CS7CS3 Advanced Software Engineering Group Project

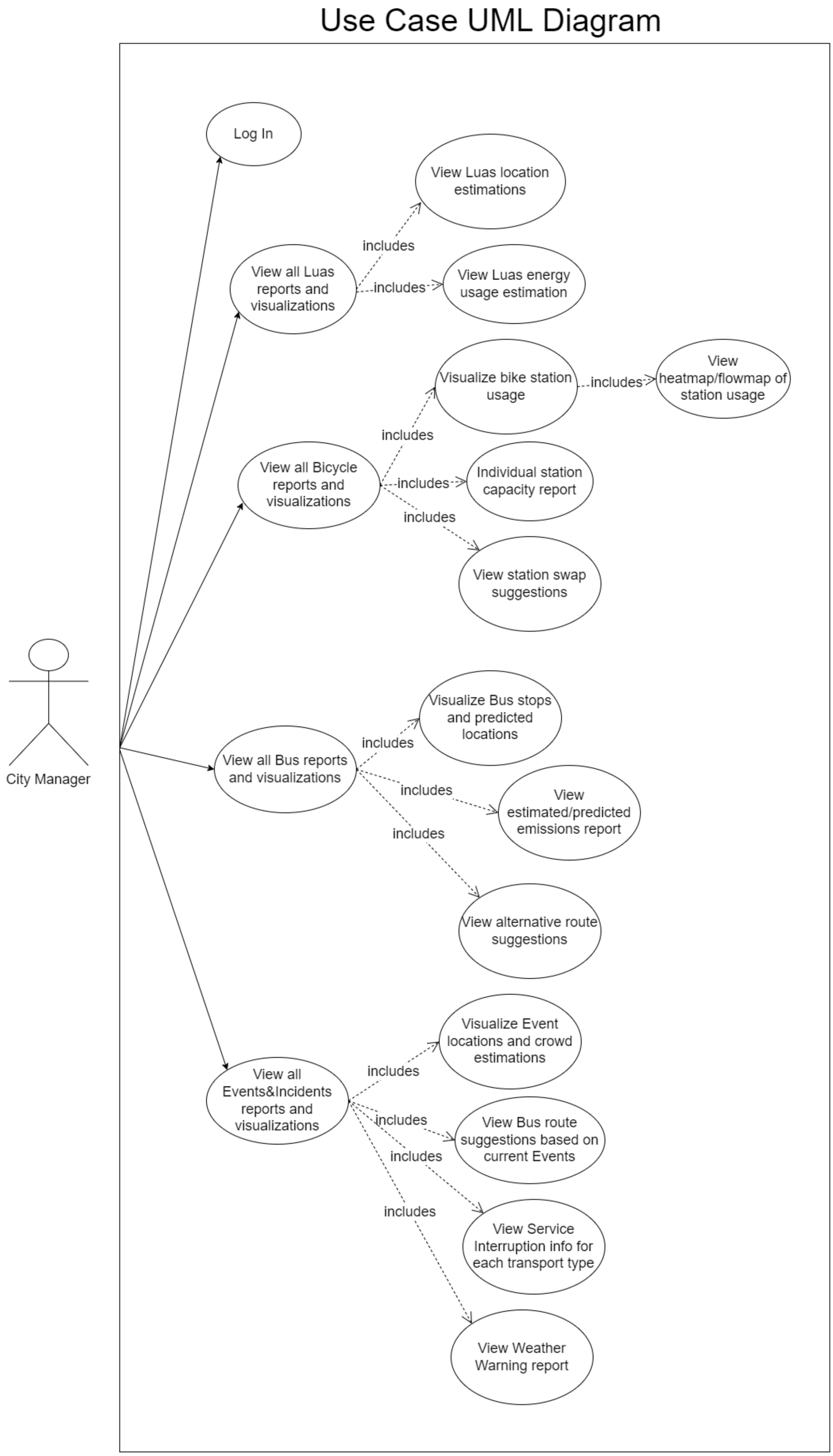
Requirements/Use Cases

Project Name: *SUSTAINABLE CITY MANAGEMENT*

Group: *5*

Yatheendra Pravan K M 20305901	Lokesh Selvakumar 21331969	Karina Salem 20323363	Kevin Reynolds 20308229
Kalaivani Kandasamy 21332195	Deeksha Vyas 21334447	Ayush Kalra 20327429	Shubham Maurya 20310455

