Aug 2021**				
5	**06 Aug 2021**				

Blockers

	Date	Adam	Daniel	Danning	Turlough
1	**02 Aug 2021**				
2	**03 Aug 2021**				
3	**04 Aug 2021**				
4	**05 Aug 2021**				
5	**06 Aug 2021**				

Action items

- [x] @ Adam Ryan - Sample
- [x] Sample

Decisions

- May not be relevant

Daily Standups - WK10

Information	Values
Date:	11 Jun 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan
Daily Standup Time:	Time
Agenda:	WIP
Link:	Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- Determine what team members are working on, development progress, blockers.

Discussion Topics

- Adam's Work in Progress + Blockers
- Daniel's Work in Progress + Blockers
- Danning's Work in Progress + Blockers
- Turlough's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Daniel	Danning	Turlough
1	**09 Aug 2021**				
2	**10 Aug 2021**				
3	**11 Aug 2021**				
4	**12 Aug 2021**				
5	**13 Aug 2021**				

Work in Progress

	Date	Adam	Daniel	Danning	Turlough
1	**09 Aug 2021**				
2	**10 Aug 2021**				
3	**11 Aug 2021**				
4	**12 Aug 2021**				
5	**13 Aug 2021**				

Blockers

	Date	Adam	Daniel	Danning	Turlough
1	**09 Aug 2021**				
2	**10 Aug 2021**				
3	**11 Aug 2021**				
4	**12 Aug 2021**				
5	**13 Aug 2021**				

Action items

- [x] @ Adam Ryan - Sample
- [x] Sample

Decisions

- May not be relevant

Daily Standups - WK11

Information	Values
Date:	11 Jun 2021

Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan
Daily Standup Time:	Time
Agenda:	WIP
Link:	Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- Determine what team members are working on, development progress, blockers.

Discussion Topics

- Adam's Work in Progress + Blockers
- Daniel's Work in Progress + Blockers
- Danning's Work in Progress + Blockers
- Turlough's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Daniel	Danning	Turlough
1	**16 Aug 2021**				
2	**17 Aug 2021**				
3	**18 Aug 2021**				
4	**19 Aug 2021**				
5	**20 Aug 2021**				

Work in Progress

	Date	Adam	Daniel	Danning	Turlough
1	**16 Aug 2021**				
2	**17 Aug 2021**				
3	**18 Aug 2021**				
4	**19 Aug 2021**				
5	**20 Aug 2021**				

Blockers

	Date	Adam	Daniel	Danning	Turlough
1	**16 Aug 2021**				
2	**17 Aug 2021**				
3	**18 Aug 2021**				
4	**19 Aug 2021**				
5	**20 Aug 2021**				

Action items

- [x] @ Adam Ryan - Sample
- [x] Sample

Decisions

- May not be relevant

Daily Standups - WK12

Information	Values
Date:	11 Jun 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan
Daily Standup Time:	Time
Agenda:	WIP
Link:	Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- Determine what team members are working on, development progress, blockers.

Discussion Topics

- Adam's Work in Progress + Blockers
- Daniel's Work in Progress + Blockers
- Danning's Work in Progress + Blockers
- Turlough's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Daniel	Danning	Turlough
1	**23 Aug 2021**				
2	**24 Aug 2021**				
3	**25 Aug 2021**				
4	**26 Aug 2021**				
5	**27 Aug 2021**				

Work in Progress

	Date	Adam	Daniel	Danning	Turlough
1	**23 Aug 2021**				
2	**24 Aug 2021**				
3	**25 Aug 2021**				
4	**26 Aug 2021**				
5	**27 Aug 2021**				

Blockers

	Date	Adam	Daniel	Danning	Turlough
1	**23 Aug 2021**				
2	**24 Aug 2021**				
3	**25 Aug 2021**				
4	**26 Aug 2021**				
5	**27 Aug 2021**				

Action items

- [x] @ Adam Ryan - Sample
- [x] Sample

Decisions

- May not be relevant

Workshops

This page is designed to track workshops.

Workshop 1 - WK01

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details
Last Modified by:	@ Adam Ryan

People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3
- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Workshop 2 - WK02

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3
- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Workshop 3 - WK03

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
    - What was complete?
    - What is in progress?
    - What are the issues?

- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3
- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Workshop 4 - WK04

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3
- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Workshop 5 - WK05

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2

- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3
- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Workshop 6 - WK06

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
    - What was complete?
    - What is in progress?
    - What are the issues?
- Presentation
    - What is the topic?
    - What is prepared?
    - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3
- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Workshop 7 - WK07

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3
- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Workshop 8 - WK08

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details

Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
  - [ ]

Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3

- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Workshop 9 - WK09

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?

        - What is prepared?
        - Run through of slides.
    - Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3
- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Workshop 10 - WK10

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3
- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Workshop 11 - WK11

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time

Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
  - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3
- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Workshop 12 - WK12

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Workshop Details
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Attendants

- Student - @ Adam Ryan
- Student - Daniel
- Student - Danning
- Student - Turlough

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Notes

- Retrospective on Previous Week's Work
  - What was complete?
  - What is in progress?
  - What are the issues?
- Presentation
  - What is the topic?
  - What is prepared?
  - Run through of slides.
- Other business

Discussion Note

Action items

- [ ] @ Adam Ryan - Action 1
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 2
- [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - Action 3
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 1
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 2
    - [ ] Demo (@ Mayuri Srinivasan (Unlicensed)) - SubAction 3
- [ ] @ScrumMaster - Create Tickets for Actions

Decisions

- Sample Decision 1
- Sample Decision 2
- Sample Decision 3

Sprint Pre-Planning

Notes pertaining to the sprint pre-planning meetings can be found in “Sprints” under the Project Management tab. Links to these pages are shown below:

Sprint 1 - Pre-Planning

Sprint 2 - Pre-Planning

Sprint 3 - Pre-Planning

Sprint 4 - Pre-Planning

Sprint 5 - Pre-Planning

Presentations

This page is to track the presentations.

Presentation Templates

This page should contain the template for our presentation templates for SGL, QBL, and PMU and the automations to populate them if relevant.

Presentation 1 - Initial Big Idea

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Presentation 1
Presentation Title:	Initial Big Idea
Last Modified by:	@ Adam Ryan
People To Attend:	@ Adam Ryan @ Daniel Danev @ Danning Zhan @ Turlough Hannon
Meeting Time:	2pm - 5pm
Agenda:	Complete the Presentation

Meeting Link:	Discord

Presentation Grading Reminder

Please keep in mind the marking scheme for the presentation

> Remember to keep in mind the marking scheme for the presentation when designing the structure!
>
> Presentations last for 8 minutes, with 4 minutes for questions and answers.

- Criteria 1: **Awareness of Audience & Delivery**

Excellent	Very Good	Fair	Poor
– Builds trust and holds attention of the audience; – Fluctuation in volume and inflection help to maintain audience interest and emphasise key points.	– Quick recovery from minor mistakes; – Fairly consistent use of direct eye contact with the audience. – Satisfactory variation of volume and inflection.	– Some tension or indifference apparent – Occasional but unsustained eye contact with audience; – Uneven volume with little or no inflection.	– Nervous tension obvious – No effort to make eye contact with audience; – Low volume and/or monotonous tone cause audience to disengage.

• Criteria 2: **Technical Content & Organisation**

Excellent	Very Good	Fair	Poor
– Major technical details summarised and effectively persuades the audience re the validity of the approach taken. – Conclusions/ideas are supported by evidence where appropriate. – Excellent organisation and use of illustrations.	– Clear description of approach taken although audience may not be persuaded of it's validity/appropriateness. – Reasonable attempt at providing evidence to support claims. – Very good structure, consistency and use of illustrations.	– Some major technical details left un-clear. – Includes very thin data or evidence in support of ideas or conclusions; – Minor issues in relation to the structure, consistency and general organisation of the presentation.	– Fails to effectively persuade the audience of usefulness of the approach taken. – Very weak or no support of technical proposal by way of examples, facts, and/or statistics; – No major ideas proposed and/or audience left confused.

• Criteria 3: **Timing & Handling of Questions**

Excellent	Very Good	Fair	Poor
– Excellent answers given for all questions. – Excellent use of allocated time.	– Clear and satisfactory answers given for all questions. – Finished presentation of relevant content on time.	– Reasonable answers given to all most questions. – Covered most of the relevant content in the allotted time.	– Incomplete or vague answers for the majority of questions. – Fails to cover the main points of the presentation in the allotted time.

Initial Big Idea

The big picture is your ideas to address the **specification** and **innovation** and any thoughts on how your team will **function/development process.**

Discussion

This section should detail meeting/discussion notes:

Intro/start: Context on problem; background reading and research

-What is the lay of the land? Existing Apps?

-Transport for London - Topology of Bus Routes

-Moovit: Responsive to user commentary

Innovation section: Discuss what we've looked at, what exists, what's new

-How what we want to implement compares with the existing apps

-See what the weather will be when you arrive (contain within the UI)

-Weather not so relevant for model but maybe UI? Traffic Data (consideration of counsel data - different traffic data)

-Sustainability.

-Fare Info: Leap API (saving info against it)

-Account Info (GDPR considerations)

-Business Intelligence Tools to identify customer

Achieving outcomes: How we'll function as a team, tools we're thinking about

-refer to Technical Content Area

Technical Content Area:

-Defer to next.

-Past experiences/Tool/Stacks.

- Timing:
  - Adam Ryan - Initial Big Picture
  - Daniel - Detailed Timing
  - Danning - Data Analytics
  - Turlough - Final

Structure and Reading

This section should recap the key background reading for each section.

Section 1 - Intro

- Examined the Specification.
- Key points.

Section 2 - Existing Solutions

- Detail existing app features
- User Reviews
- UX's for applications

Section 3 - Our Big Idea

- Wireframe
- Guiding Principles and Features
- DevOps and Tech

Section 4 -Roadmap

- Sprint Structure

- Documentation
- Meeting Type
- Team Structure.

Section 5 - Questions and Answers

- Felix complimented research on existing apps - Asked about user feedback and the analytics overhead which could be present by lots of user feedback
    - Structured Responses.
    - Google Surveys in Initial Stages
    - Forms built into app prompting user
    - Structured Responses in both to avoid overhead of Text Analytics.
- What are users complaining about in the App? What is the Key Problem? (Padraig)
    - UX one issue. Stereographic Design Principles foregoing modern flat design.
    - Forecasting second issue. Users don't trust the application so reject continued usage.

General Presentation Notes

The following are general notes on the presentations at large

- QR Code Innovation not very well received.
- Occupancy cited as difficult problem - Potential to get from users or link to travel times/user info (cleanliness, etc)?
- Probabilistic display of info very well received.
- Sustainability not touched on Maybe not useful?
- Docker as a choice questioned/criticised from Ellie's team by Padraig.
- Challenges tbe frequently asked.
- Common framework choices:
    - Docker (for some) > MySQL/Azure/SQLite > Django > Vue/React/Bootstrap (drag and drop)
        - No Angular is weird
    - Github
        - UCD Git barely touched on
- Machine Learning Models:
    - KNN/ANN looked at.
    - Mostly under discussion
- Architecture
    - Barely approached
- Split of backend/frontend/analytics HEAVILY pushed by Padraig.
- Eugene's Team had an incredible integrated Zoom presentation.

Final Presentation

Upcoming Presentations

> Remember, these are the upcoming presentations and schedule!

Date	Presentation Title
16-June-2021	Initial Big Picture
30-June-2021	Detailed Requirements and Architecture
14-July-2021	Prototype and Data Analytics
11-August-2021	Pre-final Prototype
25-August-2021	Final Presentation

Presentation 2 - Detailed Requirements and Architecture

Information	Values
Date:	15 Jun 2021
Author:	@ Daniel Danev
Page Description:	This looks at the Detailed Requirements
Presentation Title:	Detailed Requirements and Architecture.
Last Modified by:	@ Adam Ryan
People To Attend:	@ Adam Ryan @ Daniel Danev @ Danning Zhan @ Turlough Hannon
Meeting Time:	11:00
Agenda:	Finish the presentation.
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Presentation Grading Reminder

Please keep in mind the marking scheme for the presentation

> Remember to keep in mind the marking scheme for the presentation when designing the structure!
>
> Presentations last for 8 minutes, with 4 minutes for questions and answers.

- Criteria 1: **Awareness of Audience & Delivery**

Excellent	Very Good	Fair	Poor
– Builds trust and holds attention of the audience; – Fluctuation in volume and inflection help to maintain audience interest and emphasise key points.	– Quick recovery from minor mistakes; – Fairly consistent use of direct eye contact with the audience. – Satisfactory variation of volume and inflection.	– Some tension or indifference apparent – Occasional but unsustained eye contact with audience; – Uneven volume with little or no inflection.	– Nervous tension obvious – No effort to make eye contact with audience; – Low volume and/or monotonous tone cause audience to disengage.

- Criteria 2: **Technical Content & Organisation**

Excellent	Very Good	Fair	Poor
– Major technical details summarised and effectively persuades the audience re the validity of the approach taken. – Conclusions/ideas are supported by evidence where appropriate. – Excellent organisation and use of illustrations.	– Clear description of approach taken although audience may not be persuaded of it's validity/appropriateness. – Reasonable attempt at providing evidence to support claims. – Very good structure, consistency and use of illustrations.	– Some major technical details left un-clear. – Includes very thin data or evidence in support of ideas or conclusions; – Minor issues in relation to the structure, consistency and general organisation of the presentation.	– Fails to effectively persuade the audience of usefulness of the approach taken. – Very weak or no support of technical proposal by way of examples, facts, and/or statistics; – No major ideas proposed and/or audience left confused.

- Criteria 3: **Timing & Handling of Questions**

Excellent	Very Good	Fair	Poor
– Excellent answers given for all questions. – Excellent use of allocated time.	– Clear and satisfactory answers given for all questions. – Finished presentation of relevant content on time.	– Reasonable answers given to all most questions. – Covered most of the relevant content in the allotted time.	– Incomplete or vague answers for the majority of questions. – Fails to cover the main points of the presentation in the allotted time.

Presentation - Name

(this should be replaced with a brief overview of the presentation)

Background Reading

This section should recap the key background reading for each section.

Section 1 - User Journeys

Product Requirements

Section 2 - Requirements

Product Requirements

Section 3 - Architecture

Overall Data Model

Structure

This section should broadly touch what the structure is.

Section 1 - User Journeys

This section should provide context for the rest of the slides.

- Touch on user roles.
- Touch on user journeys by role.

Section 2 - Requirements

This section should expand the user journeys into more detailed requirements

- Touch on the requirement

Section 3 - Architecture

This section should expand the requirements into the architecture to fulfil the requirements

- Google API for Map
- Flask-Login for User accounts
- MVC architecture for scalability (data model)
- Overall data flow Flask with Vue
    - Why these?
        - Flask is modular
            - Not constrained to a structure
            - Want to understand the components would work
        - Vue is upcoming
            - Talk about features

Additional Notes

This section should detail key notes on the presentation

Upcoming Presentations

> Remember, these are the upcoming presentations and schedule!

Date	Presentation Title
16-June-2021	Initial Big Picture
30-June-2021	Detailed Requirements and Architecture
14-July-2021	Prototype and Data Analytics
11-August-2021	Pre-final Prototype
25-August-2021	Final Presentation

Presentation 3 - Prototype and Data Analytics

Information	Values
Date:	15 Jun 2021
Author:	@ Danning Zhan
Page Description:	Presentation 1
Presentation Title:	
Last Modified by:	@ Adam Ryan
People To Attend:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Presentation Grading Reminder

Please keep in mind the marking scheme for the presentation

Remember to keep in mind the marking scheme for the presentation when designing the structure!

Presentations last for 8 minutes, with 4 minutes for questions and answers.

- Criteria 1: **Awareness of Audience & Delivery**

Excellent	Very Good	Fair	Poor
– Builds trust and holds attention of the audience; – Fluctuation in volume and inflection help to maintain audience interest and emphasise key points.	– Quick recovery from minor mistakes; – Fairly consistent use of direct eye contact with the audience. – Satisfactory variation of volume and inflection.	– Some tension or indifference apparent – Occasional but unsustained eye contact with audience; – Uneven volume with little or no inflection.	– Nervous tension obvious – No effort to make eye contact with audience; – Low volume and/or monotonous tone cause audience to disengage.

- Criteria 2: **Technical Content & Organisation**

Excellent	Very Good	Fair	Poor
– Major technical details summarised and effectively persuades the audience re the validity of the approach taken. – Conclusions/ideas are supported by evidence where appropriate. – Excellent organisation and use of illustrations.	– Clear description of approach taken although audience may not be persuaded of it's validity/appropriateness. – Reasonable attempt at providing evidence to support claims. – Very good structure, consistency and use of illustrations.	– Some major technical details left un-clear. – Includes very thin data or evidence in support of ideas or conclusions; – Minor issues in relation to the structure, consistency and general organisation of the presentation.	– Fails to effectively persuade the audience of usefulness of the approach taken. – Very weak or no support of technical proposal by way of examples, facts, and/or statistics; – No major ideas proposed and/or audience left confused.

- Criteria 3: **Timing & Handling of Questions**

Excellent	Very Good	Fair	Poor
– Excellent answers given for all questions. – Excellent use of allocated time.	– Clear and satisfactory answers given for all questions. – Finished presentation of relevant content on time.	– Reasonable answers given to all most questions. – Covered most of the relevant content in the allotted time.	– Incomplete or vague answers for the majority of questions. – Fails to cover the main points of the presentation in the allotted time.

Presentation - Name

(this should be replaced with a brief overview of the presentation)

Background Reading

This section should recap the key background reading for each section.

Section 1 - Name/Key Idea

Section 1 Description

Section 2 - Name/Key Idea

Section 2 Description

Section 3 - Name/Key Idea

Section 3 Description

Section 4 - Name/Key Idea

Section 4 Description

Structure

This section should broadly touch what the structure is.

Section 1 - Name/Key Idea

Section 1 Description

Section 2 - Name/Key Idea

Section 2 Description

Section 3 - Name/Key Idea

Section 3 Description

Section 4 - Name/Key Idea

Section 4 Description

Additional Notes

This section should detail key notes on the presentation

Upcoming Presentations

> Remember, these are the upcoming presentations and schedule!

Date	Presentation Title
16-June-2021	Initial Big Picture
30-June-2021	Detailed Requirements and Architecture
14-July-2021	Prototype and Data Analytics
11-August-2021	Pre-final Prototype
25-August-2021	Final Presentation

Presentation 4 - Pre-final Prototype

Information	Values
Date:	15 Jun 2021
Author:	@ Turlough Hannon
Page Description:	Presentation 1
Presentation Title:	
Last Modified by:	@ Adam Ryan
People To Attend:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Presentation Grading Reminder

Please keep in mind the marking scheme for the presentation

> Remember to keep in mind the marking scheme for the presentation when designing the structure!

Presentations last for 8 minutes, with 4 minutes for questions and answers.

• Criteria 1: **Awareness of Audience & Delivery**

Excellent	Very Good	Fair	Poor
– Builds trust and holds attention of the audience; – Fluctuation in volume and inflection help to maintain audience interest and emphasise key points.	– Quick recovery from minor mistakes; – Fairly consistent use of direct eye contact with the audience. – Satisfactory variation of volume and inflection.	– Some tension or indifference apparent – Occasional but unsustained eye contact with audience; – Uneven volume with little or no inflection.	– Nervous tension obvious – No effort to make eye contact with audience; – Low volume and/or monotonous tone cause audience to disengage.

• Criteria 2: **Technical Content & Organisation**

Excellent	Very Good	Fair	Poor
– Major technical details summarised and effectively persuades the audience re the validity of the approach taken. – Conclusions/ideas are supported by evidence where appropriate. – Excellent organisation and use of illustrations.	– Clear description of approach taken although audience may not be persuaded of it's validity/appropriateness. – Reasonable attempt at providing evidence to support claims. – Very good structure, consistency and use of illustrations.	– Some major technical details left un-clear. – Includes very thin data or evidence in support of ideas or conclusions; – Minor issues in relation to the structure, consistency and general organisation of the presentation.	– Fails to effectively persuade the audience of usefulness of the approach taken. – Very weak or no support of technical proposal by way of examples, facts, and/or statistics; – No major ideas proposed and/or audience left confused.

- Criteria 3: **Timing & Handling of Questions**

Excellent	Very Good	Fair	Poor
– Excellent answers given for all questions. – Excellent use of allocated time.	– Clear and satisfactory answers given for all questions. – Finished presentation of relevant content on time.	– Reasonable answers given to all most questions. – Covered most of the relevant content in the allotted time.	– Incomplete or vague answers for the majority of questions. – Fails to cover the main points of the presentation in the allotted time.

Presentation - Name

(this should be replaced with a brief overview of the presentation)

Background Reading

This section should recap the key background reading for each section.

Section 1 - Name/Key Idea

Section 1 Description

Section 2 - Name/Key Idea

Section 2 Description

Section 3 - Name/Key Idea

Section 3 Description

Section 4 - Name/Key Idea

Section 4 Description

Structure

This section should broadly touch what the structure is.

Section 1 - Name/Key Idea

Section 1 Description

Section 2 - Name/Key Idea

Section 2 Description

Section 3 - Name/Key Idea

Section 3 Description

Section 4 - Name/Key Idea

Section 4 Description

Additional Notes

This section should detail key notes on the presentation

Upcoming Presentations

> Remember, these are the upcoming presentations and schedule!

Date	Presentation Title
16-June-2021	Initial Big Picture
30-June-2021	Detailed Requirements and Architecture
14-July-2021	Prototype and Data Analytics
11-August-2021	Pre-final Prototype
25-August-2021	Final Presentation

Presentation 5 - Final Presentation

Information	Values
Date:	15 Jun 2021
Author:	@ Adam Ryan
Page Description:	Presentation 1
Presentation Title:	
Last Modified by:	@ Adam Ryan
People To Attend:	@ Adam Ryan Demo (@ Mayuri Srinivasan (Unlicensed))
Meeting Time:	Insert Time
Agenda:	Update
Meeting Link:	MeetingLinkGoesHere - Discord if Discord

Presentation Grading Reminder

Please keep in mind the marking scheme for the presentation

> Remember to keep in mind the marking scheme for the presentation when designing the structure!

Presentations last for 8 minutes, with 4 minutes for questions and answers.

• Criteria 1: **Awareness of Audience & Delivery**

Excellent	Very Good	Fair	Poor
– Builds trust and holds attention of the audience; – Fluctuation in volume and inflection help to maintain audience interest and emphasise key points.	– Quick recovery from minor mistakes; – Fairly consistent use of direct eye contact with the audience. – Satisfactory variation of volume and inflection.	– Some tension or indifference apparent – Occasional but unsustained eye contact with audience; – Uneven volume with little or no inflection.	– Nervous tension obvious – No effort to make eye contact with audience; – Low volume and/or monotonous tone cause audience to disengage.

• Criteria 2: **Technical Content & Organisation**

Excellent	Very Good	Fair	Poor
– Major technical details summarised and effectively persuades the audience re the validity of the approach taken. – Conclusions/ideas are supported by evidence where appropriate. – Excellent organisation and use of illustrations.	– Clear description of approach taken although audience may not be persuaded of it's validity/appropriateness. – Reasonable attempt at providing evidence to support claims. – Very good structure, consistency and use of illustrations.	– Some major technical details left un-clear. – Includes very thin data or evidence in support of ideas or conclusions; – Minor issues in relation to the structure, consistency and general organisation of the presentation.	– Fails to effectively persuade the audience of usefulness of the approach taken. – Very weak or no support of technical proposal by way of examples, facts, and/or statistics; – No major ideas proposed and/or audience left confused.

- Criteria 3: **Timing & Handling of Questions**

Excellent	Very Good	Fair	Poor
– Excellent answers given for all questions. – Excellent use of allocated time.	– Clear and satisfactory answers given for all questions. – Finished presentation of relevant content on time.	– Reasonable answers given to all most questions. – Covered most of the relevant content in the allotted time.	– Incomplete or vague answers for the majority of questions. – Fails to cover the main points of the presentation in the allotted time.

Presentation - Name

(this should be replaced with a brief overview of the presentation)

Background Reading

This section should recap the key background reading for each section.

Section 1 - Name/Key Idea

Section 1 Description

Section 2 - Name/Key Idea

Section 2 Description

Section 3 - Name/Key Idea

Section 3 Description

Section 4 - Name/Key Idea

Section 4 Description

Structure

This section should broadly touch what the structure is.

Section 1 - Name/Key Idea

Section 1 Description

Section 2 - Name/Key Idea

Section 2 Description

Section 3 - Name/Key Idea

Section 3 Description

Section 4 - Name/Key Idea

Section 4 Description

Additional Notes

This section should detail key notes on the presentation

Upcoming Presentations

> Remember, these are the upcoming presentations and schedule!

Date	Presentation Title
16-June-2021	Initial Big Picture
30-June-2021	Detailed Requirements and Architecture
14-July-2021	Prototype and Data Analytics
11-August-2021	Pre-final Prototype
25-August-2021	Final Presentation

Final Report

This page should contain all pages for the final report

Reviewed Literature

This page is to track the key documents which were read.

	Article	Journal	Link	Topic	Description	Useful?
1						
2						
3						
4						

Reviewed Apps

This page should contain a list of similar applications, both web apps and also mobile apps to provide context.

App Type (mobile /web)	App	Confluence Link	App Link	Key Features	Team (Good/Bad)	Rating if applicable	Sample Comments

Mobile	Dublin Bus App	Dublin Bus App	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjFtoe035bxAhWRRUEAHZ2NDV8QFjAAegQIAxAD&url=https%3A%2F%2Fapps.apple.com%2Fie%2Fapp%2Fdublin-bus%2Fid450455266&usg=AOvVaw3o9QPP_hC3Hj6HN8wzR-OH	Real Time Bus Info. Official App. Journey Planner.	Bad - UX, Wrong Times, Inconsistent. Good - Efficient, finds stops near you.	1.9 (298 Reviews)	“So so so so so bad. Genuinely don’t know how you can keep such a poor app in use. It’s actually counterproductive. Missing buses because the RTPI on this app is on drugs. Buses never correspond with the RTPI. Sometimes the RTPI won’t even load the times . FORCES PEOPLE TO BE LATE MO MATTER WHAT. “Bus in 10 minutes” actually means a bus in 45 minutes. Actually an insult to have this bus app still on the market . Many people suffer everyday because this is the only alternative.” “If you want to browse stop locations near you one by one and find lots of stops for busses you have no intention of taking, this app is great. If you want to find a way to get from A to B in Dublin on a bus, this is useless. Use google maps, Apple maps, or talk to a random person on the street. All will be more helpful. User interface design was not a priority for this poor app, which is not a surprise for Dublin Bus, who don't even post decent maps of their routes at physical stops. This app fits right into their “if you have to ask where this bus goes, you shouldn't be taking it" attitude.” “I don't know why people are complaining about this app not working. I've been using it fun for over a year now. I plan when I need to leave my house using it. Yes sometimes a bus doesn't show up but for me that very rare. This app displays exactly what's on the live times at the bus stop itself. But please update the layout!!”

Mobile	TFI Real Time Planner	TFI Journey Plan App	https://apps.apple.com/ie/app/real-time-ireland/id590250800	Real-time departure information and travel updates from Bus Éireann, Dublin Bus, Go-Ahead Ireland, Iarnród Éireann and Luas services See up to date information so you can tell when your bus/Luas /train will be at a stop or station. - Combines Sources Start and End Time filters.	Good - More options than Dublin Bus. Provides info on multiple transport services. User retention for ease of use. Bad - Confusing UX. Info not reliable - travel times inaccurate (disappear, wrong, etc.). Not trusted by Users.	1.5 (449 reviews)	“This app has left me standing in the rain for a bus that never comes more times than I can count. The information it provides is often completely incorrect, with times to arrival counting down according to the timetable regardless of whether or not the bus is actually due or not, resulting in buses disappearing from the list without arriving or arriving far later than the app would have you believe.” “In my experience the real time displayed is not live ‘real time’. The bus that I went to catch was arriving in 4 mins, then ‘due’, it then disappeared off the app and 15 mins after the scheduled arrival time the bus hadn’t shown up.” “Ok, it’s not always reliable, I can live with that. Nor are the timetables at stops. But what’s really annoying is that pop-up message about location settings. No, you don’t need my location, if I’m going to select a stop myself. No, you don’t need to notify me that location settings are off three times in a row!” “Better than Dublin bus app Easy to use and very fast they really fixed this up”
Web/Mobile https://support.google.com/transitpartners/#topic=3521043	Google Maps	Google Travel Planner App	https://www.google.com/maps https://apps.apple.com/us/app/google-maps/id585027354	Street View. Traffic View. Estimated Journey Times and Stop Info. Easy UX. Not just limited to travel. Shows other features on map. Shows multiple transport methods. Info on standard bus occupancy.	Good - Easy UX. Multiple Transport Methods. Multiple bus routes. Estimated times for each. Traffic data.. Bus business. Bad - Data Privacy Concerns.	4.7 (3.9M reviews)	(searching for user reviews specific to Dublin Bus but tricky to find)
Browser (mobile and desktop)	Browser	Dublin Bus website	Dublin Bus	Simple layout (list-based) Simple data presentation (bus number, destination, time of arrival)	Bad - map view doesn’t work (not updated in years), outdated design		

Dublin Bus App

Please find screenshots below of the Dublin Bus App RTI interface.

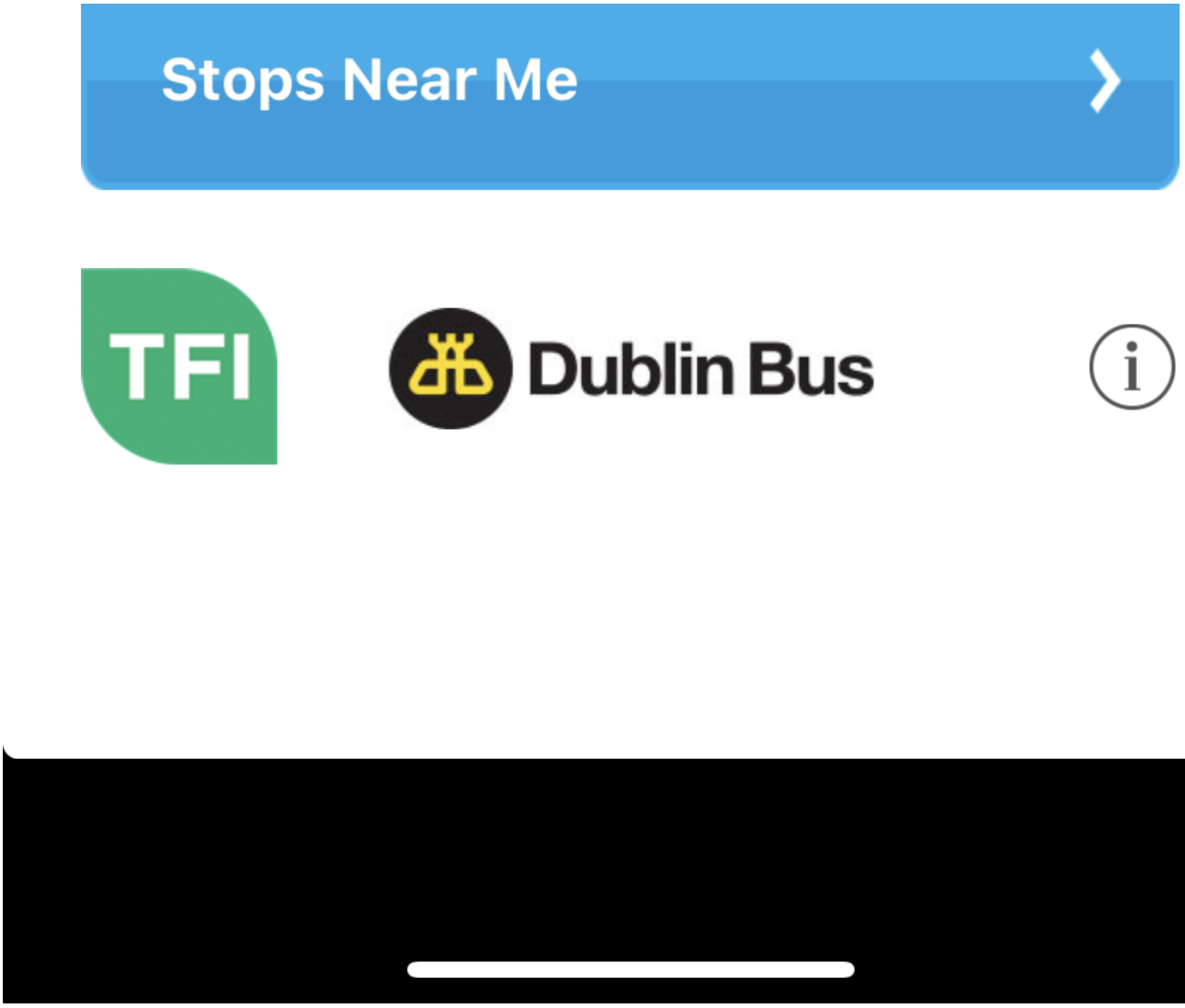
Menu

For real time bus information please select your search preference

Search By Stop Number

Search By Route

Search By Address



Search by Stop Number

Stop 35

St. Pappin's Road, Dean Swift Road

Data Refreshed at 13:16

Tá ár dtoilleadh ar bord teoranta do 50% den lántoilleadh. Caith clúdach aghaidhe agus bain úsáid as Cárta TFI Leap. Íoslódáil Aip Breisithe Creidmheasa Leap ar iPhone agus ar Android.

On board capacity is limited to 50% Wear a face covering & use a TFI Leap Card. Download the Leap Top-Up App on iPhone and Android.

Less info

Bus	Destination	Time
11	Sandyford B.D.	14 mins
11	Sandyford B.D.	44 mins

Show on Map

Entered Stop

Stop 35

St. Pappin's Road, Dean Swift Road

Data Refreshed at 13:16

Tá ár dtoilleadh ar bord teoranta do 50% den lántoilleadh. Caith clúdach aghaidhe agus bain úsáid as Cárta TFI Leap. Íoslódáil Aip Breisithe

Creidmheasa Leap ar iPhone agus ar Android.

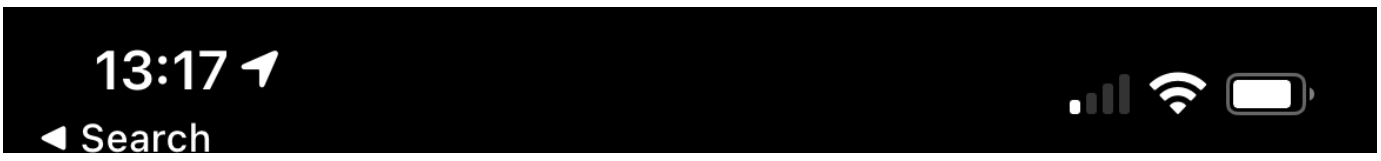
On board capacity is limited to 50% Wear a face covering & use a TFI Leap Card. Download the Leap Top-Up App on iPhone and Android.

