# CS7CS3 Advanced Software Engineering Group Project

# Requirements/Use Cases

# Project Name: *SUSTAINABLE CITY MANAGEMENT*

**Group: *5***

***<List of Group Members>***

# 1. Use Case Diagram

Please include a UML Use Case Diagram (see slides on Blackboard) for the project.

*Diagram here.*

*<From <single use case description start> to <single use case description end> contains the structure of the information that should be here for* ***each*** *use case. Copy and fill all sections for* ***EACH******USE CASE****>*

*<single use case description start>*

### Use Case Name: Luas Map Visualization

1. Description

*Describe the goals and responsibilities of the Use Case*

*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

*List the actors that are involved, and their roles in the Use Case*

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

*List and describe the triggers that start this use case executing, and the subsequent 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

*Here is where you indicate if the use case has any special requirements or expectations as to the existence of other systems*

* 1. This data requires the existence of live Luas’s data sources.

4. Preconditions

*Describe what must be have occurred previously for this use case to execute*

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

*Describe the state of the system, or what should be seen to have been achieved, when this use case has completed its processing.*

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’s electricity usage estimates.

1. Description

*Describe the goals and responsibilities of the Use Case*

*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

*List the actors that are involved, and their roles in the Use Case*

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

*List and describe the triggers that start this use case executing, and the subsequent 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

*Here is where you indicate if the use case has any special requirements or expectations as to the existence of other systems*

This data requires the existence of live Luas’s data sources.

4. Preconditions

*Describe what must be have occurred previously for this use case to execute*

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

*Describe the state of the system, or what should be seen to have been achieved, when this use case has completed its processing.*

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.

*<single use case description end>*

### Use Case Name: Dublin Bikes Station Visualization

1. Description

*Describe the goals and responsibilities of the Use Case*

*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

*List the actors that are involved, and their roles in the Use Case*

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

Triggers and Inputs

*List and describe the triggers that start this use case executing, and the subsequent 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

*Here is where you indicate if the use case has any special requirements or expectations as to the existence of other systems*

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

4. Preconditions

*Describe what must be have occurred previously for this use case to execute*

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

*Describe the state of the system, or what should be seen to have been achieved, when this use case has completed its processing.*

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.

*<single use case description end>*

### Use Case Name: Dublin Bikes Station Usage

1. Description

*Describe the goals and responsibilities of the Use Case*

*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

*List the actors that are involved, and their roles in the Use Case*

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

Triggers and Inputs

*List and describe the triggers that start this use case executing, and the subsequent 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

*Here is where you indicate if the use case has any special requirements or expectations as to the existence of other systems*

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

4. Preconditions

*Describe what must have occurred previously for this use case to execute*

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

*Describe the state of the system, or what should be seen to have been achieved, when this use case has completed its processing.*

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.

*<single use case description end>*

### Use Case Name: Login and firebase subscription.

1. Description

*Describe the goals and responsibilities of the Use Case*

*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

*List the actors that are involved, and their roles in the Use Case*

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

Triggers and Inputs

*List and describe the triggers that start this use case executing, and the subsequent inputs*

*Triggers:*

1. 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

*Using a bulleted list, describe the sequence of steps that should occur (basic flow all going well) in order to complete the use case, and what should happen if there are any conditions that mean the basic flow will not happen as described.*

*NOTE FILLED IN PURELY AS AN EXAMPLE:*

|  |  |  |  |
| --- | --- | --- | --- |
| Basic Flow | | | |
| User | | System | |
| 1 | User opens the application |  |  |
| 2 | User inputs username and password |  |  |
| 3 | User clicks on the button “Login” |  |  |
|  |  | 3 | The system validates credentials. |
|  |  | 4 | 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. |
|  |  | 5 | 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

*Here is where you indicate if the use case has any special requirements or expectations as to the existence of other systems*

User accounts should have been already created.

4. Preconditions

*Describe what must be have occurred previously for this use case to execute*

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”.

*<single use case description end>*