M.O.P

MELBOURNE OPENDATA PLAYGROUND

Project Handover Document

Data 2 Intelligence Consulting

Melbourne City

Trimester 3, 2021

Project Handover Document (T3 2021)

Table of Contents

1. Project Information

1.1. Client/Product Owner/s

Mr Remi Cattier
Data Hub Lead
remi.cattier@melbourne.vic.gov.au

Mr. Will McIntosh
Senior Advisor to the CEO
will.mcintosh@melbourne.vic.gov.au

1.2. Academic Mentor/Supervisor

Dr Fatemeh Ansarizadeh
Casual Academic
f.ansarizadeh@deakin.edu.au

1.3. Project Team

Deakin 2 Intelligence Consulting - Melbourne City

Student ID	Student Name	Role
220400024	Albert Hon	Senior Student (Squad Leader)
220013964	Josh Sansom-Sherwill	Senior Student
219235988	Ngoc Dung Huynh	Senior Student
220051603	Tithra Chap	Senior Student
220597584	William Hebblewhite	Junior Student
219279071	Shalom Chin	Junior Student
217495571	Alex Vuong	Junior Student
220541733	Steven James Tuften	Junior Student
220595465	Angus Maiden	Junior Student

Project Handover Document (T3 2021)

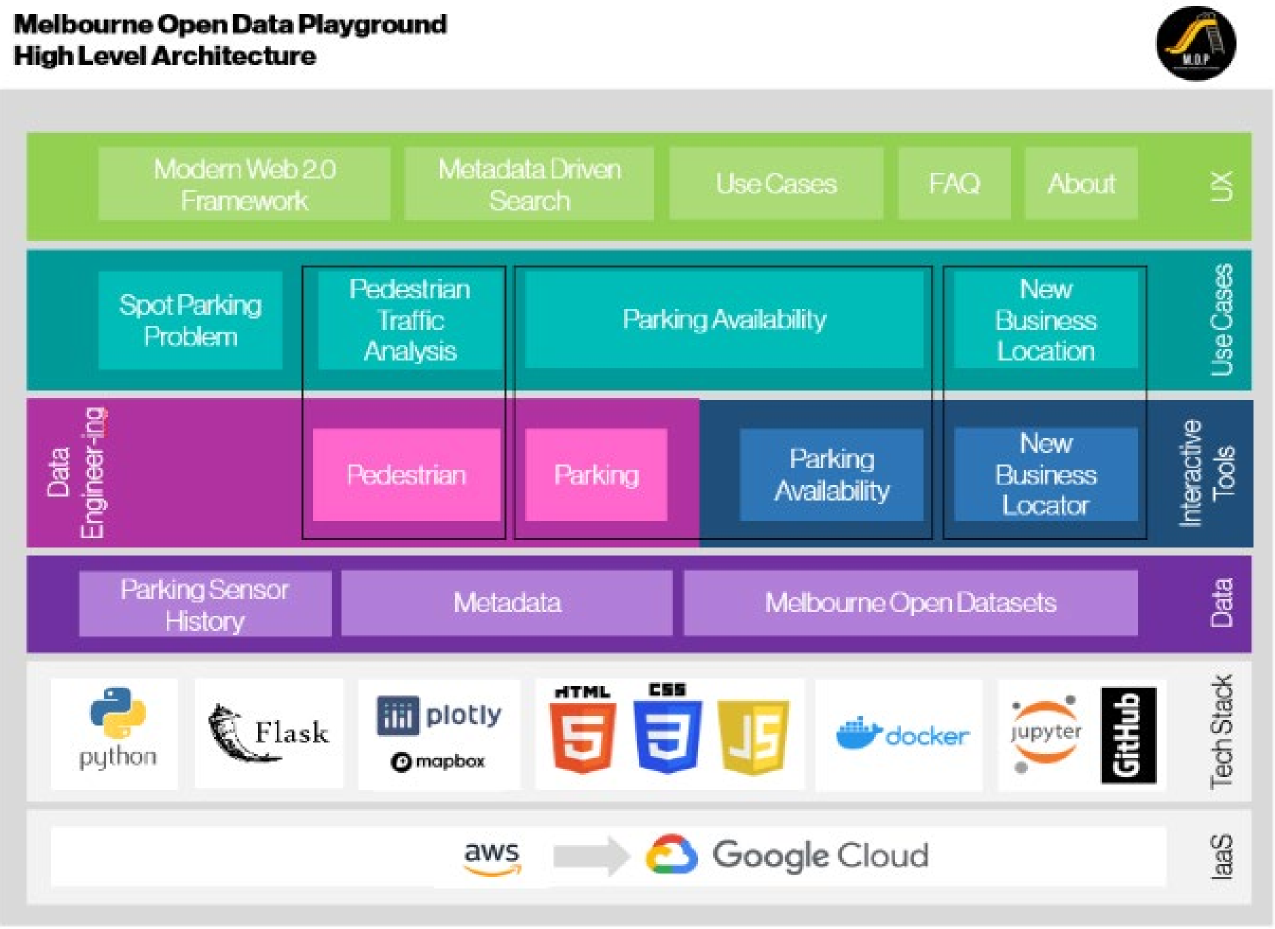
2. Project Overview

In the previous trimester (T2 2021) the project had its scope redefined by the client. This introduced the current problem statement:

The City of Melbourne has been an Australian leader in Open Data since 2014. The latest research and local user engagement have identified a gap where users would like access to Open Data example tools so that our users can re-use these tools in their apps and city solutions.

This project delivers a proof-of-concept example of how calls to Open Data API can be made to deliver a solution. Figure 2.1 below illustrates the major components of the Melbourne Open Data Playground platform.

Figure 2.1: Melbourne Open Data Platform High Level Architecture



This trimester, the client identified two key benefits that this re-definition of the scope would deliver for this project:

- The toolkit examples created will be developed to maximise efficiency and time saving for the council's staff. This will save council time through the reduction of calls and emails relating to questions of how to use council's Open Data platform API.

- The use cases lists are the most common customer requests for assistance in creating a data solution using City of Melbourne's open data/API.

Building on the foundation of work completed in the previous semester, our squad focussed on delivering two key outputs: (1) Adding additional toolkit examples in line with the scope and key benefits of the project. (2) Making platform improvements that aid not only in user experience but also help streamline integration for future squads.

A major result of this focus has been the development of style templates and an integration framework. This will provide future squads with consistency and uniformity across products and streamlined delivery of toolkit examples.

The project this semester also saw the addition of three toolkit examples: New Business Location, Spot Parking Problem, and Pedestrian Traffic Analysis. These use cases explored and analysed a variety of datasets available from Melbourne Open Data. Utilising python libraries such as Selenium, folium, Plotly, and mapbox, our data science team was able to deliver highly stylized, interactive tools to help in developing business insights using the City of Melbourne's Open Data API.

The web development team made further improvements to the web portal adding more informative pages to improve user experience, as well as enhancing backend processing performance, and website security. A crucial part of the web development team's deliverables this semester has been the planning and implementation of a framework for use case integration onto the web portal. This framework will aid in future toolkit integration and allow new use cases to be developed and made product-ready in a short amount of time.

The work completed this semester leaves the D2i (Melbourne City) tribe in an excellent position. This work will give future squads a strong foundation through which to quickly deliver project outputs. The development of style templates, integration frameworks, and key platform improvements means that future members of D2i (Melbourne City) can quickly meet new business demands and questions and drive further adoption and innovation of the Melbourne Open Data Platform.

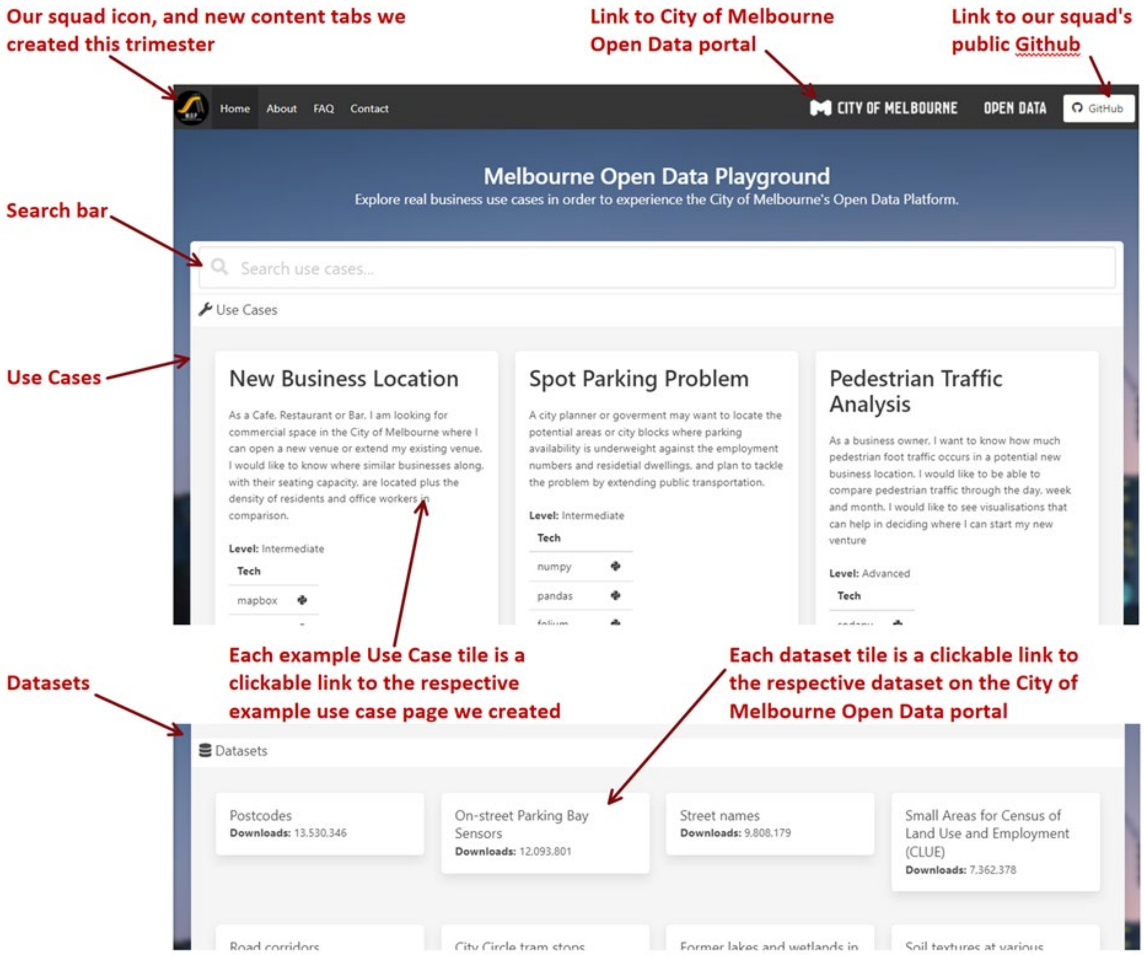
Project Handover Document (T3 2021)

3. User Manual

The following section provides a guide to the main product of this project being the Melbourne Open Data Playground.