Less info

Bus	Destination	Time
11	Sandyford B.D.	14 mins
11	Sandyford B.D.	44 mins

Show on Map

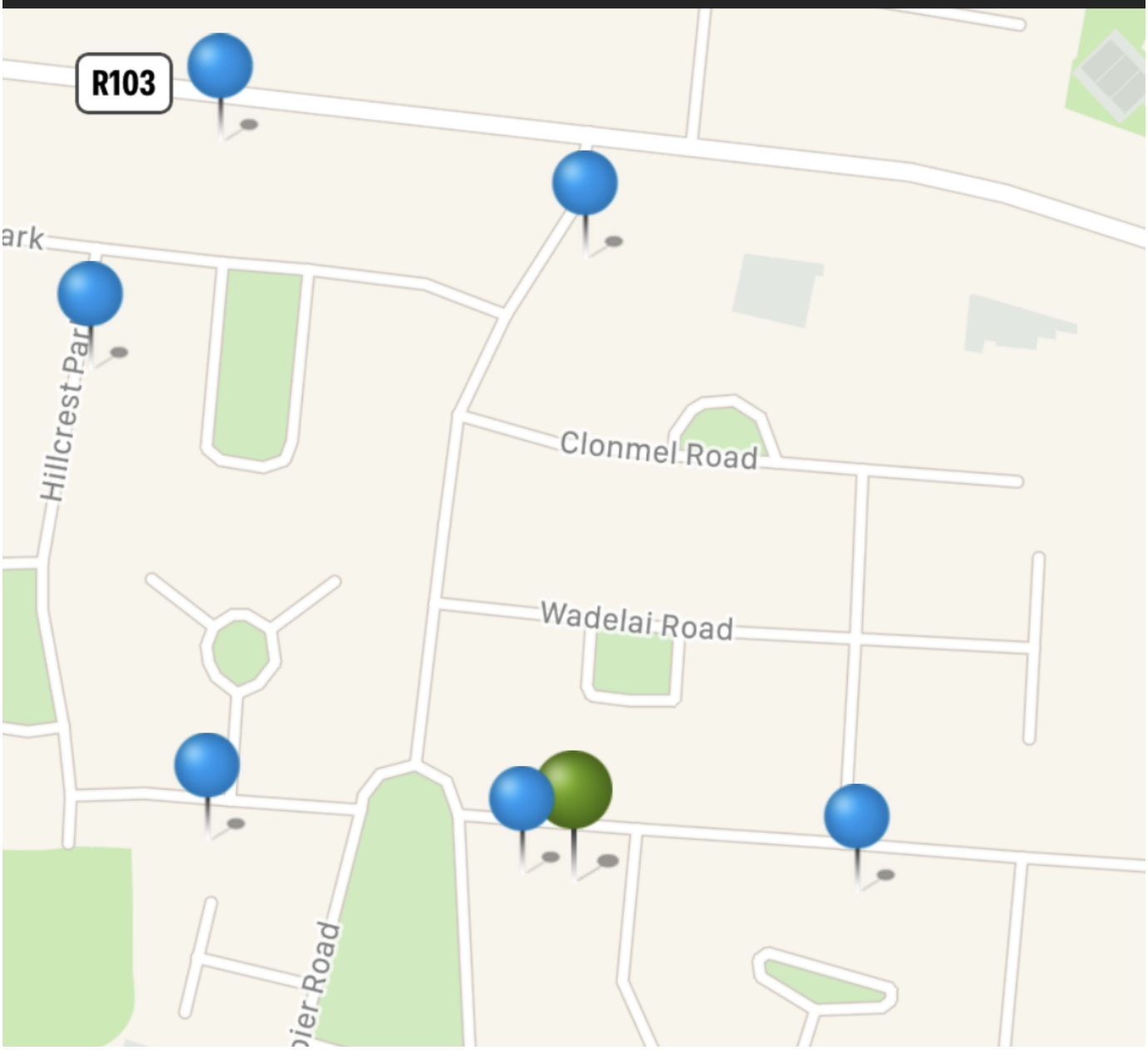
View Stop on Map

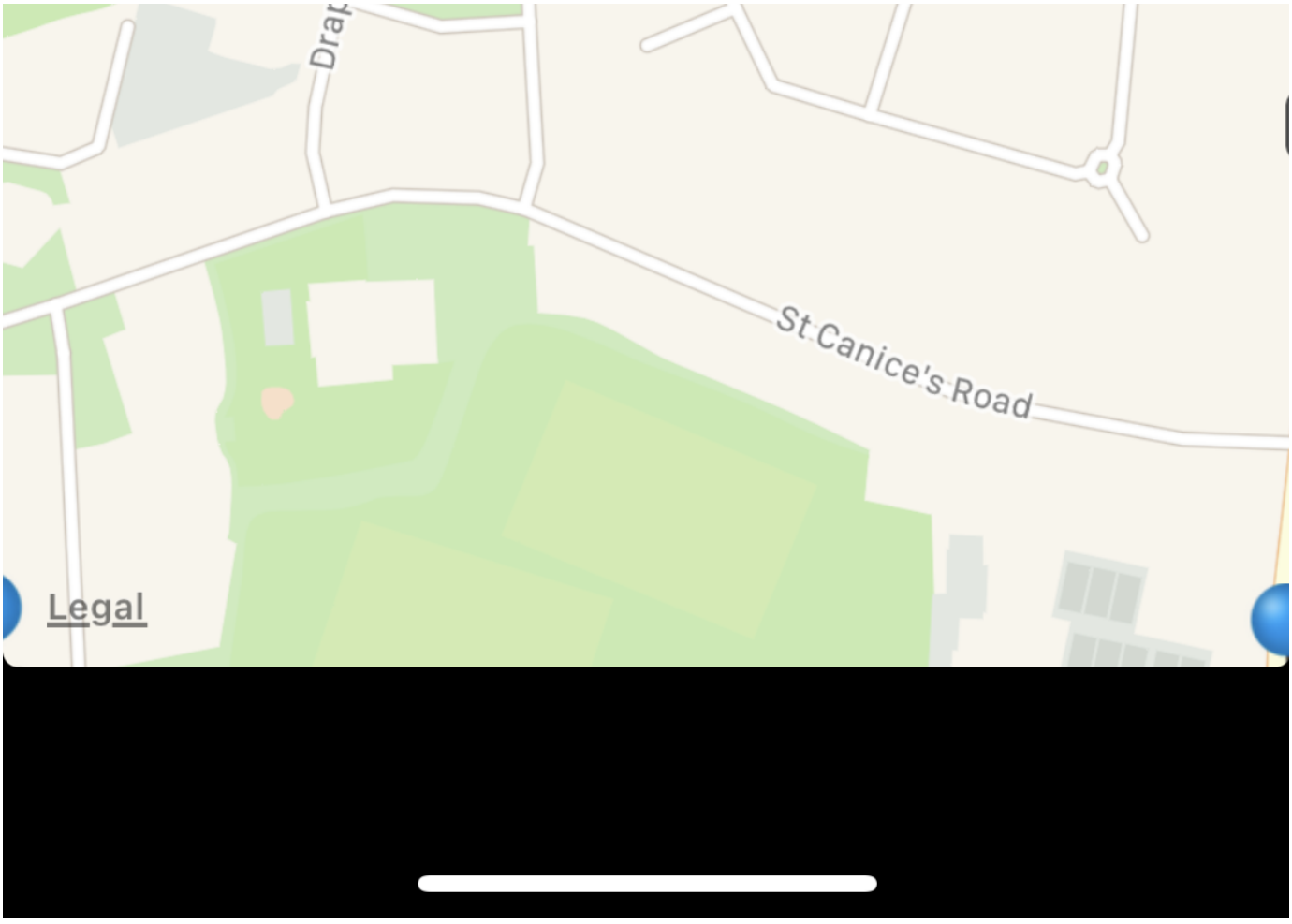


Back

Stop 35

Get Walking Directions to St. Pappin's Road, Dean Swift Road





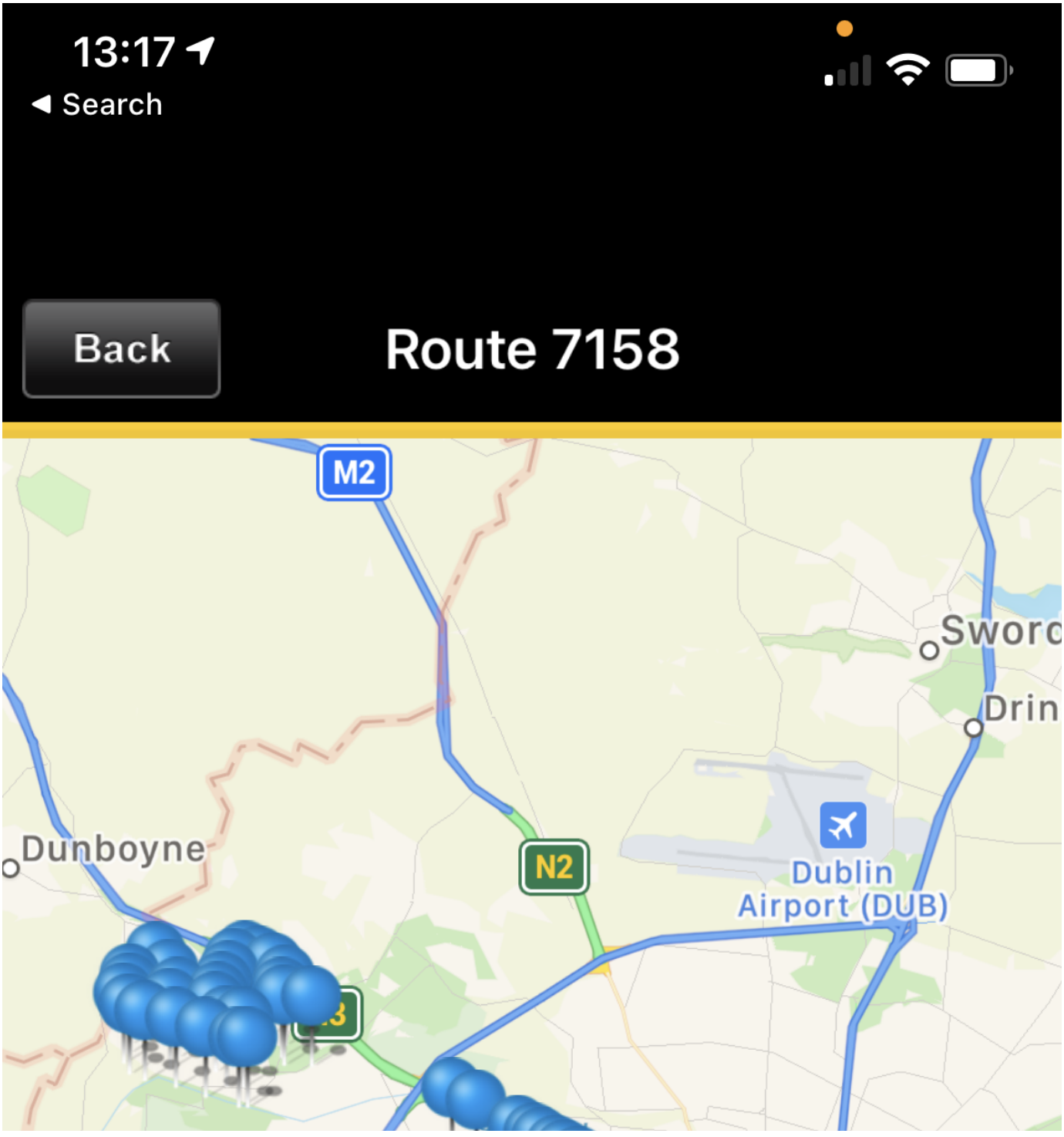
Search by Route

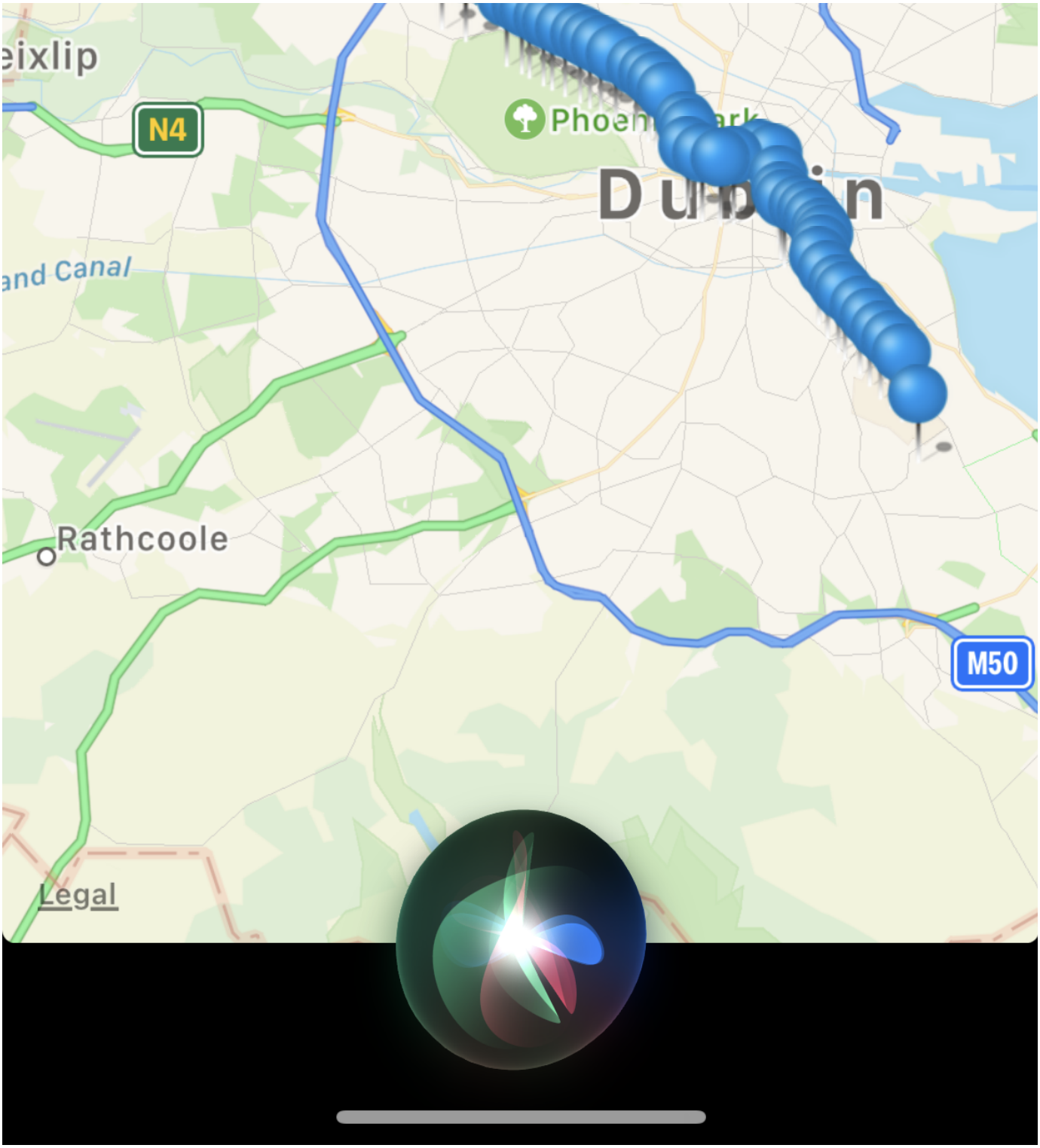
Select Direction

Towards UCD

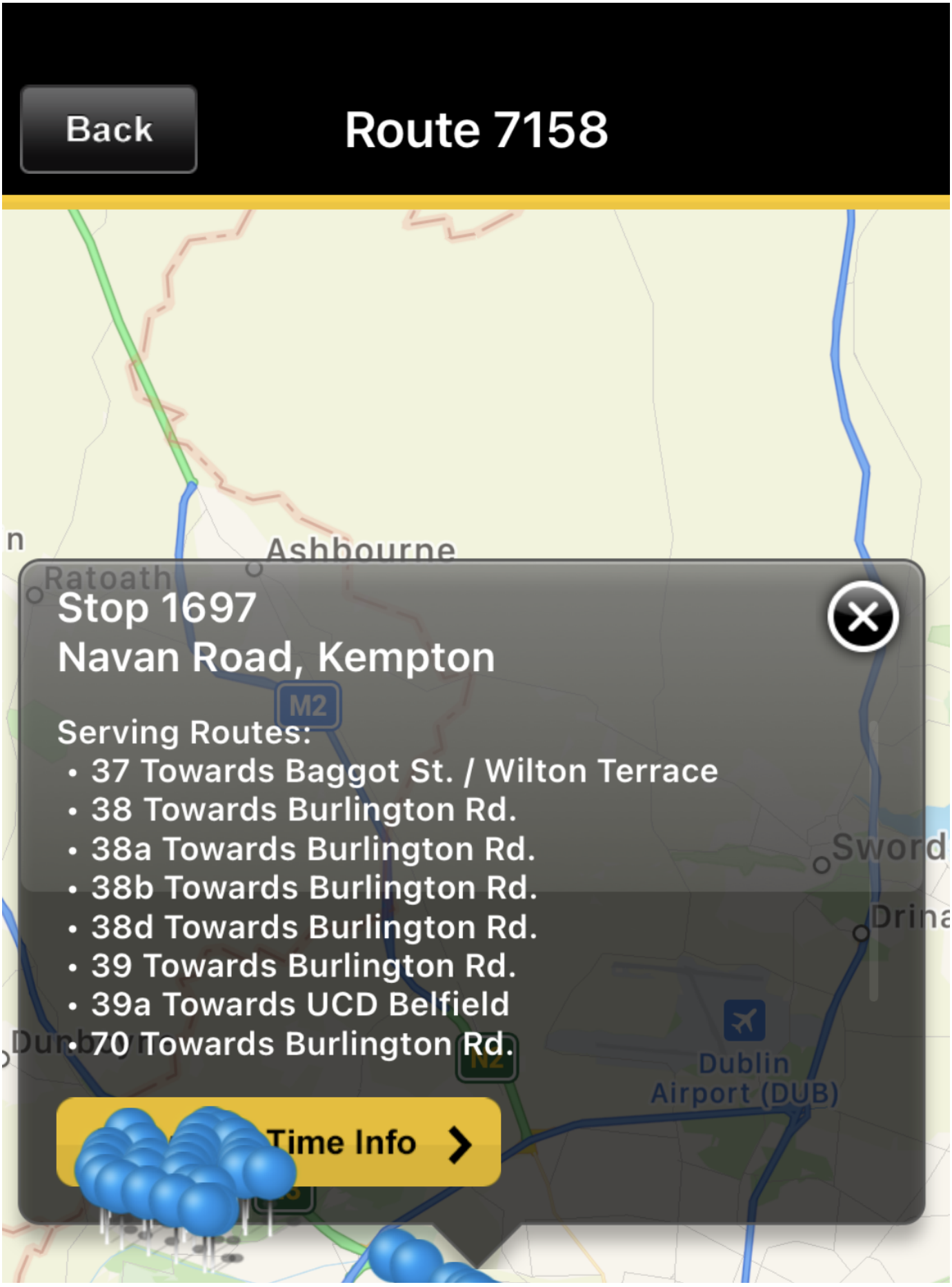
Towards Ongar

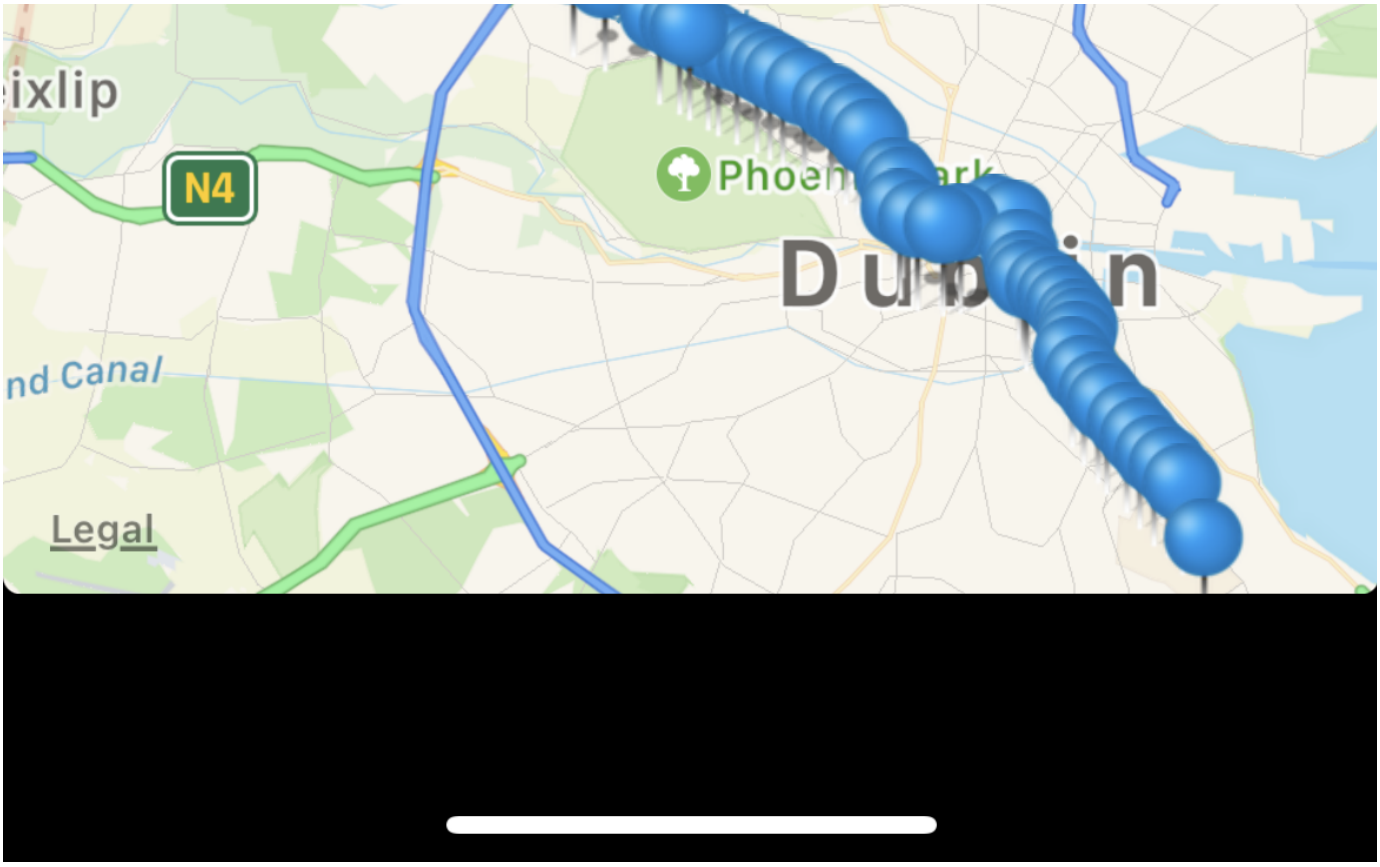
Enter Route


13:17
Search
Back
Route 7158
M2
N2
Dunboyne
Dublin
Airport (DUB)

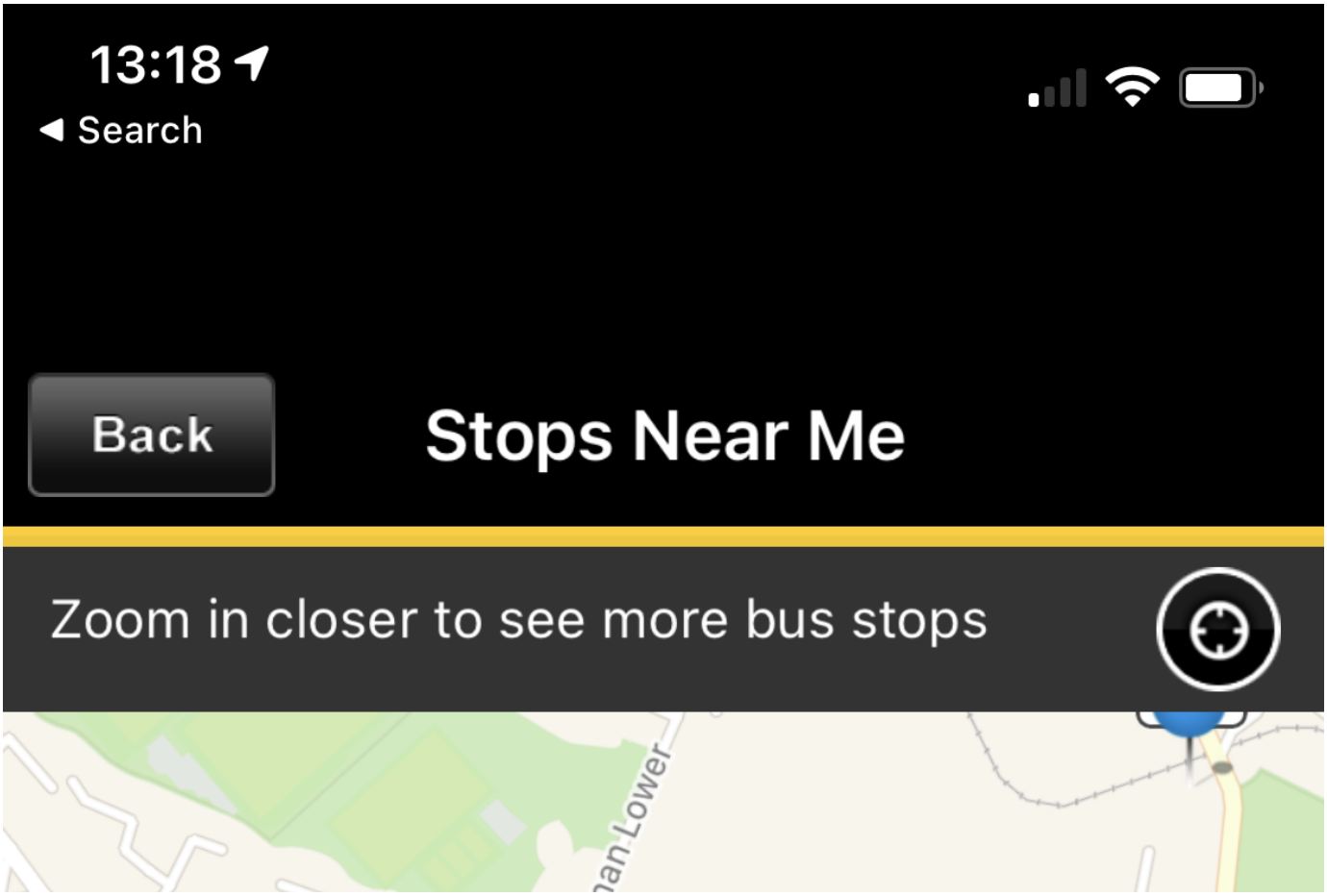


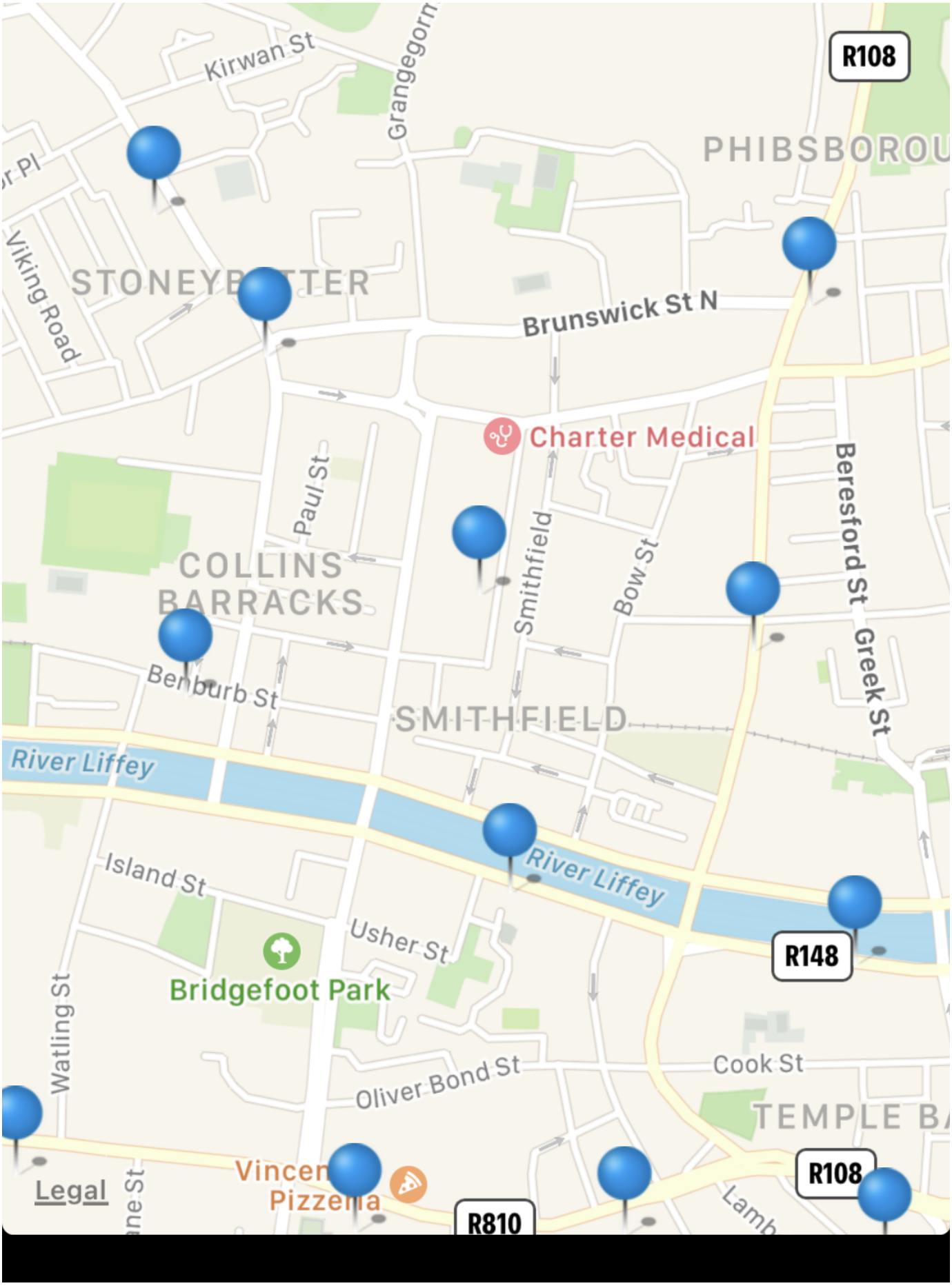
Click On Stop


Back
Route 7158
Ashbourne
Ratoath
Stop 1697
Navan Road, Kempton
M2
Serving Routes:
• 37 Towards Baggot St. / Wilton Terrace
• 38 Towards Burlington Rd.
• 38a Towards Burlington Rd.
• 38b Towards Burlington Rd.
• 38d Towards Burlington Rd.
• 39 Towards Burlington Rd.
• 39a Towards UCD Belfield
• 70 Towards Burlington Rd.
Sword
N2
Dublin
Airport (DUB)
Time Info

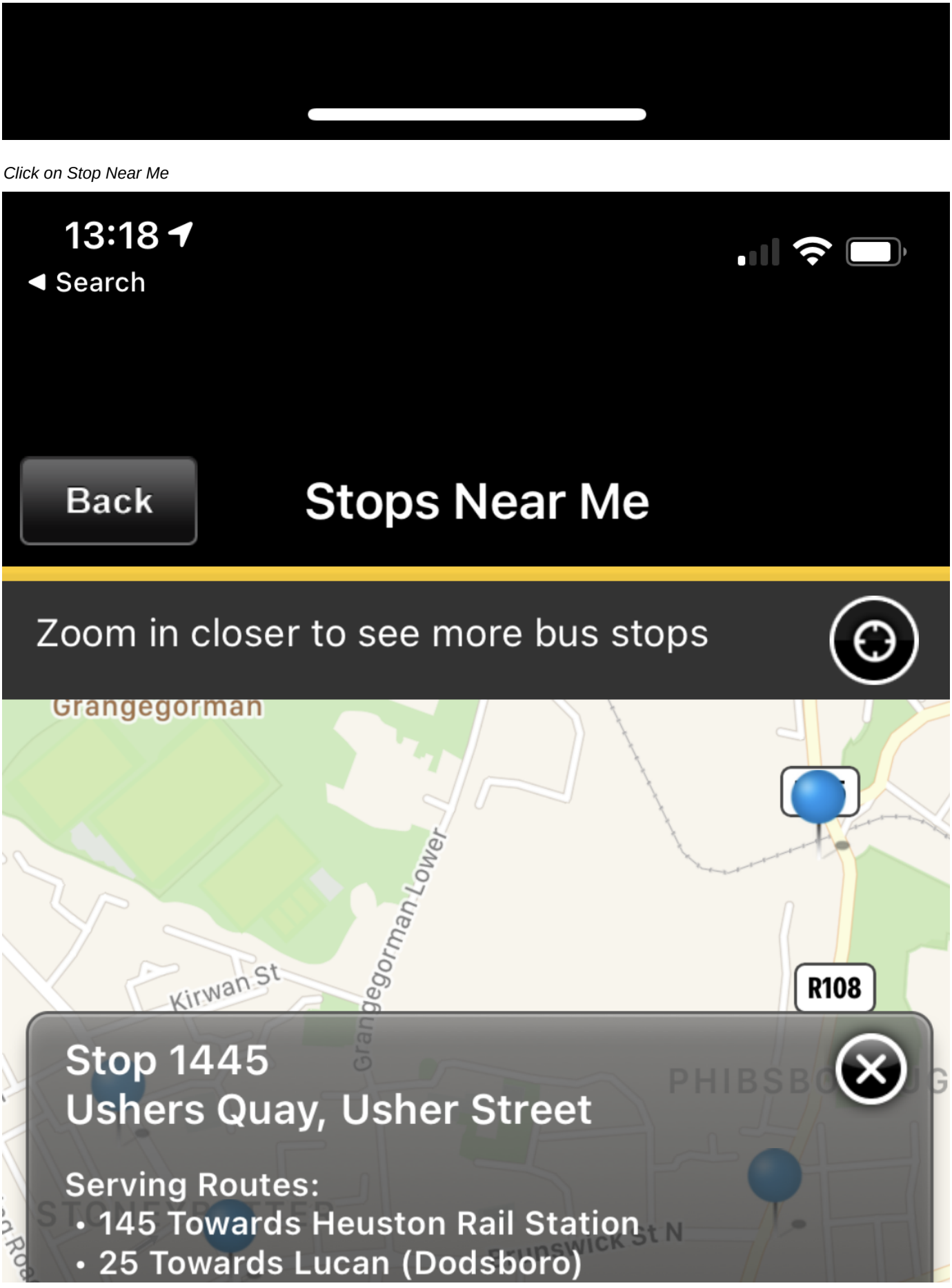


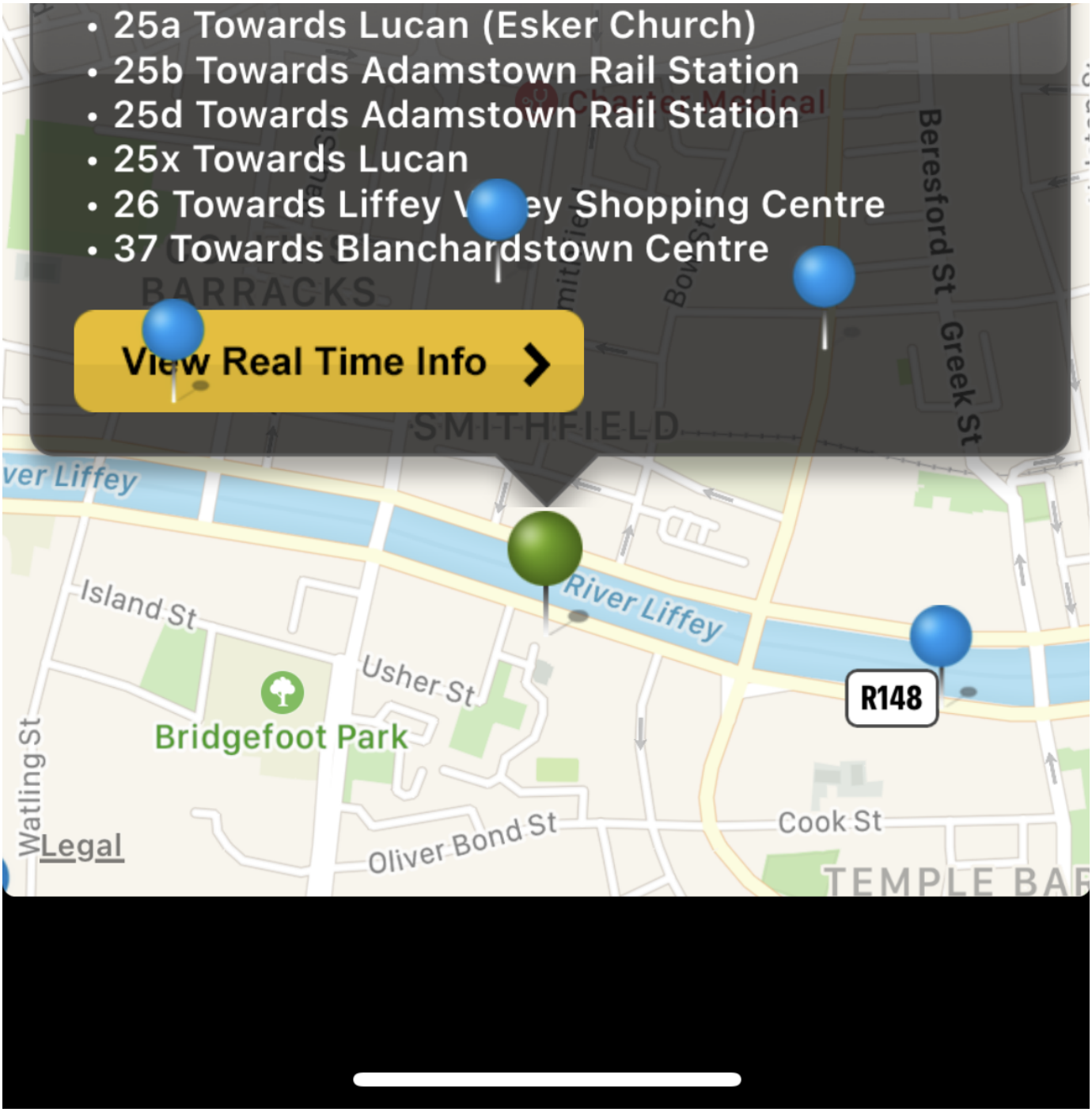
Stops Near Me




Kirwan St
Grangegorm
R108
PHIBSBOROU
Viking Road
Brunswick St N
Charter Medical
Paul St
Smithfield
Bow St
Beresford St
Greek St
COLLINS
BARRACKS
Benburb St
SMITHFIELD
River Liffey
River Liffey
Island St
Usher St
R148
Bridgefoot Park
Watling St
Oliver Bond St
Cook St
TEMPLE BA
Legal
Vincen
Pizzeria
R810
R108
Lamb

Click on Stop Near Me

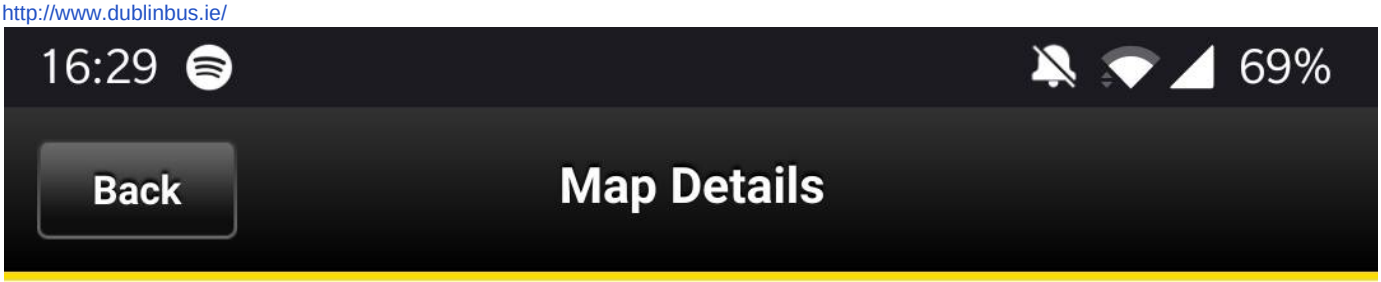


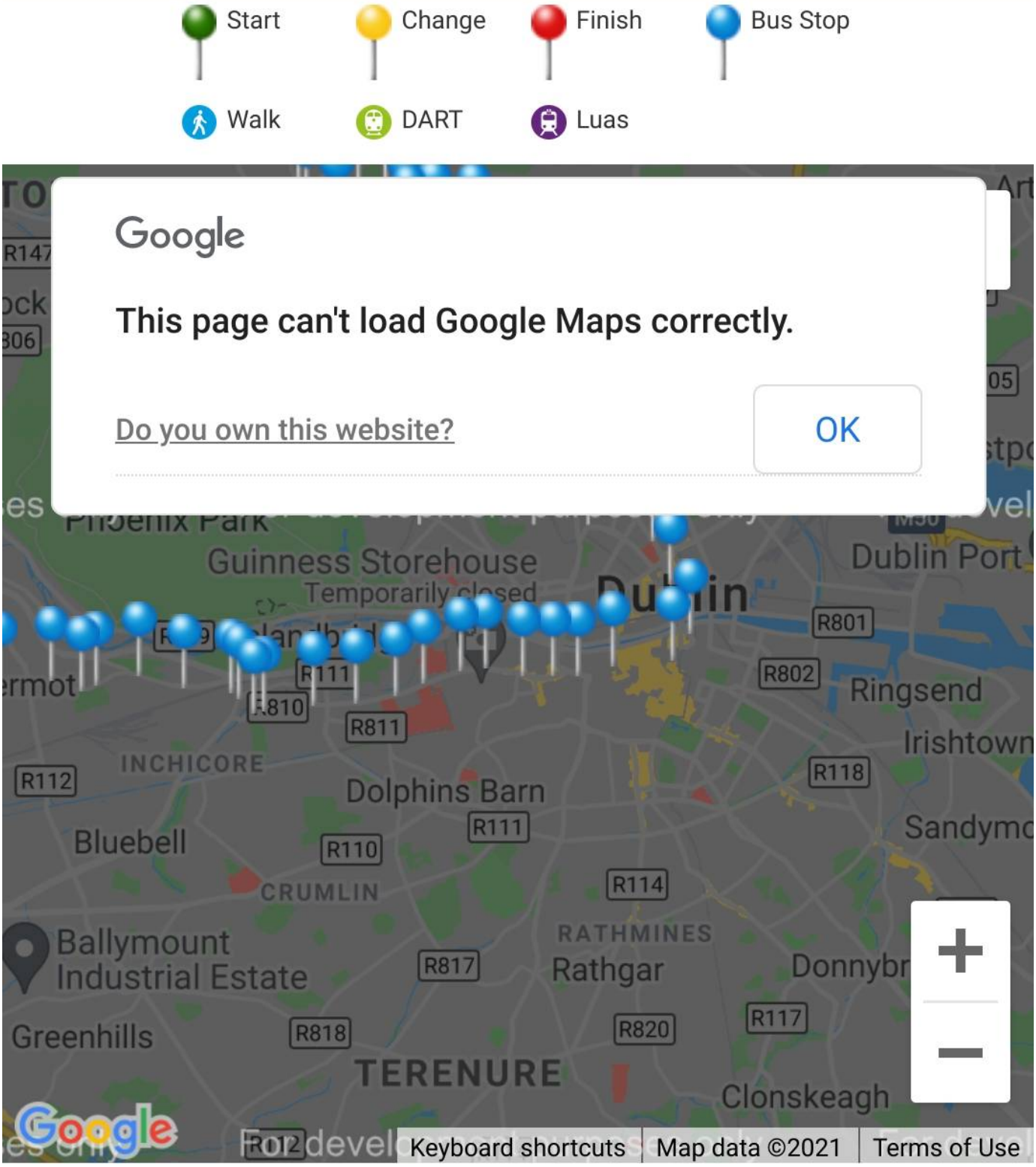


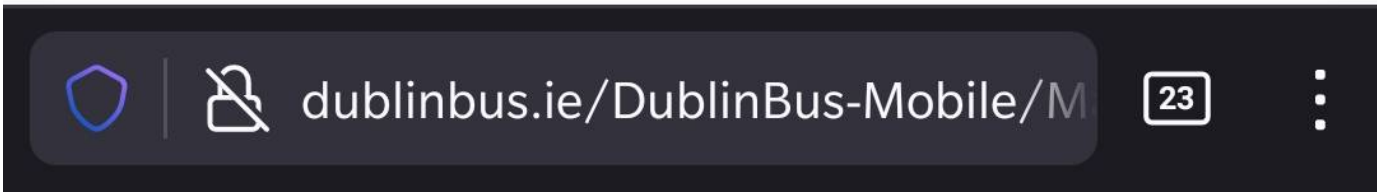
Dublin Bus website

A screenshot of the Dublin Bus mobile interface through a web browser; the system has not been updated in years, as the Google Maps implementation is in developer mode.

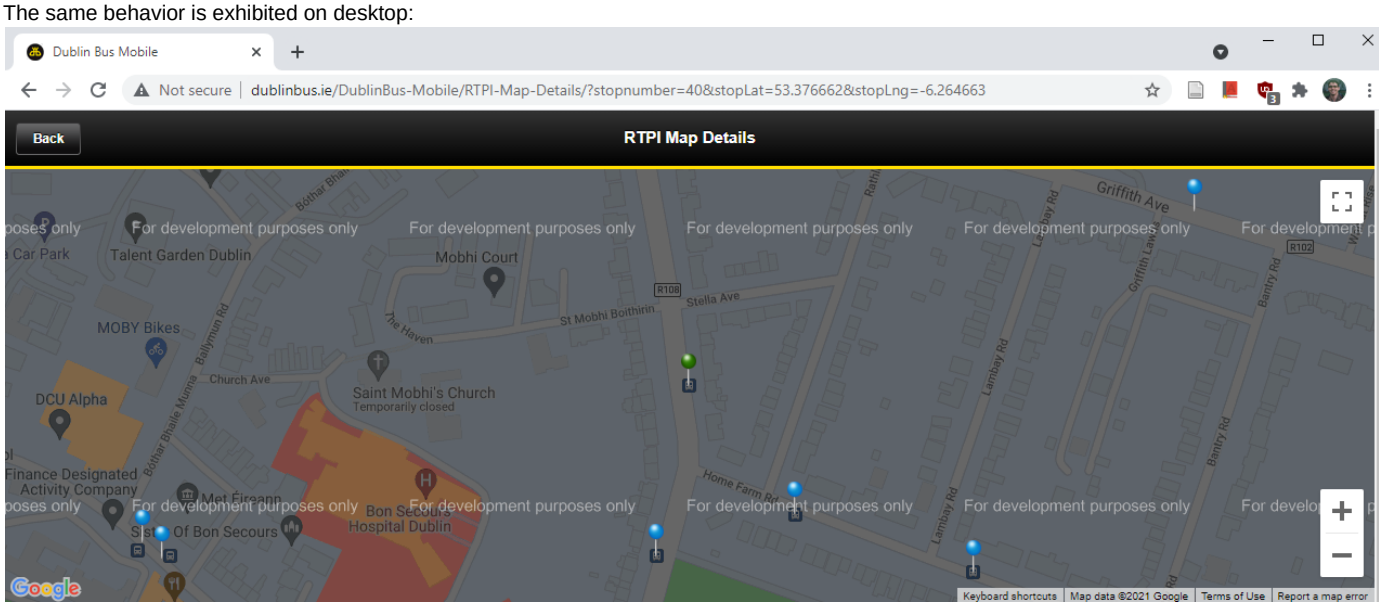
http://www.dublinbus.ie/




Start
Change
Finish
Bus Stop
Walk
DART
Luas
Google
This page can't load Google Maps correctly.
Do you own this website?
OK
Guinness Storehouse
Temporarily closed
Dublin Port
Dublin
Ringsend
Irishtown
INCHICORE
Dolphins Barn
Bluebell
CRUMLIN
RATHMINES
Rathgar
Ballymount
Industrial Estate
Greenhills
TERENURE
Clonskeagh
R801
R802
R118
R111
R810
R811
R112
R110
R114
R817
R818
R820
R117
Keyboard shortcuts
Map data ©2021
Terms of Use



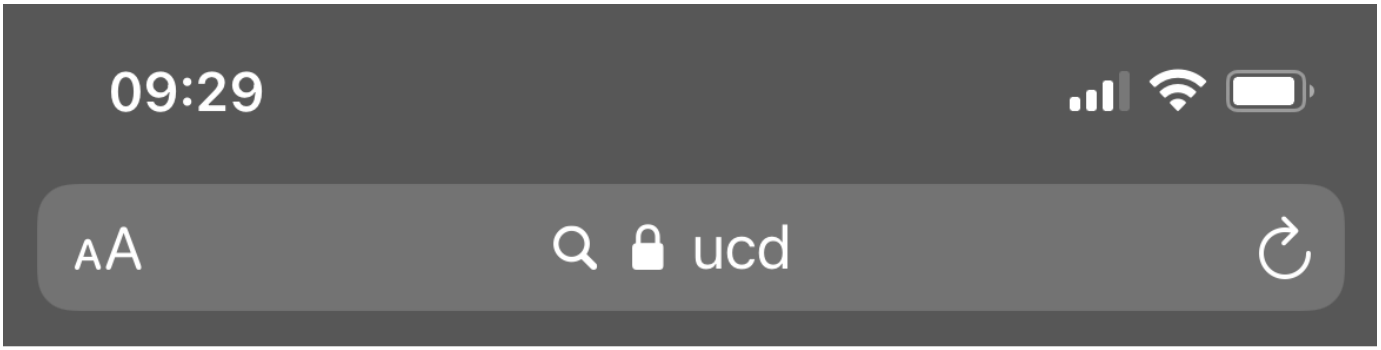
The same behavior is exhibited on desktop:



Google Travel Planner App

Please find screenshots of Google Map's Travel Planning App

Search Location



All Maps Images News Videos Shoppin

See results about

University College Dublin …
Football club

University College Dublin

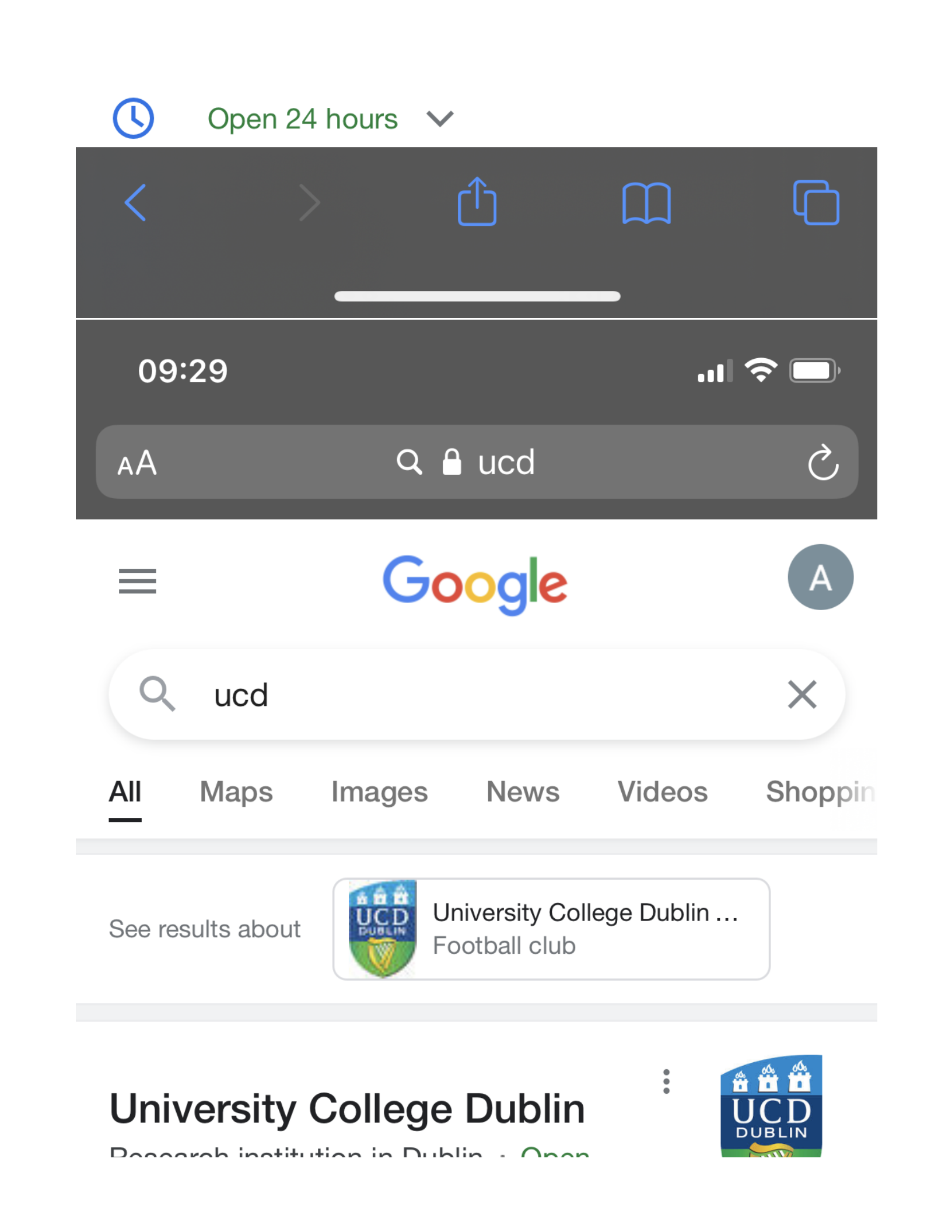
Research institution in Dublin · Open

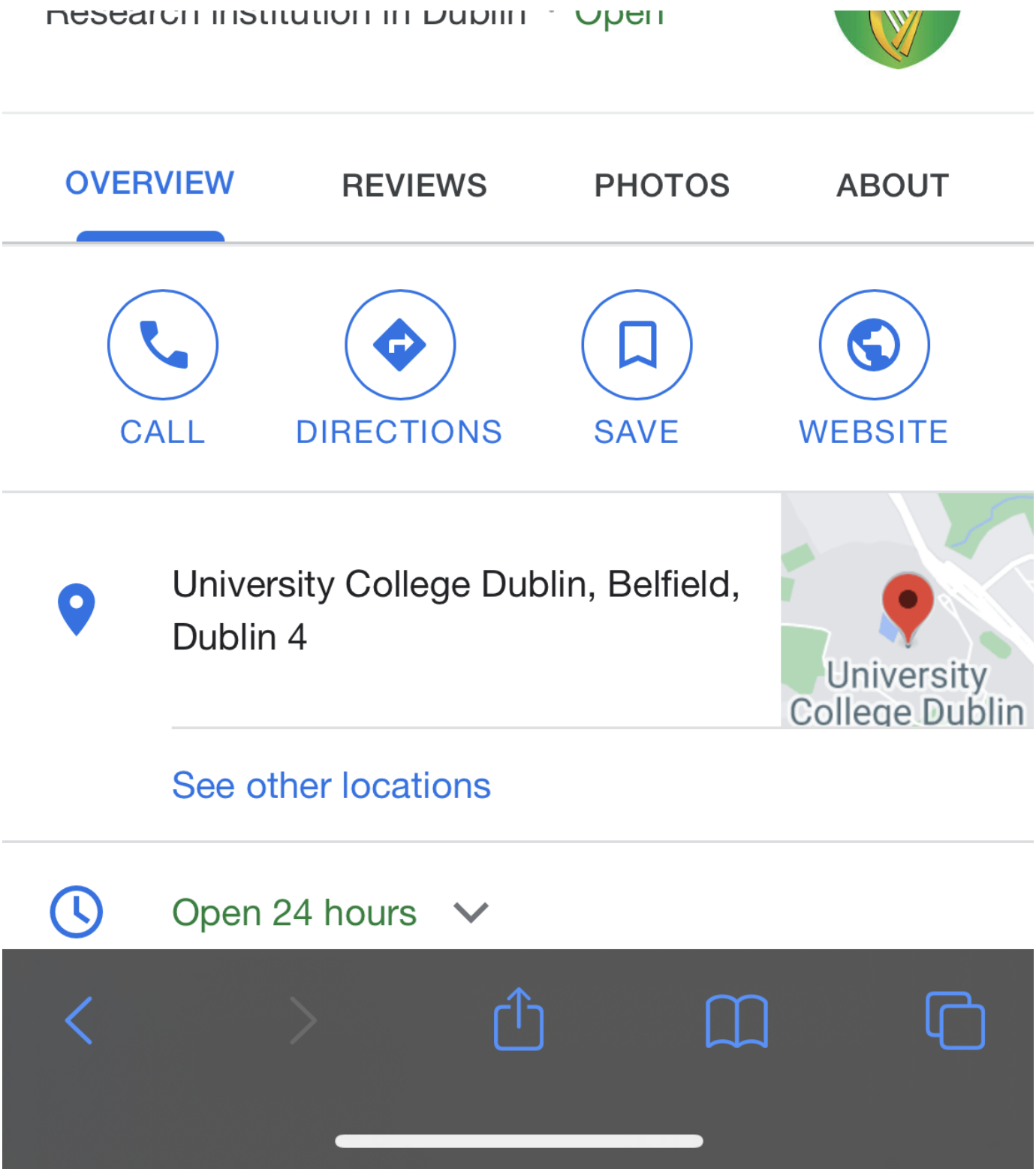
OVERVIEW REVIEWS PHOTOS ABOUT

CALL DIRECTIONS SAVE WEBSITE

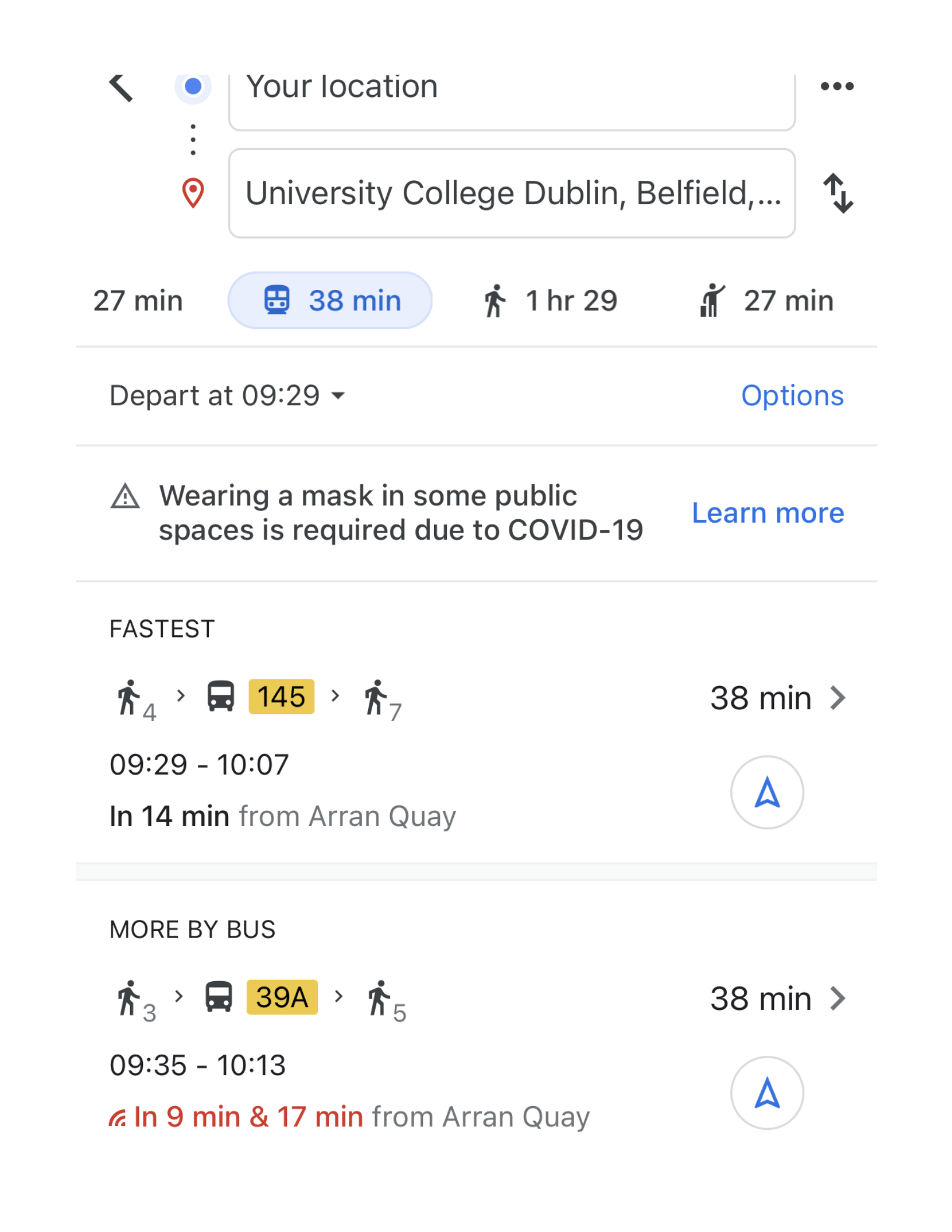
University College Dublin, Belfield, Dublin 4

See other locations


Open 24 hours
09:29
AA
ucd
Google
A
ucd
All
Maps
Images
News
Videos
Shoppin
See results about
UCD DUBLIN
University College Dublin …
Football club
University College Dublin
Research institution in Dublin · Open
UCD DUBLIN



Click Directions


Your location
University College Dublin, Belfield,...
27 min
38 min
1 hr 29
27 min
Depart at 09:29
Options
Wearing a mask in some public spaces is required due to COVID-19
Learn more
FASTEST
4
145
7
38 min
09:29 - 10:07
In 14 min from Arran Quay
MORE BY BUS
3
39A
5
38 min
09:35 - 10:13
In 9 min & 17 min from Arran Quay

09:43 - 10:21

In 18 min & 22 min from Arran Quay

ALSO CONSIDER

FREENOW 26 min

1 min away €13–15

Ad · *Estimate for Taxi*

09:29

◀ Safari

Your location

University College Dublin, Belfield,...

