Disappster Home

Hey all,

This is the homepage of our Confluence.

Confluence is primarily used for Software Engineering project documentation and comes with many cool features and automations which can be run (currently an automation is in place for automatic epic completion once sub-tasks are marked!). 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:

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 Requirements

Target release	16 Apr 2021
MVP Release Date	14 Apr 2021
Document status	FINAL
Confluence owner	@ Adam Ryan
Product Owner	@ karl.roe

Objective

Deliverable Project

This is not an exhaustive list, but contains the main elements of a working solution

1. data collection through JCDecaux API
2. data management/storage in RDS DB on AWS 3. display bike stations on map
3. flask web application (python)
4. occupancy information
5. weather information
6. interactivity (click, API request, handle response)
7. ML model for predicting occupancy based on weather patterns, trained on collected data
8. project served on a named host over the web on your EC2 instance NAMED HOST REQUIREMENT DROPPED

9. Github repo, including source code, logs, commit history, branches, etc

Project Management

1. Design, planning notes and materials, discussion of Requirements, mock
2. Scrum project management (trello board, slack logs, google docs, etc)
3. Meeting notes (log of daily standup, notes from sprint reviews, retrospec- tives
4. feature selection/product backlog for each sprint (× 4)
5. burndown charts, (sprint and/or release) for each sprint (× 4)
6. managing deliverables, prototypes

Personal Notes (Individual Submission)

1. personal review
2. retrospective of the project

Key Goal: As a user, I wish to be able to access an interactable flask web application wherein I can view occupancy and weather information and get a prediction as to what the available bikes will be like for a certain day based on an ML model, so that I can take bike journeys using a bike sharing scheme.

Success metrics

Goal	Metric
I can access a website	URL is Valid
Weather info is on that website	Weather Info is Displayed
Availability info is on that website	Availability Info is Displayed
Station info is on that website	Station Info is Displayed
I can interact with the website	The user can Click Stations
I can predict future bike availability	The user can predict availability in the Future

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.
https://disappster.atlassian.net/jira/software/projects/COMP30830/boards/1/roadmap

Product Roadmap

Team mission

The goal of this project is to develop a web application to display occupancy and weather information for Dublin Bikes

🔗 Project information

- Insert Assignment Name Here

📋 Roadmap overview

https://disappster.atlassian.net/jira/software/projects/COMP30830/boards/1/roadmap

🔍 Detailed sprint roadmap

Sprint 1 - SM: AR

Feature	Initiative	Dates	Priority	Effort	Status	Description
Establish Jira Board	N/A	21 Feb 2021	HIGH	HIGH	SHIPPED	Get on Confluence and Jira. Get Page structure and template set up.
Get Set Up on Platforms and Retrieve Data	https://disappster.atlassian.net/browse/COMP30830-42 https://disappster.atlassian.net/browse/COMP30830-31https://disappster.atlassian.net/browse/COMP30830-52	21 Feb 2021	HIGH	LOW	IN PROGRESS	Begin retrieving data daily.
Start Pulling Weather Data	https://disappster.atlassian.net/browse/COMP30830-66 https://disappster.atlassian.net/browse/COMP30830-73 https://disappster.atlassian.net/browse/COMP30830-79	28 Feb 2021	MEDIUM	LOW	NOT STARTED	As an extended sprint goal, begin retrieving the weather data.

Sprint 2

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

Sprint 3

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

Sprint 4

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

Sprint 5

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

Project Management

Sprint Overview

Meeting Notes

Workshops

Other

Team Member Contact List

This page is to track the contact information of the key team project members

	FirstName	LastName	Internal /External	Role	Email	PhoneNumber	Availability	Jira @
1	Adam	Ryan	Internal	Developer	adam.ryan@ucdconnect.ie	0834240698	Tuesdays, Thursdays.	@ Adam Ryan
2	Finnian	Rogers	Internal	Developer	finnian.rogers@ucdconnect.ie		All	@ Finnian Rogers
3	Jane	Slevin	Internal	Developer	jane.slevin@ucdconnect.ie		All	@ Jane Slevin
4	Karl	Roe	Internal	Product Owner	karl.roe@ucdconnect.ie		Thursday Mornings	@ karl.roe
5	Aonghus	Lawlor	External	Client	aonghus.lawlor@ucd.ie		Project Review.	@ aonghus.lawlor

Requirements - Discovery Phase

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

Assignment Submissions

These pages detail the assignments submitted.

Sprints

This page is designed to detail a copy of the sprint Epics and Tasks listed under the burndown charts.

On this version of Confluence and Jira (non-Cloud) I cannot embed reports or a macro detailing the tasks per sprint. Thus they will be copied from the burndown report on Jira.

Sprint 1

*Issue added after sprint start

Scope changes log

View in issue navigator

Date	Key	Summary	Issue type	Details of scope change

2021-02-17	COMP30830-32*	Create a JCDecaux Account	Task	Issue added to sprint
2021-02-17	COMP30830-33*	Create an Amazon MySQL Database	Task	Issue added to sprint
2021-02-17	COMP30830-31*	TM1- As a team member, I need to begin pulling and storing Dublin Bike Info, to gather historical data.	Epic	Issue added to sprint
2021-02-17	COMP30830-31	TM1- As a team member, I need to begin pulling and storing Dublin Bike Info, to gather historical data.	Epic	Issue removed from sprint
2021-02-17	COMP30830-31*	TM1- As a team member, I need to begin pulling and storing Dublin Bike Info, to gather historical data.	Epic	Issue added to sprint
2021-02-17	COMP30830-42*	TM2- As a team member, I need to begin pulling and storing Dublin Bike Info, to gather historical data.	Epic	Issue added to sprint
2021-02-17	COMP30830-52*	TM3- As a team member, I need to begin pulling and storing Dublin Bike Info, to gather historical data.	Epic	Issue added to sprint
2021-02-17	COMP30830-34*	Write Code to retrieve Dublin Bike JSON Data	Task	Issue added to sprint
2021-02-17	COMP30830-35*	Write Code to Store Data Into Text File	Task	Issue added to sprint
2021-02-17	COMP30830-36*	Write Code to Create Table if not exist.	Task	Issue added to sprint
2021-02-17	COMP30830-37*	Write Code to Connect to mySQL Database	Task	Issue added to sprint
2021-02-17	COMP30830-38*	Write Code to Insert JSON Data Into mySQL Database	Task	Issue added to sprint
2021-02-17	COMP30830-40*	Set this function live and running every five minutes on EC2 Instance.	Task	Issue added to sprint
2021-02-17	COMP30830-39*	Write Wrapping Function to contain all operations for retrieve data and posting into database.	Task	Issue added to sprint
2021-02-17	COMP30830-43*	Create a JCDecaux Account	Task	Issue added to sprint
2021-02-17	COMP30830-44*	Create an Amazon MySQL Database	Task	Issue added to sprint
2021-02-17	COMP30830-45*	Write Code to retrieve Dublin Bike JSON Data	Task	Issue added to sprint
2021-02-17	COMP30830-46*	Write Code to Store Data Into Text File	Task	Issue added to sprint
2021-02-17	COMP30830-47*	Write Code to Create Table if not exist.	Task	Issue added to sprint
2021-02-17	COMP30830-48*	Write Code to Connect to mySQL Database	Task	Issue added to sprint
2021-02-17	COMP30830-49*	Write Code to Insert JSON Data Into mySQL Database	Task	Issue added to sprint
2021-02-17	COMP30830-50*	Write Wrapping Function to contain all operations for retrieve data and posting into database.	Task	Issue added to sprint

2021-02-17	COMP30830-51*	Set this function live and running every five minutes on EC2 Instance.	Task	Issue added to sprint
2021-02-17	COMP30830-53*	Create a JCDecaux Account	Task	Issue added to sprint
2021-02-17	COMP30830-54*	Create an Amazon MySQL Database	Task	Issue added to sprint
2021-02-17	COMP30830-55*	Write Code to retrieve Dublin Bike JSON Data	Task	Issue added to sprint
2021-02-17	COMP30830-56*	Write Code to Store Data Into Text File	Task	Issue added to sprint
2021-02-17	COMP30830-57*	Write Code to Create Table if not exist.	Task	Issue added to sprint
2021-02-17	COMP30830-58*	Write Code to Connect to mySQL Database	Task	Issue added to sprint
2021-02-17	COMP30830-59*	Write Code to Insert JSON Data Into mySQL Database	Task	Issue added to sprint
2021-02-17	COMP30830-60*	Write Wrapping Function to contain all operations for retrieve data and posting into database.	Task	Issue added to sprint
2021-02-17	COMP30830-61*	Set this function live and running every five minutes on EC2 Instance.	Task	Issue added to sprint
2021-02-18	COMP30830-62*	As a team, we need to choose a singular repository to work from, in order to have a shared code base.	Epic	Issue added to sprint
2021-02-18	COMP30830-64*	Decide as a team GIT Practices	Task	Issue added to sprint
2021-02-18	COMP30830-63*	Decide as a group what the structure should look like	Task	Issue added to sprint
2021-02-18	COMP30830-66*	TM1-2: As a team member, I need to begin pulling and storing Weather Info, as a way to set up how weather affects Bike Capacity	Epic	Issue added to sprint
2021-02-18	COMP30830-84*	Document Results on Weather Integration Confluence Page	Task	Issue added to sprint
2021-02-18	COMP30830-83*	Write Weather JSON to SQL Database Function	Task	Issue added to sprint
2021-02-18	COMP30830-82*	Retrieve Weather JSON Function	Task	Issue added to sprint
2021-02-18	COMP30830-81*	As a team, we need to decide what tables go into our Weather Table and how to structure it.	Task	Issue added to sprint
2021-02-18	COMP30830-80*	As a team, we need to agree where we pull Weather Data from	Task	Issue added to sprint
2021-02-18	COMP30830-78*	Document Results on Weather Integration Confluence Page	Task	Issue added to sprint
2021-02-18	COMP30830-77*	Write Weather JSON to SQL Database Function	Task	Issue added to sprint
2021-02-18	COMP30830-76*	Retrieve Weather JSON Function	Task	Issue added to sprint

2021-02-18	COMP30830-75*	As a team, we need to decide what tables go into our Weather Table and how to structure it.	Task	Issue added to sprint
2021-02-18	COMP30830-74*	As a team, we need to agree where we pull Weather Data from	Task	Issue added to sprint
2021-02-18	COMP30830-65*	Update Confluence with Details of Integration (key to access, URL, Diagram of process)	Task	Issue added to sprint
2021-02-18	COMP30830-68*	As a team, we need to decide what tables go into our Weather Table and how to structure it.	Task	Issue added to sprint
2021-02-18	COMP30830-70*	Write Weather JSON to SQL Database Function	Task	Issue added to sprint
2021-02-18	COMP30830-71*	Document Results on Weather Integration Confluence Page	Task	Issue added to sprint
2021-02-18	COMP30830-69*	Retrieve Weather JSON Function	Task	Issue added to sprint
2021-02-18	COMP30830-67*	As a team, we need to agree where we pull Weather Data from	Task	Issue added to sprint

Your sprint commitment has increased by 50 issues

Due to scope changes: You have 50 issues to complete this sprint

Incomplete issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
COMP30830-84	Document Results on Weather Integration Confluence Page	Task		**DONE**		1
COMP30830-81	As a team, we need to decide what tables go into our Weather Table and how to structure it.	Task		**DONE**		1
COMP30830-78	Document Results on Weather Integration Confluence Page	Task		**DONE**		1

Completed issues

View in issue navigator

Key	Summary	Issue type	Status
COMP30830-32	Create a JCDecaux Account	Task	**DONE**
COMP30830-33	Create an Amazon MySQL Database	Task	**DONE**
COMP30830-31	TM1- As a team member, I need to begin pulling and storing Dublin Bike Info, to gather historical data.	Epic	**DONE**

COMP30830-42	TM2- As a team member, I need to begin pulling and storing Dublin Bike Info, to gather historical data.	Epic	**DONE**
COMP30830-52	TM3- As a team member, I need to begin pulling and storing Dublin Bike Info, to gather historical data.	Epic	**DONE**
COMP30830-34	Write Code to retrieve Dublin Bike JSON Data	Task	**DONE**
COMP30830-35	Write Code to Store Data Into Text File	Task	**DONE**
COMP30830-36	Write Code to Create Table if not exist.	Task	**DONE**
COMP30830-37	Write Code to Connect to mySQL Database	Task	**DONE**
COMP30830-38	Write Code to Insert JSON Data Into mySQL Database	Task	**DONE**
COMP30830-40	Set this function live and running every five minutes on EC2 Instance.	Task	**DONE**
COMP30830-39	Write Wrapping Function to contain all operations for retrieve data and posting into database.	Task	**DONE**
COMP30830-43	Create a JCDecaux Account	Task	**DONE**
COMP30830-44	Create an Amazon MySQL Database	Task	**DONE**
COMP30830-45	Write Code to retrieve Dublin Bike JSON Data	Task	**DONE**
COMP30830-46	Write Code to Store Data Into Text File	Task	**DONE**
COMP30830-47	Write Code to Create Table if not exist.	Task	**DONE**
COMP30830-48	Write Code to Connect to mySQL Database	Task	**DONE**
COMP30830-49	Write Code to Insert JSON Data Into mySQL Database	Task	**DONE**
COMP30830-50	Write Wrapping Function to contain all operations for retrieve data and posting into database.	Task	**DONE**
COMP30830-51	Set this function live and running every five minutes on EC2 Instance.	Task	**DONE**
COMP30830-53	Create a JCDecaux Account	Task	**DONE**
COMP30830-54	Create an Amazon MySQL Database	Task	**DONE**
COMP30830-55	Write Code to retrieve Dublin Bike JSON Data	Task	**DONE**
COMP30830-56	Write Code to Store Data Into Text File	Task	**DONE**
COMP30830-57	Write Code to Create Table if not exist.	Task	**DONE**
COMP30830-58	Write Code to Connect to mySQL Database	Task	**DONE**
COMP30830-59	Write Code to Insert JSON Data Into mySQL Database	Task	**DONE**

COMP30830-60	Write Wrapping Function to contain all operations for retrieve data and posting into database.	Task	**DONE**
COMP30830-61	Set this function live and running every five minutes on EC2 Instance.	Task	**DONE**
COMP30830-62	As a team, we need to choose a singular repository to work from, in order to have a shared code base.	Epic	**DONE**
COMP30830-64	Decide as a team GIT Practices	Task	**DONE**
COMP30830-63	Decide as a group what the structure should look like	Task	**DONE**
COMP30830-66	TM1-2: As a team member, I need to begin pulling and storing Weather Info, as a way to set up how weather affects Bike Capacity	Epic	**DONE**
COMP30830-83	Write Weather JSON to SQL Database Function	Task	**DONE**
COMP30830-82	Retrieve Weather JSON Function	Task	**DONE**
COMP30830-80	As a team, we need to agree where we pull Weather Data from	Task	**DONE**
COMP30830-77	Write Weather JSON to SQL Database Function	Task	**DONE**
COMP30830-76	Retrieve Weather JSON Function	Task	**DONE**
COMP30830-75	As a team, we need to decide what tables go into our Weather Table and how to structure it.	Task	**DONE**
COMP30830-74	As a team, we need to agree where we pull Weather Data from	Task	**DONE**
COMP30830-65	Update Confluence with Details of Integration (key to access, URL, Diagram of process)	Task	**DONE**
COMP30830-68	As a team, we need to decide what tables go into our Weather Table and how to structure it.	Task	**DONE**
COMP30830-70	Write Weather JSON to SQL Database Function	Task	**DONE**
COMP30830-71	Document Results on Weather Integration Confluence Page	Task	**DONE**
COMP30830-69	Retrieve Weather JSON Function	Task	**DONE**
COMP30830-67	As a team, we need to agree where we pull Weather Data from	Task	**DONE**

Issues completed outside of sprint

Key	Summary	Issue type	Epic	Status	Assignee	Issue count

Sprint 2

Date - 28 February 2021 - 18 March 2021Date - 28 February 2021 to 18 March 2021

Sprint goal - Skeleton Flask app, Wireframe, Stations on Map

Remaining work

Number of issues left to complete this sprint

Guideline

Ideal burn rate

Report: COMP30830 Sprint 2

*Issue added after sprint start

Scope changes log

View in issue navigator

Date	Key	Summary	Issue type	Details of scope change
2021-03-03	COMP30830-78*	Document Results on Weather Integration Confluence Page	Task	Issue added to sprint
2021-03-03	COMP30830-84*	Document Results on Weather Integration Confluence Page	Task	Issue added to sprint
2021-03-03	COMP30830-81*	As a team, we need to decide what tables go into our Weather Table and how to structure it.	Task	Issue added to sprint
2021-03-23	COMP30830-101*	Decide on a weather API	Task	Issue added to sprint
2021-03-23	COMP30830-102*	Show map on HTML Page	Task	Issue added to sprint
2021-03-23	COMP30830-108*	Show Markers on Map in Frontend	Task	Issue added to sprint
2021-03-23	COMP30830-110*	Initial setup of flask app.	Task	Issue added to sprint
2021-03-23	COMP30830-111*	Decide on pages to have in website.	Task	Issue added to sprint
2021-03-23	COMP30830-115*	Function to get latest availability data	Task	Issue added to sprint
2021-03-23	COMP30830-119*	TM1 - Show Bike Station Names	Task	Issue added to sprint
2021-03-23	COMP30830-121*	TM3 - Show Weather Positions	Task	Issue added to sprint
2021-03-23	COMP30830-120*	TM2 - Show availability info (last five minutes)	Task	Issue added to sprint
2021-03-23	COMP30830-122*	Build a wireframe of the front end view.	Task	Issue added to sprint
2021-03-23	COMP30830-123*	Create a burndown chart	Task	Issue added to sprint
2021-03-23	COMP30830-124*	Create a dropdown to filter station markers	Task	Issue added to sprint

Your sprint commitment has increased by 15 issues

Due to scope changes: You have 15 issues to complete this sprint

Incomplete issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
COMP30830-123	Create a burndown chart	Task		**DONE**		1

Completed issues

View in issue navigator

Key	Summary	Issue type	Status
COMP30830-78	Document Results on Weather Integration Confluence Page	Task	**DONE**
COMP30830-84	Document Results on Weather Integration Confluence Page	Task	**DONE**
COMP30830-81	As a team, we need to decide what tables go into our Weather Table and how to structure it.	Task	**DONE**
COMP30830-101	Decide on a weather API	Task	**DONE**
COMP30830-102	Show map on HTML Page	Task	**DONE**
COMP30830-108	Show Markers on Map in Frontend	Task	**DONE**
COMP30830-110	Initial setup of flask app.	Task	**DONE**
COMP30830-111	Decide on pages to have in website.	Task	**DONE**
COMP30830-115	Function to get latest availability data	Task	**DONE**
COMP30830-119	TM1 - Show Bike Station Names	Task	**DONE**
COMP30830-121	TM3 - Show Weather Positions	Task	**DONE**
COMP30830-120	TM2 - Show availability info (last five minutes)	Task	**DONE**
COMP30830-122	Build a wireframe of the front end view.	Task	**DONE**
COMP30830-124	Create a dropdown to filter station markers	Task	**DONE**