3.1. Home Page – Discovery Portal

Figure 3.1: Home Page - Discovery Portal



Our squad icon, and new content tabs we created this trimester

Link to City of Melbourne Open Data portal

Link to our squad's public Github

Search bar

Use Cases

Datasets

Each example Use Case tile is a clickable link to the respective example use case page we created

Each dataset tile is a clickable link to the respective dataset on the City of Melbourne Open Data portal

Melbourne Open Data Playground

Explore real business use cases in order to experience the City of Melbourne's Open Data Platform.

New Business Location

As a Cafe, Restaurant or Bar, I am looking for commercial space in the City of Melbourne where I can open a new venue or extend my existing venue. I would like to know where similar businesses along, with their seating capacity, are located plus the density of residents and office workers in comparison.

Level: Intermediate

Tech

mapbox

Spot Parking Problem

A city planner or goverment may want to locate the potential areas or city blocks where parking availability is underweight against the employment numbers and residetial dwellings, and plan to tackle the problem by extending public transportation.

Level: Intermediate

Tech

numpy

pandas

Pedestrian Traffic Analysis

As a business owner, I want to know how much pedestrian foot traffic occurs in a potential new business location. I would like to be able to compare pedestrian traffic through the day, week and month. I would like to see visualisations that can help in deciding where I can start my new venture

Level: Advanced

Tech

Postcodes
Downloads: 13,530,346

On-street Parking Bay Sensors
Downloads: 12,093,801

Street names
Downloads: 9,808,179

Small Areas for Census of Land Use and Employment (CLUE)
Downloads: 7,362,378

Road corridors

City Circle tram stops

Former lakes and wetlands in

Soil textures at various

The Melbourne Open Data Playground's home page acts as a discovery portal. Showcasing a dynamic, user-friendly interface, this portal allows easy access to datasets, use-cases and project information.

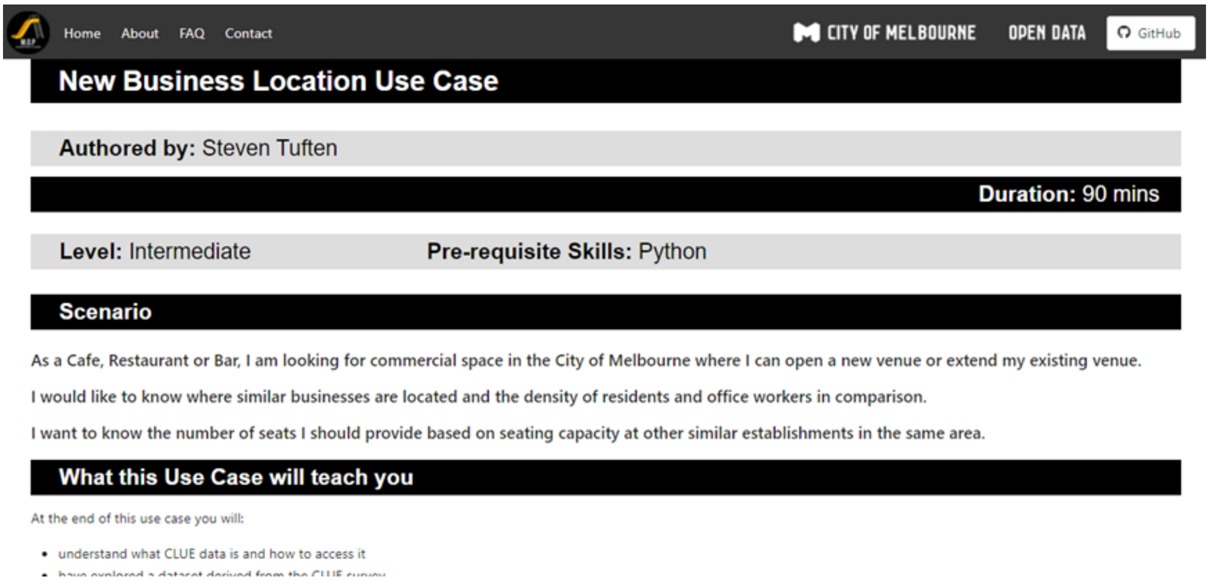
In the top left of the page, users can see our newly created project logo. Also visible are tabs to Home, About, FAQ and Contact pages. In the top right users can access the City of Melbourne Open Data portal and the project's public GitHub which hosts the developed python notebooks from this project.

Central to the discovery portal is a newly enhanced search bar. Users can perform keyword searches on both use cases that are integrated onto the portal as well as direct access over 200 datasets from Melbourne Open Data.

The middle of the discovery portal is dominated by the project's use cases. These tiles provide a general overview of the use cases.

3.2. Example Use Case page

Figure 3.2 Use case example



New Business Location Use Case

Authored by: Steven Tuften

Duration: 90 mins

Level: Intermediate **Pre-requisite Skills:** Python

Scenario

As a Cafe, Restaurant or Bar, I am looking for commercial space in the City of Melbourne where I can open a new venue or extend my existing venue.

I would like to know where similar businesses are located and the density of residents and office workers in comparison.

I want to know the number of seats I should provide based on seating capacity at other similar establishments in the same area.

What this Use Case will teach you

At the end of this use case you will:

- understand what CLUE data is and how to access it
- have explored a dataset derived from the CLUE survey

Users can access these examples by clicking on the tile which will take them to the corresponding detailed use case page.

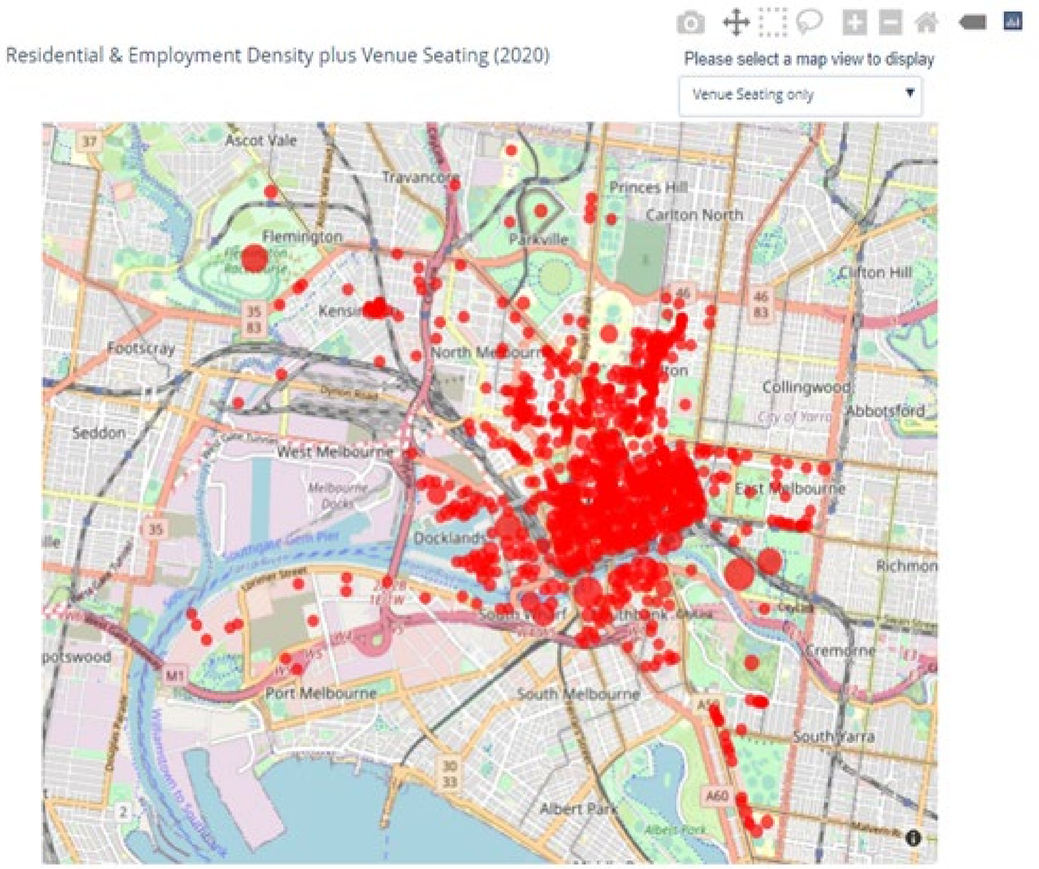
Project Handover Document (T3 2021)

Consistent formatting, styling and integration frameworks provide the user with uniform toolkits to explore. This uniformity allows the user to easily identify the structure of the toolkits and aids in the learning experience.

The new format provides an improved educational experience. The adaption of a content hierarchy provides helpful information to the user and allows them to make an informed choice about which toolkit to explore regarding the business scenario, level of difficulty, estimated duration and pre-requisite skills needed to get the most out of the learning experience.

This is followed by a step-by-step guide to exploring and understanding the code and data that is being developed. Additionally, interactive widgets are embedded in the use-cases, where possible, to better demonstrate the example scenario.

Figure 3.3 Use case interactivity example from the New Business Location use case



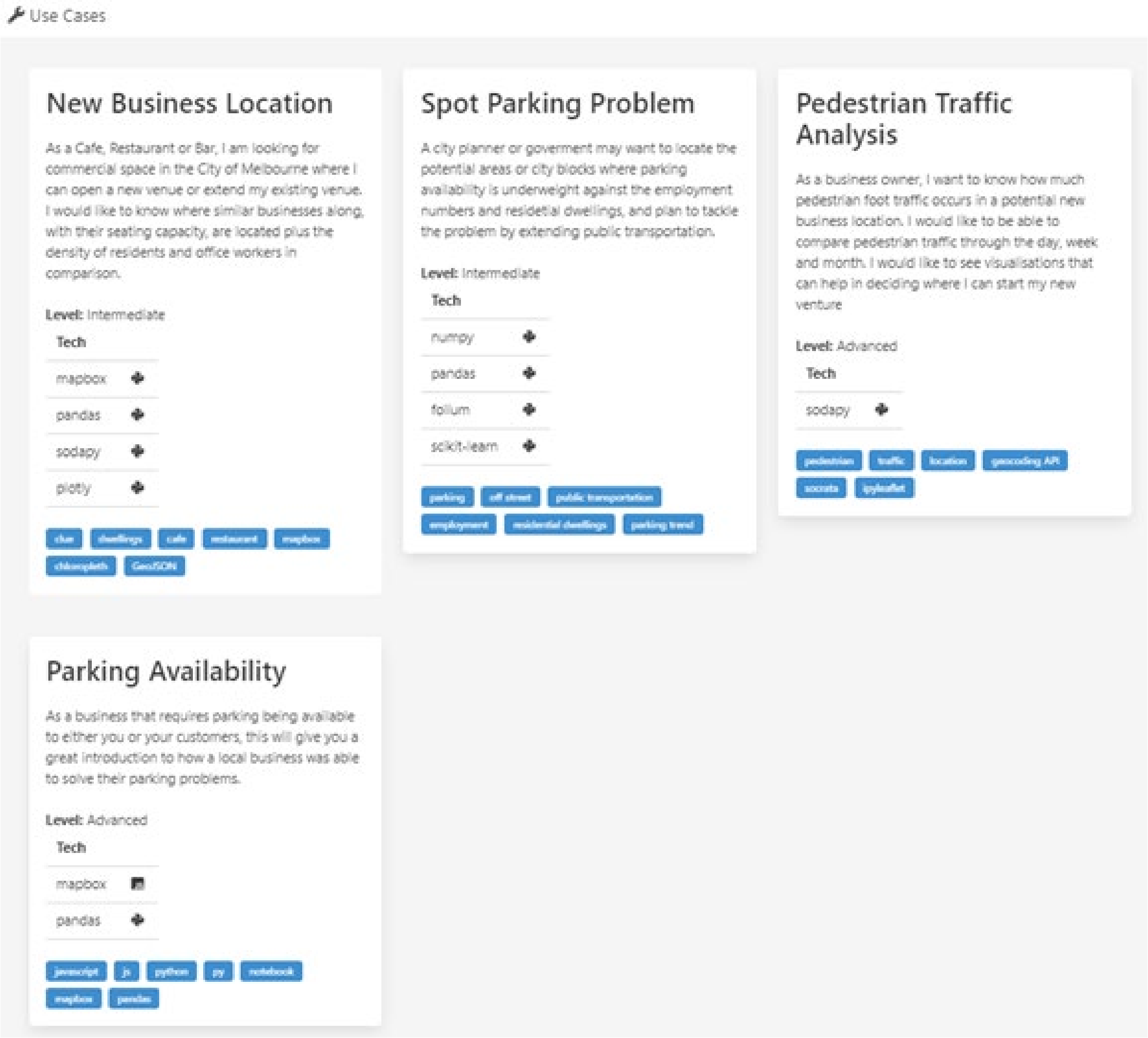
Residential & Employment Density plus Venue Seating (2020)