27 min | 38 min | 1 hr 29 | 27 min

Depart at 09:29 Options

Wearing a mask in some public

spaces is required due to COVID-19 Learn more

FASTEST

4 › 145 › 7 38 min ›

09:29 - 10:07

In 14 min from Arran Quay

MORE BY BUS

3 › 39A › 5 38 min ›

09:35 - 10:13

In 9 min & 17 min from Arran Quay

4 › 25A 26 66 67 › › 5 38 min ›

09:43 - 10:21

In 18 min & 22 min from Arran Quay

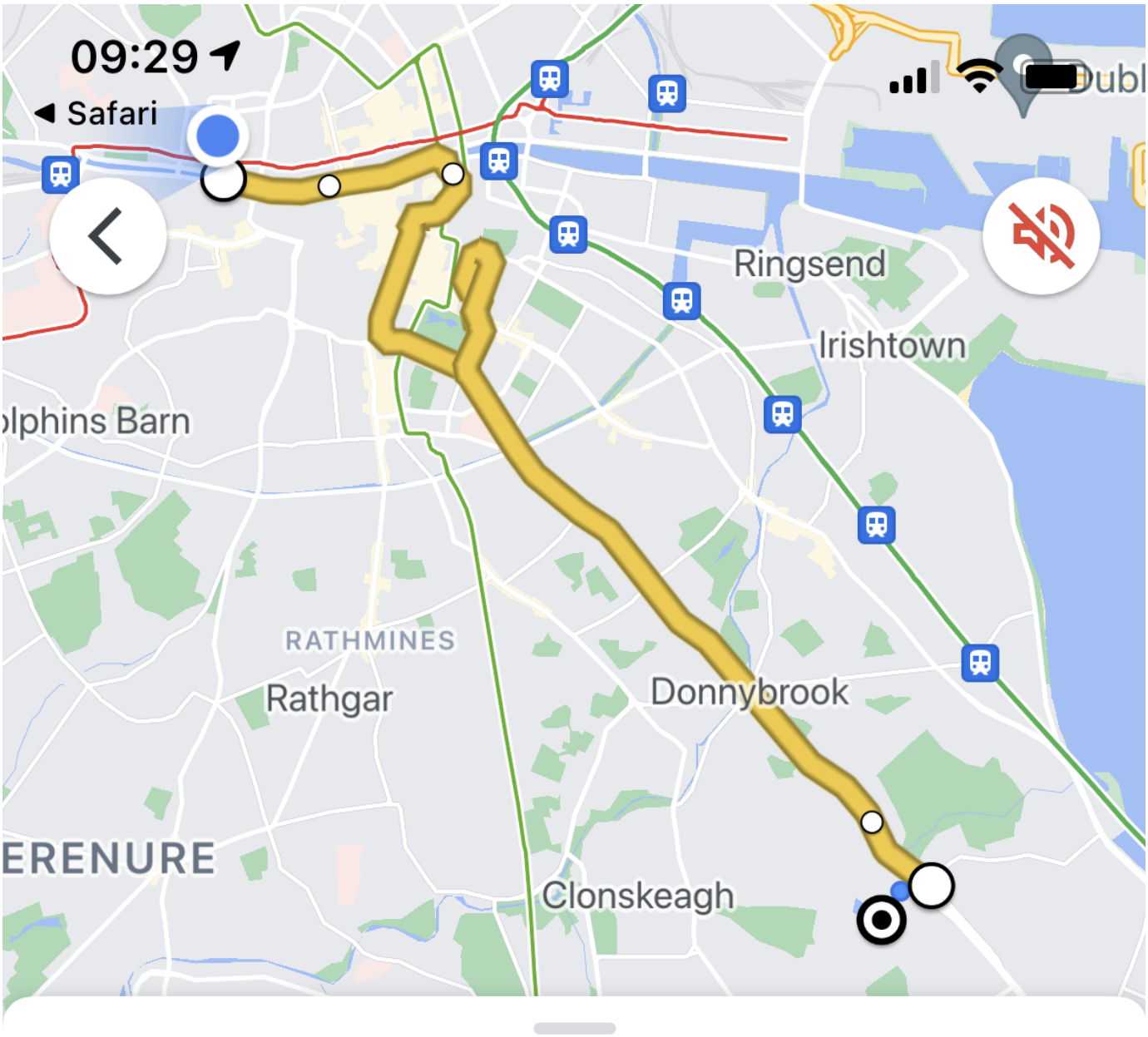
ALSO CONSIDER

FREENOW 26 min ›

1 min away €13–15

Ad · *Estimate for Taxi*

Select Route



4 › 145 › 7

Your location 09:29

Walk 4 min (300 m)

Live View

Arran Quay

145 Ballywaltrim

09:33

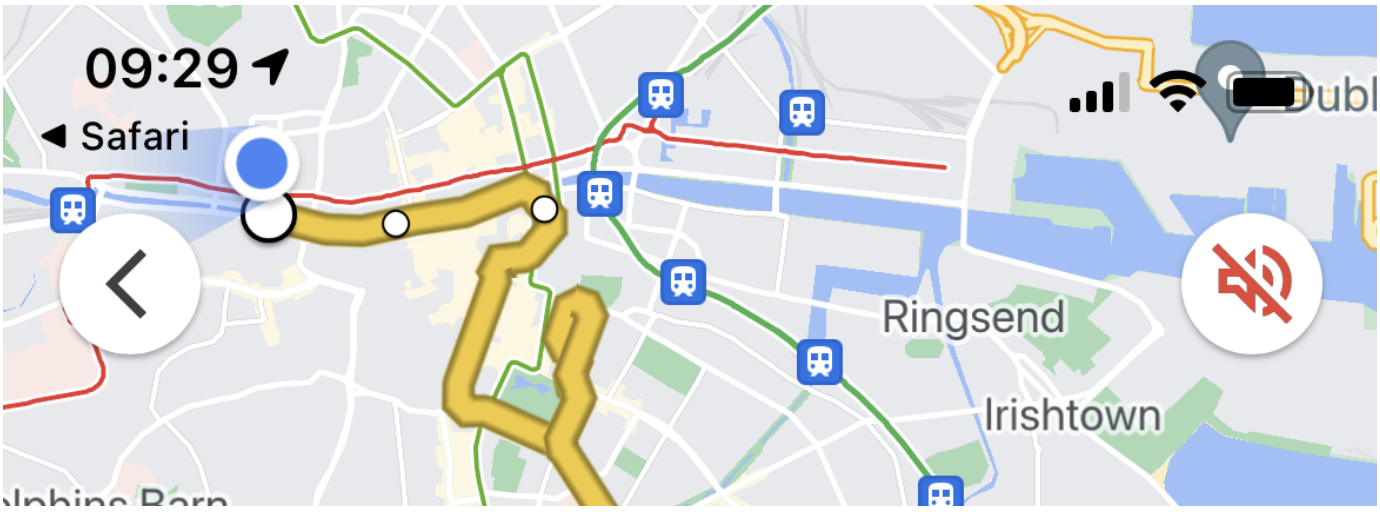
Scheduled · in 3 min

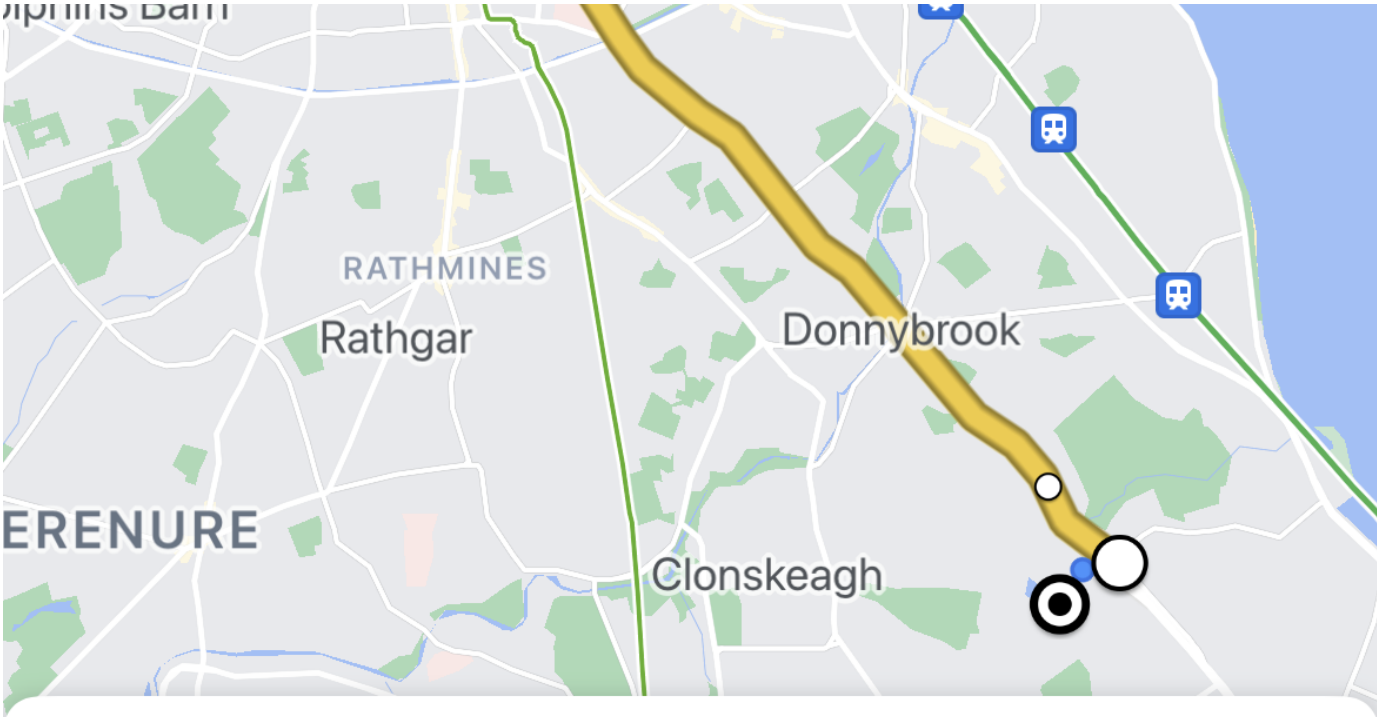
Also in 13 min & 23 min

Start

Pin

10:07 · 38 min





4 › 145 › 7

Your location 09:29

Walk 4 min (300 m)

Live View

Arran Quay

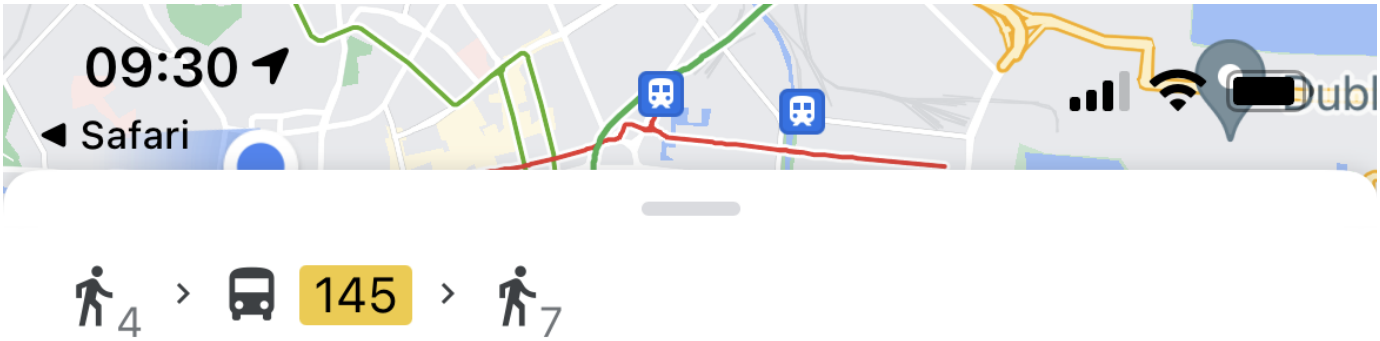
145 Ballywaltrim 09:33

Scheduled · in 3 min

Also in 13 min & 23 min

Start Pin 10:07 · 38 min

Expand Route Details



4 › 145 › 7

Arran Quay

145 Ballywaltrim

Scheduled · in 3 min

09:33

Other departures

Ballywaltrim 7 min ago

Ballywaltrim 09:33

Ballywaltrim 09:43

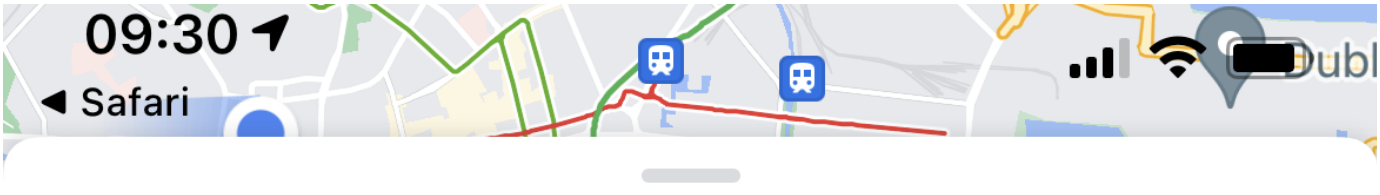
Ballywaltrim 09:53

More departures

Ride 21 stops (27 min)

- Four Courts, stop 1478
- Ormond Quay Upper
- Bachelors Walk, stop 7622
- D'Olier Street
- Nassau Street, stop 406
- Kildare Street, stop 747
- Stephen's Green East

Start | Pin | 10:07 · 38 min



4 › 145 › 7

Arran Quay

145 Ballywaltrim

Scheduled · in 3 min

09:33

Other departures

Ballywaltrim	7 min ago
Ballywaltrim	**09:33**
Ballywaltrim	09:43
Ballywaltrim	09:53

More departures

Ride 21 stops (27 min)

Four Courts, stop 1478

Ormond Quay Upper

Bachelors Walk, stop 7622

D'Olier Street

Nassau Street, stop 406

Kildare Street, stop 747

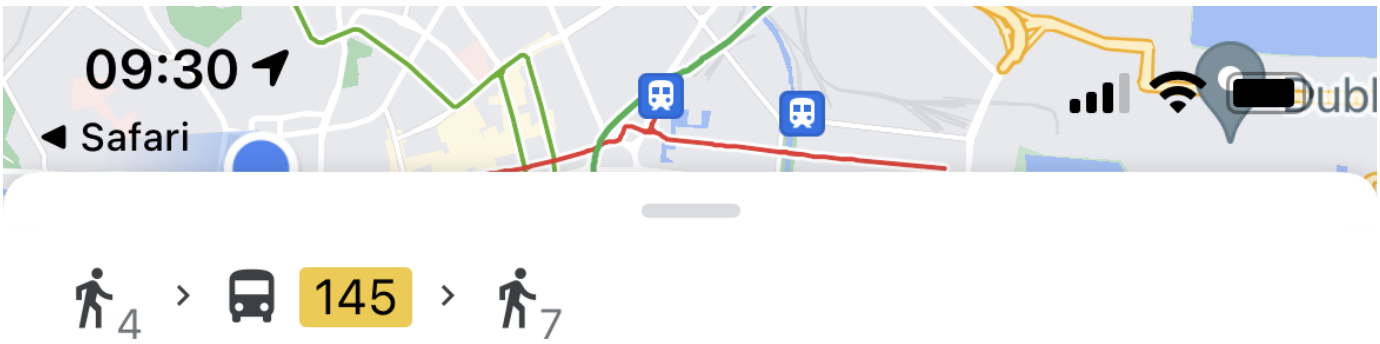
Stephen's Green East

Start

Pin

10:07 · **38 min**

Final Route Options



Donnybrook, Bloomfield Avenue

Donnybrook Village

Donnybrook

Donnybrook (Stadium)

Donnybrook Church

Teresian School, stop 761

Dublin (RTE)

Belfield Court

Booterstown, Woodbine Road 10:00

9 a.m.: Usually not busy

Walk 7 min (500 m)

University College Dublin 10:07

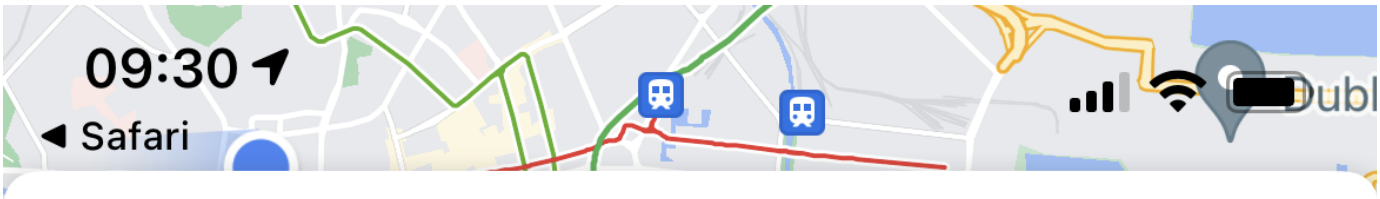
Belfield, Dublin 4

Add to Calendar

Remind you to leave on time

Agency info

Start | Pin | 10:07 · 38 min



4 › 145 › 7

Donnybrook, Bloomfield Avenue

Donnybrook Village

Donnybrook

Donnybrook (Stadium)

Donnybrook Church

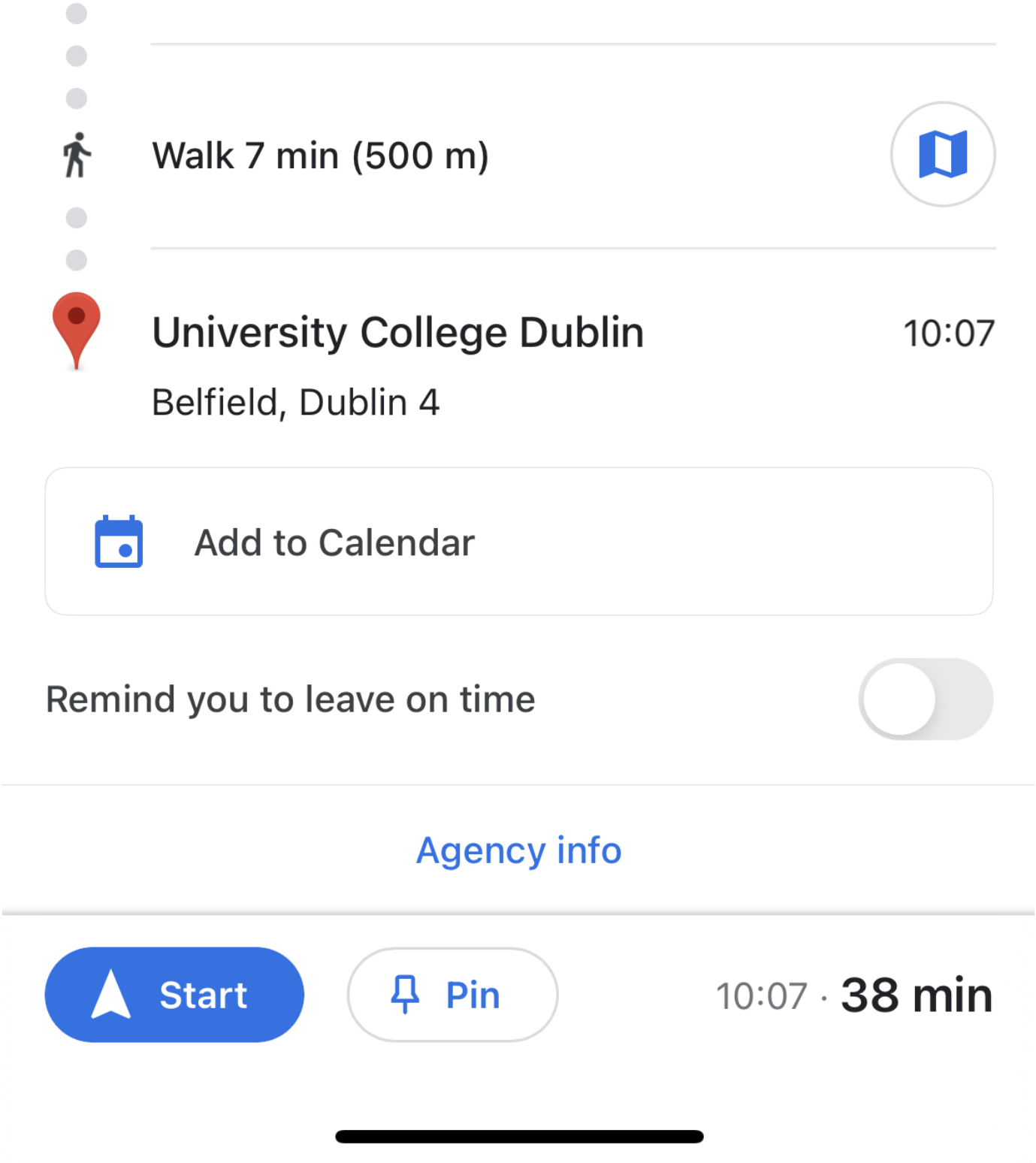
Teresian School, stop 761

Dublin (RTE)

Belfield Court

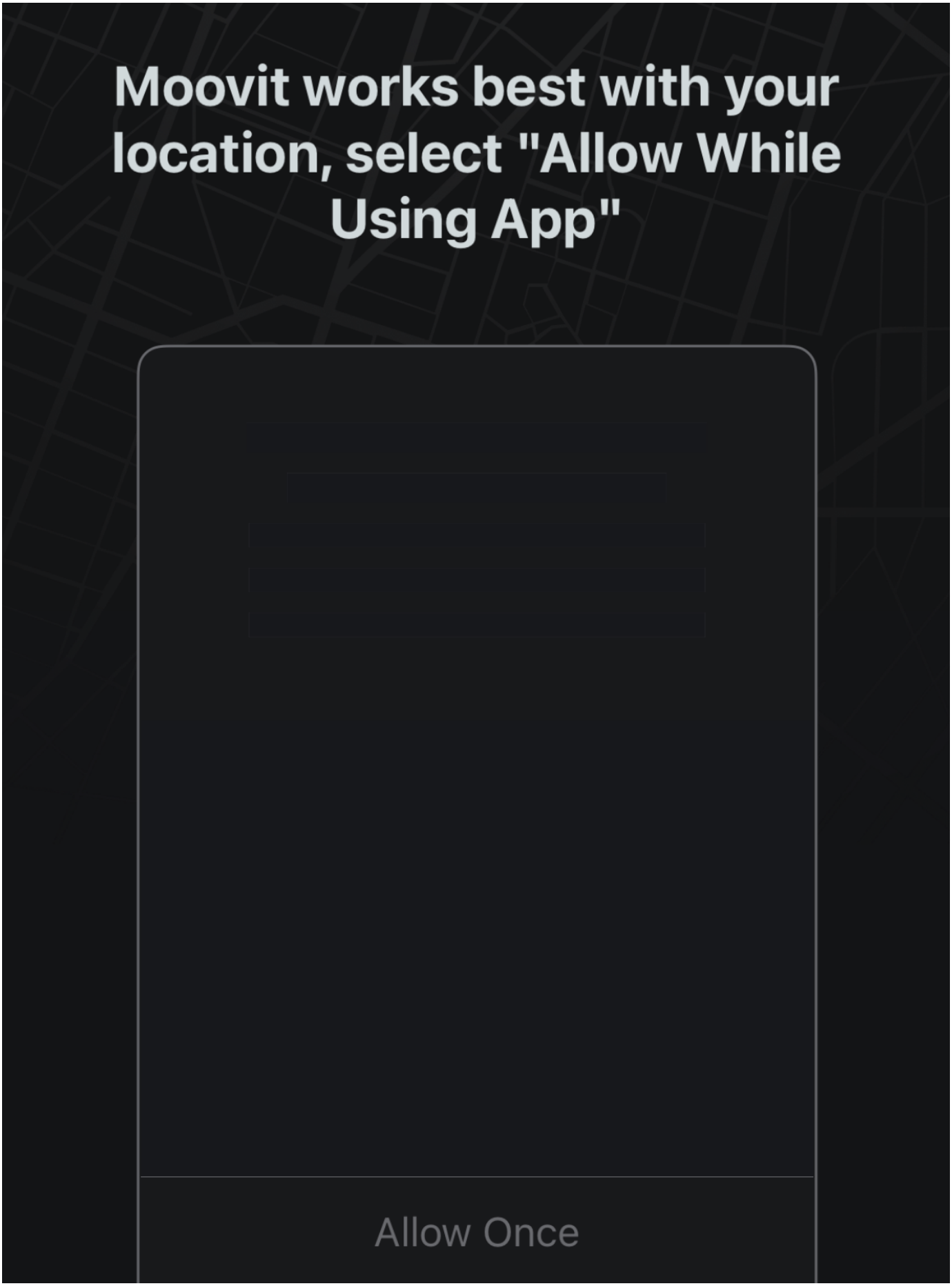
Booterstown, Woodbine Road 10:00

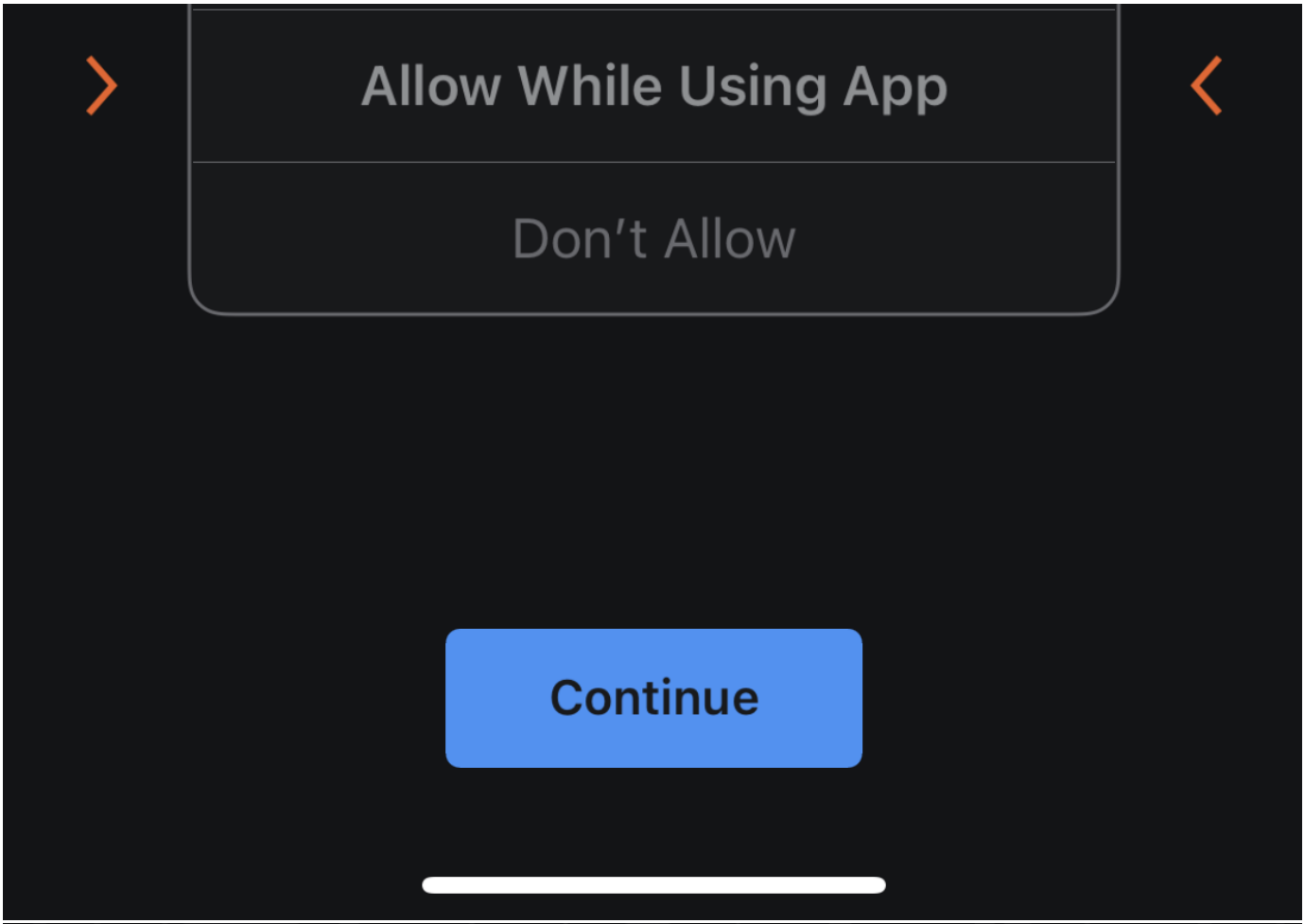
9 a.m.: Usually not busy

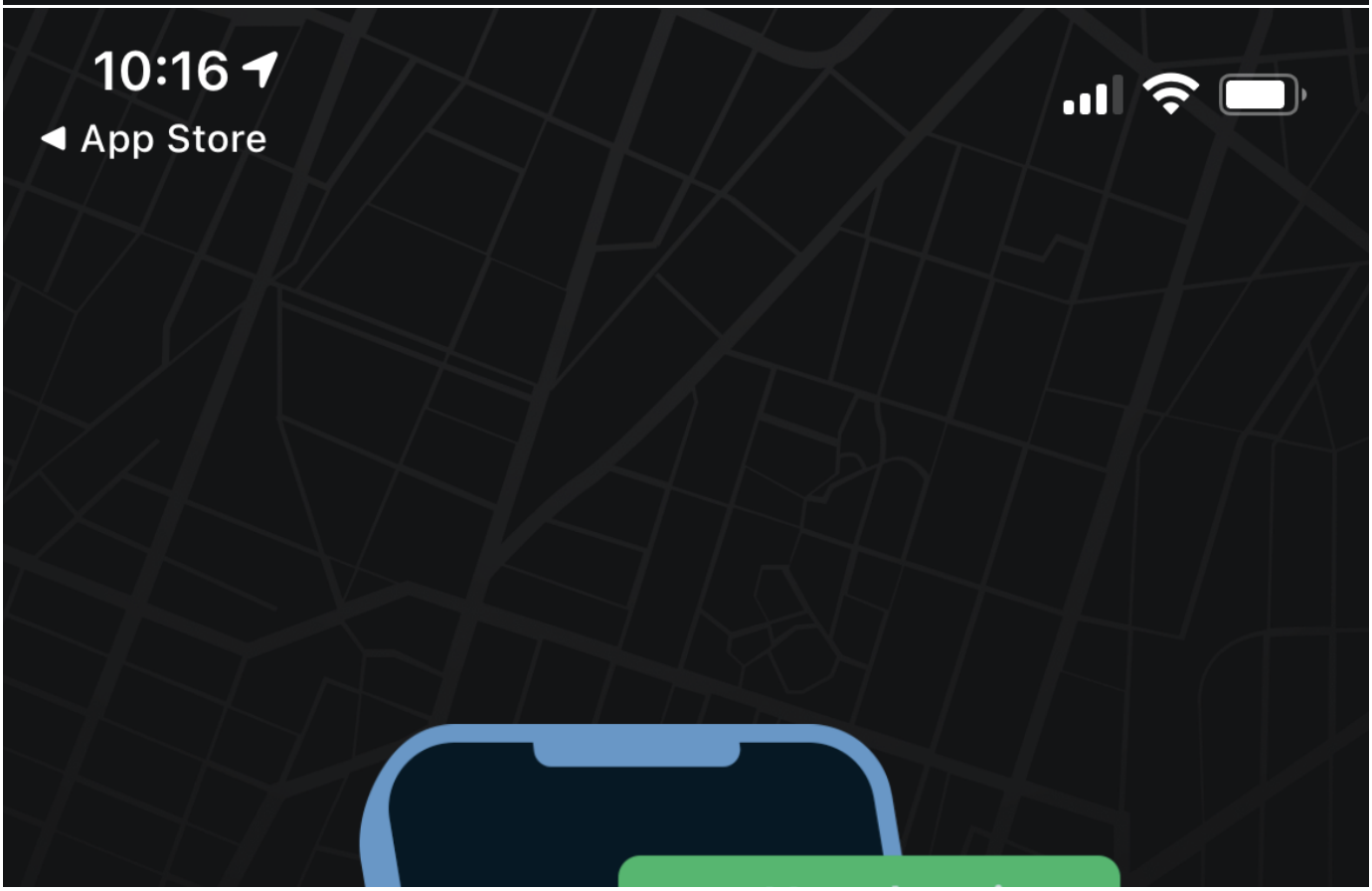


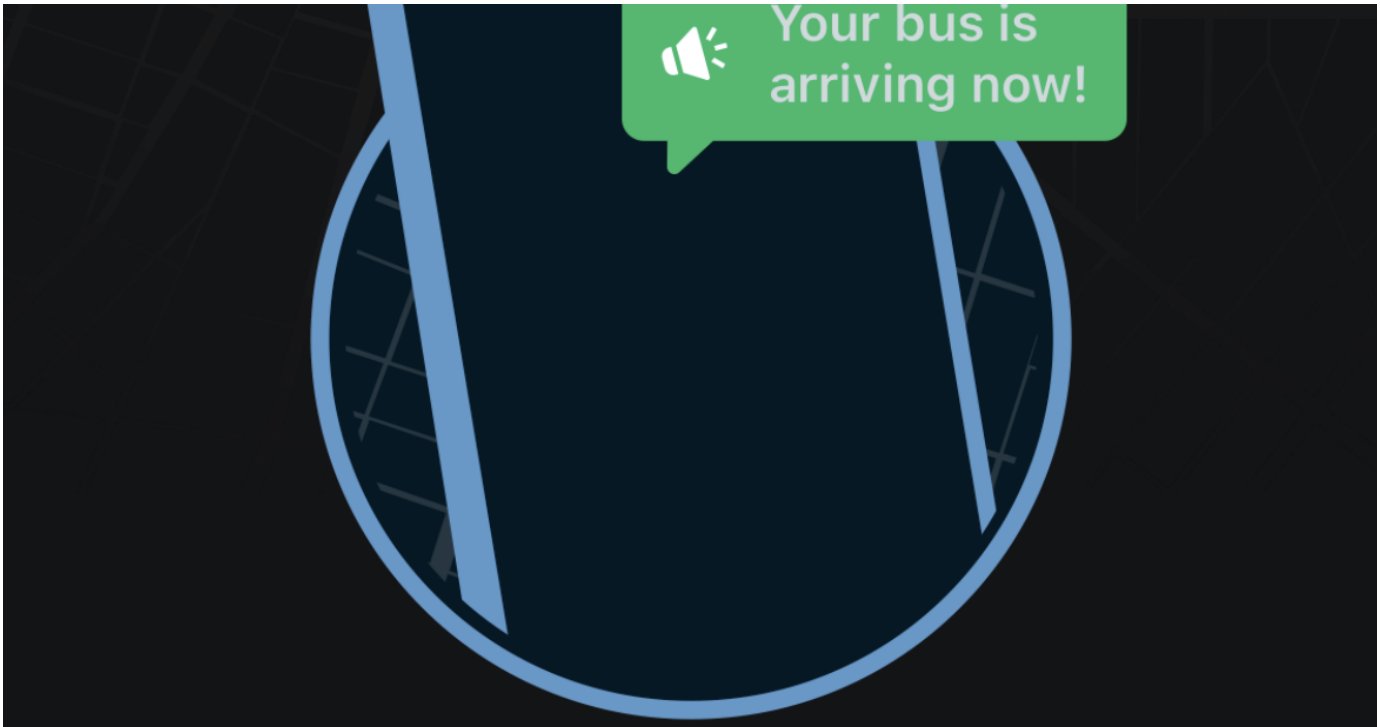
Moovit

The user journey for Moovit:


Moovit works best with your location, select "Allow While Using App"
Allow Once


Allow While Using App
Don’t Allow
Continue


10:16
App Store

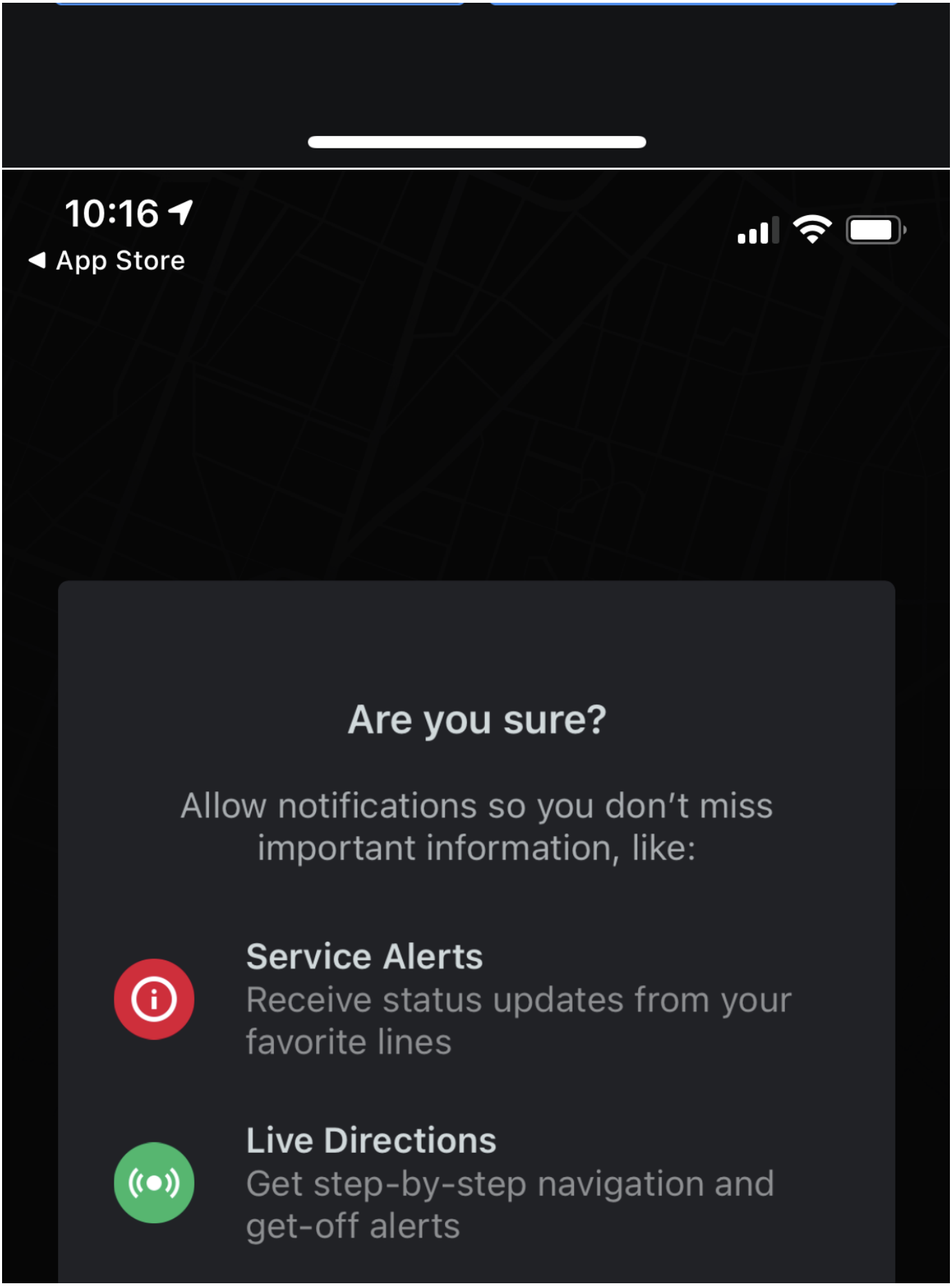


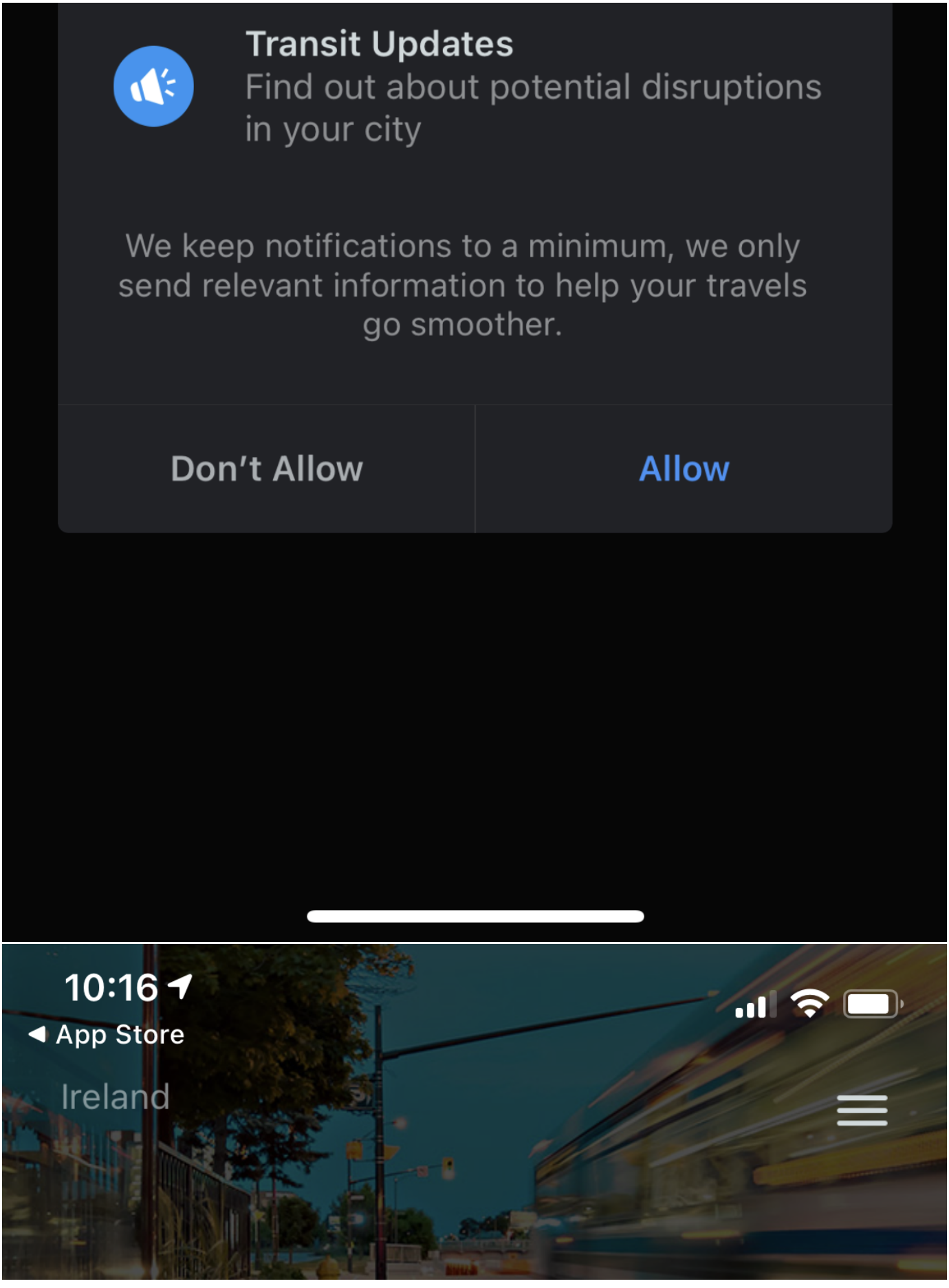
Get Live Updates!

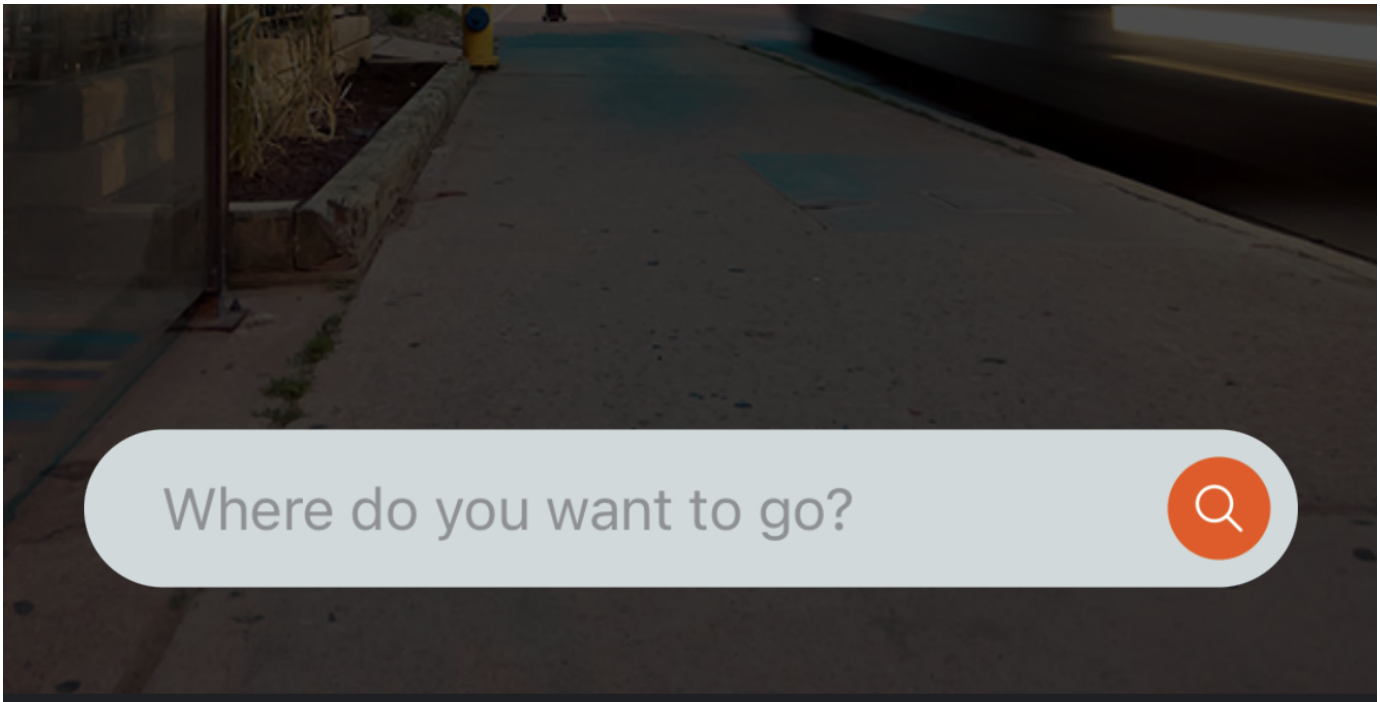
Allow notifications to receive real-time transit alerts, service updates and more!

