Fleming Flamingos
Development and GIS Solutions

FINAL REPORT

Development Activity Tracker – City of Toronto (2221)

Authors

Terrie-Ann Broomfield, Xin Wen, Peggy Wong

GIS Application Specialist and GIS Cartographic Specialist

Client

Scott Whynot

City of Toronto | Graphics and Visualization Supervisor

Advisor

Kendra Chalmers

Sir Sandford Fleming College | Instructor

June 22, 2022

This final report is presented in partial fulfilment of the academic requirements for APST62, GIS Collaborative Project course, Fleming College.

Abstract

Toronto is the largest city in Canada and the fourth largest in North America. Development activities take place daily which makes tracking these activities difficult without an available map illustrating the process. The City of Toronto presented this issue and our team set out to provide a solution to track all the development activities within the city using Geographic Information System technology. The data acquired from the City of Toronto Open Data Portal, was published as hosted feature layers through ArcGIS API for Python to create a web map symbolizing unique values for each development application, and a heat map to show the development hotspots. The web maps were made in compliance with Accessibility for Ontarians with Disabilities Act, thus people with colour deficiencies will be able to differentiate between the application types and clearly understand the information illustrated. A dashboard was generated from the web maps to display the statistics of development activities throughout the city. A web application and a story map were also created to further enhance the interactions of the users. A maintenance script was created to allow the hosted feature layers to be updated automatically, minimizing the maintenance effort by the client. Finally, a setup guide was created to provide step-by-step instructions to set up web maps and automated updates.

Keywords: ArcGIS, Canada, City, Development, Planning, Toronto

Table of Contents

List of Figures and Tables

Terms of Reference

ArcGIS API for Python: An Application Programming Interface (API) developed by Esri to perform GIS visualization, analysis, data management, and GIS system administration tasks

ArcGIS Dashboards: An ArcGIS Online application to convey information by presenting location-based analytics using intuitive and interactive data visualizations on a single screen.

ArcGIS Experience Builder: An ArcGIS Online application to quickly transform your data into compelling web apps without writing a single line of code.

ArcGIS Map Viewer: An ArcGIS Online application to create interactive web maps.

ArcGIS Notebooks: An ArcGIS Online application to perform analysis, automate workflows, and immediately visualize data and analysis results in a geographic context.

ArcGIS StoryMaps: An ArcGIS Online application to create remarkable stories that give the maps meaning.

Accessibility for Ontarians with Disabilities Act (AODA): A law that sets out a process for developing and enforcing accessibility standards.

Geographic Information System (GIS): A system that manage and analyze geographical data to solve spatial problems.

Hosted feature layer: The layer that have been published to ArcGIS Online. The feature data in these layers is hosted by, or stored on, ArcGIS Online.

Manage Notebook Code Dependencies (MNCD): A tool developed by Esri that manages cached Python “Code Sample” and “Notebook” items within the notebooks.

Modified Transverse Mercator (MTM): A coordinate system that is common in Eastern Canada. It uses a transverse Mercator projection with zones spaced 3° of longitude apart.

North American Datum 1927 (NAD27): A horizontal datum that is based on the Clarke Ellipsoid of 1866.

Overwrite Hosted Feature Services (OverwriteFS): A python script developed by Esri to overwrite a feature service. It can be leveraged by the Notebook environment and automatically managed by the MNCD Tool.

1. Introduction

City development is a dynamic process which plays an important role on sustainable growth of modern cities. As the largest city in Canada, the amount of development activities in Toronto is extremely high. As of June 2022, there were 58,864 development applications received by the City of Toronto since 2008 (City of Toronto, 2022). The high amount of development activities can be daunting if people were to look for developments in a certain area without a map. Thus, the City of Toronto was interested in mapping all the development activities within the city, allowing the public to access information of development applications interactively.

1.1 Previous Work

The City of Toronto has developed a web application for development applications in the past (https://secure.toronto.ca/AIC/index.do). The web application allows the users to perform a quick search by entering an address or clicking anywhere on the map, with the ability to filter results. However, it can only display the development applications within a specific radius, and the spatial pattern of the development activities cannot be revealed from the map.

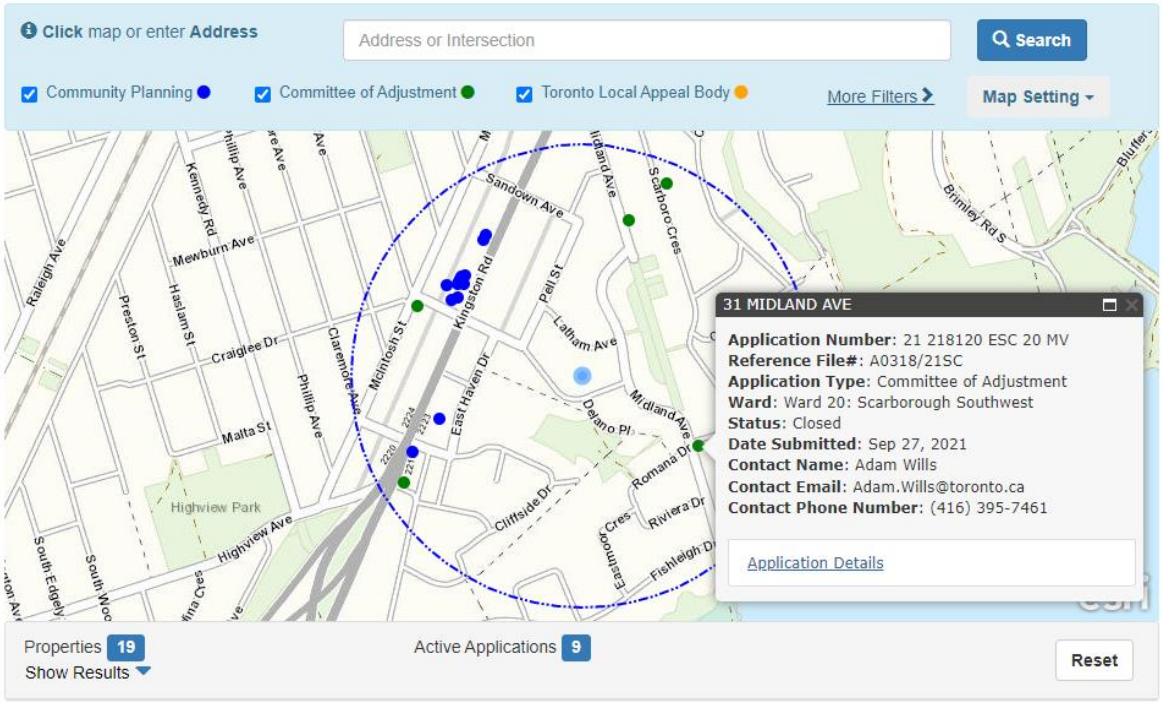

Figure 1 Web application of development applications developed by the City of Toronto

The City of Kenosha, Wisconsin in the United States has also developed an interactive dashboard for the overall development in the city (https://www.arcgis.com/apps/dashboards/c1620b1dfaec4c5fa6897df5f74e1a90). The dashboard displays a map and some statistical graphs of the development projects that have been reviewed by Kenosha's Department of City Development since 2018, providing a visual representation of the development activities and statistics at a glance.

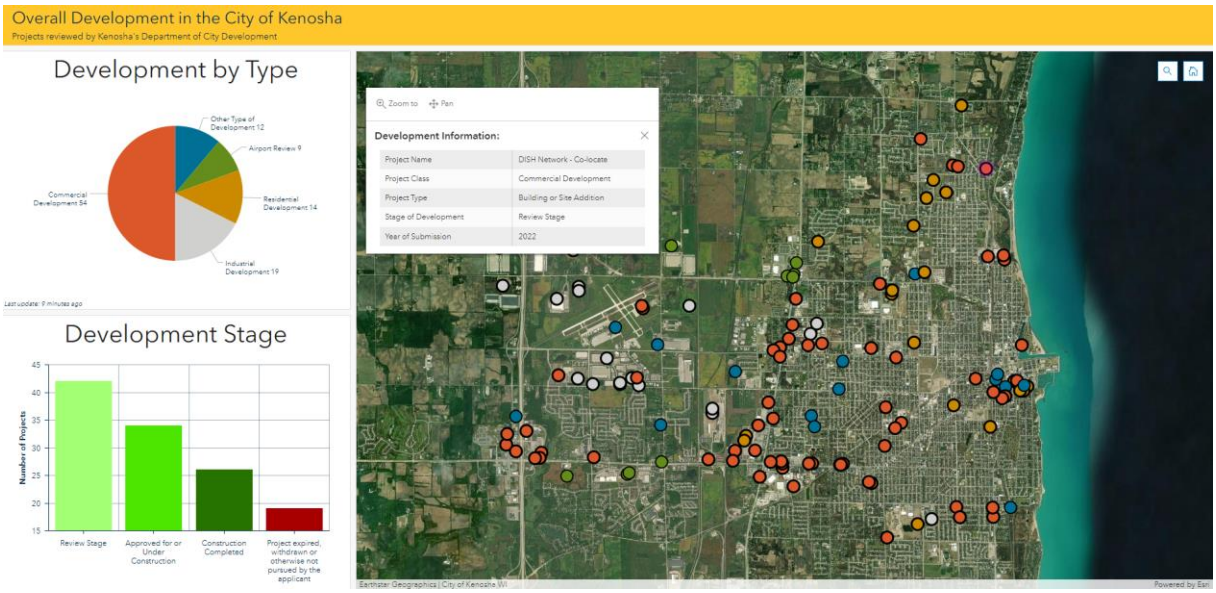

Figure 2 Dashboard of overall development developed by the City of Kenosha

1.2 Client

The client for this project is the City of Toronto, represented by Scott Whynot, the Graphics and Visualization Supervisor. Dulini Ratnayake, the Graphics and Visualization Program Manager, and Harrison Thomas, the Graphics and Visualization Supervisor, have also participated in the meetings to discuss the project and provide feedbacks.

1.3 Study Area

Toronto, the capital of the province of Ontario, is located on the north-western shore of Lake Ontario. It is the largest city in Canada and the fourth largest city in North America, with a population of about 2.9 million. People have lived in Toronto since shortly after the last ice age, where the first urban community was established as the Town of York in 1793. Through years of

development, Toronto has emerged as one of the most liveable and multicultural urban places in the world today (City of Toronto, 2022).

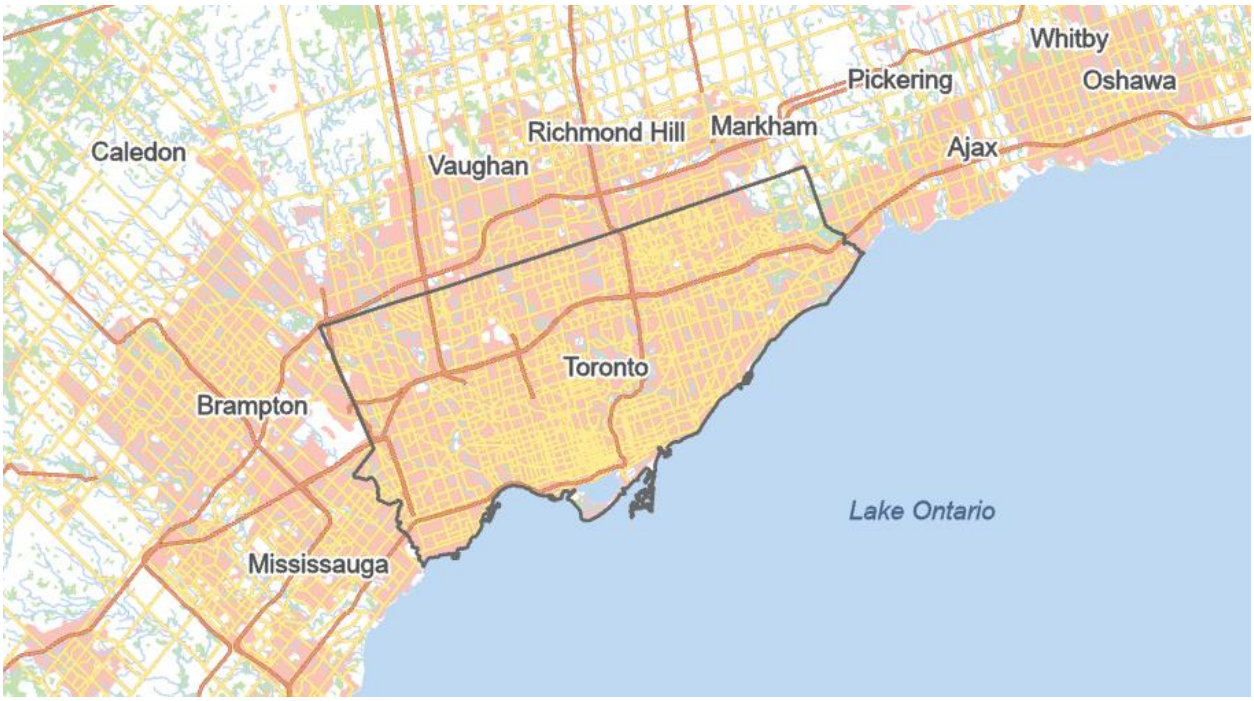

Figure 3 Study area extent (black outline) at a scale of 1:450,000

1.4 Problem Definition and Objectives

Currently, the development activity dataset can be downloaded from the City of Toronto Open Data Portal in non-spatial formats only. The current web application developed by the City of Toronto has several limitations, such as displaying the development applications within a specific radius only. Thus, the main goal of this project was to alleviate the difficulties when researching development activities in Toronto by providing a visual presentation of the process of development activities, with the following objectives to meet the project requirements:

- **To track all the development activities in Toronto** using ArcGIS Map Viewer, ArcGIS Dashboards, ArcGIS Experience Builder, and ArcGIS StoryMaps, including all currently active (open) and inactive (closed) Community Planning applications, Committee of Adjustment applications and Toronto Local Appeal Body appeals received by the City of Toronto since January 01, 2008 for public queries.

- **To promote effective communication** using a visually engaging design while meeting the standards of the Accessibility for Ontarians with Disabilities Act (AODA). AODA is an Ontario law mandating that organizations must follow standards to become more accessible to people with disabilities (Kovac, 2018). All deliverables should be designed strategically to provide information to the public in a user-friendly manner.
- **To automate the updating process** where the data can be synchronized with the City of Toronto Open Data Portal on a daily basis, minimizing the effort of maintenance by the client.

1.5 Literature Review

To have a better understanding about the project, a literature review was conducted on the development application process in Toronto and Accessibility for Ontarians with Disabilities Act (AODA) during the planning stage of the project. The findings from the literature review are provided in the Appendix A and the Appendix B.

2. Methodology

To meet the client requirements and the project objectives, a series of web solutions were developed to visualize the development activities in Toronto. Although the client initially requested a web map and a corresponding dashboard only, our team decided to expand the project scope and worked on a web application and a story map as well, allowing the public to access the information in multiple ways. ArcGIS API for Python was also used to publish the web maps and manage the automation process. The chart below illustrated the main components of this project:

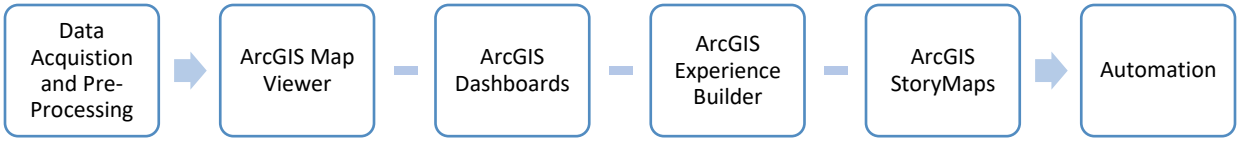

Figure 4 Main components of the project

2.1 Data Acquisition and Pre-Processing

The first step of the project was data acquisition and pre-processing, which was implemented using ArcGIS API for Python. ArcGIS API for Python is a powerful, modern Pythonic library for performing GIS visualization, analysis, data management, and GIS system administration tasks (Esri, 2022). The script was hosted and run through ArcGIS Notebooks, where the Standard license was sufficient for this project. Although the latest version of ArcGIS API for Python was 2.0.1 at the time of project implementation, the version 1.9.1 was used for this project as the version 2.0 was not available at Fleming College. The classes and methods that were used in the scripts can be found in the API Reference for the ArcGIS API for Python (Esri, 2022). The data acquisition and pre-processing involved creating connection to GIS environment, creating folder for project items, publishing hosted feature layers, sanitizing problematic data, adding geometries to existing features, and sharing items. Here is the workflow of data acquisition and pre-processing:

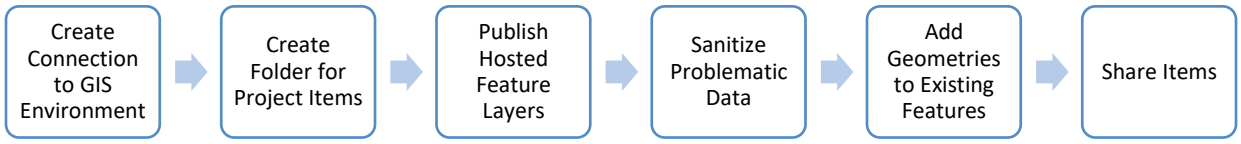

Figure 5 Workflow of data acquisition and pre-processing

Note: Since the June 2022 update after project completion, ArcGIS Online can no longer support publishing hosted feature layers with unsupported coordinates formats from the source file directly (e.g. development applications dataset). The workaround is to publish a hosted table layer from the source file (as an intermediate layer) and transfer features to an empty hosted feature layer using Spatially Enabled DataFrame. The following workflow is outdated but the scripts has been modified to reflect the updates.

2.1.1 Data Resources

The client has provided all the required datasets through the City of Toronto Open Data Portal and the City of Toronto ArcGIS Server. Development Applications lists all currently active and inactive development applications received by the City of Toronto since January 1, 2008, which is updated on a monthly basis. Regional Municipal Boundary outlines the geographical administrative boundary of the City of Toronto. Historical Aerial Imagery is a black and white

aerial photography of Toronto captured in 1939, which was applied to the story map only for educational purposes. Here is a list of the datasets used in this project:

Name	Source	Format	Projection	URL	Last Update
Development Applications	City of Toronto	CSV, JSON, XML	NAD 27 – MTM Projection	https://open.toronto.ca/dataset/development-applications/	2022-06-22
Regional Municipal Boundary	City of Toronto	SHP	WGS84	https://open.toronto.ca/dataset/regional-municipal-boundary/	2019-07-23
Historic Aerial Imagery - 1939	City of Toronto	Map Server	WGS84	https://gis.toronto.ca/arcgis/rest/services/basemap/cot_historic_aerial_1939/MapServer/	2018-07-27

Table 1 List of datasets used in the project

2.1.2 Create Connection to GIS Environment

To get started, the script was connected to the GIS environment by creating a GIS object. GIS object is the "entry point" in the Python script when using the ArcGIS API for Python, allowing the users to manage the resources on ArcGIS Online or ArcGIS Enterprise. It was created by passing in the URL and the login credentials using the following syntax:

```
arcgis.gis.GIS(url, username, password, key_file, cert_file, verify_cert,
                    set_active, client_id, profile)
```

The URL should be a web address to either an ArcGIS Enterprise Portal or to ArcGIS Online, where ArcGIS Online is used if no URL is specified. The currently logged-in user's credentials or anonymous access is used if the username or password are not provided. Since we wanted to use the currently logged-in user's credentials to login to ArcGIS Online, the URL and the login credentials were not specified in the script.

