# 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 |

# 1. Use Case Diagram

Diagram

Description automatically generated

### Use Case Name: Data Pipeline

1. Description

*Goals:*

1. Every 5 minutes, retrieve the latest 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 data suitable for visualization.
4. Statistics are calculated based on most recent live data and historical statistics kept in backend database.
5. Transformed data is passed to the Web Service Provider, which will publish data to all local databases on user 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 converting (transforming) this data into geo-markers for vehicles, bus stop locations, bike station usage statistics etc., which is then passed to 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 process by retrieve the external live data and passing it to the Prediction Engine

Inputs:

No inputs. The Use Case will execute in the same manner each time.

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 predicted data that can be visualised in our UI (i.e., map markers). It also calculates statistics (e.g., CO2 emissions) from the live data, predicted data andpreviously calculated historical statistics kept in the **Backend DB**. All transformed data is pushed to the **Web Service Provider** and thennew statistics are pushed to the **Backend DB**.
4. On receiving new data, **Web Service Provider** will publish the updated data to registered devices.

3. Special Requirements

This data 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 for the current 5-minute window (from live data) and for every 5-minute window for the next 24 hours (from predicted data). This data will be pushed to all registered devices.

In the case that a user’s device is unable to connect to the Web Service Provider, each device will contain enough predicted data to operate the application for 24 hours from when they were last online.

### 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, front end subscribes the data from firebase which will be pushed to the local data buffer.

*Responsibilities:*

This use case is responsible for the authentication and authorization of users. Upon successful authentication data is subscribed from the firebase.

Actors

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

Triggers and Inputs

*Triggers:*

User opens the application and log in 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 log in 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 button “Login” |  |  |
|  |  | 4 | The system validates credentials. |
|  |  | 5 | If the credentials are correct, authentication is successful, and the front end subscribes the data in the firebase which is pushed to local data buffer in the client side. |
|  |  | 6 | The system redirects the user to the homepage if the credentials are correct |

|  |  |  |  |
| --- | --- | --- | --- |
| Alternative Flow 1 | | | |
| User | | System | |
| 1 | User opens the application |  |  |
| 2 | User inputs username and password |  |  |
| 3 | User clicks on the button “Login” |  |  |
|  |  | 4 | The system validates credentials. |
|  |  | 5 | If the credentials are wrong, the system shows an alert message “credentials are incorrect, try again” and denies the 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 database for authentication and authorization.

Application data should be available in the firebase which is pushed from the data engine located at the backend.

5. Postconditions

If the User successfully logged in, the home page is shown. The front end of the system subscribes and gets the data from the firebase and stores the data required to render visualization in the local data buffer.

If the user login failed, the login page is retained with alert message “credentials are incorrect and try again”.

### Use Case Name: Dublin Bikes - Station Visualization

1. Description

*Goals:*

1. Use live (or predicted) data to visualize bike availability at bike stations, categorise stations based on number of bikes available.
2. Use station usage to estimate number of bikes currently in use.

*Responsibilities:*

This use case is responsible for visualizing all Dublin Bike station 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 ‘Dublin Bikes’ 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 locations and station information overlaid |
| 4 | User selects a usage filter. |  |  |
|  |  | 5 | Chart is updated to show only the stations that match the filter. |

3. Special Requirements

This data 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 has been completed, the Dublin Bikes Station Visualization has been 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 calculation 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 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 over.
2. User can select a usage statistic (i.e., over-use, under-use) to rank stations.

2. Flow of Events

|  |  |  |  |
| --- | --- | --- | --- |
| Basic Flow | | | |
| User | | System | |
| 1 | User selects the ‘Dublin Bikes’ 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 given time period. |

3. Special Requirements

This data 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 has been completed, the Dublin Bikes Station Usage chart has been 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/flow-map

1. Description

*Goals:*

1. Displaying a heat map for different bicycles stations present based on the bike data for different locations on the city.

2. Selection of a particular area on map and to display the number of stations and bikes present

3. 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 and also helps to visualise the areas where there is need for more bike stations.

Actors

City Managers - When selecting the map option for the ‘bicycles’, City Managers will be able to see the heatmap based on the data fetched from server.

Triggers and Inputs

Triggers:

1. User Logs into the application

2. User selects ‘bikes’ from the dashboard view.

3. User clicks on the ‘Generate Heat map’ option

4. Map of Dublin city is displayed, with all the Bike stations information displayed.

5. Clicking on particular area on the map and displaying 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 map option on ‘Bicycle’ menu of the dashboard |  |  |
|  |  | 2 | The system retrieves the most-recent bike data present on the buffer |
|  |  | 3 | The live data is converted to geographical co-ordinates and numbers of bike stations in different  areas |
|  |  | 4 | Heat Map is displayed of the various areas in the Dublin City  With no particular area selected(default) |
| 5 | User selects on a particular area |  |  |
|  |  | 6 | Heat Map of the particular area is displayed with all the necessary information |
| 7 | User selects on a particular bike station |  |  |

3. Special Requirements

This data requires the existence of historical bike stations 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 has been completed, the Luas map visualization has been rendered with information based on:

the most recent live or predicted data and

the filters selected by the user.