Use Case Diagram



Use Case Name: Data Pipeline

1. Description

Goals:

1. Every 5 minutes, retrieve the live Luas, Dublin Bus, Dublin Bikes and events data from external data sources.
2. From this live data, predict the next 24 hours of data.
3. Live & predicted data is transformed into suitable data for visualization.
4. Statistics are calculated based on most recent live data and historical statistics, then new statistics are saved in the backend database.
5. Transformed data is passed to the Web Service Provider, which will publish data to all local databases on subscribed devices.

Responsibilities:

The Data Pipeline is responsible for retrieving live data for all data indicators, using this data to generate predicted data for offline use, and transforming this data into geo-markers for vehicles, bus stop locations, bike station usage statistics etc., which is then passed to Web Service Provider (Firebase).

Actors:

Live Data Handler – every 5 minutes, it retrieves the live data for each data indicator and passes it to the Prediction Engine, beginning the Data Pipeline process.

Triggers and Inputs

Triggers:

Every 5 minutes, the Live Data Handler begins the data pipeline process by retrieving the live data and passing it to the Prediction Engine

Inputs:

No inputs. The Use Case will execute in the same manner every 5 minutes.

2. Flow of Events

1. **Live Data Handler** validates the external APIs and retrieves the live data for each data indicator. This data is passed to the Prediction Engine.
2. **Prediction Engine** uses the live data to predict the next 24 hours of data, using a pre-trained prediction model. Live and predicted data is passed to the Data Transformer.
3. **Data Transformer** transforms live and the predicted data that can be visualised in our UI (i.e., map markers). It also calculates statistics (e.g.,

CO_2 emissions) from the live data, predicted data and previously calculated historical statistics kept in the **backend database**. All transformed data is pushed to the **Web Service Provider** and then new statistics are pushed to the **backend database**.

3. Special Requirements

This use case requires the existence of external live data sources for each of the data indicators.

4. Preconditions

This use-case has no pre-conditions, as it is the basis of our systems. All other use cases will depend on the operation of this use case.

5. Postconditions

On completion, the Web Service Provider will contain transformed data (live and predicted data) for the next 24 hours, split into 5-minute intervals. The first 5-minute interval is based on the most recently retrieved live data, and the rest are based on predicted data.

In the case that a user's device is unable to connect to the Web Service Provider, the device will contain enough predicted data stored in the local data buffer to operate the application for 24 hours from when they last received data.

Use Case Name: Login and Firebase subscription

1. Description

Goals

1. Validate credentials entered by the user using Django/backend server.
2. On successful authentication, the UI Handler subscribes to Firebase, which will publish data to the local data buffer.

Responsibilities:

This use case is responsible for the authentication of users. Upon successful authentication the device will be subscribed to Firebase, which will update the local data buffer.

Actors

City Managers – Users trying to log-in to the application by accessing the login page and entering their credentials.

Triggers and Inputs

Triggers:

User opens the application and login page is the default page rendered in the UI.

Inputs:

1. User enters the username and password in the corresponding input boxes.
2. User presses the login button.

2. Flow of Events

Basic Flow			
User		System	
1	User opens the application.		
2	User inputs username and password.		
3	User clicks on the “Login” button.		
		4	The system validates credentials.
		5	If the credentials are correct, authentication is successful, and the device is subscribed to the Firebase server.
		6	The Firebase server pushes data to the local data buffer.
		7	The system redirects the user to the homepage.

Alternative Flow 1			
User		System	
1	User opens the application.		
2	User inputs username and password.		
3	User clicks on the “Login” button.		
		4	The system attempts to validate the credentials.
		5	If the credentials are wrong, the system shows an alert message and denies access.

3. Special Requirements

User accounts should have been already created.

4. Preconditions

The user details related to every user must be present in the backend database for authentication.