2.1.3 Create Folder

The create_folder method creates a folder with the given folder name for the given owner. To keep the project items organized, a folder called "Development Activity Tracker" was created in the Portal using the create_folder method:

```
create_folder(folder, owner)
```

All the generated items, such as hosted feature layers and web maps, were stored in the generated folder.

2.1.4 Publish Hosted Feature Layers

Once the folder was created, the data and the hosted feature layers were added and published. Here is an overview of the workflow of publishing hosted feature layers:

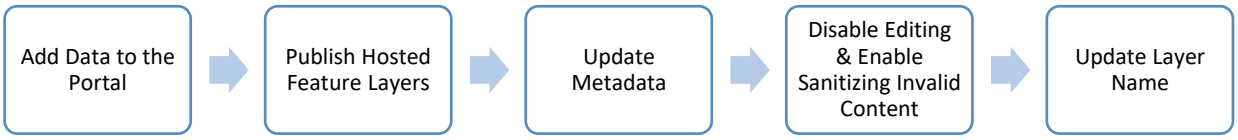

Figure 6 Workflow of publishing hosted feature layers

2.1.4.1 Add Data to the Portal

As Historical Aerial Imagery is already hosted on the City of Toronto ArcGIS Server, it can be added to the web map directly without publishing hosted feature layer. However, it is not the case for Development Applications and Regional Municipal Boundary, as the data source is the City of Toronto Open Data Portal. Thus, Development Applications and Regional Municipal Boundary were loaded to the Portal directly, using the add method:

```
add(item_properties, data, thumbnail, metadata, owner, folder, item_id)
```

2.1.4.2 Publish Hosted Feature Layers

The publish method is used to publish a hosted service based on an existing source item. Once the datasets were added to the Portal, the hosted feature layers were published using the publish method:

```
publish(publish_parameters, address_fields, output_type, overwrite,
     file_type, build_initial_cache, item_id, geocode_service)
```

2.1.4.3 Update Metadata

To make the hosted feature layers easier to find and understand, metadata were added using the update method, where the dictionary of the item properties was provided:

```
update(item_properties, data, thumbnail, metadata)
```

2.1.4.4 Disable Editing & Enable Sanitizing Invalid Content

Since the hosted feature layers were initially editable and shared to the public, the editing capabilities had to be turned off to prevent unwanted editing from the public. Also, some features in the development applications dataset contains invalid HTML content, which may cause errors when adding geometries later.

```
7174353,,CO,2015-07-10,"withdrawn ?

Wed 09/30/2015 10:19 AM - email - Tony Evangelista <aevangel@sympatico.ca>
Good Morning Sai-Man,

Further to a voice mail message I left for you yesterday please accept this email as authorization to

Thank you

Regards,

Tony
",,M2N,,Tentatively Scheduled, ,DUNFOREST,288,AVE,313150.859,4848669.17,15 188452 000 23 CO,B0027/15NY
```

Figure 7 Feature with invalid HTML content

Thus, the definition properties of the hosted feature layers were updated using the update_definition method, at a hosted feature layer collection services level:

```
update_definition(json_dict, future)
```

The capabilities were set to Query only to turn off editing capabilities, and the XXS input rule was set to “sanitizeInvalid” to encode invalid HTML content and prevent invalid HTML errors.

2.1.4.5 Update Layer Name

Since the default names of the hosted feature layers may not be interpretable for the users, the names of the hosted feature layers were updated using the update_definition method, but at a hosted feature layer level only.

2.1.5 Sanitize Problematic Data

Before adding geometries to the existing features in the development applications dataset, the problematic data was first sanitized. There were three types of problematic data identified in the dataset: invalid HTML content, invalid X and Y coordinates, and invalid dates. Since we have already updated the service definitions to sanitize invalid HTML content, the invalid X and Y coordinates and the invalid dates were sanitized manually using the edit_features method.

```
edit_features(adds, updates, deletes, gdb_version, use_global_ids,
rollback_on_failure, return_edit_moment, attachments, true_curve_client,
session_id, use_previous_moment, datum_transformation, future)
```

2.1.5.1 Invalid X and Y coordinates

Some features in the development applications dataset have incorrect X and Y coordinates that are out of range and not located in Toronto. At the time of project implementation, there were 57 features located in the Tug Hill region of the United States, and 1 feature located near Columbia. To prevent the features with incorrect coordinates being visualized, the X and Y coordinates of those features were set to null using the edit_features method.

Figure 8 Map with incorrect X and Y coordinates (in red borders)

2.1.5.2 Invalid Dates

Some features in the Development Applications dataset have incorrect dates.

1	_id	APPLICATION#	APPLICATION_TYPE	DATE_SUBMITTED	DESCRIPTION	HEARING_DATE	POSTAL
17294	7139076		MV	2008-10-29	The applicant is s	3008-12-03	M1C
21977	7143759		MV	2010-06-11		1010-07-02	M3H
29822	7151604		MV	2010-01-27	To construct a ne	0010-06-03	M8W
29823	7151605		MV	2010-01-27	To construct a ne	0010-06-03	M8W

Figure 9 Features with incorrect dates (in red border)

If the incorrect dates were not sanitized, the dates of all the other features will be affected and changed to 1970-01-01 in the Spatially Enabled DataFrame when adding geometries. Therefore, all incorrect dates were set to null using the edit_features method to ensure data accuracy.

	F_id	APPLICATION_	APPLICATION_TYPE	DATE_SUBMITTED	DESCRIPTION	HEARING_DATE	POSTAL	REFERENCE_FILE_	STATUS
0	7121784	None	MV	2022-01-19	To construct a two storey rear addition and de...	1970-01-01 00:27:29.894400	M4G	None	Closed
1	7121785	None	MV	2022-03-30	To construct a new SFD w/ a building height fo...	NaT	M8W	None	Postponed
2	7121786	None	MV	2022-03-30	To construct a new SFD w/ a building height fo...	NaT	M8W	None	Postponed
3	7121787	None	MV	2022-03-30	To convert the ground and second floors of the...	1970-01-01 00:27:41.904000	M5R	None	Accepted
4	7121788	None	MV	2022-03-30	Proposal to construct a new dwelling.	1970-01-01 00:27:34.732800	M3B	None	Hearing Scheduled

Figure 10 Incorrect hearing dates in the Spatially Enabled DataFrame

2.1.6 Add Geometries to Existing Features

The City of Toronto's operational coordinate system is currently the North American Datum 1927 (NAD27) Modified Transverse Mercator (MTM) projection (City of Toronto, 2022). Since ArcGIS Online does not support MTM projection, the features cannot be visualized without obtaining the Spatially Enabled DataFrame with geometries. Thus, the Spatially Enabled DataFrame was obtained by specifying the X and Y columns and the spatial reference, which is 7991 for NAD 27 MTM Zone 10 (Esri, 2021), using the from_xy method:

```
from_xy(df, x_column, y_column, sr)
```

The geometries were then added to the existing features using the Spatially Enabled DataFrame, using the edit_features method. Since the server may not be able to handle large

dataset, the Spatially Enabled DataFrame was sliced with a batch size of 250 to ensure service stability. General error handling was also added to the script, allowing the script to scan through the individual features in the problematic batches and pass the individual problematic features only.

2.1.7 Share Items

To enable the public to view the hosted feature layers through ArcGIS Map Viewer, the hosted feature layers were shared to everyone, using the share_items method:

```
share_items(items, everyone, org, groups, allow_members_to_edit)
```

2.2 ArcGIS Map Viewer

ArcGIS Map Viewer is an ArcGIS Online application that allows the users to create interactive web maps to tell stories and answer questions. Main Map, Heat Map, and Historic Aerial Imagery Map were created using ArcGIS Map Viewer. The following table illustrated the web maps created in this project, and the relevant layers used in these web maps:

Web Map	**Layers**
Development Activity Tracker – Main Map	- Development Applications - Regional Municipal Boundary
Development Activity Tracker – Heat Map	- Development Applications - Regional Municipal Boundary
Historic Aerial Imagery – 1939	- Historic Aerial Imagery – 1939

Table 2 Web maps and relevant layers

Main Map is the main deliverable for this project, which visualizes all currently active and inactive Community Planning applications, Committee of Adjustment applications, and Toronto Local Appeal Body appeals received by the City of Toronto since 2008. On the other side, Heat Map is a supplemental map which visualizes the magnitude of development activities in different locations, helping the users to understand the patterns of the development activities and identify the development hotspots in Toronto. Both maps were used in the dashboard, the web application, and the story map. Historic Aerial Imagery Map was applied to the story map

only for educational purposes, allowing the users to compare the historical aerial imagery with the current aerial imagery.

2.2.1 Base Map

A basemap provides background context for a map, where layers can be added on top of it. For this project, Topographic was selected as the basemap for all the web maps. Topographic is a basemap designed by Esri which provides an appropriate amount of context information such as roads and labels, without having overwhelming details.

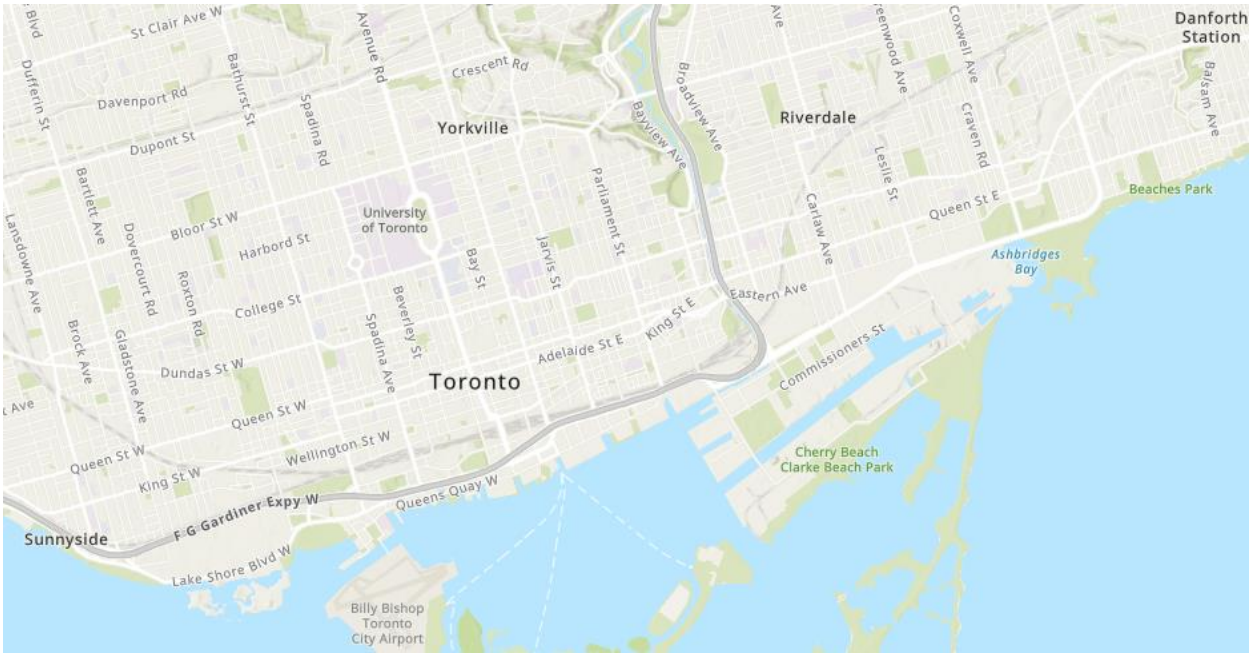

Figure 11 Topographic basemap

2.2.2 Symbology

Since the people with colour deficiencies and the people without colour deficiencies see colours differently, it was challenging to select a colour palette that works well for both types of people. Therefore, Masataka Okabe and Kei Ito purposed a colour palette that is unambiguous to both people with colour deficiencies and people without colour deficiencies using scientific methods (Okabe & Ito, 2002). This colour palette was applied to the application types in the main map.

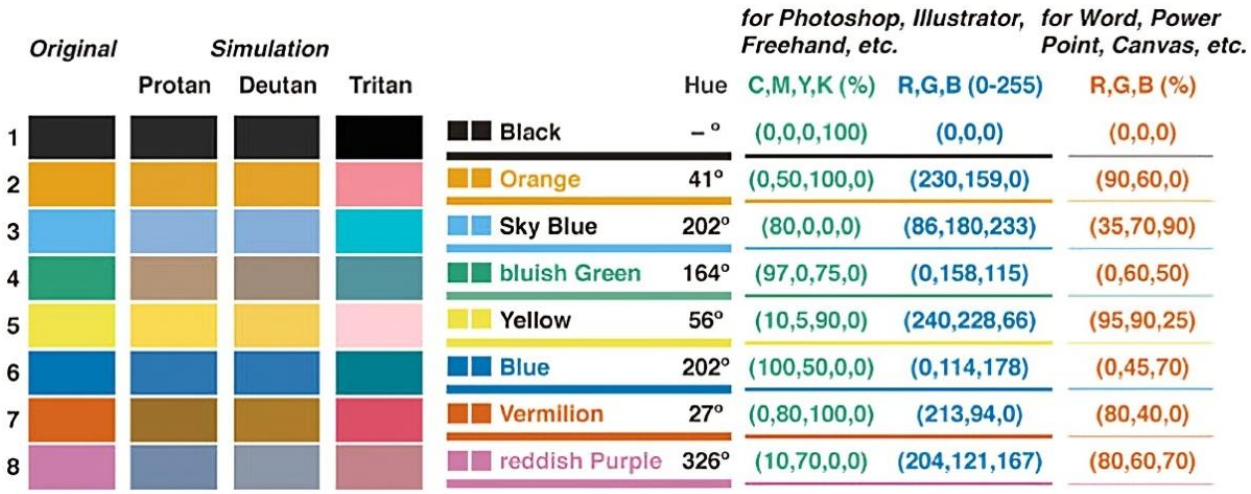

Figure 12 Colour palette unambiguous to both people with and without colour deficiencies

For the heat map, purple (115, 0, 115) was selected for the low values and yellow (225, 234, 0) was selected for the high values. Based on the colour palette generated by David Nichols' tool (https://davidmathlogic.com/colorblind/), the two colours work well for people with colour deficiencies and people without colour deficiencies.

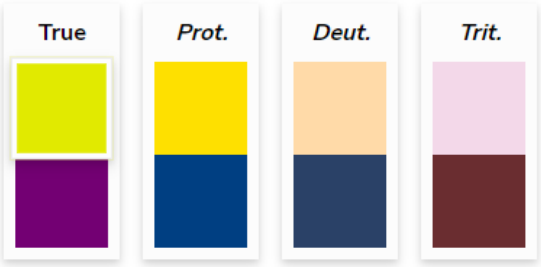

Figure 13 Colour palette for heat map renderer, generated by David Nichol's tool

2.2.3 Clustering

Since there are over 70,000 development activities in Toronto since 2008, the main map may look disorganized with the overlap of the features. The loading speed may also be reduced as more information is processed simultaneously. Thus, clustering was enabled in the main map to display a large number of features. The cluster labels and the cluster pop-ups were configured to provide information about the clusters. As the application type values in the dataset are abbreviations only, an Arcade expression was applied to define rendering properties for the application type attribute.

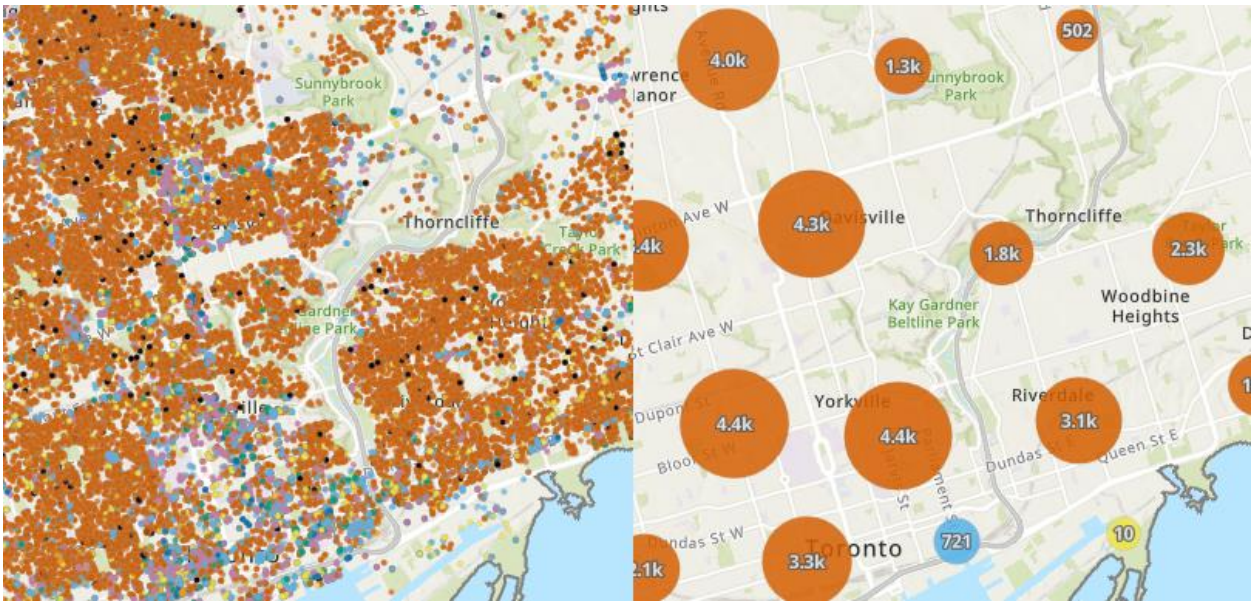

Figure 14 Main map without clustering (left) and with clustering (right)

2.2.4 Pop-Ups

The pop-ups of the individual features in the main map were configured to display information about the development application, such as application type and submission date. As the application type values in the dataset are abbreviations only, and the reference file number values, hearing date values, and the description values are empty for some features, Arcade expressions were applied to define rendering properties for the application type attribute, reference file number attribute, hearing date attribute, and description attribute.

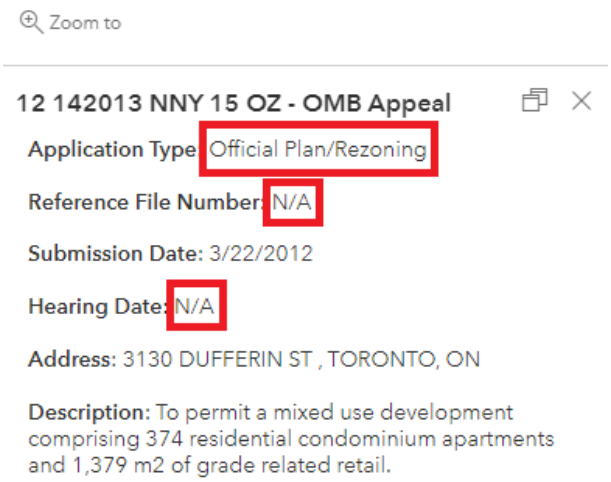

Figure 15 Arcade expressions define rendering properties of attributes (in red borders)

2.2.5 Search Bar

The search bar allows the users to search by address by default. To enhance its functionality, the search bar was configured through Settings on the item page, allowing the users to locate individual development applications by entering application number.