Please select a map view to display: Venue Seating only

3.3. More examples

Over the course of this trimester (T3/2021) we have developed three use-cases in addition to a prior use case developed in the previous trimester (T2/2021). Use case tiles, which appear on the home page of the discovery portal, provide users with a brief description of the use case, as well as an assumed difficulty level and the types of technology involved in the use-case. Metadata tags appear along the bottom of each use case tile and aid in the search for relevant use cases.

Figure 3.4 Use case tiles - Home Page



Use Cases

New Business Location

As a Cafe, Restaurant or Bar, I am looking for commercial space in the City of Melbourne where I can open a new venue or extend my existing venue. I would like to know where similar businesses along, with their seating capacity, are located plus the density of residents and office workers in comparison.

Level: Intermediate

Tech

mapbox ✚
pandas ✚
sodapy ✚
plotly ✚

clue · dwellings · cafe · restaurant · mapbox · chloropleth · GeoJSON

Spot Parking Problem

A city planner or goverment may want to locate the potential areas or city blocks where parking availability is underweight against the employment numbers and resideltial dwellings, and plan to tackle the problem by extending public transportation.

Level: Intermediate

Tech

numpy ✚
pandas ✚
follum ✚
scikit-learn ✚

parking · off street · public transportation · employment · residential dwellings · parking trend

Pedestrian Traffic Analysis

As a business owner, I want to know how much pedestrian foot traffic occurs in a potential new business location. I would like to be able to compare pedestrian traffic through the day, week and month. I would like to see visualisations that can help in deciding where I can start my new venture

Level: Advanced

Tech

sodapy ✚

pedestrian · traffic · location · geocoding API · socrata · ipyleaflet

Parking Availability

As a business that requires parking being available to either you or your customers, this will give you a great introduction to how a local business was able to solve their parking problems.

Level: Advanced

Tech

mapbox ■
pandas ✚

javascript · js · python · py · notebook · mapbox · pandas

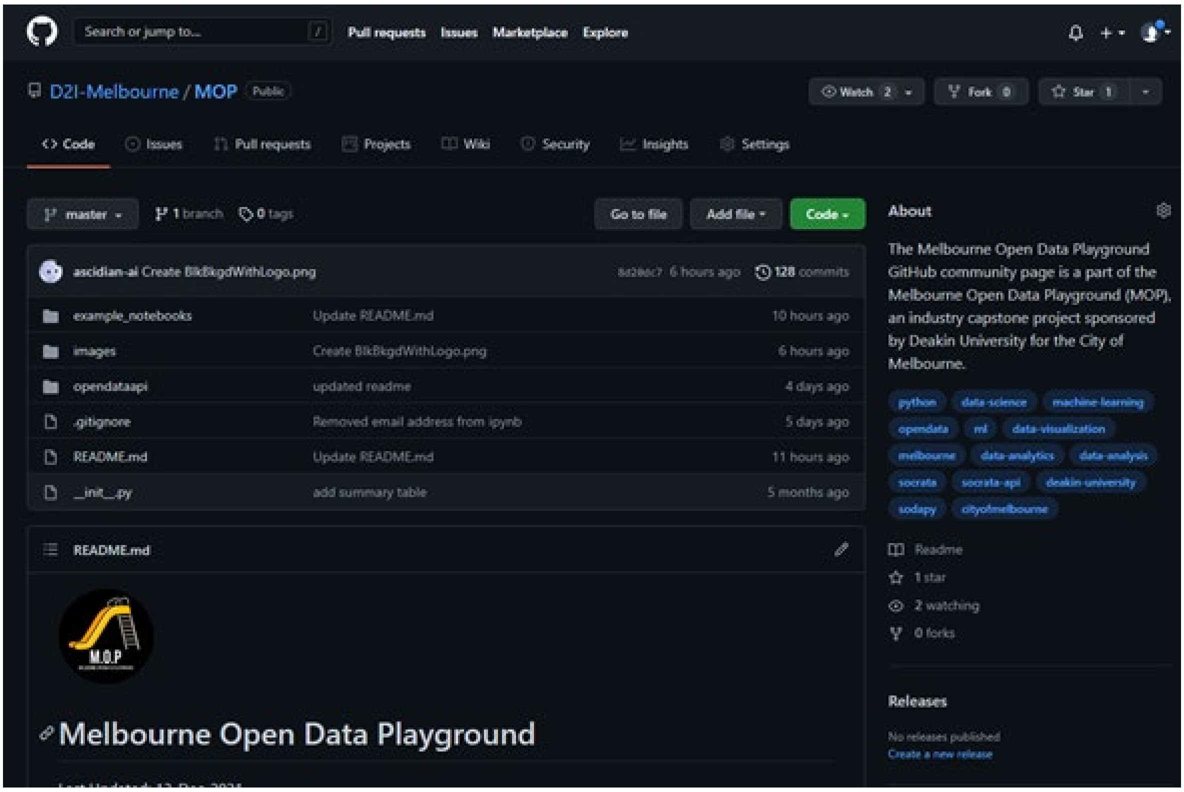
4. GitHub

In addition to the Melbourne Open Data playground, the squad has also curated a public-facing GitHub. In this environment we showcase python code that supports and extends the use cases integrated onto the web portal.

The Jupyter notebooks are shared for the benefit of those who may be more technically motivated to explore and build upon our examples.

Readme pages help users to navigate to and explore these works. The squad believes that such access will aid in building a community around the playground and Melbourne Open Data more generally. Figure 4.1 below illustrates the home page of the GitHub site.

Figure 4.1: Public-facing GitHub repository



ascidian-ai Create BlkBkgdWithLogo.png 8d28dc7 6 hours ago 128 commits

example_notebooks	Update README.md	10 hours ago
images	Create BlkBkgdWithLogo.png	6 hours ago
opendataapi	updated readme	4 days ago
.gitignore	Removed email address from ipynb	5 days ago
README.md	Update README.md	11 hours ago
__init__.py	add summary table	5 months ago

README.md

Melbourne Open Data Playground

About

The Melbourne Open Data Playground GitHub community page is a part of the Melbourne Open Data Playground (MOP), an industry capstone project sponsored by Deakin University for the City of Melbourne.

Releases

No releases published
Create a new release

5. Completed Deliverables

In this section we describe the key deliverables along with their main contributor(s) for the project this trimester.

5.1. Improved Search Performance (Josh)

The search performance for datasets was drastically improved and reduced to one-third the time after refactoring the JavaScript code. This code can be found in the 'search.js' file of the 'static' folder of the webapp.

Trello:	https://trello.com/c/qtBqsJO0
Bitbucket:	https://bitbucket-students.deakin.edu.au/projects/D2IC-PG/repos/d2i---melbourne-city/browse/webapp/flaskr/static/search.js

5.2. UX Improvements (Angus, Josh, Shalom)

Three new pages (Contact, FAQ, About) were added to the website to give more information on what the website was about, how to use it and whom to contact. These pages can be found in "flaskr/templates/home" folder.

Trello:	https://trello.com/c/i8BmHHdK, https://trello.com/c/uDqfqjGN
Bitbucket:	https://bitbucket-students.deakin.edu.au/projects/D2IC-PG/repos/d2i---melbourne-city/browse/webapp/flaskr/templates/home

In addition, a new logo was created and added to the site's menu bar (base.html) to give identity to the project. The logo asset can be found in the 'static/images' folder.

Trello:	https://trello.com/c/HmbyAa9w
Bitbucket:	https://bitbucket-students.deakin.edu.au/projects/D2IC-PG/repos/d2i---melbourne-city/browse/webapp/flaskr/templates/base.html

There are two searchable categories – use cases and datasets. The use case cards were furnished with more information such as the difficulty level to give more details to potential users of the data. Links to the dataset cards have also been included so that visitors to the site can get more information on the datasets from the City of Melbourne (CoM) Open Data website when clicked on. JavaScript is used to call the data in the 'search.js' file. The front-end code is found in the 'index.html' page.

Trello: https://trello.com/c/RGzPJCMr, https://trello.com/c/SrE252w1
Bitbucket: https://bitbucket-students.deakin.edu.au/projects/D2IC-PG/repos/d2i---melbourne-city/browse/webapp/flaskr/templates/home/index.html

5.3. Web Security (Angus, Shalom)

We did a security scan using hostedscan.com to identify any potential security issues. A total of 3 medium and 4 low security issues were identified, of which we resolved 1 medium and 1 low security risks to do with cross-site origin. This was resolved with the implementation of the Talisman web package in the '__inti.py__' file. The other two medium security risks with regards to open http and https ports were not considered as a problem because the ports are needed for users to view the public website.

There was also an additional security risk identified by AWS on outdated software and packages. This was resolved through an additional line of code added to the deployment script to update any packages.

Trello: https://trello.com/c/5jIDoeTV
Bitbucket: https://bitbucket-students.deakin.edu.au/projects/D2IC-PG/repos/d2i---melbourne-city/browse/webapp/flaskr/__init__.py

The results of the first security scan done on 17th November 2021 can be found here:

https://deakin365.sharepoint.com/:b:/r/sites/Data2IntelligenceConsulting/Shared%20Documents/D2I%20(Melbourne%20City)/T3%202021/WEB%20team%20Documents/open_plaground_report.pdf?csf=1&web=1&e=eTzxgh

The second security scan was completed on 2nd December 2021 and the results of that scan can be found here:

https://deakin365.sharepoint.com/:b:/r/sites/Data2IntelligenceConsulting/Shared%20Documents/D2I%20(Melbourne%20City)/T3%202021/WEB%20team%20Documents/MoP_scan_report_011221.pdf?csf=1&web=1&e=WrUcmG

5.4. "Cold Start" to Parking Availability Website (Josh)

Initially, the parking availability use case experienced performance issues when loading the data into the interactive maps. This was caused by the initiation time required to execute the lambda function on AWS. This issue was resolved by purchasing a reserved concurrency capacity for the function.