### Use Case Name: Dublin Bikes – swap suggestions

1. Description

*Goals:*

Suggest bike swaps between overcapacity and under-capacity stations based on live and historical bike stations information.

*Responsibilities:*

This use case is responsible for using the historical data to create suggestions that would solve the problem of overcapacity and under-capacity of bike stations in Dublin.

Actors

1. City Managers – When selecting the ‘Bicycles’ view of the site, City Managers will cause the Suggested bike swaps between stations to be created.

Triggers and Inputs

Triggers:

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

Inputs:

2. Flow of Events

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

3. Special Requirements

This data requires the existence of historical bike stations data sources.

4. Preconditions

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

Bike stations 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 - heatmap of bus/bus stops locations

1. Description

*Goals:*

1. Use Historical data for all the bus stops in Dublin.
2. Use Live data to determine current bus locations.
3. Display map which shows all the bus stops.
4. Display a heatmap that shows the current number of busses at each bus stop within the city. It will also show the probable number of buses at each location for the next few days based on the bus time-table gathered from historical data. This can change on the basis of live data for each day.

*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 the time period in the Heatmap to know the count of buses at any Bus Stop.

2. Flow of Events

|  |  |  |  |
| --- | --- | --- | --- |
| Basic Flow | | | |
| User | | System | |
| 1 | User selects the ‘Bus’ dashboard view in the application. |  |  |
|  |  | 2 | The system retrieves the historic and most-recent Bus data from the local database. |
|  |  | 3 | Map of Dublin city is displayed, with the Bus stops overlaid. Heatmap displayed on the side with each bus stop and bus detail. |
| 4 | User selects a time range. |  |  |
|  |  | 5 | Heatmap is updated with bus stop location and count of buses at that period of time. |

3. Special Requirements

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

4. Preconditions

Users must have logged in to the system and have sufficient privileges to view the Bus locations along with the Bus Stops.

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:

1. the most recent live data of the bus,
2. historic data of the bus stops and
3. the time period selected by the user.

### Use Case Name: Buses - Environmental Impact

1. Description

*Goals:*

1. To estimate the number of buses in the area
2. To calculate the CO2 emission on the basis on number of buses

*Responsibilities:*

1. The use case will visualize the live CO2 emission levels which can be used by city mangers to check environmental impact.

Actors

1. City Managers – When selecting the ‘Buses’ view of the site, It will show CO2 emissions levels throughout the city.

Triggers and Inputs

Triggers:

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

Inputs:

1. User can select an area to view the CO2 levels

2. Flow of Events

|  |  |  |  |
| --- | --- | --- | --- |
| Basic Flow | | | |
| User | | System | |
| 1 | User selects the ‘Buses’ dashboard view in the application. |  |  |
|  |  | 2 | The system retrieves the most-recent Buses data from local database. |
|  |  | 3 | Map of Dublin city is displayed, with the Buses locations and CO2 emission levels. |
| 4 | User selects Max and Min CO2 levels view-filter. |  |  |
|  |  | 5 | System shows the areas which has highest and lowest CO2 levels. |

3. Special Requirements

* 1. This data requires the existence of live Buses data sources and average emission of bus.

4. Preconditions

User must have logged in to the system and have sufficient privileges to view the CO2 emissions.

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

5. Postconditions

Once this use case has been completed, the CO2 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 CO2 emissions

1. Description

*Goals:*

1. Display CO2 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:*

1. This use case is responsible for visualizing CO2 emission comparison caused by the busses in Dublin city.

Actors

1. City Managers – When selecting the ‘Dublin Bus’ view of the site, City Managers will request for CO2 emission comparison.

Triggers and Inputs

Triggers:

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

Inputs:

1. User can select option to show the CO2 emission information

2. Flow of Events

|  |  |  |  |
| --- | --- | --- | --- |
| Basic Flow | | | |
| User | | System | |
| 1 | User selects the ‘Dublin Bus’ dashboard view in the application. |  |  |
|  |  | 3 | The system retrieves the most-recent Bus data from the local database. |
|  |  | 4 | Map of Dublin city is displayed, with the bus data overlaid |
| 5 | User selects option to show bus frequency suggestion |  |  |
|  |  | 6 | The system calculates the CO2 emission comparison for the following days |
|  |  | 7 | Graph is displayed with CO2 emission comparison |

3. Special Requirements

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

4. Preconditions

Users must be logged in to the system and have sufficient privileges to view the Dublin Bus 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 CO2 emission comparisons for the following days based on:

1. the most recent live or predicted data

### Use Case Name: Buses - rerouting suggestions

1. Description

*Goals:*

1. Suggesting re-routing of the buses based on congestions/Co2
2. Suggesting change 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 CO2 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:

1. .