Issues completed outside of sprint

Key	Summary	Issue type	Epic	Status	Assignee	Issue count

Sprint 3

Date - 19 March 2021 - 4 April 2021Date - 19 March 2021 to 4 April 2021

Sprint goal - Visualisations, Graphs, Frontend Features

Remaining work

Number of issues left to complete this sprint

Guideline

Ideal burn rate

Report: COMP30830 Sprint 3

*Issue added after sprint start

Scope changes log

View in issue navigator

Date	Key	Summary	Issue type	Epic	Details of scope change	Change in estimation
2021-03-23	COMP30830-94*	As a user, I want to select stations from a dropdown.	Epic		Issue added to sprint	1
2021-03-23	COMP30830-129*	As a user I want to be able to choose to filter graphs by Weather Type	Epic		Issue added to sprint	1
2021-03-23	COMP30830-109*	Colour station icons by availability (?)	Task		Issue added to sprint	1
2021-03-23	COMP30830-112*	Decide visualisations to show.	Task		Issue added to sprint	1
2021-03-23	COMP30830-116*	Hide already open Markers on Selection	Task		Issue added to sprint	1
2021-03-23	COMP30830-117*	Dropdown menu to show the station list and choose a Station	Task		Issue added to sprint	1
2021-03-23	COMP30830-133*	I need to create a pandas function to summarise availability by day	Task	**AS A USER, I WANT TO SEE THE AVAILABILITY BY DAY OVER ALL TIME.**	Issue added to sprint	1
2021-03-23	COMP30830-135*	I need to create a JS function to graph the data by day over all time.	Task	**AS A USER, I WANT TO SEE THE AVAILABILITY BY DAY OVER ALL TIME.**	Issue added to sprint	1
2021-03-23	COMP30830-136*	I need to create a pandas function to summarise the data	Task	**AS A USER, I WANT TO SEE VISUALISATION OF THE AVAILABILITY INFO BY DAY.**	Issue added to sprint	1
2021-03-23	COMP30830-138*	I need to create a JS function to graph the data by hour	Task	**AS A USER, I WANT TO SEE VISUALISATION OF THE AVAILABILITY INFO BY DAY.**	Issue added to sprint	1
2021-03-23	COMP30830-139*	I need to create a pandas function to summarise the data	Task		Issue added to sprint	1
2021-03-23	COMP30830-141*	I need to create a Javascript function to graph the data	Task		Issue added to sprint	1
2021-03-23	COMP30830-142*	I need to create a header	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1

2021-03-23	COMP30830-143*	I need footer on the page	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-144*	I need a section for charts	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-145*	I need a section for graphs	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-147*	I need a default view for chart section	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-148*	I need CSS for my sections to look nice.	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-150*	We will merge our code from Sprint 1 and 2 into a single Flask App.	Task		Issue added to sprint	1
2021-03-23	COMP30830-153*	Build a function to show the nearest station with available bikes.	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	Issue added to sprint	1
2021-03-23	COMP30830-149*	If we can add in animations that would be cool	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-152*	Build the javascript function to show that location on the map	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	Issue added to sprint	1
2021-03-23	COMP30830-155*	Add Engine Unit Tests	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	Issue added to sprint	1
2021-03-24	COMP30830-154*	As a group, we need to implement unit tests into our application to ensure proper functionality	Epic		Issue added to sprint	1
2021-03-24	COMP30830-157*	Add database schema unit tests (check against data_dictionary)	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	Issue added to sprint	1
2021-03-24	COMP30830-156*	Add dataframe Unit Tests	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	Issue added to sprint	1
2021-03-25	COMP30830-158*	As a group we need a section in the body for charting areas and selectors at side or bottom of page	Epic		Issue added to sprint	1
2021-03-25	COMP30830-159*	Add a container underneath/at side of chart.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1
2021-03-25	COMP30830-160*	Move the selector into the container.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1

2021-03-25	COMP30830-161*	Edit CSS of Map to fit both into body	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1
2021-03-25	COMP30830-162*	Javascript function to Edit section and add charts on station selection	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1
2021-03-25	COMP30830-163*	Add radio button and Javascript function to filter markers by colour	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1
2021-03-25	COMP30830-164*	Add radio button and based on selection filter markers by available bikes or stands.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1
2021-03-25	COMP30830-165*	Open info window for that station after selecting it in the selector	Task		Issue added to sprint	1
2021-03-25	COMP30830-166*	Sort selector so the stations display alphabetically	Task		Issue added to sprint	1
2021-03-25	COMP30830-166	Sort selector so the stations display alphabetically	Task		Issue removed from sprint	-1
2021-03-25	COMP30830-166*	Sort selector so the stations display alphabetically	Task		Issue added to sprint	1
2021-03-30	COMP30830-123*	Create a burndown chart	Task		Issue added to sprint	1
2021-03-30	COMP30830-128*	As a user, I want to see visualisation of the availability info by week.	Epic		Issue added to sprint	1
2021-03-30	COMP30830-127*	As a group we need a unified codebase	Epic		Issue added to sprint	1
2021-03-30	COMP30830-125*	As a project manager, I want to see what the work progress is like	Epic		Issue added to sprint	1
2021-04-01	COMP30830-167*	Update wireframe to include planned contents of side panel.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1

Your sprint commitment has increased by 40 issues

Due to scope changes: You have 40 issues to complete this sprint

Incomplete issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count

COMP30830-129	As a user I want to be able to choose to filter graphs by Weather Type	Epic		**TO DO**		1
COMP30830-135	I need to create a JS function to graph the data by day over all time.	Task	**AS A USER, I WANT TO SEE THE AVAILABILITY BY DAY OVER ALL TIME.**	**DONE**		1
COMP30830-138	I need to create a JS function to graph the data by hour	Task	**AS A USER, I WANT TO SEE VISUALISATION OF THE AVAILABILITY INFO BY DAY.**	**DONE**		1
COMP30830-147	I need a default view for chart section	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-153	Build a function to show the nearest station with available bikes.	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	**TO DO**		1
COMP30830-149	If we can add in animations that would be cool	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-152	Build the javascript function to show that location on the map	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	**DONE**		1
COMP30830-154	As a group, we need to implement unit tests into our application to ensure proper functionality	Epic		**TO DO**		1
COMP30830-157	Add database schema unit tests (check against data_dictionary)	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	**BLOCKED**		1
COMP30830-156	Add dataframe Unit Tests	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	**BLOCKED**		1
COMP30830-158	As a group we need a section in the body for charting areas and selectors at side or bottom of page	Epic		**TO DO**		1
COMP30830-164	Add radio button and based on selection filter markers by available bikes or stands.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**BLOCKED**		1

Completed issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
COMP30830-94	As a user, I want to select stations from a dropdown.	Epic		**DONE**		1
COMP30830-109	Colour station icons by availability (?)	Task		**DONE**		1
COMP30830-112	Decide visualisations to show.	Task		**DONE**		1
COMP30830-116	Hide already open Markers on Selection	Task		**DONE**		1
COMP30830-117	Dropdown menu to show the station list and choose a Station	Task		**DONE**		1
COMP30830-133	I need to create a pandas function to summarise availability by day	Task	**AS A USER, I WANT TO SEE THE AVAILABILITY BY DAY OVER ALL TIME.**	**DONE**		1

COMP30830-136	I need to create a pandas function to summarise the data	Task	**AS A USER, I WANT TO SEE VISUALISATION OF THE AVAILABILITY INFO BY DAY.**	**DONE**		1
COMP30830-139	I need to create a pandas function to summarise the data	Task		**DONE**		1
COMP30830-141	I need to create a Javascript function to graph the data	Task		**DONE**		1
COMP30830-142	I need to create a header	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-143	I need footer on the page	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-144	I need a section for charts	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-145	I need a section for graphs	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-148	I need CSS for my sections to look nice.	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-150	We will merge our code from Sprint 1 and 2 into a single Flask App.	Task		**DONE**		1
COMP30830-155	Add Engine Unit Tests	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	**DONE**		1
COMP30830-159	Add a container underneath/at side of chart.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1
COMP30830-160	Move the selector into the container.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1
COMP30830-161	Edit CSS of Map to fit both into body	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1
COMP30830-162	Javascript function to Edit section and add charts on station selection	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1
COMP30830-163	Add radio button and Javascript function to filter markers by colour	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1
COMP30830-165	Open info window for that station after selecting it in the selector	Task		**DONE**		1
COMP30830-166	Sort selector so the stations display alphabetically	Task		**DONE**		1
COMP30830-123	Create a burndown chart	Task		**DONE**		1
COMP30830-128	As a user, I want to see visualisation of the availability info by week.	Epic		**DONE**		1
COMP30830-167	Update wireframe to include planned contents of side panel.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1

Issues completed outside of sprint

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
COMP30830-127	As a group we need a unified codebase	Epic		**DONE**		1
COMP30830-125	As a project manager, I want to see what the work progress is like	Epic		**DONE**		1

Sprint 4

Sprint 3 and Sprint 4 Epic List

This page will detail a copy of the burndown report task list for Sprint 3 and Sprint 4. These cannot be added to the Sprints page due to page limits.

Sprint 3

Date - 19 March 2021 - 4 April 2021Date - 19 March 2021 to 4 April 2021

Sprint goal - Visualisations, Graphs, Frontend Features

Remaining work

Number of issues left to complete this sprint

Guideline

Ideal burn rate

Report: COMP30830 Sprint 3

*Issue added after sprint start

Scope changes log

View in issue navigator

Date	Key	Summary	Issue type	Epic	Details of scope change	Change in estimation
2021-03-23	COMP30830-94*	As a user, I want to select stations from a dropdown.	Epic		Issue added to sprint	1
2021-03-23	COMP30830-129*	As a user I want to be able to choose to filter graphs by Weather Type	Epic		Issue added to sprint	1
2021-03-23	COMP30830-109*	Colour station icons by availability (?)	Task		Issue added to sprint	1
2021-03-23	COMP30830-112*	Decide visualisations to show.	Task		Issue added to sprint	1
2021-03-23	COMP30830-116*	Hide already open Markers on Selection	Task		Issue added to sprint	1

2021-03-23	COMP30830-117*	Dropdown menu to show the station list and choose a Station	Task		Issue added to sprint	1
2021-03-23	COMP30830-133*	I need to create a pandas function to summarise availability by day	Task	**AS A USER, I WANT TO SEE THE AVAILABILITY BY DAY OVER ALL TIME.**	Issue added to sprint	1
2021-03-23	COMP30830-135*	I need to create a JS function to graph the data by day over all time.	Task	**AS A USER, I WANT TO SEE THE AVAILABILITY BY DAY OVER ALL TIME.**	Issue added to sprint	1
2021-03-23	COMP30830-136*	I need to create a pandas function to summarise the data	Task	**AS A USER, I WANT TO SEE VISUALISATION OF THE AVAILABILITY INFO BY DAY.**	Issue added to sprint	1
2021-03-23	COMP30830-138*	I need to create a JS function to graph the data by hour	Task	**AS A USER, I WANT TO SEE VISUALISATION OF THE AVAILABILITY INFO BY DAY.**	Issue added to sprint	1
2021-03-23	COMP30830-139*	I need to create a pandas function to summarise the data	Task		Issue added to sprint	1
2021-03-23	COMP30830-141*	I need to create a Javascript function to graph the data	Task		Issue added to sprint	1
2021-03-23	COMP30830-142*	I need to create a header	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-143*	I need footer on the page	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-144*	I need a section for charts	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-145*	I need a section for graphs	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-147*	I need a default view for chart section	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-148*	I need CSS for my sections to look nice.	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1
2021-03-23	COMP30830-150*	We will merge our code from Sprint 1 and 2 into a single Flask App.	Task		Issue added to sprint	1
2021-03-23	COMP30830-153*	Build a function to show the nearest station with available bikes.	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	Issue added to sprint	1
2021-03-23	COMP30830-149*	If we can add in animations that would be cool	Task	**AS USER, I WANT A NICE USER INTERFACE.**	Issue added to sprint	1

2021-03-23	COMP30830-152*	Build the javascript function to show that location on the map	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	Issue added to sprint	1
2021-03-23	COMP30830-155*	Add Engine Unit Tests	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	Issue added to sprint	1
2021-03-24	COMP30830-154*	As a group, we need to implement unit tests into our application to ensure proper functionality	Epic		Issue added to sprint	1
2021-03-24	COMP30830-157*	Add database schema unit tests (check against data_dictionary)	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	Issue added to sprint	1
2021-03-24	COMP30830-156*	Add dataframe Unit Tests	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	Issue added to sprint	1
2021-03-25	COMP30830-158*	As a group we need a section in the body for charting areas and selectors at side or bottom of page	Epic		Issue added to sprint	1
2021-03-25	COMP30830-159*	Add a container underneath/at side of chart.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1
2021-03-25	COMP30830-160*	Move the selector into the container.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1
2021-03-25	COMP30830-161*	Edit CSS of Map to fit both into body	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1
2021-03-25	COMP30830-162*	Javascript function to Edit section and add charts on station selection	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1
2021-03-25	COMP30830-163*	Add radio button and Javascript function to filter markers by colour	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1
2021-03-25	COMP30830-164*	Add radio button and based on selection filter markers by available bikes or stands.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1
2021-03-25	COMP30830-165*	Open info window for that station after selecting it in the selector	Task		Issue added to sprint	1
2021-03-25	COMP30830-166*	Sort selector so the stations display alphabetically	Task		Issue added to sprint	1
2021-03-25	COMP30830-166	Sort selector so the stations display alphabetically	Task		Issue removed from sprint	-1
2021-03-25	COMP30830-166*	Sort selector so the stations display alphabetically	Task		Issue added to sprint	1

2021-03-30	COMP30830-123*	Create a burndown chart	Task		Issue added to sprint	1
2021-03-30	COMP30830-128*	As a user, I want to see visualisation of the availability info by week.	Epic		Issue added to sprint	1
2021-03-30	COMP30830-127*	As a group we need a unified codebase	Epic		Issue added to sprint	1
2021-03-30	COMP30830-125*	As a project manager, I want to see what the work progress is like	Epic		Issue added to sprint	1
2021-04-01	COMP30830-167*	Update wireframe to include planned contents of side panel.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	Issue added to sprint	1

Your sprint commitment has increased by 40 issues

Due to scope changes: You have 40 issues to complete this sprint

Incomplete issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
COMP30830-129	As a user I want to be able to choose to filter graphs by Weather Type	Epic		**TO DO**		1
COMP30830-135	I need to create a JS function to graph the data by day over all time.	Task	**AS A USER, I WANT TO SEE THE AVAILABILITY BY DAY OVER ALL TIME.**	**DONE**		1
COMP30830-138	I need to create a JS function to graph the data by hour	Task	**AS A USER, I WANT TO SEE VISUALISATION OF THE AVAILABILITY INFO BY DAY.**	**DONE**		1
COMP30830-147	I need a default view for chart section	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-153	Build a function to show the nearest station with available bikes.	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	**TO DO**		1
COMP30830-149	If we can add in animations that would be cool	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-152	Build the javascript function to show that location on the map	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	**DONE**		1
COMP30830-154	As a group, we need to implement unit tests into our application to ensure proper functionality	Epic		**TO DO**		1
COMP30830-157	Add database schema unit tests (check against data_dictionary)	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	**BLOCKED**		1
COMP30830-156	Add dataframe Unit Tests	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	**BLOCKED**		1

COMP30830-158	As a group we need a section in the body for charting areas and selectors at side or bottom of page	Epic		TO DO		1
COMP30830-164	Add radio button and based on selection filter markers by available bikes or stands.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	BLOCKED		1

Completed issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
COMP30830-94	As a user, I want to select stations from a dropdown.	Epic		DONE		1
COMP30830-109	Colour station icons by availability (?)	Task		DONE		1
COMP30830-112	Decide visualisations to show.	Task		DONE		1
COMP30830-116	Hide already open Markers on Selection	Task		DONE		1
COMP30830-117	Dropdown menu to show the station list and choose a Station	Task		DONE		1
COMP30830-133	I need to create a pandas function to summarise availability by day	Task	**AS A USER, I WANT TO SEE THE AVAILABILITY BY DAY OVER ALL TIME.**	DONE		1
COMP30830-136	I need to create a pandas function to summarise the data	Task	**AS A USER, I WANT TO SEE VISUALISATION OF THE AVAILABILITY INFO BY DAY.**	DONE		1
COMP30830-139	I need to create a pandas function to summarise the data	Task		DONE		1
COMP30830-141	I need to create a Javascript function to graph the data	Task		DONE		1
COMP30830-142	I need to create a header	Task	**AS USER, I WANT A NICE USER INTERFACE.**	DONE		1
COMP30830-143	I need footer on the page	Task	**AS USER, I WANT A NICE USER INTERFACE.**	DONE		1
COMP30830-144	I need a section for charts	Task	**AS USER, I WANT A NICE USER INTERFACE.**	DONE		1
COMP30830-145	I need a section for graphs	Task	**AS USER, I WANT A NICE USER INTERFACE.**	DONE		1
COMP30830-148	I need CSS for my sections to look nice.	Task	**AS USER, I WANT A NICE USER INTERFACE.**	DONE		1
COMP30830-150	We will merge our code from Sprint 1 and 2 into a single Flask App.	Task		DONE		1
COMP30830-155	Add Engine Unit Tests	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	DONE		1

COMP30830-159	Add a container underneath/at side of chart.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1
COMP30830-160	Move the selector into the container.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1
COMP30830-161	Edit CSS of Map to fit both into body	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1
COMP30830-162	Javascript function to Edit section and add charts on station selection	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1
COMP30830-163	Add radio button and Javascript function to filter markers by colour	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1
COMP30830-165	Open info window for that station after selecting it in the selector	Task		**DONE**		1
COMP30830-166	Sort selector so the stations display alphabetically	Task		**DONE**		1
COMP30830-123	Create a burndown chart	Task		**DONE**		1
COMP30830-128	As a user, I want to see visualisation of the availability info by week.	Epic		**DONE**		1
COMP30830-167	Update wireframe to include planned contents of side panel.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**DONE**		1

Issues completed outside of sprint

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
COMP30830-127	As a group we need a unified codebase	Epic		**DONE**		1
COMP30830-125	As a project manager, I want to see what the work progress is like	Epic		**DONE**		1

Sprint 4

This is the status as of 9pm. 18 hours remain so this is subject to change.

Date - 2 April 2021 - 18 April 2021Date - 2 April 2021 to 18 April 2021

Sprint goal - Complete the project. Deploy the app.