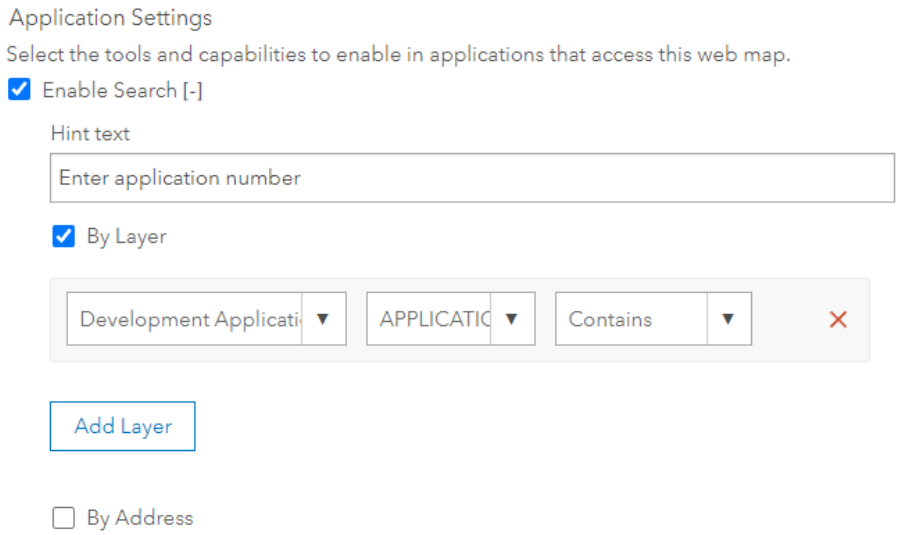

Figure 16 Application setting to enable feature search

2.2.6 Publish Web Maps through ArcGIS API for Python

To allow the client to replicate or enhance the web maps in the future, a script for creating web maps was created. Here is an overview of the workflow for publishing web maps using ArcGIS API for Python:

Figure 17 Workflow of publishing web maps through ArcGIS API for Python

2.2.6.1 Publish Web Maps

To publish the web maps, empty web map objects were created for each web map using the WebMap class. The relevant layers were then added to the web map objects using the add_layer method:

```
add_layer(layer, options)
```

The hosted feature layers for the development applications and the regional municipal boundary were added to the main map and the heat map, while the tile layer for the historical aerial imagery was added directly from the City of Toronto ArcGIS Server to the historical aerial imagery map. The web maps were then published using the save method:

```
save(item_properties, thumbnail, metadata, owner, folder)
```

2.2.6.2 Configure Web Maps

The JSONs of the web maps are required in order to update the map symbology and configure the web maps. Since the web maps were already created earlier using ArcGIS Map Viewer and the script was for replication purposes only, the JSONs were extracted directly from the created web maps through https://www.arcgis.com/sharing/content/items/<itemID>/data (where itemID is the item ID of the web map) (EarlMedina1, 2019). The JSON was then added to the script and was converted to a dictionary using the loads method of the json module, allowing its content to be manipulated. The following variables were changed:

- operationalLayers: id (unique ID of the layer)
- operationalLayers: url (REST endpoint of hosted feature layer) – *except historical aerial imagery map*
- operationalLayers: itemID (item ID of hosted feature layer) – *except historical aerial imagery map*
- applicationProperties: viewing: search: layers: id (same as operationalLayers: id) – *main map only*

Once the JSON dictionary was converted back to a string using the dumps method of the json module, the web maps were configured using the update method, where the dictionary of the item properties was provided (JSON string was the value of the dictionary key “text”).

2.2.6.3 Share Items

Same as sharing the hosted feature layers, the share_items method was used to share the web maps with everyone.

2.3 ArcGIS Dashboards

ArcGIS Dashboards enables users to convey information by presenting location-based analytics using intuitive and interactive data visualizations on a single screen, helping them to make informed decision (Esri, 2022). Here is a list of elements that were used in the dashboard:

Figure 18 Elements of the dashboard

2.3.1 User Interface Design

The user interface design plays an important role in the effectiveness of the dashboard. The dark theme was used to minimize digital eye strain of the users. The sidebar with the selectors and the tabs were utilized to help displaying content in a limited space. Responsive design was also integrated to the dashboard by using CSS font size property "vw". As the default dashboard is not fully responsive and it only looks good when it is displayed in the original screen resolution (Full HD 1920 x 1080), the font size was configured to help displaying content in various screen resolutions.

2.3.2 Header

A header is a reserved area along the top of the dashboard to give the dashboard a unique identity, apply corporate branding standards, and provide links to additional content (Esri, 2022). It was customized to provide basic information about the dashboard to the users. The logo of the City of Toronto was added, and the header colour was set to the theme colour of the City of Toronto (dark blue), helping the users to identify the owner of the dashboard. The links to the relevant websites were also added to the menu of the header, allowing the users to explore other relevant information quickly.

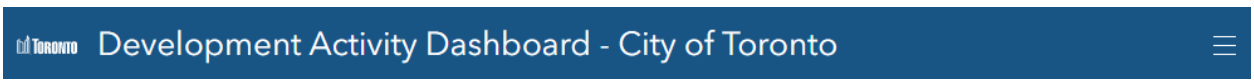

Figure 19 Header of the dashboard

2.3.3 Map

The main map and the heat map were added to the dashboard. The map extents were configured so that the map extents of both web maps were synchronized. The map actions were also configured so that other dashboard elements such as indicators can be updated based on the current map extent, allowing the users to explore information interactively.

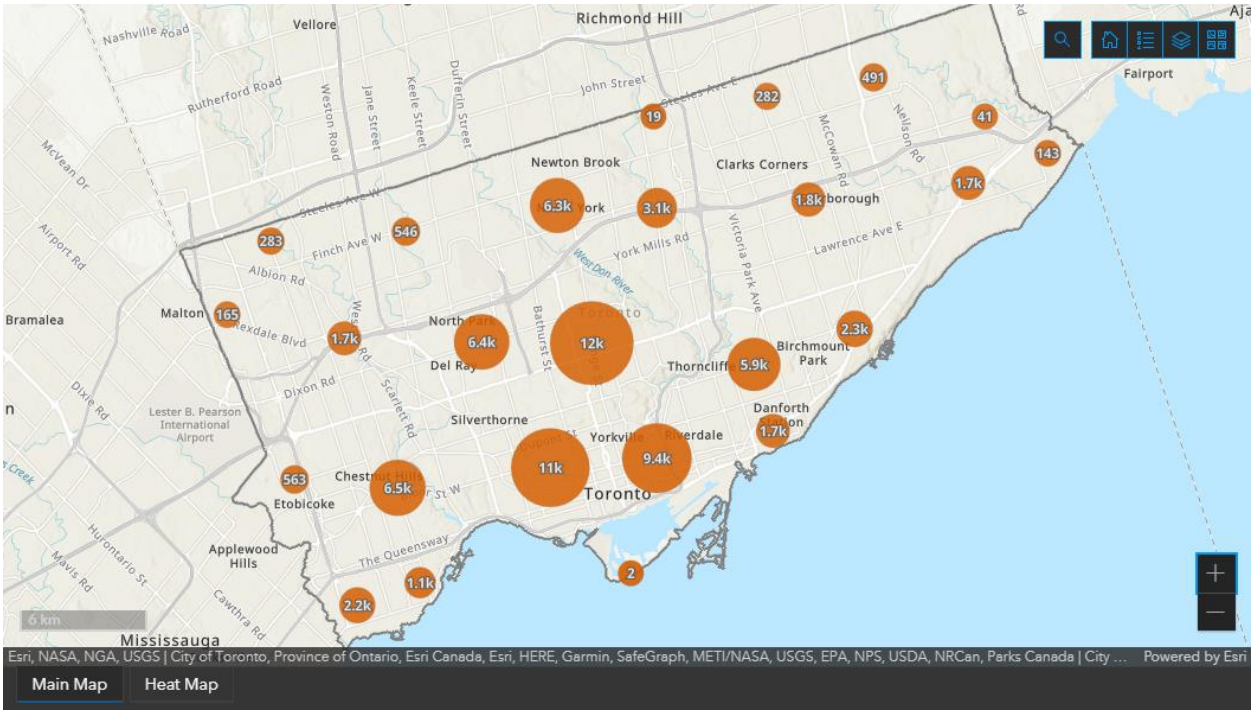

Figure 20 Map section of the dashboard

2.3.4 Indicator

An indicator is a card that can be added to the dashboard to show the numeric attributes of individual features or display a summary statistic (Esri, 2022). Multiple indicators were added to display the total number of development applications since 2008, the total number of development applications, and the number of development applications of different types. The colours of the application type indicators were also set to match the colour of the application types on the main map, helping the users to identify the application types quickly.

Figure 21 Minor variance indicator in the dashboard

2.3.5 Selectors

A selector supports a selection change event that can trigger actions (Esri, 2022). Logos were applied to help the users to identity the selectors easily. There were two types of selectors added to the sidebar of the dashboard: date selectors and category selectors.

2.3.5.1 Date Selector

Date selector allows the users to select a time range using a date picker. The application date selector and the hearing date selector were added to allow the users to filter data based on the submission date and the hearing date, within a specific time range.

2.3.5.1 Category Selector

Category selector allows the users to choose categories derived from the features. The application type selector, the application status selector, and the postal code selector were added to the dashboard. The application type selector allows the users to select the development applications of a specific application type, where the application status selector allows the users to select the development applications of a specific application status. The postal code selector allows the user to display the development applications within a certain area based on the first three digits of the postal code.

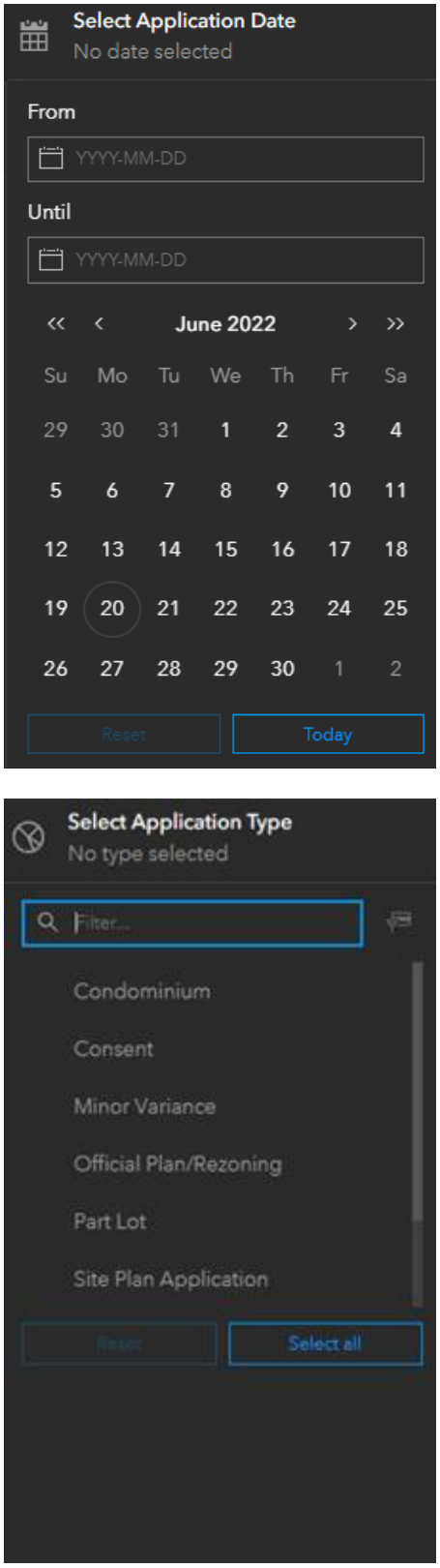

Figure 22 Application date selector (top) and application type selector (bottom)

2.3.6 List

A list is used to show features or rows from a layer (Esri, 2022). The recent applications list and the active application list were added to display the information of the development applications. As the application type values in the dataset were abbreviations only, an Arcade expression was applied to define rendering properties for the application type attribute. The lists were also configured to trigger actions. For example, selecting the individual features on the list will trigger the web map to pan, zoom or display pop-ups.

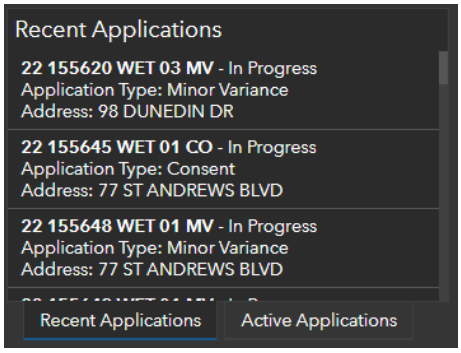

Figure 23 Lists in the dashboard

2.3.7 Serial Chart

A serial chart visualized series of data points along a horizontal axis and a vertical axis (Esri, 2022). A column chart was added to display the development activity trend by month, with hover texts displaying the number of development applications in a specific month. A scroll bar was also added to the chart to control the time range displayed.

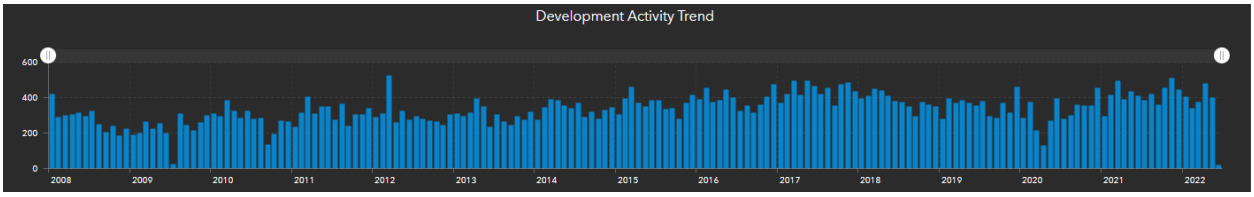

Figure 24 Serial chart in the dashboard

2.3.8 Rich Text

Rich text elements were used to provide additional information about the dashboard. An introduction section was added to provide background information about the dataset. A glossary of terms section was added to provide definitions of different application types. A disclaimer section was added to prevent civil liability arising for particular acts or omissions.

2.4 ArcGIS Experience Builder

ArcGIS Experience Builder is an ArcGIS Online application to transform the data into web applications easily. An interactive web application was created using ArcGIS Experience Builder for this project. It is made of a set of widgets to give information about the distribution of the development activities within the city. It is highly responsive that the content can be displayed on regular desktop, laptop, and mobile devices. The web application is AODA compliant, where the used colour palette allows both people with colour deficiencies and people without colour deficiencies to use the web application without difficulties. The web application includes a navigation bar, the web maps, the application list, a map legend, and a feedback button linked to the survey. The following widgets were added to the web application:

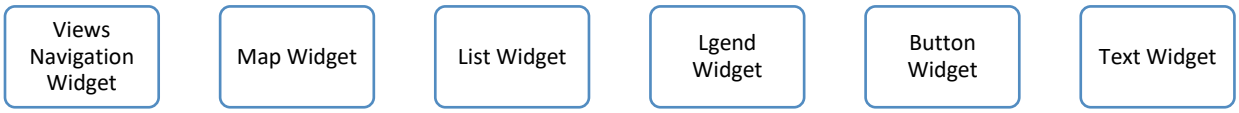

Figure 25 Overview of the widgets used in the web application

2.4.1 Header

Similar to the dashboard, the header was customized to provide basic information about the web application to the users. The logo of the City of Toronto was added to the header which redirects the users to the main page.

2.4.2 Views Navigation Widget

The views navigation widget was added to the header, allowing the users to switch between the main map and the heat map easily. The underline indicates the current section.

Figure 26 Header with views navigation widget in the web application

2.4.3 Map Widget

The map widget displays 2D and 3D geographic information (Esri, 2022). The main map and the heat map were added to the web application to display the development activities in Toronto.

2.4.4 List Widget

The list widget displays data records in a custom list view (Esri, 2022). A list of applications was added to the web application, where the application type, the application ID, and the address of the individual development applications were displayed. A search box was added to allow the users to search development applications by application type or application ID. A sort tool was added to allow the users to sort the list by application type or application ID. An application type filter was added to the list, allowing the users to select the development applications based on the application type.

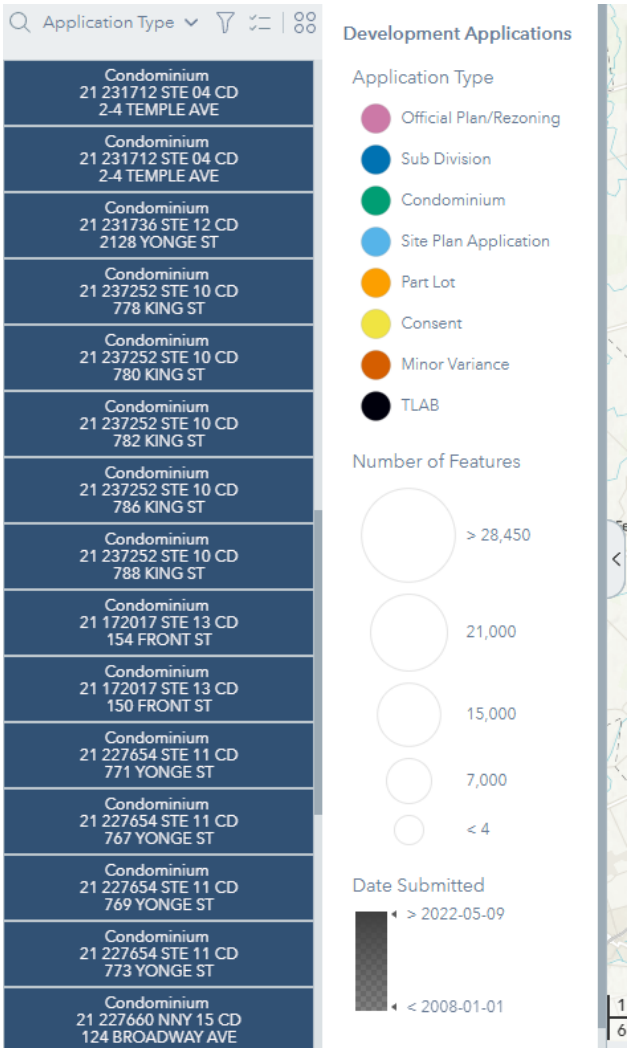

Figure 27 Application list (left) and retractable legend (right) in the web application

2.4.5 Legend Widget

The legend widget displays labels and symbols for layers in a map (Esri, 2022). Retractable legend widgets were added for both web maps, allowing the users to understand the map symbology easily.

2.4.6 Button Widget

The button widget provides links to open pages, windows, and section views in your app, scroll to a particular block or to the top of the page, or go to a web address (Esri, 2022). In the web application, a feedback button was added to the bottom of the application list, linking to a feedback survey directly. The feedback survey allows the user to provide feedbacks about the web application.