Transformed data should be available in Firebase.

5. Postconditions

On successful login, the home page is shown. The device is subscribed to the Firebase server, and data has been published from the Firebase server to the local data buffer.

If the user login failed, the login page is retained with an alert message displayed.

Use Case Name: Dublin Bikes - Station Visualization

1. Description

Goals:

1. Use live or predicted data to visualize bike availability at bike stations, and categorise stations based on the number of bikes available.
2. Show estimations for the number of bikes currently in use.

Responsibilities:

This use case is responsible for visualizing all Dublin Bike stations' information on a map of Dublin city. The visualization can also highlight stations that are almost full/empty.

Actors

City Managers – When selecting the 'Dublin Bikes' view, the Dublin bikes station visualization will be rendered.

Triggers and Inputs

Triggers:

1. User logs in to application.
2. User selects the 'Bike' dashboard view to display the Dublin Bikes Station Visualization.

Inputs:

1. User can select a filter to display only stations that are >90% full or >90% empty.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the 'Bike' dashboard view in the application.		

		2	The system retrieves the most-recent Dublin Bikes data from local database.
		3	Map of Dublin city is displayed, with the Dublin Bike stations' location and station information overlaid.
4	User selects station usage filter.		
		5	Chart is updated to show only the stations that match the filter.

3. Special Requirements

This use case requires the existence of live Dublin Bike data sources.

4. Preconditions

User must have logged in to the system and have sufficient privileges to view the Dublin Bikes Station Visualization.
Live and predicted data must have been pushed to the local data buffer.

5. Postconditions

Once this use case is completed, the Dublin Bikes Station Visualization will be rendered with information based on:
1. the most recent live or predicted data and
2. the filters selected by the user.

Use Case Name: Dublin Bikes - Station Usage

1. Description

Goals:
1. Use live (or predicted) data to calculate current usage statistics for each of the stations.
2. Use historical data to generate historical usage statistics.
3. Rank stations based on over/under use, display their information on a bar chart.

Responsibilities:
This use case is responsible for calculating usage statistics for Dublin Bikes stations, i.e., what percentage of the day was any given station:
- Full
- >90% Full
- >90% Empty
- Empty

The worst performing stations will be displayed on a bar chart. The bar chart can be filtered based on time-period selected, showing station statistics for last 24 hours, 48 hours, 7 days etc.

Actors
1. City Managers – When selecting the ‘Dublin Bikes’ view, the Dublin Bikes Station Usage chart will be rendered.

Triggers and Inputs

Triggers:
1. User logs in to application.
2. User selects the ‘Bike’ dashboard view to display the Dublin Bikes Station Usage chart.

Inputs:
1. User can select a date range to calculate station statistics.
2. User can select a usage statistic (i.e., over-use, under-use) to rank stations by.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the ‘Bike’ dashboard view in the application.		
		2	The system retrieves the most-recent station statistics from the local database.
		3	Bar chart is displayed using the default usage statistic (over-use) and default time-period (24hrs)
4	User selects a usage filter.		
		5	Chart is updated to show the usage statistics for each station.
6	User selects a date range.		
		7	Chart is updated to show the station statistics across the selected time-period.

3. Special Requirements

This use case requires the existence of live Dublin Bike data sources.

4. Preconditions

User must have logged in to the system and have sufficient privileges to view the Dublin Bikes Station Usage chart.
Live and predicted data must have been pushed to the local data buffer.

5. Postconditions

Once this use case is completed, the Dublin Bikes Station Usage chart will be rendered with information based on:

1. the most recent live or predicted data and
2. the filters selected by the user.

Use Case Name: Dublin Bikes - Heat-map

1. Description

Goals:

1. Displaying a heat map for Dublin bike stations based on the bike station usage for various locations in the city.
2. Provide filtering of visualisations based on the stations present in the different areas on the map.

Responsibilities:

This use case will be responsible for getting the information regarding the availability of bikes and bike stations on the map, highlighting areas of over/under-use.