Remaining work

Number of issues left to complete this sprint

Guideline

Ideal burn rate

Report: COMP30830 Sprint 4

*Issue added after sprint start

Scope changes log

View in issue navigator

Date	Key	Summary	Issue type	Epic	Details of scope change	Change in estimation
2021-04-03	COMP30830-202*	Add 'click' functionality to list of nearest stations	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	Issue added to sprint	1
2021-04-03	COMP30830-203*	Extend scope of function to view nearest stations regardless of location (currently only works if user is within 600m of a station/stations)	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	Issue added to sprint	1
2021-04-03	COMP30830-204*	Add CSS to list of nearest stations	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	Issue added to sprint	1
2021-04-15	COMP30830-200*	Display predicted bikes for a single station.	Task	**AS A USER, I WANT TO ENTER A DATE, A TIME, AND A LOCATION, IN ORDER TO SEE WHICH NEARBY STATIONS WILL HAVE A BIKE AT THAT TIME.**	Issue added to sprint	1

Your sprint commitment has increased by 4 issues

Due to scope changes: You have 50 issues to complete this sprint

Incomplete issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
COMP30830-174	Develop a table to hold what the user enters to trigger the model and what the returned predictions were.	Task	**AS A USER, I WANT TO ENTER A DATE, A TIME, AND A LOCATION, IN ORDER TO SEE WHICH NEARBY STATIONS WILL HAVE A BIKE AT THAT TIME.**	**BLOCKED**		1
COMP30830-179	Develop a radio button that reruns the JS functions displaying Available Stands instead of Available Bikes.	Task	**AS A USER, I WANT TO BE ABLE TO CHOOSE WHETHER THE DISPLAY RELATES TO AVAILABLE BIKES OR AVAILABLE STANDS**	**BLOCKED**		1
COMP30830-183	An about page should be built explaining what the station is, showing high level details, and potentially comparing with the Dublin Bikes website.	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**TO DO**		1
COMP30830-185	Ensure all design decisions are documented on JIRA.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**IN PROGRESS**		1

COMP30830-189	Ensure the Website Wireframe is documented on Jira.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**TO DO**		1
COMP30830-190	Ensure the GitKraken timeline is documented in the DevOps process.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**IN PROGRESS**		1
COMP30830-195	Ensure the final report on Intro, Sprint, pre/during/post planning are documented on Jira.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**TO DO**		1
COMP30830-197	Ensure all of the tickets are categorised with priorities on JIRA.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**BLOCKED**		1
COMP30830-198	Ensure redundant code is removed from the codebase or archived in GIT.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**BLOCKED**		1
COMP30830-97	As a group, we need to establish a domain name to visit the website.	Epic		**TO DO**		1
COMP30830-132	As a user, I want to enter a date and time to see what stations will be available.	Epic		**TO DO**		1
COMP30830-168	As a user, I want to enter a date, a time, and a location, in order to see which nearby stations will have a bike at that time.	Epic		**TO DO**		1
COMP30830-169	As a user, I want to enable Geolocation, to see nearest bikes	Epic		**TO DO**		1
COMP30830-178	As a user, I want to be able to choose whether the display relates to Available Bikes or Available Stands	Epic		**TO DO**		1
COMP30830-184	As a product owner, I want the documentation to be fully complete, to grade the group.	Epic		**TO DO**		1
COMP30830-129	As a user I want to be able to choose to filter graphs by Weather Type	Epic		**TO DO**		1
COMP30830-164	Add radio button and based on selection filter markers by available bikes or stands.	Task	**AS A GROUP WE NEED A SECTION IN THE BODY FOR CHARTING AREAS AND SELECTORS AT SIDE OR BOTTOM OF PAGE**	**BLOCKED**		1
COMP30830-158	As a group we need a section in the body for charting areas and selectors at side or bottom of page	Epic		**TO DO**		1
COMP30830-153	Build a function to show the nearest station with available bikes.	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	**TO DO**		1
COMP30830-156	Add dataframe Unit Tests	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	**BLOCKED**		1

COMP30830-157	Add database schema unit tests (check against data_dictionary)	Task	**AS A GROUP, WE NEED TO IMPLEMENT UNIT TESTS INTO OUR APPLICATION TO ENSURE PROPER FUNCTIONALITY**	**BLOCKED**		1
COMP30830-154	As a group, we need to implement unit tests into our application to ensure proper functionality	Epic		**TO DO**		1

Completed issues

View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Issue count
COMP30830-151	Investigate API for how to put a location in for a user	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	**DONE**		1
COMP30830-170	Develop a table in the database to hold training data. This should be labelled "02_..." to note it's part of the transformation process.	Task	**AS A USER, I WANT TO ENTER A DATE, A TIME, AND A LOCATION, IN ORDER TO SEE WHICH NEARBY STATIONS WILL HAVE A BIKE AT THAT TIME.**	**DONE**		1
COMP30830-171	Develop a table in the database to hold test data. This should be labelled "02_..." to note it's part of the transformation process.	Task	**AS A USER, I WANT TO ENTER A DATE, A TIME, AND A LOCATION, IN ORDER TO SEE WHICH NEARBY STATIONS WILL HAVE A BIKE AT THAT TIME.**	**DONE**		1
COMP30830-172	Document the new tables created on Confluence.	Task	**AS A USER, I WANT TO ENTER A DATE, A TIME, AND A LOCATION, IN ORDER TO SEE WHICH NEARBY STATIONS WILL HAVE A BIKE AT THAT TIME.**	**DONE**		1
COMP30830-173	Develop a table to hold forecasting predictions for the next five/seven days. This should run daily. Tables should start with "01_" as import data.	Task	**AS A USER, I WANT TO ENTER A DATE, A TIME, AND A LOCATION, IN ORDER TO SEE WHICH NEARBY STATIONS WILL HAVE A BIKE AT THAT TIME.**	**DONE**		1
COMP30830-175	Build a method to call the model and return the prediction results.	Task	**AS A USER, I WANT TO ENTER A DATE, A TIME, AND A LOCATION, IN ORDER TO SEE WHICH NEARBY STATIONS WILL HAVE A BIKE AT THAT TIME.**	**DONE**		1
COMP30830-176	Build out how the prediction results appear on the front end.	Task	**AS A USER, I WANT TO ENTER A DATE, A TIME, AND A LOCATION, IN ORDER TO SEE WHICH NEARBY STATIONS WILL HAVE A BIKE AT THAT TIME.**	**DONE**		1
COMP30830-177	Develop a model per station and document the results on Confluence. Embed the script into the FlaskApp.	Task	**AS A USER, I WANT TO ENTER A DATE, A TIME, AND A LOCATION, IN ORDER TO SEE WHICH NEARBY STATIONS WILL HAVE A BIKE AT THAT TIME.**	**DONE**		1
COMP30830-180	The map overlays with the footer.	Bug	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-181	The sidebar dropdown occasionally shows duplicates.	Bug	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-182	The charts are sometimes not very responsive in how quickly they load.	Bug	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1

COMP30830-186	Ensure all of the tables and integration keys are detailed in Jira.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**DONE**		1
COMP30830-187	Ensure all of the Sprints are documented on Confluence.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**DONE**		1
COMP30830-188	Ensure all of the Workshops are documented on Jira.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**DONE**		1
COMP30830-191	Ensure the Project Requirements are documented on Confluence.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**DONE**		1
COMP30830-192	Ensure the most recent data flow is documented on Confluence	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**DONE**		1
COMP30830-194	Ensure the routes are documented on Confluence.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**DONE**		1
COMP30830-196	Ensure the Burndown and Velocity Reports are pasted into Jira and Add in Sprint 2's chart.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**DONE**		1
COMP30830-199	Ensure the code is modularised.	Task	**AS A PRODUCT OWNER, I WANT THE DOCUMENTATION TO BE FULLY COMPLETE, TO GRADE THE GROUP.**	**DONE**		1
COMP30830-138	I need to create a JS function to graph the data by hour	Task	**AS A USER, I WANT TO SEE VISUALISATION OF THE AVAILABILITY INFO BY DAY.**	**DONE**		1
COMP30830-135	I need to create a JS function to graph the data by day over all time.	Task	**AS A USER, I WANT TO SEE THE AVAILABILITY BY DAY OVER ALL TIME.**	**DONE**		1
COMP30830-149	If we can add in animations that would be cool	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-147	I need a default view for chart section	Task	**AS USER, I WANT A NICE USER INTERFACE.**	**DONE**		1
COMP30830-152	Build the javascript function to show that location on the map	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	**DONE**		1
COMP30830-202	Add 'click' functionality to list of nearest stations	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	**DONE**		1
COMP30830-203	Extend scope of function to view nearest stations regardless of location (currently only works if user is within 600m of a station/stations)	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	**DONE**		1
COMP30830-204	Add CSS to list of nearest stations	Task	**(OPTIONAL) AS A USER, I WANT TO SEE THE NEAREST STATION TO ME.**	**DONE**		1

COMP30830-200	Display predicted bikes for a single station.	Task	**AS A USER, I WANT TO ENTER A DATE, A TIME, AND A LOCATION, IN ORDER TO SEE WHICH NEARBY STATIONS WILL HAVE A BIKE AT THAT TIME.**	**DONE**		1

Issues completed outside of sprint

Key	Summary	Issue type	Epic	Status	Assignee	Issue count

No issues have been completed outside of the sprint

Meetings

This page is designed to track meeting notes.

Daily Standups

These pages are to track daily standup notes.

Daily Standup - WK05

Information	Values
Date:	13 Feb 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan @ Jane Slevin @ Finnian Rogers
Daily Standup Time:	08:45am - 09:00am
Agenda:	WIP
Link:	meet.google.com/sme-tpoy-wxt

Attendants

- Student - @ Adam Ryan
- Student - @ Jane Slevin
- Student - @ Finnian Rogers

Meeting Goals:

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

Discussion Topics

- Adam's Work in Progress + Blockers
- Finnian's Work in Progress + Blockers
- Jane's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Jane	Finnian
1	**16 Feb 2021**	-	-	-
2	**17 Feb 2021**	Store/Requests/DB	Store/Requests/DB	Store/Requests/DB
3	**18 Feb 2021**	Jira and Confluence	Connecting to DB	Connecting to DB and Writing to DB
4	**19 Feb 2021**	Jira and Confluence	Has Store/DB done	Store/DB done

Work in Progress

	Date	Adam	Jane	Finnian
1	**16 Feb 2021**	Store Requests	Store/Requests/DB	Store/Requests/DB
2	**17 Feb 2021**	Working RDS/Confluence Notes	Finishing Store Function	Finishing Store Function
3	**18 Feb 2021**	Wrapping function + git	Finishing connecting to DB piece	Finishing connecting to DB and Writing to DB piece
4	**19 Feb 2021**	Wrapping function	Wrapping function	Wrapping function

Blockers

	Date	Adam	Jane	Finnian
1	**16 Feb 2021**	-	-	-
2	**17 Feb 2021**	-	-	-
3	**18 Feb 2021**	-	SQL Alchemy Connection Error --See 1_to_1	-
4	**19 Feb 2021**	-	-	-

Action items

- [x] @ Adam Ryan - Send Updated Code as sample into Discord.
- [x] Resolve SQLAlchemy Blocker

Decisions

- May not be relevant

Daily Standup - WK06

Information	Values
Date:	13 Feb 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan @ Jane Slevin @ Finnian Rogers
Daily Standup Time:	08:45am - 09:00am
Agenda:	WIP
Link:	meet.google.com/sme-tpoy-wxt

Attendants

- Student - @ Adam Ryan
- Student - @ Finnian Rogers
- Student - @ Jane Slevin

Meeting Goals:

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

Discussion Topics

- Adam's Work in Progress + Blockers
- Finnian's Work in Progress + Blockers
- Jane's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Jane	Finnian
1	**22 Feb 2021**	Push the EC2 Code	Looking at Weather	Looking at Weather
2	**23 Feb 2021**	Investigated Dupes, worked on weather	Looking into AWS push	Looking into AWS push
3	**24 Feb 2021**	Weather up, dupe solves	Working on EC2.	Github setup, harvester working.
4	**25 Feb 2021**	Everythinging up and running, pushed to Git, no issues.	Up on EC2 - weather to do	Up on EC2 - Weather to do.
5	**26 Feb 2021**	N/A	N.A - Midterm other module	Weather sorted - waiting on API key

Work in Progress

	Date	Adam	Jane	Finnian
1	**22 Feb 2021**	Fixing EC2 Bug	Wrapping it up	Wrapping it up
2	**23 Feb 2021**	Finished weather push initial, duplicates?	Plan to finish pushing on EC2 today + Github	Plan to finish pushing on EC2 today + Github
3	**24 Feb 2021**	Supporting as needed	Working on EC2.	Pushing harvester to EC2.
4	**25 Feb 2021**	Waiting for Flask lecture.	Working on weather	Working on weather
5	**26 Feb 2021**	Weather running okay	Working on weather piece	Waiting for Weather Wrapper

Blockers

	Date	Adam	Jane	Finnian
1	**22 Feb 2021**	-	-	-
2	**23 Feb 2021**	-	-	-
3	**24 Feb 2021**	-	-	-
4	**25 Feb 2021**	-	-	-
5	**26 Feb 2021**	-	-	OpenWeatherMapAPI

Action items

- [ ] @ScrumMaster - Meet with X to discuss Ticket Y

Decisions

- The user journeys have been finalised

Daily Standup - WK07

Information	Values
Date:	13 Feb 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan @ Finnian Rogers @ Jane Slevin
Daily Standup Time:	08:45am - 09:00am
Agenda:	WIP
Link:	meet.google.com/sme-tpoy-wxt

Attendants

- Student - @ Adam Ryan
- Student - @ Finnian Rogers
- Student - @ Jane Slevin

Meeting Goals:

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

Discussion Topics

- Adam's Work in Progress + Blockers
- Finnian's Work in Progress + Blockers
- Jane's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Jane	Finnian
1	**01 Mar 2021**	No update	Finished Weather	Finished Weather
2	**02 Mar 2021**	Signed up for Gmaps API	Getting Adam's scraper on EC2	Getting Adam's scraper on EC2
3	**03 Mar 2021**	Working on flask app	Wireframe, has Adam's scraper running on EC2	Working on flask app (pulling and displaying tables ok now) has Adam's scraper running on EC2
4	**04 Mar 2021**	Basic flask app working	Working on flask app	Google maps displaying in web page
5	**05 Mar 2021**	no update	no update	no update

Work in Progress

	Date	Adam	Jane	Finnian
1	**01 Mar 2021**	Sprint 2 Preplanning	Deploy weather to EC2	Deploy weather to EC2
2	**02 Mar 2021**	Basic flask app	Getting Adam's scraper on EC2, sign up for Gmaps API	Getting Adam's scraper on EC2, sign up for Gmaps API
3	**03 Mar 2021**	working on flask app	going to watch flask lectures and start flask app	Going to watch google maps lecture and look at google maps api
4	**04 Mar 2021**	Build out flask app	Work on flask app	Try to get markers on google map
5	**05 Mar 2021**	Same as yesterday	Same as yesterday	Same as yesterday

Blockers

	Date	Adam	Jane	Finnian
1	**01 Mar 2021**	-	-	-
2	**02 Mar 2021**	-	-	-
3	**03 Mar 2021**	-	-	-
4	**04 Mar 2021**	-	-	-
5	**05 Mar 2021**	-	-	-

Action items

- [ ] @ScrumMaster - Meet with X to discuss Ticket Y

Decisions

- The user journeys have been finalised

Daily Standup WK07.1 - MT1

Information	Values
Date:	13 Feb 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan @ Jane Slevin @ Finnian Rogers
Daily Standup Time:	08:45am - 09:00am
Agenda:	WIP
Link:	meet.google.com/sme-tpoy-wxt

Attendants

- Student - @ Adam Ryan
- Student - @ Finnian Rogers
- Student - @ Jane Slevin

Meeting Goals:

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

Discussion Topics

- Adam's Work in Progress + Blockers
- Finnian's Work in Progress + Blockers
- Jane's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Jane	Finnian
1	**08 Mar 2021**	Got markers to display on map	Working through google maps tutorial	Working through google maps tutorial
2	**09 Mar 2021**	No update - working on other modules	No update - working on other modules	No update - working on other modules
3	**10 Mar 2021**	Has all station markers showing on map with basic information displaying on click	Has all station markers showing on map with basic information displaying on click	Has all station markers showing on map with basic information displaying on click
4	**11 Mar 2021**	No update - working on other modules	No update - working on other modules	Attempted to fill drop down with station names - did not work
5	**12 Mar 2021**	No update - working on other modules	No update - working on other modules	No update - working on other modules

Work in Progress

	Date	Adam	Jane	Finnian
1	**08 Mar 2021**	Adding interactivity to markers	Working through google maps tutorial	Working through google maps tutorial
2	**09 Mar 2021**	No update - working on other modules	No update - working on other modules	No update - working on other modules
3	**10 Mar 2021**	Work on making a dropdown list with all the stations in it	Work on making a dropdown list with all the stations in it	Work on making a dropdown list with all the stations in it
4	**11 Mar 2021**	No update - working on other modules	No update - working on other modules	No update - working on other modules
5	**12 Mar 2021**	No update - working on other modules	No update - working on other modules	No update - working on other

Blockers

	Date	Adam	Jane	Finnian
1	**08 Mar 2021**	making markers interactive	-	Markers not adding to map
2	**09 Mar 2021**	No update - working on other modules	-	No update - working on other modules
3	**10 Mar 2021**	-	-	-
4	**11 Mar 2021**	-	-	-
5	**12 Mar 2021**	-	-	-

Action items

- [ ] @ScrumMaster - Meet with X to discuss Ticket Y

Decisions

- The user journeys have been finalised

Daily Standup - WK07.2 - MT2

Information	Values
Date:	13 Feb 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan @ Finnian Rogers @ Jane Slevin
Daily Standup Time:	08:45am - 09:00am
Agenda:	WIP
Link:	meet.google.com/sme-tpoy-wxt

Attendants

- Student - @ Adam Ryan
- Student - Finnian
- Student - Jane

Meeting Goals:

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

Discussion Topics

- Adam's Work in Progress + Blockers
- Finnian's Work in Progress + Blockers
- Jane's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Jane	Finnian
1	**15 Mar 2021**	Filling selector with station names	Filling selector with station names	Filling selector with station names
2	**16 Mar 2021**	No update	Filled selector	No update
3	**17 Mar 2021**	Holiday (no meeting)	Holiday (no meeting)	Holiday (no meeting)
4	**18 Mar 2021**	Using flask SQL-alchemy instead of SQl alchemy, split app.py into models.py, config.py and views.py - Object oriented approachc	Filtered markers	Filled selector and filtered markers
5	**19 Mar 2021**	SQL queries to availability table	SQL queries to availability table	SQL queries to availability table

Work in Progress

	Date	Adam	Jane	Finnian
1	**15 Mar 2021**	Filling selector with station names	Filling selector with station names	Filling selector with station names
2	**16 Mar 2021**	Same as yesterday	Connecting markers to selector	Same as yesterday

3	**17 Mar 2021**	Holiday (no meeting)	Holiday (no meeting)	Holiday (no meeting)
4	**18 Mar 2021**	SQL queries to availability table	SQL queries to availability table	SQL queries to availability table
5	**19 Mar 2021**	Same as yesterday	Same as yesterday	Same as yesterday

Blockers

	Date	Adam	Jane	Finnian
1	**15 Mar 2021**	-	-	-
2	**16 Mar 2021**	-	-	2300 stations appearing instead of 109 in json object
3	**17 Mar 2021**	Holiday (no meeting)	Holiday (no meeting)	Holiday (no meeting)
4	**18 Mar 2021**	-	-	-
5	**19 Mar 2021**	-	-	-

Action items

- [ ] @ScrumMaster - Meet with X to discuss Ticket Y

Decisions

- The user journeys have been finalised

Daily Standup - WK08

Information	Values
Date:	13 Feb 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan @ Jane Slevin @ Finnian Rogers
Daily Standup Time:	08:45am - 09:00am
Agenda:	WIP
Link:	meet.google.com/sme-tpoy-wxt

Attendants

- Student - @ Adam Ryan
- Student - @ Finnian Rogers
- Student - @ Jane Slevin

Meeting Goals:

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

Discussion Topics

- Adam’s Work in Progress + Blockers

- Finnian's Work in Progress + Blockers
- Jane's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Jane	Finnian
1	**22 Mar 2021**	SQL query - most recent update.	SQL query - most recent update.	SQL query - most recent update.
2	**23 Mar 2021**	Further SQL queries.	SQL query - most recent update.	SQL query - most recent update.
3	**24 Mar 2021**	Code merge.	Code merge.	Code merge.
4	**25 Mar 2021**	Adding CSS to homepage. Completion of code merge.	Adding number of available bikes to marker.	Colour coding markers based on number of available bikes.
5	**26 Mar 2021**	Pandas function to summarise availability.	No update, working on other modules.	Modified home route to display station list in alphabetical order in dropdown.