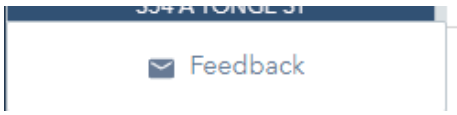

Figure 28 Feedback button in the web application

2.4.7 Text Widget

A text widget was added to the bottom of the web maps for the disclaimer section, explaining the ownership and the limitations of the web applications.

2.5 ArcGIS StoryMaps

ArcGIS StoryMaps is a story authoring web-based application to share maps in the context of narrative text and other multimedia content (Esri, 2022). This application was used to create an educational article about the development activities in Toronto. Here is an overview of the main sections of the story map:

- Introduction
- Evolution of the City (Toronto's development history)
- Development Applications (overview of development application types)
- Current Development Activities (web solutions to display current development activities)
- Conclusion

The content of the story map was authored based on the official website of the City of Toronto, with the approval from the client. A copyright statement was added to the footer to indicate the copyright of the content.

2.5.1 User Interface Design

Tidal theme is one of the default themes that is available on ArcGIS StoryMaps. It was applied to the story map as its dark blue theme matches the theme colour of the City of Toronto, allowing the users to identify the owner of the story map. Navigation was enabled to allow the users to jump to a specific section of the story map.

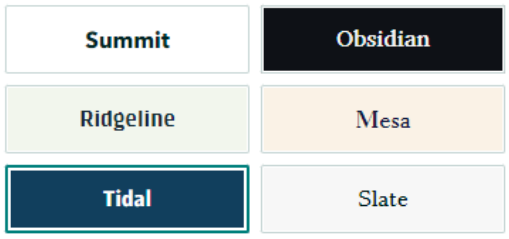

Figure 29 Themes that are available on ArcGIS StoryMaps

2.5.2 Swipe

A swipe block is an interactive experience that allows readers to compare two maps or images (Esri, 2022). To allow the users to compare the differences between Toronto in 1939 and Toronto nowadays, the historical aerial imagery map and the imagery hybrid map that was created by Esri Canada were added to the swipe block.

Figure 30 Swipe block in the story map

2.5.3 Sidecar

Sidecar blocks are a combination of media and story narrative that fill the display, creating an immersive experience in stories (Esri, 2022). This functionality was utilized to display the web solutions for this project, including the web maps, the dashboard, and the web application, allowing the users to explore the development activities in Toronto nowadays.

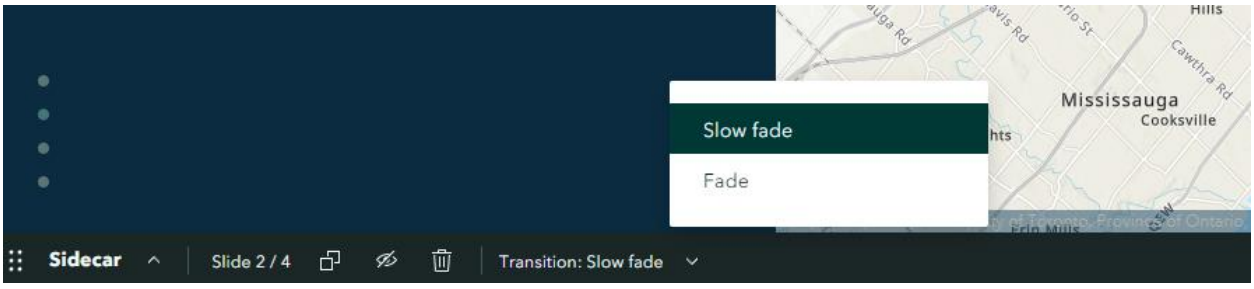

Figure 31 Transition was set to slow fade for smoother transition

2.6 Maintenance

The maintenance process was implemented using ArcGIS API for Python. The maintenance script was hosted and run through ArcGIS Notebooks, where the Standard license was sufficient for the automation process. The maintenance process involves installing Manage Notebook Code Dependencies (MNCD), running OverwriteFS tool, sanitizing problematic data, adding geometries to existing features, and setting up scheduled tasks. Here is an overview of the workflow of the maintenance process:

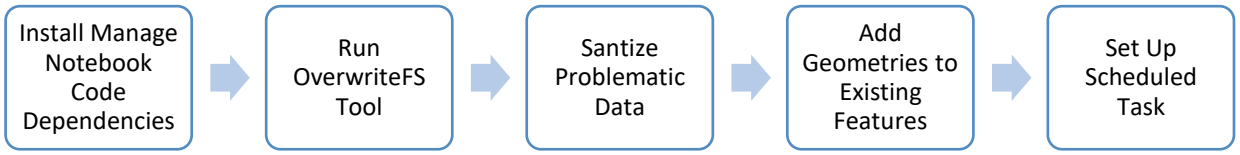

Figure 32 Workflow of maintenance process

Note: Since the June 2022 update after project completion, ArcGIS Online can no longer support publishing hosted feature layers with unsupported coordinates formats from the source file directly (e.g. development applications dataset). The workaround is to update the hosted table layer and transfer features to the hosted feature layer using Spatially Enabled DataFrame. The following workflow is outdated but the scripts has been modified to reflect the updates.

2.6.1 Manage Notebook Code Dependencies (MNCD)

To update the hosted feature services using the OverwriteFS tool, the Manage Notebook Code Dependencies (MNCD) must be first installed and applied to the Notebook environment. MNCD tool allows the users to manage a cache of Python "Code Sample" and "Notebook" item content in the user's home folder. Once cached, the content from these items can be imported into the script. This enables the users to build or leverage existing libraries of reusable code and greatly simplifies the script (Dodd, 2021). The MNCD installer can be downloaded from ArcGIS Online (https://www.arcgis.com/home/item.html?id=46c7512604654601ab4338f9299c5414).

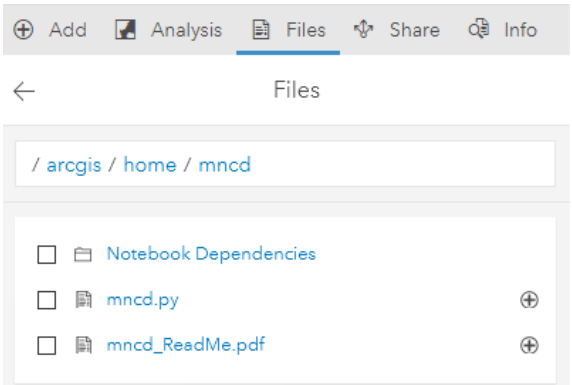

Figure 33 The contents of MNCD Code Sample was download to the user's home folder

To apply MNCD to the Notebook environment, the MNCD installer was added to the Portal and run through ArcGIS Notebooks. During the installation, the contents of the MNCD Code Sample item was downloaded and extracted to the user's home folder "/arcgis/home" under a newly created folder called "mncd".

2.6.2 OverwriteFS

After applying MNCD to the Notebook environment, the OverwriteFS tool was imported to update the existing hosted feature layers for the development applications dataset and the regional municipal boundary shapefile. OverwriteFS tool allows the users to automate and standardize the data updates using a Python script (Esri, 2022). The sample script created by Esri was further enhanced in this project. A variable called "dev_update" was created and initialized to False. If the hosted feature layers were updated successfully, the "dev_update" variable will be updated to True, allowing further processing of the hosted feature layers. If the hosted feature layers were not updated, the rest of the script will not be run since no updates were required.

2.6.3 Sanitize Problematic Data and Add Geometries to Existing Features

If the hosted feature layers were updated successfully, the problematic data such as invalid X and Y coordinates and invalid dates will be sanitized. The geometries will be also added to the existing features using Spatially Enabled DataFrame. The details of the process were discussed in the 2.1.5 Sanitize Problematic Data and 2.1.6. Add Geometries to Existing Features.

2.6.4 Scheduled Task

The maintenance script will not be run automatically without the setup of a scheduled task. A scheduled task "Daily Maintenance" was created on ArcGIS Notebooks to run the script automatically at midnight every day. The hosted feature layers will be updated

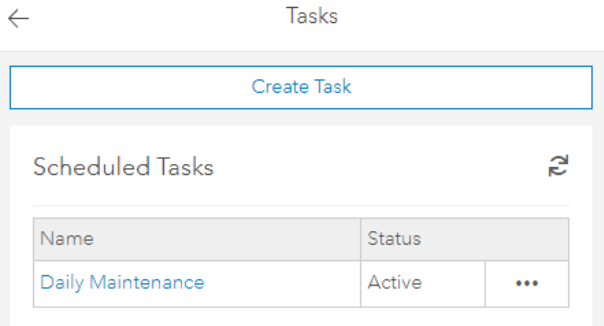

Figure 34 Scheduled task for daily maintenance in ArcGIS Notebooks

automatically and the effort of maintenance by the client can be minimized. The web maps, the dashboard, the web application, and the story map will be updated automatically once the hosted feature layers are updated.

3. Results

To alleviate the difficulties when researching development activities in Toronto, a series of web solutions were created to meet the project objectives. Here is an overview of the main deliverables of this project:

Figure 35 List of main deliverables

The following supplemental deliverables are not listed as the main deliverables of the project but were created to support the main deliverables.

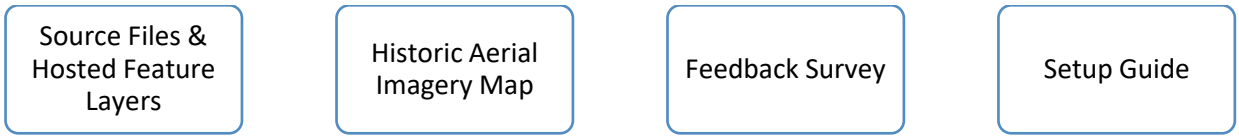

Figure 36 List of supplemental deliverables

3.1 Web Maps

3.1.1 Main Map

The main map lists all currently active and inactive Community Planning applications, Committee of Adjustment applications and Toronto Local Appeal Body appeals received by the City between January 1st 2008 till present. It can be viewed through ArcGIS Map Viewer. https://fleming.maps.arcgis.com/apps/mapviewer/index.html?webmap=d7455aa93d2440508a07e872415059eb

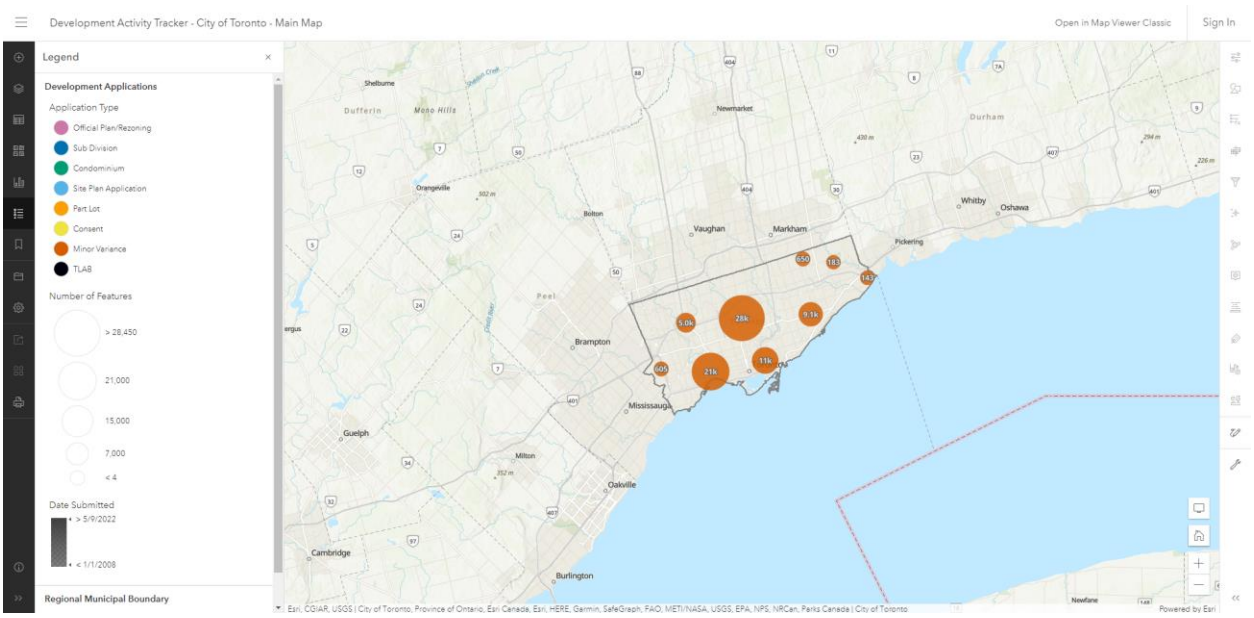

Figure 37 Overview of the main map

The clusters are represented by the colour of the most common application type in the area.

The sizes of the clusters are based on the number of development applications in the area.

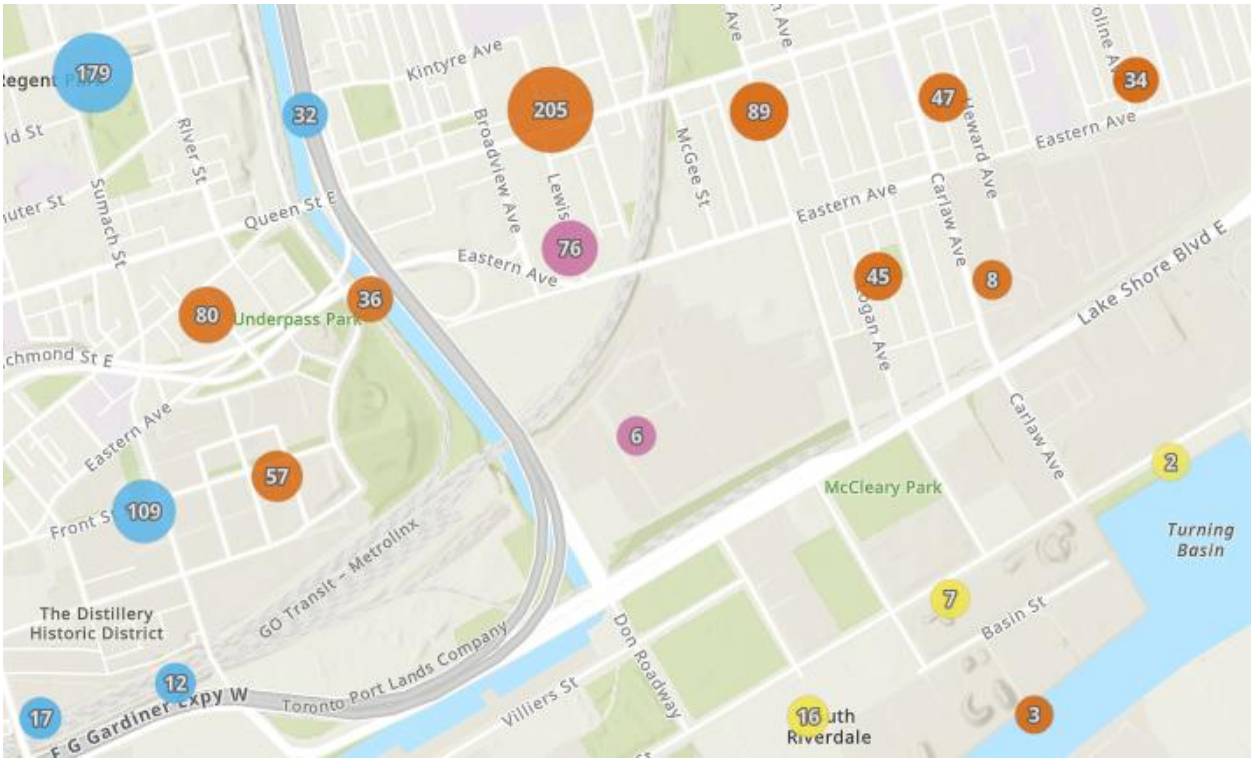

Figure 38 Clusters in the main map

The cluster pop-ups show the number of development activities and the most common application type in the area. The pop-ups also allow the users to browse features within the cluster.

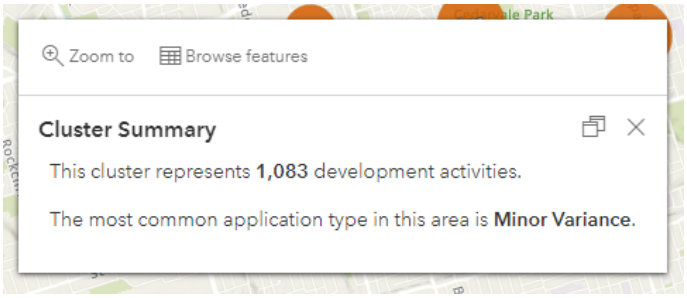

Figure 39 Cluster pop-up in the main map

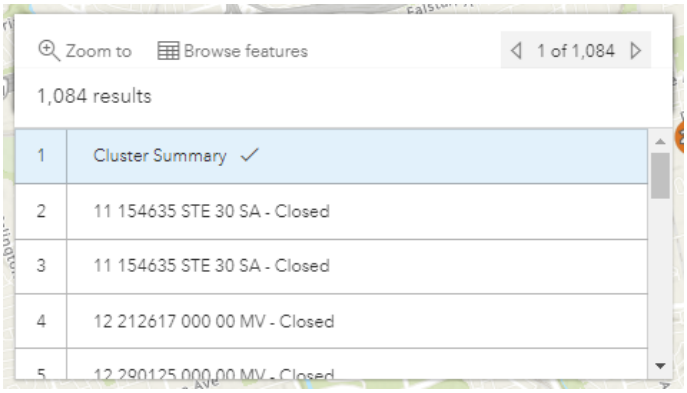

Figure 40 Browse features of the cluster pop-up in the main map

The individual feature pop-ups display all the relevant information, including the application number, the application status, the application type, the reference file number, the submission date, the hearing date, the address, and the description. The null values are shown as “N/A” in the pop-ups to avoid confusions.

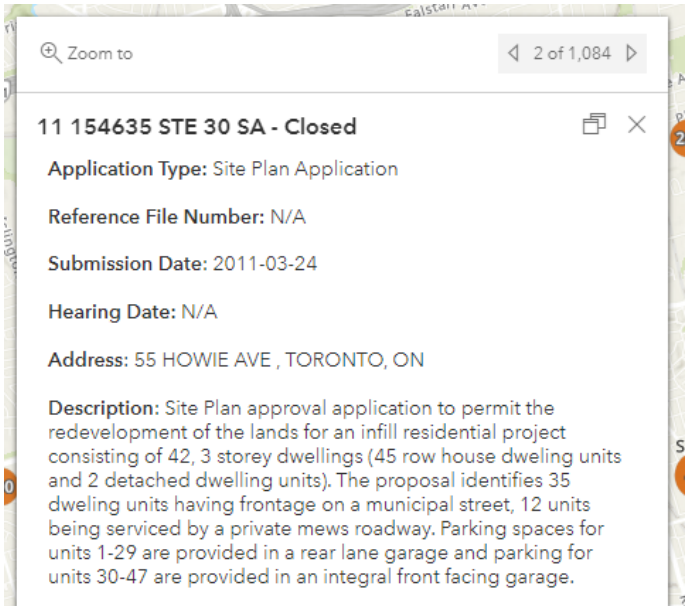

Figure 41 Individual feature's pop-up in the main map

The search bar allows the users to search by application number or address.