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 suggestions colum on dashboard,. |

3. Special Requirements

This data requires the existence of historical and live Bus and CO2 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

### Use Case Name: Luas - Map Visualization

1. Description

*Goals:*

1. Display accurately the most recent location of all Luas trams, Luas stops, and service interruptions in the city on a map of Dublin city.
2. If live data not available, simulate with most-recent historic data.
3. Provide filtering of visualizations based on Red/Green line or travel direction.

*Responsibilities:*

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

Actors

1. 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 database. |
|  |  | 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’s 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 and
2. the filters selected by the user.

*<single use case description end>*

### 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’s 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 time 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:

1. 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 | Map of Dublin city is displayed, with the Luas locations overlaid, with no filters (default). |
| 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’s 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 and
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.
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 crow 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:**

1. 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. |
| 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 history data of crowd size

4. Preconditions

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

Size of the crowd is subjective to various conditions (Covid-19, Raining etc.)

5. Postconditions

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

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

*<single use case description end>*

### 

*<single use case description start>*

### Use Case Name: Increasing bus route frequency suggestions

1. Description

*Goals:*

1. Display suggestions for increasing the number of busses 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 frequency of bus routes based on the events that are happening in that route on a map of Dublin city.

Actors

1. City Managers – When selecting the ‘Events/Incidents’ view of the site, City Managers will request for suggestions on bus route frequency visualization to be created.

Triggers and Inputs

Triggers:

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

Inputs:

1. User can select option to show the bus frequency suggestions

2. Flow of Events

|  |  |  |  |
| --- | --- | --- | --- |
| Basic Flow | | | |
| User | | System | |
| 1 | User selects the ‘Events’ dashboard view in the application. |  |  |
|  |  | 3 | The system retrieves the most-recent Events data from the local database. |
|  |  | 4 | Map of Dublin city is displayed, with the Events data overlaid |
| 5 | User selects option to show bus frequency suggestion |  |  |
|  |  | 6 | The system retrieves the most-recent Bus data from the local database. |
|  |  | 7 | Map of Dublin city is displayed, with bus route frequency suggestions |

3. Special Requirements

* 1. This data requires the existence of live Bus and Events data sources.

4. Preconditions

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

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

5. Postconditions

Once this use case has been completed, the Bus map visualization has been rendered with bus route frequency suggestions based on:

1. the most recent live or predicted data

*<single use case description end>*

### 

*<single use case description start>*

### Use Case Name: Maintenance events for 3 transport types

1. Description

*Goals:*

1. Use live data to display maintenance for Bikes, Luas and Buses.

*Responsibilities:*

This use case is responsible for showing maintenance details for each of the three modes of transport. These details include the location, vehicle information and time of maintenance.

Actors

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

Triggers and Inputs

Triggers:

1. User logs in to the application.
2. User selects the ‘Events’ dashboard to view the maintenance details of the three modes of transport.

Inputs:

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

2. Flow of Events

|  |  |  |  |
| --- | --- | --- | --- |
| Basic Flow | | | |
| User | | System | |
| 1 | User selects the ‘Events’ dashboard view in the application. |  |  |
|  |  | 2 | The system retrieves the most-recent Maintenance data from the local database for each of the discussed transport types. |
|  |  | 3 | A tabular display will occur for each transport’s maintenance detail. |
| 4 | User selects the mode of transport |  |  |
|  |  | 5 | Table is updated to show only the transport, user wants to see. |

3. Special Requirements

This data requires the existence of live Transport 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 details of Maintenance of all the specified modes has been rendered based on:

1. the most recent data and
2. the type of transport.

*<single use case description end>*

*<single use case description start>*

### Use Case Name: Warning notifications of extreme weather conditions

1. Description

*Goals:*

Display MET Eireann 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 in the affected areas.

Actors

1. 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:

1. User can view the weather warning notifications

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 Eireann weather warnings source. |
|  |  | 3 | If there are any extreme weather conditions happening in the near future, they will be displayed to the user. |

3. Special Requirements

1. This data requires the existence and availability of MET Eireann weather warnings data source.
2. It requires the existence of a weather warning happening in the near future.

4. Preconditions

The user must have logged in to the system and have sufficient privileges to view the weather warnings notification section.

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 Eireann warning data.

*<single use case description end>*