Actors

1. City Managers – When selecting the 'Dublin Bikes' view, the Dublin Bikes Heat-map Visualization will be rendered.

Triggers and Inputs

Triggers:

1. User Logs into the application.
2. User selects 'Bikes' from the dashboard view.
3. Heat-map of Dublin city is displayed, with all the Dublin Bike station information displayed.
4. Clicking on a particular area on the map will display the relevant information.

Inputs:

1. User can filter between the various areas present.
2. Based on the area selected the corresponding information can be displayed.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the 'Bike' dashboard view in the application.		

		2	The system retrieves the most-recent bike data present from the local data buffer.
		3	Heat-map is displayed of the various areas the Dublin City with no particular area selected (default).
4	User selects an area.		
		5	Heat-map of the selected area is displayed with all the necessary information.

3. Special Requirements

This use case requires the existence of live Dublin Bike data sources.

4. Preconditions

User must have logged in to the system and have sufficient privileges to view the Dublin Bikes Heat-map visualization.
Live and predicted data must have been pushed to the local data buffer.

5. Postconditions

Once this use case has been completed, the Dublin bikes Heat-map visualization will be rendered with information based on:

1. The most recent live or predicted data.
2. The filters selected by the user.

Use Case Name: Dublin Bikes – Swap Suggestions

1. Description

Goals:

Suggest bike swaps between over-capacity and under-capacity stations based on live bike station information.

Responsibilities:

This use case is responsible for using the live data to create suggestions that would help evenly distribute the bikes within the Dublin Bike stations throughout the city.

Actors

1. City Managers – When selecting the ‘Bikes’ view of the site, City Managers will cause the Suggested Bike Swap dashboard view to be displayed.

Triggers and Inputs

Triggers:

1. User logs in to the application.
2. User selects the ‘Bikes’ dashboard view.
3. The usage of each bike station is calculated.
4. Bike swaps suggestions based on over-capacity and under-capacity of the stations are created.

Inputs:

This use case has no inputs, as suggestions are dependent on the live data.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the ‘Bike’ dashboard view in the application.		
		2	The system retrieves the live bike station data from the local data buffer.
		3	The usage of each bike station is calculated and swaps between stations with over-capacity and under-capacity are created.
		4	The bike swaps suggestions are displayed to the user.

3. Special Requirements

This data requires the existence of live Dublin Bikes data sources.

4. Preconditions

The user must have logged in to the system and have sufficient privileges to view the Bike swap suggestions.

Bike station data must have been pushed to the local data buffer.

5. Postconditions

Once this use case has been completed, the Bike swaps suggestions have been created based on the available bike stations’ data.

Use Case Name: Buses – Map & Heat-map of Bus/Bus Stop Locations

1. Description

Goals:

1. Use Live data to determine current bus locations.
2. Display map which shows all the bus stops.

3. Display a heatmap that shows the current distribution of buses within the city.

Responsibilities:

This use case is responsible for displaying the location of buses and bus stops so the City Manager can know the availability of buses at any given time and location to make rapid decisions during the time of emergencies.

Actors

1. City Managers – City Managers can select the Map or Heatmap to get the bus and bus stop locations in the desired view.

Triggers and Inputs

Triggers:

1. User logs in to the application.
2. User selects the ‘Bus’ dashboard to view the Map and Heatmap inside it.

Inputs:

1. Users can select between the standard map and heat-map views.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the ‘Bus’ dashboard view in the application.		
		2	The system retrieves most-recent Dublin Bus data from the local database.
		3	Map of Dublin city is displayed, with the bus stops and estimated bus locations overlaid.
4	User selects the Heat-map view		
		5	Heatmap is displayed over the standard map.

3. Special Requirements

This data requires the existence of live Dublin Bus data sources.

4. Preconditions

Users must have logged in to the system and have sufficient privileges to view the Bus Map/Heat-Map visualization.

Live data must have been pushed to the local data buffer.

5. Postconditions

Once this use case has been completed, the Bus and Bus Stop locations will have rendered based on the most recent live data of the bus.

Use Case Name: Buses - Environmental Impact

1. Description

Goals:

1. To estimate the number of buses in the area.
2. To calculate the CO_2 emission based on the number of buses.

Responsibilities:

1. Use case will visualize the live CO_2 emission levels which can be used by city mangers to check environmental impact.