Trello: https://trello.com/c/7LOkelaw

5.5. Google Cloud Plan (Josh, Angus, Shalom)

The website is sitting on an AWS infrastructure and, for this trimester, being independently funded by one of the Senior team members Josh. This cost will be handed over to the next trimester's seniors team at the completion of this trimester.

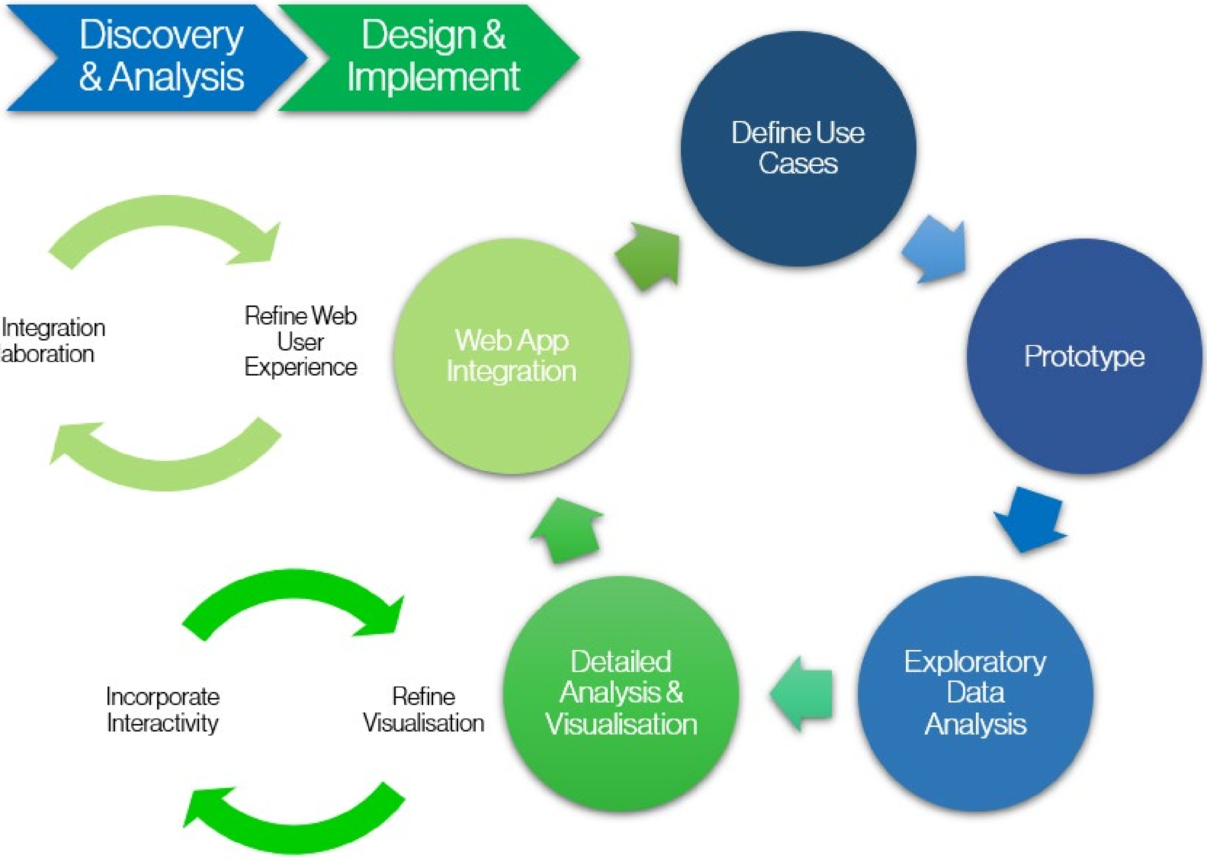
The team wanted to explore tapping into the University's resources to continue this project without the team members having to bear the burden of the infrastructure costs. The University recommended that we use Google Cloud Platform (GCP) as a free resource. Moreover, the Client has expressed interest in making our website Cloud agnostic for scalability. The AWS services were mapped to the GCP services, and a plan was created to figure out how to port the current website over to GCP. Josh developed the code for bringing the app over to GCP leveraging Google Cloud Functions and Cloud Run. Details of this plan can be found in the "google_cloud_test" folder of the M2I project folder.

Trello: https://trello.com/c/49TMAR4K, https://trello.com/c/9BWJnOXi
Bitbucket: https://bitbucket-students.deakin.edu.au/projects/D2IC-PG/repos/d2i---melbourne-city/browse/google_cloud_test

5.6. Addition of New Use Cases (Entire Team)

The entire team is involved in the end-to-end process of creating toolkit use cases in Jupyter notebook and transforming the notebook into a HTML file for web integration. Creation of the use case and preparation of the base HTML file is done by the data science (DS) team with some support from the web development (WEB) team. Once the base HTML is ready, the WEB team will style the page and make sure that the use case is made accessible within the website.

Figure 5.1: Stages on how a use case is created and integrated into the website


Discovery & Analysis
Design & Implement
Define Use Cases
Prototype
Exploratory Data Analysis
Detailed Analysis & Visualisation
Web App Integration
Web Integration Collaboration
Refine Web User Experience
Incorporate Interactivity
Refine Visualisation

Members of the DS team develop detailed scenarios using datasets from Melbourne Open Data. They then prototype, provide exploratory and detailed data analysis, visualisation, and interactivity. The DS team then pass on the Jupyter notebooks and converted HTML file to the web team for web app integration.

5.6.1. Creating the use case (Data Science Team)

1. *Define the use case:* The use case that meets the needs of the client is first identified. Based on the usefulness of the data for businesses and the problem statement, the decision to what to work on is made.

Trello: https://trello.com/c/Mf00DFqr, https://trello.com/c/L8IagwuG

2. *Prototype:* The data is then used to generate a prototype to validate if the solution is feasible.

Trello: https://trello.com/c/lMe76Of3, https://trello.com/c/Ksnq078e

3. *Exploratory Data Analysis:* This is where the data is being explored and profiled.

Trello: https://trello.com/c/ghF4u9XP

4. *Detailed Analysis & Visualisation:* The selected use case is then analysed, and interactivity is incorporated into the notebook. Once the notebook is ready, it is checked for a consistent narrative and copy-edited to fit a pre-defined use case template. The narrative must pass copywriting before being converted into a HTML file for integration. The readied HTML file and Jupyter notebook are put into a folder called "datascience/usecases". Only one HTML file and Jupyter notebook per use case can be in the folder.

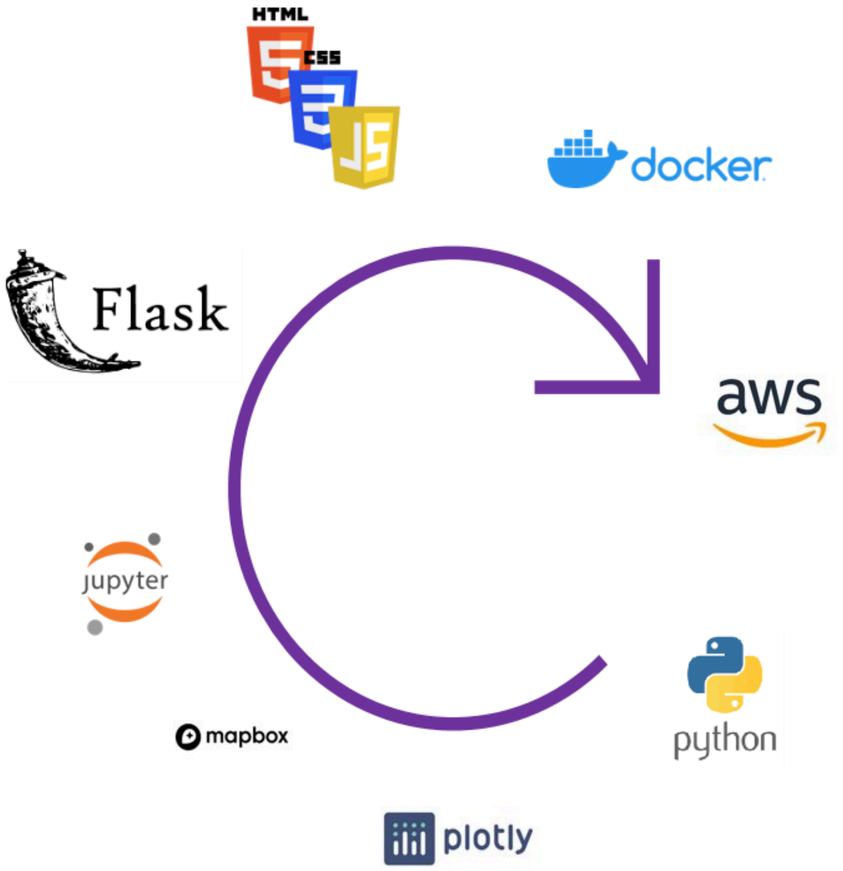
Trello: https://trello.com/c/AebGOLzU, https://trello.com/c/ghF4u9XP, https://trello.com/c/4ov0hbRi, https://trello.com/c/DPg4XWse

Bitbucket: https://bitbucket-students.deakin.edu.au/projects/D2IC-PG/repos/d2i---melbourne-city/browse/datascience

5.6.2. Integrating the use case (WEB Team)

Integration of each use case into the website follows the steps outlined in Figure 5.2 below.

Figure 5.2: Flow of use case integration to the website



1. *Collect latest Jupyter notebook from data science team:* The WEB team ensures that there is one notebook per use case. The notebook is used as a reference to check if the HTML file reflects the contents in the notebook.

2. *Convert into HTML using 'nbconvert' Jupyter library:* The WEB team inspects the HTML file to see if it conforms to the expected template. If it does not, it is sent back to the author for further edits.

3. *Move new HTML into use-cases folder:* A new branch is created based on the name of the use case. In the newly created branch, the HTML file is moved into the webapp's use case folder and renamed into the correct format.

4. *Manipulate HTML into respective Flask template:* The HTML file is converted into a flask template to ensure that the menu bar and page can be accessible on the website.

5. *CSS into header & remove overriding styles:* The HTML file is further styled to ensure that the styling confirms to the overall website's style.

6. *Main content into content block:* The content is moved into a content block so that it can be viewed within the Flask framework.

7. *Package JavaScript to reduce the size of page requests:* As there is a size limit in which a visitor can see the page on AWS infrastructure, the JavaScript code is separated as a different module to reduce the size content.

8. *Update code blocks into code snippet blocks:* The code from the Jupyter notebook is transformed into a code block.