Work in Progress

	Date	Adam	Jane	Finnian
1	**22 Mar 2021**	SQL query - most recent update. Review visualisation lecture.	SQL query - most recent update. Review visualisation lecture.	SQL query - most recent update. Review visualisation lecture.
2	**23 Mar 2021**	Meet to review division of tasks. Work on visualisations.	Meet to review division of tasks. Work on visualisations.	Meet to review division of tasks. Work on visualisations.
3	**24 Mar 2021**	Code merge. Work on visualisations.	Code merge. Work on visualisations.	Code merge. Work on visualisations.
4	**25 Mar 2021**	Pandas functions to summarise availability.	Add container to hold graphs.	Javascript functions to graph availability data.
5	**26 Mar 2021**	Pandas function to summarise availability.	Add container to hold graphs.	Javascript functions to graph availability data.

Blockers

	Date	Adam	Jane	Finnian
1	**22 Mar 2021**	-	-	Ill over weekend
2	**23 Mar 2021**	-	-	-
3	**24 Mar 2021**	-	-	-
4	**25 Mar 2021**	-	-	Adam completing Pandas functions to return required data for graphing.
5	**26 Mar 2021**	-	-	-

Action items

- [ ] @ScrumMaster - Meet with X to discuss Ticket Y

Decisions

- The user journeys have been finalised

Daily Standup - WK09

Information	Values

Date:	13 Feb 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan @ Jane Slevin @ Finnian Rogers
Daily Standup Time:	08:45am - 09:00am
Agenda:	WIP
Link:	meet.google.com/sme-tpoy-wxt

Attendants

- Student - @ Adam Ryan
- Student - @ Finnian Rogers
- Student - @ Jane Slevin

Meeting Goals:

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

Discussion Topics

- Adam's Work in Progress + Blockers
- Finnian's Work in Progress + Blockers
- Jane's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Jane	Finnian
1	**29 Mar 2021**	Pandas function to summarise availability.	Adding containers to hold charts.	Panning map on click of marker. Opening info window on selection from dropdown.
2	**30 Mar 2021**	Pandas function to summarise availability.	Open info window on selection from dropdown.	Getting started with chartist.js
3	**31 Mar 2021**	Pandas function to summarise availability pushed.	Option to enter location and view nearest stations.	Getting started with chartist.js
4	**01 Apr 2021**	Reviewing queries/pandas functions as necessary for charts.	Option to enter location and view nearest stations.	Charting availability data.
5	**02 Apr 2021**	Sprint 4 preplanning. Creating Jira tasks.	Sprint 4 preplanning. Option to enter location and view nearest stations.	Sprint 4 preplanning. Working on charts.

Work in Progress

	Date	Adam	Jane	Finnian
1	**29 Mar 2021**	Pandas function to summarise availability.	Default view for side panel.	Get familiar with Chartist.js.
2	**30 Mar 2021**	Finishing off pandas function to summarise availability.	Default view for side panel.	Charting availability data.

3	**31 Mar 2021**	Reviewing queries/pandas functions as necessary for charts.	Option to enter location and view nearest stations.	Charting availability data.
4	**01 Apr 2021**	Reviewing queries/pandas functions as necessary for charts.	Option to enter location and view nearest stations.	Charting availability data.
5	**02 Apr 2021**	Machine learning model.	Option to enter location and view nearest stations.	Creating tables to hold data for machine learning.

Blockers

	Date	Adam	Jane	Finnian
1	**29 Mar 2021**	-	-	-
2	**30 Mar 2021**	-	-	-
3	**31 Mar 2021**	-	-	Completion of pandas functions.
4	**01 Apr 2021**	-	-	-
5	**02 Apr 2021**	-	-	-

Action items

- [ ] @ScrumMaster - Meet with X to discuss Ticket Y

Decisions

- The user journeys have been finalised

Daily Standup - WK10

Information	Values
Date:	13 Feb 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan @ Jane Slevin @ Finnian Rogers
Daily Standup Time:	08:45am - 09:00am
Agenda:	WIP
Link:	meet.google.com/sme-tpoy-wxt

Attendants

- Student - @ Adam Ryan
- Student - @ Finnian Rogers
- Student - @ Jane Slevin

Meeting Goals:

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

Discussion Topics

- Adam's Work in Progress + Blockers
- Finnian's Work in Progress + Blockers
- Jane's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Jane	Finnian
1	**05 Apr 2021**	(Bank holiday)	(Bank holiday)	(Bank holiday)
2	**06 Apr 2021**	Machine learning model.	Adding direction information to nearest stations.	Creating test/training tables and weather scraper.
3	**07 Apr 2021**	Machine learning model.	Adding direction information & fixing bugs in nearest station function.	Splitting test/training data by date.
4	**08 Apr 2021**	Creating routes for prediction data.	Tidying up front-end.	Front-end for prediction data.
5	**09 Apr 2021**	Optimising prediction models to reduce runtime. Creating route for prediction data.	Layout of side panel.	Adding selectors to choose time and date for prediction.
6	**10 Apr 2021**	Creating route for prediction model.	Creating divs for separate displays for side panel and functions to call divs.	Adding selectors to choose time and date for prediction. Adding loading spinner for charts.

Work in Progress

	Date	Adam	Jane	Finnian
1	**05 Apr 2021**	-	-	-
2	**06 Apr 2021**	Testing machine learning model, modifying to accept input.	Fixing bugs in nearest station function.	Creating test/training tables and weather scraper.
3	**07 Apr 2021**	Finalising machine learning model.	Adding direction information to nearest stations.	Splitting test/training data by date.
4	**08 Apr 2021**	Creating routes for prediction data.	Tidying up front-end.	Front-end for prediction data.
5	**09 Apr 2021**	Creating route for prediction data.	Default view for side-panel and CSS of side-panel.	Front-end for prediction data.
6	**10 Apr 2021**	Creating route for prediction data. Modifying route to accept input.	Default view for side-panel and CSS of side-panel.	Setting time and date range for prediction input (no past dates and no dates beyond 5 days in the future).

Blockers

	Date	Adam	Jane	Finnian
1	**05 Apr 2021**	-	-	-
2	**06 Apr 2021**	-	-	-
3	**07 Apr 2021**	-	-	-
4	**08 Apr 2021**	-	-	-
5	**09 Apr 2021**	-	-	-
6	**10 Apr 2021**	-	-	-

Action items

- [ ] @ScrumMaster - Meet with X to discuss Ticket Y

Decisions

- The user journeys have been finalised

Daily Standup - WK11

Information	Values
Date:	12 Apr 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan @ Jane Slevin @ Finnian Rogers
Daily Standup Time:	08:45am - 09:00am
Agenda:	WIP
Link:	meet.google.com/sme-tpoy-wxt

Attendants

- Student - @ Adam Ryan
- Student - Finnian
- Student - Jane

Meeting Goals:

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

Discussion Topics

- Adam's Work in Progress + Blockers
- Finnian's Work in Progress + Blockers
- Jane's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Jane	Finnian
1	**12 Apr 2021**	Initial deployment to EC2.	Initial deployment to EC2.	Initial deployment to EC2.
2	**13 Apr 2021**	Report writing, Sprint 1. Final adjustments to app.	Report writing, Sprints 3&4. Final adjustments to app.	Report writing, Sprint 2. Final adjustments to app.
3	**14 Apr 2021**	Final adjustments to app. No update on report, working on other modules.	Final adjustments to app. No update on report, working on other modules.	Report writing, Sprint 2. Final adjustments to app.
4	**15 Apr 2021**	Final adjustments to app. Report writing, Sprint 1 and data analytics.	Final adjustments to app. Report writing, Sprints 3&4.	Final adjustments to app. No update on report, working on other modules.
5	**16 Apr 2021**	Scheduling forecast scraper, refreshing models, programming fall-back SKLearn Linear Regression Model in case of XGBoost failure, filling About page. Data analytics and Sprint 1 for report.	Making front-end adjustments suggested by Product Owner. Design and Front-End Architecture for report.	Future Work and Project Overview for report.

Work in Progress

	Date	Adam	Jane	Finnian
1	**12 Apr 2021**	Begin report writing - covering data analytics and Sprint 1. Final adjustments to app.	Begin report writing - covering G Maps Distance Matrix and Sprints 3&4. Final adjustments to app.	Begin report writing - covering G Maps Charts and Sprint 2. Final adjustments to app.
2	**13 Apr 2021**	Report writing, Sprint 1 and data analytics. Final adjustments to app.	Report writing, Sprints 3&4. Final adjustments to app.	Report writing, Sprint 2. Final adjustments to app.
3	**14 Apr 2021**	Report writing, Sprint 1 and data analytics. Final adjustments to app.	Report writing, Sprints 3&4. Final adjustments to app.	Report writing, future work. Final adjustments to app.
4	**15 Apr 2021**	Report writing, data analytics. Final adjustments to app and final deployment.	Report writing, design. Final adjustments to app and final deployment.	Report writing, future work and project overview. Final adjustments to app and final deployment.
5	**16 Apr 2021**	Finish report writing.	Finish report writing.	Finish report writing.

Blockers

	Date	Adam	Jane	Finnian
1	**12 Apr 2021**	-	-	-
2	**13 Apr 2021**	-	-	-
3	**14 Apr 2021**	-	-	-
4	**15 Apr 2021**	-	-	-
5	**16 Apr 2021**	-	-	-

Action items

- [ ] @ScrumMaster - Meet with X to discuss Ticket Y

Decisions

- The user journeys have been finalised

Daily Standup - WK12

Information	Values
Date:	13 Feb 2021
Author:	@ Adam Ryan
Page Description:	Daily Standup Notes
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan @ Finnian Rogers @ Jane Slevin
Daily Standup Time:	08:45am - 09:00am
Agenda:	WIP
Link:	meet.google.com/sme-tpoy-wxt

Attendants

- Student - @ Adam Ryan
- Student - @ Jane Slevin

- Student - @ Finnian Rogers

Meeting Goals:

- Determine what team members are working on, development progress, blockers.
    - We should hopefully be nearly finished at this stage; if COVID is no longer

Discussion Topics

- Adam's Work in Progress + Blockers
- Finnian's Work in Progress + Blockers
- Jane's Work in Progress + Blockers

Discussion Notes

Yesterday:

	Date	Adam	Jane	Finnian
1	**19 Apr 2021**			
2	**20 Apr 2021**			
3	**21 Apr 2021**			
4	**22 Apr 2021**			
5	**23 Apr 2021**			

Work in Progress

	Date	Adam	Jane	Finnian
1	**19 Apr 2021**			
2	**20 Apr 2021**			
3	**21 Apr 2021**			
4	**22 Apr 2021**			
5	**23 Apr 2021**			

Blockers

	Date	Adam	Jane	Finnian
1	**19 Apr 2021**			
2	**20 Apr 2021**			
3	**21 Apr 2021**			
4	**22 Apr 2021**			
5	**23 Apr 2021**			

Action items

- [ ] @ScrumMaster - Meet with X to discuss Ticket Y

Decisions

- The user journeys have been finalised

Workshop

Information	Values
Date:	13 Feb 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:	8:45am - 9:00am
Agenda:	Workshop for Weather App
Meeting Link:	MeetingLinkGoesHere

Attendants

- Student - @ Adam Ryan
- Student - Finnian
- Student - Jane

Meeting Goals:

- What is the key purpose of the meeting.

Discussion Topics

- Work Complete
- Tickets Today
- Blockers?

Discussion Notes

- Work Complete
- Tickets Today
- Blockers?

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

- The user journeys have been finalised

Fill in retrospective details in the table below.

Fill in retrospective details in the table below.

Type /date to quickly add the date and @mention participants to add their details

Type /date to quickly add the date and @mention participants to help them find the page.

Sprint 1 Preplanning and Initial Introduction

Information	Values
Date:	13 Feb 2021
Author:	@ Adam Ryan
Page Description:	Documents the outcomes of our first discovery workshop.
Last Modified by:	@ Adam Ryan
People To Attend Daily Standup:	@ Adam Ryan @ Jane Slevin @ Finnian Rogers
Daily Standup Time:	10:00am - 12pm
Agenda:	Initial Workshop to Scope out Project
Link:	https://us02web.zoom.us/j/85793516043?pwd=MEFKckhGN2FJTUZnZTJEREwxWitvUT09

Attendants

- Student - @ Adam Ryan
- Student - @ Finnian Rogers
- Student - @ Jane Slevin

Meeting Goals:

- Review the project requirements, discuss broad timeline view and potential tasks involved, and determine project management structure.

Discussion Topics

- What sort of Project Requirements do we have?
- What software will we use to track and document our work?
- What sort of Git structure will we complete?
- What should we look to do in Sprint 1?
  - How should initial meetings be organised and when should we start?
- When should we look for setup to be complete?

Discussion Notes

- Within Sprint 1:
  - We want to have the Project Management Documentation Largely Setup and Complete.
  - We want to begin Scoping Requirements
  - We want to be set up on each platform.
  - We want to have started pulling and storing data.
- Tickets Today
- Blockers?

Sprint Goals

1. Establish the Jira Board and Confluence Page.

2. Establish the extractor and have it running daily.
   https://disappster.atlassian.net/browse/COMP30830-31
   https://disappster.atlassian.net/browse/COMP30830-42
   https://disappster.atlassian.net/browse/COMP30830-52
3. Set up the weather API
   https://disappster.atlassian.net/browse/COMP30830-66
   https://disappster.atlassian.net/browse/COMP30830-73
   https://disappster.atlassian.net/browse/COMP30830-79
4. Setup the various platforms which we need.

Duration

- Ending 21 Feb 2021
  - Revised Ending. 28 Feb 2021

Backlog

-Sprints with expended stuff

Action items

- [x] @ Adam Ryan - Set-up daily standup for 8:45 from Tuesday
- [x] @ Adam Ryan - Establish Jira and Confluence.
- [x] @ScrumMaster - Create Tickets for Actions

Decisions

- The user journeys have been finalised

Fill in retrospective details in the table below.

> Fill in retrospective details in the table below.
>
> Type /date to quickly add the date and @mention participants to add their details

Type /date to quickly add the date and @mention participants to help them find the page.

Workshop - WK05

Fill in workshop details in the table below.

> Fill in retrospective details in the table below.
>
> Type /date to quickly add the date and @mention participants to add their details

Information	Values
Date:	18 Feb 2021 09:00a, - 09:20am
Author:	@ Adam Ryan
Page Description	Weekly Product Owner Workshop
Last Modified by:	@ Adam Ryan
People in Sprint:	@ Adam Ryan @ karl.roe @ Jane Slevin @ Finnian Rogers

Background

Hi all.

These pages are designed to capture what

> For context on what these workshops should consist of, these are essentially the equivalent of Iteration Reviews: (https://www.atlassian.com/agile/scrum/ceremonies):
>
> If you are unfamiliar with the context of an interation review, please see the list here:
>
> *Iteration review*
>
> **Attendees:**
>
> **Required:** development team, scrum master, product owner
> **Optional:** project stakeholders
>
> **When:** At the end of a sprint or milestone.
>
> **Duration:** 30-60 minutes.
>
> **Agile Framework:** Scrum and kanban. Like planning, review for kanban teams should be aligned with team milestones rather than on a fixed cadence.
>
> **Purpose:** Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

Agenda:

As this is the first meeting, there has been no set agenda. Key items include:

- What is being worked on?
  - How is this being approached (e.g. distributed workload, shared, etc.)
- What is the overall management structure?
- How will IR reviews work going forward.
- Requirement gathering - if relevant.
- Overall sprint plan - expectations on MVP structure.

Discussion:

Brief overview of items discussed include:

- What is being worked on?
  - Extractor being worked on, only current item in sprint 1 (sprint 1 ending 21 Feb 2021), all working on this component for now. All team members completing individually.
- What is the overall management structure?
  - Demonstration of Jira board + Confluence.
- How will IR reviews work going forward.
  - Weekly meeting, largely set by team. Agenda to be set by team prior.
- Requirement gathering - if relevant.
  - Not covered. Broad expectation/note on ability to enter one location, enter a second location, travel there. ACTION RQ-01.
- Overall sprint plan - expectations on MVP structure .
  - Sprint 1: Extraction, Github, examine flask.
  - Sprint 2: Skeleton Flask app, basic front end, backend functional. MVP BY BEGINNING SPRINT 3!!!
  - Sprint 3: JS/React, Frontend feature added, advanced features e.g. ML focus.
  - Sprint 4: PROD PUSH!
    - GENERAL Note: Sprint structure given as 4 rather than 5 we have; expectations broadly should align.

Actions:

Actions include:

- [ ] Begin Setting up Sprint 2 Tasks on Jira commencing 22 Feb 2021 .
- [x] Continue using the Confluence/Jira Structure.
- [ ]

COMP30830 (@ Adam Ryan @ Finnian Rogers @ Jane Slevin) Going forward add in expected estimate onto owned Jira tasks to create burndown charts.

- [ ] Create requirements and tasks for next sprint.
- [ ] Set agenda for these meetings going forward (e.g. questions we have etc).

Please edit this page to capture anything missed.

Workshop - WK06

Fill in workshop details in the table below.

Fill in retrospective details in the table below.

Type /date to quickly add the date and @mention participants to add their details

Information	Values
Date:	25 Feb 2021 09:00a, - 09:20am
Author:	@ Adam Ryan
Page Description	Weekly Product Owner Workshop
Last Modified by:	@ Adam Ryan
People in Sprint:	@ Adam Ryan @ karl.roe @ Jane Slevin @ Finnian Rogers

Background

Hi all.

These pages are designed to capture what happens in our weekly workshops with the product owner.

For context on what these workshops should consist of, these are essentially the equivalent of Iteration Reviews: (https://www.atlassian.com/agile/scrum/ceremonies):

If you are unfamiliar with the context of an interation review, please see the list here:

Iteration review

Attendees:

Required: development team, scrum master, product owner
Optional: project stakeholders

When: At the end of a sprint or milestone.

Duration: 30-60 minutes.

Agile Framework: Scrum and kanban. Like planning, review for kanban teams should be aligned with team milestones rather than on a fixed cadence.

Purpose: Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

Agenda:

Agenda for the second meeting:

- What's our update on EC2 harvester?
- What's our update on the database?

- What's our update on the documentation?
- What should Sprint 2 Detail?
  - How is this being approached (e.g. distributed workload, shared, etc.)
- How are we doing relative to other groups?

Discussion:

- What's our update on EC2 harvester?
  - Mostly finished; JS, FR adding in weather.
- What's our update on the database?
  - Setup, data populated.
- What's our update on the documentation?
  - Data Model Added. Integration Added. Sprints recorded. Github setup complete.
- What should Sprint 2 Detail?
  - Flask focused.
    - Set up Flask and have skeleton flask app.
    - Sprint Meetings required - S1 retrospective and S2 preplanning.
    - Setup the backlog of tasks for S2 and S3.
    - Requirements should be detailed.
  - Frontend focused.
    - Fluid UI setup complete - Frontend Wireframe.
    - Decide a frontend JS Framework. REACT, Dash, JQuery, etc.
    - Decide a JS Map API and schema.
    - Tasks should be split and divvied.
  - Confirm Scrum Master
- How are we doing relative to other groups?
  - First meeting so hard to tell.
  - First indicators - S1 on schedule, documentation setup well.
  - Goal is to maintain.