Not now

Allow


10:16
App Store
Are you sure?
Allow notifications so you don't miss important information, like:
Service Alerts
Receive status updates from your favorite lines
Live Directions
Get step-by-step navigation and get-off alerts


Transit Updates
Find out about potential disruptions in your city
We keep notifications to a minimum, we only send relevant information to help your travels go smoother.
Don’t Allow
Allow
10:16
App Store
Ireland



Favourites Add

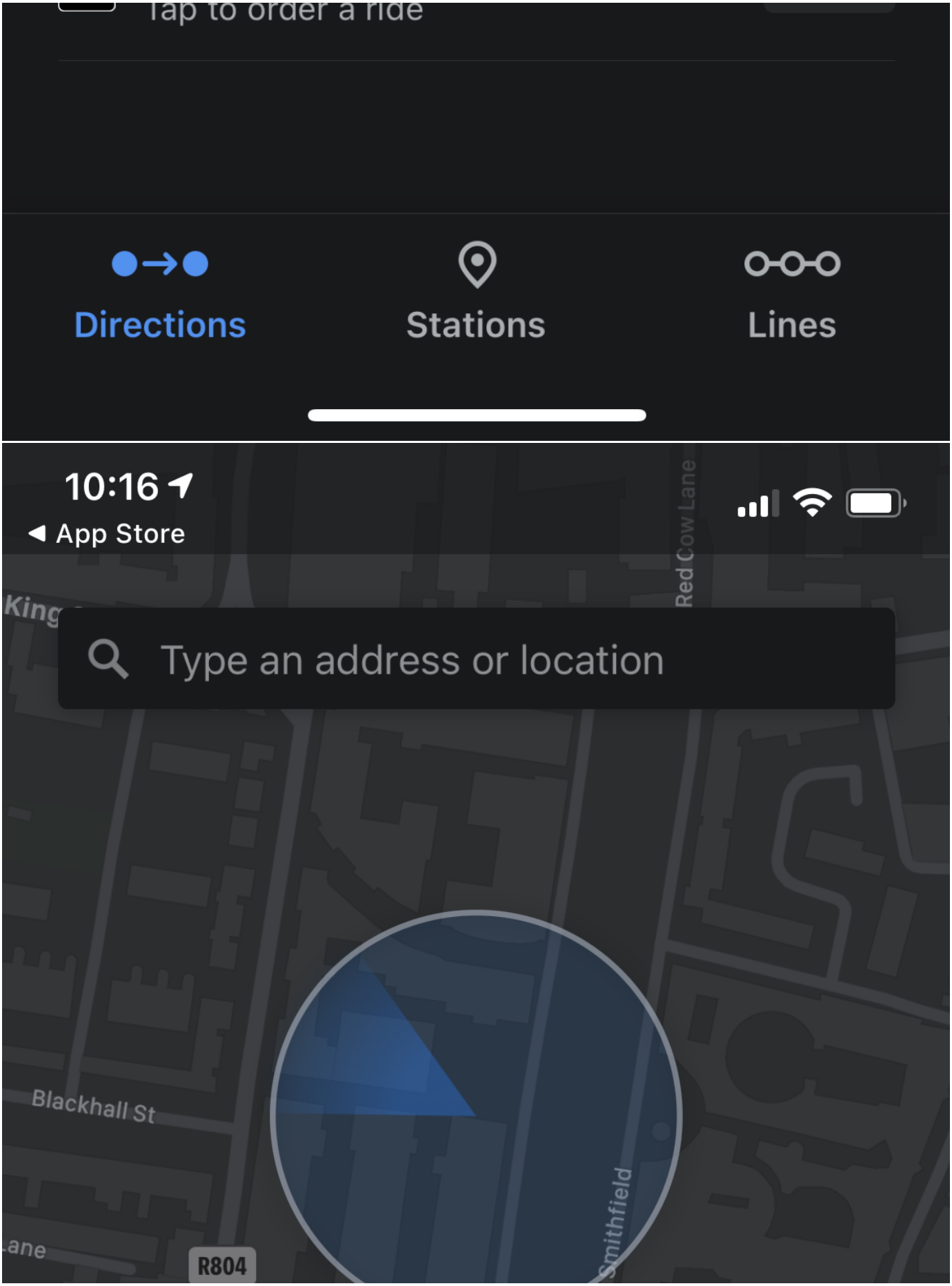
Home
Tap to set

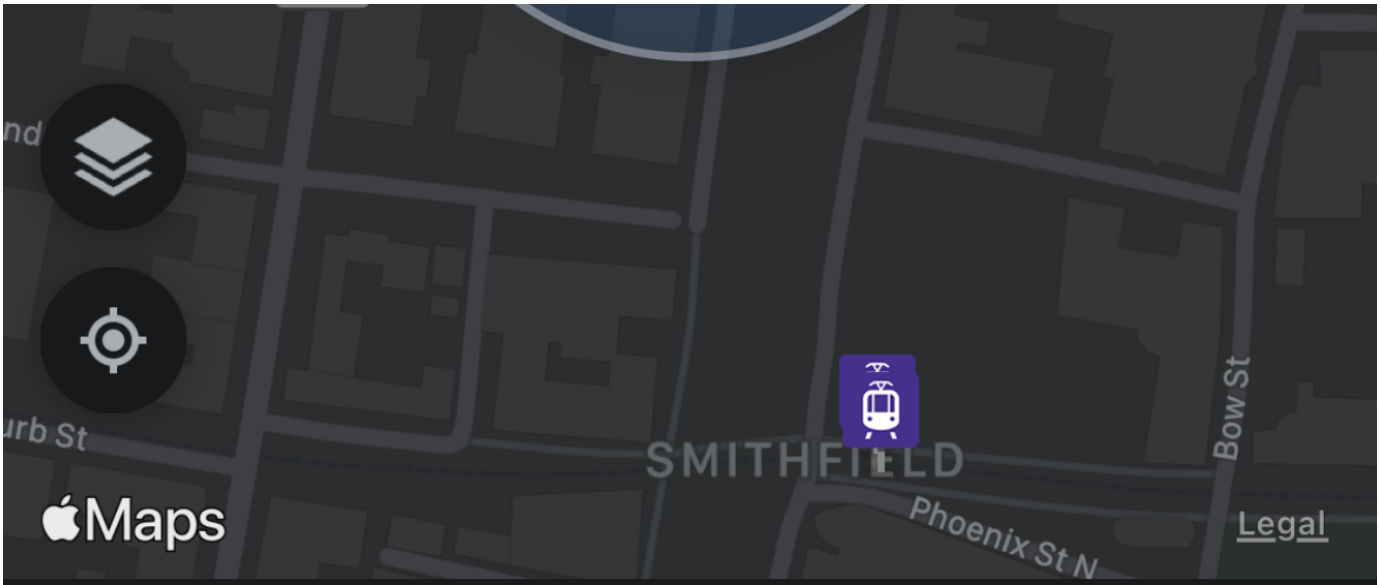
Work
Tap to set

Sync Calendar
Easily navigate to all your events

Taxi & Ride Hailing

Request an Uber
Order


Tap to order a ride
Directions
Stations
Lines
10:16
App Store
Type an address or location
Red Cow Lane
Blackhall St
Smithfield
R804



Nearby Stops

Favourites

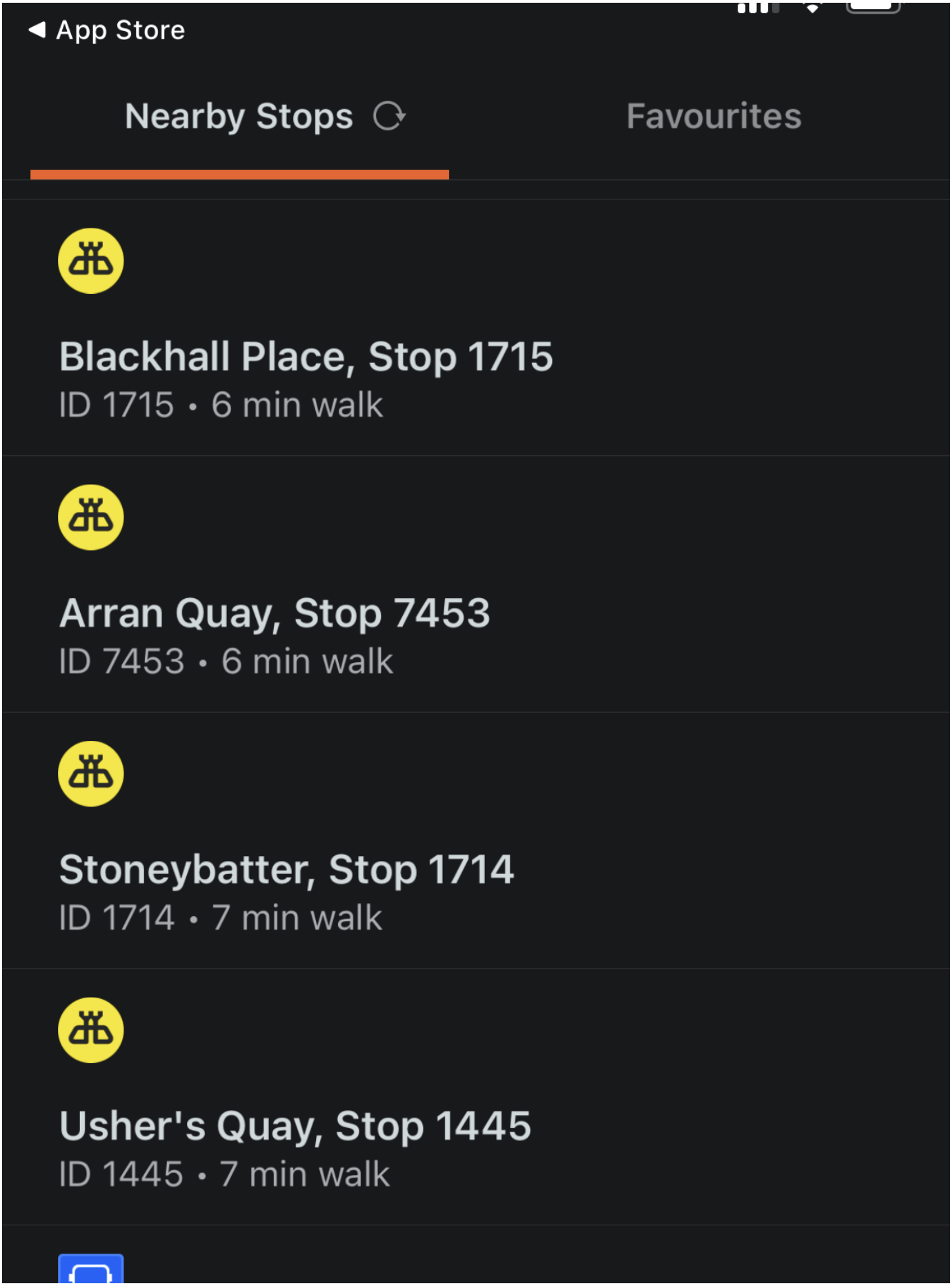
Smithfield

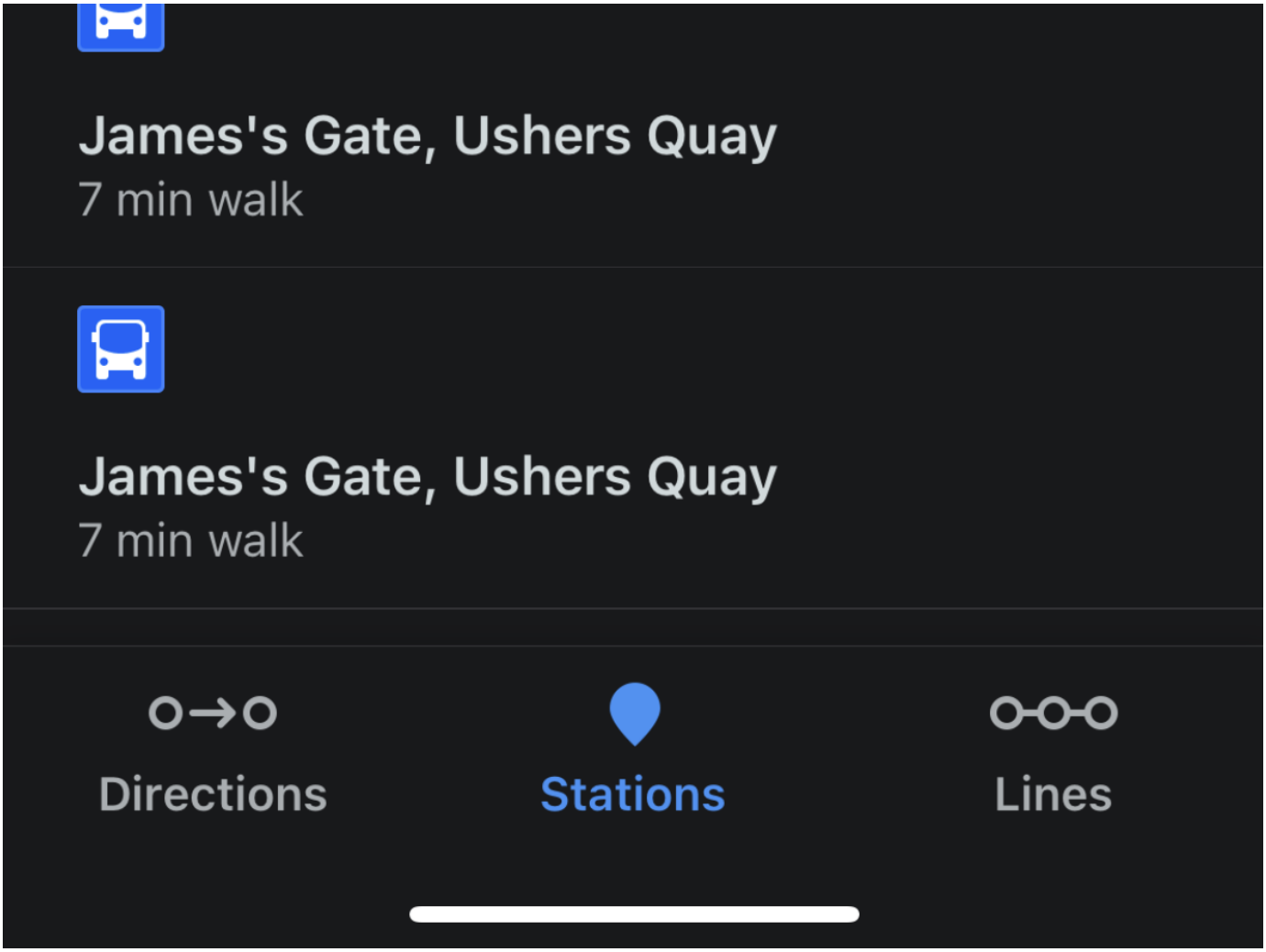
4 min walk • Red

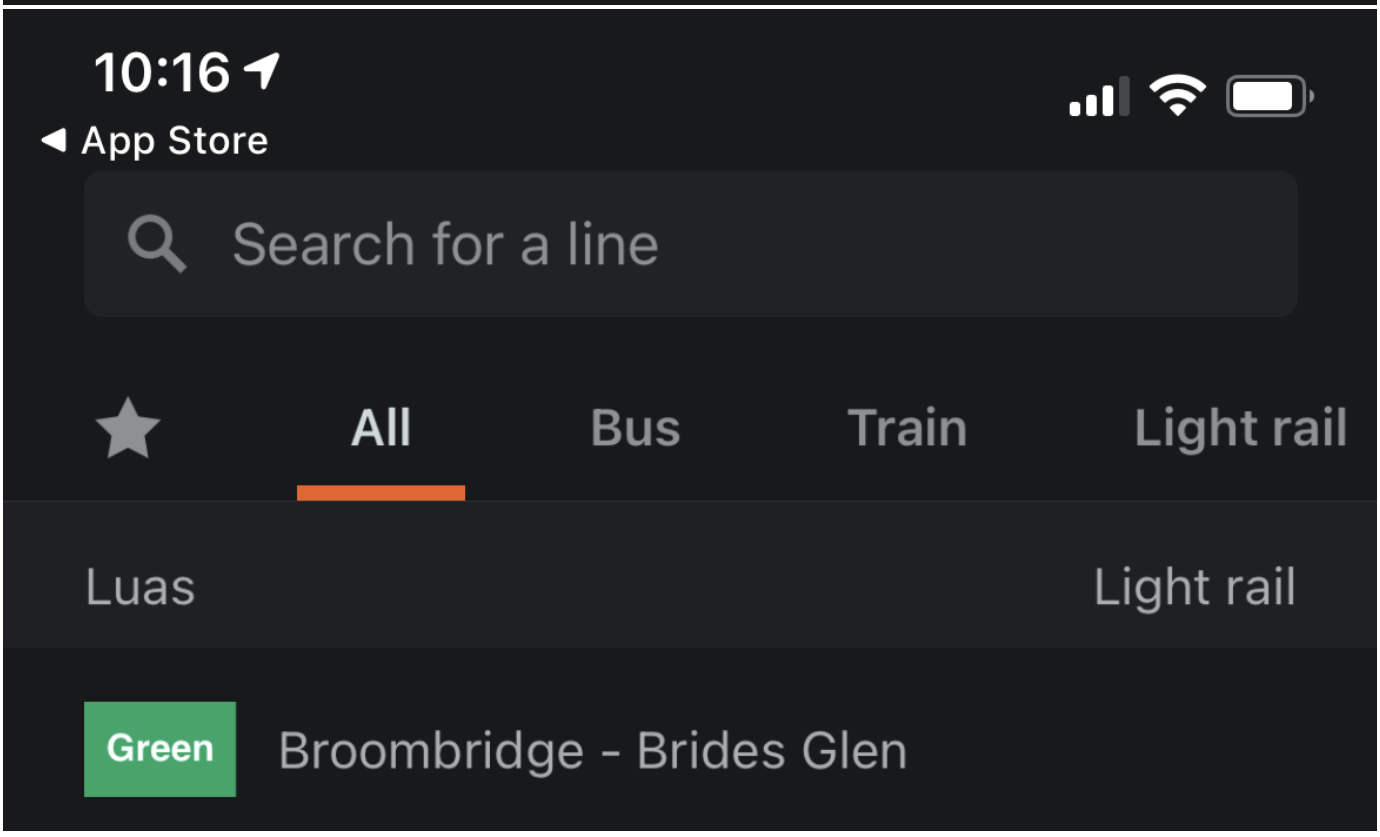
Red **The Point** Now 7 min

Directions Stations Lines

10:16


App Store
Nearby Stops
Favourites
Blackhall Place, Stop 1715
ID 1715 • 6 min walk
Arran Quay, Stop 7453
ID 7453 • 6 min walk
Stoneybatter, Stop 1714
ID 1714 • 7 min walk
Usher's Quay, Stop 1445
ID 1445 • 7 min walk


James's Gate, Ushers Quay
7 min walk
James's Gate, Ushers Quay
7 min walk
Directions
Stations
Lines


10:16
App Store
Search for a line
All
Bus
Train
Light rail
Luas
Light rail
Green
Broombridge - Brides Glen

Red The Point - Saggart

Dublin Bus Bus

1 St John's Church - Shanard Avenue

4 Bluepool Pitches - Harristown

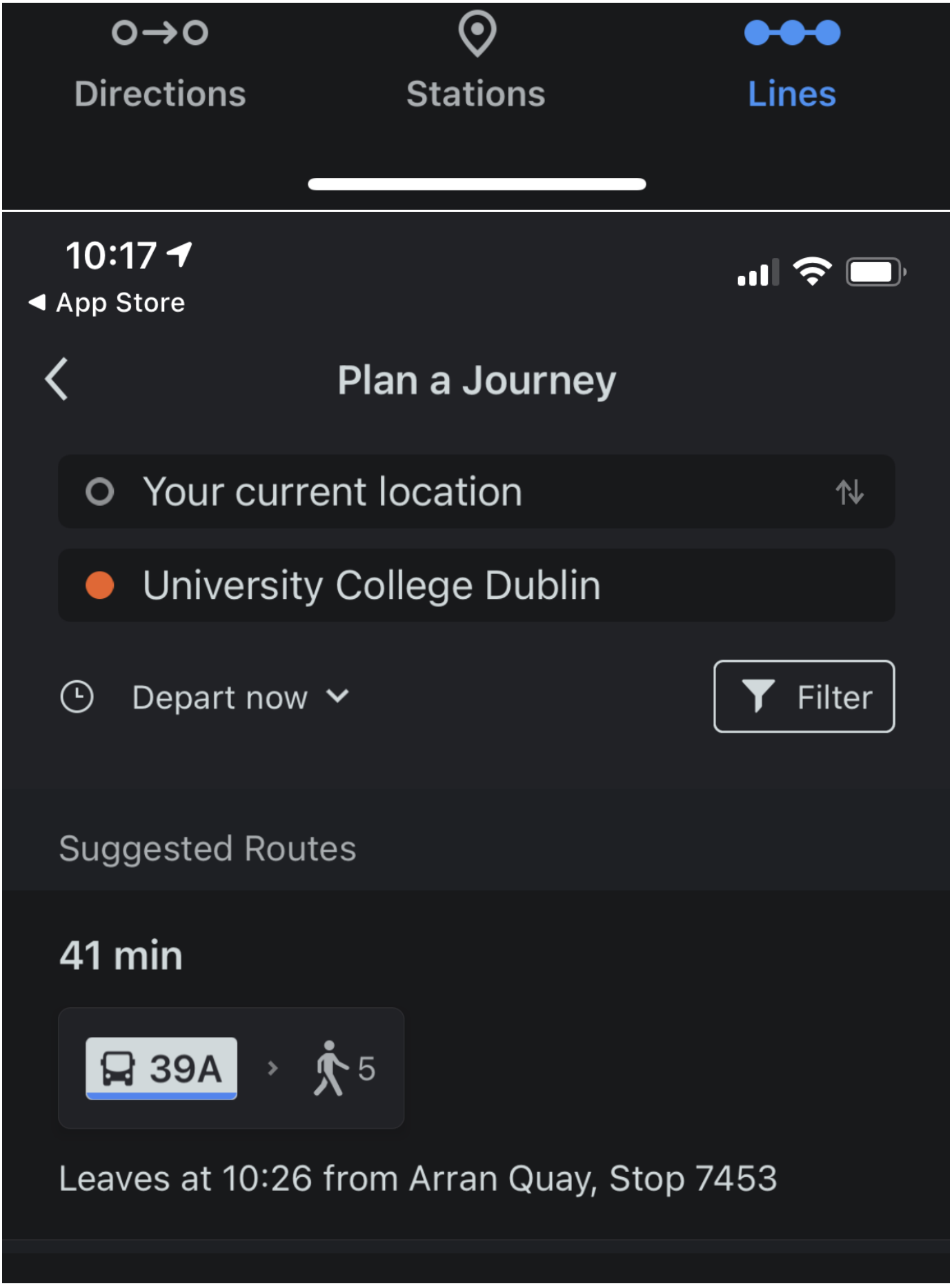
7 Brides Glen Luas - Mountjoy Square

7A Loughlinstown Park - Mountjoy Square

7B Beechfield Manor - Mountjoy Square

7D Dalkey Village - Mountjoy Square

9 Greenhills College - Charlestown SC


Directions
Stations
Lines
10:17
App Store
Plan a Journey
Your current location
University College Dublin
Depart now
Filter
Suggested Routes
41 min
39A
5
Leaves at 10:26 from Arran Quay, Stop 7453

38 min

145 › 6

Leaves at 10:23 from Arran Quay, Stop 7453

Walking & Biking Routes

28 min

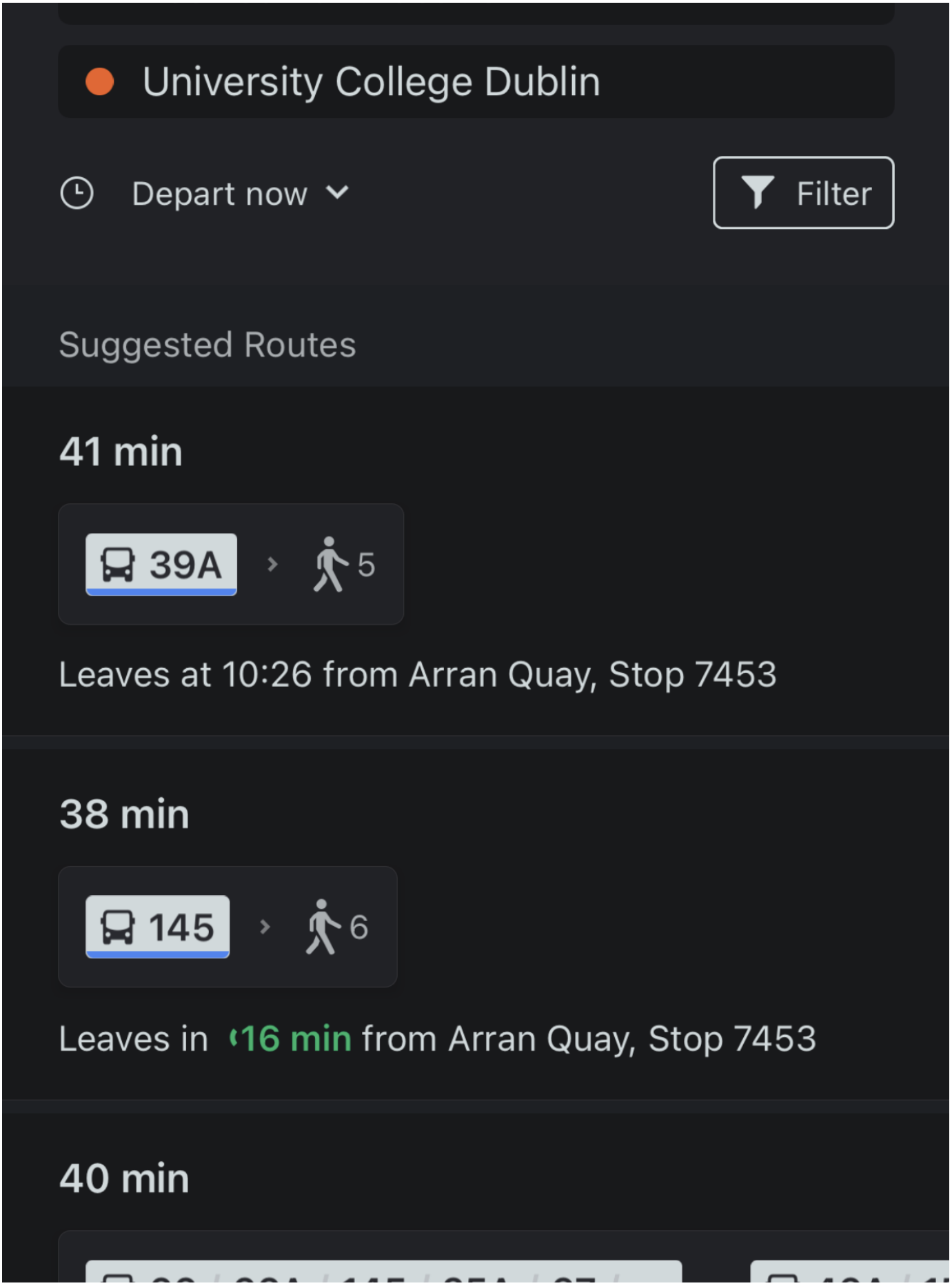
7.8 km

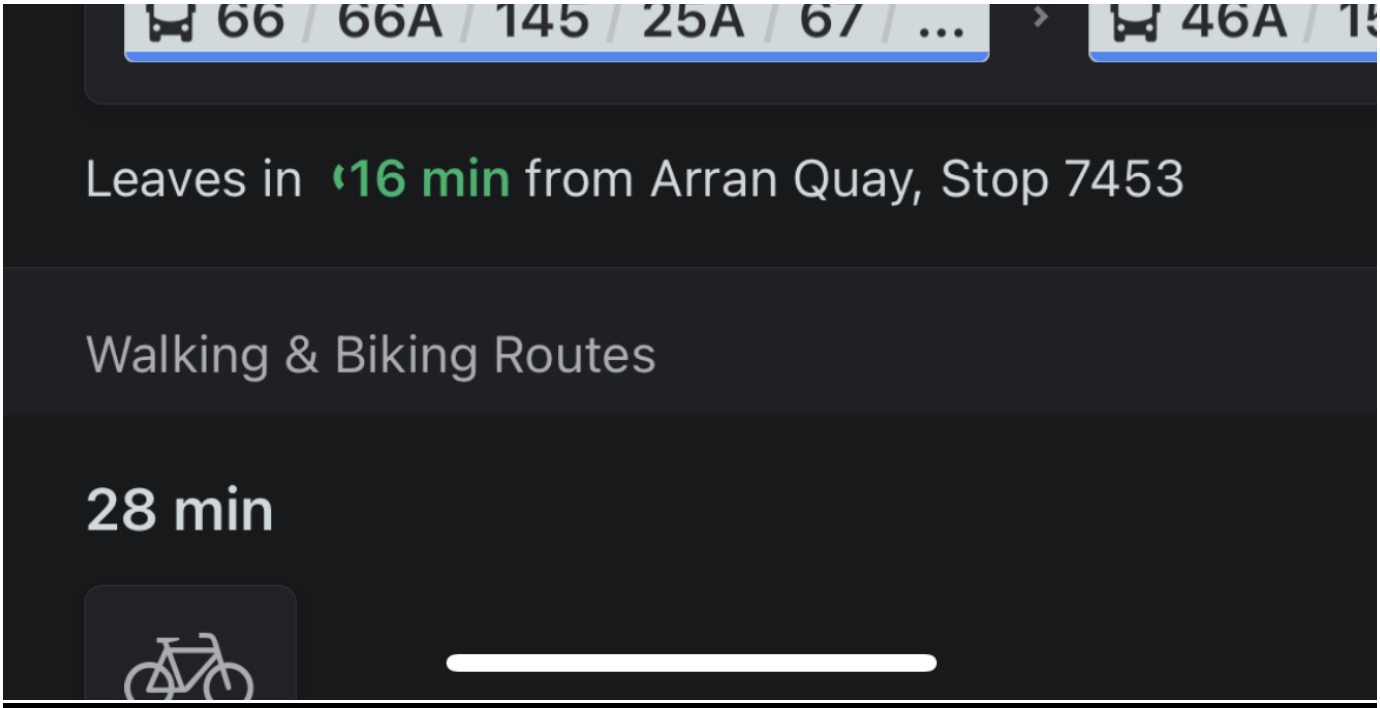
10:17

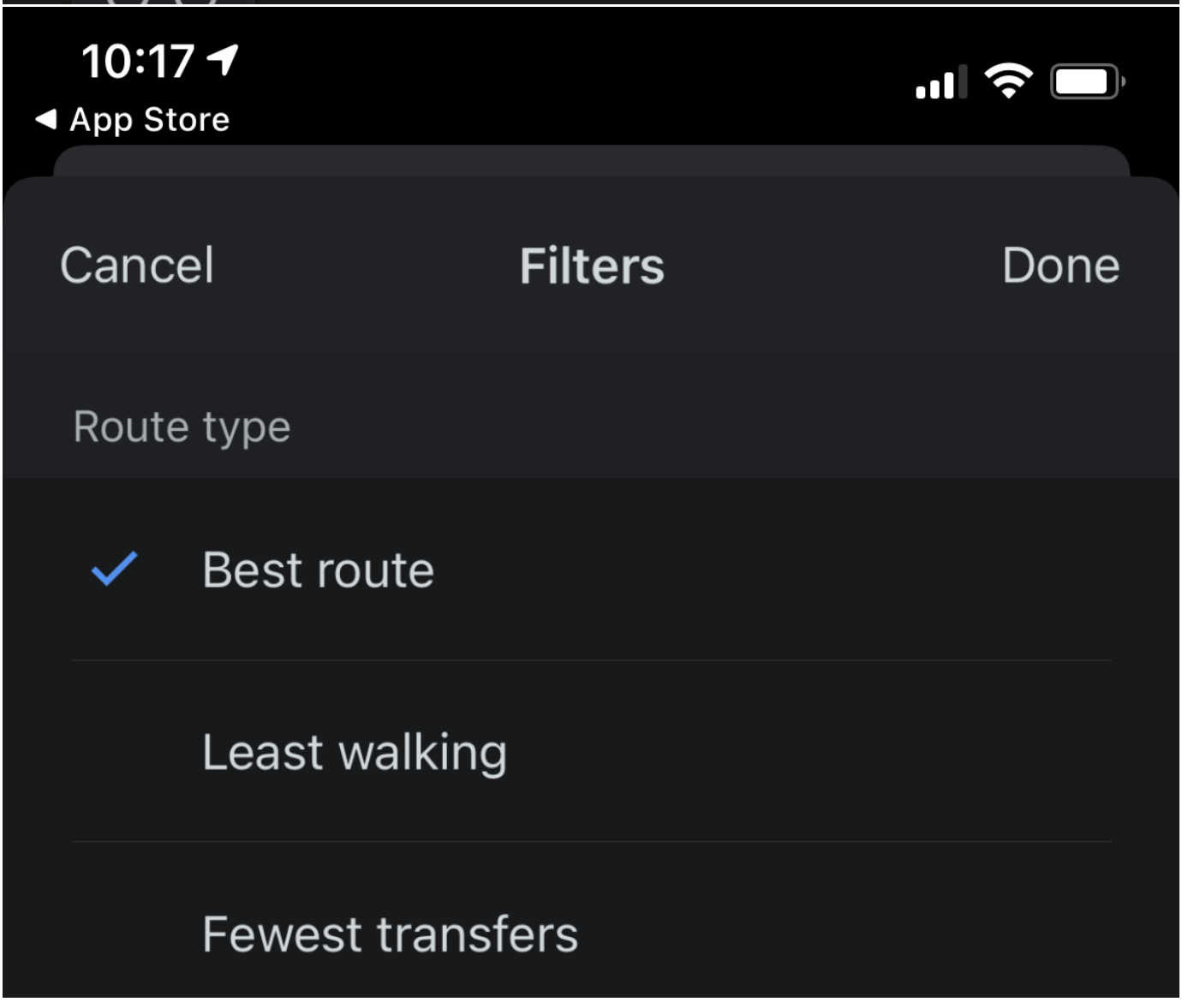
◀ App Store

Plan a Journey

Your current location


University College Dublin
Depart now
Filter
Suggested Routes
41 min
39A
5
Leaves at 10:26 from Arran Quay, Stop 7453
38 min
145
6
Leaves in 16 min from Arran Quay, Stop 7453
40 min


66 / 66A / 145 / 25A / 67 / ...
46A / 15
Leaves in 16 min from Arran Quay, Stop 7453
Walking & Biking Routes
28 min


10:17
App Store
Cancel
Filters
Done
Route type
Best route
Least walking
Fewest transfers

Transport types

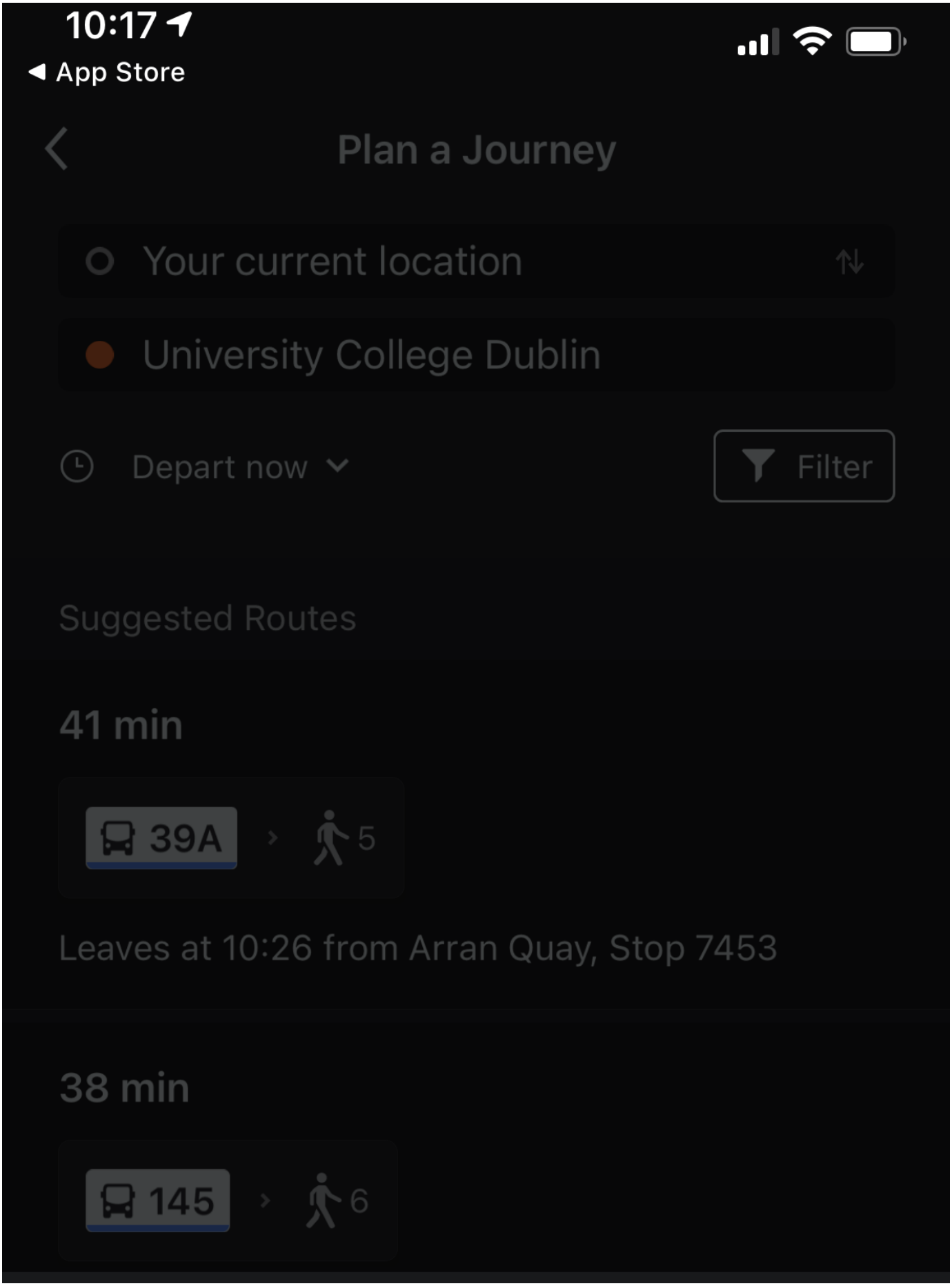
Bus

Train

Light rail

Ferry

Bike


10:17
App Store
Plan a Journey
Your current location
University College Dublin
Depart now
Filter
Suggested Routes
41 min
39A
5
Leaves at 10:26 from Arran Quay, Stop 7453
38 min
145
6

Depart Now

Set departure time

Set desired arrival time

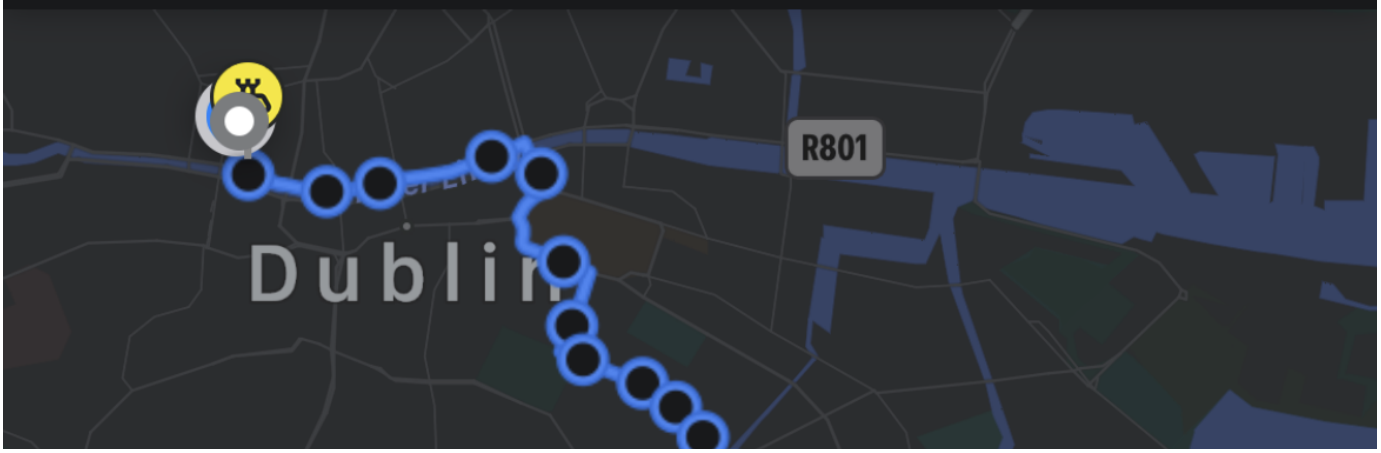
Last lines for today

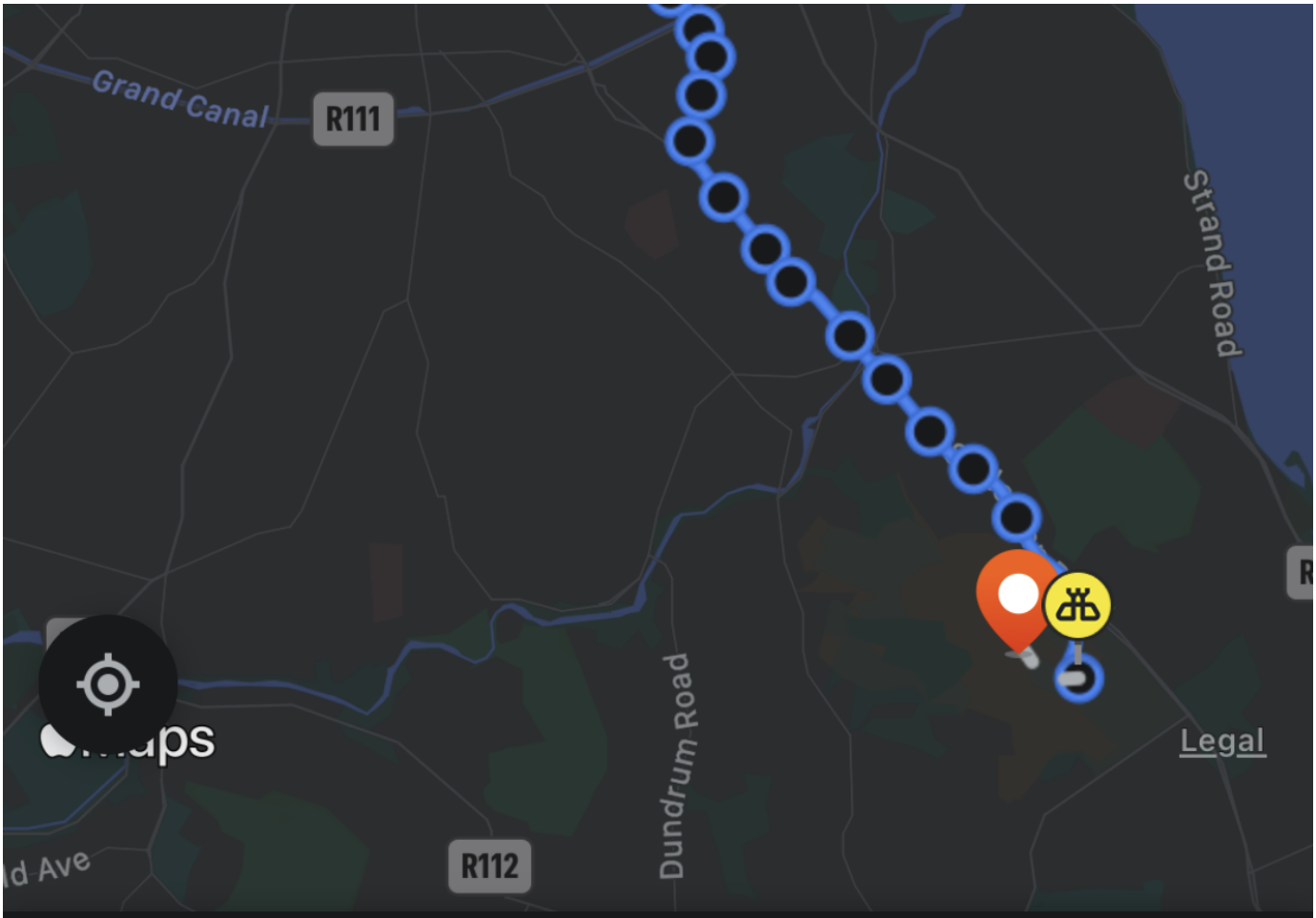
Cancel

10:17

App Store

Directions





University College Dublin

41 min

Earlier 10:20 - 11:01 Later

Leave at 10:20 from

Your current location

Walk to

Report Share Go

10:17

App Store

Directions

Arran Quay, Stop 7453

ID 7453

350 m • 4 min

Wait for

39A Delhurst Estate - University College Dublin 10:26 10:36

Show more options

Travel to

Ucd, Stop 767
ID 767

23 stops • 30 min

Walk to

University College Dublin
Roebuck, Dublin 4

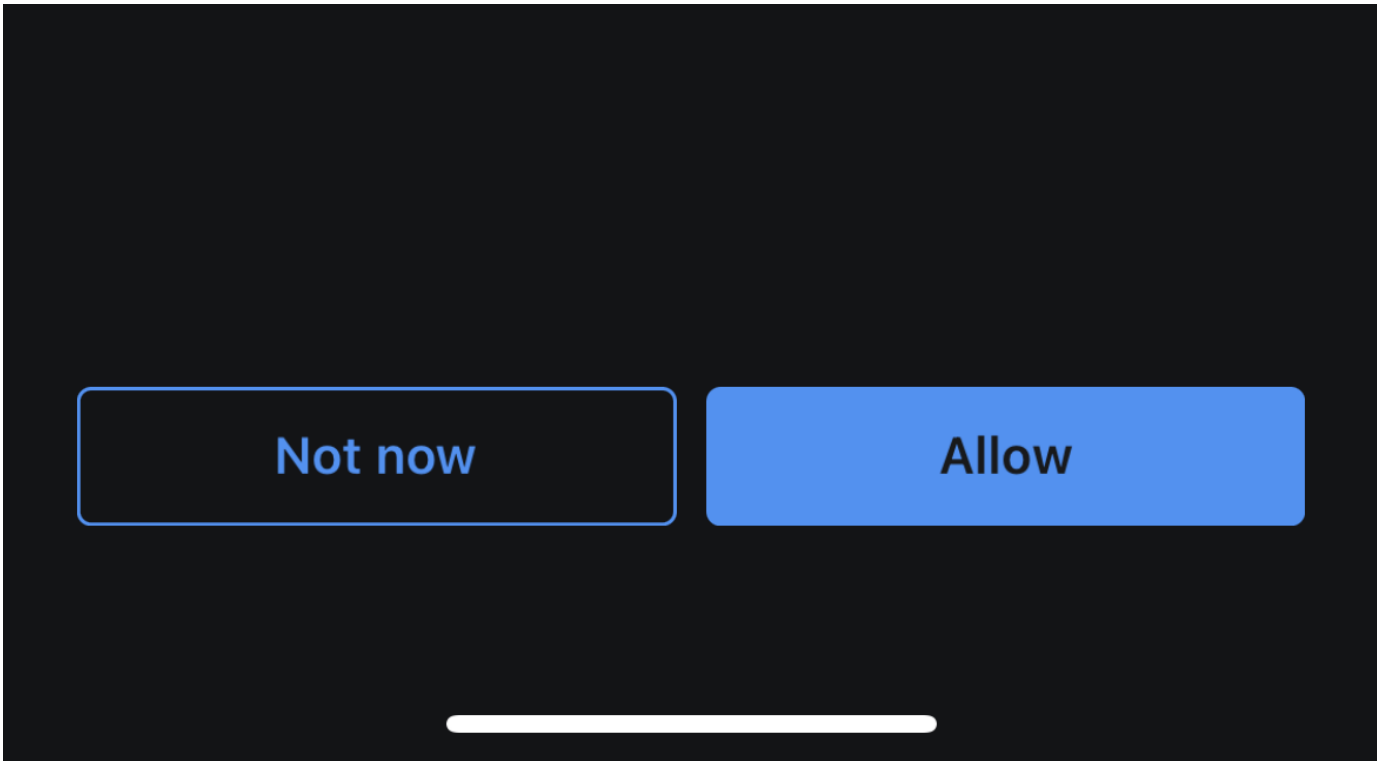
400 m • 5 min

Report Share Go

10:17
App Store

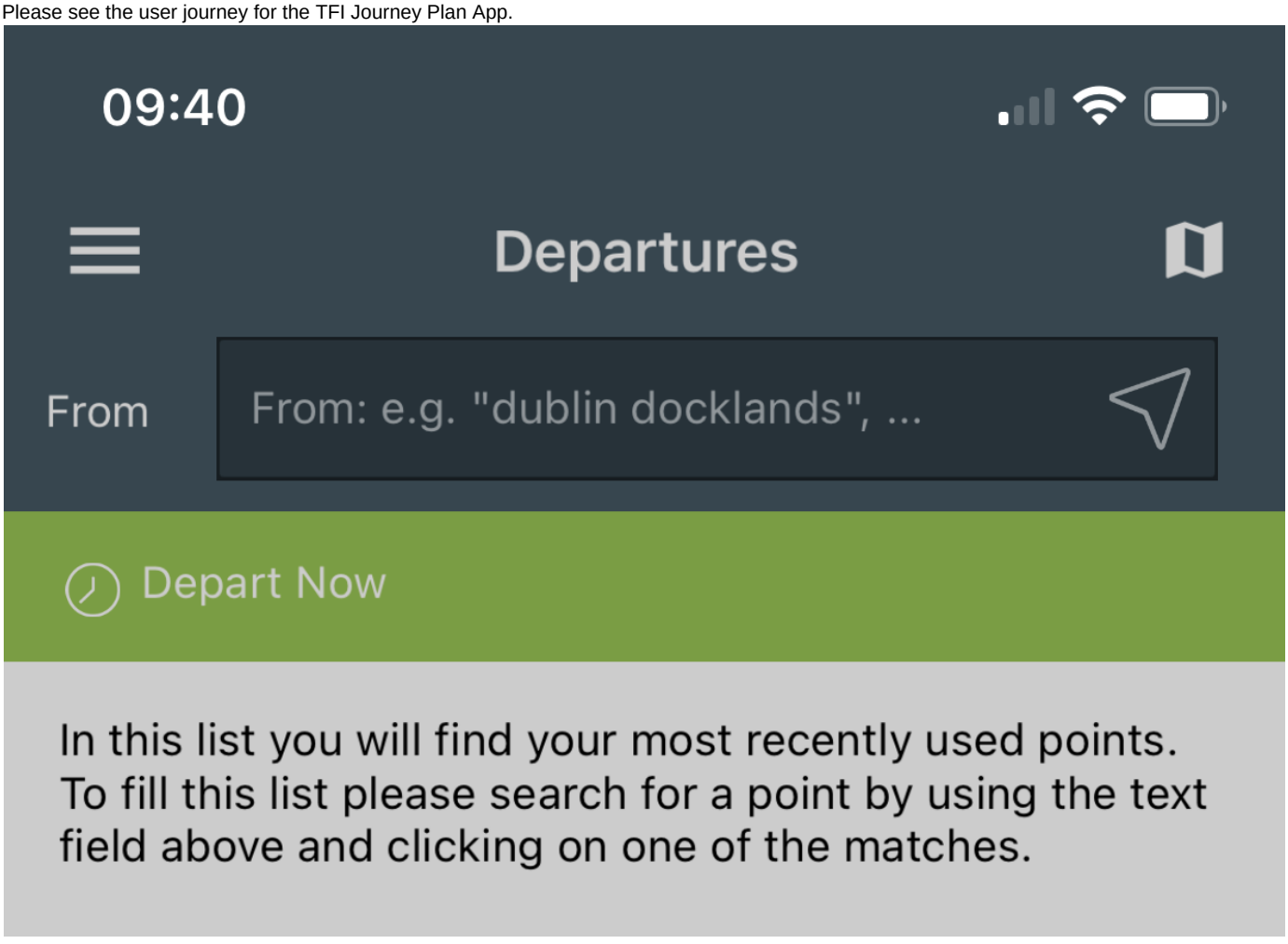
Get Live Updates!

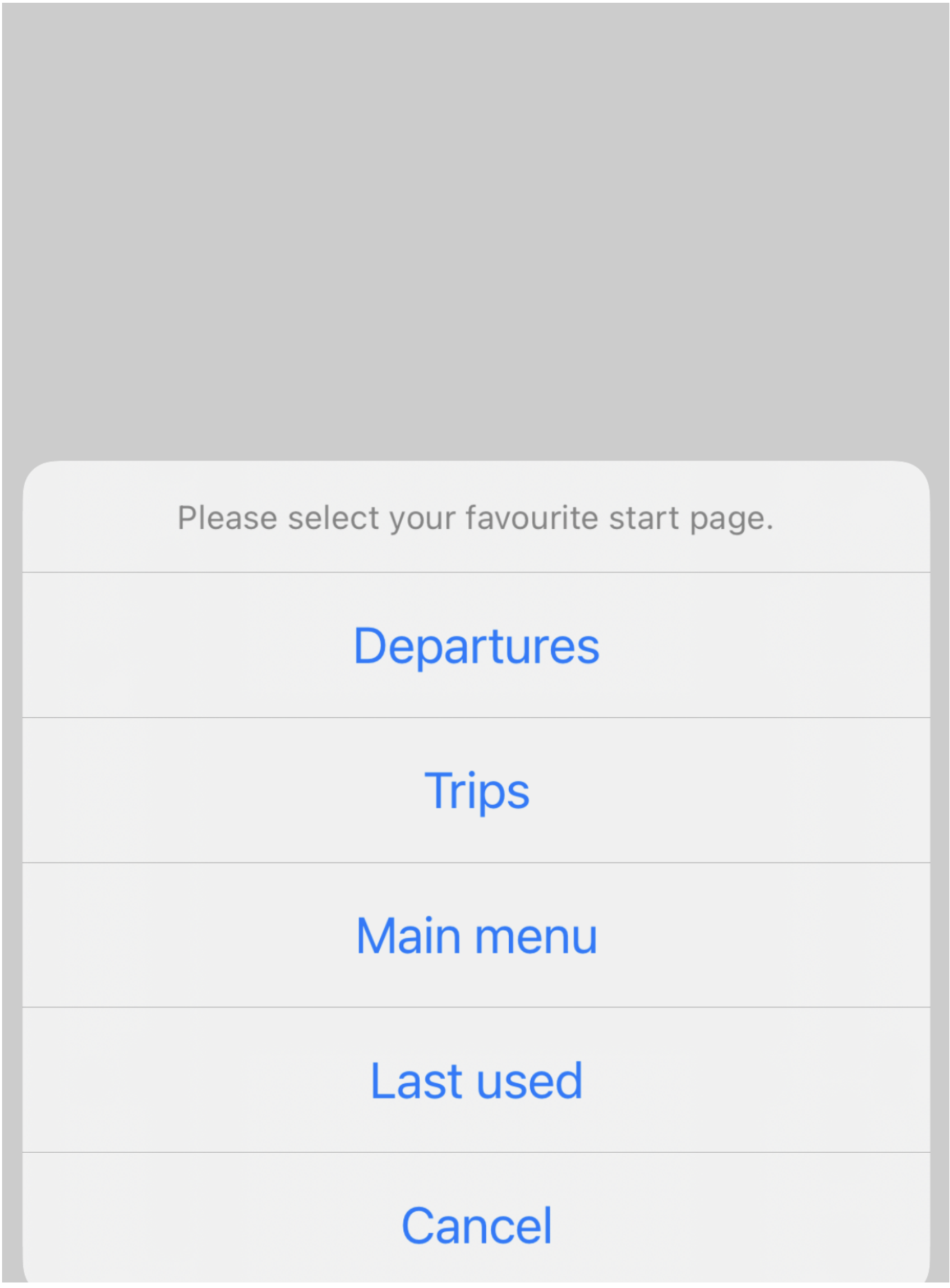
Allow notifications to receive real-time transit alerts, service updates and more!