9. *Index content for search portal:* Finally, the index page is being populated with the use case details in the 'search.json' file. A pull request is then created for that branch and once approved, the branch is merged to master. The webapp is then deployed.

Trello: https://trello.com/c/UirlSYJt, https://trello.com/c/B6X7AhFv, https://trello.com/c/rHJUJJMa

5.7. Scalability of Webapp (Josh)

The webapp was growing larger in the number of use cases and logic. To cater for this increased number of use cases, the webapp folder structure was re-designed. Logic files were to a ‘logic’ folder and a new ‘controller’ folder was created to control the routing. Use case templates were now placed in a ‘use-cases’ folder.

Trello:	https://trello.com/c/3DtoM7x6

5.8. GitHub Re-design (Steven, Angus)

A public GitHub repository was established in T2 2021 as a late addition to the project and place to share project collateral with the City of Melbourne open data community. As a result, the repository was organised by Contributor rather than in a structure which lent itself to easier navigation by the community.

Reorganising the GitHub involved migrating from an individual student's account to an organisational GitHub account and a re-organisation and documenting of the content. Readme pages have been setup to aid navigability and a systematic structure to organise open data tools and Jupyter notebooks according to their capability e.g., tutorials, use cases etc.

Many of the Jupyter notebooks have been standardised to ensure a consistent user experience.

Trello:	https://trello.com/c/QxbSNP7K

5.9. ETL Re-design (Josh)

The ETL process for parking availability has been redesigned to collect the parking sensor data daily. The data for the last 28 days is being updated and replaces existing parking sensor data. Previously, this updating of data did not exist. Details of the files involved can be found in the Bitbucket link below.

Trello: https://trello.com/c/QBzJbkEI

Bitbucket: https://bitbucket-students.deakin.edu.au/projects/D2IC-PG/repos/d2i---melbourne-city/commits/b7826467c1aada8f6ac0aa5b2ce935ed08925439

6. Roadmap

6.1. Progress in 2021

The Melbourne Open Data Playground is the result of two years of development and industry collaboration with the City of Melbourne to support greater community and industry use of its published open data sets.

The current implementation of the Melbourne Open Data Playground web application (site) and its associated GitHub repository came to fruition in Trimester 2 of 2021 and has experienced significant enhancements this trimester (T3 2021).

T3 2021 saw three new use cases added plus performance, security, stability, and user experience enhancements that better position the platform for productionisation and community use.

6.2. Future Direction

Having established a process for converting Jupyter notebooks into web application use cases, the platform is now well positioned to realise rapid implementation of future use cases. Future use cases for implementation will be agreed with the City of Melbourne in Week 1 and delivered iteratively over multiple sprints across the remainder of each trimester**.**

In parallel, the web team will continue the work started in T3 2021 and work towards migrating the current web application from AWS to the Google Cloud Platform to align it with both Deakin University and the City of Melbourne preferred cloud infrastructure.

6.2.1. T1 2022 proposed enhancements

In addition to the AWS to GCP migration, the web team will implement further enhancements to the user experience to support mobile device compatibility and use cases accessible through the menu.

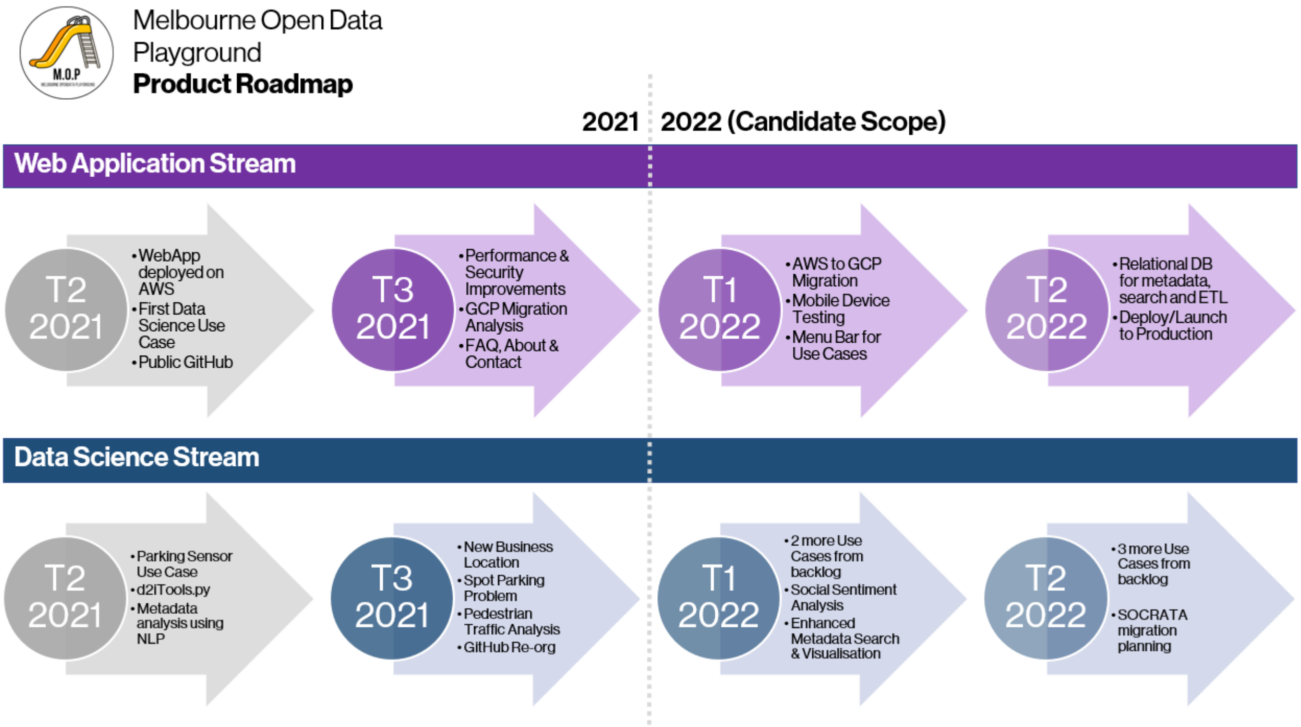
In addition to implementation of selected use cases, the data science team will aim to implement metadata analysis using natural language processing that was initiated in T2 2021 and investigate sentiment analysis for the City of Melbourne Open Data community.

6.2.2. T2 2022 proposed enhancements

In the second half of 2022 the squad could work towards GCP migration finalisation and may need to commence migrating Open Data from SOCRATA to another open data platform provider. Further use cases will be added to the site and a possible migration of flat file data sources into a relational database for improved performance of the search function as well as possible data transformations of Open Data in support of more advanced use cases.

It should be noted that all scope shown for 2022 has been proposed by the D2I Melbourne squad but not yet ratified by the City of Melbourne. This roadmap will serve as a starting point for discussion and agreement of scope with the City of Melbourne. Figure 6.1 below illustrates the most significant deliverables completed and anticipated across 2021 and into 2022.

Figure 6.1: Roadmap for 2021-2022



Melbourne Open Data Playground
Product Roadmap

2021 | 2022 (Candidate Scope)

Web Application Stream

T2 2021
- WebApp deployed on AWS
- First Data Science Use Case
- Public GitHub

T3 2021
- Performance & Security Improvements
- GCP Migration Analysis
- FAQ, About & Contact

T1 2022
- AWS to GCP Migration
- Mobile Device Testing
- Menu Bar for Use Cases

T2 2022
- Relational DB for metadata, search and ETL
- Deploy/Launch to Production

Data Science Stream

T2 2021
- Parking Sensor Use Case
- d2iTools.py
- Metadata analysis using NLP

T3 2021
- New Business Location
- Spot Parking Problem
- Pedestrian Traffic Analysis
- GitHub Re-org

T1 2022
- 2 more Use Cases from backlog
- Social Sentiment Analysis
- Enhanced Metadata Search & Visualisation

T2 2022
- 3 more Use Cases from backlog
- SOCRATA migration planning

7. Open Issues

Induction Processes for Style and Integration Templates

A key deliverable this trimester has been the development of style and integration frameworks. This means consistency in style across use cases and ease of integration of use cases onto the web portal.

The team has made excellent progress in creating notebook templates and examples to follow as well as a step-by-step guide for web integration. That said, the process of web integration requires manual and tactful work by a skilled squad of members. Further, the libraries and content used in the notebooks, which serve as the foundation of the use cases, contribute to the complexity of this task.

Good standardisation of tasks, better induction of the processes of new members, and limiting the use of new libraries or content help make this process even more efficient. Retaining, building upon, and transferring these skills to future squads is a benefit of this standardisation work.

Interactivity Concerns in Notebooks

The spot parking problem use case utilises a third-party service (Binder) to initialise interactivity. The time for initialisation ranges from a few seconds up to a minute. The delay has a negative impact on the user experience and is antithetical to the streamlined product that the tribe is hoping to develop. Future squads may wish to explore alternative techniques to enable this level of interactivity including cloud service hosting of notebooks or other possible solutions.

8. Lessons Learned

The team's biggest challenge this trimester (T3/2021) was the intensive nature of the subject; compressed from 12 weeks down to 6 weeks. Compounding the intense workload was the fact that everyone on the team had different time commitments due to work or other responsibilities.

This initially led to a situation in which meeting as a team, and planning the project was difficult. Unless this issue was rapidly fixed, it would have led to delays in project output. Thanks to the leadership and organisational skills of Albert Hon, and the flexibility of team members such issues were largely avoided.

To further mitigate the challenges inherent in an intensive unit of study, the team made effective use of calendar scheduling and timeboxing. The Agile and Scrum skills of Shalom Chin were crucial to our effective teamwork in this regard. The results of this were:

- Identifying and accommodating squad members' commitments.
- Planning to divide and integrate work.
- Following through with communication and support for one another.

The project planning and scheduling skills learned by the team this semester will be useful next semester.

Another important lesson was the ability to manage both small- and large-scale projects effectively. After identifying the problem scope provided by the client, we split into smaller teams according to our skillsets (Web development and Data Science) overseen by our squad lead.

This division of labour allowed us to develop a multitude of deliverables such as web app improvements, security fixes and the use cases. Once deliverables had been drafted, we would come together as a whole team to discuss integration into the web app, at which point we would discuss the use cases that best fit the scope of the project. We learned to work as part of small (3-4 people) and larger teams (9 people) and manage the production of deliverables effectively to allow ample time for each team to work on the key outputs according to their skillsets.

One challenge that our team was able to overcome quickly was project and tech stack familiarity. Although our team was made up of nine members, only two of those members (Albert and Josh) had worked on the project previously. It was therefore important that the team got up-to-date quickly on what the project was and how the technologies used related to one another.

After receiving guidance on the group project from Albert and Josh, new members of the team began to upskill in the software, packages, and skills needed to better perform within the project. For instance, the data science team had to learn new libraries such as Plotly, mapbox as well as develop the skills to embed interactive elements in the python notebooks.

The web team learned the Flask web development framework and web architectural skills. The team took time to learn how to see the different parts of the framework, and how it all fit together in terms of frontend and backend elements. The members also mentored each other to improve skills more effectively. Sansom-Sherwill, Shalom Chin and Steven James Tuften all provided key mentoring sessions, in the early stages of the project, on tech stacks, agile/scrum management and presentation skills respectively.