Actions:

Actions include:

- [ ] COMP30830 (@ Adam Ryan @ Finnian Rogers @ Jane Slevin) to set up a Sprint 1 Retrospective and Sprint 2 Pre-Planning Meeting
  - [ ] Epic backlog to be created.
    - [ ] Tasks backlog to be created.
  - [ ] All to update burndown chart and add times to tasks - should be part of S2 pre-planning meeting.
  - [ ] Requirements to be detailed and added to confluence.
  - [ ] Burndown chart for S1 done and prepped for S2 Time to tasks must be updated daily to facilitate, either all of us in standups or EoD.
  - [ ] All should review DB integrity (e.g. still running. etc.)
- [ ] By Sprint 2 we should have
  - [ ] Frontend JS framework set.
  - [ ] One master database.
  - [ ] ETL process established and scoped.
  - [ ] Decided and began pulling JS API details for Map.
  - [ ] An answer to the question 'pull from API vs pull from DB' (AR Note: Typically you want to pull from DB for reliability and speed. REST API is slower but has advantage of being live. For DBBikes application where minimal risk of outdated data you don't really care. Similarly, because we're in a non-Production environment with no backing budget, API requests typically have max limit so more sensible for us ton control. All to formulate detailed answer).
  - [ ] Front End Requirements for end of sprint include:
    - [ ] A map showing station markers.
    - [ ] Station markets intractable (on_click() x).

- [ ] A dropdown list of all stations with some option to search and select (adv. feature non req)
- [ ] Overall Sprint 2 focus is FlaskApp + basic Frontend interacting with Backend.

Please edit this page to capture anything missed.

Note next week is last meeting with @ karl.roe until 25 Mar 2021 . Please prep. agenda as needed as S2 is a key sprint.

Workshop - WK07

Fill in workshop details in the table below.

Fill in retrospective details in the table below.

Type /date to quickly add the date and @mention participants to add their details

Information	Values
Date:	04 Mar 2021 09:00a, - 09:20am
Author:	@ Finnian Rogers
Page Description	Weekly Product Owner Workshop
Last Modified by:	@ Finnian Rogers
People in Sprint:	@ Adam Ryan @ karl.roe @ Jane Slevin @ Finnian Rogers

Background

Hi all.

These pages are designed to capture what happens in our weekly workshops with the product owner.

For context on what these workshops should consist of, these are essentially the equivalent of Iteration Reviews: (https://www.atlassian.com/agile/scrum/ceremonies):

If you are unfamiliar with the context of an interation review, please see the list here:

Iteration review

Attendees:

Required: development team, scrum master, product owner
Optional: project stakeholders

When: At the end of a sprint or milestone.

Duration: 30-60 minutes.

Agile Framework: Scrum and kanban. Like planning, review for kanban teams should be aligned with team milestones rather than on a fixed cadence.

Purpose: Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

Agenda:

Agenda for the third meeting:

- What's our update on EC2 harvester?
- What's our update on the database?
- What's our update on the flask app?

- What's our update on the front end map?
- What's our update on the documentation?
- What should Sprint 2 and the first half of Sprint 3 Detail?
    - How is this being approached (e.g. distributed workload, shared, etc.)

Discussion:

- What's our update on EC2 harvester?
    - All running Adam's harvester and storing in our repective databases
- What's our update on the documentation?
    - Data Model Added. Integration Added. Sprints recorded. Github setup complete.
    - Backlog still needs to be done
- What should Sprint 2 Detail?
    - Flask focused.
        - Set up Flask and have skeleton flask app running locally
        - Pulling relevant data from database to respond to front end queries
    - Frontend focused.
        - Modify wireframe so map takes up less space and to leave space for visualisations and other features (requested by product owner)
        - Decide a frontend JS Framework. REACT, Dash, JQuery, etc.
        - Create basic User Interface with Google Maps API
        - Display clickable markers for all stations on the map
        - Basic data about that station should be displayed on click
        - Dropdown for station names that's linked to the markers
        - (Optional for this sprint) Display the most recent update for a clicked marker by querying the availability table
        - Tasks should be split and divvied.
- What should the first half of Sprint 3 Detail?
    - Frontend focused.
        - Build out the UI with more structure and detail
        - Create weekly and hourly visualisations for each station

Actions:

Actions include:

- [ ] Burndown chart for S1 done and prepped for S2 Time to tasks must be updated daily to facilitate, either all of us in standups or EoD.
- [ ] By Sprint 2 we should have
    - [ ] Frontend JS framework set (if any).
    - [ ] One master database.
    - [ ] ETL process established and scoped.
    - [ ] Decided and began pulling JS API details for Map.
    - [ ] An answer to the question 'pull from API vs pull from DB' (AR Note: Typically you want to pull from DB for reliability and speed. REST API is slower but has advantage of being live. For DBBikes application where minimal risk of outdated data you don't really care. Similarly, because we're in a non-Production environment with no backing budget, API requests typically have max limit so more sensible for us ton control. All to formulate detailed answer).
    - [ ] Front End Requirements for end of sprint include:
        - [ ] A map showing station markers.
        - [ ] Station markets intractable (on_click() x).
        - [ ] A dropdown list of all stations with some option to search and select (adv. feature non req)
    - [ ] Overall Sprint 2 focus is FlaskApp + basic Frontend interacting with Backend.
- [ ] By end of midterm we should have
    - [ ] A structured and interactive user interface displaying relevant information
    - [ ] Display visualisations for availability in each station

Please edit this page to capture anything missed.

Sprint 2 Retrospective, Sprint 3 Planning - WK07.2

Fill in workshop details in the table below.

Fill in retrospective details in the table below.

Type /date to quickly add the date and @mention participants to add their details

Information	Values
Date:	18 Mar 2021 12pm-1:30pm
Author:	@ Jane Slevin
Page Description	Weekly Product Owner Workshop
Last Modified by:	@ Jane Slevin
People in Sprint:	@ Adam Ryan @ Finnian Rogers @ Jane Slevin

Background

Hi all.

These pages are designed to capture what happens in our weekly workshops with the product owner.

For context on what these workshops should consist of, these are essentially the equivalent of Iteration Reviews: (https://www.atlassian.com/agile/scrum/ceremonies):

If you are unfamiliar with the context of an interation review, please see the list here:

Iteration review

Attendees:

Required: development team, scrum master, product owner
Optional: project stakeholders

When: At the end of a sprint or milestone.

Duration: 30-60 minutes.

Agile Framework: Scrum and kanban. Like planning, review for kanban teams should be aligned with team milestones rather than on a fixed cadence.

Purpose: Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

Agenda:

Agenda for sprint 2 retrospective and sprint 3 pre-planning:

- Progress with Sprint 2 tasks?
- Sprint 2 backlog?
- Reflection on Sprint 2 - what went well, what could be improved?
- Sprint 3 planning - tasks to be completed?
- How should work be divided for Sprint 3?

Discussion:

- Progress with Sprint 2 tasks:

        - All running flask app locally.
            - Have routes to return station data from database.
            - Started work on routes to return availability/most recent update data.
        - All working on front-end individually. So far:
            - Displaying map on homepage (index.html) using Google Maps API.
            - Displaying stations as clickable markers on map and in dropdown menu.
            - Station selection in dropdown linked to markers.
            - Displaying station name on click of marker.
        - Continuing data collection (bike and weather) using Adam's harvester. Approximately 2-3 weeks worth of data collected.
        - Wireframe updated following meeting with product owner.
            - Less map-centric.
            - Charts (detailing availability stats) displayed in side panel.
- Sprint 2 backlog:
    - Pull availability/last update data.
    - Display basic data i.e. station name and up-to-date availability on click of marker.
    - Create visualisations for hourly/daily/weekly (?) availability data.
    - Sprint 2 burndown chart.
- Reflection on Sprint 2.
    - Greater focus on Confluence over Jira.
    - Continuing to work on individual applications, sharing code/ideas through GitHub.
- Sprint 3 planning:
    - To have started/completed before next week's meeting with Product Owner:
        - Create visualisations to display availability data.
            - Pull historical availability and weather data.
            - Create 3 visualisations for hourly, daily, weekly (?) availability data.
            - Option to filter availability based on weather conditions.
        - Continue to build out UI with more structure and detail.
            - Request and display most recent update per station.
            - Close pop-up for first station when second station marker is clicked.
            - Colour-code markers e.g. green for >60% availability, yellow for 30-60%, red for <30%.
            - Display number of available bikes or number of available stands on marker? Filter/radio buttons to choose which is displayed?
            - Display visualisations in side panel - how to render?
            - Display summary information in side panel before a station is chosen e.g. average availability for each station.
        - Investigate best method of loading data into front end (load all JSON data on page load or on events? read from CSV?). App currently loading very slowly. Contact Karl for advice.
    - Going forward/second half of sprint 3:
        - Machine learning, weather/availability forecasting.
- Sprint 3 division of work:
    - Jane to replace Finnian as scrum-master.
    - All continuing to work on front end individually.
    - Will merge flask when complete/optimised.

Actions for Sprint 3:

Actions include:

- [ ] Complete burndown for Sprint 2 and prepare for Sprint 3 estimated time/time spent for tasks must be updated to facilitate.
- [ ] Tasks for coming week:
    - [ ] Create visualisations to display availability data.
        - [ ] Pull historical availability and weather data
        - [ ] Investigate methods for creating visualisations.
        - [ ] Create 3 visualisations for hourly, daily, weekly (?) availability data.
        - [ ] Create option to filter availability data based on weather.
    - [ ] Continue to build out UI with more structure and detail.
        - [ ] Request and display data regarding most recent update per station (marker on_click, see wireframe).
        - [ ]

- [ ] Include function to close pop-up for first station when second station marker is clicked.
- [ ] Colour-code markers e.g. green for 60% availability, yellow for 30-60%, red for <30% (non req).
- [ ] Display number of available bikes or number of available stands on marker (non req)? Filter/radio buttons to choose which is displayed?
- [ ] Display visualisations in side panel.
- [ ] Display summary information in side panel before a station is chosen e.g. average availability for each station.
- [ ] Investigate best method of loading data into front end (load all JSON data on page load or on events? read from CSV?). App currently loading very slowly. Contact Karl for advice.

- [ ] Machine learning, weather/availability forecasting.
- [ ] By end of Sprint 3 we should have:
  - [ ] Visualisations for hourly/daily/weekly availability for each station
  - [ ] Interactive UI with clickable markers - most recent update displayed on click, visualisations updated on click.
  - [ ] Weather and availability forecast information.

Please edit this page to capture anything missed.

Workshop - WK08

Fill in workshop details in the table below.

Fill in retrospective details in the table below.

Type /date to quickly add the date and @mention participants to add their details

Information	Values
Date:	25 Mar 2021 09:00a, - 09:20am
Author:	@ Jane Slevin
Page Description	Weekly Product Owner Workshop
Last Modified by:	@ Jane Slevin
People in Sprint:	@ Adam Ryan @ karl.roe @ Jane Slevin @ Finnian Rogers

Background

Hi all.

These pages are designed to capture what happens in our weekly workshops with the product owner.

For context on what these workshops should consist of, these are essentially the equivalent of Iteration Reviews: (https://www.atlassian.com/agile/scrum/ceremonies):

If you are unfamiliar with the context of an interation review, please see the list here:

Iteration review

Attendees:

Required: development team, scrum master, product owner
Optional: project stakeholders

When: At the end of a sprint or milestone.

Duration: 30-60 minutes.

Agile Framework: Scrum and kanban. Like planning, review for kanban teams should be aligned with team milestones rather than on a fixed cadence.

Purpose: Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

Agenda:

Agenda for the third meeting:

- Update on database queries?
- Update on front end?
- Update on visualisations?
- Update on the documentation/management?
- Targets for the remainder of Sprint 3?
    - How is this being approached (e.g. distributed workload, shared, etc.)
- Questions for product owner:
    - Best approach to working with shared codebase on GitHub.
    - Charting library recommendations.
    - Preferences regarding frontend i.e. positioning of chart container, marker styling etc.

Discussion:

- Update on database queries?
    - sql.py file created to hold codebase for all SQL database queries required.
    - All SQL queries required to date and envisaged to be required contained within sql.py (availabilty_last_update, weather_by_station_number etc.)
- Update on front end?
    - Displaying map on homepage using Google Maps API
    - Currently full page map with small space for header and footer, dropdown menu contained in header
    - Dublin Bikes stations displayed on map as clickable markers
        - Option of colour-coding markers based on percentage availability or displaying number of bikes/stands available as label on marker - product owner preference?
        - On-click of marker pop-up info-window displaying:
            - Station name
            - Number of bikes currently available
            - Number of bike stands currently available
            - Time/date of last update
        - Info-window for first station hidden on-click of another marker
    - List of Dublin Bikes stations displayed in dropdown menu. When station is selected from the dropdown all map markers apart from that for the selected station are hidden.
    - Wireframe updated to include container for graphs.
- Update on visualisations?
    - Work on pandas functions to summarise data retrieved in progress.
    - Yet to decide on JavaScript charting library.

        - Yet to create container to display graphs (frontend). Second webpage created to use for testing chart display in the meantime.
- Update on the documentation/management?
    - Sprint 2 restrospective and sprint 3 preplanning held on March 18th.
        - Sprint 2 closed off.
        - Tasks for sprint 3 decided and created.
        - Burndowns need to be done.
    - Code merge completed. Worked independently for Sprints 1 and 2 so each team member had understanding of how key elements work. Dividing tasks going forward (Jira tickets). Working on shared codebase on GitHub.
- Targets for the remainder of Sprint 3.
    - Continue to build out UI with more structure and detail.
        - Add section for visualisations.
        - Move dropdown to visualisation section. Visualisations (hourly, daily, weekly availability) will be displayed for selected station.
        - Decide on default view for visualisation section (potentially hidden until station is selected?).
    - Add visualisations.
- AOB
    - Current approach to code sharing/editing on GitHub ok.
    - Product owner no preference as to location of container for charts. Prefers colour-coded markers. Potentially add radio button to filter by available bikes vs available bike stands.
    - Recommended Chartist charting library.

Actions:

Actions include:

- [ ] Burndown chart for Sprint 2
- [ ] By the end of Sprint 3 we should have:
    - [ ] Container for charts/visualisations on homepage.
    - [ ] Move dropdown selector to visualisation container.
    - [ ] Render visualisations (hourly, daily, weekly availability data).
    - [ ] JavaScript function to update visualisations displayed based on station selection from dropdown.
    - [ ] Decide on default view for visualisation container or function to display container only when station has been selected.
    - [ ] JavaScript function to filter coloured markers to show available bikes or available stations.
    - [ ] JavaScript function to open info-window when station is station is selected from dropdown.
- [ ] Focus for remainder of Sprint 3 is displaying visualisations and creating nice user interface.

Please edit this page to capture anything missed.

Workshop - WK09

Fill in workshop details in the table below.

> Fill in retrospective details in the table below.
>
> Type /date to quickly add the date and @mention participants to add their details

Information	Values
Date:	01 Apr 2021 09:00a, - 09:20am
Author:	@ Jane Slevin
Page Description	Weekly Product Owner Workshop
Last Modified by:	@ Jane Slevin
People in Sprint:	@ Adam Ryan @ karl.roe @ Jane Slevin @ Finnian Rogers

Background

Hi all.

These pages are designed to capture what happens in our weekly workshops with the product owner.

For context on what these workshops should consist of, these are essentially the equivalent of Iteration Reviews: (https://www.atlassian.com/agile/scrum/ceremonies):

If you are unfamiliar with the context of an interation review, please see the list here:

Iteration review

Attendees:

Required: development team, scrum master, product owner
Optional: project stakeholders

When: At the end of a sprint or milestone.

Duration: 30-60 minutes.

Agile Framework: Scrum and kanban. Like planning, review for kanban teams should be aligned with team milestones rather than on a fixed cadence.

Purpose: Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

Agenda:

Agenda for the third meeting:

- What's our update on the front end map?
- What's our update on displaying visualisations?
- What's our update on additional features?
- What are our targets for the end of Sprint 3?
- Targets for Sprint 4?

Discussion:

- What's our update on the front end map?
  - Markers colour-coded by availability of bikes (red for 0% available, yellow for 0-25%, green for >25%). Can be filtered to display only red/yellow/green.
  - Side panel added to hold visualisations, availability predictions and any additional features. Contents of side panel will be replaced based on selection and input, rather than adding additional webpages. See wireframe.
  - Info-window for relevant marker opens on selection of station from dropdown.
  - Issues:
    - Overlap of station selector and graph div - CSS to be modified.
- What's our update on displaying visualisations?
  - Charts displaying average hourly availability and average daily availability per station displayed in side panel on selection of station from dropdown/selection of station marker on map.
  - Issues:
    - Availability queries optimised but loading time for charts is still slow. Should be some speed-up on deployment to EC2. Option: process making queries running in the background and loading data to JSON file at regular intervals (probably won't do this). Add loading symbol.
    - Duplicates in dropdown - need to be removed.
- What's our update on additional features?
  - Option to enter a location and be presented with nearest stations in progress, aim to have completed by the end of Sprint 3.
- What are our targets for the end of Sprint 3?
  - Mostly on track.
  - Option to view nearest stations.
  - Polish charts - add labels etc.

- Targets for Sprint 4?
  - App:
    - Machine learning model built in.
    - Optimise performance (as mentioned above, process continually making queries?).
    - Complete, polished UI.
    - Deploy to EC2.
    - Optional additional features:
      - Filter markers by colour representing availabilty. Feature added. (Add further filter to switch to bike stand availabilty.)
      - Storing last station chosen. Use local storage to negate GDPR issues - all information stored on client side, no location information sent to server.
      - Customise google maps appearance.
      - Geolocation. Adding feature to allow user to enter location - take this further?
      - Responsive design. (App is to be used on laptop, does not need to be suitable for mobile use.)
      - Directions API. Provide user with directions to a station based on whether they want to use a bike or return a bike?
      - CO2 emission calculation per journey i.e. “by making this journey by bike you have reduced your CO2 emissions by 80%!”.
      - Weather warnings. Possibly incorporate into default view.
  - Report:
    - Intro, background, scope.
    - Details for each sprint:
      - Targets, management, division of tasks.
      - Burndown and retrospective.
      - Preplanning.
    - Access to Jira included or provided on email request. Generated report included in appendix.
    - GitHub public, provide link.
    - Link to EC2

Actions:

Actions include:

- [ ] Remainder of Sprint 3:
  - [ ] Explore machine learning models (lecture at 4pm today).
  - [ ] Modify CSS to fit graphs.
  - [ ] Remove duplicates in dropdown menu.
  - [ ] Add feature to view nearest stations.
  - [ ] Add labels etc. to charts.
- [ ] Sprint 4 preplanning, Sprint 3 retrospective.
  - [ ] Decide which additional features we want to add.
  - [ ] Complete burndown charts.
- [ ] Priorities:
  - [ ] Polished UI.
  - [ ] Deployment to EC2.