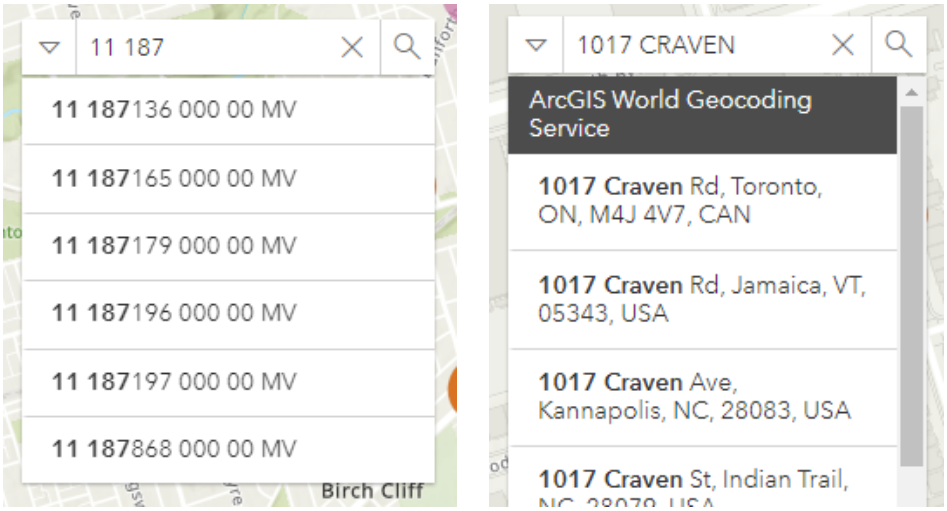

Figure 42 Search by application number (left) and address (right) in the main map

3.1.2 Heat Map

The heat map visualizes the magnitude of development activities in different locations, helping the users to understand the patterns and identify the hot spots of the development activities in Toronto. Similar to the main map, the heat map can be viewed through ArcGIS Map Viewer. https://fleming.maps.arcgis.com/apps/mapviewer/index.html?webmap=1ff4b037ec8d420b928d853b1de122f5

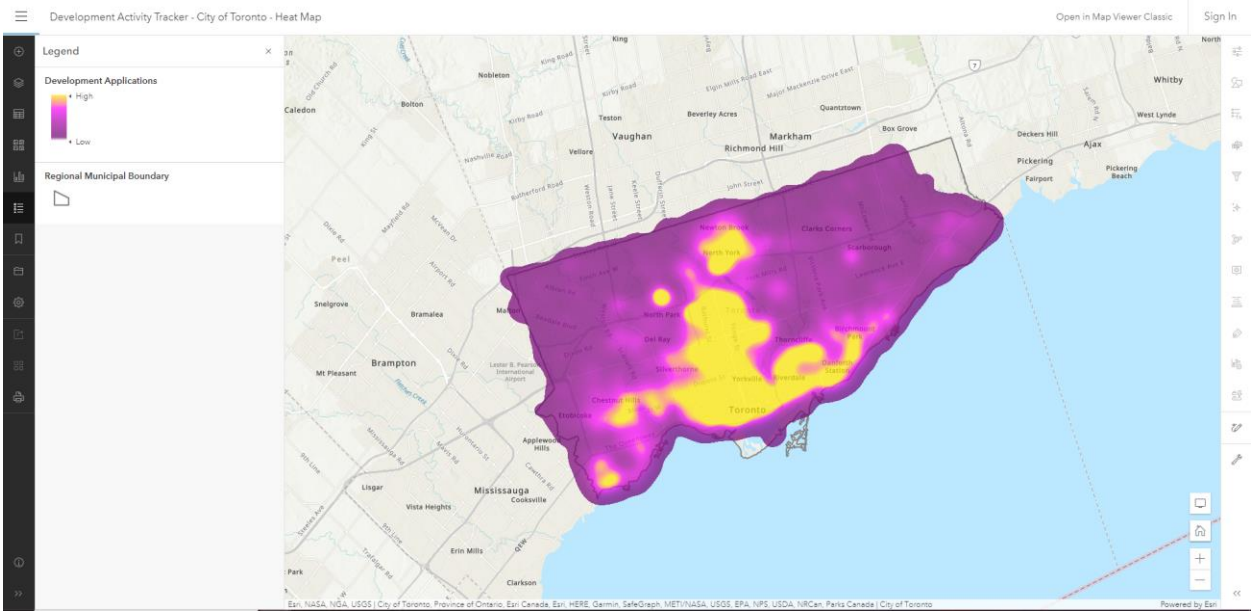

Figure 43 Overview of the heat map

3.2 Dashboard

An operational dashboard was created using ArcGIS Dashboards. It integrates the web maps and all relevant information on a single screen, presenting dynamic results in an easy-to-read format and allowing the users to explore the data interactively. Here is an overview of the dashboard with default screen resolution of 1920 x 1080 (1080p).

https://fleming.maps.arcgis.com/apps/dashboards/5b5195282e4f4d959022efecc33c232b

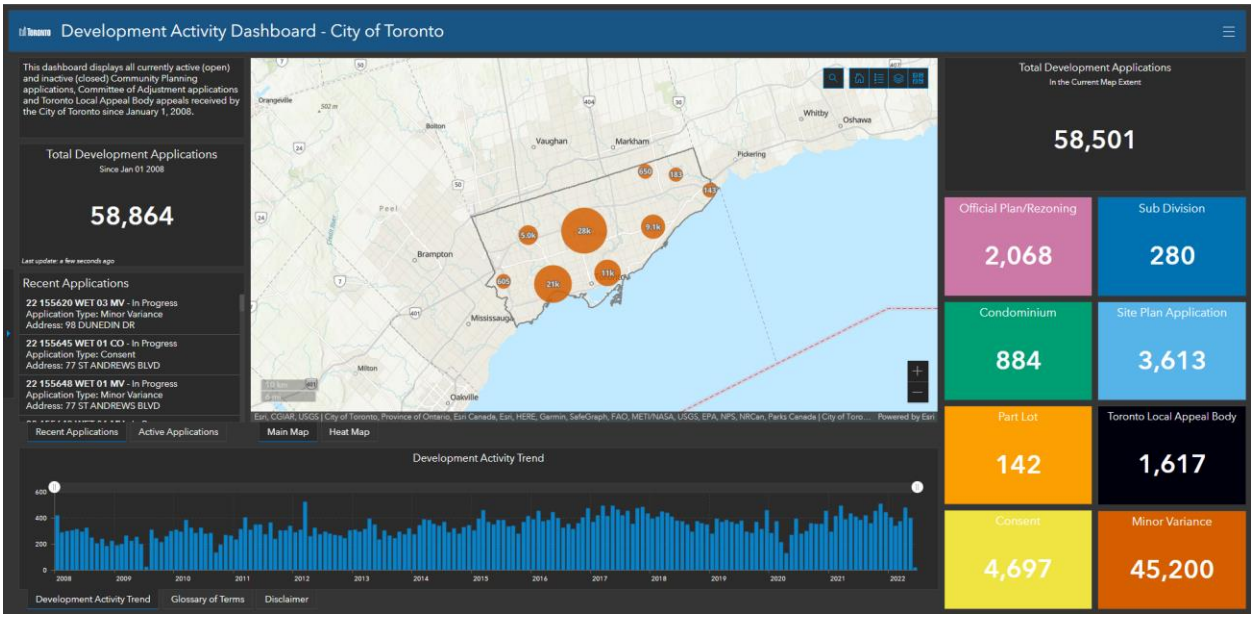

Figure 44 Overview of the dashboard

The dashboard is responsive and works with different screen resolutions.

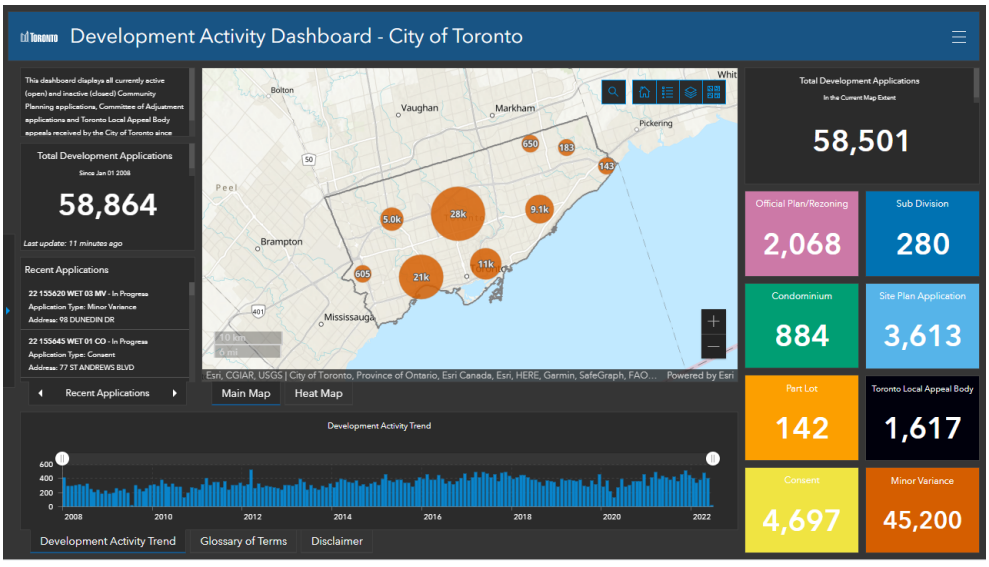

Figure 45 Dashboard with a screen resolution of 1280 x 720 (720p)

The lists on the left, the indicators on the right, and the chart at the bottom will be updated based on the current map extent.

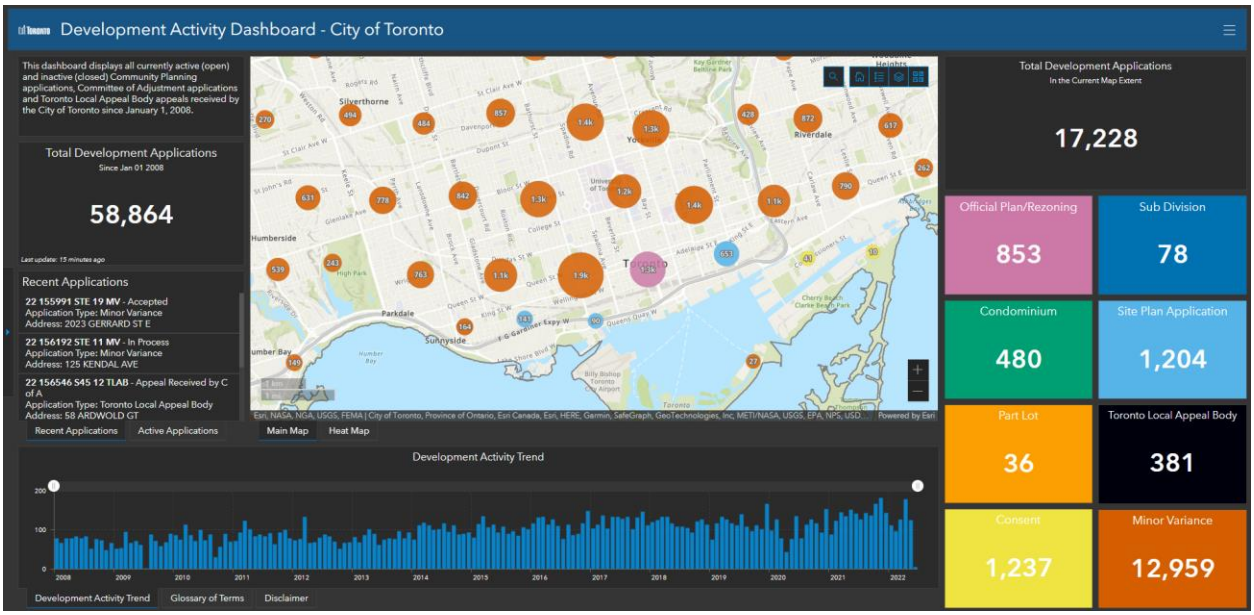

Figure 46 Lists, indicators, and charts updated based on the current map extent in the dashboard

The filters are located in the retractable side panel. The lists on the left, the indicators on the right, and the chart at the bottom will be updated based on the selected filters as well.

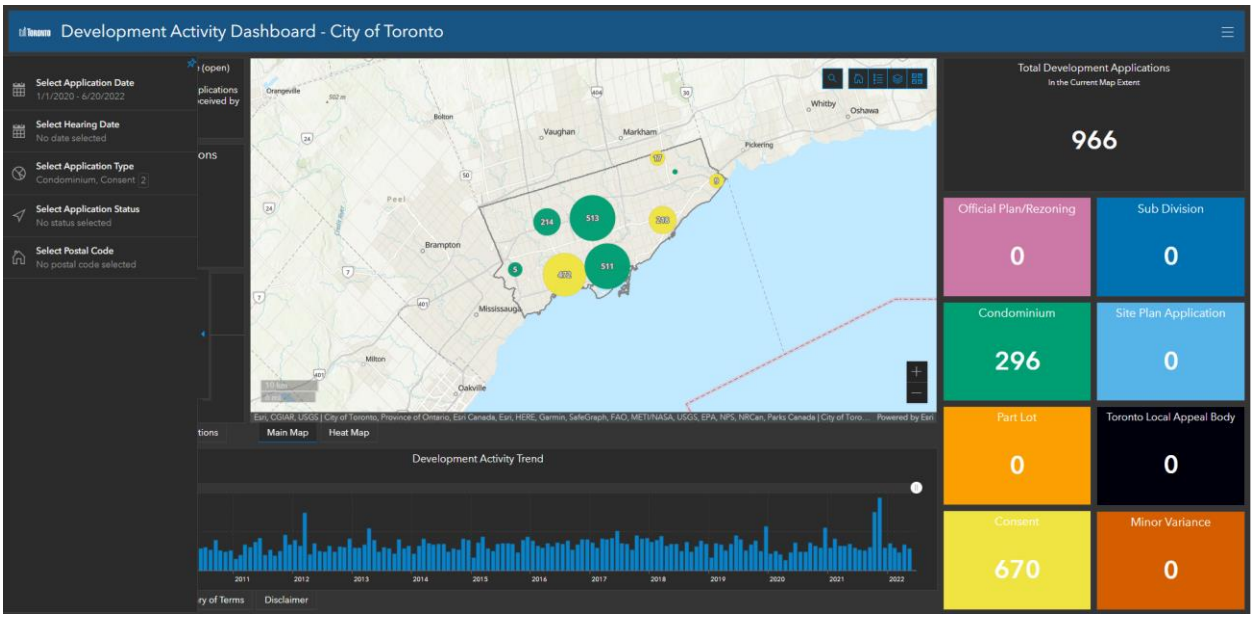

Figure 47 Lists, indicators, and charts updated based on the filters in the dashboard

The scroll bar of the chart controls the time range of the data displayed.

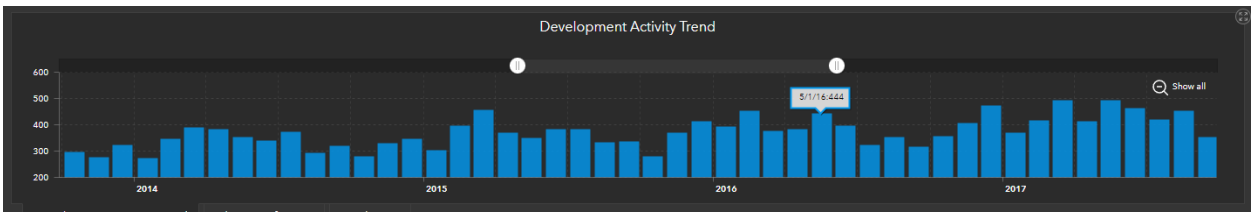

Figure 48 Scroll bar of the chart in the dashboard

The menu links in the header allow the users to access other related websites or web applications quickly.

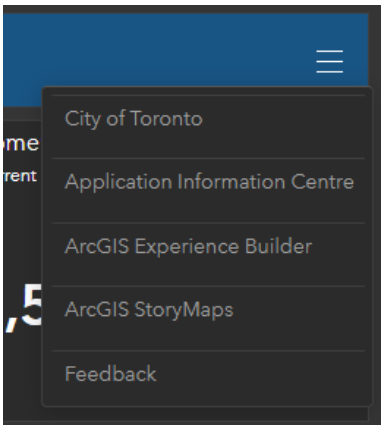

Figure 49 Menu links in the dashboard

3.3 Web Application

A web application was created using ArcGIS Experience Builder in this project. The web application displays the web maps and information using customized widgets and templates, where the responsive design allows the web application to run seamlessly on desktop and mobile devices.

https://experience.arcgis.com/experience/499362549fde45edb0f3b958060ac1c6

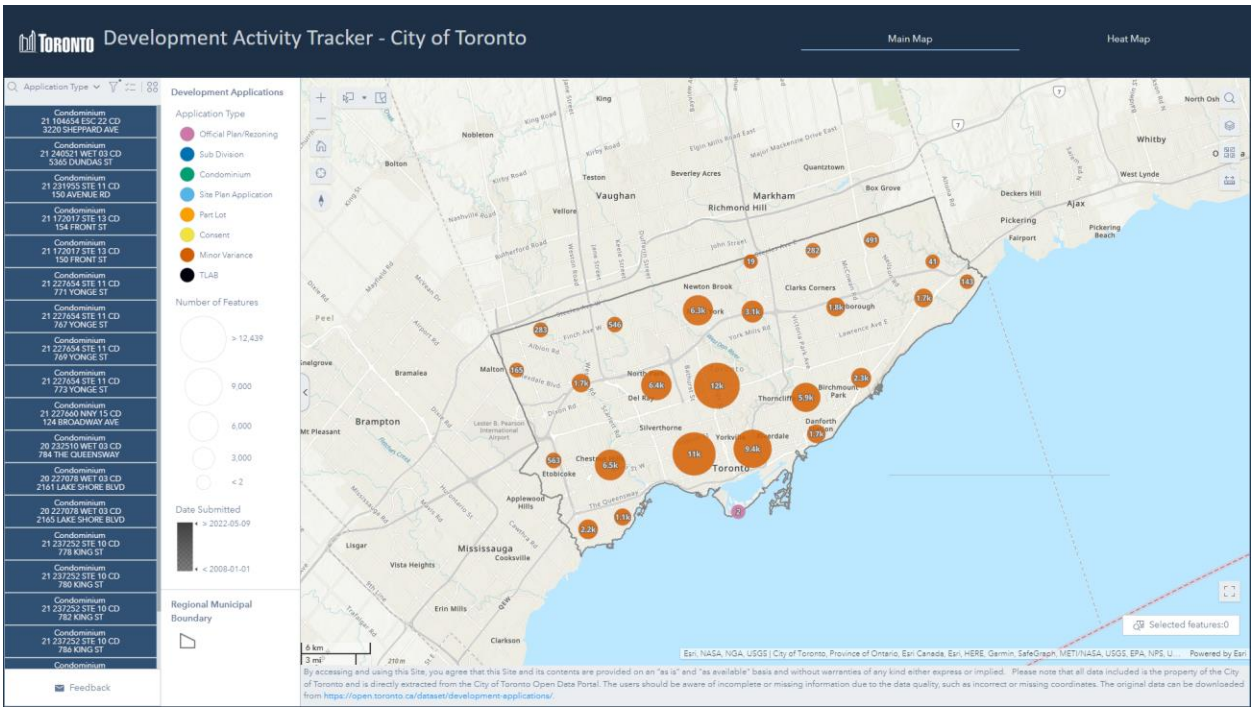

Figure 50 Overview of the web application (main map)