Our fast upskilling and project management skills have allowed us to be as productive in terms of deliverables as previous semesters, despite the shortened period of the subject.

9. Product Development Life Cycle

The tribe is split into two squads – the Web Development (WEB) team and the Data Science (DS) team. The WEB team focusses on elements of the project related to the web application. The DS team focuses elements of the project related to the exploration, analysis and development of use cases based on datasets found on Melbourne Open Data.

First, the tribe meets to discuss the overall schema for the project. Second, the web development and data science team work separately in developing their products. Once the initial draft of the product was ready, the tribe comes together again to complete shared tasks such as walkthroughs of the use-case, plans for integration and documentation.

Over time, we have found that a stable and productive way of keeping issues transparent and communication flowing among the different team members and squads.

Trello is being used to establish the goals to be met and tasks to be completed as well as visualise the progress. We use Bitbucket to store our source code. Finally, we use Microsoft Teams for communication of the individual tasks' progress.

The tribe is self-organising, so how the work is distributed and who takes charge of the work is decided by the team members as the Iteration progresses.

9.1. New Tasks

The overall vision of the product is established in the first few meetings to determine the goals to be accomplished at the end of the trimester. Thereafter, Squads are given the autonomy on how they go about creating new tasks.

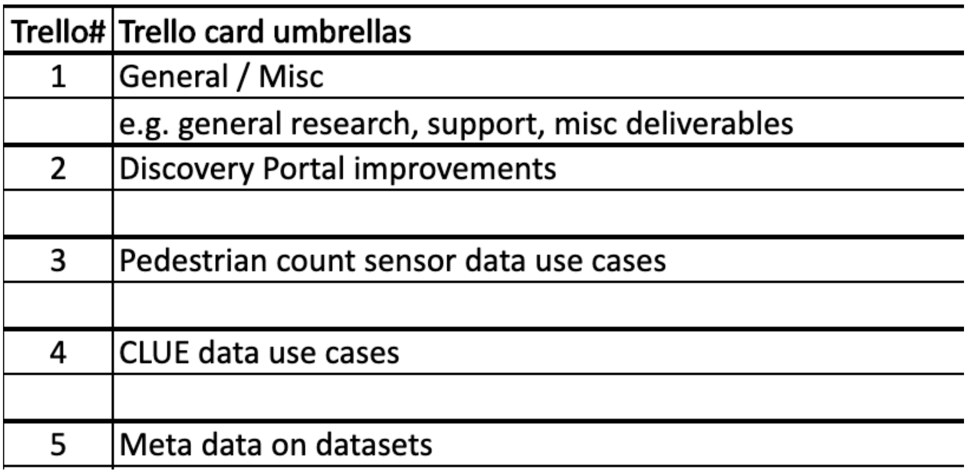
For the DS team, the datasets available on CoM's website are reviewed and new use cases are developed based on the problem statement. The tasks are created on the Trello board and put in the "Product Backlog" stage. It is encouraged that each user story in the board created should follow the template outline in https://trello.com/c/Z3Gvf4Pz.
We mark each user story with the relevant cover picture to distinguish WEB stories and DS stories. Each story is numbered based on the nature of the work to be done.

Project Handover Document (T3 2021)

Figure 9.1: Cover Pictures

Deakin University Industry Capstone
Melbourne Open Data Playground
WEB Team
T3 2021
Project Client
CITY OF MELBOURNE
Deakin University Industry Capstone
Melbourne Open Data Playground
DS Team
T3 2021
Project Client
CITY OF MELBOURNE

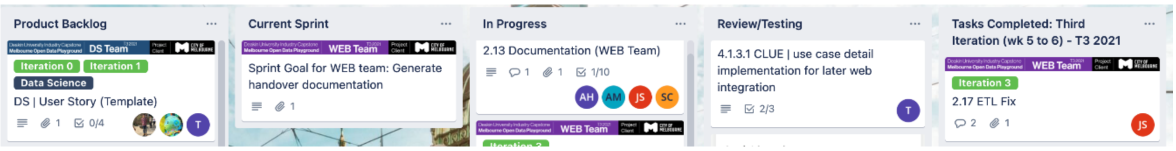
Figure 9.2: Trello Card Numbering



Trello#	Trello card umbrellas
1	General / Misc
	e.g. general research, support, misc deliverables
2	Discovery Portal improvements
3	Pedestrian count sensor data use cases
4	CLUE data use cases
5	Meta data on datasets

Once it is decided that the tasks should be completed in the current iteration, the user story is dragged to the "Current Sprint" stage. All user stories committed to for the iteration are put in the "Current Sprint" at the beginning of each new iteration. The goal is to move the user story through each stage to "Tasks Completed" stage. When a team member is working on a story, he/she puts the user story under the "In Progress" stage. Members who are involved in the user story are added to the card.

Figure 9.2: Trello board stages



Product Backlog
DS Team
Iteration 0
Iteration 1
Data Science
DS | User Story (Template)
1 0/4

Current Sprint
WEB Team
Sprint Goal for WEB team: Generate handover documentation
1

In Progress
2.13 Documentation (WEB Team)
1 1 1/10
AH AM JS SC

Review/Testing
4.1.3.1 CLUE | use case detail implementation for later web integration
2/3
T

Tasks Completed: Third Iteration (wk 5 to 6) - T3 2021
WEB Team
Iteration 3
2.17 ETL Fix
2 1
JS

9.1.1. Trello board stages

The WEB team conducts a Product Backlog Refinement (PBR) and Sprint Planning session to create new tasks. In the PBR session, the team identifies all potential work by splitting them into 5 categories:

1. User Experience – feature that improve the user experience of visitors to the website such as new use cases, better navigation etc.
2. Bugs - outstanding bugs that require fixing
3. Performance - improvements that can be made to speed up the website
4. Product Scalability – improvements that can be made to the maintenance of the webapp.
5. Security - web security of the application

The team prioritises the identified work based on effort and value. Work that is high value and low effort is usually prioritised first. Other factors that might affect this prioritisation includes time-sensitivity. Once the work is selected, it is translated into a user story in "Product Backlog" stage of the Trello board.

The Sprint Planning session is to determine and confirm the user story that will be played in the current Iteration. How the story will be executed is discussed in that meeting. The iteration goal is also established and agreed among the team members. The confirmed stories for the iteration are then put into the "Sprint Backlog" stage.

Communication on the progress of the user story in Teams is established by posting a message with the title of the conversation (Squad name | Trello card number, card name) and the squad banner to make for easy referencing. Members keep the conversation of that topic within that thread so that the team's chat does not become too messy.

Figure 9.3: Squad Banners

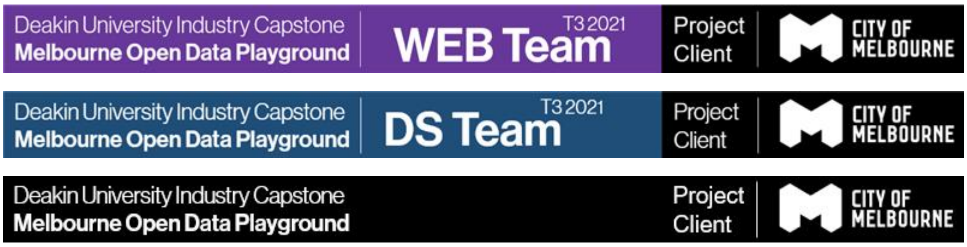
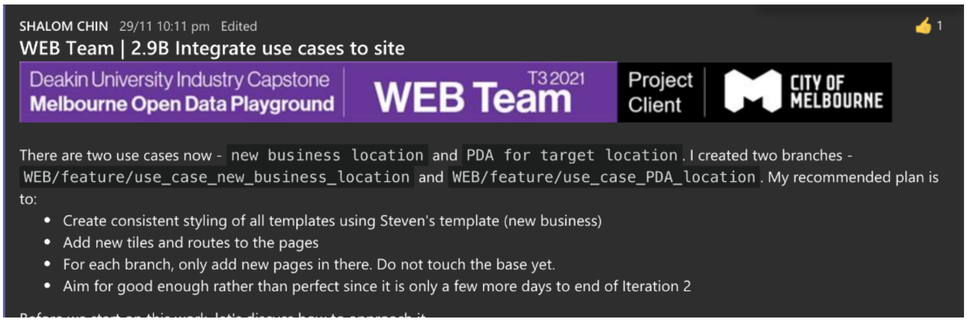

Deakin University Industry Capstone
Melbourne Open Data Playground
WEB Team
T3 2021
Project Client
CITY OF MELBOURNE
Deakin University Industry Capstone
Melbourne Open Data Playground
DS Team
T3 2021
Project Client
CITY OF MELBOURNE
Deakin University Industry Capstone
Melbourne Open Data Playground
Project Client
CITY OF MELBOURNE

Figure 9.4: Example of team communications


Deakin University Industry Capstone
Melbourne Open Data Playground
T3 2021
WEB Team
Project Client
CITY OF MELBOURNE
1

SHALOM CHIN 29/11 10:11 pm Edited

WEB Team | 2.9B Integrate use cases to site

There are two use cases now - `new business location` and `PDA for target location`. I created two branches - `WEB/feature/use_case_new_business_location` and `WEB/feature/use_case_PDA_location`. My recommended plan is to:

- Create consistent styling of all templates using Steven's template (new business)
- Add new tiles and routes to the pages
- For each branch, only add new pages in there. Do not touch the base yet.
- Aim for good enough rather than perfect since it is only a few more days to end of Iteration 2

Before we start on this work, let's discuss how to approach it

Discussion of shared takes between the WEB and DS team takes place in our daily stand-up sessions where new issues requiring every member of the team to be present are raised. Project planning also takes place in this discussion.

9.2. Definition of Done (DoD)

For the DS team's DoD, the tasks must have passed peer review and testing. By addressing all concerns, the task is then merged into master.

The WEB team adopts the same definition as the DS team with an additional stage of once the code has been merged into the master branch, the feature must also be live on the web portal.

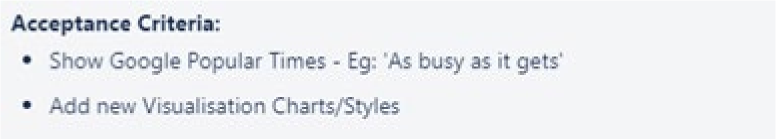
9.3. Task Review

Work review and testing are required before the task is declared to be complete. The individual(s) allocated to the pull request in Bitbucket are responsible of reviewing the work. Once the peer is satisfied with the work, they will approve the work in Bitbucket.

After implementing the changes, the individual would then ask squad-members to re-check their work. The responsible reviewer(s) allocated to the task verify that best practises have been exhibited, in terms of committing on Bitbucket, code quality and comments in Trello. When all the criteria are satisfied, this task is done and moved to the "Tasks Completed" stage.