Please edit this page to capture anything missed.

Sprint 3 Retrospective, Sprint 4 Planning - WK09

Fill in workshop details in the table below.

Fill in retrospective details in the table below.

Type /date to quickly add the date and @mention participants to add their details

Information	Values
Date:	01 Apr 2021 5pm-5.45pm
Author:	@ Jane Slevin
Page Description	Weekly Product Owner Workshop
Last Modified by:	@ Jane Slevin
People in Sprint:	@ Adam Ryan @ Finnian Rogers @ Jane Slevin

Background

Hi all.

These pages are designed to capture what happens in our sprint retrospective and preplanning meetings.

For context on what these workshops should consist of, these are essentially the equivalent of Iteration Reviews: (https://www.atlassian.com/agile/scrum/ceremonies):

If you are unfamiliar with the context of an interation review, please see the list here:

Iteration review

Attendees:

Required: development team, scrum master, product owner
Optional: project stakeholders

When: At the end of a sprint or milestone.

Duration: 30-60 minutes.

Agile Framework: Scrum and kanban. Like planning, review for kanban teams should be aligned with team milestones rather than on a fixed cadence.

Purpose: Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

Agenda:

Agenda for sprint 3 retrospective and sprint 4 pre-planning:

- Progress with Sprint 3 tasks?
- Sprint 3 backlog?
- Reflection on Sprint 3 - what went well, what could be improved?
- Sprint 4 planning - tasks to be completed?
- How should work be divided for Sprint 4?

Discussion:

This meeting is a follow-up to our product owner workshop earlier today.

- Progress with Sprint 3 tasks:
    - Markers colour-coded by availability of bikes (red for 0% available, yellow for 0-25%, green for >25%). Can be filtered to display only red/yellow/green.
    - Info-window for relevant marker opens on selection of station from dropdown.
    - Side panel added to hold visualisations, availability predictions and any additional features. Contents of side panel will be replaced based on selection and input, rather than adding additional webpages. See wireframe.
    - Charts displaying average hourly availability and average daily availability per station displayed in side panel on selection of station from dropdown/selection of station marker on map.

        - Issues:
            - Availability queries optimised but loading time for charts is still slow. Should be some speed-up on deployment to EC2. Option: process making queries running in the background and loading data to JSON file at regular intervals (probably won't do this). Add loading symbol.
            - Duplicates in dropdown - need to be removed.
            - Overlap of station selector and graph div - CSS to be modified.
    - Outstanding Sprint 3 backlog:
        - Option to enter a location and be presented with nearest stations.
        - Ability to filter colouring of markers to reflect available bike stands instead of available bikes.
        - Front end bugs mentioned above.
    - Reflection on Sprint 3:
        - Better division of workload. See DevOps.
        - Better use of Jira board tickets.
    - Sprint 4 planning:
        - Choose a machine learning model (investigate XGBoost).
        - Gather data required to build prediction model and display predictions.
            - New tables to contain relevant data returned from database queries.
            - Collection and storage of weather-forecast data (API).
        - Integrate machine learning model. We want the user to be able to enter a date and time and view availability information:
            - Choose station and view predicted availability for chosen station?
            - Enter location and view nearest stations that are predicted to have bikes available?
        - Modify UI to create space for input of time and date/display of prediction information.
        - Fix front end bugs listed above.
        - Build out and customise UI.
            - Add loading symbol if performance cannot be improved.
        - Additional features:
            - Considering geolocation. Prioritising feature to manually enter location and view nearest stations (in development) and work on prediction model.
            - Explore the addition of direction information when displaying nearest stations.
        - Add unit tests.
        - Review and remove redundant code. Ensure code is modularised.
        - Documentation:
            - Document all design decisions on Confluence.
            - Ensure all sprints are documented on Confluence, including records of daily stand-ups, product owner workshops, sprint retrospective and preplanning meetings, any other meetings (e.g. code merge).
            - Ensure methods, data flows, routes, tables are documented on Confluence.
            - Ensure burndown charts are completed and documented on Clonfluence.
    - Division of work for Sprint 4:
        - As before, the aim is for larger tasks integral to the system to be completed by all team members and follow-on tasks to be divided through the use of Jira tickets. Approach will be reviewed pending further understanding of what is involved in development of machine learning model.

Actions for coming week:

Actions include:

- [ ] Choose machine learning model. Develop model for each station. Integrate into flask app.
- [ ] Develop tables to hold:
    - [ ] Training data.
    - [ ] Test data.
    - [ ] Weather forecast data.
    - [ ] User input.
- [ ] Develop method to call model on user input and return prediction results. Aim is to provide prediction data for nearest stations based on location input but as a start we will provide prediction data for a chosen station.
- [ ] Complete and add in feature to show nearest stations on location entry. Explore addition of direction information.
- [ ] Build out UI.
    - [ ] Input fields for user to select location, date and time for prediction.

- [ ] Area to display prediction results.
- [ ] Default view for side-panel.

Please edit this page to capture anything missed.

Workshop - WK10

Fill in workshop details in the table below.

> Fill in retrospective details in the table below.
>
> Type /date to quickly add the date and @mention participants to add their details

Information	Values
Date:	08 Apr 2021 09:00a, - 09:20am
Author:	@ Jane Slevin
Page Description	Weekly Product Owner Workshop
Last Modified by:	@ Jane Slevin
People in Sprint:	@ Adam Ryan @ karl.roe @ Jane Slevin @ Finnian Rogers

Background

Hi all.

These pages are designed to capture what happens in our weekly workshops with the product owner.

> For context on what these workshops should consist of, these are essentially the equivalent of Iteration Reviews: (https://www.atlassian.com/agile/scrum/ceremonies):
>
> If you are unfamiliar with the context of an interation review, please see the list here:
>
> *Iteration review*
>
> **Attendees:**
>
> **Required:** development team, scrum master, product owner
> **Optional:** project stakeholders
>
> **When:** At the end of a sprint or milestone.
>
> **Duration:** 30-60 minutes.
>
> **Agile Framework:** Scrum and kanban. Like planning, review for kanban teams should be aligned with team milestones rather than on a fixed cadence.
>
> **Purpose:** Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

Agenda:

Agenda for the third meeting:

- What's our update on the flask app?
- What's our update on the prediction model?
- What's our update on the documentation?

- What's our plan for the final week of the project?

Discussion:

- What's our update on the flask app?
  - Original features complete - filter markers by availability (bikes only), select marker on map to view info window with current availability, select station from dropdown to view on map and open info window.
  - Additional features: enter a location to see 5 nearest stations (does not take availability into account), select a station from list to view on map and open info window, select "get directions" to get directions to that station from current location (transport mode set to walking).
  - Priority now:
    - Fix bugs.
    - Polish UI.
    - Build out front-end to display prediction information.
    - Remove redundant code.
    - Ensure code is modularised.
    - Ensure code is appropriately commented.
- What's our update on the prediction model?
  - Prediction model created for each station using XGBoost regression model, yet to be integrated.
  - Predictions based on historical availability for time of day/day of week/day of month and the following weather characteristics: weather category, temperature, "feels like" temperature, humidity and air pressure.
  - Training/test data split based on created date i.e. first 70% of the data used for training and latter 30% for testing, to represent the manner in which the app will function i.e. using a database of historical data to make predictions for a future date.
  - Database tables created to hold training and test data.
  - Scraper for weather forecast data built, yet to be integrated. Intention is to run once daily.
- What's our update on the documentation?
  - Records of meetings (stand-up, scrum, planning and retrospective) up-to-date.
  - Additional documentation largely up-to-date. Methods, routes, mock-ups, design-choices etc must be finalised.
- What's our plan for the final week of the project?
  - Application:
    - Integration of prediction models - create routes, build out how prediction results appear on front-end.
    - Integration of weather forecast scraper. Schedule to run at intervals.
    - Finalising/polishing UI prioritised over addition of other extra/non-required features.
    - Move to classroom EC2 (conditions of use of starter accounts changing on April 14th).
    - Deploy to EC2. Domain name not required (not included in free tier) - expose port so application is remotely accessible for grading.
    - Soft deadline for finalised application set for Sunday 11th April - then deployment.
  - Report:
    - Documentation up-to-date on Confluence.
    - Product owner suggested structure for report:
      - Introduction, scope, target.
      - Links to EC2, GitHub, Jira/Confluence etc.
      - Individual contribution/division of work.
      - Discuss each Sprint.
        - Sprint 1: initial scrum, developing backlog, initial mock-up, progress/what tasks were completed, daily stand-ups, burndown and discussion.
        - Sprints 2&3: as above, including retrospective for previous sprints.
        - Sprint 4: as above, including discussion of data analytics.
      - "Tell story" of project.
      - Possible additional sections: architecture, functionality, design (mock-ups), future work.
      - Approx. 15 page report (guideline).
  - Personal Note

Actions:

Actions include:

- [ ] Priorities:
  - [ ] Clean up UI.
    - [ ] Fix bugs.

- [ ] Appropriate divs, to open/close on input/selection of options.
- [ ] Integrate prediction models.
  - [ ] Create routes for prediction data.
  - [ ] Build out front-end display of predictions.
- [ ] Remove redundant code.
- [ ] Ensure code is modularised.
- [ ] Deploy application to EC2. Planned deployment date Monday 12th.
- [ ] Gather and complete documentation.
  - [ ] Ensure all tables and integrations keys are documented.
  - [ ] Ensure all design decisions are documented.
  - [ ] Ensure all Workshops are documented.
  - [ ] Ensure all Sprints including Retrospective and Pre-Planning meetings are documented.
  - [ ] Ensure the wireframe is documented.
  - [ ] Ensure the GitKracken timeline is documented.
  - [ ] Ensure all data flows are documented.
  - [ ] Ensure all methods are documented.
  - [ ] Ensure all routes are documented.
  - [ ] Ensure burndown reports are documented.
- [ ] Write report. PDF document. Include link to GitHub (ensure access enabled) and EC2 instance
- [ ] Submit 😱

Please edit this page to capture anything missed.

Workshop - WK11

Fill in workshop details in the table below.

> Fill in retrospective details in the table below.
>
> Type /date to quickly add the date and @mention participants to add their details

Information	Values
Date:	15 Apr 2021 09:00a, - 09:20am
Author:	@ Jane Slevin
Page Description	Weekly Product Owner Workshop
Last Modified by:	@ Jane Slevin
People in Sprint:	@ Adam Ryan @ karl.roe @ Jane Slevin @ Finnian Rogers

Background

Hi all.

These pages are designed to capture what happens in our weekly workshops with the product owner.

For context on what these workshops should consist of, these are essentially the equivalent of Iteration Reviews: (https://www.atlassian.com/agile/scrum/ceremonies):

If you are unfamiliar with the context of an interation review, please see the list here:

Iteration review

Attendees:

Required: development team, scrum master, product owner
Optional: project stakeholders

When: At the end of a sprint or milestone.

Duration: 30-60 minutes.

Agile Framework: Scrum and kanban. Like planning, review for kanban teams should be aligned with team milestones rather than on a fixed cadence.

Purpose: Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

Agenda:

- What's our final update on the web application?
- What's our update on the accompanying report?
- What do we have left to do by the deadline tomorrow?
- Any required features we have neglected to include?

Discussion:

- What's our final update on the web application?
    - Web application 99% complete.
        - Prediction models integrated.
        - Weather scraper integrated.
        - UI finalised.
        - Test deployment successful. Intend to complete final deployment later today after some last minute adjustments e.g. cleaning up code.
- Report writing in progress.
    - Each team member writing about the sprints for which they were Scrum Master.
    - Each team member writing about the areas in which they were most involved e.g. creating charts, developing prediction models etc.
    - Remaining sections divided between team members.
    - Documentation (Jira and Confluence) 99% complete. Final review today.
- What do we have left to do by the deadline tomorrow?
    - Main task: complete the report.
    - Make any final adjustments to web application and complete final deployment.
    - Review documentation to ensure all areas are up-to-date.
- Any required features we have neglected to include?
    - Product Owner happy with web application overall. All required functionality included. UI easy to navigate.
    - Some minor adjustments suggested, to consider if time allows:
        - Change background colour of default side-panel view, currently difficult to see weather icon - poor accessibility.
        - Add whitespace between stations in nearest stations list.
        - Move nearest station "View on Map" option to same line as "Get Directions" option, style as buttons.
        - Center station selector and label in Availability tab
        - Vertically center "Stations" and "About Disappster" in main navbar.

Actions:

Priorities:

- [ ] Complete report.
- [ ] Complete documentation.
- [ ] Final deployment of web application.
- [ ] Submit project.

Please edit this page to capture anything missed.

Sprint 4 Retrospective and Project End

Fill in workshop details in the table below.

> Fill in retrospective details in the table below.
>
> Type /date to quickly add the date and @mention participants to add their details

Information	Values
Date:	16 Apr 2021 1pm-2pm
Author:	@ Adam Ryan
Page Description	End of Project
Last Modified by:	@ Adam Ryan
People in Sprint:	@ Adam Ryan @ Finnian Rogers @ Jane Slevin

Background

Hi all.

These pages are designed to capture what happens in our sprint retrospective and preplanning meetings.

> For context on what these workshops should consist of, these are essentially the equivalent of Iteration Reviews: (https://www.atlassian.com/agile/scrum/ceremonies):
>
> If you are unfamiliar with the context of an interation review, please see the list here:
>
> *Iteration review*
>
> **Attendees:**
>
> **Required:** development team, scrum master, product owner
> **Optional:** project stakeholders
>
> **When:** At the end of a sprint or milestone.
>
> **Duration:** 30-60 minutes.
>
> **Agile Framework:** Scrum and kanban. Like planning, review for kanban teams should be aligned with team milestones rather than on a fixed cadence.
>
> **Purpose:** Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

Agenda:

Agenda for sprint 4 retrospective:

- Sprint 4 Progress
  - Report Progress

- Application Final Testing

Discussion:

This meeting is a conclusion of the project and is designed to prepare for the final project submission.

- Report - What is the final progress?
- Application - Any final changes which are needed ahead of the application deployment.

Future Items:

What needs to be moved to Disappster Phase 2?

System Design

This is to document the System Design

Flask App

This section details the Flask App Structure and Details on What is Present

App Folder Structure

This page details the folder structure of the Flask App

	Primary Folder	File	Primary Codebase	Description
1	static	javascript.js	Javascript	Javascript Functions
2	static	stylesheet.css	CSS	Stylesheet
3	models	varied	pickle and xgboost. save_model.	ML Models
4	templates	about.html	HTML	About Page - About This App
5	templates	index.html	HTML	Station Page
6	FlaskApp	__init__.py	Python	Initialiser
7	FlaskApp	config.py	Python	App Config
8	FlaskApp	data_dictionary.py	Python	Database dictionary, Database connection details, API services
9	FlaskApp	sql.py	Python	APP SQL Queries
10	FlaskApp	tests.py	Python	Test Statements
11	FlaskApp	methods.py	Python	App Methods
12	FlaskApp	app.py	Python	App Routes

Import Structure

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

- App.py
    - External Libraries (limited)
    - Data Dictionary
    - Routes
        - Methods
            - Data Dictionary
            - SQL
            - External Libraries
    - Methods
        - Data Dictionary
        - SQL
        - External Libraries
    - Unit Tests
        - Methods.
            - Data Dictionary

- SQL
- External Libraries

Due to the folder and import structure, there is no risk of circular import impact app running.

The largest risk of circular imports exists within the importing of the unit tests into app.py however due to the placement of the import at the end of the routes and before main, this route is closed.

Routes

This page details the routes in the app.

This page details the folder structure of the Flask App

	Route	RouteName	Methods
1	home, index, (blank)	`home`	`station_availability_last_update_table_df`
2	debug	debug_hello	none
3	`station`	`station`	`station_availability_last_update_table_df`
4	`availability`	`availability_request`	`station_availability_last_update_table_df`
5	`current_weather`	`current_weather_request`	`weather_last_update_df`
6	`station_availability_stat_by_weekdayno`	`station_availability_stat_by_weekdayno`	`avg_station_availability_by_weekdayno_df`
7	`station_availability_stat_by_date`	`station_availability_stat_by_date`	`avg_station_availability_by_date_df`
8	`station_availability_stat_by_monthno`	`station_availability_stat_by_monthno`	`avg_station_availability_by_monthno_df`
9	`station_availability_stat_by_hourno`	`station_availability_stat_by_hourno`	`avg_station_availability_by_hourno_df`
10	`station_availability_stat_by_weekno`	`station_availability_stat_by_weekno`	`avg_station_availability_by_weekno_df`
11	`/single_station_availability_stat_by_date/<no>`	`single_station_availability_stat_by_date`	`avg_station_availability_by_date_df_forstat`
12	`/single_station_availability_stat_by_monthno/<no>`	`single_station_availability_stat_by_monthno`	`avg_station_availability_by_monthno_df_forstat`
13	`/single_station_availability_stat_by_hourno/<no>`	`single_station_availability_stat_by_hourno`	`avg_station_availability_by_hourno_df_forstat`
14	`single_station_availability_stat_by_weekno/<no>`	`single_station_availability_stat_by_weekno`	`avg_station_availability_by_weekno_df_forstat`
15	`/metric/<metric_type>/<station_no>`	`metric_type_average_avail_for_station`	(all time avg methods)

16	`/test_model/<requested_time>/<no>`	`getPrediction`	`get_forecast_for_time`
			
			`predict_from_station_time`

Tests

This page details the tests which were implemented. Due to time constraints and focusing on key features within the app, the tests are limited to just a database connection test. However, many of the methods contain detail error dictionaries and conditional statements designed to compensate for having failures.

One failing which would need to be fixed in a future iteration of the app is that we should have default data in the instance that the database connection fails to be established so that the app does not completely crash.

	Test Function	Tested Method	Pass Condition	Fail Outcome
1	`connection_test`	`connect_db_engine`	`Database Conenction Established`	Database connection method returns error.

2	`run_tests`	`All runnable tests`	`All tests True`	Flask App Fails to Run

DevOps

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

DevOps Process

For Sprint 1 and 2, following the instruction that we should all be involved in various components of the application, we initially elected to develop the early stages of the app individually so that we were all familiar with the processes and challenges involved.

The net result of Sprint 1 was the completion of a data harvester added into our Github repository, and the net result of Sprint 2 is we all had Flask Apps complete (up to end of Sprint 2 goals).

Following the end of Sprint 2, as aspects of the application became more specialised and following the understanding of how Flask Apps are built we conducted a code merge of our applications which commenced the beginning of our unified code base and adoption of the app being within Github in our shared repository.

For Sprint 3 and Sprint 4, we have elected to use two primary branches for the development of features; a feature development branch, and a main development branch. The main development branch is used for the completion of sprint goals (e.g. code merge in sprint 3, and code push at the end).