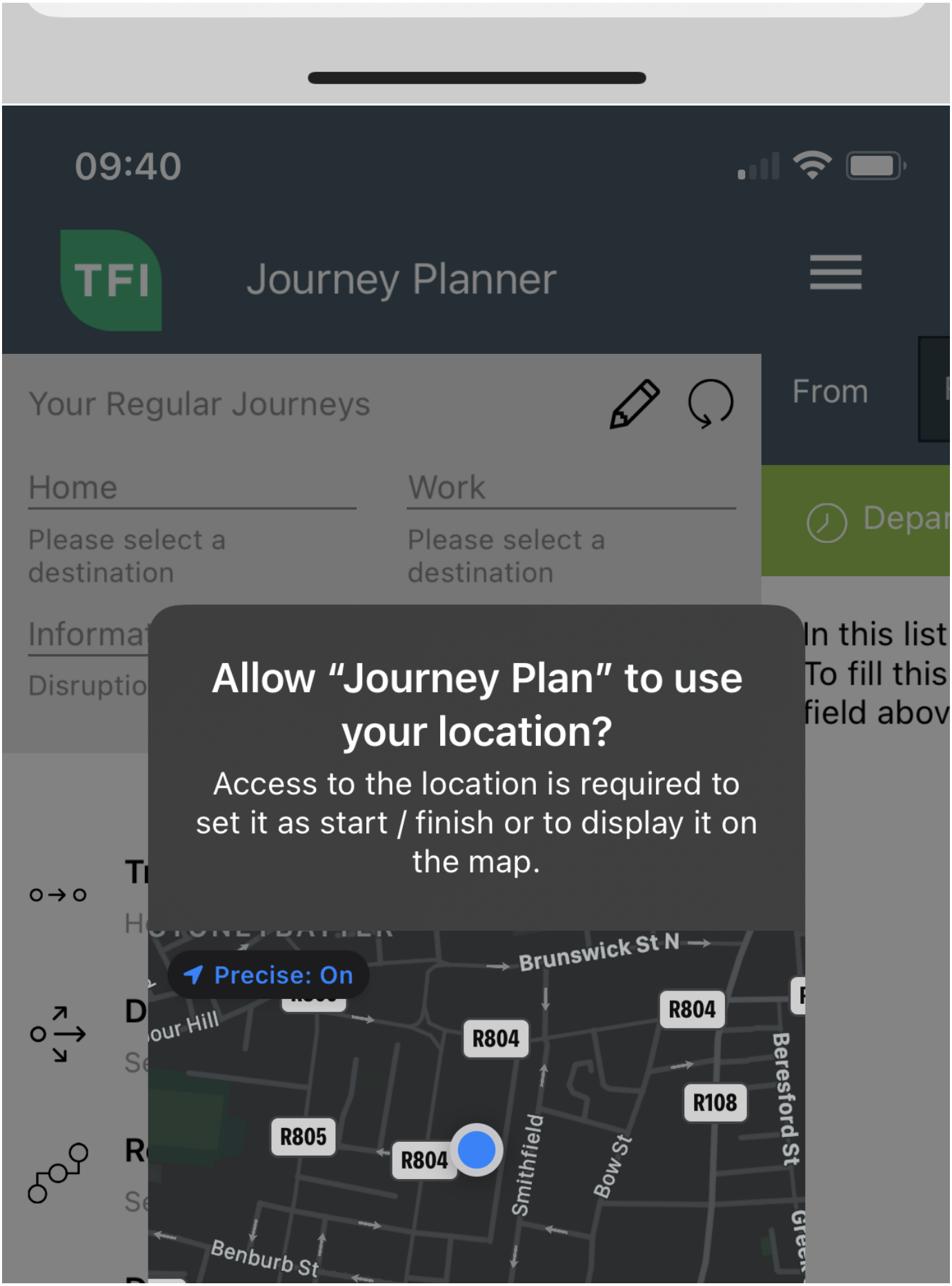


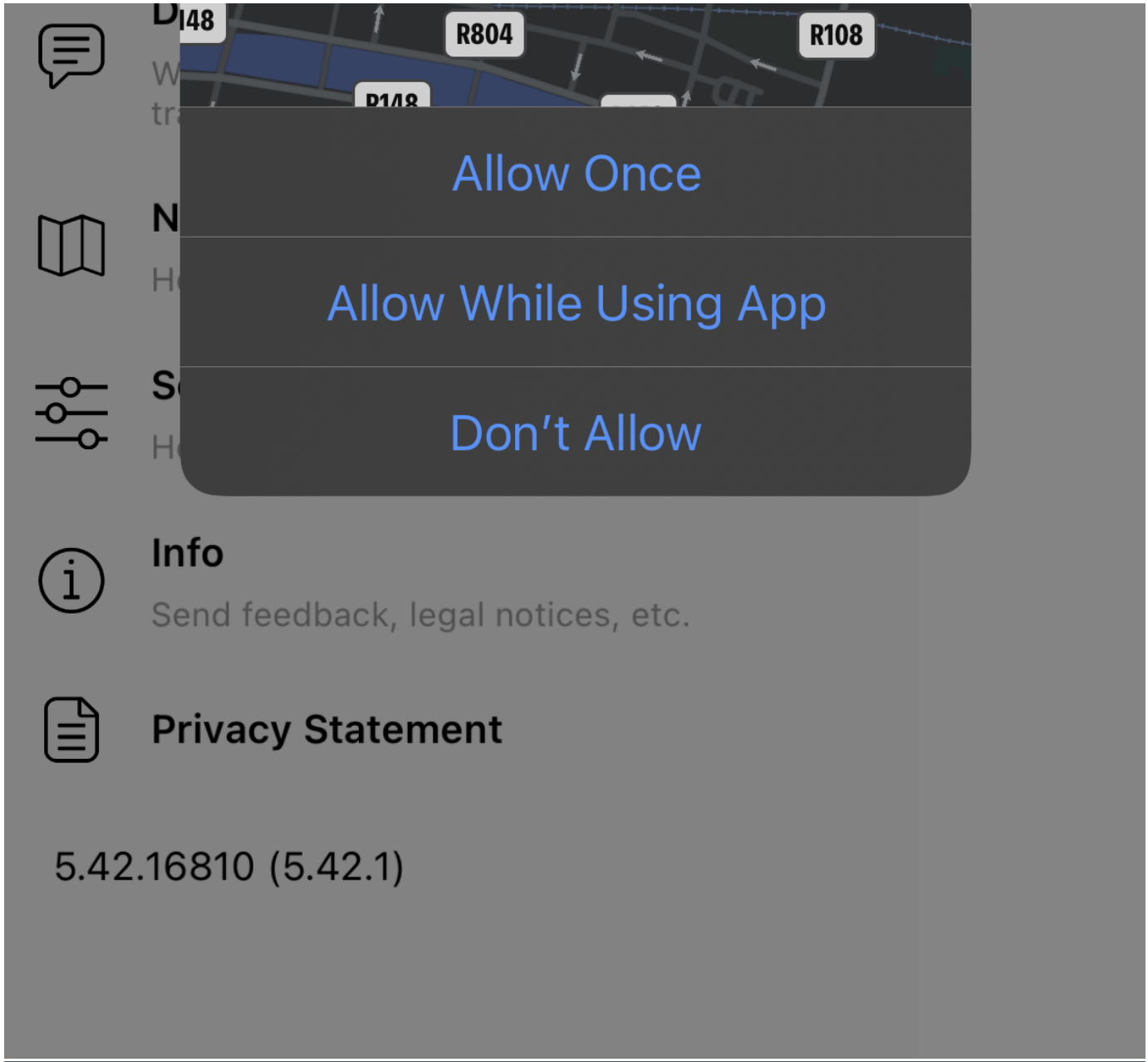
TFI Journey Plan App

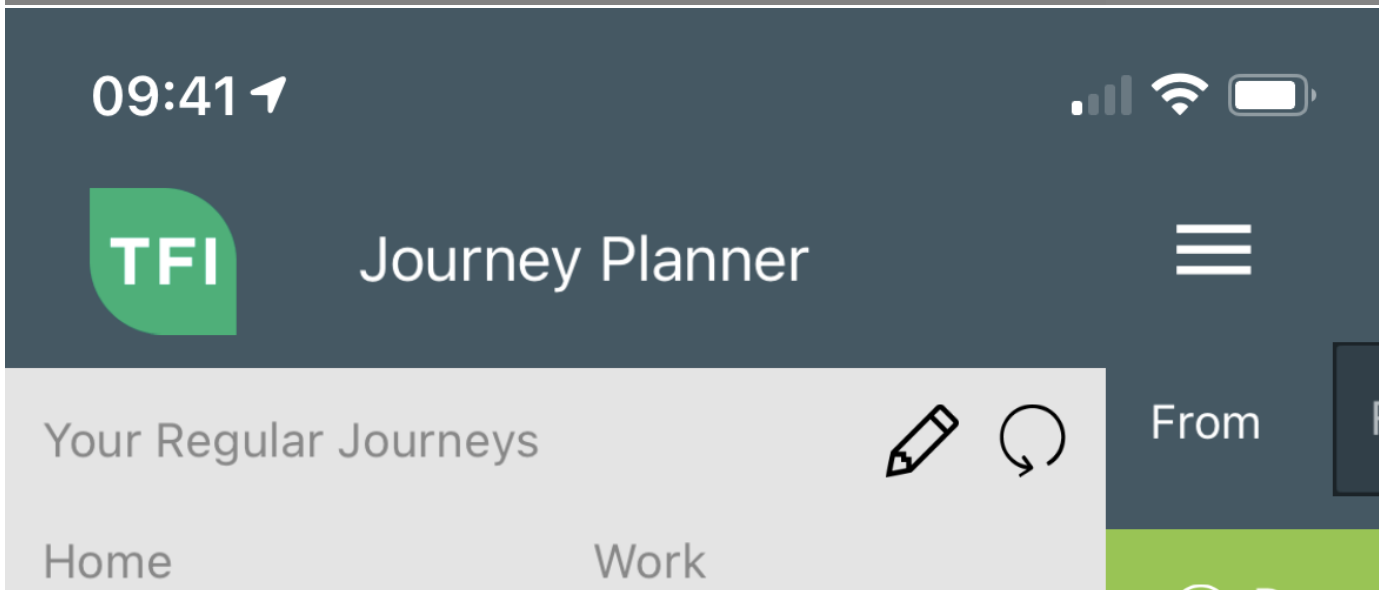
Please see the user journey for the TFI Journey Plan App.




Please select your favourite start page.
Departures
Trips
Main menu
Last used
Cancel


09:40
TFI
Journey Planner
Your Regular Journeys
From
Home
Please select a destination
Work
Please select a destination
Allow “Journey Plan” to use your location?
Access to the location is required to set it as start / finish or to display it on the map.
Precise: On
Brunswick St N
R804
R804
R108
R805
R804
Smithfield
Bow St
Beresford St
Benburb St


R804
R108
Allow Once
Allow While Using App
Don’t Allow
Info
Send feedback, legal notices, etc.
Privacy Statement
5.42.16810 (5.42.1)


09:41
TFI
Journey Planner
Your Regular Journeys
From
Home
Work

Please select a destination

Please select a destination

Depar

Information

Disruption by route

In this list
To fill this
field abov

...

Trips

How do I get from A to B?

Departures

Search for departures by Stop or Station

Route Diagrams

Search for departures for stops along a Route

Disruptions

What disruptions are currently affecting the transport network?

Network Maps

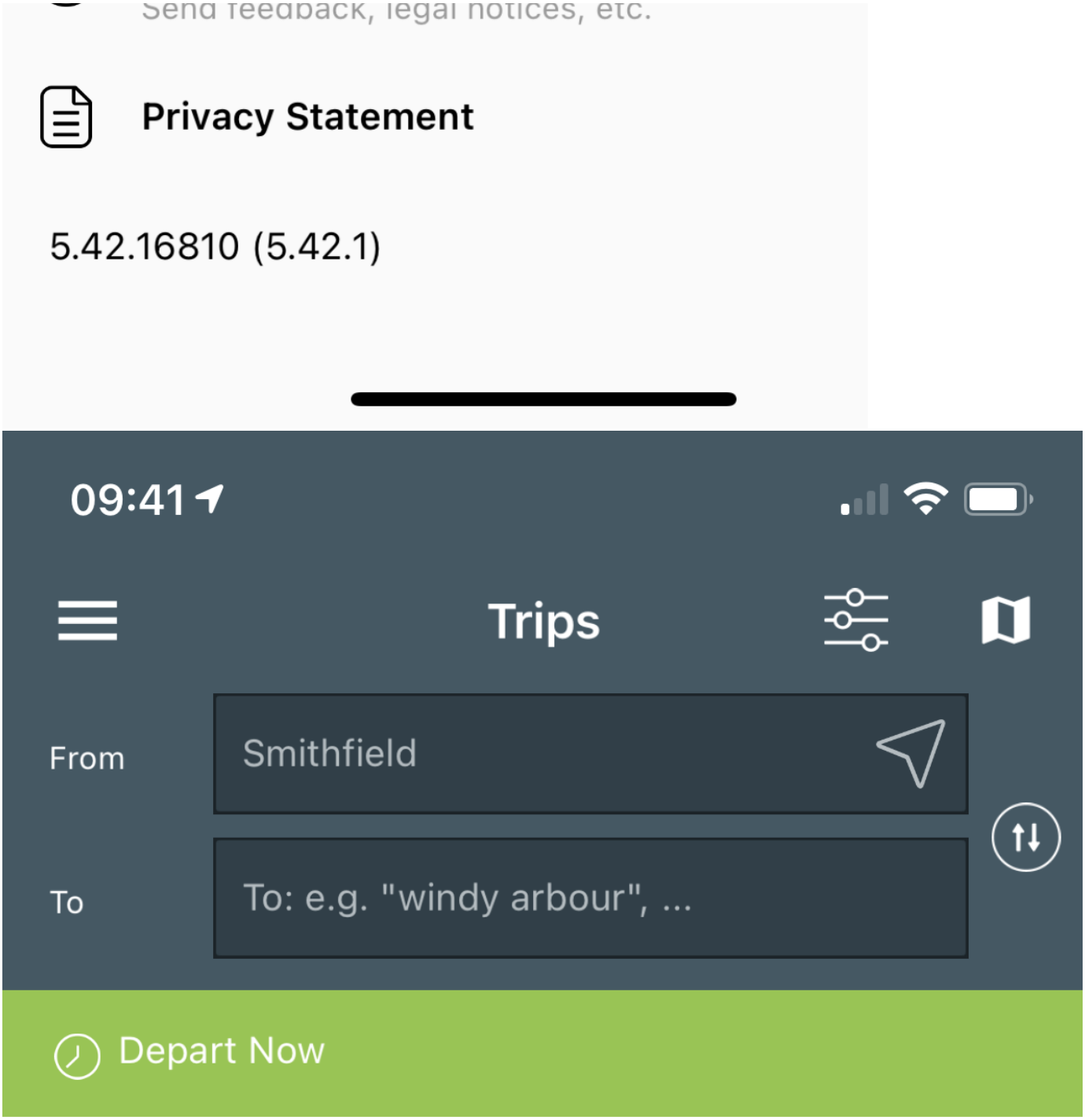
How can I get an overview of the network?

Settings

How can I adjust the app to my needs?

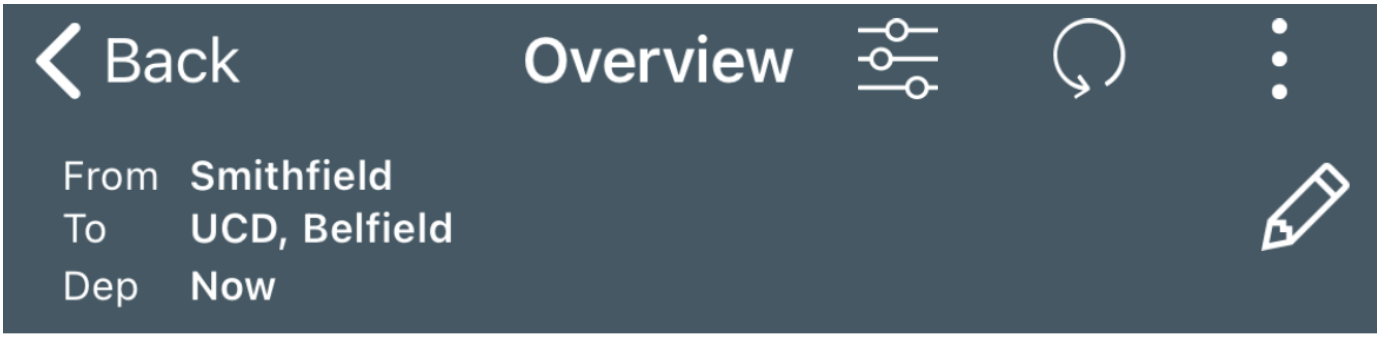
Info

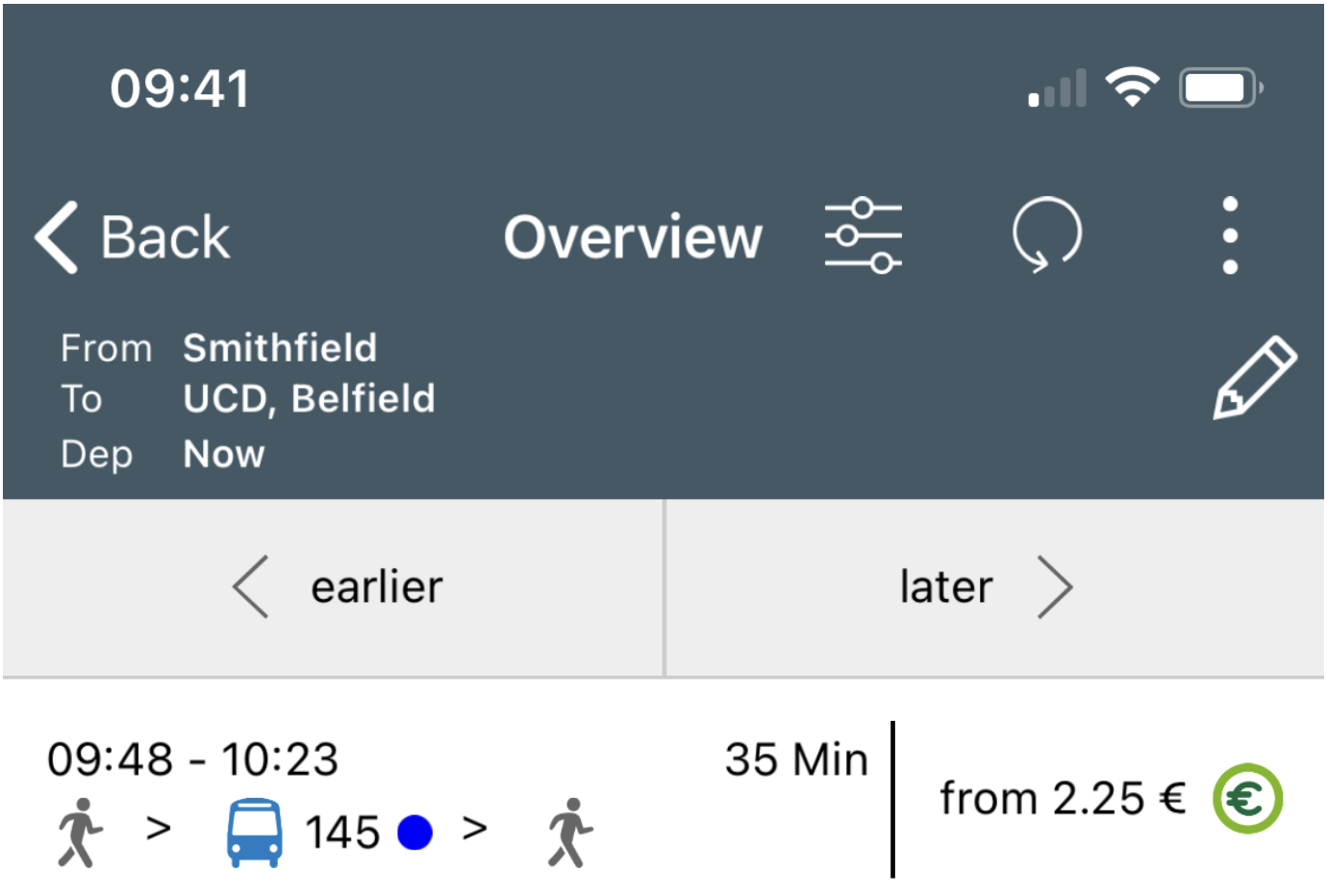
Send feedback, legal notices, etc.



In this list you will find your most recently used trips. To fill this list just calculate a trip.


09:41


Back
Overview
From Smithfield
To UCD, Belfield
Dep Now


09:41
Back
Overview
From Smithfield
To UCD, Belfield
Dep Now
earlier
later
09:48 - 10:23
>
145
>
35 Min
from 2.25 €

09:58 - 10:33 35 Min from 2.25 €

> 145 >

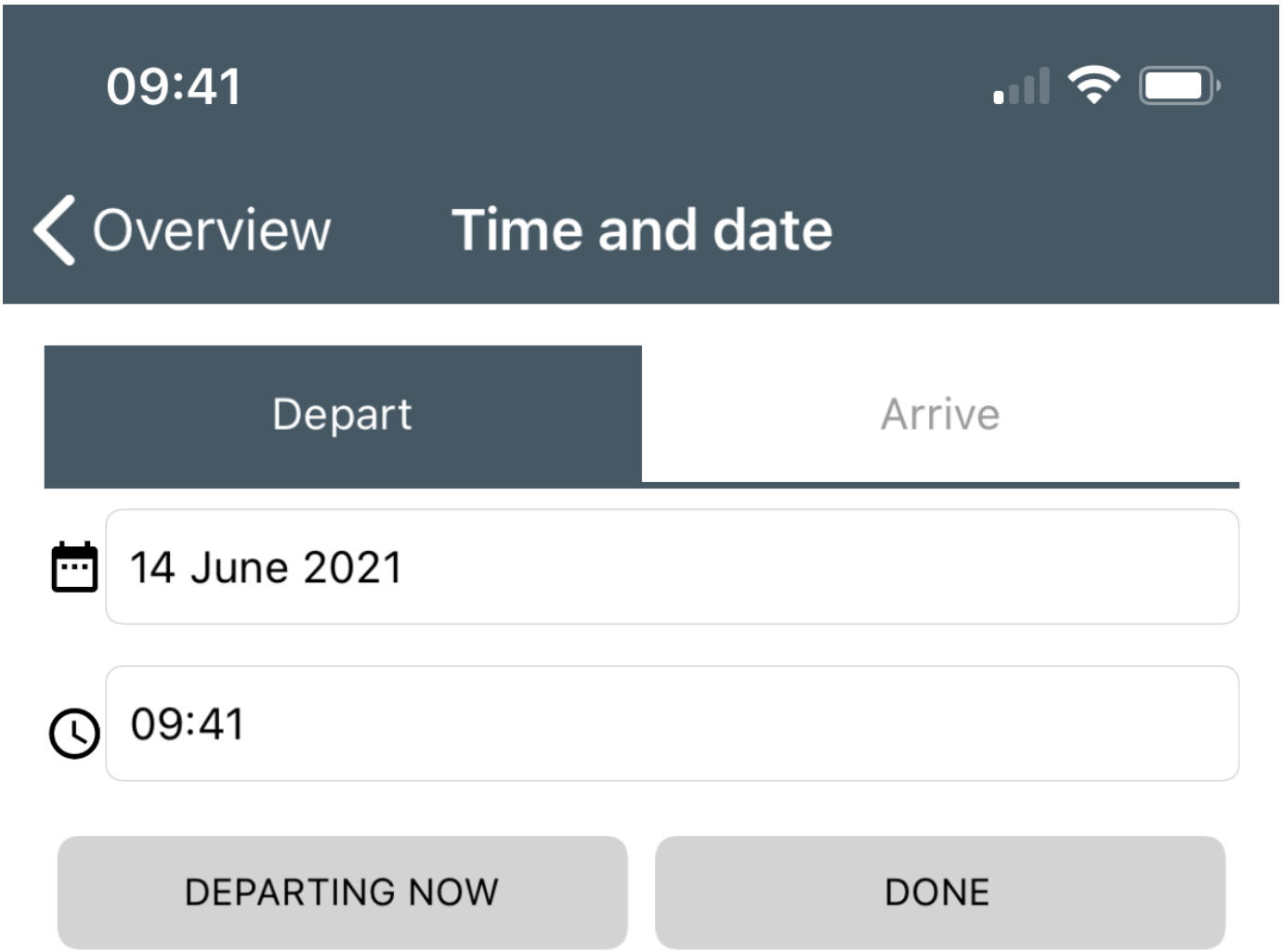
10:08 - 10:43 35 Min from 2.25 €

> 145 >

10:11 - 10:52 41 Min from 2.25 €

> 39A >

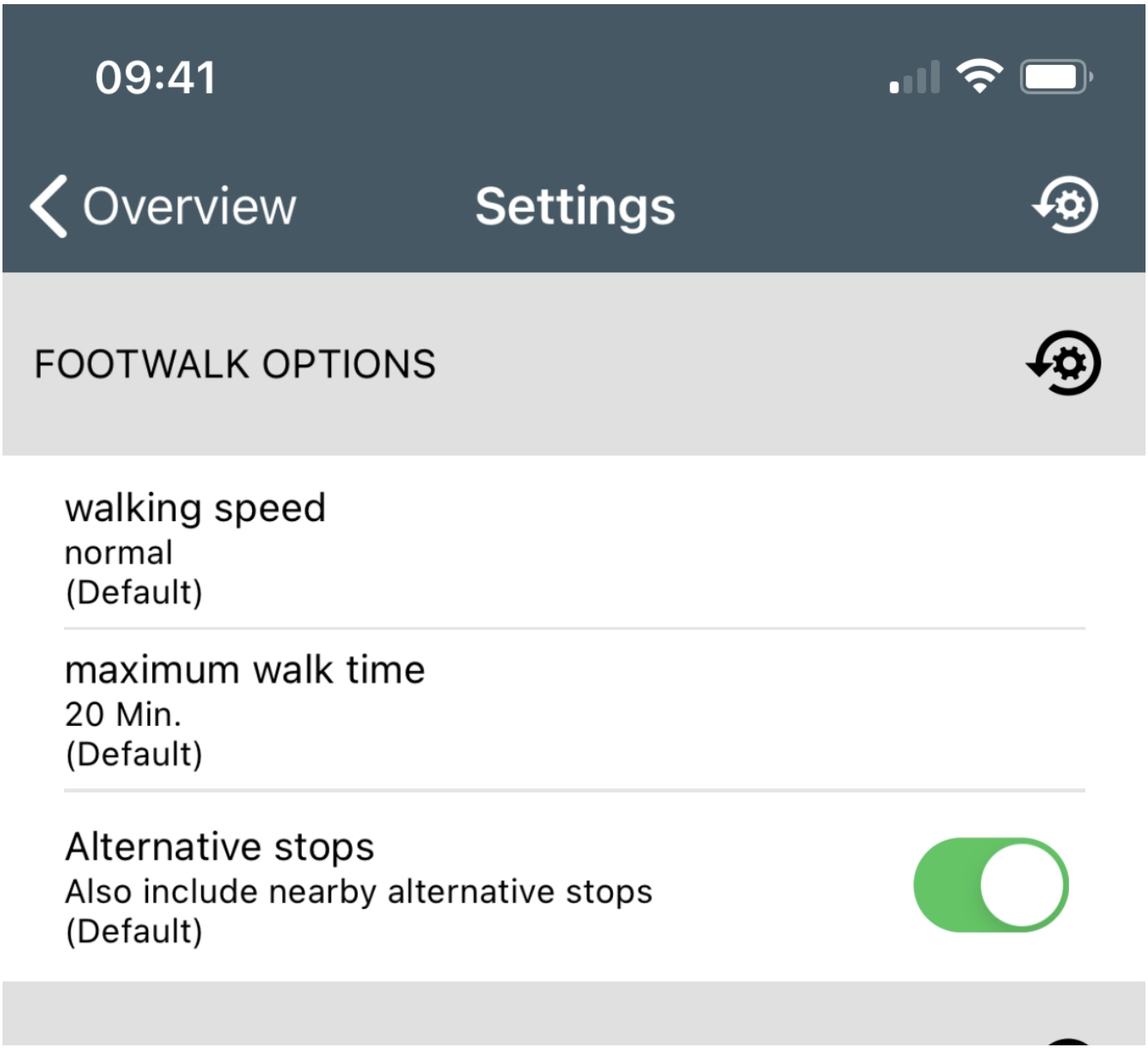
09:41 - 09:58 17 Min


09:41
Overview
Time and date
Depart
Arrive
14 June 2021
09:41
DEPARTING NOW
DONE


09:41
Back
Overview
From Smithfield
To UCD, Belfield
Dep Mon, 14/06/21 at 12:41

earlier	later

12:38 - 13:14 	36 Min	from 2.25 €
12:48 - 13:24 	36 Min	from 2.25 €
12:58 - 13:34 	36 Min	from 2.25 €
13:08 - 13:44 	36 Min	from 2.25 €
12:41 - 12:58 	17 Min	


09:41
Overview
Settings
FOOTWALK OPTIONS
walking speed
normal
(Default)
maximum walk time
20 Min.
(Default)
Alternative stops
Also include nearby alternative stops
(Default)

ALLOWED MEANS OF TRANSPORT

Route type
Quickest connection
(Default)

Train

Tram

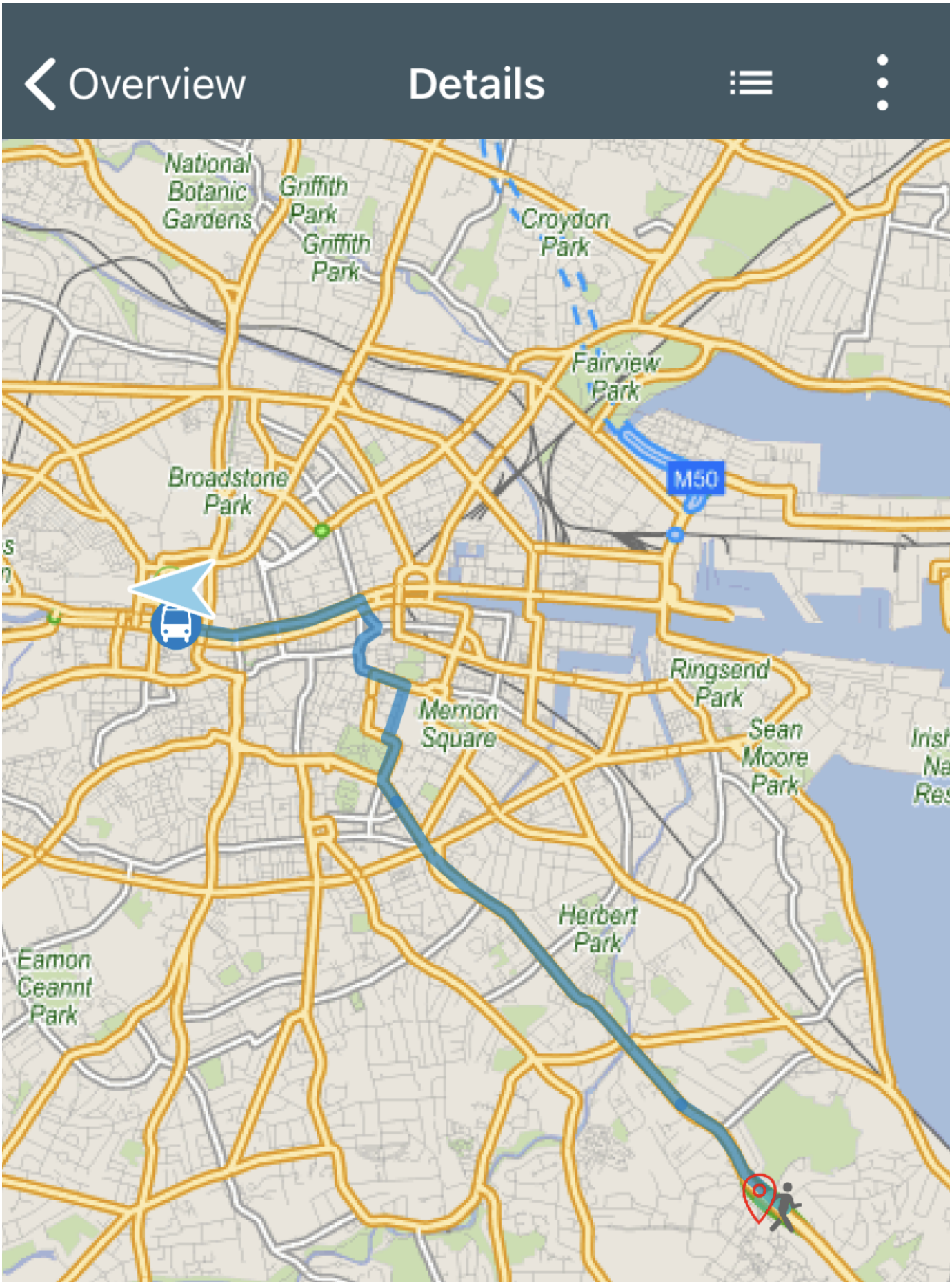
Bus services

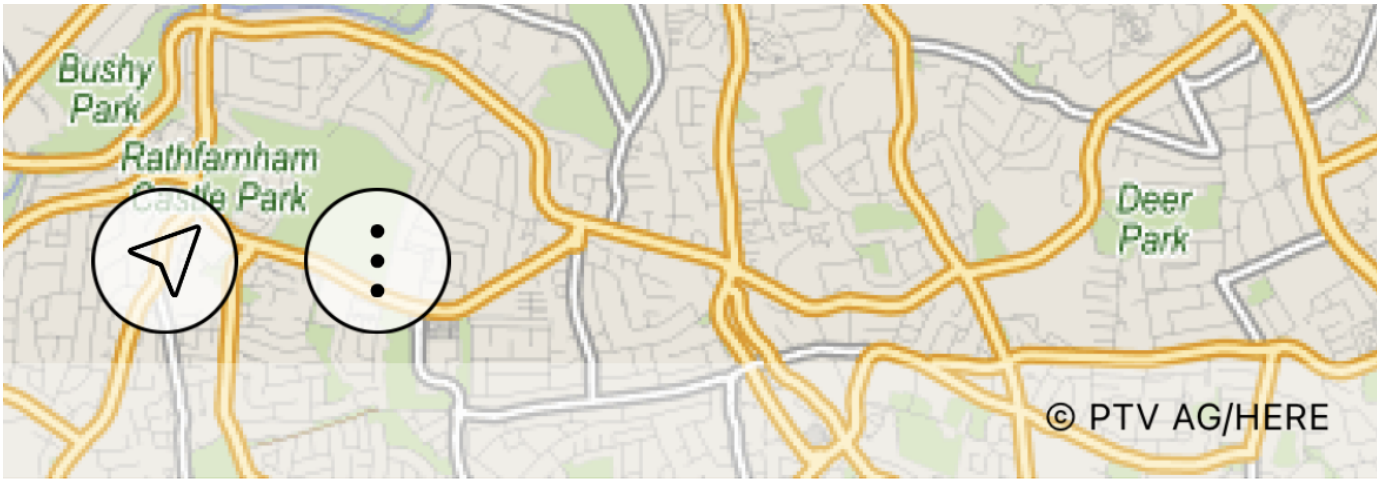
Ferry

Taxi
(Default)

MISCELLANEOUS

Use statistics


Overview
Details
National Botanic Gardens
Griffith Park
Griffith Park
Croydon Park
Fairview Park
M50
Broadstone Park
Ringsend Park
Merrion Square
Sean Moore Park
Herbert Park
Eamon Ceannt Park

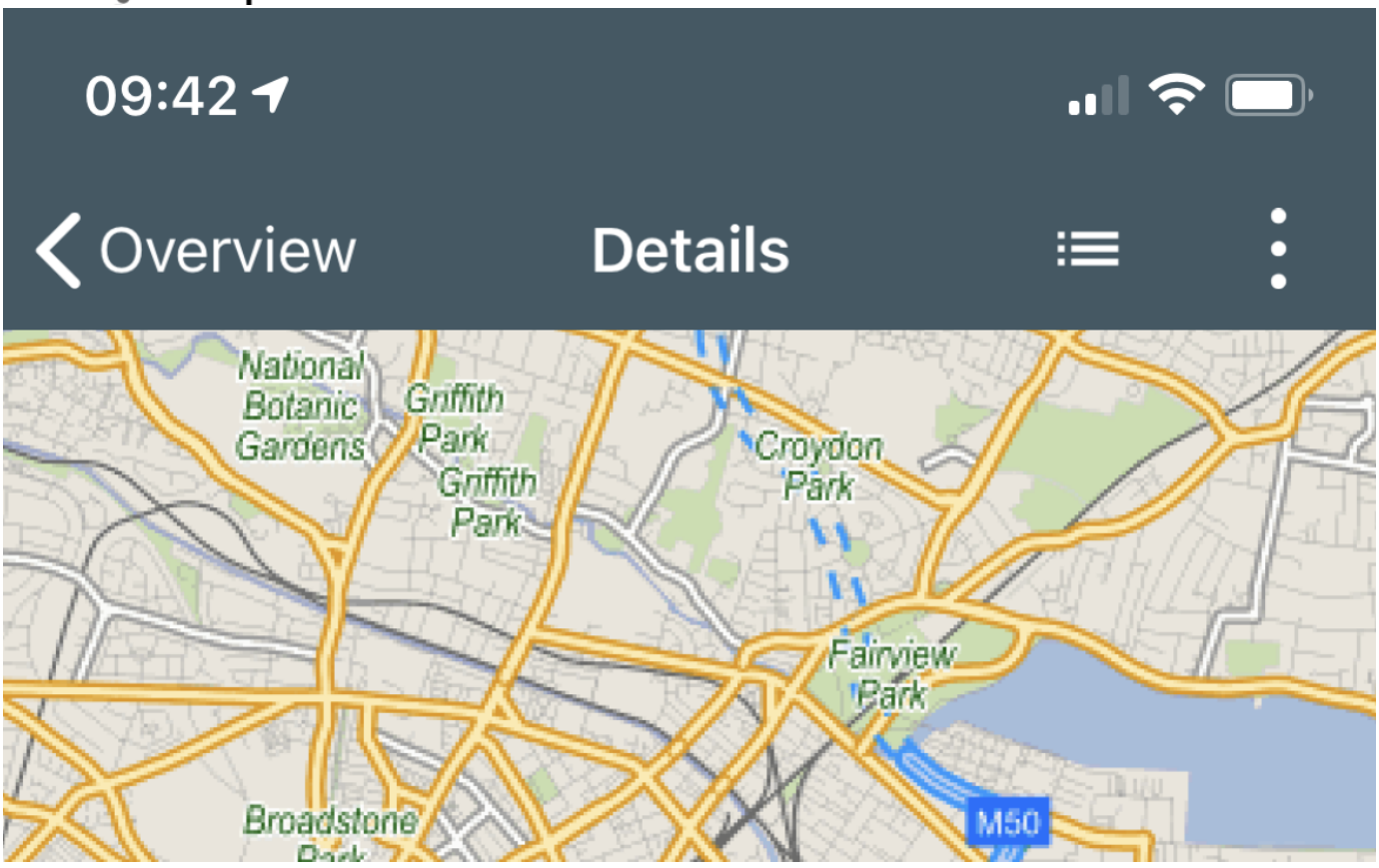


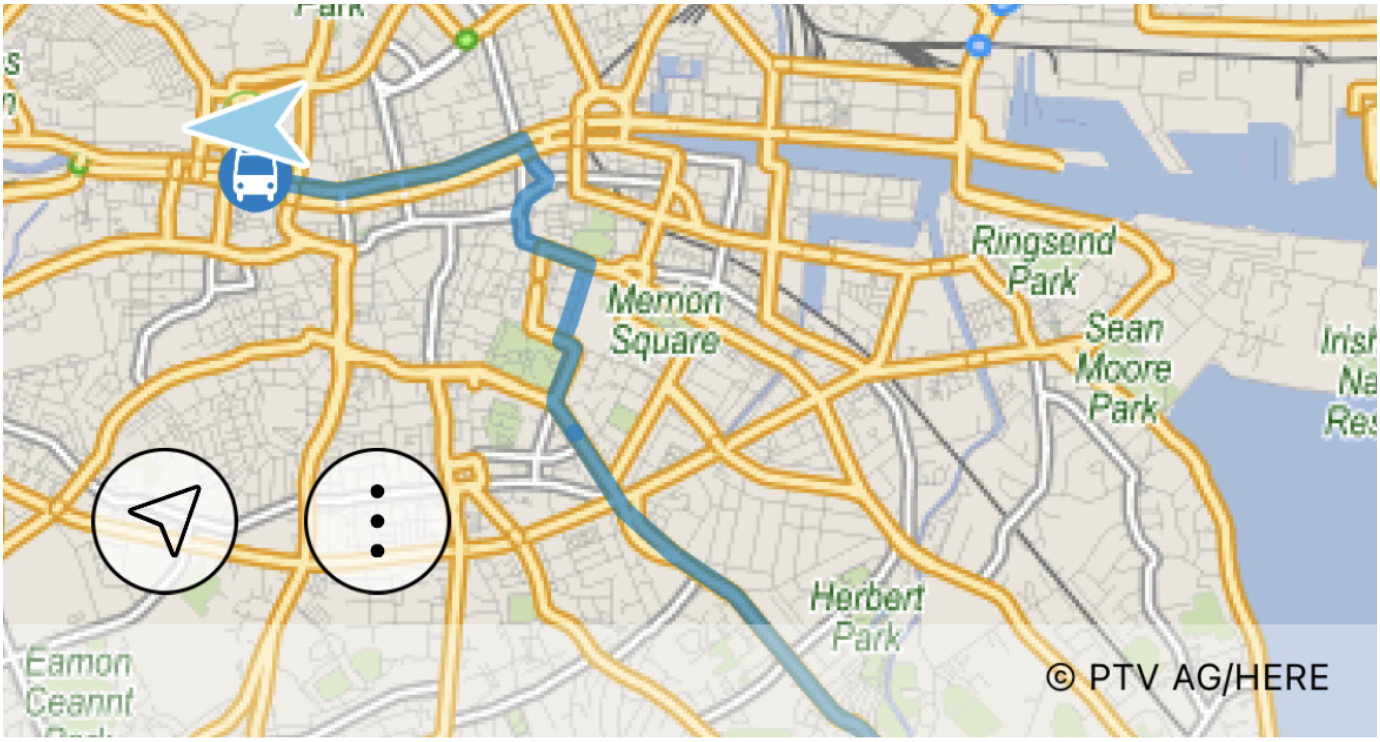
12:58 - 13:34 36 Min

> 145 >

LEAP 2.25 € Cash 3.00 €

12:58 **Smithfield**, Smithfield





12:58 - 13:34 36 Min

> 145 >

€ LEAP 2.25 € Cash 3.00 €

12:58 **Smithfield**, Smithfield

Walk: 360 m, 5 Min

13:03 **Arran Quay**, stop 7453, Smithfield

Dublin Bus 145 towards Balywaltrim

21 stops, 27 Min

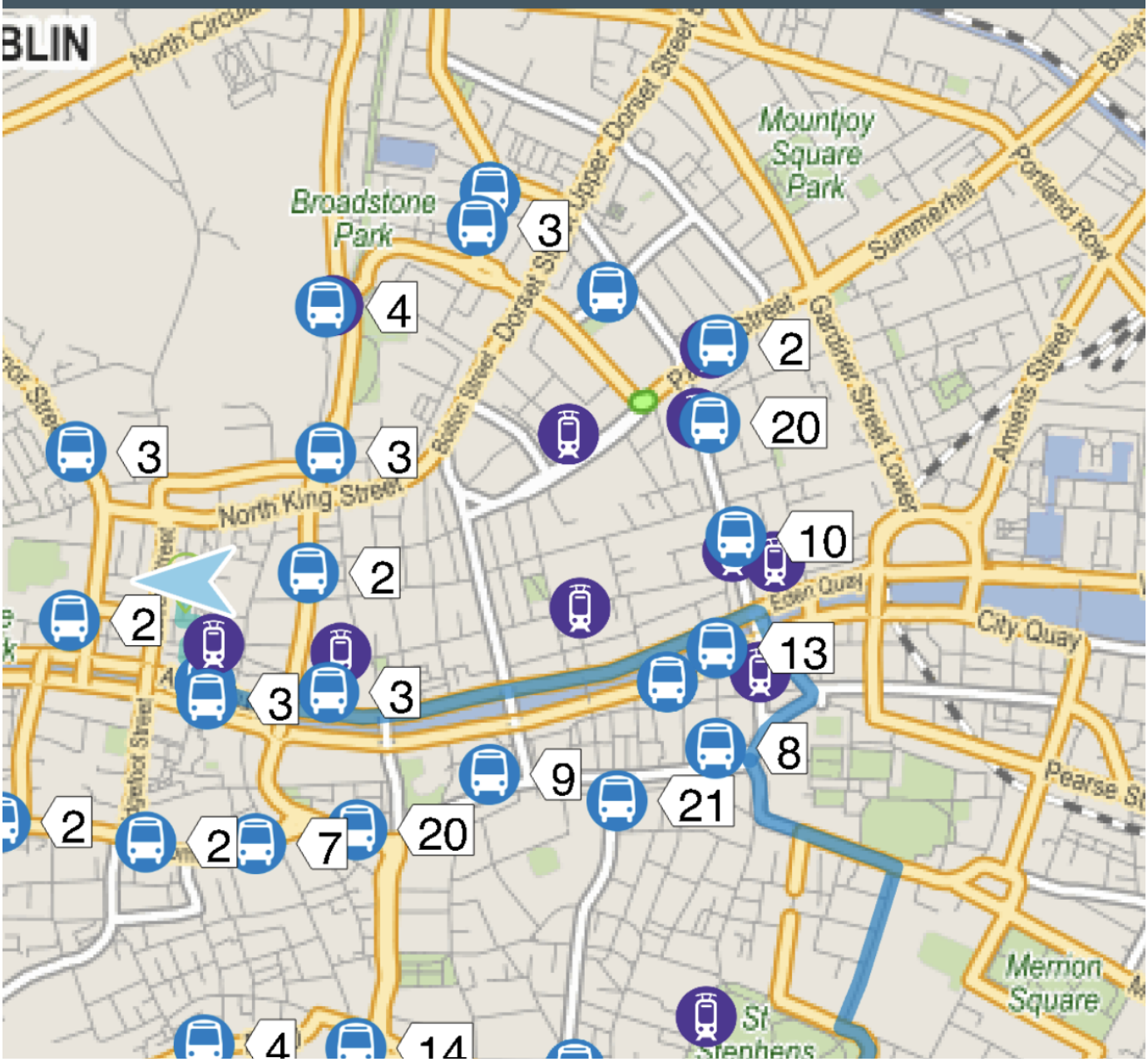
13:30 **Woodbine Road**, stop 2007, Booterstown

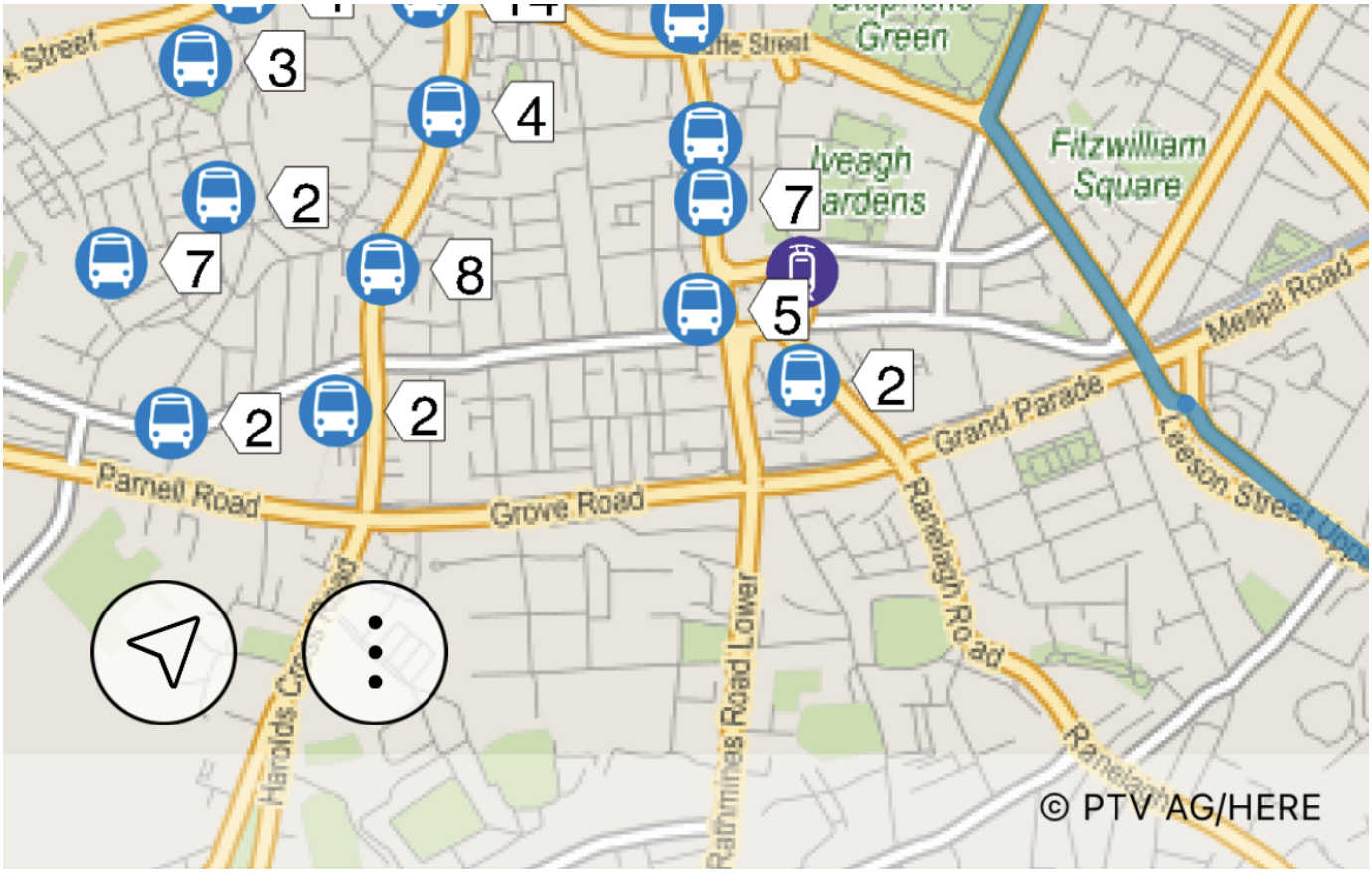
Walk: 230 m, 4 Min

13:34 **UCD**, Belfield

09:42

Overview Details





12:58 - 13:34 36 Min

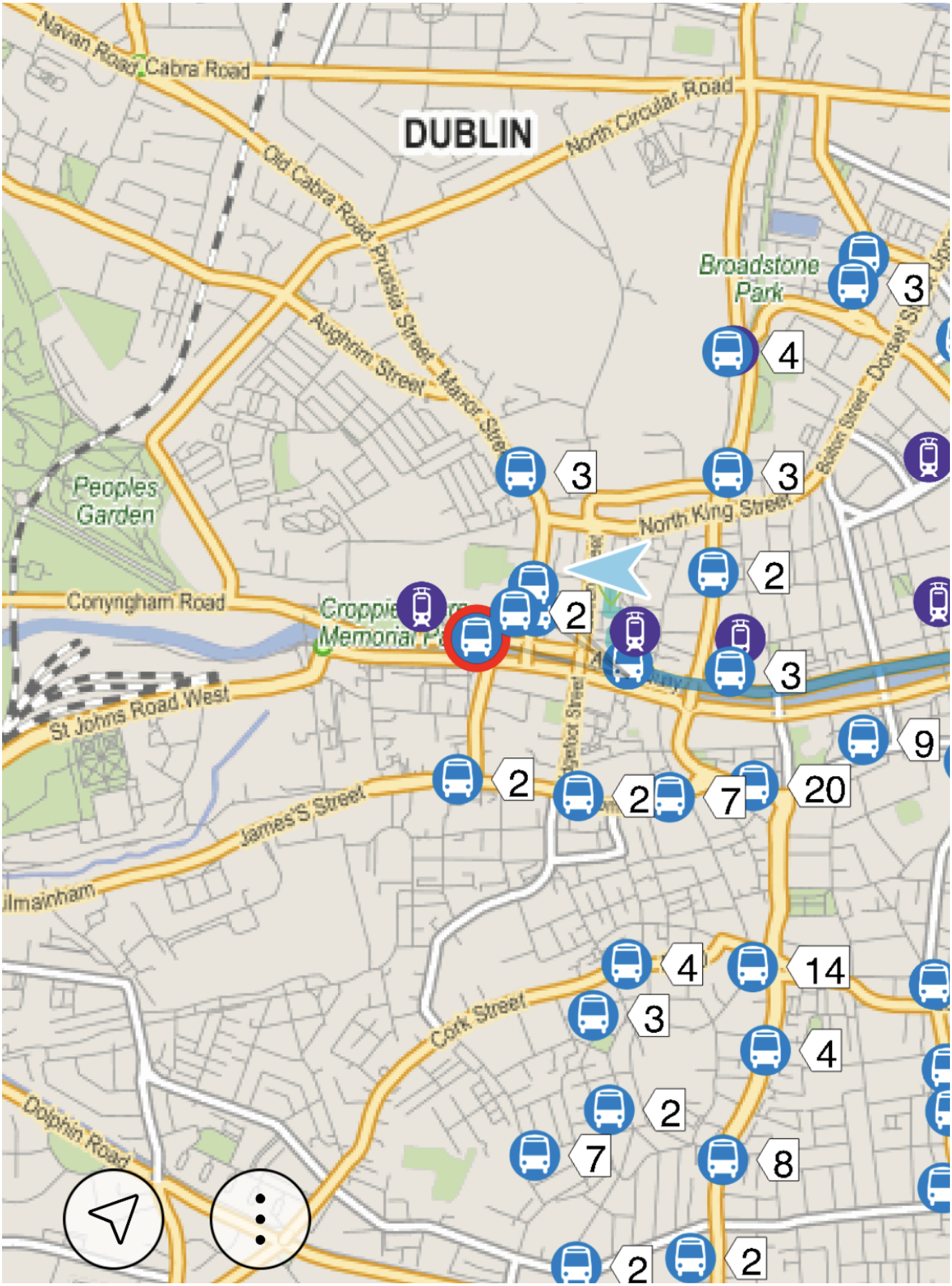
> 145 >

LEAP 2.25 € Cash 3.00 €

12:58 **Smithfield**, Smithfield

09:42

Overview **Details**


DUBLIN
Navan Road
Cabra Road
North Circular Road
Old Cabra Road
Prussia Street
Manor Street
Aughrim Street
Broadstone Park
Dorset Street
Peoples Garden
North King Street
Conyngham Road
Croppies Memorial Park
St Johns Road West
James's Street
Cork Street
Dolphin Road
3
4
3
3
2
2
3
9
2
2
7
20
4
14
3
4
2
7
8
2
2

Sarsfield Quay, stop 1476
Wolfe Tone Quay

From here | To here | departures

09:43

TFI Journey Planner

Your Regular Journeys

Home
Please select a destination

Work
Please select a destination

Information
Disruption by route

From

To

Depar

Trips
How do I get from A to B?