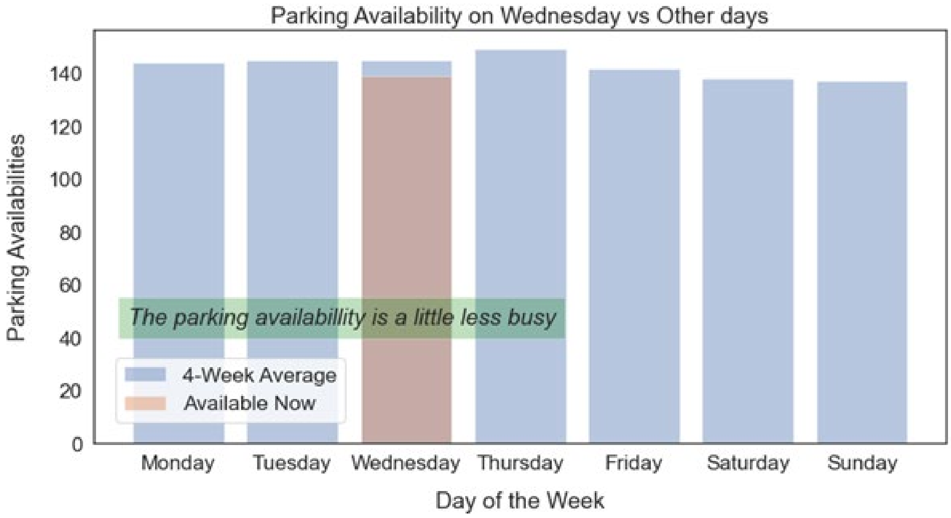
9.4. Testing

Product testing is completed by ensuring that different aspects of the product match the ‘acceptance criteria’ under the relevant Trello card. By comparing the product features to this acceptance criteria, the team can be certain that the product is of high quality. As such, we were able to use Trello effectively to enhance the product feature. An example of this has been shown below:



Acceptance Criteria:

- Show Google Popular Times - Eg: 'As busy as it gets'
- Add new Visualisation Charts/Styles



This process allows the team to be confident that the accomplished product meets the acceptance criteria and is ready to be implemented into the master.

User testing is performed on our local machines before deploying to the website. When the feature is deployed, we will do another test on the live website to make sure the product is deployed as per the acceptance criteria.

9.5. Branching Strategy

Team members are encouraged to create their own branch if they wish to contribute to any work instead of committing directly to master so that wrong commits that might break the web application can be prevented.

The WEB team creates a new branch via Bitbucket for each new feature to be developed upon starting a new user story. The created branch name will be announced in Teams and updated in the description of the Trello card.

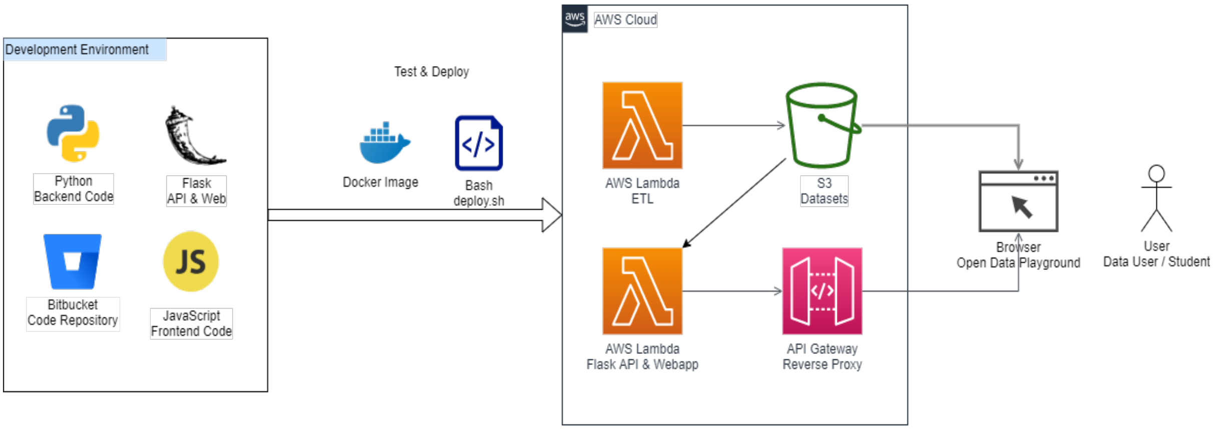
During this project, it was recommended that every tribe member commit their work to the branch after the successful completion and review of allocated tasks. Additionally, it was recommended that relevant comments be made in Bitbucket. This allows other members of the tribe to understand the status of the work completed. The commit message should contain enough high-level information to inform the reader of the new features which have been added and what has changed. This also allows members to understand the progression of the work.

Once the feature has been tested locally and understood to work, the branch is peer reviewed using the pull request function on Bitbucket. After at least one approval, the branch is merged into the master, either by the reviewer or the requestor. All commits made to the branch or to master should be a working copy of the web application. Rebasing of commits is discouraged so that the git history is not unnecessarily re-written.

10. Product Architecture

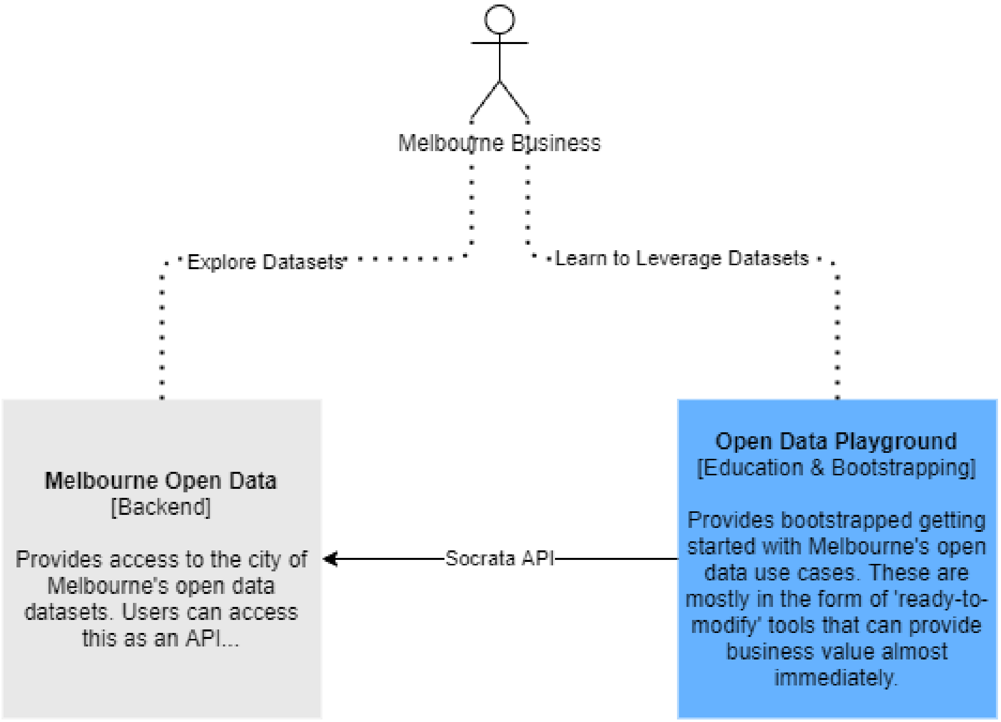
All project technology components and how they relate to each other are illustrated in Figure 10.1 below.

Figure 10.1 Product Component Relations



Our website (the Melbourne Open Data Playground) sits alongside the Melbourne Open Data Platform as an additional tool to educate users, with examples of how to use the platform to derive business insights and integrate Open Data into their existing services. Our website interacts directly with the platform through Socrata to power our tools and examples, as well as providing an enhanced data search function.

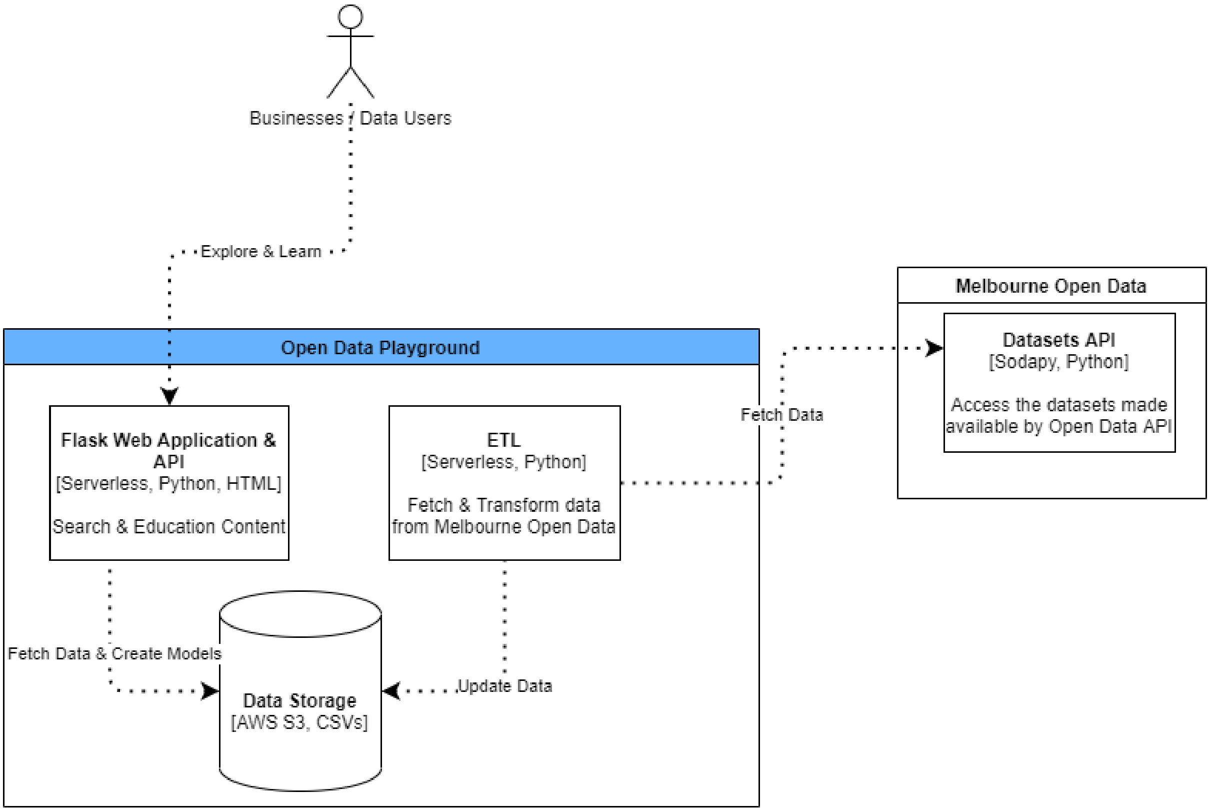
Figure 10.2 Overview of project relations


Melbourne Business
Explore Datasets
Learn to Leverage Datasets
Melbourne Open Data [Backend]
Provides access to the city of Melbourne's open data datasets. Users can access this as an API...
Socrata API
Open Data Playground [Education & Bootstrapping]
Provides bootstrapped getting started with Melbourne's open data use cases. These are mostly in the form of 'ready-to-modify' tools that can provide business value almost immediately.