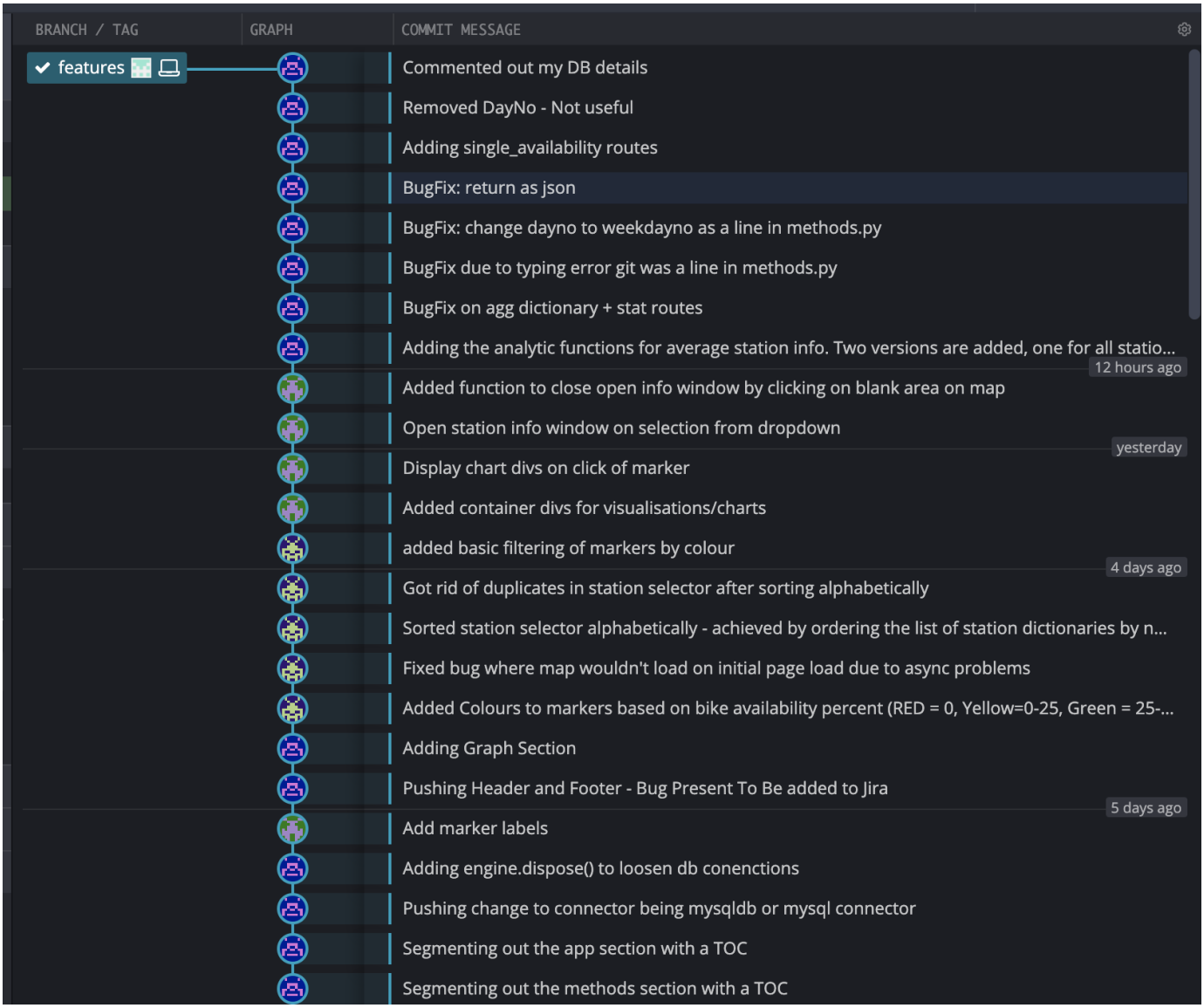
The feature development branch is the primary development branch due to the small team size. Within this branch a **Centralised Workflow** (https://www.atlassian.com/git/tutorials/comparing-workflows#centralized-workflow) process was enabled.

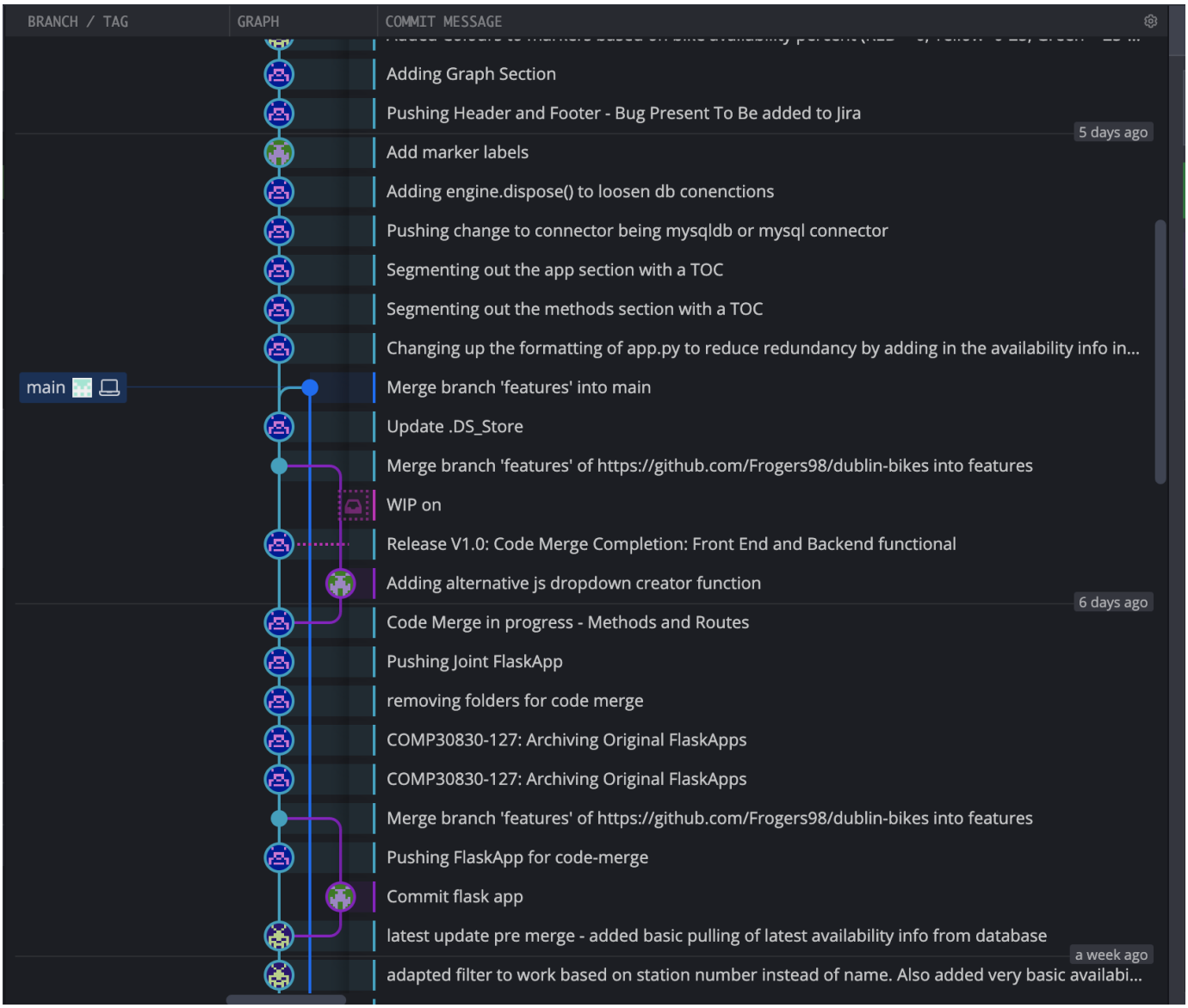
The team process for conducting git pushes to features therefore became:

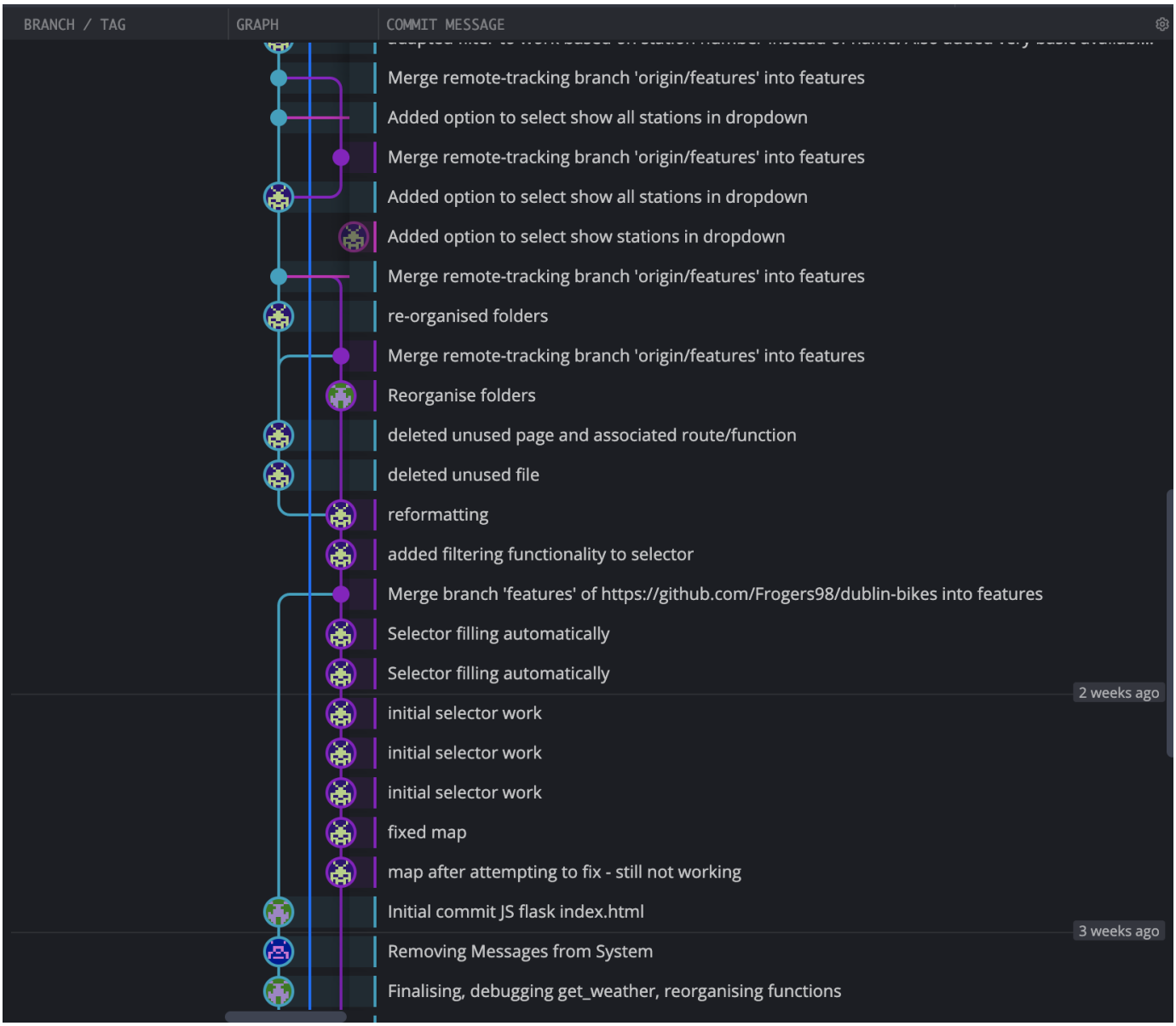
1. git pull
2. git add <file>
3. git commit -m '<Message>'
4. git push

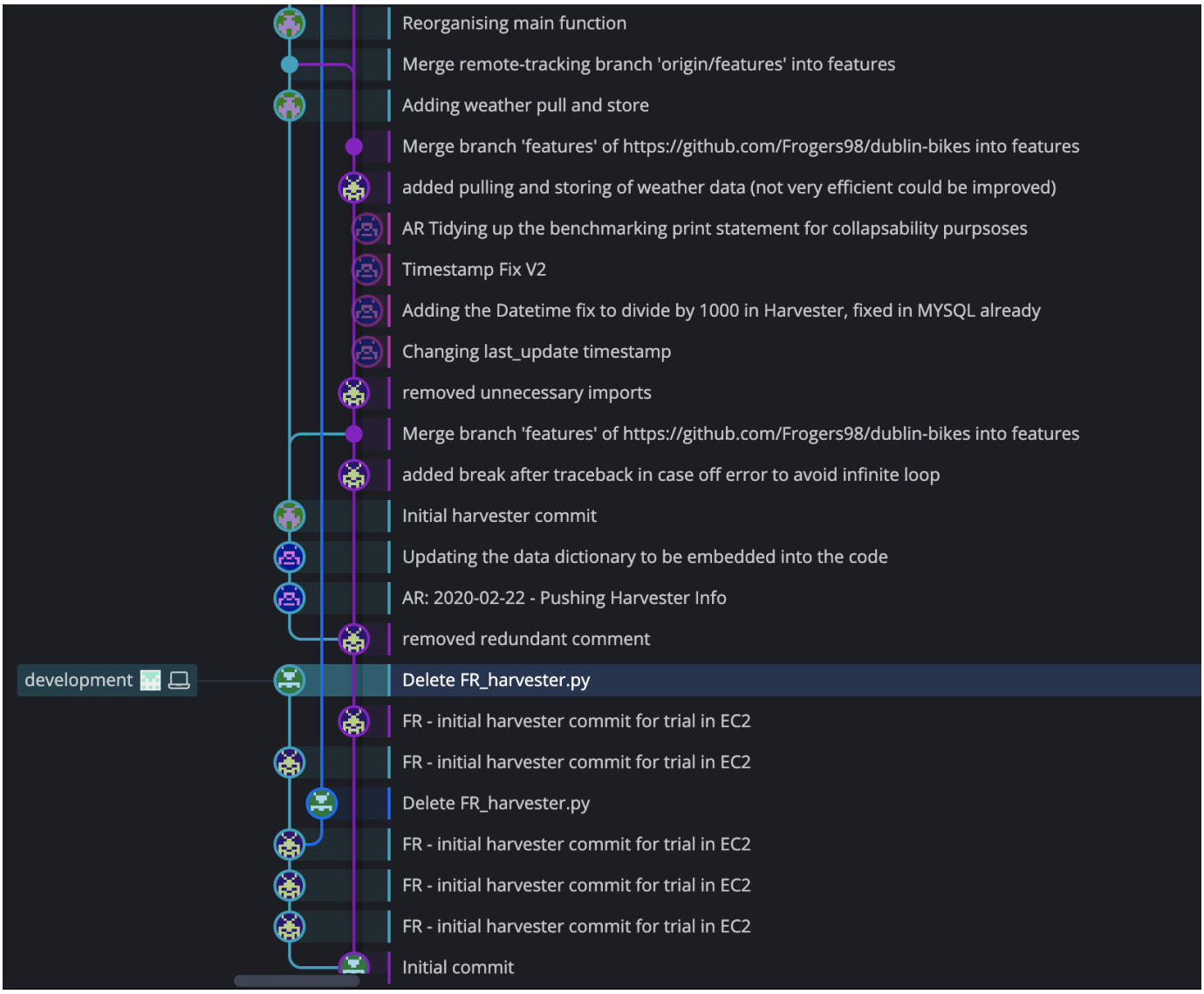
Git Branching Graph

Current GitHub repository is located at the following link: https://github.com/Frogers98/dublin-bikes


BRANCH / TAG
GRAPH
COMMIT MESSAGE
features
Commented out my DB details
Removed DayNo - Not useful
Adding single_availability routes
BugFix: return as json
BugFix: change dayno to weekdayno as a line in methods.py
BugFix due to typing error git was a line in methods.py
BugFix on agg dictionary + stat routes
Adding the analytic functions for average station info. Two versions are added, one for all statio...
12 hours ago
Added function to close open info window by clicking on blank area on map
Open station info window on selection from dropdown
yesterday
Display chart divs on click of marker
Added container divs for visualisations/charts
added basic filtering of markers by colour
4 days ago
Got rid of duplicates in station selector after sorting alphabetically
Sorted station selector alphabetically - achieved by ordering the list of station dictionaries by n...
Fixed bug where map wouldn't load on initial page load due to async problems
Added Colours to markers based on bike availability percent (RED = 0, Yellow=0-25, Green = 25-...
Adding Graph Section
Pushing Header and Footer - Bug Present To Be added to Jira
5 days ago
Add marker labels
Adding engine.dispose() to loosen db conenctions
Pushing change to connector being mysqldb or mysql connector
Segmenting out the app section with a TOC
Segmenting out the methods section with a TOC


BRANCH / TAG
GRAPH
COMMIT MESSAGE
Adding Graph Section
Pushing Header and Footer - Bug Present To Be added to Jira
5 days ago
Add marker labels
Adding engine.dispose() to loosen db conenctions
Pushing change to connector being mysqldb or mysql connector
Segmenting out the app section with a TOC
Segmenting out the methods section with a TOC
Changing up the formatting of app.py to reduce redundancy by adding in the availability info in...
main
Merge branch 'features' into main
Update .DS_Store
Merge branch 'features' of https://github.com/Frogers98/dublin-bikes into features
WIP on
Release V1.0: Code Merge Completion: Front End and Backend functional
Adding alternative js dropdown creator function
6 days ago
Code Merge in progress - Methods and Routes
Pushing Joint FlaskApp
removing folders for code merge
COMP30830-127: Archiving Original FlaskApps
COMP30830-127: Archiving Original FlaskApps
Merge branch 'features' of https://github.com/Frogers98/dublin-bikes into features
Pushing FlaskApp for code-merge
Commit flask app
latest update pre merge - added basic pulling of latest availability info from database
a week ago
adapted filter to work based on station number instead of name. Also added very basic availabi...


BRANCH / TAG
GRAPH
COMMIT MESSAGE
Merge remote-tracking branch 'origin/features' into features
Added option to select show all stations in dropdown
Merge remote-tracking branch 'origin/features' into features
Added option to select show all stations in dropdown
Added option to select show stations in dropdown
Merge remote-tracking branch 'origin/features' into features
re-organised folders
Merge remote-tracking branch 'origin/features' into features
Reorganise folders
deleted unused page and associated route/function
deleted unused file
reformatting
added filtering functionality to selector
Merge branch 'features' of https://github.com/Frogers98/dublin-bikes into features
Selector filling automatically
Selector filling automatically
2 weeks ago
initial selector work
initial selector work
initial selector work
fixed map
map after attempting to fix - still not working
Initial commit JS flask index.html
3 weeks ago
Removing Messages from System
Finalising, debugging get_weather, reorganising functions



Integrations

These pages are designed to document data integrations.

Weather 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

Bike Data - JC Decaux

This page details integration details (keys) of the application to JC Decaux to pull Dublin Bike Data.
The Table Below Lists the Integration Details and Endpoints.