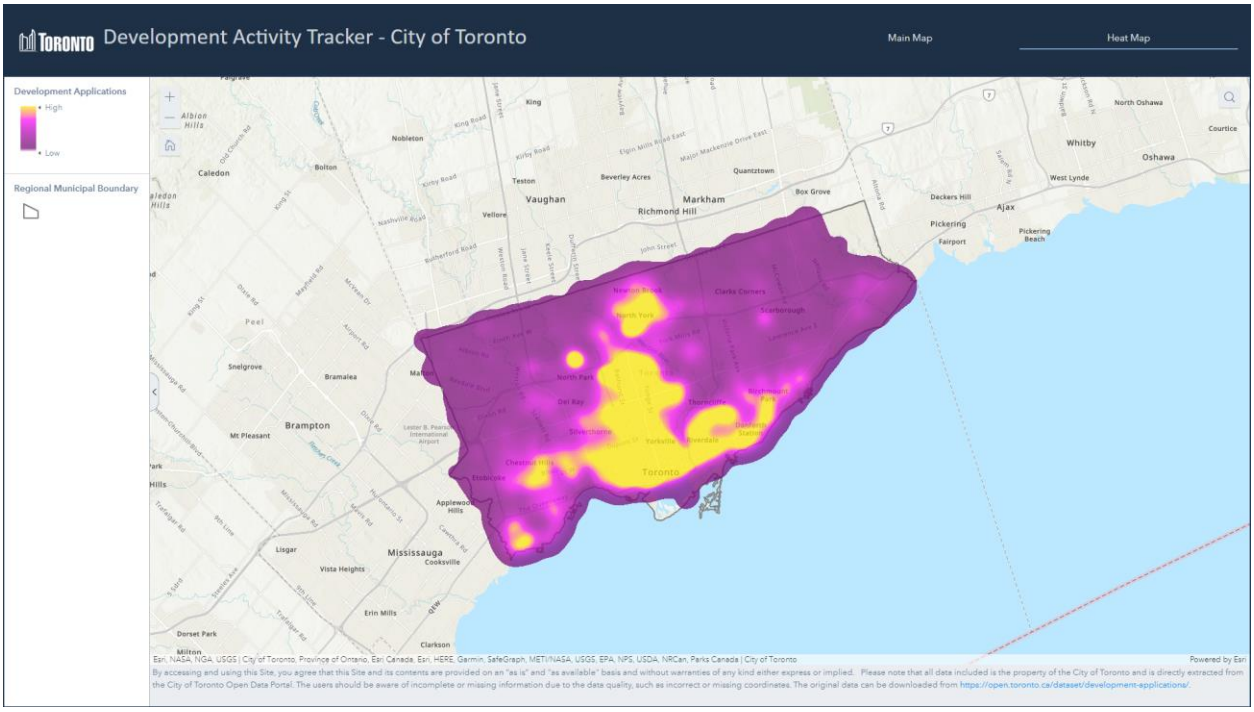

Figure 51 Overview of the web application (heat map)

The responsive design of the web application works with different screen resolutions.

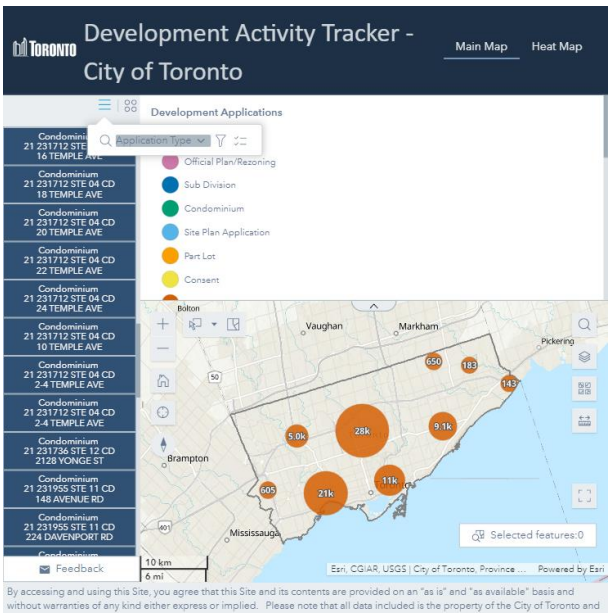

Figure 52 Web application with a screen resolution of 800 x 800

The users can search the development applications by entering the application ID in the search box of the list to filter the results in the main map.

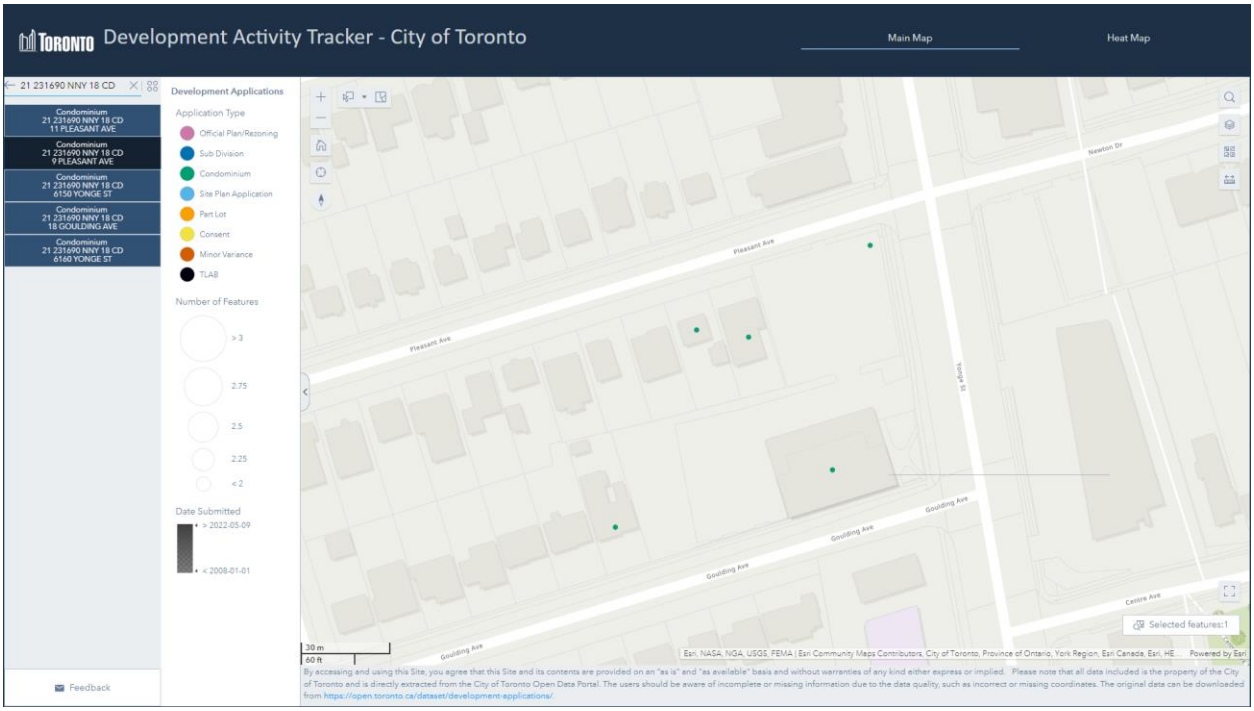

Figure 53 Search by application ID in the web application

The users can either search the development applications by entering the application type in the search box or use the application type filter of the list to filter the results in the main map.

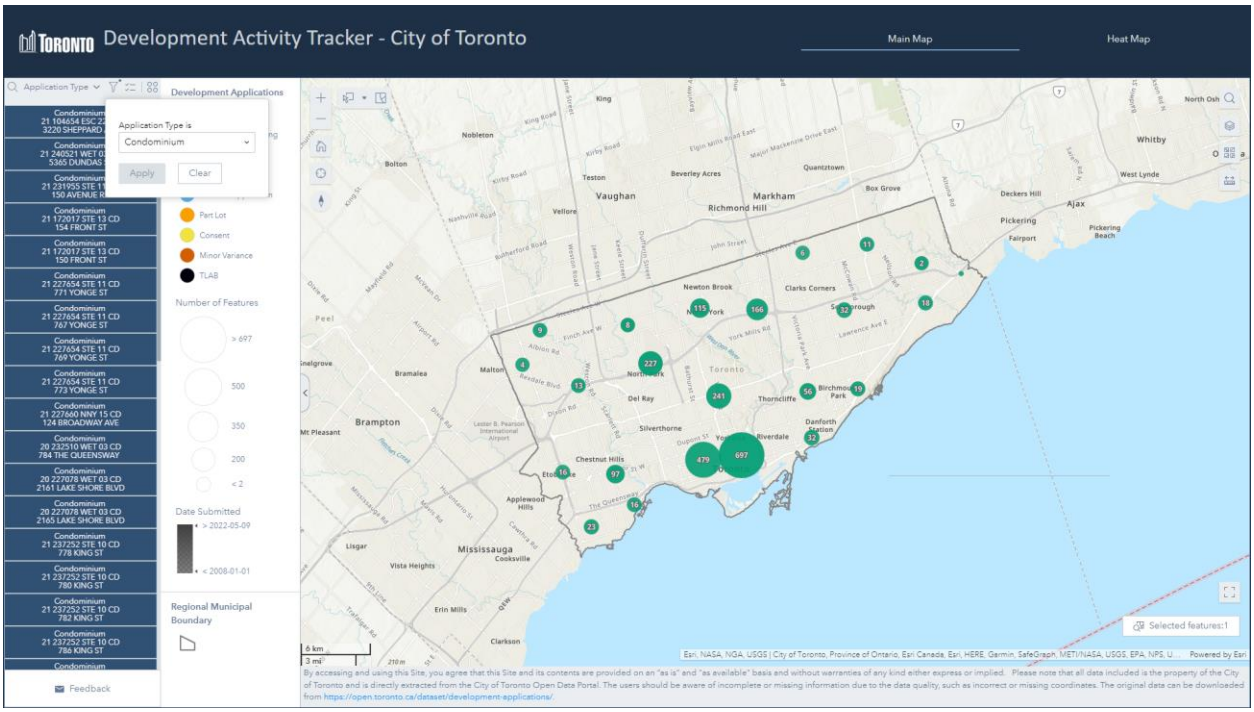

Figure 54 Filter application type in the web application

3.4 Story Map

The story map was created using ArcGIS StoryMaps in this project. It provides information about the development activities in Toronto, in addition to the web maps. Through the story map, the users can learn more about the development history of Toronto and the current development application types that are available. They can also review the web maps and the web application within the story map, with direct links to allow quick access.

https://storymaps.arcgis.com/stories/110e1015cfc8455590998a8050d1998b

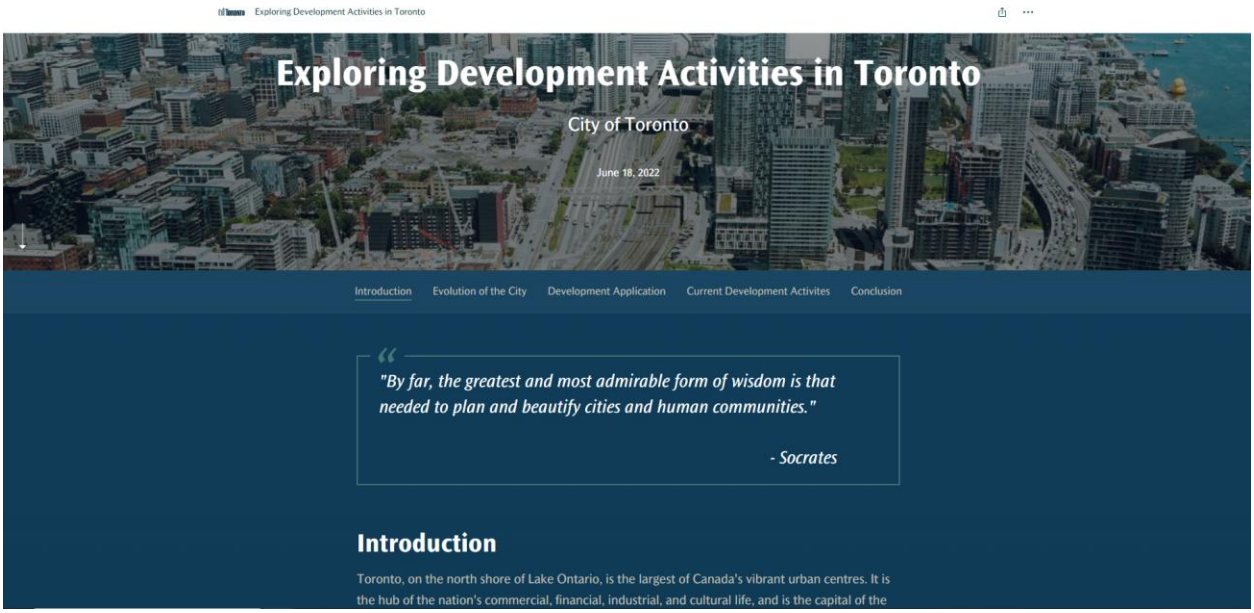

Figure 55 Overview of the story map

The users can jump to a specific section of the story map by clicking links on the navigation bar.

Figure 56 Navigation bar and the selected section in the story map

The users can also compare the differences between the historical aerial imagery and the current historical imagery of Toronto interactively using the swipe block.

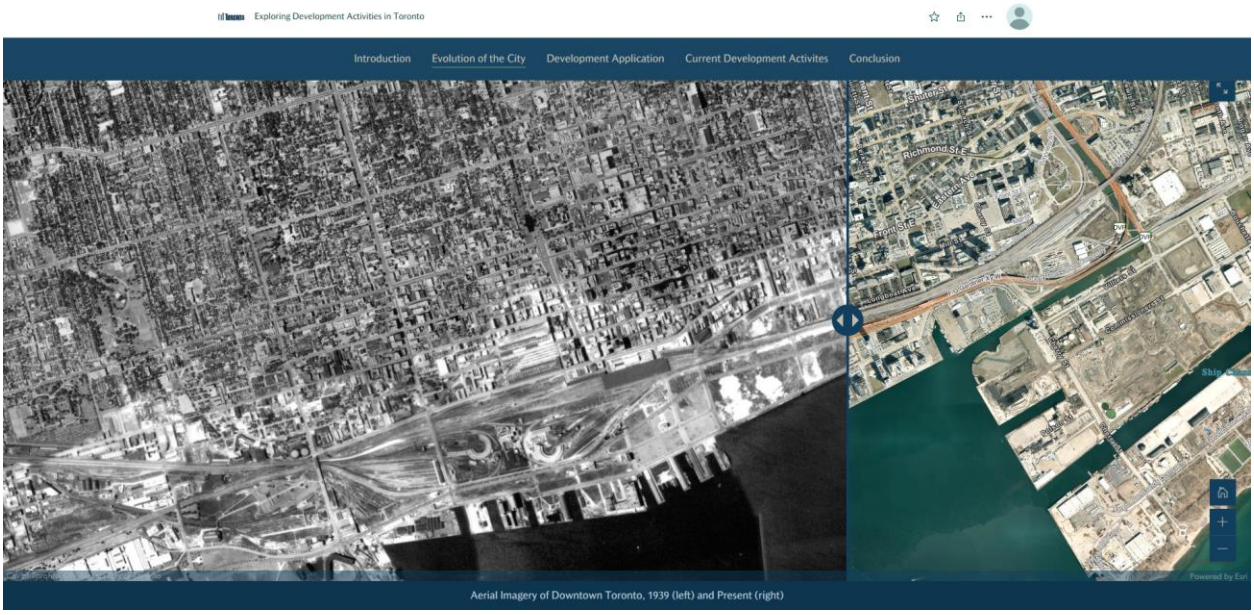

Figure 57 Swipe in the dashboard

The users can interact with the web maps within the sidecar element in the story map.

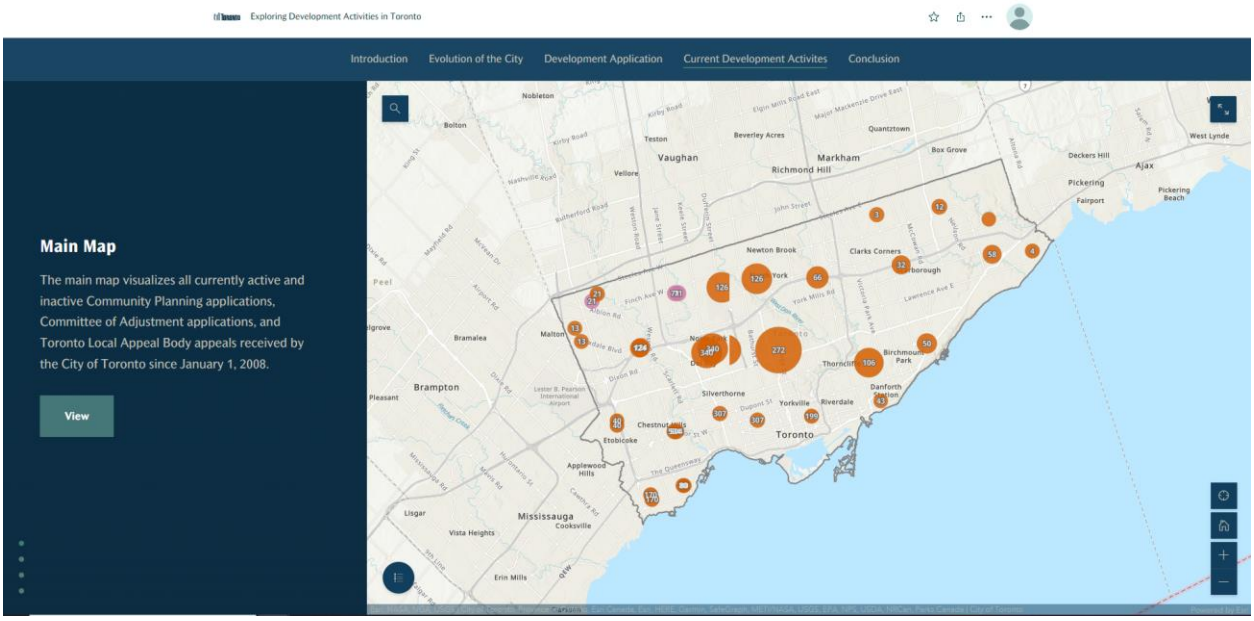

Figure 58 Sidecar in the story map

3.5 Python Scripts

3.5.1 Setup Script

The setup script utilizes ArcGIS API for Python to publish hosted feature layers and set up web maps for this project easily. The client can run the setup script through ArcGIS Notebooks. https://fleming.maps.arcgis.com/home/notebook/notebook.html?id=0d2c3177fe4241cd8b6ee0e036ebc0da