Actors

City Managers – When selecting the 'Bus' view of the site, it will show CO_2
emissions levels throughout the city.

Triggers and Inputs

Triggers:

1. User logs in to application.
2. User selects the 'Bus' dashboard view to display the visualization.

Inputs:

1. User can select an area to view the CO_2 levels.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the 'Bus' dashboard view in the application.		
		2	The system retrieves the most-recent Dublin Bus data from local data buffer.
		3	Map of Dublin city is displayed with the bus locations and CO_2 emission levels.
4	User selects Max and Min CO_2 levels view-filter.		
		5	System shows the areas which has highest and lowest CO_2 levels.

3. Special Requirements

This data requires the existence of live Dublin Bus data sources and average emission of buses within each area.

4. Preconditions

User must have logged in to the system and have sufficient privileges to view the CO_2 emissions.
Live and predicted data must have been pushed to the local data buffer.

5. Postconditions

Once this use case has been completed, the CO_2 visualization has been rendered with information based on:

1. The most recent live data.
2. The filters selected by the user.

These are subject to change as the live location of buses will change thus data generated will be dynamic and it must be refreshed.

Use Case Name: Buses **-** Predict CO_2 Emissions

1. Description

Goals:

1. Display CO_2 emission comparisons of the buses in the city on a graph for the following days.
2. If live data is not available, simulate with most-recent historic data.

Responsibilities:

This use case is responsible for visualizing predicted CO_2 emissions caused by the busses in Dublin city.

Actors

City Managers – When selecting the ‘Bus’ view of the site, City Managers will request for CO_2 emission comparison.

Triggers and Inputs

Triggers:

1. User logs in to the application.
2. User selects the ‘Bus’ dashboard view to display the visualization.

Inputs:

No inputs. This chart will be shown every time the user selects the ‘Bus’ dashboard view.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the 'Bus' dashboard view in the application.		
		2	The system retrieves the most-recent Dublin Bus data from the local data buffer.
		3	The system calculates the CO_2 emission predictions for the following days.
		4	Graph is displayed with CO_2 emission predictions per area.

3. Special Requirements

This data requires the existence of live Dublin Bus data sources.

4. Preconditions

Users must be logged in to the system and have sufficient privileges to view the Predicted CO2 visualisation.
Live and predicted data must have been pushed to the local data buffer.

5. Postconditions

Once this use case has been completed, Graph with CO_2 emission comparisons for the following days based on the most recent live or predicted data.

Use Case Name: Buses **-** Rerouting Suggestions

1. Description

Goals:
1. Suggest re-routing of the buses based on congestion/ CO_2.
2. Suggest changes in timings and increasing the frequency based on busy periods.

Responsibilities:

The application will be able to suggest new routes for buses in the areas where the CO_2 emissions are high thereby reducing pollution levels in the affected area.

Actors
1. City Managers – City Managers will be able to visualise new routes for the buses.

Triggers and Inputs

Triggers:
1. User logs in to the application.
2. User selects the ‘Bus’ dashboard to view the suggestions.

Inputs:

No inputs. This chart will be shown every time the user selects the ‘Bus’ dashboard view.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the ‘Bus’ dashboard view in the application.		
		2	The system retrieves the new routes suggestions from the local database.
		3	New routes are overlaid in the suggestion's column on dashboard.

3. Special Requirements

This data requires the existence of historical and live Bus and CO_2 emissions data sources.

4. Preconditions

Users must have logged in to the system and have sufficient privileges. Live data must have been pushed to the local data buffer containing the new routes for the buses.

5. Postconditions

Once this use case has been completed, the Bus Route Suggestions chart will be displayed, based on the most recent live or predicted data.

Use Case Name: Luas - Map Visualization

1. Description

Goals:
1. Display the most recent location of all Luas trams, Luas stops, and service interruptions in the city on a map of Dublin city.
2. Provide filtering of visualizations based on Red/Green line or travel direction.

Responsibilities:

This use case is responsible for visualizing the live (or predicted) Luas locations on a map of Dublin city.