The info returned is a json

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

`Dublin Stations: https://api.jcdecaux.com/vls/v1/stations?contract=dublin&apiKey=12345`

	Description	Service Endpoint	Parameters	Keys
1	Station Data, Given Contract	`https://api.jcdecaux.com/vls/v1/stations/{station_number}?contract={contract_name}`	{station_number} {contract_name}	ADD WHEN GROUP KEY
2	All Stations	`https://api.jcdecaux.com/vls/v1/stations`	{stations}	ADD WHEN GROUP KEY
3	Parks Given Contract	`https://api.jcdecaux.com/parking/v1/contracts/{contract_name}/parks`	{contract_name}	ADD WHEN GROUP KEY
4	Park Info	`https://api.jcdecaux.com/parking/v1/contracts/{contract_name}/parks/{park_number}`	{contract_name} {park_number}	ADD WHEN GROUP KEY
5	Get Dublin Stations	`https://api.jcdecaux.com/vls/v1/stations?contract=dublin`	N/A	ADD WHEN GROUP KEY

Dublin Station Mapping

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	number	int64	Yes	Yes	No	1, 2, … , 109	No	PK of Station
2	contract_name	varchar(6)	No	No	Yes	dublin	No	Denotes JCDecaus's contract **dublin**
3	name	varchar()	Yes	No	No	'MATER HOSPITAL'	No	Name of Station
4	address	varchar()	Yes	No	Maybe	Wolfe Tone Street	No	Address Name
5	position	Dictionary	Yes	No	Maybe	{'lat': 53.333653, 'lng': -6.248345}	No	Position - Latitude and Longitude as Dictionary
6	banking	bool	No	No	No	True/False	No	??? - Determine Meaning
7	bonus	bool	No	No	No	True/False	No	??? - Determine Meaning

8	bike_stands	int64	No	No	No	0,…,40	No	Maximum Number of Bikes at the Station Current Maximum = 40
9	available_bikes	int64	No	No	No	0,…,x-1	No	Number of Bikes available at the station
10	status	object	No	No	No	Open/Close	No	Open or Close - Identify meaning
11	last_update	float64	No	No	No	Date of Last Update	No	Timestamp updated

Summary Data

```
number                      int64
contract_name              object
name                       object
address                    object
position                   object
banking                      bool
bonus                        bool
bike_stands                 int64
available_bike_stands       int64
available_bikes             int64
status                     object
last_update               float64
dtype: object
```

	count	unique	top	freq
contract_name	109	1	dublin	109
name	109	109	MATER HOSPITAL	1
address	109	109	Wolfe Tone Street	1
position	109	109	{'lat': 53.333653, 'lng': -6.248345}	1
status	109	1	OPEN	109

	count	mean	std	min	25%	50%	75%	max
number	109.0	6.033028e+01	34.021250	2.000000e+00	3.100000e+01	6.100000e+01	9.000000e+01	1.170000e+02
bike_stands	109.0	3.211009e+01	7.683965	1.600000e+01	2.900000e+01	3.000000e+01	4.000000e+01	4.000000e+01
available_bike_stands	109.0	2.025688e+01	9.019941	0.000000e+00	1.300000e+01	2.000000e+01	2.800000e+01	3.800000e+01
available_bikes	109.0	1.111009e+01	5.276326	0.000000e+00	8.000000e+00	1.100000e+01	1.400000e+01	2.400000e+01
last_update	108.0	1.613474e+12	178619.620555	1.613474e+12	1.613474e+12	1.613474e+12	1.613474e+12	1.613475e+12

Header Summary:

	number	contract_name	name	address	position	banking	bonus	bike_stands	available_bike_stands	available_bikes	status	last_update
0	42	dublin	SMITHFIELD NORTH	Smithfield North	{'lat': 53.349562, 'lng': -6.278198}	True	False	30	8	22	OPEN	1.613474e+12
1	30	dublin	PARNELL SQUARE NORTH	Parnell Square North	{'lat': 53.353462, 'lng': -6.265305}	True	False	20	17	3	OPEN	1.613474e+12
2	54	dublin	CLONMEL STREET	Clonmel Street	{'lat': 53.336021, 'lng': -6.26298}	False	False	33	27	6	OPEN	1.613475e+12
3	108	dublin	AVONDALE ROAD	Avondale Road	{'lat': 53.359405, 'lng': -6.276142}	False	False	40	34	6	OPEN	1.613474e+12

Amazon Web Service

The application is hosted on Amazon Web Service.

The IP address for now of the instance is: http://34.229.255.100:5000/

Amazon Raw Data Store

This page details integration details (keys) of the application to RDS.

IP permission is required.

The Table Below Lists the Integration Details and Endpoints.

	User	Password	Database	Port	Endpoint
1	adamryan	adam.ryan1	dbbikes	3306	dbbikes.cmbuuvrlonfv.us-east-1.rds.amazonaws.com
2	admin	DublinBikesProject2201	dublin_bikes	3306	dublin-bikes.ciu0f2oznjig.us-east-1.rds.amazonaws.com
3	janeslevin	js2021dbbikes	dbbikes30830	3306	dbbikes30830.cfv8ckdtpwoq.us-east-1.rds.amazonaws.com

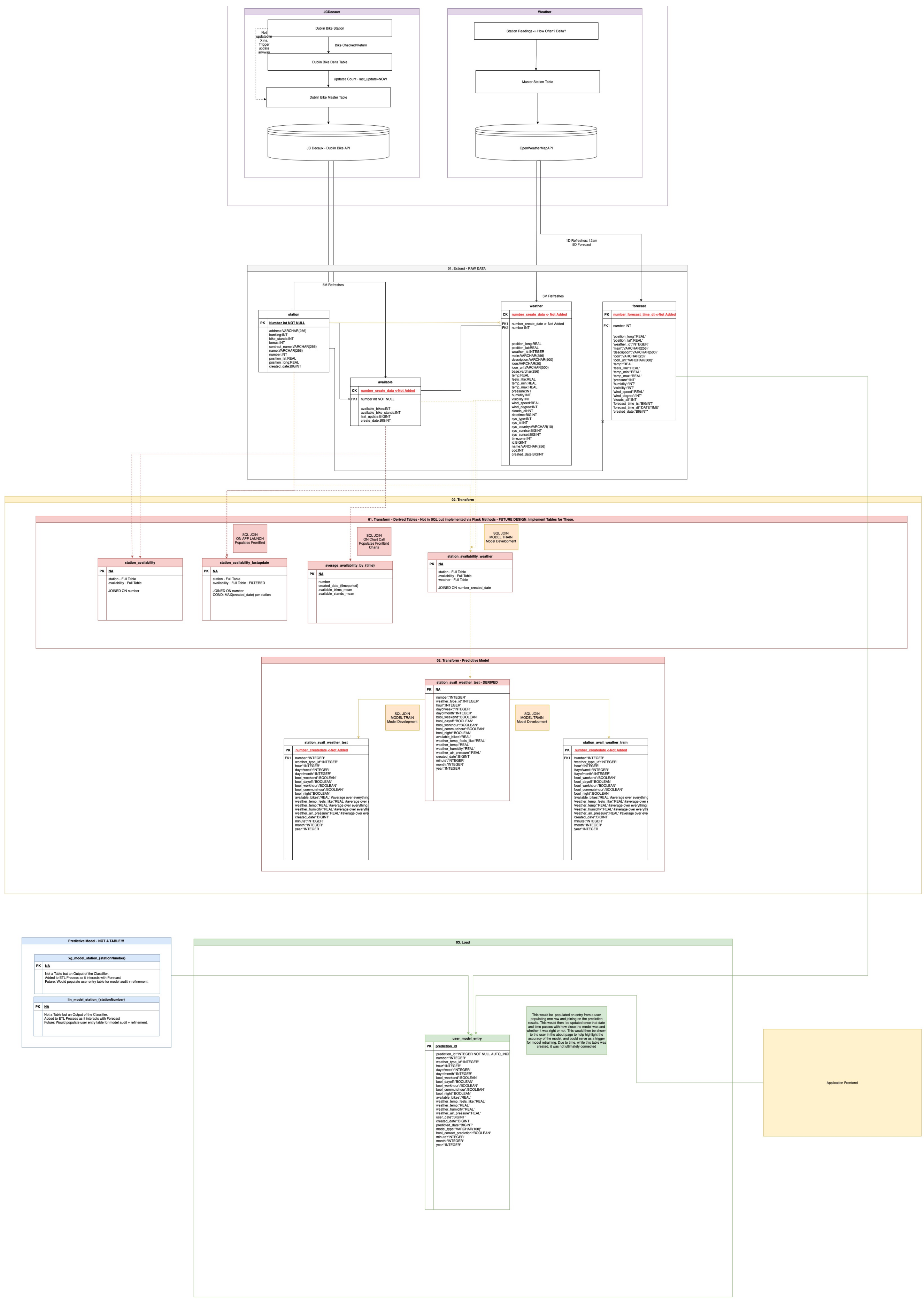
Data Model

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


JCDecaux
Dublin Bike Station
Not updated in X ns. Trigger update anyway
Bike Checked/Return
Dublin Bike Delta Table
Updates Count - last_update=NOW
Dublin Bike Master Table
JC Decaux - Dublin Bike API
Weather
Station Readings <- How Often? Delta?
Master Station Table
OpenWeatherMapAPI
1D Refreshes: 12am
5D Forecast
01. Extract - RAW DATA
5M Refreshes
5M Refreshes
station
PK Number int NOT NULL
address:VARCHAR(256)
banking:INT
bike_stands:INT
bonus:INT
contract_name:VARCHAR(256)
name:VARCHAR(256)
number:INT
position_lat:REAL
position_long:REAL
created_date:BIGINT
available
CK number_create_data <-Not Added
FK1 number int NOT NULL
available_bikes:INT
available_bike_stands:INT
last_update:BIGINT
create_date:BIGINT
weather
CK number_create_data <- Not Added
FK1 FK2 number_create_date <- Not Added
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
timezone:INT
id:BIGINT
name:VARCHAR(256)
cod:INT
created_date:BIGINT
forecast
PK number_forecast_time_dt <-Not Added
FK1 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. Transform
01. Transform - Derived Tables - Not in SQL but implemented via Flask Methods - FUTURE DESIGN: Implement Tables for These.
SQL JOIN
ON APP LAUNCH
Populates FrontEnd
SQL JOIN
ON Chart Call
Populates FrontEnd
Charts
SQL JOIN
MODEL TRAIN
Model Development
station_availability
PK NA
station - Full Table
availability - Full Table
JOINED ON number
station_availability_lastupdate
PK NA
station - Full Table
availability - Full Table - FILTERED
JOINED ON number
COND: MAX(created_date) per station
average_availability_by_{time}
PK NA
number
created_date_{timeperiod}
available_bikes_mean
available_stands_mean
station_availability_weather
PK NA
station - Full Table
availability - Full Table
weather - Full Table
JOINED ON number_created_date
02. Transform - Predictive Model
station_avail_weather_test - DERIVED
PK NA
'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'
'weather_temp_feels_like':'REAL'
'weather_temp':'REAL'
'weather_humidity':'REAL'
'weather_air_pressure':'REAL'
'created_date':'BIGINT'
'minute':'INTEGER'
'month':'INTEGER'
'year':'INTEGER
SQL JOIN
MODEL TRAIN
Model Development
SQL JOIN
MODEL TRAIN
Model Development
station_avail_weather_test
PK number_createdate <-Not Added
FK1 '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
'weather_temp':'REAL' #average over everything
'weather_humidity':'REAL' #average over everythi
'weather_air_pressure':'REAL' #average over eve
'created_date':'BIGINT'
'minute':'INTEGER'
'month':'INTEGER'
'year':'INTEGER
station_avail_weather_train
PK number_createdate <-Not Added
FK1 '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
'weather_temp':'REAL' #average over everything
'weather_humidity':'REAL' #average over everythi
'weather_air_pressure':'REAL' #average over eve
'created_date':'BIGINT'
'minute':'INTEGER'
'month':'INTEGER'
'year':'INTEGER
Predictive Model - NOT A TABLE!!!
xg_model_station_{stationNumber}
PK NA
Not a Table but an Output of the Classifier.
Added to ETL Process as it interacts with Forecast
Future: Would populate user entry table for model audit + refinement.
lin_model_station_{stationNumber}
PK NA
Not a Table but an Output of the Classifier.
Added to ETL Process as it interacts with Forecast
Future: Would populate user entry table for model audit + refinement.
03. Load
user_model_entry
PK prediction_id
'prediction_id':'INTEGER NOT NULL AUTO_INCR
'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'
'weather_temp_feels_like':'REAL'
'weather_temp':'REAL'
'weather_humidity':'REAL'
'weather_air_pressure':'REAL'
'user_date':'BIGINT'
'created_date':'BIGINT'
'predicted_date':'BIGINT'
'model_type':'VARCHAR(100)'
'bool_correct_prediction':'BOOLEAN'
'minute':'INTEGER'
'month':'INTEGER'
'year':'INTEGER'
This would be populated on entry from a user populating one row and joining on the prediction results. This would then be updated once that date and time passes with how close the model was and whether it was right or not. This would then be shown to the user in the about page to help highlight the accuracy of the model, and could serve as a trigger for model retraining. Due to time, while this table was created, it was not ultimately connected
Application Frontend

Source File: Keep Up to Date

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.

> The root page Page: Transform - Frontend Store could not be found in space Disappster.

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={
                  '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'
                                }
                }
```

Extract Data Store

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

Station

This page details the availability table.

It captures bike stand numbers.

Station Table

Dublin Station Mapping

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	number	INT	Yes	Yes	No	1, 2, … , 109	No	PK of Station
2	contract_name	varchar(6)	No	No	Yes	dublin	No	Denotes JCDecaus's contract **dublin**
3	name	varchar(256)	Yes	No	No	'MATER HOSPITAL'	No	Name of Station
4	address	varchar(256)	Yes	No	Maybe	Wolfe Tone Street	No	Address Name
5	position_lat	REAL	Yes	No	Maybe	53.333653	No	Position - Latitude
6	position_long	REAL	Yes	No	Maybe	-6.248345	No	Position - Longitude
7	banking	int64	No	No	No	True/False	No	??? - Determine Meaning
8	bonus	int64	No	No	No	True/False	No	??? - Determine Meaning
9	status	varchar(256)	No	No	No	Open/Close	No	Open or Close - Identify meaning
10	created_date	BIGINT	No	No	No	Date of Last Update	No	Timestamp updated

Note

Number is a primary key.

Summary Data

	count	unique	top	freq
contract_name	109	1	dublin	109
name	109	109	MATER HOSPITAL	1
address	109	109	Wolfe Tone Street	1
position	109	109	{'lat': 53.333653, 'lng': -6.248345}	1
status	109	1	OPEN	109

	count	mean	std	min	25%	50%	75%	max
number	109.0	6.033028e+01	34.021250	2.000000e+00	3.100000e+01	6.100000e+01	9.000000e+01	1.170000e+02
bike_stands	109.0	3.211009e+01	7.683965	1.600000e+01	2.900000e+01	3.000000e+01	4.000000e+01	4.000000e+01
available_bike_stands	109.0	2.025688e+01	9.019941	0.000000e+00	1.300000e+01	2.000000e+01	2.800000e+01	3.800000e+01
available_bikes	109.0	1.111009e+01	5.276326	0.000000e+00	8.000000e+00	1.100000e+01	1.400000e+01	2.400000e+01
last_update	108.0	1.613474e+12	178619.620555	1.613474e+12	1.613474e+12	1.613474e+12	1.613474e+12	1.613475e+12

Header Summary:

	number	contract_name	name	address	position	banking	bonus	bike_stands	available_bike_stands	available_bikes	status	last_update
0	42	dublin	SMITHFIELD NORTH	Smithfield North	{'lat': 53.349562, 'lng': -6.278198}	True	False	30	8	22	OPEN	1.613474e+12
1	30	dublin	PARNELL SQUARE NORTH	Parnell Square North	{'lat': 53.353462, 'lng': -6.265305}	True	False	20	17	3	OPEN	1.613474e+12
2	54	dublin	CLONMEL STREET	Clonmel Street	{'lat': 53.336021, 'lng': -6.26298}	False	False	33	27	6	OPEN	1.613475e+12
3	108	dublin	AVONDALE ROAD	Avondale Road	{'lat': 53.359405, 'lng': -6.276142}	False	False	40	34	6	OPEN	1.613474e+12

Availability

This page details the availability table.

It captures bike stand numbers.

Availability Table

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	number	int64	No	No	Yes	1, 2, … , 109	No	FK to Station
2	bike_stands	int64	No	No	No	0,…,40	No	Maximum Number of Bikes at the Station Current Maximum = 40
3	available_bikes	int64	No	No	No	0,…,x-1	No	Number of Bikes available at the station
4	last_update	bigint	No	No	No	Timestamp of Update	No	Timestamp updated
5	created_date	bigint	No	No	No	Timestamp of Add	No	Timestamp Added to TB

Note

Number is a foreign key.

Number_LastUpdate is a composite key.

created_date will be a joinable piece for weather, one-to-one.

No Primary Key Set yet.

Summary Data

	count	unique	top	freq
contract_name	109	1	dublin	109
name	109	109	MATER HOSPITAL	1
address	109	109	Wolfe Tone Street	1
position	109	109	{'lat': 53.333653, 'lng': -6.248345}	1
status	109	1	OPEN	109

	count	mean	std	min	25%	50%	75%	max
number	109.0	6.033028e+01	34.021250	2.000000e+00	3.100000e+01	6.100000e+01	9.000000e+01	1.170000e+02
bike_stands	109.0	3.211009e+01	7.683965	1.600000e+01	2.900000e+01	3.000000e+01	4.000000e+01	4.000000e+01
available_bike_stands	109.0	2.025688e+01	9.019941	0.000000e+00	1.300000e+01	2.000000e+01	2.800000e+01	3.800000e+01
available_bikes	109.0	1.111009e+01	5.276326	0.000000e+00	8.000000e+00	1.100000e+01	1.400000e+01	2.400000e+01
last_update	108.0	1.613474e+12	178619.620555	1.613474e+12	1.613474e+12	1.613474e+12	1.613474e+12	1.613475e+12

Header Summary:

	number	contract_name	name	address	position	banking	bonus	bike_stands	available_bike_stands	available_bikes	status	last_update
0	42	dublin	SMITHFIELD NORTH	Smithfield North	{'lat': 53.349562, 'lng': -6.278198}	True	False	30	8	22	OPEN	1.613474e+12
1	30	dublin	PARNELL SQUARE NORTH	Parnell Square North	{'lat': 53.353462, 'lng': -6.265305}	True	False	20	17	3	OPEN	1.613474e+12
2	54	dublin	CLONMEL STREET	Clonmel Street	{'lat': 53.336021, 'lng': -6.26298}	False	False	33	27	6	OPEN	1.613475e+12
3	108	dublin	AVONDALE ROAD	Avondale Road	{'lat': 53.359405, 'lng': -6.276142}	False	False	40	34	6	OPEN	1.613474e+12

Weather

Weather Table

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	number	INT	No	No	Yes - station	1…109	No	Station Number
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	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	BIGINT	No	No	No	1614101644	No	Datetime added to DB

Note

Number_Created_date is a FK to Availability

Forecast

Forecast Table

	Name	Type	All Instances Unique?	PK	FK	Field Values	Nullable	Description
1	number	INT	No	No	Yes - station	1…109	No	Station Number
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	forecast_time_ts	BIGINT	No	No	No	1614079641	No	Datetime - TS format
21	forecast_time_dt	DATETIME	No	No	No	2021-01-01	No	Datetime - DT Format
22	created_date	BIGINT	No	No	No	1614101644	No	Datetime added to DB

Note

Number_Created_date is a FK to Availability

Transform Data Store

The Transform Data Store involves any staging tables which are used to transform the raw data into loadable data

Transform - Machine Learning Data Store

Machine Learning Tables

Training Data Set

Training Data Set

Transform - Frontend Store

Active Count

Loading Data Store

The Load tables feature any table which are then loaded into Analytic Data Stores

ETL - Data Flows For Integrations

This page should be updated to contain:

1. A list of all integrations into the app.
2. A data flow diagram.
3. Integration Details where relevant.

The purpose of this page is to create a high level overview.

Design Decisions

These pages track design decision discussions

UAT and PROD Plan

As this application only consists of a production MVP release, this page is not relevant for this phase of the project.