Project Handover Document (T3 2021)

Figure 10.3 breaks down this process into its component services on AWS.

Figure 10.3 Component Services on AWS


Businesses / Data Users
Explore & Learn
Open Data Playground
Flask Web Application & API
[Serverless, Python, HTML]
Search & Education Content
ETL
[Serverless, Python]
Fetch & Transform data from Melbourne Open Data
Fetch Data & Create Models
Data Storage
[AWS S3, CSVs]
Update Data
Fetch Data
Melbourne Open Data
Datasets API
[Sodapy, Python]
Access the datasets made available by Open Data API

10.1. Technology Stack

The two teams have different workflows and goals along with different technologies and tools. Some common technologies, such as Python and HTML, help with the integration of Data Science team generated use cases into the web application by the WEB Team.

The technologies used by each team are listed below.

10.1.1. Data Science Team

- **General - Python, SodaPy, Pandas, NumPy**: The DS Team works with the powerful, open-source Python language and its well-known utility libraries such as NumPy and Pandas. SodaPy is also used to access Melbourne Open Data through the Socrata API.
- **NLP Libraries - PyLDAvis, NLTK, Spacy, Genism:** Some Natural Language processing libraries may be used for future work in the backlog.
- **Geospatial Libraries - GeoPy, Shapely, GeoPandas:** These libraries are used for the various mapping tools and visualisations used in our use-cases.

- **IDE - Jupyter Notebooks:** Jupyter Notebooks are the tool of choice for data scientists working with Python, allowing us to create and share documents that integrate live code, equations, computational output, visualizations, and other multimedia, along with explanatory text in a single document. Notebooks are created by the DS Team for each use-case, which can be converted to HTML by the WEB Team to display on the site.

10.1.2. WEB Team

- **Frontend - JavaScript, HTML, CSS:** The website content, styling, and functionality such as navbar, search input, and clickable buttons are all coded using these ubiquitous tools.
- **Backend - Python, Flask:** In the backend, the webapp runs on Flask, a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or other components where pre-existing third-party libraries provide these functions.
- **Security - Flask-Talisman:** Talisman is a small Flask extension that handles setting HTTP headers that can help protect against a few common web application security issues.
- **Deployments - Docker, AWS, Google Cloud Platform (GCP):** Docker is the most popular solution for creating and working with containers. The AWS Cloud service *Lambda Functions* uses containers to deploy the webapp. There is a similar service in GCP called *Cloud Run* that we will use when migrating from AWS to GCP. There are many other services available in these Cloud Providers that we can use as we expand the project.
- **IDE - Visual Studio Code:** Visual Studio Code is a lightweight code editor which the WEB Team uses for coding in the webapp. It allows for running the Flask server in a local development environment so we can see changes live in a browser, and it also has several useful extensions for working with Git, debugging, etc.

11. Source Code Management

All source code for the project is managed through the team's Deakin Bitbucket account and repository for **"D2I – Melbourne City"**. This Bitbucket repository is also a staging area for all changes to the Melbourne Open Data Playground Public GitHub Repository.

All non-source code files or artefacts may also be stored in the teams MS Teams Channel Files folder under the directory named after your trimester e.g., **"T1 2022"**. The MS Teams File folder should be used for MS Office documents, video, audio, images, or other assets not forming part of the web application or Public GitHub deliverable.

Git is used to manage your local cloned copy of both the Bitbucket and GitHub repository. SourceTree may optionally be used for managing git commands via a Graphical User Interface.

Source code for the web application is developed in Microsoft Visual Studio Code.
Jupyter Python Notebooks may be designed and developed in either Jupyter Notebook or Google Colab.

Figure 11.1 below illustrates the software tools used locally and remotely for managing source code.

Table 11.1 describes the primary top-level folders in the Bitbucket account.
Table 11.2 describes the primary top-level folders in the GitHub account.

Figure 11.1: Source Code Management & Version Control tools

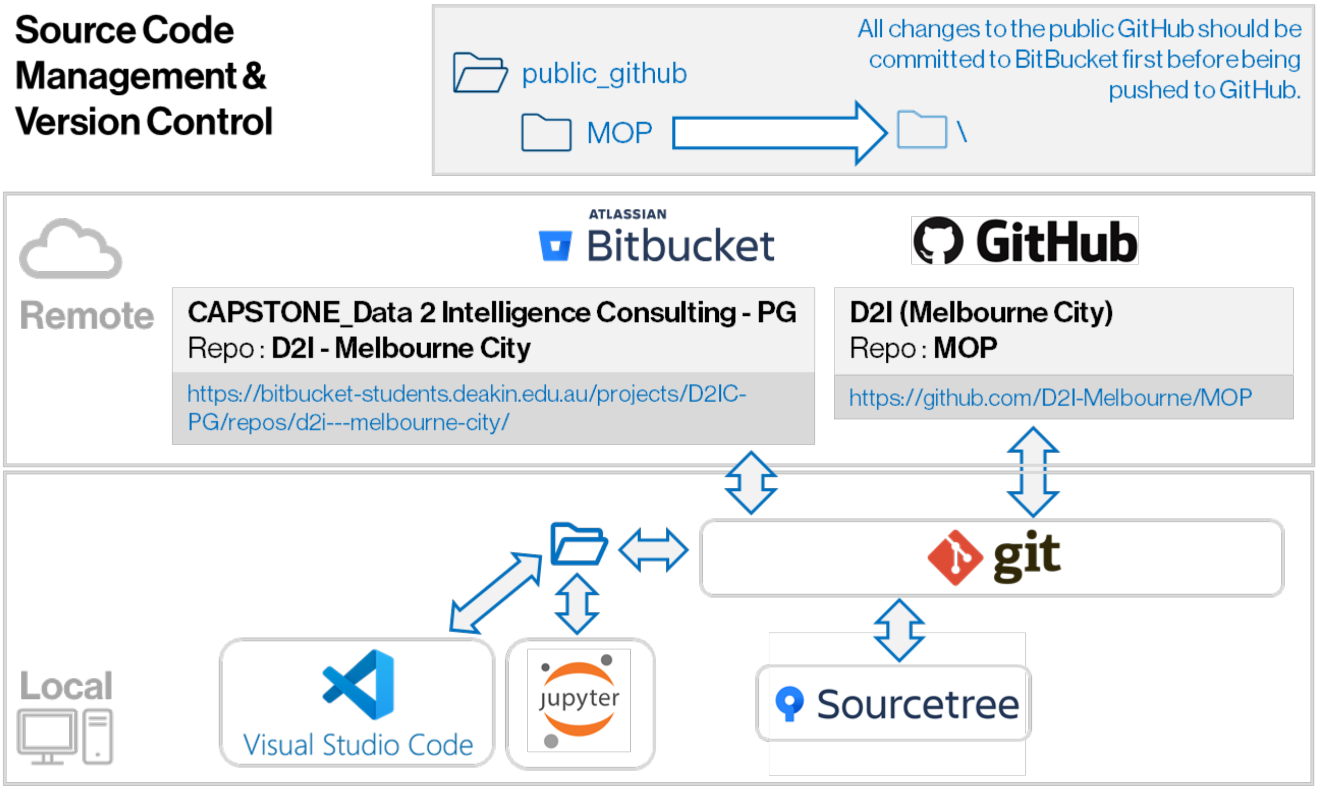

Source Code Management & Version Control
public_github
MOP
\
All changes to the public GitHub should be committed to BitBucket first before being pushed to GitHub.
Remote
ATLASSIAN Bitbucket
GitHub
CAPSTONE_Data 2 Intelligence Consulting - PG
Repo : D2I - Melbourne City
https://bitbucket-students.deakin.edu.au/projects/D2IC-PG/repos/d2i---melbourne-city/
D2I (Melbourne City)
Repo : MOP
https://github.com/D2I-Melbourne/MOP
git
Local
Visual Studio Code
jupyter
Sourcetree

Table 11.1: Bitbucket folder structure

Folder	Description
data science	All artefacts produced by the data science sub team.
ETL	Code for data transformations relating to moving and accumulating open data in the web app.
FAQ Document files	Files related to publishing the FAQ in the Web App.
google_cloud_test	Files for testing connectivity and use of docker with Google Cloud Platform.
Playground	Location for storing individual WIP files not yet ready for assembling into team deliverables.
Prior Work	Artefacts from prior trimesters.
public_github\MOP	Contains a copy of the contents of the GitHub folder for the MOP repository.

Folder	Description
	The MOP Repo is replicated here since Bitbucket is the only source control system recognised by the Deakin DISC program. Thus, all changes to the public GitHub repo should be committed in this subfolder first before being pushed to the public GitHub folder.
webapp	Artefacts created by the web sub team for implementation of the Melbourne Open Data Playground web application.

Table 11.2: GitHub folder structure

Folder	Description
example_notebooks	All example Jupyter notebooks are stored under this folder.
\condaconfiguration	Contains instructions for setting up a Conda Environment with the required packages for use case development.
\dataanalysis	Contains Jupyter notebooks for exploratory data analysis of City of Melbourne Open Data Sets.
\tutorials	Notebooks demonstrating specific data engineering and data science techniques.
\usecases	Notebooks corresponding to the use cases published in the Melbourne Open Data Playground web application.
images	Images used on the home Readme page are located here.
opendataapi	Open Data API python command line functions are in this folder.

12. Login Credentials

In this trimester we leveraged AWS Cloud infrastructure service to support development and deployment of the Melbourne Open Data Playground. As such there is an AWS root account, we have created that can be transferred between squads in the tribe. The credentials for this account can be acquired by getting in contact with one of this trimester's junior members (will likely be your senior members):

- Angus Maiden
- Steven Tuften
- Alex Vuong
- Shalom Chin
- William Hebblewhite

There are no other passwords required for working with the web application. Rather, access to AWS resources should be given through AWS IAM accounts, roles, groups, and policies.

13. Appendices

13.1. Project Links

[1] Link to handover artefacts folder where this document is stored

https://deakin365.sharepoint.com/:f:/r/sites/Data2IntelligenceConsulting/Shared%20Documents/D2I%20(Melbourne%20City)/Handover%20and%20Showcase/T3%202021?csf=1&web=1&e=czBKhu

[2] Link to the team's pitch video

https://deakin365.sharepoint.com/:v:/r/sites/Data2IntelligenceConsulting/Shared%20Documents/D2I%20(Melbourne%20City)/Handover%20and%20Showcase/T3%202021/T3%202021%20Showcase%20Video.mp4?csf=1&web=1&e=pDBwWO

[3] Link to our Trello board (T3/2021)

https://trello.com/b/sxBfi5DY/d2i-melbourne-city-trello

[4] Link to our Trello Roadmap (T3/2021)

https://trello.com/b/unvtjPUX/d2i-melbourne-city-roadmap

[5] Link to Bitbucket repository (T3/2021)

https://bitbucket-students.deakin.edu.au/projects/D2IC-PG/repos/d2i---melbourne-city/browse

[6] Link to the live website (T3/2021)

https://6pdglgxshl.execute-api.ap-southeast-2.amazonaws.com/Prod/

[7] Link to the GitHub site

https://github.com/D2I-Melbourne/MOP