Actors

City Managers – When selecting the ‘Luas’ view of the site, City Managers will cause the Luas Location Visualization to be created.

Triggers and Inputs

Triggers:

1. User logs in to application.
2. User selects the ‘Luas’ dashboard view to display the visualization.

Inputs:

1. User can filter between ‘Red Line’/’Green Line’/’Both’ view to show Luas trams travelling on either the Red Line, Green Line, or on both Lines.
2. User can filter between ‘Northbound’/ ‘Southbound’ to only show Luas trams travelling in that direction.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the ‘Luas’ dashboard view in the application.		
		2	The system retrieves the most-recent Luas data from local data buffer.
		3	Map of Dublin city is displayed, with the Luas locations overlaid, with no filters (default).
4	User selects ‘Line’ or ‘Direction’ view-filter.		
		5	System removes filtered entities from the map.
6	User de-selects/changes filter.		
		7	System displays the required Luas entities.

3. Special Requirements

1. This data requires the existence of live Luas data sources.

4. Preconditions

User must have logged in to the system and have sufficient privileges to view the Luas visualisation.

Live and predicted data must have been pushed to the local data buffer.

5. Postconditions

Once this use case has been completed, the Luas map visualization has been rendered with information based on:

1. The most recent live or predicted data.
2. The filters selected by the user.

Use Case Name: Luas - Electricity Usage Estimates

1. Description

Goals:

1. Use live (or predicted) and historical data to estimate Luas's travel distances.
2. Calculate estimations for Luas electricity usage based on travel distances.
3. Display estimations on a chart within the 'Luas' view.

Responsibilities:

This use case is responsible for generating electricity usage estimates for all Luas trams in Dublin, allowing for comparison across a given period (days, weeks, months etc.).

Actors

1. City Managers – When selecting the 'Luas' view of the site, City Managers will cause the Luas Energy Estimation Chart to be created.

Triggers and Inputs

Triggers:

1. User logs in to application.
2. User selects the 'Luas' dashboard view to display the Luas Energy Estimation Chart.

Inputs:

User can select a date range to display all estimations within the selected range.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the 'Luas' dashboard view in the application.		
		2	The system retrieves the most-recent Luas data from local database.

		3	Luas energy usage estimations are displayed on a bar chart, within the default date range.
4	User selects a date range.		
		5	Chart is updated with daily estimations for the given date range.

3. Special Requirements

This data requires the existence of live Luas data sources.

4. Preconditions

User must have logged in to the system and have sufficient privileges to view the Luas electricity estimation.
Live and predicted data must have been pushed to the local data buffer.

5. Postconditions

Once this use case has been completed, the Luas Energy Estimation chart has been rendered with information based on:

1. The most recent live or predicted data.
2. The date range selected by the user.

Use Case Name: Event Locations and Estimated Crowd Size

1. Description

Goals:

1. To get a list of all the events scheduled within the city.
2. To get an estimation of the size of crowd from historic data.
3. Visualize the data on Dublin city map.

Responsibilities:

The use case is used to visualize events and estimated crowd size for events, which can be leveraged by city managers for crowd handling.

Actors

1. City Managers – When selecting the ‘Events/Incidents’ view of the site, City Managers will visualize the events and crowd size.

Triggers and Inputs

Triggers:

1. User logs in to the application.
2. User selects the ‘Events/Incidents’ dashboard view to display the events happening in the city and estimated crowd size on the map.

Inputs:

Users can select a date range to display all events within the selected range.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the 'Events/Incidents' dashboard view in the application.		
		2	The system retrieves the most-recent events data from a local database.
		3	Map of Dublin city is displayed, with the events locations and crowd size estimations.
4	User selects a date range.		
		5	Map will only display events and crowd size in the date range.

3. Special Requirements

This data requires the existence of live Events data sources and historical data of crowd sizes.

4. Preconditions

Users must have logged in to the system and have sufficient privileges to view the Events visualization.

5. Postconditions

Once this use case has been completed, the Events visualization has been rendered with information based on:

1. the most recent live or predicted data and
2. the date range selected by the user.