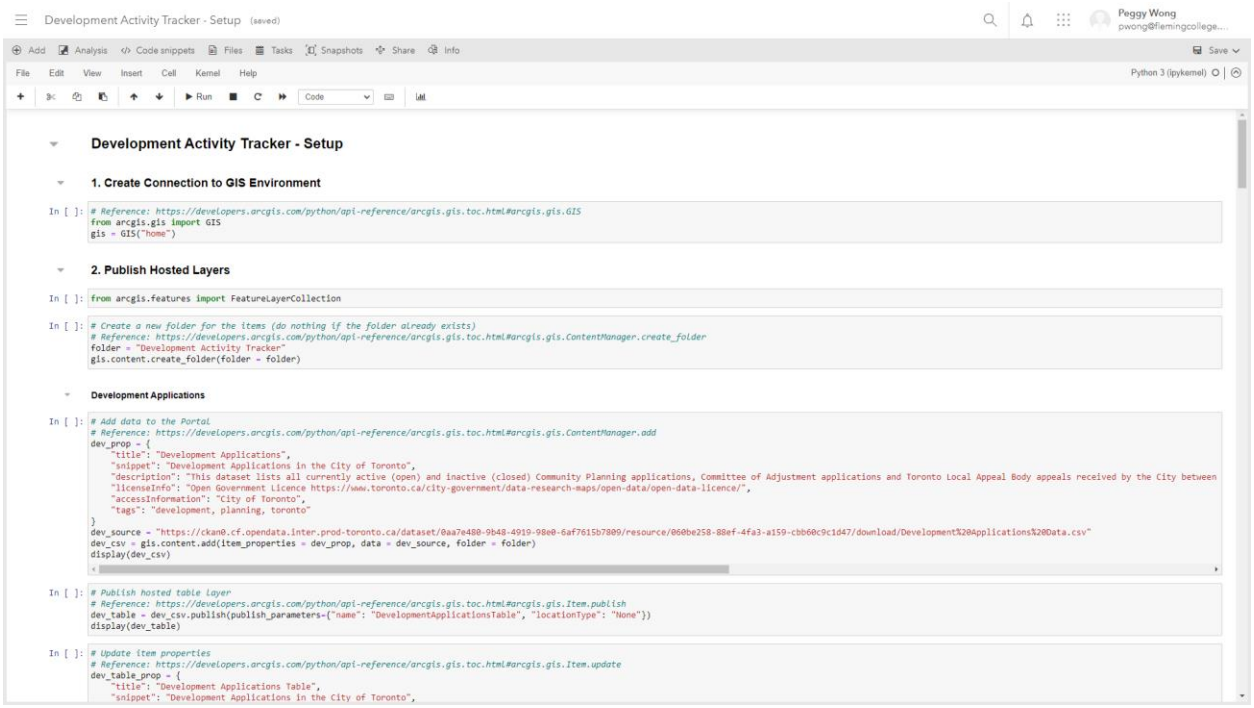

Figure 59 Overview of the setup script

3.5.2 Maintenance Script

The maintenance script utilizes ArcGIS API for Python to update the hosted feature layers of the Development Activity Tracker on a daily basis. The client can host the maintenance script through ArcGIS Notebooks to utilize the scheduled task functionality. https://fleming.maps.arcgis.com/home/notebook/notebook.html?id=4adc613f0696473883a28ac3d5acc360

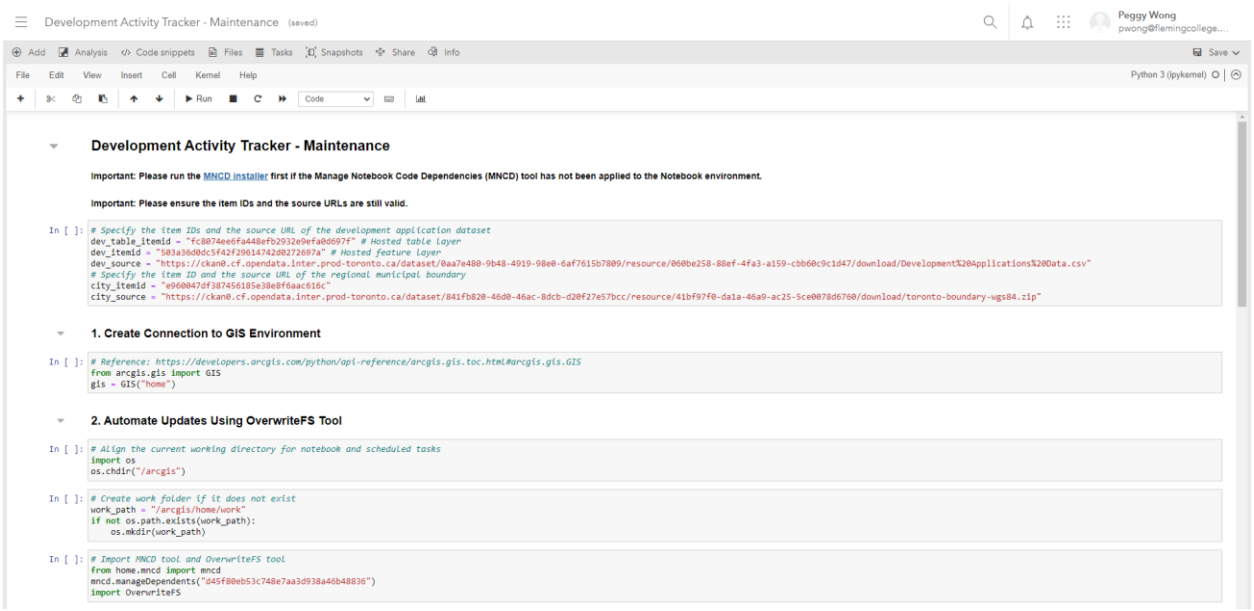

Figure 60 Overview of the maintenance script

The client can also view the scheduled task history in the maintenance script and check the results of the previous runs.

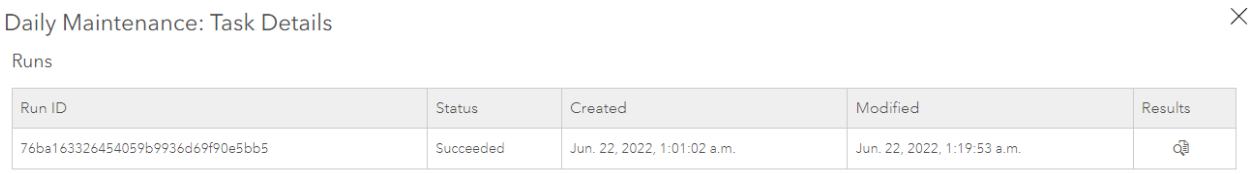

Figure 61 Task details of scheduled task in the maintenance script

3.6 Supplemental Deliverables

3.6.1 Source Files & Hosted Feature Layers

The hosted feature layer of the development applications dataset was published from the source CSV file. The hosted feature layer of the regional municipal boundary was published from the source shapefile. Both hosted feature layers were added to the web maps.

Note: Since the June 2022 update after project completion, ArcGIS Online can no longer support publishing hosted feature layers with unsupported coordinates formats from the source file directly (e.g. development applications dataset). The workaround is to publish a hosted table layer from the source file (as an intermediate layer) and transfer features to an empty hosted feature layer using Spatially Enabled DataFrame.

https://fleming.maps.arcgis.com/home/item.html?id=503a36d0dc5f42f29614742d0272697a

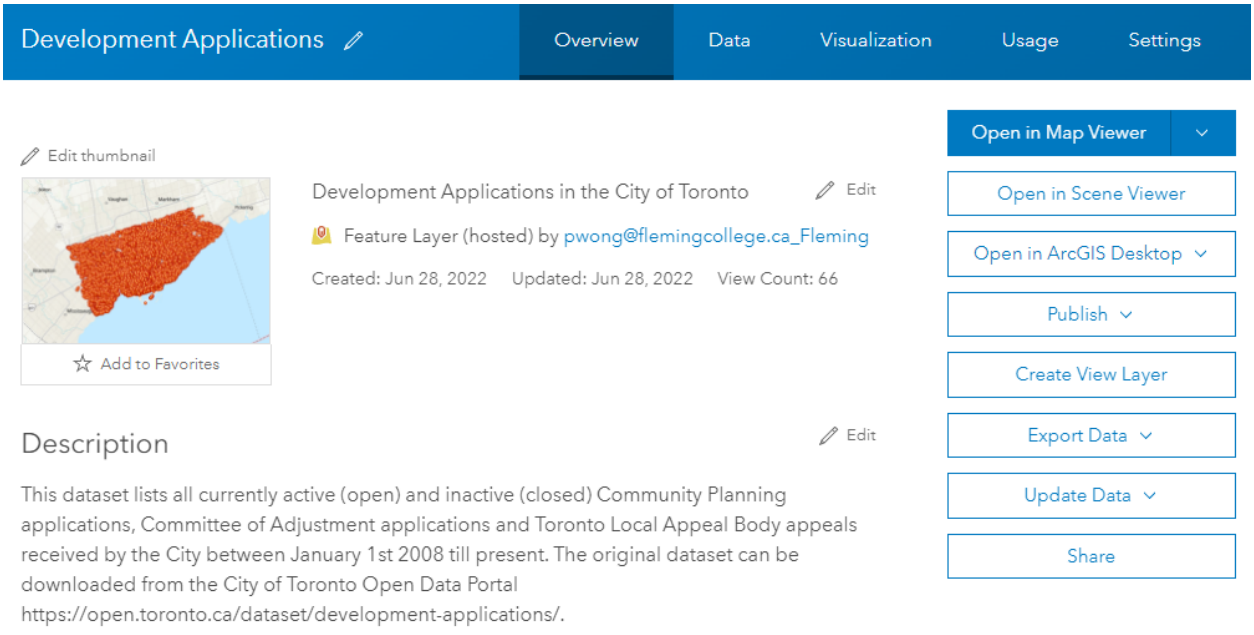

Figure 62 Item page of the development applications hosted feature layer

https://fleming.maps.arcgis.com/home/item.html?id=e960047df387456185e38e8f6aac616c

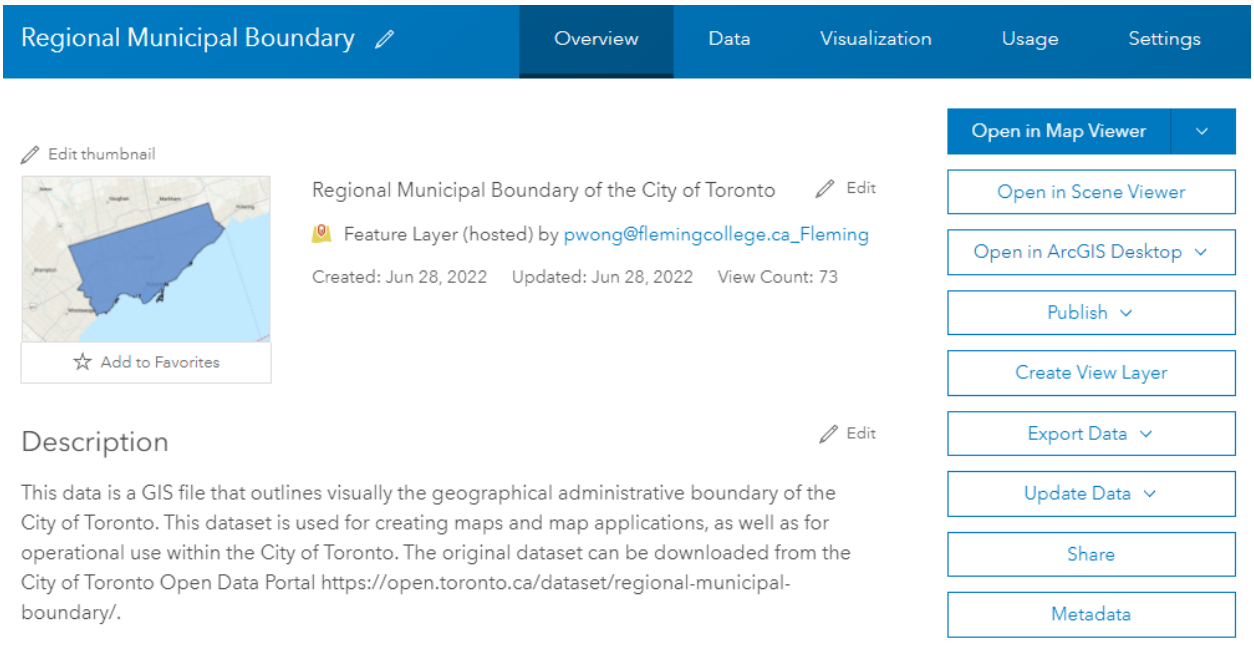

Figure 63 Item page of the regional municipal boundary hosted feature layer

3.6.2 Historic Aerial Imagery Map

The historic aerial imagery map was used in the story map only for educational purposes. It was applied to the swipe block in the story map, allowing the users to compare with the current aerial imagery interactively.

https://fleming.maps.arcgis.com/apps/mapviewer/index.html?webmap=4564d41ac2ca40bca239908543c20c55

Figure 64 Overview of the historic aerial imagery map

3.6.3 Feedback Survey

The feedback survey was powered by ArcGIS Survey 123 and was linked from the dashboard and the web application, allowing the users to provide feedbacks such as problematic data. All submissions will be stored in the client's account for review, and the client can use the received feedbacks for future enhancements.

https://survey123.arcgis.com/share/6bf1627106fa4fbc8f8010a9b90e98a7

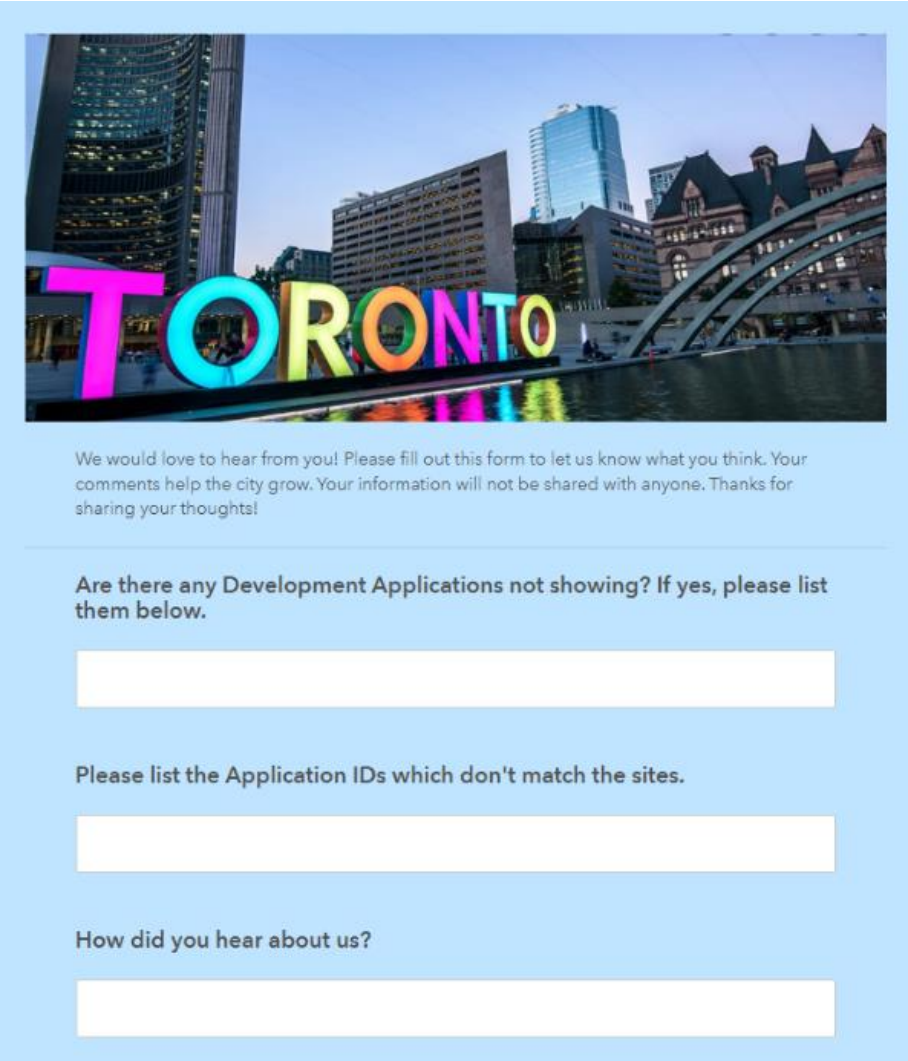

Figure 65 Overview of the feedback survey

3.6.4 Python Setup Guide

A step-by-step Python setup guide was created to assist the client to set up the web maps and the automated updates. The setup guide includes detailed instructions with screenshots in plain language, thus the staffs who did not use ArcGIS Notebooks before can still follow the instructions and set up the scripts easily.

https://flemingflamingos.github.io/development-activity-tracker/python-setup-guide.pdf

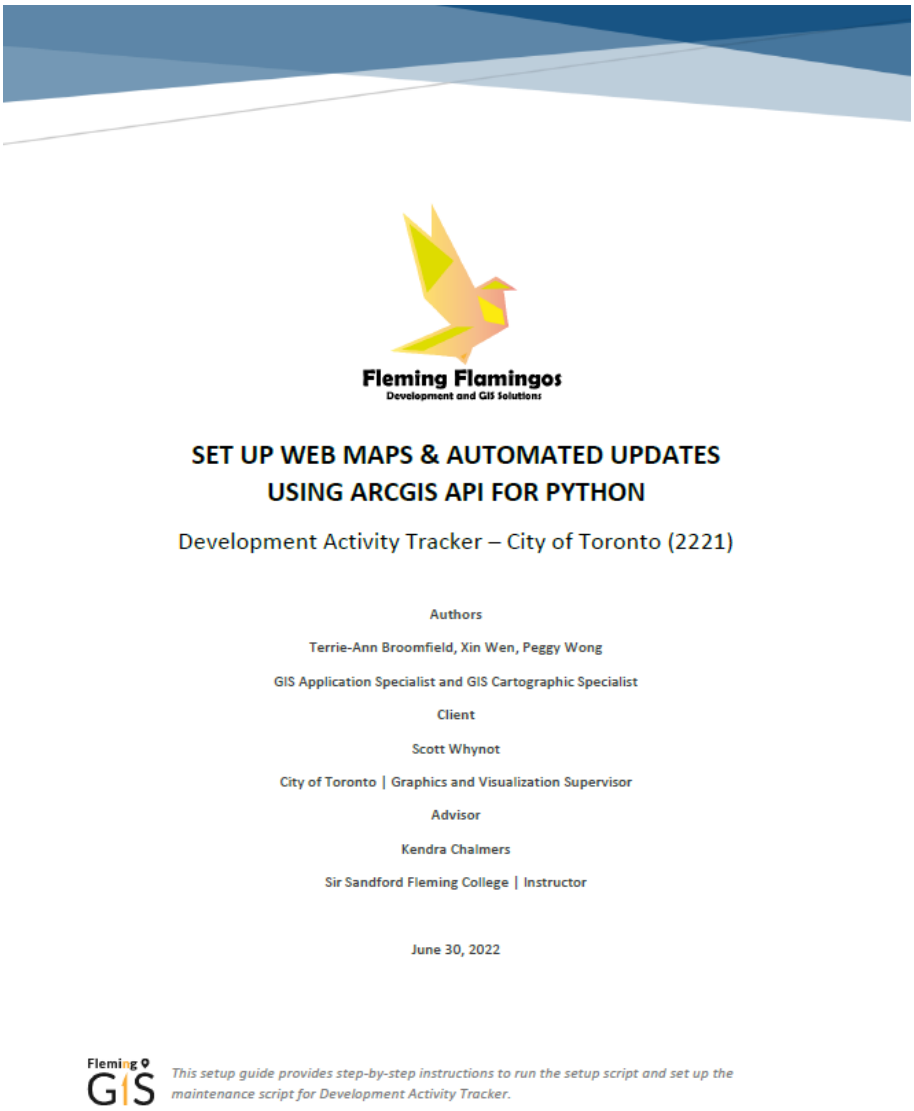

Figure 66 Cover page of the setup guide

4. Discussion

4.1 Summary

The project has met the three main objectives that were identified successfully. The web maps and the web applications allow the users to track the development activities in Toronto interactively, with strategic designs to promote effective communication. After receiving numerous feedbacks during the project implementation, the current interfaces are easy to navigate and user friendly. On the other hand, the maintenance script checks the updates on the City of Toronto Open Data Portal daily to minimize the maintenance effort of the client.