Street

How do I get from A to B?

Departures

Search for departures by Stop or Station

Route Diagrams

Search for departures for stops along a Route

Disruptions

What disruptions are currently affecting the transport network?

Network Maps

How can I get an overview of the network?

Settings

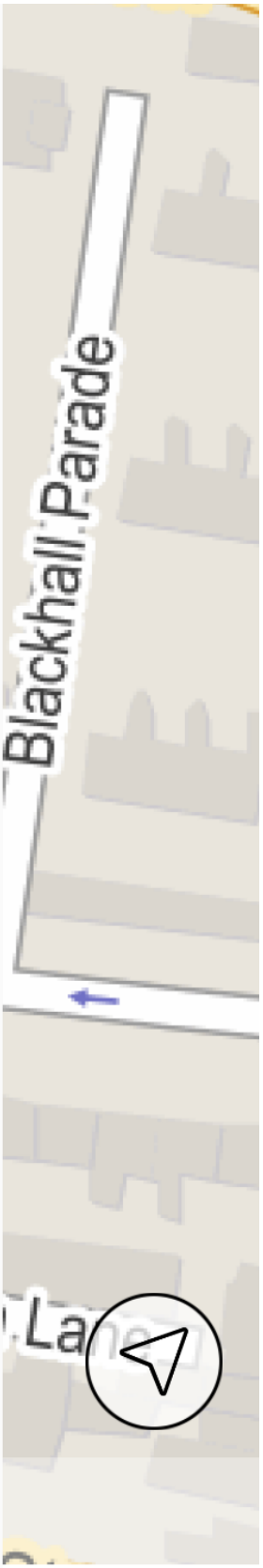
How can I adjust the app to my needs?

Info

Send feedback, legal notices, etc.

Privacy Statement

5.42.16810 (5.42.1)



Help

Please visit our

ONLINE HELP

or send us your question via

EMAIL

TFI TRANSPORT FOR IRELAND

Údarás

Náisiúnta Iompair
National Transport Authority

Journey Plan

This app is provided by:

National Transport Authority of Ireland

Address: Iveagh Court
Dublin 2

Tel. +353 1 8798300

Web www.nationaltransport.ie

MENTZ

09:43

Network Maps

Cork City 2019

As of 24/05/2019

Dublin Area 2019

Dublin Area 2019

As of 24/05/2019

Galway City 2019

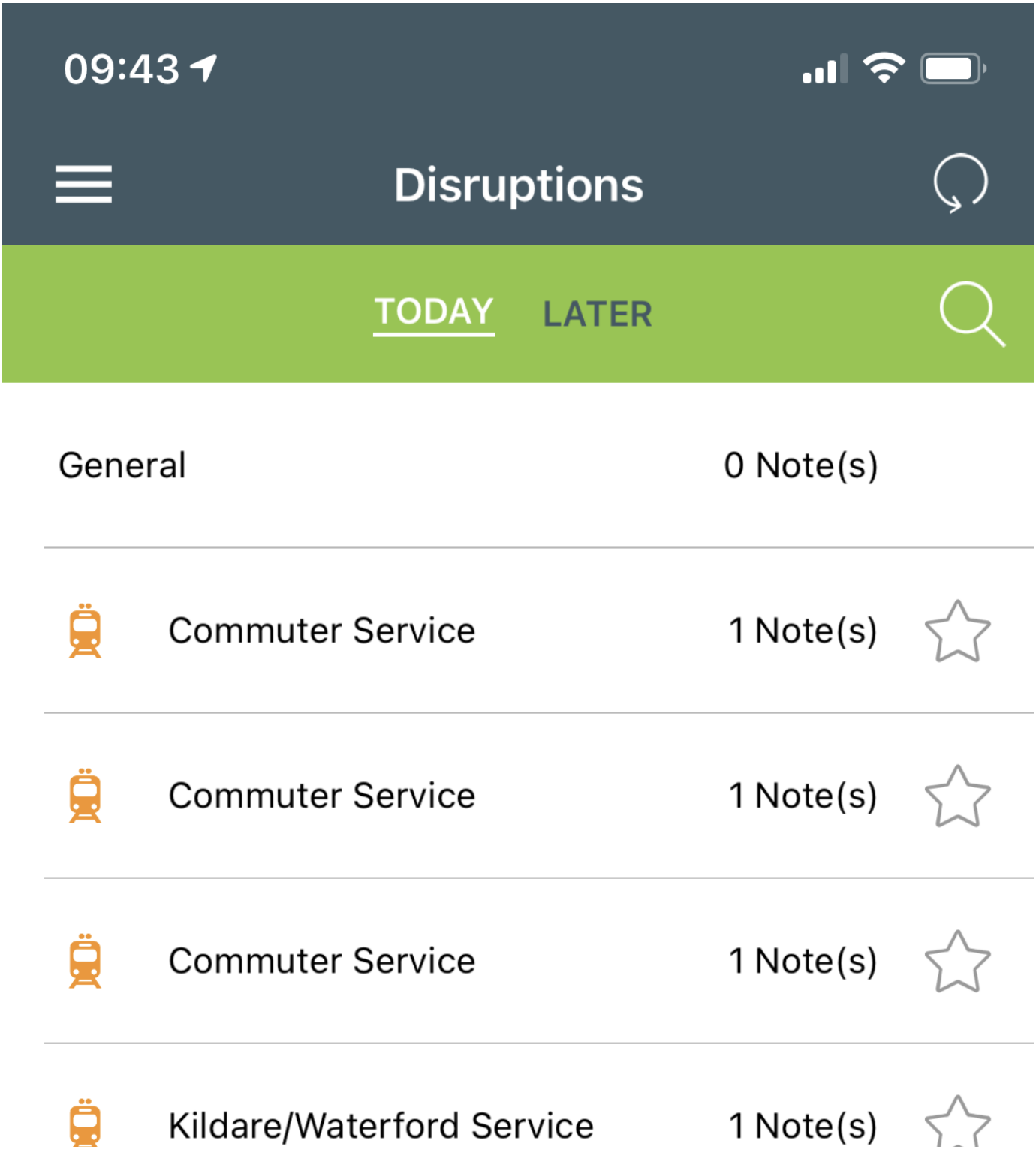
As of 24/05/2019

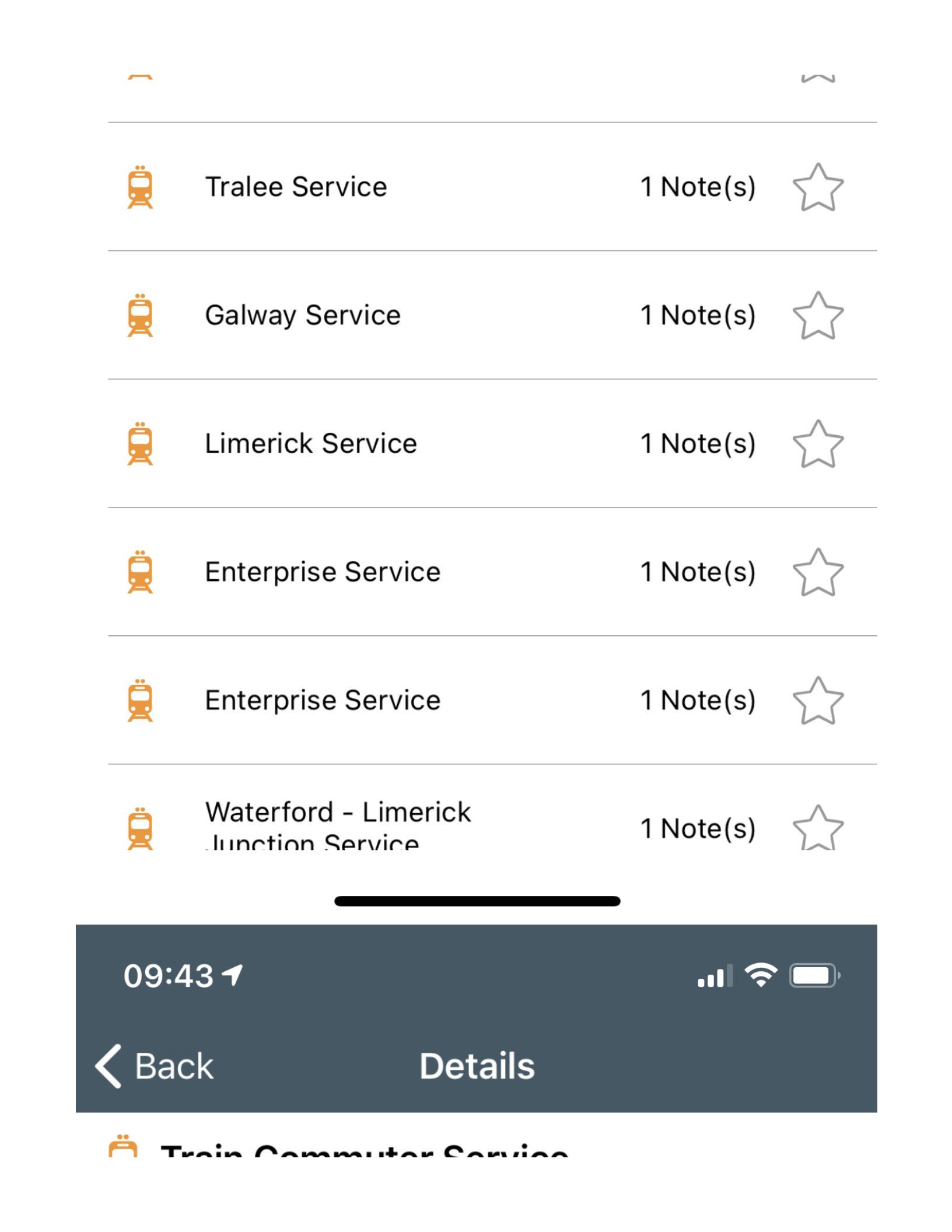
Limerick City 2019

As of 24/05/2019

Waterford City 2019

As of 24/05/2019

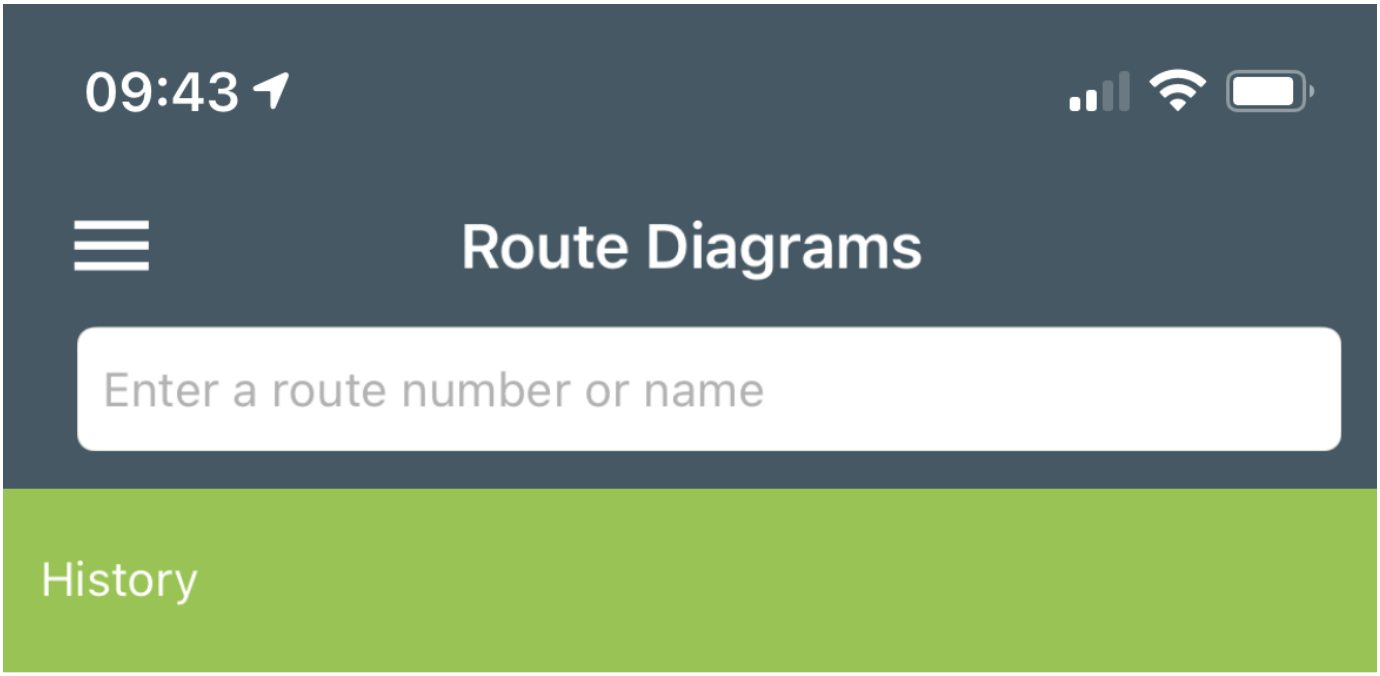

09:43
Disruptions
TODAY
LATER
General
0 Note(s)
Commuter Service
1 Note(s)
Commuter Service
1 Note(s)
Commuter Service
1 Note(s)
Kildare/Waterford Service
1 Note(s)


Tralee Service
1 Note(s)
Galway Service
1 Note(s)
Limerick Service
1 Note(s)
Enterprise Service
1 Note(s)
Enterprise Service
1 Note(s)
Waterford - Limerick
1 Note(s)
09:43
Back
Details

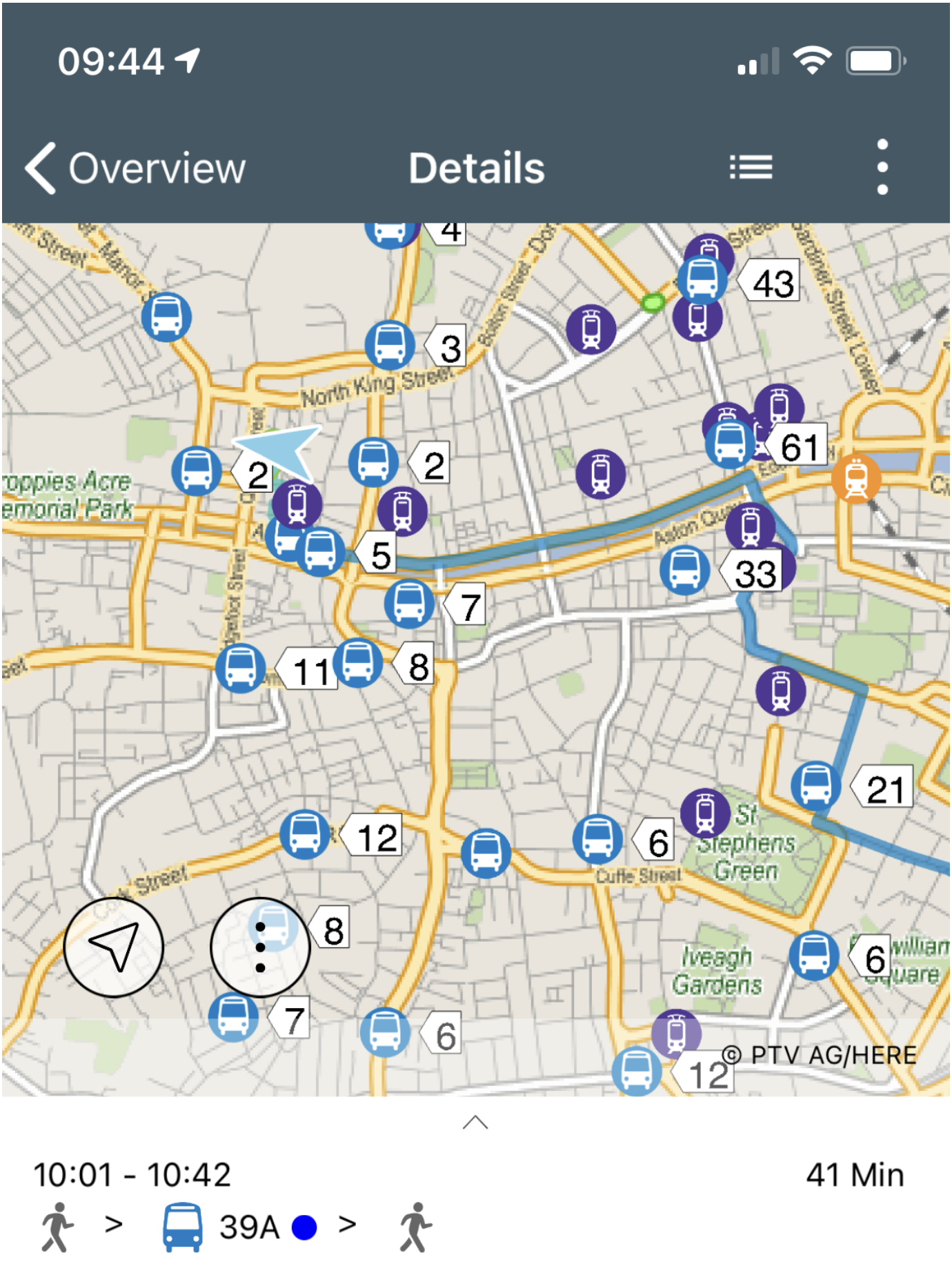
Train Commuter Service

Web Fares available online

Discounted webfares are often available at www.irishrail.ie, especially when purchased in advance.



In this list you will find your most recently uses lines. To fill this just search and select a line.



10:01 - 10:42 41 Min

> 39A >

LEAP 2.25 € Cash 3.00 €

10:01 **Smithfield**, Smithfield

Walk: 360 m, 5 Min

10:06 **Arran Quay**, stop 7453, Smithfield

Dublin Bus 39A
towards Belfield, University College Dublin

REAL TIME

23 stops, 30 Min

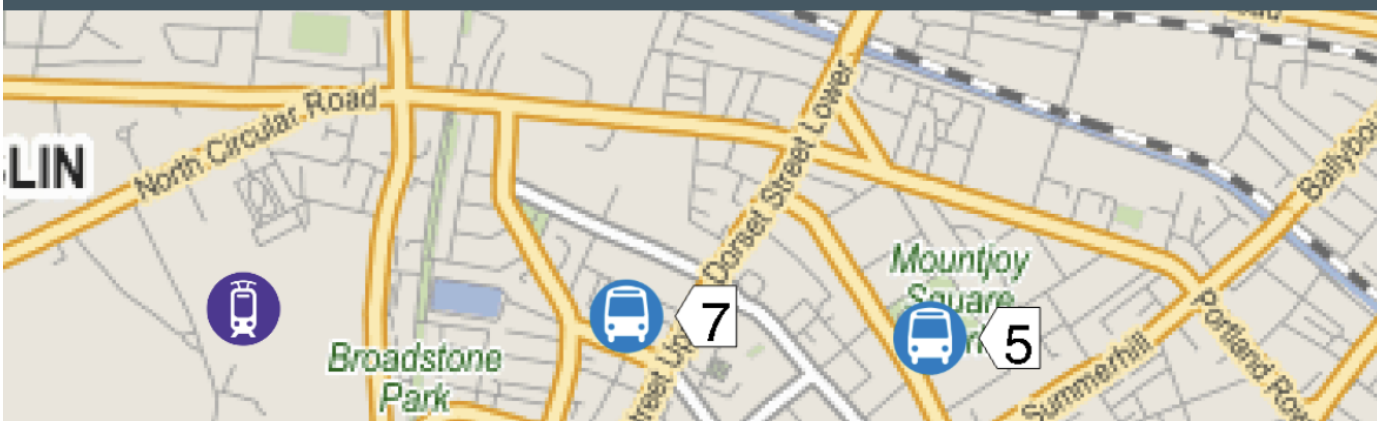
10:36 **UCD**, stop 767, Belfield

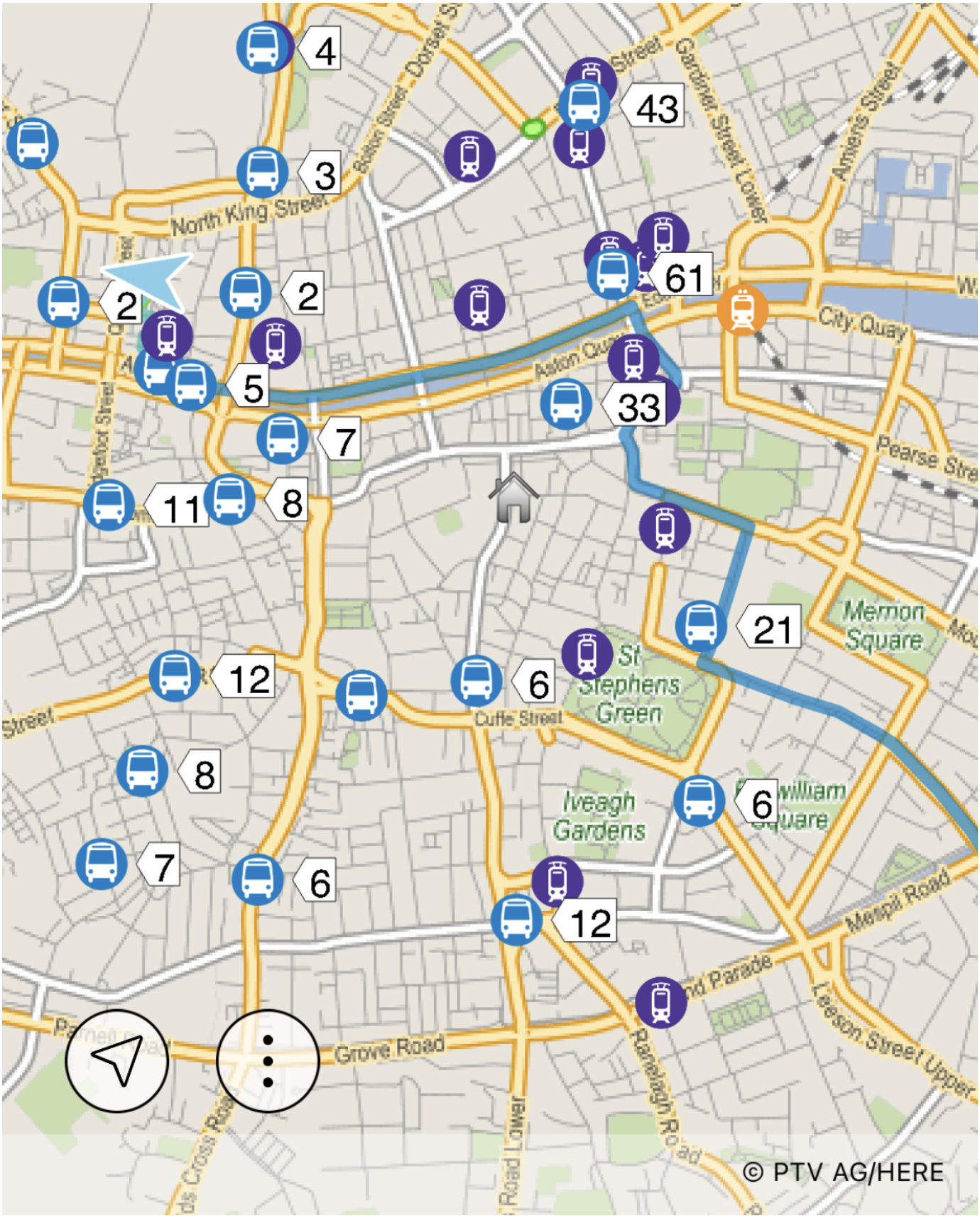
Walk: 560 m, 6 Min

10:42 **UCD** Belfield

09:44

Overview **Details**





South Great George's Street
Dublin City South

Page Title Here

Brief Description

Page	Title	Name

System Design

This section is to document the system design.

Data Exploration

This page contains details on the analytics discovery phase.

Machine Learning Models

This page should contain details on the Machine Learning Methods and Features Tested.

We would like to use the background knowledge to create a model. As this should speed up the process of creating a model without testing for all the features through permutations. We can use data exploration and correlation plots to justify our choices, but I believe that it is better to go after causality than to go after correlation.

As this would mean that we understand the data and that the outcome of the model can be easily interpreted by users, which is a good thing in my opinion.

The models that we should try to use are models that will work in theory for out data domain, rather than solely trying to obtain the lowest error. So rather than dumping things into the model we should try to use our background knowledge as best as possible to choose the features.

The models dont have to be too computationally expensive for it to be usable and for it to be a good model. A good model would be a model that gives us a decent result and the user trusts the result also. If the result is amazing, but the users does not trust the model, then there is no point as the user may not believe the result. similar with the other argument, if the user recieves a bad result from the model and completely understands the model, then the question can be asked “why are you using this bad models for this task?” .

So a balance should be struck between these things, so that the most optimal AI is used for the task at hand.

There are a few models that we can try linear models can work, but would not recommend as some of our features are not linear, and there is no clear relational between some features and the target.

We could use a random forest as this can treat non-linearity and outliers.

We could use a recurrent neural network of some sorts as the data itself is of the sort that is based on previous data.

There are many that we can use, but we do not know which model would be the most optimal combination of trustworthiness and accuracy. Because if we do not know/ understand the inner workings of a model, then the odds are the user would be less likely to understand the workings of the model and how the result can be trusted. Because I believe that simply telling the user, this is the result is not enough, we should show them the reasoning/proof behind the result.

Model 1 : time Series Analysis

he task at hand is to create a model form the historical dataset. The conclusion from exploring the data is that the degree of applicability is very low

Background

There are some things that we can extract from the the historical dataset that I believe can be used, would like thoughts about this from you guys.

From looking at the dataset, the dataset is a time series dataset, so theoretically, a time series model would be great for the model. But for the feature itself there is a lot of noise as it is a real world time series. I tried to create a time series model using Prophet, a package published by Facebook for time series data/models.

Because of the nature of our data, even though we have a time series, the time series can be thought of as a time series that begins at the start of the year 2018 and ends at the end of the year 2018.

But we can also think of our dataset as a number of different time series, whereby each starts at the start of each day. But then there would not be any way to predict for the next day. The model would mean we should not apply the historical dataset. Also, because there is a discontinuous timeline which occurs in real life due to scheduling, a time series model may not be the best.

Because we have such a large dataset, a more effective way to use the time series would be to create a more general time-series, which may not be as applicable in the real life scenario, or user friendly.

Features

The features I decided to use, are the planned arrival times

The y-component value I decided to use the delay time in seconds.

ETL Process Description and Logic

the planned arrival time can extract from bus schedules schedules and we can extract from the historical dataset by using the “dayofservice” and the planned arrival time at the stop. We can combine them together to form a time value.

The delay time can be found by subtracting the actual arrival time form the planned arrival time. This means that the value can be negative which means the bus was early and positive which means the bus was delayed.

I removed some of the outliers for which the bus were delayed by too long, such as more than 30 minutes as this is an anomaly, only happened 30000 times out of 1.3 million bus arrivals.

Prophet would create a time series model from the time series data.

Result

The result is to be expected, with time series analysis forecasts, we can get the expected value and the variance of the expected value. So the forecast is essentially a range of values for which we are sure the forecasted value falls into.

The result is not great as the range given to us is very large since we are very far into the future compared to the time of data collection.

```
In [58]: forecast = m.predict(current_time)
         forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

Out[58]:

	ds	yhat	yhat_lower	yhat_upper
0	2021-07-06 17:23:53.727602	-8511.387137	-47843.163465	35923.847228

The image above shows the range which is a very large range which is to be expected, since the future is not a very predictable thing. so the further into the future the less sure the model becomes.

Conclusion

So if we were in 2018/2019, time series would be a good model for the data provided, but since we are in 2021 it is not exactly a model that will give us accurate expected forecasts but the forecast range is not incorrect in the sense that the delay time will definitely fall in the range.

Recommendations

- For 2021 I would not recommend time series analysis due to the time that has passed
- The amount of data is quite significant in size so the training time for the model is quite long

Model 2 : Random Forest Regression

Random forest model for the historical bus and weather data.

Background

From the historical dataset we have categorical and numerical data. Some of the numerical features should be treated as categorical and others as continuous features. The target that we decided to use the delay time, which is the difference between the actual arrival time of the bus and the planned arrival time of the bus at each stop. The reason we decided to go with this approach rather than to try to make the target the arrival time at the stops, is because there has been changes to each bus route that existed in 2018. Changed such as the number of stops, the roads it takes and each schedule changes. So, we deemed that it would be more appropriate to look at the time that was delayed for each stop, since even if the routes/ bus stop sequence changes, the overall direction of the bus did not change too much.

We need a model that can handle different types of data. Linear regression as a whole cannot handle large amounts of categorical features well as we would need to encode the features in such a way that the dimension of the input feature space, which will impact the model as a whole, but this does not mean a linear regression would not work. Which is why I decided to create a random forest model for the task at hand.

As we have decided to create models for each route, I decided to create models for a number of routes to evaluate the models and see how the models perform. The routes I picked were all at random from the set of routes that existed in the 2018 dataset. I performed a train test split by training on the first 70% of the data with respect to time, then I tested on the rest of the year as that is the purpose of our models, to be able to somewhat predict the future to some degree of accuracy.

For the random forest there are a series of hyper-parameters that needs to be tuned to achieve an efficient model. The hyper-parameters that I had tuned are the number of tree in the forest, the depth and the number of features that are considered at each split. I had tried out 3 values of for each hyper-parameter, and ended up with the parameters that give us the best model.

Features

The features from the bus data that I decided to utilize are the time values and weather data.

Time values:

From the date, I extracted the month and day value, and created an index for if the date is a holiday or not. The criteria that i utilized are if we are in the summer months of June, July, August as these are school holidays and I expect there are large amounts of travel in the social environment.

Other criteria are if we are on a national public holiday such as if it is a bank holiday or not.

For the time I also created a hours and minutes columns for the planned arrival times as I do not know if the schedule has changed since 2018 so I think the time of day would be important but not an exact time as given in the dataset, which is in seconds.

The other data I decided to include are which stop we are arriving at, as in the sequence number of the stop.

I plan to include the delay time of the previous stop and also the id of the current stop. but there are some issues in mind with the encoding /inclusion of those features.

Weather Values:

After looking at the weather data for the five routes that I have tested, I compared the weather data that we have available with respect to the target feature. Most of features that did not follow a any trend when compared to the target feature.

The only features that did not follow a constant flat-line trend was the feature for rain-fall for 1 hour, and the classification of the weather of the time interval. There are entries with missing values, but if we look at the classification of the weather at that interval, it can be implied that the rain-fall for that 1 hour interval is 0 for those missing values. So I compared all of the missing values with the weather classification so that I know for sure that the values should be 0. I encoded the data

ETL Process Description and Logic

The historical data has been split into individual routes, so the bus data is extracted for a single route.

The categorical data can be encoded using one hot encoding or just kept as is, due to the nature of the random forest model, decision trees. We just need to encode any non-numerical categorical data into numbers. The magnitude of the categories should not matter significantly to the decision trees.

I left the numerical data as is, as the random forest can use the values. The weather categorical data I encoded with one hot encoding, or dummy encoding. As there is no ranking/relation between the categories. The weekdays I decided to encode in two separate methods. The first one I tried was also one-hot encoding. The second I tried was to form a direct mapping from the range of 1 to 7 to Monday to Sunday. As there is a relation between the days of the week which is more clear cut than the weather categories.

Result

From training the models I found that the mean absolute error was between 200 and 210 seconds, which means that the mean error is less than 5 minutes away from the actual arrival time.

The encoding from using one hot encoding for the weekdays presented me with an error that is 10 seconds less than the encoding using the numerical mapping.

I also trained the models using no weather information, which yielded an error that was 30 seconds greater than with the weather features I choose.

I also tested the data using linear regression but for the routes I choose it had a greater error by around 30 seconds for the routes.

Conclusion

I believe that the model that I have settled on is the best of the models that I decided to try out. From the results on the models that I have tested. I believe that the model itself is has an error that is not too high. That the hyper-parameters have been tuned as best as possible. The hyper-parameters were chosen so that we do not overfit the data to 2018, which is quite different to the data that will be presented in 2021, due to many factors such as restructuring, new bus routes, green initiatives and etc.

Recommendations

One thing I would expect is that the error would be quite high for our present 2021 data, but not too much higher. But I would have rathered if we had more recent data for the training of the models os that the models are more accurate.

Other models

We also considered using linear regression as a baseline model, we performed the same processing on the data for the linear models, we created linear models with the same data on the same routes so that we can compare the model performance using different metrics. So, we trained 4 linear regression routes that were initially chosen at random and we compared the metrics for the random forest models on those routes.

The findings were that the metrics measured very similar values. So there was not much improvement in the performance of the models just from a numbers point of view. But, if we were to compare the actual results of the models we see that linear regression will give us an output that follows a normal distribution which is to be expected. But the testing data does not have a normal distribution as the distribution of the output. But from the random forest we get a distribution that is similar to the actual testing data. Which means that it is a better model for predicting the feature that we want.

There are other models that we would have liked to consider such as neural networks and xgboost. But the main concern with these models is that they are a lot more computationally complicated than the other models that we looked at. So they will take up more time when training the model, especially with the fact that out training data is still of a substantial size. This time spent may lead to an decrease in error but the time spent verses the actual error decreased may not have been worth the effort. That was my main concern.

As the resources on the server was shared, I believed that doing something that extensive would be unfair to everyone else using the server, and when everyone would be training models at the same time the resources would be shared, this increasing the training times even more. When our group tested a single xgboost model to compare, the training time took considerably longer than the other models considered. That is why we did not end up going with those models.

Application Structure

This page is to detail the application structure.

App Folder Structure

This page details the folder structure of the Django App

	Primary Folder	File	Primary Codebase	Description
1	static	javascript.js	Javascript	Javascript Functions
2	static	stylesheet.css	CSS	Stylesheet
3	models	varied	pickled models.	ML Models
4	templates	base.html	HTML/Jinja	Template Extension
5	templates	about.html	HTML	About Page - About This App
6	templates	index.html	HTML	Index Page
7	Main	__init__.py	Python	Initialiser
8	Main	config.py	Python	App Config
9	Main	data_dictionary.py	Python	Static Variable dictionary, Database connection details, API services
10	Main	sql.py	Python	Model Queries

11	Main	tests.py	Python	Test Statements
12	Main	methods.py	Python	App Methods
13	Main	app.py	Python	App Routes
14	Main	models.py	Python	Models for the App.

Import Structure

This page details the import structure within the app to avoid cyclic imports.

Routes

This page details the routes in the app.

	Route	RouteName	Methods
1	home, index, (blank)	`home`	`Method1`
2	debug	debug_hello	none
3	`route`	`route`	`M`
4	`route`	`route`	`M`
5	`route`	`route`	`M`
6	`route`	`route`	`M`
7	`route`	`route`	`M`
8	`route`	`route`	`M`
9	`route`	`route`	`M`
10	`route`	`route`	`M`
11	`route`	`route`	`M`
12	`route`	`route`	`M`
13	`route`	`route`	`M`
14	`route`	`route`	`M`
15	`route`	`route`	`M`
16	`route`	`route`	`M`

Tests

This page details the tests which are implemented.

	Test Function	Tested Method	Pass Condition	Fail Outcome
1	`connection_test`	`connect_db_engine`	`Database Connection Established`	Database connection method returns error.
2	`run_tests`	`All runnable tests`	`All tests True`	Django App Fails to Run

DevOps

This page is to document the Development Operations process (i.e. Git Process).

DevOps Process

Discussion on Devops Process

The {Name} development branch is the primary development branch due to the small team size. Within this branch a **{WorkflowName} Workflow** (https://www.atlassian.com/git/tutorials/comparing-workflows#centralized-workflow) process was enabled.

The team process for conducting git pushes to features therefore became:

1. git pull
2. git add <file>
3. git commit -m '<Message>'
4. git push

Git Branching Graph

Current GitHub repository is located at the following link: (link here)

Integrations

These pages are designed to document data integrations.

Dublin Bus Data

This pages details the bus data integration

RT_Trips Data

This page details integration details (keys) of the application to the historic trip data.
The Table Below Lists the Integration Details and Endpoints.

The info returned is a csv

For all integrations the appid parameter must be added to connect. Eg.

	Description	Service Endpoint	Parameters	Keys
1	Location	`CS UCD`	N/A	`N/A`

Data Mapping

Description

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	`DATASOURCE`	VARCHAR(4)	No	Yes	Yes	DB	No	DublinBus - Org Identifier
2	`DAYOFSERVICE`	DateTime	No	Yes	No	07-FEB-18 00:00:00	No	Day of Service - Days More than 24 hours
3	`TRIPID`	VARCHAR(30)	No	Yes	Yes	6253783	No	Identifier for a trip - a journey taken for a bus
4	`LINEID`	VARCHAR(10)	No	No	Yes	68	Yes	Identifier for a line - a series of routes a bus might take grouped together
5	`ROUTEID`	VARCHAR(20)	No	No	Yes	104_16, 104_15, etc.	Yes	Individual Journeys taken by buses
6	`DIRECTION`	VARCHAR(2)	No	No	No	1,2	Yes	Forward or Backwards
7	`PLANNEDTIME_ARR`	INT64	No	No	No	87245	Yes	Planned time of arrived in seconds
8	`PLANNEDTIME_DEP`	INT64	No	No	No	84600	Yes	Planned time of arrival in seconds
9	`ACTUALTIME_ARR`	INT64	No	No	No	87524	Yes	Actual time of arrival in second
10	`ACTUALTIME_DEP`	INT64	No	No	No	84600	Yes	

								Actual time of departure in seconds
11	BASIN	VARCHAR(20)	No	No	No	BasDef	Yes	BasDef - Unclear.
12	TENDERLOT	VARCHAR(30)	No	No	No	Tbd - NaN	Yes	Unclear
13	SUPPRESSED	INT32	No	No	No	NaN, 0	Yes	0 if it's suppressed. These contain only planned data. Partially suppressed trips are not flagged. For some reason only 0 is present but: 0=Achieved 1=Suppressed
14	JUSTIFICATIONID	VARCHAR64	No	Yes	Yes	177856, 297896	Yes	Specific fault codes for trips - These are not unique.
15	LASTUPDATE	Field Type	No	No	No	28-FEB-18 12:05:11	No	Date last modified
16	NOTE	Field Type	No	No	No	,2967409,	Yes	Free text - Looks like an ID of some type?

Documentation Summary:

API documentation available here:

RT_LeaveTimes Data

The fields in this dataset are similar to the rt_trips_2018 dataset.

There are various details that can be noted between the two datasets.

This dataset contains the information about all of the buses and the times at which the bus stops at every single bus stop along its route. It also contains the details regarding to the day of service, the source of the data, the timetable expected times and the actual expected times of arrival and departure. There are also the identification for which route and which line of the route the bus belongs to. There is a suppression index, which tells us if there are any issues with the data, such as if there are values that are missing in the actual arrival/departure times. There are also, justification for the suppression which is a code that explains why the data becomes suppressed. There is also a note on the particular bus at each stop.

The data quality issues that we have are mostly with the suppressed, justification and notes column. The suppressed and justification columns exist with each other as the justification is dependent on the suppression. But there are so many missing values that it is most likely unable to contribute anything to any models that are constructed due to the bias in the representation of the feature. But I would still expect the feature to

be of some use. The justification and notes features, have no missing unless we can understand what the entries mean, because they seem to be in code. But we do not have a key for the codes. So these two columns have no logical meaning to us which indicates that we are unable to use these features, because using information blindly will lead us into future problems.

GTFS-R Data

This page details integration details (keys) of the application to the realtime trip data.

The Table Below Lists the Integration Details and Endpoints.

The info returned is a json

For all integrations the appid parameter must be added to connect. Eg.

`{URL HERE}`

	Description	Service Endpoint	Parameters	Keys
1	Endpoint	`Eg?var1={var1}&var2={var2}`	{var1} {var2}	`API KEY`

Data Mapping

Description

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	Trip id	String	Yes	Yes/No	Yes/No	`1954939.1.10-64-e20-1.155.O`	No	The unique identifier of a trip
2	Route ID	String	No	Yes/No	Yes/No	`10-64-e20-1`	No	Identifier of the route, there is a mapping to the short version which appears on the bus.
3	Start Date	Date	No	Yes/No	Yes/No	`20210627`	No	The date on which the bus is travelling
4	Start time	Datetime	No	Yes/No	Yes/No	`07:15:00`	No	The time at which the the bus leaves the initial station
5	StopTimeUpdate	List	Yes/No	Yes/No	Yes/No		Yes/No	The list of stations at which an update is made
6	StopTimeUpdate-stopSequence	Int	No	Yes/No	Yes/No	1	No	The stop number along the route at which the bus stopped
7	SopTimeUpdate-arrival	Dictionary	No	Yes/No	Yes/No	`{'delay': 480}`	Yes/No	information regarding the arrival of the bus at the station, what time it arrived at, how many seconds the bus is delayed by
8	SopTimeUpdate-departure	Dictionary	No	Yes/No	Yes/No	`{'delay': 480}`	Yes/No	Details about when the bus departed, the timestamp, how many seconds the bus is delayed when departing the station/stop
9	StopTimeUpdate-stop_id	String	No	Yes/No	Yes/No	`'8490B5531001'`	Yes/No	The identification of the stop, there is a conversion from the formal id

								to the id shown on the sign in real life
10	StopTimeUpdate-schedule_relationship	String	No	Yes/No	Yes/No	'SCHEDULED'	Yes/No	If the trip is a scheduled trip or not
11	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
12	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
13	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
14	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
15	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
16	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
17	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
18	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
19	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
20	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
21	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
22	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
23	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
24	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
25	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
26	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field

Documentation Summary:

API documentation available here: Link

Weather Data

These pages detail the weather integrations:

Historic Data

This page details integration details (keys) of the application to the historic trip data.
The Table Below Lists the Integration Details and Endpoints.

The info returned is a json

For all integrations the appid parameter must be added to connect. Eg.

`{URL HERE}`

	Description	Parameters	Keys
1	Website	URL	`https://history.openweathermap.org/storage/df6333d386c1a7b47a47d42b452964ed.csv`

Data Mapping

Description

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	`dt`	BIGINT	Yes	Yes	No	1545102000	Yes/No	Datetime - Timestamp
2	`dt_iso`	DateTime (ISO)	Yes	N/A	No	2018-04-12 10:00:00 +0000 UTC	Yes/No	Datetime
3	`timezone`	INT	No	N/A	No	0,3600	Yes/No	Shift in seconds from UTC
4	`city_name`	VARCHAR(6)	No	N/A	No	Dublin	Yes/No	Dublin

5	lat	float64	No	Yes	No	53.349805	Yes/No	53.349805
6	lon	float64	No	Yes	No	-6.26031	Yes/No	-6.26031
7	temp	float64	No	No	No	280.16	Yes/No	Temperature in Kelvin
8	feels_like	float64	No	No	No	275.54	Yes/No	Temperature in Kelvin
9	temp_min	float64	No	No	No	279.15	Yes/No	Temperature in Kelvin
10	temp_max	float64	No	No	No	289.15	Yes/No	Temperature in Kelvin
11	pressure	int64	No	No	No	1017	Yes/No	Air Pressure
12	sea_level	float64	No	No	No	NaN	Yes/No	Null
13	grnd_level	float64	No	No	No	NaN	Yes/No	Null
14	humidity	int64	No	No	No	93	Yes/No	Humidity
15	wind_speed	float64	No	No	No	3.6	Yes/No	Wind speed
16	wind_deg	int64	No	No	No	250	Yes/No	Wind direction
17	rain_1h	float64	No	No	No	0.5	Yes/No	Rain volume for the last hour, mm
18	rain_3h	float64	No	No	No	NaN	Yes/No	Null
19	snow_1h	float64	No	No	No	0.71	Yes/No	Snow volume for the last hour, mm
20	snow_3h	float64	No	No	No	NaN	Yes/No	Null
21	clouds_all	VARCHAR(255)	No	No	No	75	Yes/No	Check - Cloud Types? Cloud Percent? Unclear
22	weather_id	VARCHAR(10)	No	No	Yes	803	Yes/No	Weather Identifier
23	weather_main	VARCHAR(255)	No	No	No	Clouds	Yes/No	Main Weather
24	weather_description	VARCHAR(500)	No	No	No	broken clouds	Yes/No	Main Weather Description
25	weather_icon	VARCHAR(2000)	No	No	No	04d	Yes/No	Weather Icon

Documentation Summary:

API documentation available here: Link

Quality summary:

The dataset is defined through unique values of datetime at 1 hour intervals. Most of the data seem to be quite normal and acceptable.

The features rain_1h, rain_3h, snow_1h, snow_3h have missing values in them. All other descriptors are quite clean with regards to missing values and data quality, there are no extreme outliers or any unusual values.

Potential Issues:

The datetime values seem to include a summer time transition, so may create an issue when combining with other datasets using the datetime.

The missing values in the above mentioned features may cause issues if we were to use them in our model training.

Cleaning Method:

The way we cleaned the above mentioned features, filling in the missing values, was by looking at the other features and how they relate. What we found was that the rainfall and snow fall features with missing values had a descriptor category of sunny or cloudy. So at those times there were no rain. Thus we decided to fill in the values with 0 rainfall, as it means the missing values represent the times where the rainfall was negligible or was not raining.

Weather Summary:
https://openweathermap.org/weather-conditions

Current and Forecast Data

This page details integration details (keys) of the application to OpenWeatherMapAPI to pull weather data.
The Table Below Lists the Integration Details and Endpoints.

The info returned is a json

For all integrations the appid parameter must be added to connect. Eg.

`Open Weather Stations: https://api.openweathermap.org/data/2.5/weather?lat={}&lon={}&appid={API KEY GOES HERE}`

	Description	Service Endpoint	Parameters	Keys
1	Weather Info, given Latitude and Longitude	`http://api.openweathermap.org/data/2.5/weather?lat={}&lon={}`	{position_latitude} {position_longitude}	`fa4a1ef5fe110a5b66dbe8f58890b6f1`
2	Forecast Info, Given Latitude and Longitude	`http://api.openweathermap.org/data/2.5/forecast?lat={}&lon={}&appid={}`	{position_latitude} {position_longitude}	`fa4a1ef5fe110a5b66dbe8f58890b6f1`

Weather Mapping

IMPORTANT NOTE!!! MULTIPLE WEATHER TYPES POSSIBLE! CHOSE weather[0] TO GET THE PRIMARY WEATHER TYPE NB NB NB!!!!

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	position_long	REAL	No	No	No	-6.2656	No	Longitude
2	position_lat	REAL	No	No	No	53.3581	No	Latitude
3	weather_id	INTEGER	No	No	No	803	No	Weather ID Type
4	main	VARCHAR(256)	No	No	No	Clouds	No	Short Description
5	description	VARCHAR(500)	No	No	No	broken clouds	No	Weather Description
6	icon	VARCHAR(20)	No	No	No	04d	No	Weather Icon
7	icon_url	VARCHAR(500)	No	No	No	http://openweathermap.org/img/wn/04d@2x.png	No	Icon URL
8	base	varchar(256)	No	No	No	stations	No	Station
9	temp	REAL	No	No	No	286.21	No	Temp
10	feels_like	REAL	No	No	No	277.02	No	Feels Like
11	temp_min	REAL	No	No	No	284.82	No	Min Temp
12	temp_max	REAL	No	No	No	287.15	No	Max Temp
13	pressure	INT	No	No	No	1001	No	Air Pressure
14	humidity	INT	No	No	No	77	No	Humidity
15	visibility	INT	No	No	No	10000	No	Visibility
16	wind_speed	REAL	No	No	No	12.86	No	Wind Speed
17	wind_degree	INT	No	No	No	200	No	Wind Direction
18	clouds_all	INT	No	No	No	75	No	Clouds???
19	datetime	BIGINT	No	No	No	1614079641	No	Datetime
20	sys_type	INT	No	No	No	1	No	?
21	sys_id	INT	No	No	No	1565	No	System ID?
22	sys_country	VARCHAR(10)	No	No	No	IE	No	Country
23	sys_sunrise	BIGINT	No	No	No	1614065172	No	Sunrise Time
24	sys_sunset	BIGINT	No	No	No	1614102658	No	Sunset Time
25	timezone	INT	No	No	No	0	No	Timezone ID
26	id	BIGINT	No	No	No	6691027	No	???
27	name	VARCHAR(256)	No	No	No	Drumcondra	No	Location Name
28	cod	INT	No	No	No	200	No	???

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	position_long	REAL	No	No	No	-6.2656	No	Longitude
2	position_lat	REAL	No	No	No	53.3581	No	Latitude
3	weather_id	INTEGER	No	No	No	803	No	Weather ID Type
4	main	VARCHAR(256)	No	No	No	Clouds	No	Short Description
5	description	VARCHAR(500)	No	No	No	broken clouds	No	Weather Description
6	icon	VARCHAR(20)	No	No	No	04d	No	Weather Icon
7	icon_url	VARCHAR(500)	No	No	No	http://openweathermap.org/img/wn/04d@2x.png	No	Icon URL
8	base	varchar(256)	No	No	No	stations	No	Station
9	temp	REAL	No	No	No	286.21	No	Temp
10	feels_like	REAL	No	No	No	277.02	No	Feels Like
11	temp_min	REAL	No	No	No	284.82	No	Min Temp
12	temp_max	REAL	No	No	No	287.15	No	Max Temp
13	pressure	INT	No	No	No	1001	No	Air Pressure
14	humidity	INT	No	No	No	77	No	Humidity
15	visibility	INT	No	No	No	10000	No	Visibility
16	wind_speed	REAL	No	No	No	12.86	No	Wind Speed
17	wind_degree	INT	No	No	No	200	No	Wind Direction
18	clouds_all	INT	No	No	No	75	No	Clouds???
19	`forecast_time_dt`	DATETIME	No	No	No	2021-01-01	No	Datetime
20	`forecast_time_ts`	BIGINT	No	No	No	1614102658	No	Timestamp of DT

Documentation Summary:

OpenWeatherMap API documentation available here: https://openweathermap.org/current

Scraper Code

The following code is a sample to scrape these:

```
database_schema={
                    '01_station':{
                              'address':'VARCHAR(256)'
                              ,'banking':'INTEGER'
                              ,'bike_stands':'INTEGER'
                              ,'bonus':'INTEGER'
                              ,'contract_name':'VARCHAR(256)'
                              ,'name':'VARCHAR(256)'
                              ,'number':'INTEGER'
                              ,'position_lat':'REAL'
                              ,'position_long':'REAL'
                              ,'created_date': 'BIGINT'
                                }
                   ,'01_availability':{
                                'number':'INTEGER'
                                ,'available_bikes':'INTEGER'
                                ,'available_bike_stands':'INTEGER'
                                ,'last_update':'BIGINT'
```

```
                    ,'created_date':'BIGINT'
                    }

                ,'01_weather':{
                    'number': 'INT'
                    ,'position_long':'REAL'
                    ,'position_lat':'REAL'
                    ,'weather_id':'INTEGER'
                    ,'main':'VARCHAR(256)'
                    ,'description':'VARCHAR(500)'
                    ,'icon':'VARCHAR(20)'
                    ,'icon_url':'VARCHAR(500)'
                    ,'base':'varchar(256)'
                    ,'temp':'REAL'
                    ,'feels_like':'REAL'
                    ,'temp_min':'REAL'
                    ,'temp_max':'REAL'
                    ,'pressure':'INT'
                    ,'humidity':'INT'
                    ,'visibility':'INT'
                    ,'wind_speed':'REAL'
                    ,'wind_degree':'INT'
                    ,'clouds_all':'INT'
                    ,'datetime':'BIGINT'
                    ,'sys_id':'INT'
                    ,'sys_country':'VARCHAR(10)'
                    ,'sys_sunrise':'BIGINT'
                    ,'sys_sunset':'BIGINT'
                    ,'sys_type':'INT'
                    ,'timezone':'INT'
                    ,'id':'BIGINT'
                    ,'name':'VARCHAR(256)'
                    ,'cod':'INT'
                    ,'created_date':'BIGINT'
                    }

                ,'01_forecast':{
                    'number': 'INT'
                    ,'position_long':'REAL'
                    ,'position_lat':'REAL'
                    ,'weather_id':'INTEGER'
                    ,'main':'VARCHAR(256)'
                    ,'description':'VARCHAR(500)'
                    ,'icon':'VARCHAR(20)'
                    ,'icon_url':'VARCHAR(500)'
                    ,'temp':'REAL'
                    ,'feels_like':'REAL'
                    ,'temp_min':'REAL'
                    ,'temp_max':'REAL'
                    ,'pressure':'INT'
```

```
                                    ,'humidity':'INT'
                                    ,'visibility':'INT'
                                    ,'wind_speed':'REAL'
                                    ,'wind_degree':'INT'
                                    ,'clouds_all':'INT'
                                    ,'forecast_time_ts':'BIGINT'
                                    ,'forecast_time_dt':'DATETIME'
                                    ,'created_date':'BIGINT'
                                    }

                        ,'02_station_avail_weather_train':{
                                    'number':'INTEGER'
                                    , 'weather_type_id':'INTEGER'
                                    , 'hour':'INTEGER'
                                    , 'dayofweek':'INTEGER'
                                    , 'dayofmonth':'INTEGER'
                                    , 'bool_weekend':'BOOLEAN'
                                    , 'bool_dayoff':'BOOLEAN'
                                    , 'bool_workhour':'BOOLEAN'
                                    , 'bool_commutehour':'BOOLEAN'
                                    , 'bool_night':'BOOLEAN'
                                    , 'available_bikes':'REAL' #average
over everything
                                    , 'weather_temp_feels_like':'REAL'
#average over everything
                                    , 'weather_temp':'REAL' #average over
everything
                                   ,  'weather_humidity':'REAL' #average
over everything
                                    , 'weather_air_pressure':'REAL'
#average over everything
                                    , 'created_date':'BIGINT'
                                    ,'minute':'INTEGER'
                                    ,'month':'INTEGER'
                                    ,'year':'INTEGER'
                                    }

,                       '02_station_avail_weather_test':{
                                    'number':'INTEGER'
                                    , 'weather_type_id':'INTEGER'
                                    , 'hour':'INTEGER'
                                    , 'dayofweek':'INTEGER'
                                    , 'dayofmonth':'INTEGER'
                                    , 'bool_weekend':'BOOLEAN'
                                    , 'bool_dayoff':'BOOLEAN'
                                    , 'bool_workhour':'BOOLEAN'
                                    , 'bool_commutehour':'BOOLEAN'
                                    , 'bool_night':'BOOLEAN'
                                    , 'available_bikes':'REAL' #average
over everything
```

```
                                , 'weather_temp_feels_like':'REAL'
#average over everything
                                , 'weather_temp':'REAL' #average over
everything
                                , 'weather_humidity':'REAL' #average
over everything
                                , 'weather_air_pressure':'REAL'
#average over everything
                                , 'created_date':'BIGINT'
                                ,'minute':'INTEGER'
                                ,'month':'INTEGER'
                                ,'year':'INTEGER'
                                }

    ,                   '03_user_model_entry':{
                                'prediction_id':'INTEGER NOT NULL
AUTO_INCREMENT'
                                ,'number':'INTEGER'
                                , 'weather_type_id':'INTEGER'
                                , 'hour':'INTEGER'
                                , 'dayofweek':'INTEGER'
                                , 'dayofmonth':'INTEGER'
                                , 'bool_weekend':'BOOLEAN'
                                , 'bool_dayoff':'BOOLEAN'
                                , 'bool_workhour':'BOOLEAN'
                                , 'bool_commutehour':'BOOLEAN'
                                , 'bool_night':'BOOLEAN'
                                , 'available_bikes':'REAL' #average
over everything
                                , 'weather_temp_feels_like':'REAL'
#average over everything
                                , 'weather_temp':'REAL' #average over
everything
                                , 'weather_humidity':'REAL' #average
over everything
                                , 'weather_air_pressure':'REAL'
#average over everything
                                , 'user_date':'BIGINT'
                                , 'created_date':'BIGINT'
                                , 'predicted_date':'BIGINT'
                                , 'model_type':'VARCHAR(100)'
                                , 'bool_correct_prediction':'BOOLEAN'
                                ,'minute':'INTEGER'
                                ,'month':'INTEGER'
                                ,'year':'INTEGER'
                                }
                }


services_dictionary={
```

```
                'Dublin Bikes':
                    {
                       'Service Provider':'JCDecaux'
                       ,'API Reason':'Dublin Bikes'
                       ,'Security':'secret'

                       ,'Endpoint':{
                           'Station':'https://api.jcdecaux.com/vls/v1
/stations'
                           ,'Contract':'https://api.jcdecaux.com/vls/v1
/contracts'
                           ,'Park of Contract':'https://api.jcdecaux.com
/parking/v1/contracts/{}/parks'
                           ,'Park Info':'https://api.jcdecaux.com
/parking/v1/contracts/{}/parks/{}'

                                        }
                       ,'API Key':''
                    },
                'OpenWeatherAPI':
                    {
                        'Service Provider':'OpenWeatherMap'
                        ,'API Reason':'Weather Data'
                        ,'Security':'secret'

                        ,'Endpoint':{
                                'weather_at_coord':'http://api.
openweathermap.org/data/2.5/weather' #?lat={}&lon={}&appid={}
                                    }
                        ,'API Key':''
                    },

                'OpenWeatherMapForecast':
                    {
                        'Service Provider':'OpenWeatherMap'
                        ,'API Reason':'Weather Data'
                        ,'Security':'secret'

                        ,'Endpoint':{
                                'weather_at_coord':'http://api.
openweathermap.org/data/2.5/forecast' #?lat={}&lon={}&appid={}
                                    }
                        ,'API Key':''
                    },
                }
```

```
ar_database_dictionary={
                        'username':''
```

```
                        ,'password':''
                        ,'database':''
                        ,'endpoint':''
                        ,'port':''
                    }
```

```
##------------------Extractor---------------##
##
##User:     Aryan
##DC:       2021-02-15
##DLM:      2021-02-15
##MC:       COMP30830
##SD:       Save Dublin Bike Info
##
##------------------Extractor---------------##



####--------------------------------------
#00.Import Modules
####--------------------------------------

######---------BEGIN
#      ML
######--------END

#import nltk as nl
#import sklearn as sk
#import matplotlib as mp
#import xgboost as xg
#import pymc3 as pymc
#import sympy as sym



######---------BEGIN
#      SQL
######--------END


import requests as rq
import sqlalchemy as sqla
#import pyodbc
#import cx_oracle as cx


######---------BEGIN
#     GENERAL
```

```
######--------END

import pandas as pd
import datetime as dt
import numpy as np
#import pyodbc
import sys
import os
import json
import time
import socket
import traceback as tb
import platform
from psutil import virtual_memory

#BeExplicit <- Works for Me Locally but needs to have installed module.
Replacing with explicit
#from station_info.data_dictionary import services_dictionary
#from station_info.data_dictionary import database_dictionary
#from station_info.data_dictionary import database_schema


database_dictionary={
                        'username':'adamryan'
                        ,'password':'adam.ryan1'
                        ,'database':'dbbikes'
                        ,'endpoint':'dbbikes.cmbuuvrlonfv.us-east-1.rds.
amazonaws.com'
                        ,'port':'3306'
                    }

database_schema={
                    '01_station':{
                              'address':'VARCHAR(256)'
                              ,'banking':'INTEGER'
                              ,'bike_stands':'INTEGER'
                              ,'bonus':'INTEGER'
                              ,'contract_name':'VARCHAR(256)'
                              ,'name':'VARCHAR(256)'
                              ,'number':'INTEGER'
                              ,'position_lat':'REAL'
                              ,'position_long':'REAL'
                              ,'created_date': 'BIGINT'
                                }
                    ,'01_availability':{
                                'number':'INTEGER'
                                ,'available_bikes':'INTEGER'
                                ,'available_bike_stands':'INTEGER'
                                ,'last_update':'BIGINT'
                                ,'created_date':'BIGINT'
```

```
                                }

                    ,'01_weather':{
                                'number': 'INT'
                                ,'position_long':'REAL'
                                ,'position_lat':'REAL'
                                ,'weather_id':'INTEGER'
                                ,'main':'VARCHAR(256)'
                                ,'description':'VARCHAR(500)'
                                ,'icon':'VARCHAR(20)'
                                ,'icon_url':'VARCHAR(500)'
                                ,'base':'varchar(256)'
                                ,'temp':'REAL'
                                ,'feels_like':'REAL'
                                ,'temp_min':'REAL'
                                ,'temp_max':'REAL'
                                ,'pressure':'INT'
                                ,'humidity':'INT'
                                ,'visibility':'INT'
                                ,'wind_speed':'REAL'
                                ,'wind_degree':'INT'
                                ,'clouds_all':'INT'
                                ,'datetime':'BIGINT'
                                ,'sys_type':'INT'
                                ,'sys_id':'INT'
                                ,'sys_country':'VARCHAR(10)'
                                ,'sys_sunrise':'BIGINT'
                                ,'sys_sunset':'BIGINT'
                                ,'sys_type':'INT'
                                ,'timezone':'INT'
                                ,'id':'BIGINT'
                                ,'name':'VARCHAR(256)'
                                ,'cod':'INT'
                                ,'created_date':'BIGINT'
                                }
                }

services_dictionary={
                'Dublin Bikes':
                    {
                       'Service Provider':'JCDecaux'
                       ,'API Reason':'Dublin Bikes'
                       ,'Security':'secret'

                       ,'Endpoint':{
                           'Station':'https://api.jcdecaux.com/vls/v1
/stations'
                           ,'Contract':'https://api.jcdecaux.com/vls/v1
/contracts'
                           ,'Park of Contract':'https://api.jcdecaux.com
```

```
/parking/v1/contracts/{}/parks'
                          ,'Park Info':'https://api.jcdecaux.com
/parking/v1/contracts/{}/parks/{}'

                                      }
                        ,'API
Key':'fce18e526613c8451be601a23ce591ed36b2b209'
                    },
                'OpenWeatherAPI':
                    {
                        'Service Provider':'OpenWeatherMap'
                        ,'API Reason':'Weather Data'
                        ,'Security':'secret'

                        ,'Endpoint':{
                                'weather_at_coord':'http://api.
openweathermap.org/data/2.5/weather' #?lat={}&lon={}&appid={}
                                    }
                        ,'API Key':'fa4a1ef5fe110a5b66dbe8f58890b6f1'
                    },
               }

######---------BEGIN
#     DATA VIS
######--------END

#import seaborn as sb
#import matplotlib as mp
#from bokeh import *
#from dash import *


####--------------------------------------
#01. Dictionary
####--------------------------------------


####--------------------------------------
#02. Pull from Dublin Bike API
####--------------------------------------

def request_dublinbike_data():
    """Request Dublin Bike Data.

    Inpu: Key
    Output: JsonTEXT
    """
    print("Inside request_dublinbike_data()\n\n")
    dbikes_endpoint=services_dictionary['Dublin Bikes']['Endpoint']
['Station']
```

```
    dbikes_key=services_dictionary['Dublin Bikes']['API Key']
    dbikes_contract='dublin'

    #Error handling
    error_dictionary={
                         0:'Success'
                        ,1:'There was an error in the request'
                        }

    #Assume no error by default
    error_code=0

    #Attempt the Request
    try:
        request_response=rq.get(dbikes_endpoint,params={"apiKey":
dbikes_key,
                                                        "contract":
dbikes_contract})

        json_text=request_response.json()

    #Failed for some reason
    except:
        error_code=1
        error_message=error_dictionary[error_code]
        json_text=''
        json_response=''

        print(error_message)

    return [error_code,json_text]


####-------------------------------------
#03. Write Some Text to File if Needed
####-------------------------------------

def write_to_file(filepath,filename,text):
    """Write text to a file of your name

    filepath=Path of File e.g C://user//test
    filename=Name of File. No Extension
    text=text to write

    Output: [error,fp]
    """

    #Static
    datetime_now=dt.datetime.now()
    timestamp_format=dt.datetime.timestamp(datetime_now)
```

```
    fn='{}_{}'.format(filename,timestamp_format)
    fp='{}/{}.txt'.format(filepath,fn)


    #Error handling
    error_dictionary={
                        0:'Success'
                        ,1:'The filepath does not exist. You entered:\n
{}'
                        ,2:'There was an error writing to text.'
                        }

    #Assume no error by default
    error_code=0


    #Check that the filepath exists
    if os.path.exists(filepath) or (filepath=='' or filepath=='.'):


        #open the file
        with open(fp, 'w') as f:
            print('Writing to: \n{}\n\n'.format(fp))

            #Try Write
            try:
                print('File Written')
                f.write(text)


            #Write Error
            except:
                error_code=2
                error_message=error_dictionary[error_code].format
(filepath)
                print(error_message)

    #File path does not exist
    else:
        error_code=1
        error_message=error_dictionary[error_code].format(filepath)
        print(error_message)

    return [error_code,fp]




####-------------------------------------
#04. Flatten Dublin Bikes and Write it to File. More specific version
```

```
of 3.
####-----------------------------------


def write_flatten_dublinbikes_to_text(filepath,json):
    """Write text to a file of your name

    filepath=Path of File e.g C://user//test
    filename=Name of File. No Extension
    text=text to write

    output: [error,fp]
    """

    #Static
    datetime_now=dt.datetime.now()
    filename='DublinBike'
    timestamp_format=dt.datetime.timestamp(datetime_now)
    fn='{}_{}'.format(filename,timestamp_format)
    fp='{}/{}.txt'.format(filepath,fn)
    json_atrib_ordered=['number'
                        , 'contract_name'
                        , 'name'
                        , 'address'
                        , 'banking'
                        , 'bonus'
                        , 'bike_stands'
                        , 'available_bike_stands'
                        , 'available_bikes'
                        , 'status'
                        , 'position.lat'
                        , 'position.lng'
                        , 'last_update']

    json_atrib_rename_dict={
                          'position.lat':'position_latitude'
                        , 'position.lng':'position_longitude'
    }


    #Error handling
    error_dictionary={
                        0:'Success'
                        ,1:'The filepath does not exist. You entered:\n
{}'
                        ,2:'There was an error reading dataframe.'
                        ,3:'Json Payload Doesnt Match Columns'
                        ,4:'Error Saving to File'
                        }
```

```
    #Assume no error by default
    error_code=0


    #Check that the filepath exists
    if os.path.exists(filepath) or (filepath=='' or filepath=='.'):

        #Read Dataframe
        try:
            df=pd.json_normalize(json)

            if filepath=='' or filepath=='.':
                fp='{}/{}.txt'.format(os.getcwd(),fn)

            #Columns Match
            if set(df.columns)==set(json_atrib_ordered):
                print('Columns match')

                #Try Write CSV
                try:
                    df=df[json_atrib_ordered]
                    df=df.rename
(columns=json_atrib_rename_dict)
                    df.to_csv(fp, sep='|', index=False)
                    print('Written')

                except:
                    error_code=4
                    error_message=error_dictionary[error_code].format
(filepath)
                    print(error_message)
                    print('Error in Ordering and Saving')

            #Error In Columns
            else:
                error_code=3
                error_message=error_dictionary[error_code].format
(filepath)
                print(error_message)

        #Error Reading Dataframe
        except:
            error_code=2
            error_message=error_dictionary[error_code].format(filepath)
            print(error_message)

    #File path does not exist
    else:
        error_code=1
        error_message=error_dictionary[error_code].format(filepath)
```

```
        print(error_message)

    return [error_code,fp]




####-------------------------------------
#05. Connect to a Database Engine
####-------------------------------------

def connect_db_engine(host,user,password,port,db):
    """Connect to the db engine

    host: host
    user: user
    password: pw
    port: port
    db: Name of DB


    """

    print("Inside connect_db_engine()\n\n")

    error_code=0
    engine=''

    error_dictionary={0:'No Error'
                      ,1:'One of the parammeters is wrong'}

    try:
        connect_statement='mysql+mysqldb://{}:{}@{}:{}/{}'.format(user,
password,host,port,db)
        print(connect_statement)
        engine=sqla.create_engine(connect_statement,echo=True)

    except Exception as e:
        error_code=1
        error_message=error_dictionary[error_code]
        print(error_message)
        print("The Exception is:\n\n{}\n\n".format(e))

    return [error_code,engine]


####-------------------------------------
#06. Setup the Database Schema and all related functions (e.g. foreign
keys, primary keys)
####-------------------------------------
```

```
def setup_database(host,user,password,port,db):
    """Set up the database if it does not already exist.

    Input is the database parameters and database_dictionary
    """

    print("Inside setup_database()\n\n")

    engine_l=connect_db_engine(host,user,password,port,db)
    engine=engine_l[1]

    create_sql="""
                CREATE DATABASE IF NOT EXISTS
                                            {};
                """.format(db)

    engine.execute(create_sql)



    #Loop through every table
    for table, columns in database_schema.items():
        column_count=0
        column_number=len(columns)
        insert_sql=''
        insert_sql="""CREATE TABLE IF NOT EXISTS {} (\n""".format(table)
        insert_row=''

        #for every column and type add on a statement
        for column_name, column_type in columns.items():

            #Add the statement with , in front
            if column_count>0:
                insert_row+=",{}     {}".format(column_name,column_type)

            #First column
            else:
                insert_row+="{}     {}".format(column_name,column_type)
                column_count+=1


        insert_sql+='{})'.format(insert_row)

        #Start this madness re: creating and inserting the schema
        try:
            engine.execute(insert_sql)

        #Except for something; probably the schema
```

```
        except Exception as exc:
            print(exc)

    print('Database Schema Created, have fun!')
    engine.dispose()

    return




####-------------------------------------
#07. Retrieve the unique station numbers from the station table
####-------------------------------------



def existing_station_numbers(engine):
    """A function to check which station numbers are already in the
database"""
    station_list=[]

    try:
        Select_SQL="""
        SELECT
            number
        FROM
            01_station
        """

        result=engine.execute(Select_SQL)

        rows = result.fetchall()

        for station_number in rows:
            print(station_number)
            station_list+=[station_number[0]]

        result.close()

    except:
        print('Test')
        station_list=[]


    return station_list


####-------------------------------------
#08. Insert values to availability and station table
```

```
####-------------------------------------




####-------------------------------------
#08.1 WEATHER DATA
####-------------------------------------

def request_weather_data(latitude,longitude):
    """Request OpenWeather Data.

    Inpu: Key
    Output: JsonTEXT
    """
    print("Inside request_weather_data()\n\n")

    key=services_dictionary['OpenWeatherAPI']['API Key']
    endpoint=services_dictionary['OpenWeatherAPI']['Endpoint']
['weather_at_coord']


    #Error handling
    error_dictionary={
                        0:'Success'
                       ,1:'There was an error in the request'
                       }

    #Assume no error by default
    error_code=0

    #Attempt the Request
    try:
        request_response=rq.get(endpoint,params={"APPID":key,
                                                 "lat":latitude,
                                                 "lon":longitude
                                                })

        json_text=request_response.json()

    #Failed for some reason
    except Exception as e:
        error_code=1
        error_message=error_dictionary[error_code]
        json_text=''
        json_response=''

        print(error_message)
        print(e)
```

```
    return [error_code,json_text]


def store_weather_data(weather_json,number,avail_datetime_updated,
engine,time_added):
    """Store the Weather Data into the database"""

    print('Inside store_weather_data')
    station_number=number
    avail_dt_update=avail_datetime_updated
    position_long=weather_json['coord']['lon']
    position_lat=weather_json['coord']['lat']

    weather_id=weather_json['weather'][0]['id']
    main=weather_json['weather'][0]['main']
    description=weather_json['weather'][0]['description']
    icon=weather_json['weather'][0]['icon']
    icon_url='http://openweathermap.org/img/wn/{}@2x.png'.format(icon)

    base=weather_json['base']
    temp=weather_json['main']['temp']
    feels_like=weather_json['main']['feels_like']
    temp_min=weather_json['main']['temp_min']
    temp_max=weather_json['main']['temp_max']
    pressure=weather_json['main']['pressure']
    humidity=weather_json['main']['humidity']
    visibility=weather_json['visibility']

    wind_speed=weather_json['wind']['speed']
    wind_degree=weather_json['wind']['deg']

    clouds_all=weather_json['clouds']['all']

    datetime=weather_json['dt']
    sys_type=weather_json['sys']['type']
    sys_country=weather_json['sys']['country']
    sys_id=weather_json['sys']['id']
    sys_sunrise=weather_json['sys']['sunrise']
    sys_sunset=weather_json['sys']['sunset']

    timezone=weather_json['timezone']
    id_var=weather_json['id']
    name=weather_json['name']
    cod=weather_json['cod']

    created_date=time_added

    weather_insert='''INSERT INTO 01_weather
```

```
            (    number
                ,position_long
                ,position_lat

                ,weather_id
                ,main
                ,description
                ,icon
                ,icon_url

                ,base
                ,temp
                ,feels_like
                ,temp_min
                ,temp_max
                ,pressure
                ,humidity
                ,visibility

                ,wind_speed
                ,wind_degree

                ,clouds_all

                ,datetime
                ,sys_type
                ,sys_id
                ,sys_country

                ,sys_sunrise
                ,sys_sunset

                ,timezone
                ,id
                ,name
                ,cod
                ,created_date)

        VALUES
            (%s
              ,%s
              ,%s

              ,%s
              ,%s
              ,%s
              ,%s
              ,%s
```

```
                                ,%s
                                ,%s
                                ,%s
                                ,%s
                                ,%s
                                ,%s
                                ,%s
                                ,%s
                                ,%s

                                ,%s

                                ,%s
                                ,%s
                                ,%s
                                ,%s



                                ,%s
                                ,%s

                                ,%s
                                ,%s
                                ,%s
                                ,%s
                                ,%s)
                                  '''

    weather_values=(station_number
                    ,position_long
                    ,position_lat
                    ,weather_id
                    ,main
                    ,description
                    ,icon
                    ,icon_url
                    ,base
                    ,temp
                    ,feels_like
                    ,temp_min
                    ,temp_max
                    ,pressure
                    ,humidity
                    ,visibility
                    ,wind_speed
                    ,wind_degree
                    ,clouds_all
```

```
                    ,datetime
                    ,sys_type
                    ,sys_id
                    ,sys_country
                    ,sys_sunrise
                    ,sys_sunset
                    ,timezone
                    ,id_var
                    ,name
                    ,cod
                    ,created_date)

    engine.execute(weather_insert,weather_values)

    return

def insert_station_static_values(json_data,existing_station_numbers,
engine):
    """Insert the static values into the database"""

    print(f"Inside insert_station_static_values()\n\n")

    station_list=existing_station_numbers
    datetime_now=dt.datetime.now()
    created_date=dt.datetime.timestamp(datetime_now)

    for entry in json_data:
        address=entry['address']
        name=entry['name']
        contract_name=entry['contract_name']
        banking=int(entry['banking'])
        bonus=int(entry['bonus'])
        bike_stands=entry['bike_stands']
        available_bike_stands=entry['available_bike_stands']
        available_bikes=entry['available_bikes']
        status=entry['status']
        number=entry['number']
        position_lat=entry['position']['lat']
        position_lng=entry['position']['lng']
        last_update=entry['last_update']/1000



        ###RENAME STATION
        station_insert='''INSERT INTO 01_station
                                (address
                                  ,banking
                                  ,bike_stands
                                  ,bonus
                                  ,contract_name
```

```
                                    ,name
                                    ,number
                                    ,position_lat
                                    ,position_long
                                    ,created_date)

                                VALUES
                                    (%s
                                      ,%s
                                      ,%s
                                      ,%s
                                      ,%s
                                      ,%s
                                      ,%s
                                      ,%s
                                      ,%s
                                      ,%s)
                                        '''



        variable_insert='''INSERT INTO 01_availability
                                    (number
                                    ,available_bikes
                                    ,available_bike_stands
                                    ,last_update
                                    ,created_date)

                                VALUES
                                      (%s
                                      ,%s
                                      ,%s
                                      ,%s
                                      ,%s)
                                            '''

        #Station Data already available
        if number not in station_list:
            station_values=(address,banking,bike_stands ,bonus,
contract_name,name,number,position_lat,position_lng,created_date)
            engine.execute(station_insert,station_values)

        bike_values=(number,available_bikes,available_bike_stands,
last_update,created_date)
        engine.execute(variable_insert,bike_values)

        try:
            weather_info_json_lat_long_list=request_weather_data
(latitude=str(position_lat),longitude=str(position_lng))
```

```
weather_info_json_lat_long_json=weather_info_json_lat_long_list[1]
            store_weather_data(weather_info_json_lat_long_json,number,
last_update,engine,created_date)
        except Exception as e:
            print(e)



    return




####-------------------------------------
#10. Pull Station Data, Post to DB
####-------------------------------------

#Wrapper function to pull the bike data and store it into a database
def pull_station_data():
    """Pull Weather Data and save it into the database."""

    print("Inside pull_station_data()\n\n")

    try:
        request_list=request_dublinbike_data()
        json_data=request_list[1]

        #No error occured
        if request_list[0]==0:
            myhost=database_dictionary['endpoint']
            myuser=database_dictionary['username']
            mypassword=database_dictionary['password']
            myport=database_dictionary['port']
            mydb=database_dictionary['database']

            mysql_list=connect_db_engine(myhost,myuser,mypassword,
myport,mydb)
            mysql_engine=mysql_list[1]
            existing_station_numbers_list=existing_station_numbers
(mysql_engine)
            insert_station_static_values(json_data,
existing_station_numbers_list,mysql_engine)
            mysql_engine.dispose()

        else:
            print('There had been an error')

    except Exception as e:
        print(e)
```

```
    return



####-------------------------------------
#11. get machine info
####-------------------------------------

def machine_info():
    """Gets some info on your machine"""

    machine_name=platform.machine
    os_name=platform.os
    os_version=platform.version
    host_name=socket.gethostname()
    ip_address=socket.gethostbyname(host_name)
    #total_memory=virtual_memory.total()

    print_statement="""
        Your benchmarketing stats are as follows:\n\n
        machine_name = {}
        os_name= {}
        os_version= {}
        host_name= {}
        ip_address= {}
        total_memory= {}\n\n"""

    print(print_statement.format(machine_name
                            ,os_name
                            ,os_version
                            ,host_name
                            ,ip_address
                            ,'TBD'))

    return

####-------------------------------------
#12. Run Main
####-------------------------------------


def main():
    """Main Function"""

    print("Inside Main\n\n")

    machine_info()

    while True:
```

```
        #Pull it every five miutes
        try:
            print("------------------------------\n\n\n")
            print("------------------------------\n\n\n")
            print("------------------------------\n\n\n")
            print('''Starting: The time now is: {}'''.format(dt.
datetime.now()))
            pull_station_data()
            time.sleep(5*60)
            print('\n\n\n------------------------------')
            print('\n\n\n------------------------------')
            print('\n\n\n------------------------------')
            print('\n\n\n------------------------------')

        #Error so figure out what it is.
        except Exception as e:
            print(e)

    return


###RUN MAIN!!!

if __name__ == '__main__':
    main()
```

```
##------------------Extractor---------------##
##
##User:     Aryan
##DC:       2021-02-15
##DLM:      2021-02-15
##MC:       COMP30830
##SD:       Save Dublin Bike Info
##
##------------------Extractor---------------##



####--------------------------------------
#00.Import Modules
####--------------------------------------

######---------BEGIN
#      ML
######--------END

#import nltk as nl
#import sklearn as sk
```

```
#import matplotlib as mp
#import xgboost as xg
#import pymc3 as pymc
#import sympy as sym



######---------BEGIN
#      SQL
######--------END


import requests as rq
import sqlalchemy as sqla
#import pyodbc
#import cx_oracle as cx


######---------BEGIN
#     GENERAL
######--------END

import pandas as pd
import datetime as dt
import numpy as np
#import pyodbc
import sys
import os
import json
import time
import socket
import traceback as tb
import platform
from psutil import virtual_memory

#BeExplicit <- Works for Me Locally but needs to have installed module.
Replacing with explicit
#from station_info.data_dictionary import services_dictionary
#from station_info.data_dictionary import database_dictionary
#from station_info.data_dictionary import database_schema


database_dictionary={
                        'username':'adamryan'
                        ,'password':'adam.ryan1'
                        ,'database':'dbbikes'
                        ,'endpoint':'dbbikes.cmbuuvrlonfv.us-east-1.rds.
amazonaws.com'
                        ,'port':'3306'
                    }
```

```
database_schema={
                    '01_station':{
                              'address':'VARCHAR(256)'
                              ,'banking':'INTEGER'
                              ,'bike_stands':'INTEGER'
                              ,'bonus':'INTEGER'
                              ,'contract_name':'VARCHAR(256)'
                              ,'name':'VARCHAR(256)'
                              ,'number':'INTEGER'
                              ,'position_lat':'REAL'
                              ,'position_long':'REAL'
                              ,'created_date': 'BIGINT'
                                }
                    ,'01_availability':{
                                'number':'INTEGER'
                                ,'available_bikes':'INTEGER'
                                ,'available_bike_stands':'INTEGER'
                                ,'last_update':'BIGINT'
                                ,'created_date':'BIGINT'
                                }

                    ,'01_weather':{
                                'number': 'INT'
                                ,'position_long':'REAL'
                                ,'position_lat':'REAL'
                                ,'weather_id':'INTEGER'
                                ,'main':'VARCHAR(256)'
                                ,'description':'VARCHAR(500)'
                                ,'icon':'VARCHAR(20)'
                                ,'icon_url':'VARCHAR(500)'
                                ,'base':'varchar(256)'
                                ,'temp':'REAL'
                                ,'feels_like':'REAL'
                                ,'temp_min':'REAL'
                                ,'temp_max':'REAL'
                                ,'pressure':'INT'
                                ,'humidity':'INT'
                                ,'visibility':'INT'
                                ,'wind_speed':'REAL'
                                ,'wind_degree':'INT'
                                ,'clouds_all':'INT'
                                ,'datetime':'BIGINT'
                                ,'sys_id':'INT'
                                ,'sys_country':'VARCHAR(10)'
                                ,'sys_sunrise':'BIGINT'
                                ,'sys_sunset':'BIGINT'
                                ,'sys_type':'INT'
                                ,'timezone':'INT'
                                ,'id':'BIGINT'
```

```
                                ,'name':'VARCHAR(256)'
                                ,'cod':'INT'
                                ,'created_date':'BIGINT'
                                }

                    ,'01_forecast':{
        # Removed base, timezone, avail_update_dt, datetime, id_var,
name, cod and all 'sys' entries when compared to original scraper
        # Added forecast_time_dt and forecast_time_txt to show the
forecast times in readable format
                                'number': 'INT'
                                ,'position_long':'REAL'
                                ,'position_lat':'REAL'
                                ,'weather_id':'INTEGER'
                                ,'main':'VARCHAR(256)'
                                ,'description':'VARCHAR(500)'
                                ,'icon':'VARCHAR(20)'
                                ,'icon_url':'VARCHAR(500)'
                                ,'temp':'REAL'
                                ,'feels_like':'REAL'
                                ,'temp_min':'REAL'
                                ,'temp_max':'REAL'
                                ,'pressure':'INT'
                                ,'humidity':'INT'
                                ,'visibility':'INT'
                                ,'wind_speed':'REAL'
                                ,'wind_degree':'INT'
                                ,'clouds_all':'INT'
                                ,'forecast_time_dt':'BIGINT'
                                ,'forecast_time_txt':'VARCHAR(200)'
                                ,'created_date':'BIGINT'
                                }

                    ,'02_station_avail_weather_train':{
                                'number':'INTEGER'
                                , 'weather_type_id':'INTEGER'
                                , 'hour':'INTEGER'
                                , 'dayofweek':'INTEGER'
                                , 'dayofmonth':'INTEGER'
                                , 'bool_weekend':'BOOLEAN'
                                , 'bool_dayoff':'BOOLEAN'
                                , 'bool_workhour':'BOOLEAN'
                                , 'bool_commutehour':'BOOLEAN'
                                , 'bool_night':'BOOLEAN'
                                , 'available_bikes':'REAL' #average
over everything
                                , 'weather_temp_feels_like':'REAL'
#average over everything
                                , 'weather_temp':'REAL' #average over
everything
```

```
                                ,  'weather_humidity':'REAL' #average
over everything
                                , 'weather_air_pressure':'REAL'
#average over everything
                                , 'created_date':'BIGINT'
                                }

,                    '02_station_avail_weather_test':{
                                'number':'INTEGER'
                                , 'weather_type_id':'INTEGER'
                                , 'hour':'INTEGER'
                                , 'dayofweek':'INTEGER'
                                , 'dayofmonth':'INTEGER'
                                , 'bool_weekend':'BOOLEAN'
                                , 'bool_dayoff':'BOOLEAN'
                                , 'bool_workhour':'BOOLEAN'
                                , 'bool_commutehour':'BOOLEAN'
                                , 'bool_night':'BOOLEAN'
                                , 'available_bikes':'REAL' #average
over everything
                                , 'weather_temp_feels_like':'REAL'
#average over everything
                                , 'weather_temp':'REAL' #average over
everything
                                , 'weather_humidity':'REAL' #average
over everything
                                , 'weather_air_pressure':'REAL'
#average over everything
                                , 'created_date':'BIGINT'
                                }

    ,                    '03_user_model_entry':{
                                'prediction_id':'INTEGER'
                                ,'number':'INTEGER'
                                , 'weather_type_id':'INTEGER'
                                , 'hour':'INTEGER'
                                , 'dayofweek':'INTEGER'
                                , 'dayofmonth':'INTEGER'
                                , 'bool_weekend':'BOOLEAN'
                                , 'bool_dayoff':'BOOLEAN'
                                , 'bool_workhour':'BOOLEAN'
                                , 'bool_commutehour':'BOOLEAN'
                                , 'bool_night':'BOOLEAN'
                                , 'available_bikes':'REAL' #average
over everything
                                , 'weather_temp_feels_like':'REAL'
#average over everything
                                , 'weather_temp':'REAL' #average over
everything
                                , 'weather_humidity':'REAL' #average
```

```
over everything
                                , 'weather_air_pressure':'REAL'
#average over everything
                                , 'user_date':'BIGINT'
                                , 'created_date':'BIGINT'
                                , 'predicted_date':'BIGINT'
                                , 'model_type':'VARCHAR(100)'
                                , 'bool_correct_prediction':'BOOLEAN'
                                }
                }


services_dictionary={
                'Dublin Bikes':
                    {
                       'Service Provider':'JCDecaux'
                       ,'API Reason':'Dublin Bikes'
                       ,'Security':'secret'

                       ,'Endpoint':{
                           'Station':'https://api.jcdecaux.com/vls/v1
/stations'
                           ,'Contract':'https://api.jcdecaux.com/vls/v1
/contracts'
                           ,'Park of Contract':'https://api.jcdecaux.com
/parking/v1/contracts/{}/parks'
                           ,'Park Info':'https://api.jcdecaux.com
/parking/v1/contracts/{}/parks/{}'

                                        }
                        ,'API Key':''
                    },
                'OpenWeatherAPI':
                    {
                        'Service Provider':'OpenWeatherMap'
                        ,'API Reason':'Weather Data'
                        ,'Security':'secret'

                        ,'Endpoint':{
                                'weather_at_coord':'http://api.
openweathermap.org/data/2.5/weather' #?lat={}&lon={}&appid={}
                                    }
                        ,'API Key':''
                    },

                'OpenWeatherMapForecast':
                    {
                        'Service Provider':'OpenWeatherMap'
                        ,'API Reason':'Weather Data'
                        ,'Security':'secret'
```

```
                        ,'Endpoint':{
                                'weather_at_coord':'http://api.
openweathermap.org/data/2.5/forecast' #?lat={}&lon={}&appid={}
                                    }
                        ,'API Key':''
                    },
                }

######---------BEGIN
#     DATA VIS
######--------END

#import seaborn as sb
#import matplotlib as mp
#from bokeh import *
#from dash import *


####--------------------------------------
#01. Conncct to db
####--------------------------------------


def connect_db_engine(host,user,password,port,db):
    """Connect to the db engine

    host: host
    user: user
    password: pw
    port: port
    db: Name of DB


    """

    print("Inside connect_db_engine()\n\n")

    error_code=0
    engine=''

    error_dictionary={0:'No Error'
                     ,1:'One of the parameters is wrong'
                      ,999: 'Uncaught exception'
                     }

    try:
        connect_statement='mysql+mysqlconnector://{}:{}@{}:{}/{}'.format
(user,password,host,port,db)
        print(connect_statement)
```

```
        engine=sqla.create_engine(connect_statement,echo=True)

    except Exception as e:
        error_code=999
        print(e)

    return [error_code,engine]




####-------------------------------------
#02. Setup DB
####-------------------------------------



def setup_database(host,user,password,port,db):
    """Set up the database if it does not already exist.

    Input is the database parameters and database_dictionary
    """

    print("Inside setup_database()\n\n")

    engine_l=connect_db_engine(host,user,password,port,db)
    engine=engine_l[1]

    create_sql="""
                CREATE DATABASE IF NOT EXISTS
                                            {};
                """.format(db)

    engine.execute(create_sql)



    #Loop through every table
    for table, columns in database_schema.items():
        column_count=0
        column_number=len(columns)
        insert_sql=''
        insert_sql="""CREATE TABLE IF NOT EXISTS {} (\n""".format(table)
        insert_row=''

        #for every column and type add on a statement
        for column_name, column_type in columns.items():

            #Add the statement with , in front
            if column_count>0:
```

```
                insert_row+=",{}     {}".format(column_name,column_type)

            #First column
            else:
                insert_row+="{}     {}".format(column_name,column_type)
                column_count+=1


        insert_sql+='{})'.format(insert_row)

        #Start this madness re: creating and inserting the schema
        try:
            engine.execute(insert_sql)

        #Except for something; probably the schema
        except Exception as exc:
            print(exc)

    print('Database Schema Created, have fun!')
    engine.dispose()

    return




####-------------------------------------
#03. Retrieve Station Info
####-------------------------------------


####All Stations
SQL_select_station="""
    SELECT
         stat.{}                                AS
number
        ,stat.{}                                AS
address
        ,stat.{}                                AS
banking
        ,stat.{}                                AS
bike_status
        ,stat.{}                                AS
bike_stands
        ,stat.{}                                AS
contract_name
        ,stat.{}                                AS
name
        ,stat.{}                                AS
position_lat
```

```
        ,stat.{}                                AS
position_long
        ,FROM_UNIXTIME(stat.{})                 AS
created_date
    FROM
        {} stat
    """.format('number'
            ,'address'
            ,'banking'
            ,'bike_stands'
            ,'bonus'
            ,'contract_name'
            ,'name'
            ,'position_lat'
            ,'position_long'
            ,'created_date'
            ,'01_station')

def station_table_df(host,user,password,port,db):
    """Retrieve the station table.