Use Case Name: Bus Route Frequency Suggestions

1. Description

Goals:

1. Display suggestions for increasing the number of buses in a route based on the events that are happening in the city on a map of Dublin city.
2. If live data is not available, simulate with most-recent historic data.

Responsibilities:

1. This use case is responsible for suggesting options to increase or decrease the frequency of bus routes based on the events that are happening the city.

Actors

1. City Managers – When selecting the ‘Events/Incidents’ view, the bus route frequency suggestions will be displayed.

Triggers and Inputs

Triggers:

1. User logs in to the application.
2. User selects the ‘Events/Incidents’ dashboard view to display the suggestions.

Inputs:

No inputs. Suggestions will be displayed every time the ‘Events/Incidents’ view is selected.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the ‘Events/Incidents’ dashboard view in the application.		
		2	The system retrieves the most-recent events data from the local database.
		3	The system retrieves the most-recent Bus data from the local database.
		4	Bus route frequency suggestions are displayed to the user.

3. Special Requirements

This data requires the existence of live Dublin Bus and events data sources.

4. Preconditions

1. User must be logged in to the system and have sufficient privileges to view the bus frequency suggestions.
2. Live and predicted data must have been pushed to the local data buffer.

5. Postconditions

Once this use case has been completed, the bus route frequency suggestions are displayed based on the most recent live or predicted Dublin Bus and events data.

Use Case Name: Maintenance Events

1. Description

Goals:

1. Use live data to display maintenance events for Dublin Bikes, Luas, Dublin Bus, and general roadworks.

Responsibilities:

1. This use case is responsible for showing maintenance details for each of the three modes of transport and any area that could be under maintenance.
2. City managers can use these maintenance details to make informed decisions regarding the usage of bikes, busses, and Luas on a particular road.

Actors

1. City Managers – When selecting the ‘Events/Incidents’ view, the Maintenance Events table will be rendered.

Triggers and Inputs

Triggers:

1. User logs in to the application.
2. User selects the ‘Events/Incidents’ dashboard to view the maintenance details.

Inputs:

1. Users can select for which mode of transport they wish to see maintenance details for.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the ‘Events/Incidents’ dashboard view in the application.		
		2	The system retrieves the most-recent maintenance data from the local data buffer for each of the mentioned transport types and roads if any.
		3	Two tabular displays will occur. One is Transport Maintenance Detail and another for Roads Maintenance.

4	User selects between the modes of transport.		
		5	The maintenance table is displayed for the selected view.

3. Special Requirements

This data requires the existence of live transport and road maintenance sources.

4. Preconditions

Users must have logged into the system and have sufficient privileges to view the Events and Maintenance details.
Live data must have been pushed to the local data buffer.

5. Postconditions

Once this use case has been completed, the maintenance details for all the specified modes and roads has been rendered based on the most recent data.

Use Case Name: Weather Conditions

1. Description

Goals:

Display MET Éireann weather warning notifications in case of extreme weather conditions.

Responsibilities:

This use case is responsible for notifying the users of extreme weather conditions to allow them to make informed decisions for the affected areas.

Actors

City Managers – When selecting the 'Events/Incidents' view of the site, City Managers will cause the weather warnings notifications sections to be displayed.

Triggers and Inputs

Triggers:

1. User logs in to the application.
2. User selects the 'Events/Incidents' dashboard view to view the weather warnings.

Inputs:

No inputs. Weather warnings displayed will be based on the most recent data.

2. Flow of Events

Basic Flow			
User		System	
1	User selects the 'Events/Incidents' dashboard view in the application.		
		2	The system retrieves the data from the MET Éireann weather warnings source.
		3	If there are any extreme weather conditions happening soon, they will be displayed to the user.

3. Special Requirements

This data requires the existence and availability of MET Éireann weather warnings data source.

4. Preconditions

1. The user must have logged in to the system and have sufficient privileges to view the weather warnings notification section.
2. Weather warning data must have been pushed to the local data buffer.

5. Postconditions

Once this use case has been completed, the warning notification of extreme weather conditions would have been displayed based on MET Éireann warning data.