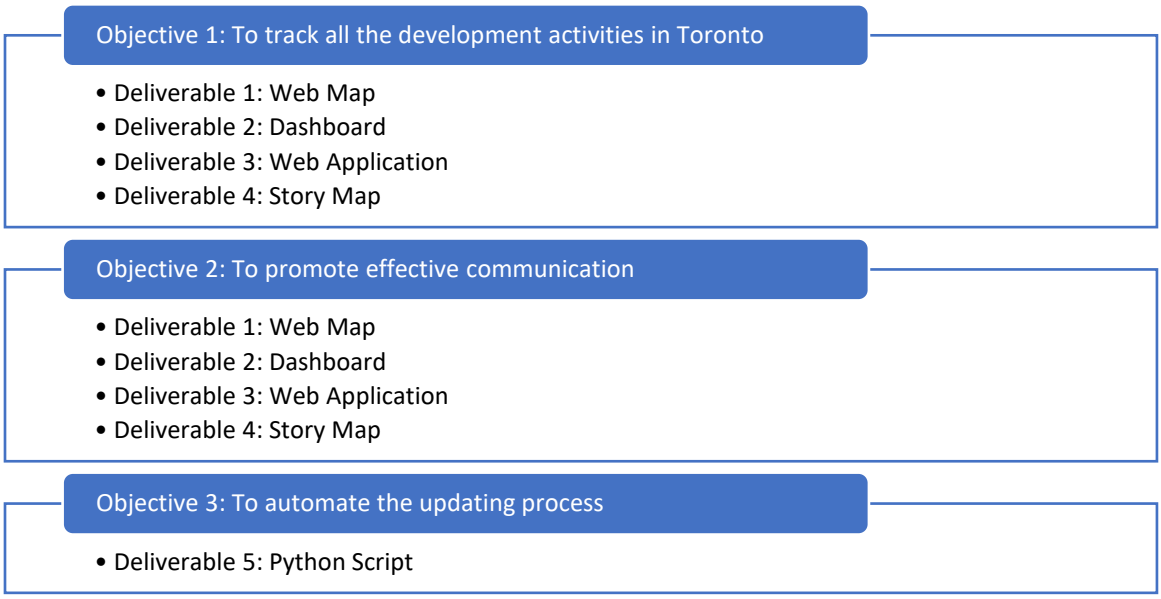

Figure 67 Projective objectives and relevant deliverables

4.2 Limitations

A number of limitations were identified throughout the project implementation. First, the filters and the temporal visualizations cannot be applied to a layer with a heatmap renderer (Esri, 2022). Therefore, the time slider of the web maps and the selector element of the dashboard do not work on the heat map. Second, the category selectors of the dashboard will be unresponsive if the category selectors use a large number of categories. Therefore, the application number selector and the reference file number selector were not added to the

dashboard. Finally, the web maps and the web applications may have a long loading time due to the size of the dataset, which may affect the user experience.

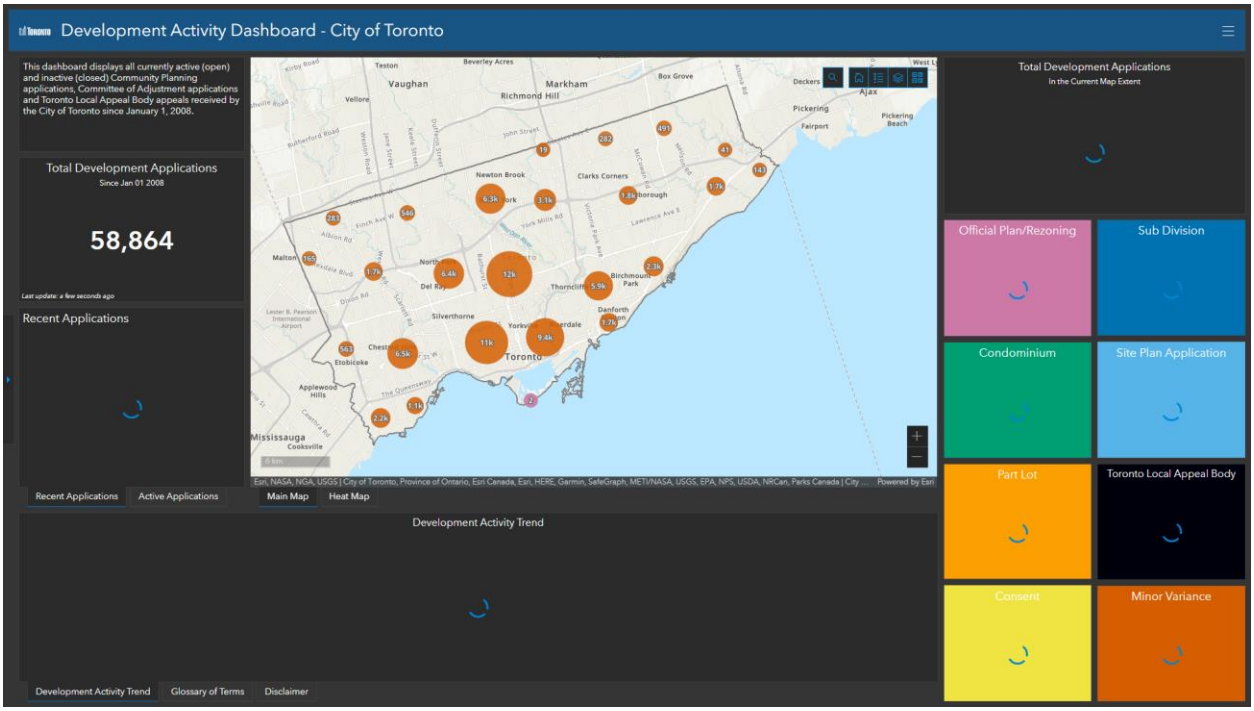

Figure 68 Loading screen of the dashboard which took around 15s to load completely

4.3 Recommendations

To further enhance the project in the future, improvement on data integrity is highly recommended. The current development applications dataset has several problematic features, such as missing or incorrect coordinates, incorrect dates, or unformatted descriptions. To maximize the project effectiveness, it is important to ensure the accuracy and the completeness of the data. Also, the development applications dataset should also include the URL of corresponding application details, providing additional information about the application such as the timeline of the application process, supporting documentation, and contact information of the planner.

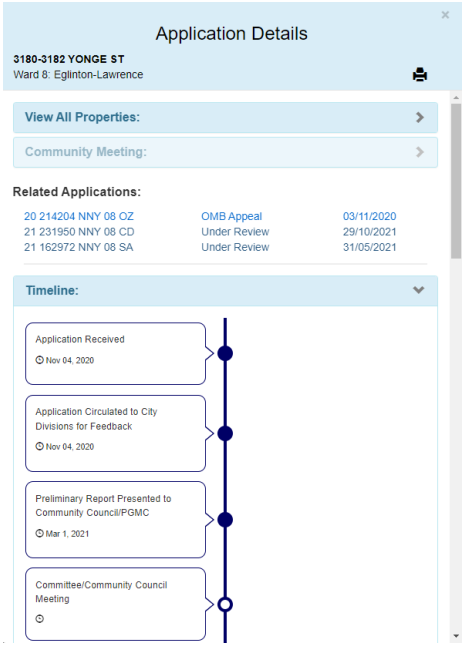

Figure 69 Development application details

Finally, regular check-ups on the web maps and the web applications are recommended to ensure the web maps and the web applications are fully updated and running optimally.

4.4 Conclusion

This project has enhanced the existing web application developed by the City of Toronto so that the users are able to explore the development activities interactively through the web maps and the web applications. It was designed for both internal and external use, allowing different stakeholders to research development activities in Toronto. Our team look forward to presenting the solution to the City of Toronto and shape the future of inclusion in the GIS community, with the created medium that is accessible and hosts all the needed information on the development activities in one place.

References

City of Toronto. (2022). Development Applications – City of Toronto Open Data Portal. Retrieved from https://open.toronto.ca/dataset/development-applications/

City of Toronto. (2022). Development Applications. Retrieved from https://secure.toronto.ca/AIC/index.do

City of Kenosha. (2021). Kenosha Developments – Overall. Retrieved from https://www.arcgis.com/apps/dashboards/c1620b1dfaec4c5fa6897df5f74e1a90

City of Toronto (2022). The History of Toronto: An 11,000 Year Journey – City of Toronto. Retrieved from https://www.toronto.ca/explore-enjoy/history-art-culture/museums/virtual-exhibits/history-of-toronto/

Kovac, Lisa (2018). What is the AODA?. Retrieved from https://aoda.ca/what-is-the-aoda/

Esri (2022). API Reference for the ArcGIS API for Python | ArcGIS API for Python. Retrieved from https://developers.arcgis.com/python/api-reference/

City of Toronto (2019). Regional Municipal Boundary - City of Toronto Open Data Portal. Retrieved from https://open.toronto.ca/dataset/regional-municipal-boundary/

City of Toronto (2018). basemap/cot_historic_aerial_1939 (MapServer). Retrieved from https://gis.toronto.ca/arcgis/rest/services/basemap/cot_historic_aerial_1939/MapServer/

City of Toronto (2022). Mapping Glossary – City of Toronto. Retrieved from https://www.toronto.ca/city-government/data-research-maps/maps/purchase-maps-data/mapping-glossary/

Esri (2021). Projected Coordinate Systems. Retrieved from https://pro.arcgis.com/en/pro-app/2.9/help/mapping/properties/pdf/projected_coordinate_systems.pdf

EarlMedina1 (2019). Methods for Updating Layer Symbology with the ArcGIS API for Python - Esri Community. Retrieved from https://community.esri.com/t5/arcgis-api-for-python-blog/methods-for-updating-layer-symbology-with-the/ba-p/902923

Okabe & Ito (2002). Color Universal Design (CUD) / Colorblind Barrier Free. Retrieved from https://jfly.uni-koeln.de/color/

Nichols, David (n.d.). Coloring for Colorblindness. Retrieved from https://davidmathlogic.com/colorblind/

Esri (2022). ArcGIS Dashboards | Data Dashboards: Operational, Strategic, Tactical, Informational. Retrieved from https://www.esri.com/en-us/arcgis/products/arcgis-dashboards/overview

Esri (2022). Header—ArcGIS Dashboards | Documentation for ArcGIS Enterprise. Retrieved from https://enterprise.arcgis.com/en/dashboards/latest/get-started/header.htm

Esri (2022). Indicator—ArcGIS Dashboards | Documentation for ArcGIS Enterprise. Retrieved from https://enterprise.arcgis.com/en/dashboards/latest/get-started/indicator.htm

Esri (2022). Selectors—ArcGIS Dashboards | Documentation. Retrieved from https://doc.arcgis.com/en/dashboards/create-and-share/selectors.htm

Esri (2022). List—ArcGIS Dashboards | Documentation. Retrieved from https://enterprise.arcgis.com/en/dashboards/latest/get-started/list.htm

Esri (2022). Serial chart—ArcGIS Dashboards | Documentation for ArcGIS Enterprise. Retrieved from https://enterprise.arcgis.com/en/dashboards/latest/get-started/serial-chart.htm

Esri (2022). Map widget—ArcGIS Experience Builder | Documentation. Retrieved from https://doc.arcgis.com/en/experience-builder/configure-widgets/map-widget.htm

Esri (2022). List widget—ArcGIS Experience Builder | Documentation. Retrieved from https://doc.arcgis.com/en/experience-builder/configure-widgets/list-widget.htm

Esri (2022). Legend widget—ArcGIS Experience Builder | Documentation. Retrieved from https://doc.arcgis.com/en/experience-builder/configure-widgets/legend-widget.htm

Esri (2022). Button widget—ArcGIS Experience Builder | Documentation. Retrieved from https://doc.arcgis.com/en/experience-builder/configure-widgets/button-widget.htm

Esri (2022). What is ArcGIS StoryMaps?—ArcGIS StoryMaps | Documentation. Retrieved from https://doc.arcgis.com/en/arcgis-storymaps/get-started/what-is-arcgis-storymaps.htm

Esri (2022). Add swipe blocks—ArcGIS StoryMaps | Documentation. Retrieved from https://doc.arcgis.com/en/arcgis-storymaps/author-and-share/add-swipes.htm

Esri (2022). Add sidecars—ArcGIS StoryMaps | Documentation. Retrieved from https://doc.arcgis.com/en/arcgis-storymaps/author-and-share/add-sidecars.htm

Dodd, Paul (2021). Manage Notebook Code Dependencies (MNCD) Installer – Overview. Retrieved from https://www.arcgis.com/home/item.html?id=46c7512604654601ab4338f9299c5414

Esri (2022). Overwrite Hosted Feature Services, v2.1.2 – Overview. Retrieved from https://www.arcgis.com/home/item.html?id=d45f80eb53c748e7aa3d938a46b48836

Esri (2022). HeatmapRenderer | API Reference | ArcGIS API for JavaScript 4.23 | ArcGIS Developer. Retrieved from https://developers.arcgis.com/javascript/latest/api-reference/esri-renderers-HeatmapRenderer.html

City of Toronto. (2022). Pre-Application Consultation. Retrieved from https://www.toronto.ca/city-government/planning-development/application-forms-fees/pre-application-consultation/

City of Toronto. (2022). Official Plan and Zoning By-law Amendment. Retrieved from https://www.toronto.ca/city-government/planning-development/application-forms-fees/building-toronto-together-a-development-guide/official-plan-and-zoning-by-law-amendment/

City of Toronto. (2022). Committee of Adjustment. Retrieved from https://www.toronto.ca/city-government/planning-development/committee-of-adjustment/

City of Toronto. (2022). Toronto Local Appeal Body – Filing an Appeal. Retrieved from https://www.toronto.ca/city-government/planning-development/committee-of-adjustment/appeals/filing-an-appeal/

Slavin, A. J. (2015). A survey of the number of legally blind university physics students in Canada during 2003-2013. Canadian Journal of Physics, 93(1), 1+

Ross, T. (2013). Advancing Ontario's accessibility: a study of linguistic, discursive, and conceptual barriers. Canadian Journal of Urban Research, 22(1), 126+

Appendix A: Literature Review – Development Application Process in Toronto

Pre-Application Consultation (City of Toronto, 2022)

Although pre-application consultation is not required for the development review process, the applicants are encouraged to meet with the city staff to discuss the development proposal and the requirements of an application before submitting a formal application. To request a meeting, the applicants will need to submit a submission package, including a completed request form, concept site plan and concept elevations or renderings. As the purpose of the consultation is to discuss the requirements of an application, the applicants will not receive any evaluation from the city staff during the consultation.

Official Plan and Zoning By-law Amendment (City of Toronto, 2022)

An Official Plan Amendment or a Zoning By-law Amendment must be applied if the applicants wish to use, alter, or develop the property in a way that does not conform with the Official Plan or Zoning By-law. The application requirements can vary, depending on the nature of the property and the proposal. Once the application is completed, it will be circulated for detailed technical review and comment. The applicants will have an opportunity to publicly present their proposals and receive feedbacks from the local community in the community consultation meeting. After that, a response form the Planner will be forwarded to the applicants for the purpose of revision and resubmission of the proposal. Once the application has been finalized, a Public Meeting will be held at Community Council and Community Council will make a final decision.

Committee of Adjustment (City of Toronto, 2022)

The Committee of Adjustment is responsible for considering applications for minor variances and consents. For example, altering or developing the property in a way that does not conform with the Zoning By-law, or dividing the land into new lots. The application process includes pre-application consultation, preliminary project review, application submission, scheduling of hearing and posting of public notice sign, notice of public hearing and application circulation,

Committee of Adjustment hearing, and Committee of Adjustment decision. If the applicants are not satisfied with the decisions, they can submit an appeal to the Toronto Local Appeal Body.

Toronto Local Appeal Body Appeals – Filing an Appeal (City of Toronto, 2022)

The applicants are able to file an appeal if they disagree with a decision of the Committee of Adjustment, or if they have not received a decision within the legislated timelines after applying for planning approval. To file an appeal, the applicants will need to complete the Notice of Appeal and submit it to the Manager & Deputy Secretary Treasurer of the Committee of Adjustment. Once the appeal information is submitted and verified, the Notice of Hearing, including the hearing date, time, and location, will be emailed to the applicants.

Appendix B: Literature Review – Accessibility for Ontarians with Disabilities Act (AODA)

The concept of accessibility in a website is making the content usable for all prospective users. The intended audience may include people with different abilities, who might require special functions to navigate the site. The Accessibility for Ontarians with Disability Act (AODA) outlines the terms, compliance, tips, and guidelines to foster learning how to make a website and other applications accessible (Province of Ontario, 2022).

A survey of the number of legally blind university physics students in Canada during 2003-2013 (Slavin, 2015)

A study reveals that 3.2% of Canadians aged 15 years or older have vision impairment, with 10.8% being classified as legally blind. Ability limitation creates a huge impact to Canadian unemployment rate and education rate. In 2006, 35.7% of legally blind people between age 15-64 are unemployed or have limited access to labor force. There is only a small portion of students with low vision have enrolled in post-secondary programs, and it is very challenging for them to meet complex school requirements. Canada is one of the countries that have legislated the provision of assistive devices to help people with disabilities.

Advancing Ontario's accessibility: A study of linguistic, discursive, and conceptual barriers (Ross, 2013)

There are two arguments after the enactment of the Accessibility for Ontarians with Disabilities Act 2005 (AODA). One is AODA texts are underpinned by a biomedical definition of disabled groups. This only helps a little support to social purpose and solely for people with disability not all Ontarians. Another one is AODA can be seen as a standard that could help to advance accessibility by regulating Ontario's planners, architects, and urban/community designers to eliminate the barriers and disabling. The process of building a standard costs money and time, and people judge in different perspectives on "who is benefited from it" and "who is served by accessibility".