    Return table as dataframe
    """

    print("Inside setup_database()\n\n")

    engine_l=connect_db_engine(host,user,password,port,db)
    engine=engine_l[1]
    df=pd.DataFrame()

    #no error
    try:
        df=pd.read_sql(SQL_select_station,engine)

    except Exception as e:
        print(e)

    engine.dispose()

    return df




####-------------------------------------
#04. Get the weather forecast
####-------------------------------------

def getWeatherForecast(latitude, longitude):
    """Function to return the weather forecast for certain co-
```

```
ordinates"""
    weather_key = services_dictionary['OpenWeatherMapForecast']['API
Key']
    endpoint=services_dictionary['OpenWeatherMapForecast']['Endpoint']
['weather_at_coord']
    r = requests.get(endpoint, params={"APPID": weather_key, "lat":
latitude, "lon": longitude})
    return r.json()

####-------------------------------------
#05. Store the forecast Info
####-------------------------------------


def store_weather_forecast(weather_json,number,time_added,host,user,
password,port,db):
    """
    Store the Weather Data into the database
    Removed base, timezone, avail_update_dt, datetime, id_var, name,
cod and all 'sys' entries when compared to scraper for current weather
    Added forecast_time_dt and forecast_time_txt
    """

    engine_l=connect_db_engine(host,user,password,port,db)
    engine=engine_l[1]

    print('Inside store_weather_forecast')
    station_number=number

    print(weather_json)

    position_long=weather_json['city']['coord']['lon']
    position_lat=weather_json['city']['coord']['lat']

    for forecast_time in weather_json['list']:
        # Loops through every forecast time at three hour intervals for
the next 5 days for the current coordinates and saves it in the database
        forecast_time_dt = forecast_time['dt']
        forecast_time_txt = forecast_time['dt_txt']
        weather_id=forecast_time['weather'][0]['id']
        main=forecast_time['weather'][0]['main']
        description=forecast_time['weather'][0]['description']
        icon=forecast_time['weather'][0]['icon']
        icon_url='http://openweathermap.org/img/wn/{}@2x.png'.format
(icon)

        temp=forecast_time['main']['temp']
        feels_like=forecast_time['main']['feels_like']
        temp_min=forecast_time['main']['temp_min']
        temp_max=forecast_time['main']['temp_max']
```

```
pressure=forecast_time['main']['pressure']
humidity=forecast_time['main']['humidity']
visibility=forecast_time['visibility']

wind_speed=forecast_time['wind']['speed']
wind_degree=forecast_time['wind']['deg']

clouds_all=forecast_time['clouds']['all']

created_date=time_added

weather_insert='''INSERT INTO 01_forecast

                            (    number
                                ,position_long
                                ,position_lat

                                ,weather_id
                                ,main
                                ,description
                                ,icon
                                ,icon_url

                                ,temp
                                ,feels_like
                                ,temp_min
                                ,temp_max
                                ,pressure
                                ,humidity
                                ,visibility

                                ,wind_speed
                                ,wind_degree

                                ,clouds_all

                                ,forecast_time_dt
                                ,forecast_time_txt

                                ,created_date)

                           VALUES
                               (%s
                                 ,%s
                                 ,%s

                                 ,%s
                                 ,%s
                                 ,%s
                                 ,%s
```

```
                                    ,%s

                                    ,%s
                                    ,%s
                                    ,%s
                                    ,%s
                                    ,%s
                                    ,%s


                                    ,%s
                                    ,%s

                                    ,%s

                                    ,%s
                                    ,%s

                                    ,%s
                                    ,%s)
                                      '''

    weather_values=(station_number
                    ,position_long
                    ,position_lat
                    ,weather_id
                    ,main
                    ,description
                    ,icon
                    ,icon_url
                    ,temp
                    ,feels_like
                    ,temp_min
                    ,temp_max
                    ,pressure
                    ,humidity
                    ,visibility
                    ,wind_speed
                    ,wind_degree
                    ,clouds_all
                    ,forecast_time_dt
                    ,forecast_time_txt
                    ,created_date)

    engine.execute(weather_insert,weather_values)

engine.dispose()

return
```

```
####-------------------------------------
#06. Get the forecast info per station
####-------------------------------------

def forecast_per_station(host,user,password,port,db):
    """Getting the forecast data for each station, leve"""

    setup_database(host,user,password,port,db)
    station_data = station_table_df(host,user,password,port,db)

    station_data=json.loads(station_data.to_json(orient='records'))
    print(station_data)

    # Set up the forecast table if it hasn't been done already
    #Get the current date for when this function is called to be able
to group all entries together
    datetime_now = dt.datetime.now()
    created_date = dt.datetime.timestamp(datetime_now)

    # Loop through each bike station to get their coordinates and make
a forecast call on those coordinates
    for station in station_data:
        position_lat = station['position_lat']
        position_lng = station['position_long']
        number = station['number']
        print("***************")
        print(f"Current station is station number {number} ")
        print("***************")
        # Get the forecast for the co-ordinates of the current station
in the loop
        weather_json = getWeatherForecast(latitude=position_lat,
longitude=position_lng)
        # Store the forecast for this station in the database
        store_weather_forecast(weather_json, number, created_date,host,
user,password,port,db)
    print("***************")
    print("Forecasts inserted for all stations!")
    print("***************")
    return



def clear_forecast_table(host,user,password,port,db):
    """A function to empty the forecast table"""

    try:
        #Engine
        engine_l=connect_db_engine(host,user,password,port,db)
```

```
        engine=engine_l[1]

        #
        enable_delete_SQL="""SET SQL_SAFE_UPDATES = 0"""
        delete_forecast_SQL="""Delete from 01_forecast"""
        disable_delete_SQL="""SET SQL_SAFE_UPDATES = 1"""

        engine.execute(enable_delete_SQL)
        engine.execute(delete_forecast_SQL)
        engine.execute(disable_delete_SQL)

        engine.dispose()
    except Exception as e:
        print("Failed to delete: {}".format(e))

####--------------------------------------
#11. get machine info
####--------------------------------------

def machine_info():
    """Gets some info on your machine"""

    machine_name=platform.machine
    os_name=platform.os
    os_version=platform.version
    host_name=socket.gethostname()
    ip_address=socket.gethostbyname(host_name)
    #total_memory=virtual_memory.total()

    print_statement="""
        Your benchmarketing stats are as follows:\n\n
        machine_name = {}
        os_name= {}
        os_version= {}
        host_name= {}
        ip_address= {}
        total_memory= {}\n\n"""

    print(print_statement.format(machine_name
                            ,os_name
                            ,os_version
                            ,host_name
                            ,ip_address
                            ,'TBD'))

    return

####--------------------------------------
#12. Run Main
####--------------------------------------
```

```
def main():
    """Main Function"""

    print("Inside Main\n\n")

    machine_info()

    myhost=database_dictionary['endpoint']
    myuser=database_dictionary['username']
    mypassword=database_dictionary['password']
    myport=database_dictionary['port']
    mydb=database_dictionary['database']

    while True:

        #Pull it every five miutes
        try:
            print("-------------------------------\n\n\n")
            print("-------------------------------\n\n\n")
            print("-------------------------------\n\n\n")
            print('''Starting: The time now is: {}'''.format(dt.
datetime.now()))

            #Delay it until 12am to avoid app functionality issue
            t1 = time.time()
            seconds_in_hour=60*60
            seconds_in_day=seconds_in_hour**24

            #Allow operation between 11pm and 1am when time in app is
likely low.
            if dt.datetime.now().hour==0 or dt.datetime.now().hour==23
or dt.datetime.now().hour==1:
                clear_forecast_table(host=myhost,user=myuser,
password=mypassword,port=myport,db=mydb)
                forecast_per_station(host=myhost,user=myuser,
password=mypassword,port=myport,db=mydb)

            else:
                print("Outside of operation time")


            runtime=time.time()-t1
            time.sleep(seconds_in_day - runtime)
            print('\n\n\n-------------------------------')
            print('\n\n\n-------------------------------')
            print('\n\n\n-------------------------------')
            print('\n\n\n-------------------------------')
```

```
        #Error so figure out what it is.
        except Exception as e:
            print(e)

    return


###RUN MAIN!!!

if __name__ == '__main__':
    main()
```

Other Integration 4

This page details integration details (keys) of the application to DublinBikes to pull bike data.
The Table Below Lists the Integration Details and Endpoints.

The info returned is a json

For all integrations the appid parameter must be added to connect. Eg.

`{URL HERE}`

	Description	Service Endpoint	Parameters	Keys
1	Endpoint	`Eg?var1={var1}&var2={var2}`	{var1} {var2}	`API KEY`

Data Mapping

Description

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
2	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
3	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
4	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
5	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
6	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
7	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
8	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
9	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
10	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
11	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
12	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
13	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
14	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
15	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
16	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
17	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
18	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
19	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
20	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field

21	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
22	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
23	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
24	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
25	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
26	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field

Documentation Summary:

API documentation available here: Link

UCD Systems

This page contains details on UCD System Integration

CS UCD Virtual Machine

This page contains the details to access the UCD virtual machine.

Hostname	User	Password	OS	Open Ports	Date Received
csi6220-4-vm1.ucd.ie	student	Team10Practicum	Ubuntu Server 20.04	22, 80, 442	14 Jun 2021

CS UCD GIT

This page details integration details (keys) of the application to DublinBikes to pull bike data.
The Table Below Lists the Integration Details and Endpoints.

The info returned is a json

For all integrations the appid parameter must be added to connect. Eg.

`{URL HERE}`

	Description	Service Endpoint	Parameters	Keys
1	Endpoint	`Eg?var1={var1}&var2={var2}`	{var1} {var2}	`API KEY`

Data Mapping

Description

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
2	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
3	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
4	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
5	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
6	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
7	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
8	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
9	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
10	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
11	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
12	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
13	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
14	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
15	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field

16	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
17	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
18	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
19	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
20	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
21	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
22	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
23	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
24	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
25	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field
26	Field Name	Field Type	Yes/No	Yes/No	Yes/No	Sample	Yes/No	Desc. Field

Documentation Summary:

API documentation available here: Link

CS UCD High Performance Server

This page contains the details to access the UCD virtual machine.

Hostname	User	Password	OS	Open Ports	Date Received
telemachus.ucd.ie	team10	4k+s*hUWwQ	?	?	15 Jun 2021

Azure MS SQL

IMPORTANT:

Figuring out Azure/MS SQL credentials has turned out to be a pain, so the below information is not currently valid! Instead, we're using my root Azure login credentials. these are:

username = 'Team10'
password = 'Practicum10'

server = 'tcp:team10-practicum.database.windows.net'
database = 'Practicum_DB'

See the below Jupyter notebook for a more detailed look at how to connect, using SQLAlchemy and PyODBC:

User credentials:

There are 2 credentials available.

team10_Login should be most of the time; it has full read and write access to database tables (SELECT, INSERT, UPDATE, DELETE).

team10_Admin should be used for administration purposes; that is, creating tables, dropping tables, etc. Please make sure that our production server isn't running on these credentials to e.g update a table with weather data info.

Note that these accounts are provisioned to work within the default database, "Practicum_DB"; if we require a different database, I'll need to set up new accounts.

User credentials:

```
Username: Team10Login
Password: Azure_ResearchPracticum
```

Here's the connection string:

```
Server=tcp:team10-practicum.database.windows.net,1433;User
ID=Team10Login;Password=Azure_ResearchPracticum;Encrypt=yes;
TrustServerCertificate=no;Connection Timeout=30;
```

Admin credentials:

```
Username: Team10Login_Admin
Password: 8%L>,./G@412/2U;#}78VFa3_=18>5
```

Here's the connection string:

```
Server=tcp:team10-practicum.database.windows.net,1433;User
ID=Team10Login_Admin;Password="8%L>,./G@412/2U;#}78VFa3_=18>5";
Encrypt=yes;TrustServerCertificate=no;Connection Timeout=30;
```

If you need REALLY serious admin powers, let me (Daniel) know; I can log in with my actual Azure account credentials which run at db_owner level privileges. I can also create another user with owner level privileges if we need to.

Data Model

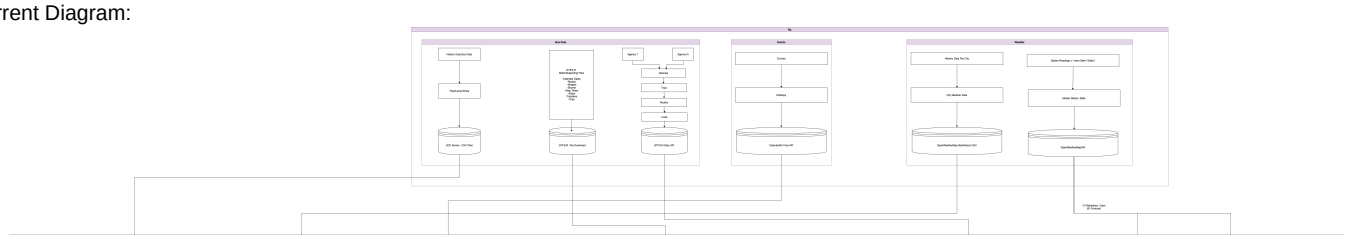
This page is to track the data model and mapping into our database.

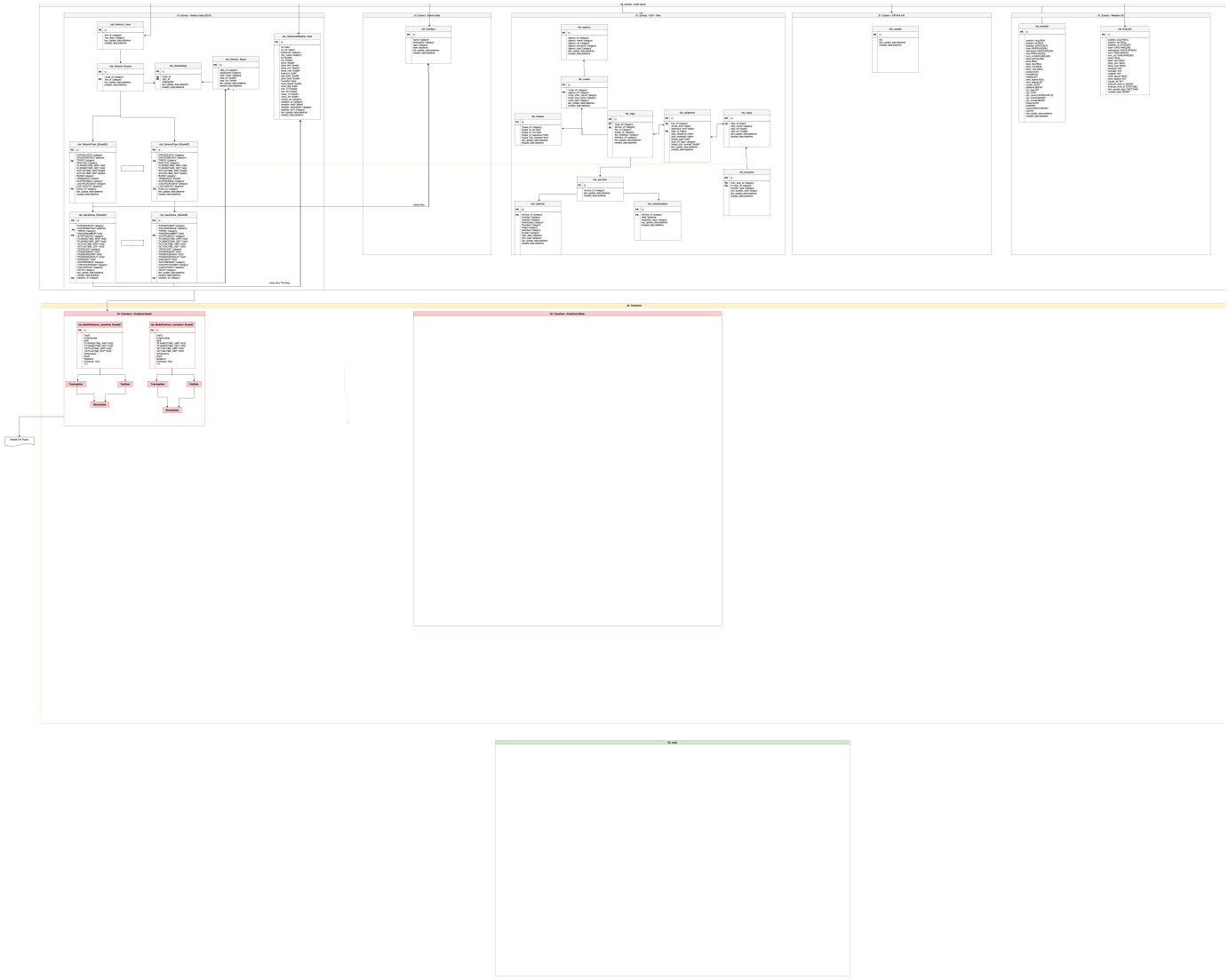
Overall Data Model

This page details the ER of the schema. Each Table should be details as a separate page in the Data Model section.

Overall ER and Data Architecture

Current Diagram:





Source File: 2021-06-28

Extract

This section details tables in the Extract Tables.

Transform

This section details tables in the Transform Section.

Front End Tables

This section details tables in the Transform Section for Front End and visualisation.

Machine Learning Tables

This section details tables in the Transform Section for Machine Learning Capabilities.

Load

This section details tables in the Load section which will be displayed to the customer.

Database Dictionary

This section contains the database schema:

```
database_schema={
                    'table':{
                              'field':'Type'
                              ,'field':'Type'
                                }
                    ,'table':{
                              'field':'Type'
                              ,'field':'Type'
                                }

                    'table':{
                              'field':'Type'
                              ,'field':'Type'
                                }
                }
```

Extract Data Store

The Raw Data Store is the set of tables designed to hold the Raw and Unaltered/Minimally Altered/Extracted Data.

RDS_HistoricBus_RT_Trips

This page details the RDS_HistoricBus_RT_Trips table.

RDS_HistoricBus_RT_Trips Table

This page stores information on trips for 2018.

A trip is a series of stops made by a bus along a specific route contained by a line. This serves as header data for the trips.

RDS_HistoricBus_RT_Trips Mapping

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	ID	BIGINT	Yes	Yes	No	1, 2, …	No	Desc.
2	DAYOFSERVICE	DateTime	No	No	No	07-FEB-18 00:00:00	No	Day of Service - Days More than 24 hours
3	TRIPID	VARCHAR(30)	No	No	Yes	6253783	No	

								Identifier for a trip - a journey taken for a bus
4	LINEID	VARCHAR(10)	No	No	Yes	68	Yes	Identifier for a line - a series of routes a bus might take grouped together
5	ROUTEID	VARCHAR(20)	No	No	Yes	104_16, 104_15, etc.	Yes	Individual Journeys taken by buses
6	DIRECTION	VARCHAR(2)	No	No	No	1,2	Yes	Forward or Backwards
7	PLANNEDTIME_ARR	INT64	No	No	No	87245	Yes	Planned time of arrived in seconds
8	PLANNEDTIME_DEP	INT64	No	No	No	84600	Yes	Planned time of arrival in seconds
9	ACTUALTIME_ARR	INT64	No	No	No	87524	Yes	Actual time of arrival in second
10	ACTUALTIME_DEP	INT64	No	No	No	84600	Yes	Actual time of departure in seconds
11	TENDERLOT	VARCHAR(30)	No	No	No	Tbd - NaN	Yes	Unclear
12	SUPPRESSED	INT32	No	No	No	NaN, 0	Yes	0 if it's suppressed. These contain only planned data. Partially suppressed trips are not flagged. For some reason only 0 is present but: 0=Achieved 1=Suppressed
13	JUSTIFICATIONID	INT64	No	No	Yes	177856, 297896	Yes	Specific fault codes for trips - These are not unique.
14	LASTUPDATE	DATETIME	No	No	No	28-FEB-18 12:05:11	No	Date last modified
15	NOTE	VARCHAR(255)	No	No	No	,2967409,	Yes	Free text - Looks like an ID of some type?
16	created_date	DateTime	No	No	No	Date of Addition	No	Timestamp updated
17	last_modified_data	DateTime	No	No	No	Date of Update	No	Timestamp updated

Note

- Not all trips are unique This could cause join errors.
- DataSource is a FK but can be dropped
- LineID: A specific bus line. E.g. 145, 39A, etc. We've 130 bus lines. Potentially this might be a way to split the data we have into more manageable chunks? Pandas can work with chunks of 10M in the larger files, so potentially splitting each of these chunks into a separate CSV and working per line might work? The key problem in this is we'd need to join on the trip ID first which will be a substantial join.
- Suppressed trips have only planned data. Partially suppressed trips won't be flagged. 4.3k are explicitly flagged.
- Justification ID contains trips with faults. 3.5k of these. -Basin: Only has basDef. Who knows what basin code this is but it's probably not relevant since it's only a single value
- Two directions, need to find out what 1 and 2 corresponds with (forward or backward). This can be done with the trip id I guess. They're not evenly split, probably from late night trips?
- The Routes are a bit weird; one line can have multiple routes (145_1, 145_2, 145_3). It looks like 1 corresponds to forward, 1 to backwards, and maybe others are forward and backward diversions? Certain routes are much more likely to be suppressed.

Summary Data

Model Class

```
class rds_HistoricBus_RT_Trips(db.Model):
    """Trips Table. Data for Historic Bus Trips"""

    __tablename__ = 'rds_HistoricBus_RT_Trips'

    #ID
    id=db.Column(db.Integer,primary_key=True)

    #Date of Service
    DAYOFSERVICE=db.Column(db.DateTime, unique=False,nullable=False)

    #Trip identifier
    TRIPID=db.Column(db.String(30),unique=False,nullable=False)

    #Line Identifier
    LINEID=db.Column(db.String(10),unique=False,nullable=False)

    #Route Identifier
    ROUTEID=db.Column(db.String(20),unique=False,nullable=False)

    #Forward/Backwards
    DIRECTION=db.Column(db.String(2),unique=False,nullable=False)

    #Planned arrival Time
    PLANNEDTIME_ARR=db.Column(db.Integer,unique=False,nullable=False)

    #Planned Departure Time
    PLANNEDTIME_DEP=db.Column(db.Integer,unique=False,nullable=False)

    #Actual Arrival Time
    ACTUALTIME_ARR=db.Column(db.Integer,unique=False,nullable=False)
```

```
    #Actual departure time
    ACTUALTIME_DEP=db.Column(db.Integer,unique=False,nullable=False)

    #Unknown
    TENDERLOT=db.Column(db.String(30),unique=False,nullable=False)

    #Suppressed trip or not
    SUPPRESSED=db.Column(db.Integer,unique=False,nullable=False)

    #Fault reason
    JUSTIFICATIONID=db.Column(db.BIGINT,unique=False,nullable=False)

    #Updated Last by DB
    LASTUPDATE=db.Column(db.DateTime, unique=False,nullable=False)

    #Freeform Note
    NOTE=db.Column(db.String(255),unique=False,nullable=False)

    #When was the entry into the Database
    created_date=db.Column(db.DateTime, default=dt.now)
    #When was the modification into the Database
    last_modified_data=db.Column(db.DateTime, default=dt.now,
onupdate=dt.now)

    ###Relationships

    #TBD
    trips_leavetimes=db.relationship('rds_HistoricBus_RT_LeaveTimes',
backref='trip',lazy=True)

    def __repr__(self):
        """Return View of Self"""
        row_id = self.id
        day = self.DAYOFSERVICE
        line_route_trip = """Line: {} Route: {} Trip: {}""".format(self.
LINEID,self.ROUTEID,self.TRIPID)
        planned_times="Planned Leave: {} Planned Arrive: {}".format
(self.PLANNEDTIME_DEP,self.PLANNEDTIME_ARR)
        actual_times="Actual Leave: {} Actual Arrive: {}".format(self.
ACTUALTIME_ARR,self.ACTUALTIME_DEP)
        suppressed=self.SUPPRESSED
        created = self.created_date

        print_statement="""Row: {}\n
                           Date: {}\n
                           Line_Route_Trip: {}\n
                           Planneds: {}\n
                           Actuals: {}\n
                           Suppressed: {}\n
                           Added to Database: {}\n\n
```

```
                                """.format(row_id
                                           ,day
                                           ,line_route_trip
                                           ,planned_times
                                           ,actual_times
                                           ,suppressed
                                           ,created).replace
("                                ","",-1)

        return print_statement

    def to_dict(self):
        """Convert Model to Dictionary

        Source: https://stackoverflow.com/questions/5022066/how-to-
serialize-sqlalchemy-result-to-json
        """

        column_attribs=inspect(self).mapper.column_attrs
        return_dict={}

        for col in column_attribs:
            col_name=col.key
            col_val=getattr(self, col_name)
            print("{}:{}".format(col_name,col_val))
            return_dict[col_name]=col_val

        return return_dict

    def to_df(self):
        """Convert a model to dataframe.

        Return dataframe
        """
        connect=db.engine.connect()
        df=pd.read_sql("Select * from {}".format(self.__tablename__),
connect)
        connect.close()
        return df
```

RDS_HistoricBus_RT_LeaveTimes

This page details integration details (keys) of the application to the historic trip data.
The Table Below Lists the Integration Details and Endpoints.

The info returned is a csv

For all integrations the appid parameter must be added to connect. Eg.

	Description	Service Endpoint	Parameters	Keys
1	Location	`CS UCD`	N/A	`N/A`

Data Mapping

Description

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	id	INTEGER	Yes	Yes	No			
2	DATASOURCE	DateTime	No	No	No	07-FEB-18 00:00:00	No	Day of Service - Days More than 24 hours
3	DAYOFSERVICE	DateTime	No	No	No	07-FEB-18 00:00:00	No	Day of Service - Days More than 24 hours
4	TRIPID	VARCHAR(30)	No	No	Yes	6253783	No	Identifier for a trip - a journey taken for a bus
5	PROGRNUMBER	INT32	No	No	No	1,4,5,7,8,…	Yes	Order of Stops: Note that some progression numbers are missing.
6	STOPPOINTID	VARCHAR(16)	No	No	Yes	7692	Yes	Stop ID.
7	PLANNEDTIME_ARR	INT64	No	No	No	87245	Yes	Planned time of arrived in seconds since start of day
8	PLANNEDTIME_DEP	INT64	No	No	No	84600	Yes	Planned time of arrival in seconds since start of day
9	ACTUALTIME_ARR	INT64	No	No	No	87524	Yes	Actual time of arrival in second
10	ACTUALTIME_DEP	INT64	No	No	No	84600	Yes	Actual time of departure in seconds
11	VEHICLEID	VARCHAR(6)	No	No	Yes	2693211	Yes	Stop ID.
12	PASSENGERS	INT64	No	No	No	NaN	Yes	All Null
13	PASSENGERSIN	INT64	No	No	No	NaN	Yes	All Null
14	PASSENGERSOUT	INT64	No	No	No	NaN	Yes	All Null
15	DISTANCE	INT64	No	No	No	84600	Yes	All Null
16	SUPPRESSED	INT32	No	No	No	0,1	Yes	Is the trip suppressed for some reason? Only values null and 0 are present. Partial suppressions are not flagged. 0=Achieved. 1=Suppressed
17	JUSTIFICATIONID	INT64	No	No	No	48498138668042 2208.0	Yes	Fault during trip identifier.
18	LASTUPDATE	DateTime	No	No	No	31-JAN-18 21:17:4231	Yes	Date a record was last modified
19	NOTE	VARCHAR(255)	No	No	No	NaN	Yes	All null
20	trip	INT64	No	Yes	No	1,2,3…	No	Primary Key of Trips Table
21	created_date	DateTime	No	No	No	31-JAN-18 21:17:4231	No	Date added
22	last_update_date	Datetime	No	No	No	31-JAN-18 21:17:4231	No	Date updated

Note

- Has over 116 million rows so poses as an issue to open and view the data

- Is related to the Trips data via the TRIPSID
- The routes from trips dataset can be a viable way of splitting the dataset , this can be done by getting all of the tripids that correspond to the route, and splitting selecting from the this dataset the rows that contain those TRIPIDs,
- The individual datasets that conatains the individual lines are around 1 million lines rather , so is much easier to work with and easier to manipulate

Summary Data

- TBD

Model Data

```
#Leavetimes
class rds_HistoricBus_RT_LeaveTimes(db.Model):
    """LeaveTimes Table. Item data for Historic Bus Trips"""

    __tablename__ = 'rds_HistoricBus_RT_Leavetimes'

    #ID
    id=db.Column(db.Integer,primary_key=True)

    #Date of Service
    DAYOFSERVICE=db.Column(db.DateTime, unique=False,nullable=False)

    #Trip identifier
    TRIPID=db.Column(db.String(30),unique=False,nullable=False)

    #Line Identifier
    PROGRNUMBER=db.Column(db.Integer,unique=False,nullable=False)

    #Route Identifier
    STOPPOINTID=db.Column(db.String(16),unique=False,nullable=False)

    #Planned arrival Time
    PLANNEDTIME_ARR=db.Column(db.Integer,unique=False,nullable=False)

    #Planned Departure Time
    PLANNEDTIME_DEP=db.Column(db.Integer,unique=False,nullable=False)

    #Actual Arrival Time
    ACTUALTIME_ARR=db.Column(db.Integer,unique=False,nullable=False)

    #Actual departure time
    ACTUALTIME_DEP=db.Column(db.Integer,unique=False,nullable=False)

    #Unknown
    VEHICLEID=db.Column(db.String(6),unique=False,nullable=False)

    #Total Passengers
    PASSENGERS=db.Column(db.Integer,unique=False,nullable=False)
```

```
    #AIn stop
    PASSENGERS_IN=db.Column(db.Integer,unique=False,nullable=False)

    #AOut Stop
    PASSENGERS_OUT=db.Column(db.Integer,unique=False,nullable=False)

    #Distance from beginning of Trip
    DISTANCE=db.Column(db.Integer,unique=False,nullable=False)

    #Suppressed trip or not
    SUPPRESSED=db.Column(db.Integer,unique=False,nullable=False)

    #Fault reason
    JUSTIFICATIONID=db.Column(db.BIGINT,unique=False,nullable=False)

    #Updated Last by DB
    LASTUPDATE=db.Column(db.DateTime, unique=False,nullable=False)

    #Freeform Note
    NOTE=db.Column(db.String(255),unique=False,nullable=False)

    #When was the entry into the Database
    created_date=db.Column(db.DateTime, default=dt.now)
    #When was the modification into the Database
    last_modified_data=db.Column(db.DateTime, default=dt.now,
onupdate=dt.now)

    ###Relationships
    trips_leavetimes=db.Column(db.Integer,db.ForeignKey
('rds_HistoricBus_RT_Trips.id'),nullable=True)
    trips_weather=db.Column(db.Integer,db.ForeignKey
('rds_HistoricalWeatherData.id'),nullable=True)

    def __repr__(self):
        """Return View of Self"""
        row_id = self.id
        day = self.DAYOFSERVICE
        bus=self.VEHICLEID
        stop_number = """Progression Number: {}""".format(self.LINEID,
self.ROUTEID,self.TRIPID)
        planned_times="Planned Leave: {} Planned Arrive: {}".format
(self.PLANNEDTIME_DEP,self.PLANNEDTIME_ARR)
        actual_times="Actual Leave: {} Actual Arrive: {}".format(self.
ACTUALTIME_ARR,self.ACTUALTIME_DEP)
        suppressed=self.SUPPRESSED
        created = self.created_date

        print_statement="""Row: {}\n
                           Date: {}\n
                           StopNo: {}\n
```

```
                            Vehicle: {}\n
                            Planneds: {}\n
                            Actuals: {}\n
                            Suppressed: {}\n
                            Added to Database: {}\n\n
                            """.format(row_id
                                       ,day
                                       ,stop_number
                                       ,bus
                                       ,planned_times
                                       ,actual_times
                                       ,suppressed
                                       ,created).replace
("                            ","",-1)

        return print_statement

    def to_dict(self):
        """Convert Model to Dictionary

        Source: https://stackoverflow.com/questions/5022066/how-to-
serialize-sqlalchemy-result-to-json
        """

        column_attribs=inspect(self).mapper.column_attrs
        return_dict={}

        for col in column_attribs:
            col_name=col.key
            col_val=getattr(self, col_name)
            print("{}:{}".format(col_name,col_val))
            return_dict[col_name]=col_val

        return return_dict

    def to_df(self):
        """Convert a model to dataframe.

        Return dataframe
        """
        connect=db.engine.connect()
        df=pd.read_sql("Select * from {}".format(self.__tablename__),
connect)
        connect.close()
        return df
```

RDS_HistoricalWeather_Data

This page contains historical weather data.

This page details integration details (keys) of the application to the historic trip data.
The Table Below Lists the Integration Details and Endpoints.

The info returned is a csv

For all integrations the appid parameter must be added to connect. Eg.

	Description	Service Endpoint	Parameters	Keys
1	Location	`CS UCD`	N/A	`N/A`

Data Mapping

Description

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	id	INTEGER	Yes	Yes	No	1,2,3,	No	Primary Key
2	`dt`	BIGINT	Yes	Yes	No	1545102000	Yes/No	Datetime - Timestamp
3	`dt_iso`	DateTime (ISO)	Yes	N/A	No	2018-04-12 10:00:00 +0000 UTC	Yes/No	Datetime
4	`timezone`	INT	No	N/A	No	0,3600	Yes/No	Maybe GMT vs BST
5	`city_name`	VARCHAR(6)	No	N/A	No	Dublin	Yes/No	Dublin
6	`lat`	`float64`	No	Yes	No	53.349805	Yes/No	53.349805
7	`lon`	`float64`	No	Yes	No	-6.26031	Yes/No	-6.26031
8	`temp`	`float64`	No	No	No	280.16	Yes/No	Temperature in Kelvin
9	`feels_like`	`float64`	No	No	No	275.54	Yes/No	Temperature in Kelvin
10	`temp_min`	`float64`	No	No	No	279.15	Yes/No	Temperature in Kelvin
11	`temp_max`	`float64`	No	No	No	289.15	Yes/No	Temperature in Kelvin
12	`pressure`	`int64`	No	No	No	1017	Yes/No	Air Pressure
13	`sea_level`	`float64`	No	No	No	NaN	Yes/No	Null
14	`grnd_level`	`float64`	No	No	No	NaN	Yes/No	Null
15	`humidity`	`int64`	No	No	No	93	Yes/No	Humidity
16	`wind_speed`	`float64`	No	No	No	3.6	Yes/No	Wind speed
17	`wind_deg`	`int64`	No	No	No	250	Yes/No	Wind direction
18	`rain_1h`	`float64`	No	No	No	0.5	Yes/No	Check specific
19	`rain_3h`	`float64`	No	No	No	NaN	Yes/No	Null
20	`snow_1h`	`float64`	No	No	No	0.71	Yes/No	Check specific
21	`snow_3h`	`float64`	No	No	No	NaN	Yes/No	Null
22	`clouds_all`	VARCHAR(255)	No	No	No	75	Yes/No	Check - Cloud Types? Cloud Percent? Unclear
23	`weather_id`	VARCHAR(10)	No	No	Yes	803	Yes/No	Weather Identifier
24	`weather_main`	VARCHAR(255)	No	No	No	Clouds	Yes/No	Main Weather
25	`weather_description`	VARCHAR(500)	No	No	No	broken clouds	Yes/No	Main Weather Description
26	`weather_icon`	VARCHAR(2000)	No	No	No	04d	Yes/No	Weather Icon
27	created_date	DateTime	No	No	No	`2021-01-01 21:17:4231`	No	Date added
28	last_update_date	Datetime	No	No	No	`2021-01-01 21:17:4231`	No	Date updated

Note

- City, lat, long are Dublin.
- Check if the timestamp aligns with the Dublin Bus data.
- 3h times are gone.
- Check act meaning of 1h times.

Summary Data

Model Data

```
#Weather
class rds_HistoricalWeatherData(db.Model):
    """Historical Weather Data"""

    __tablename__ = 'rds_HistoricalWeatherData'

    #ID
    id=db.Column(db.Integer,primary_key=True)

    #Datetime
    dt=db.Column(db.BIGINT, unique=False,nullable=False)

    #Datetime
    dt_iso=db.Column(db.DateTime, unique=False,nullable=False)

    #Timezone
    timezone=db.Column(db.Integer,unique=False,nullable=False)

    #City Name - Dublin
    city_name=db.Column(db.String(6),unique=False,nullable=False)

    #City Latitude - Dublin
    lat=db.Column(db.FLOAT,unique=False,nullable=False)

    #City Longitude - Dublin
```

```
    lon=db.Column(db.FLOAT,unique=False,nullable=False)

    #Temp Kelvin
    temp=db.Column(db.FLOAT,unique=False,nullable=False)

    #Temp Kelvin
    feels_like=db.Column(db.FLOAT,unique=False,nullable=False)

    #Temp Kelvin
    temp_min=db.Column(db.FLOAT,unique=False,nullable=False)

    #Temp Kelvin
    temp_max=db.Column(db.FLOAT,unique=False,nullable=False)

    #Air pressure
    pressure=db.Column(db.Integer,unique=False,nullable=False)

    #Sea Level
    sea_level=db.Column(db.FLOAT,unique=False,nullable=False)

    #Ground Level
    grnd_level=db.Column(db.FLOAT,unique=False,nullable=False)

    #Humidity
    humidity=db.Column(db.Integer,unique=False,nullable=False)

    #Wind speed
    wind_speed=db.Column(db.FLOAT,unique=False,nullable=False)

    #Wind Degree
    wind_deg=db.Column(db.Integer,unique=False,nullable=False)

    #Rain in last hour
    rain_1h=db.Column(db.FLOAT,unique=False,nullable=False)

    #Rain in last three hours
    rain_3h=db.Column(db.FLOAT,unique=False,nullable=False)

    #Snow in last hour
    snow_1h=db.Column(db.FLOAT,unique=False,nullable=False)

    #Snow in last three hours
    snow_3h=db.Column(db.FLOAT,unique=False,nullable=False)

    #Cloud type
    clouds_all=db.Column(db.String(255),unique=False,nullable=False)

    #Weather ID
    weather_id=db.Column(db.Integer,unique=False,nullable=False)
```

```
    #Weather Type
    weather_main=db.Column(db.String(255),unique=False,nullable=False)

    #Wind Description
    weather_description=db.Column(db.String(1000),unique=False,
nullable=False)

    #Weather Icon
    weather_icon=db.Column(db.String(10),unique=False,nullable=False)

    #When was the entry into the Database
    created_date=db.Column(db.DateTime, default=dt.now)

    #When was the modification into the Database
    last_modified_data=db.Column(db.DateTime, default=dt.now,
onupdate=dt.now)

    ###Relationships

    #TBD
    weather_leavetimes=db.relationship('rds_HistoricBus_RT_LeaveTimes',
backref='weather',lazy=True)

    def __repr__(self):
        """Return View of Self"""
        row_id = self.id
        day = self.dt_iso
        weather=self.weather_description
        clouds=self.clouds_all
        temp=self.temp-273.15
        pressure=self.pressure

        print_statement="""Row: {}\n
                           Date: {}\n
                           Weather: {}\n
                           Clouds: {}\n
                           Temp (C): {}\n
                           Pressure: {}\n\n
                           """.format(row_id
                                      ,day
                                      ,weather
                                      ,clouds
                                      ,temp
                                      ,pressure).replace
("                            ","",-1)

        return print_statement

    def to_dict(self):
        """Convert Model to Dictionary
```

```
        Source: https://stackoverflow.com/questions/5022066/how-to-
serialize-sqlalchemy-result-to-json
        """

        column_attribs=inspect(self).mapper.column_attrs
        return_dict={}

        for col in column_attribs:
            col_name=col.key
            col_val=getattr(self, col_name)
            print("{}:{}".format(col_name,col_val))
            return_dict[col_name]=col_val

        return return_dict

    def to_df(self):
        """Convert a model to dataframe.

        Return dataframe
        """
        connect=db.engine.connect()
        df=pd.read_sql("Select * from {}".format(self.__tablename__),
connect)
        connect.close()
        return df
```

RDS_CurrentWeather

This page details the weather forecast data.

Current Weather Table

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	id	INT	Yes	Yes	No	1,2,3,4,	No	Primary Key of Weather Table
2	position_long	REAL	No	No	No	-6.2656	No	Longitude
3	position_lat	REAL	No	No	No	53.3581	No	Latitude
4	weather_id	INTEGER	No	No	No	803	No	Weather ID Type
5	main	VARCHAR(256)	No	No	No	Clouds	No	Short Description
6	description	VARCHAR(500)	No	No	No	broken clouds	No	Weather Description
7	icon	VARCHAR(20)	No	No	No	04d	No	Weather Icon
8	icon_url	VARCHAR(500)	No	No	No	http://openweathermap.org/img/wn/04d@2x.png	No	Icon URL
9	base	varchar(256)	No	No	No	stations	No	Station
10	temp	REAL	No	No	No	286.21	No	Temp
11	feels_like	REAL	No	No	No	277.02	No	Feels Like
12	temp_min	REAL	No	No	No	284.82	No	Min Temp
13	temp_max	REAL	No	No	No	287.15	No	Max Temp

14	pressure	INT	No	No	No	1001	No	Air Pressure
15	humidity	INT	No	No	No	77	No	Humidity
16	visibility	INT	No	No	No	10000	No	Visibility
17	wind_speed	REAL	No	No	No	12.86	No	Wind Speed
18	wind_degree	INT	No	No	No	200	No	Wind Direction
19	clouds_all	INT	No	No	No	75	No	Clouds???
20	datetime	BIGINT	No	No	No	1614079641	No	Datetime
21	sys_type	INT	No	No	No	1	No	?
22	sys_id	INT	No	No	No	1565	No	System ID?
23	sys_country	VARCHAR(10)	No	No	No	IE	No	Country
24	sys_sunrise	BIGINT	No	No	No	1614065172	No	Sunrise Time
25	sys_sunset	BIGINT	No	No	No	1614102658	No	Sunset Time
26	timezone	INT	No	No	No	0	No	Timezone ID
27	default_id	BIGINT	No	No	No	6691027	No	???
28	name	VARCHAR(256)	No	No	No	Drumcondra	No	Location Name
29	cod	INT	No	No	No	200	No	???
30	created_date	DateTime	No	No	No	2021-01-01	No	Datetime added to DB
31	date_last_modified	DateTime	No	No	No	2021-01-01	No	Datetime last updated.

Note

- Need to take the first Weather Instance as the main weather type.
- Not so frequent

Summary Data

- TBD - Leverage Software Engineering.

Model Class

- Here

RDS_WeatherForecast

This page details the weather forecast data.

Weather Forecast Table

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	id	INT	Yes	Yes	No	1,2,3,4,	No	Primary Key of Weather Table
2	position_long	REAL	No	No	No	-6.2656	No	Longitude
3	position_lat	REAL	No	No	No	53.3581	No	Latitude
4	weather_id	INTEGER	No	No	No	803	No	Weather ID Type
5	main	VARCHAR(256)	No	No	No	Clouds	No	Short Description
6	description	VARCHAR(500)	No	No	No	broken clouds	No	Weather Description
7	icon	VARCHAR(20)	No	No	No	04d	No	Weather Icon
8	icon_url	VARCHAR(500)	No	No	No	http://openweathermap.org/img/wn/04d@2x.png	No	Icon URL
9	base	varchar(256)	No	No	No	stations	No	Station
10	temp	REAL	No	No	No	286.21	No	Temp
11	feels_like	REAL	No	No	No	277.02	No	Feels Like
12	temp_min	REAL	No	No	No	284.82	No	Min Temp

13	temp_max	REAL	No	No	No	287.15	No	Max Temp
14	pressure	INT	No	No	No	1001	No	Air Pressure
15	humidity	INT	No	No	No	77	No	Humidity
16	visibility	INT	No	No	No	10000	No	Visibility
17	wind_speed	REAL	No	No	No	12.86	No	Wind Speed
18	wind_degree	INT	No	No	No	200	No	Wind Direction
19	clouds_all	INT	No	No	No	75	No	Clouds???
20	datetime	BIGINT	No	No	No	1614079641	No	Datetime
21	sys_type	INT	No	No	No	1	No	?
22	sys_id	INT	No	No	No	1565	No	System ID?
23	sys_country	VARCHAR(10)	No	No	No	IE	No	Country
24	sys_sunrise	BIGINT	No	No	No	1614065172	No	Sunrise Time
25	sys_sunset	BIGINT	No	No	No	1614102658	No	Sunset Time
26	timezone	INT	No	No	No	0	No	Timezone ID
27	default_id	BIGINT	No	No	No	6691027	No	???
28	name	VARCHAR(256)	No	No	No	Drumcondra	No	Location Name
29	cod	INT	No	No	No	200	No	???
30	created_date	DateTime	No	No	No	2021-01-01	No	Datetime added to DB
31	date_last_modified	DateTime	No	No	No	2021-01-01	No	Datetime last updated.

Note

- Need to take the first Weather Instance as the main weather type.

Summary Data

- TBD - Leverage Software Engineering.

Model Class

- Here

rds_EventsData

rds_RealTimeBusData

rds_Calendar

rds_CalendarDates

rds_Shapes

rds_Stops

rds_Transfers

rds_Trips

rds_Agency

rds_StopTimes

rds_Routes

Transform Data Store

The Transform Data Store involves any staging tables which are used to transform the raw data into loadable data

Machine Learning Store

This section details transformations for the Machine Learning Models.

tds_ML_Model_TrainResults

This page details the {Table_Name_2} table.

It captures {Table_Name_2_Desc}

{Table_Name_2} Table

{SDesc} Mapping

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	Field Name	Field Type	Yes/No	Yes/No	Yes/No	1, 2, … , ,a,b,c, …'ab',…	Yes/No	Desc.
2	Field Name	Field Type	Yes/No	Yes/No	Yes/No	1, 2, … , ,a,b,c, …'ab',…	Yes/No	Desc.
3	Field Name	Field Type	Yes/No	Yes/No	Yes/No	1, 2, … , ,a,b,c, …'ab',…	Yes/No	Desc.
4	created_date	DateTime	No	No	No	Date of Addition	No	Timestamp updated
5	last_modified_data	DateTime	No	No	No	Date of Update	No	Timestamp updated

Note

Schema Notes

Summary Data

(include analysis of raw data)

rds_ModelTrain

tds_ML_Model

Frontend Data Store

This section details transformations for the frontend of the application.

user

This page details the {Table_Name_3} table.

It captures {Table_Name_3_Desc}

{Table_Name_3} Table

{SDesc} Mapping

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	Field Name	Field Type	Yes/No	Yes/No	Yes/No	1, 2, … , ,a,b,c, …'ab',…	Yes/No	Desc.
2	Field Name	Field Type	Yes/No	Yes/No	Yes/No	1, 2, … , ,a,b,c, …'ab',…	Yes/No	Desc.
3	Field Name	Field Type	Yes/No	Yes/No	Yes/No		Yes/No	Desc.

						1, 2, … , ,a,b,c, …'ab',…		
4	created_date	DateTime	No	No	No	Date of Addition	No	Timestamp updated
5	last_modified_data	DateTime	No	No	No	Date of Update	No	Timestamp updated

Note

Schema Notes

Summary Data

(include analysis of raw data)

tds_User_FavouriteRoutes

tds_User_TripHistory

tds_Routes_Trips_Stops

tds_Stops_Per_Route

tds_User_AppActionLog

tds_User_TravelCardTypes

Loading Data Store

The Load tables feature any table which are then loaded into Analytic Data Stores

full_routes.json

This page details the {Table_Name_4} table.

It captures {Table_Name_4_Desc}

{Table_Name_4} Table

{SDesc} Mapping

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	Field Name	Field Type	Yes/No	Yes/No	Yes/No	1, 2, … , ,a,b,c, …'ab',…	Yes/No	Desc.
2	Field Name	Field Type	Yes/No	Yes/No	Yes/No	1, 2, … , ,a,b,c, …'ab',…	Yes/No	Desc.
3	Field Name	Field Type	Yes/No	Yes/No	Yes/No	1, 2, … , ,a,b,c, …'ab',…	Yes/No	Desc.
4	created_date	DateTime	No	No	No	Date of Addition	No	Timestamp updated
5	last_modified_data	DateTime	No	No	No	Date of Update	No	Timestamp updated

Note

Schema Notes

Summary Data

(include analysis of raw data)

route_Names.json

stops_with_routes.json

Design Decisions

This page should document all design decisions.

Docker and containers

This page is to document the container-based development approach undertaken in the development and deployment of this project.

Docker containers were developed and utilised by the team. One container was a production-ready deployment, built and ran on our UCD VM instance; the other was a development environment designed to run on the machines of our team members during development.

Both containers maintain feature parity in terms of the images and major software initialised and configured to run per-container; they differ in the end setup used to enabled HTTPS on port 443. On the UCD VM, which is hosted on a named URL, we can avail of Let's Encrypt, a free provider of genuine TLS certificates.

However, to facilitate HTTPS-based development on local machines, a different approach was needed; a self-signed certificate was created and issued as part of the development branch build, and developers must modify their hosts file to enabled the self-signed certificates to work. This allows for both development and production to function on HTTPS, a traditionally tricky situation to manage.

Please see the development branch or the production branch for more detailed setup documentation.

Why Docker?

Docker allows you to spin up “containerised” instances of almost any software stack. Containers are basically microcosms of entire operating systems and software configurations; Docker containers leverage the widespread availability of virtualisation technology in modern CPUs to run instances of other operating systems from within your base OS, and each of these instances is referred to as a “container”. You specify how to set up your container in a “Dockerfile”, and when you want to run your container, you “build” your dockerfile out to a full container; you must re-build your container after changing a dockerfile.

In your dockerfile, you typically start by pulling an “image” of an operating system, e.g Ubuntu, and then specifying what next you’d like docker to; i.e installing Python 3 with “sudo apt-get python3”, or running a command to enabled or disable a built-in OS function. Docker allows you to map internal ports in each container, to actual ports on your real system; this means that a webserver running in your docker container might *think* it’s running on ports 80 and 443, but it’s actually accessible on your machine on ports 101 and 5665, as you’ve decided to tell Docker to expose those internal ports on those external ports.

This allows you to e.g run a node-express-nginx stack in one container, directly adjacent to a flask-wsgi-apache instance, on different external ports, allowing both containers to interoperate on your end system when referenced with their exposed external ports.

The biggest benefit to Docker, however, is that once a container is defined and the setup process documented (which, in itself, is almost never complicated), containers can be stopped, changed, and rebuilt very quickly. In contrast to configuring a living, breathing VM OS with your desired web stack, you are free to tinker with a docker container, knowing you can simply shut it down and roll back to your previous configuration should you break something; tinkering with a running OS, however, is a nightmare, as it’s much harder to make, store, and restore backups of a running machine and all of it’s required files, than it is to restart a docker container.

Docker also allows some convenient functionality through persistent volumes. By default, Docker container data is non-persistent; you can modify the files inside a running container, but restarting or rebuilding the container will reset those changes, as docker rebuilds a container to the exact specification of it’s dockerfile, and nothing more. Persistent volumes are a feature of docker that allows one to map a real folder on the user’s OS, and mount it to a docker image, such that it appears in the running container as a folder that persists changes from both ends.

The Jira epic related to getting our docker instances up and running is detailed in the following Jira epic board:

COMP47360-63 - As a team member, I want a synchronised development and production environment
TO DO

UAT, PREPROD, PROD Plan

This page details the version planning.

Deployment Guide

This section should contain details on how to deploy the app and set it up from scratch.

Pre-Requisite Requirements

Environments

Instructions Per Environment

Other…

Retrospectives

Error rendering macro 'create-from-template' : Failed to find net.sf.hibernate.Session from the current thread

Title	Date	Participants
Sprint 1 - Retrospective	12 Jun 2021	@ Adam Ryan @ Turlough Hannon @ Danning Zhan @ Daniel Danev