Analytics of Environmental Data

(Environmental Complaints)

CSP 586 – Software Modelling and Development with UML

Jagannath, Vinay (A20383110)

Beerappa Vanajakshi, Sashank (A20383115)

Chibe, Elliot (A20304324)

1. Overview

The project aims at providing details about environmental complaints received by the Department of Environment (DOE) from January 1993 to December 31, 2011 and by the Department of Public Health (CDPH) since January 1, 2012. The users can perform various analysis to explore and use statistical information derived from the dataset.

Complaints are regarding Air, water, and noise pollution or anything concerning the environment of particular region. In specific, the complaint types are categorized as: “Abandoned Site”, “Air Pollution Work Order”, “Asbestos Work Order”, “Construction and Demolition”, “Toxics Hazardous Materials Work Order”, ”Illegal Dumping Work Order”, “Noise Complaint”, “Permits Issued by DOE Work Order”, “Recycling Work Order”, “Service Stations/Storage Tanks Work Order”, “Vehicle Idling Work Order”, and “Water Pollution.”

In addition, the dataset also provides information regarding the complaint date, status, personnel handling the case, and location of the site as determined through the Chicago open data portal’s geocoding engine.

This serves as a great data source to generate insightful visualizations of the data determined over various factors and analyze them as per needs.

1. Feature List
2. Load data from dataset file
3. Display the selected data in tabular format
4. Visualize data as different types of charts – Bar chart, Pie chart, Polar area etc.
5. Filter datasets based on column (what needs to be included for data display)
6. Statistical metrics – total complaints, different types of complaints
7. Export data as csv spreadsheet.
8. Explore data by complaint type, over time
9. List of Actors
10. User
11. System
12. Use Cases



The use cases are defined as tasks executed by the actors to satisfy the requirements supported by the application. They can be defined as a list of actions of event steps typically defining the interactions between a user and a system to achieve a goal.

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions – use cases that some system of systems can perform in collaboration with one or more external users of the system – actors. Each use case should provide some observable and valuable result to the actors or other stake holders of the system.

The complete list of use cases is listed below.

1. User can select and loads the dataset
2. User can select the columns to be displayed – column filtering
3. User can apply filtering criteria on rows
4. User can sort rows based on column names and value
5. User can apply filtering criteria – “equal to” row value from drop down
6. User can get numerical metrics – Total number of complaints statistical data, Count of Different types of complaints statistical data
7. Users can select the chart type to be displayed
8. Users can export the data to csv.
9. Use Cases – Fully Dressed Format

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Selection and loading the dataset | |
| Triggering Event: | User confirming to select the dataset | |
| Brief Description: | The user selects the dataset to be loaded. Once the dataset gets loaded, the data is displayed in the tabular format. | |
| Primary Actors: | User | |
| Related Use Cases: | Column Filtering after data selection and loading | |
| Stakeholders: | User | |
| Preconditions: | User must have access to the Complaints dataset  Dataset should be in the correct format - csv | |
| Post conditions: | Dataset visible in tabular format | |
| Flow of events: | Actor | System |
| * The user selects the dataset | * Verify the dataset format * Loads the dataset * Displays the dataset in the tabular format * Shows column filtering option with checkbox for each available column |
| Exception Conditions | User loads a dataset which is not in the expected format. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Selecting the columns to be displayed – column filtering | |
| Triggering Event: | User confirming to selection of columns to be displayed | |
| Brief Description: | The user selects the dataset to be loaded. Once the dataset gets loaded, the data is displayed in the tabular format along with option for column filtering. User selects the columns to be displayed and applies the column filtering to display column specific data | |
| Primary Actors: | User | |
| Related Use Cases: | Selection and loading of dataset | |
| Stakeholders: | User | |
| Preconditions: | User must have access to the Complaints dataset  Dataset should be in the correct format – csv  Dataset should be displayed in the tabular format  Checkboxes with column names should be available for the dataset loaded | |
| Post conditions: | Data set visible with only selected columns | |
| Flow of events: | Actor | System |
| * The user selects the dataset * The user selects the columns to be displayed * Applies column filtering | * Verify the dataset format * Loads the dataset * Displays the dataset in the tabular format * Shows column filtering option with checkbox for each available column * Only selected columns displayed in the data table |
| Exception Conditions | User loads a dataset which is not in the expected format. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Selecting the rows to be displayed – row filtering based on column values | |
| Triggering Event: | User selecting the row filtering criteria | |
| Brief Description: | To be able to filter rows based on each column values. | |
| Primary Actors: | User | |
| Related Use Cases: | Column Filtering | |
| Stakeholders: | User | |
| Preconditions: | Dataset loaded and displayed in tabular format | |
| Post conditions: | Data set visible with only selected rows satisfying the row filter criteria | |
| Flow of events: | Actor | System |
| * The user selects the dataset * The user selects the column value for each column to be filtered * Applies row filtering | * Verify the dataset format * Loads the dataset * Displays the dataset in the tabular format * Shows row filtering option with dropdown for each available row value * Only rows satisfying the filter criteria are displayed. |
| Exception Conditions |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Selecting the rows to be displayed – sort rows based on column value | |
| Triggering Event: | User selecting the row filtering criteria | |
| Brief Description: | To be able to sort rows based on values for each column value. | |
| Primary Actors: | User | |
| Related Use Cases: | Column Filtering | |
| Stakeholders: | User | |
| Preconditions: | Dataset loaded and displayed in tabular format | |
| Post conditions: | Data set visible with only selected rows satisfying the row filter criteria | |
| Flow of events: | Actor | System |
| * The user selects the dataset * The user selects the column to sort the table on * Applies row sorting | * Verify the dataset format * Loads the dataset * Displays the dataset in the tabular format * Shows row filtering option with dropdown for each available row value * All rows are displayed in sorted order for the selected column. |
| Exception Conditions |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Selecting the rows to be displayed – search rows based on column value | |
| Triggering Event: | User selecting the row filtering criteria | |
| Brief Description: | To be able to search rows based on particular column value. | |
| Primary Actors: | User | |
| Related Use Cases: | Column Filtering | |
| Stakeholders: | User | |
| Preconditions: | Dataset loaded and displayed in tabular format | |
| Post conditions: | Data set visible with only selected rows satisfying the row filter criteria | |
| Flow of events: | Actor | System |
| * The user selects the dataset * The user searches for the rows with the particular column value. * Applies row filtering | * Verify the dataset format * Loads the dataset * Displays the dataset in the tabular format * Shows row filtering option with dropdown for each available row value * Only rows satisfying the filter criteria are displayed. |
| Exception Conditions |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Export view as csv | |
| Triggering Event: | User clicks export data option | |
| Brief Description: | To be able to export data in view as csv | |
| Primary Actors: | User | |
| Related Use Cases: |  | |
| Stakeholders: | User | |
| Preconditions: | Dataset loaded and displayed in tabular format | |
| Post conditions: | Data set visible with only selected rows satisfying the row filter criteria | |
| Flow of events: | Actor | System |
| * The user selects the dataset * The user applies the required filters on rows and columns and clicks on export data button to save the view as csv. * Pop up appears to user to download the data in view as csv file * Clicks on save button to save file to local system. | * Verify the dataset format * Loads the dataset * Displays the dataset in the tabular format * Shows row filtering option with dropdown for each available row value * Creates CSV file with all data in the current view. |
| Exception Conditions |  | |

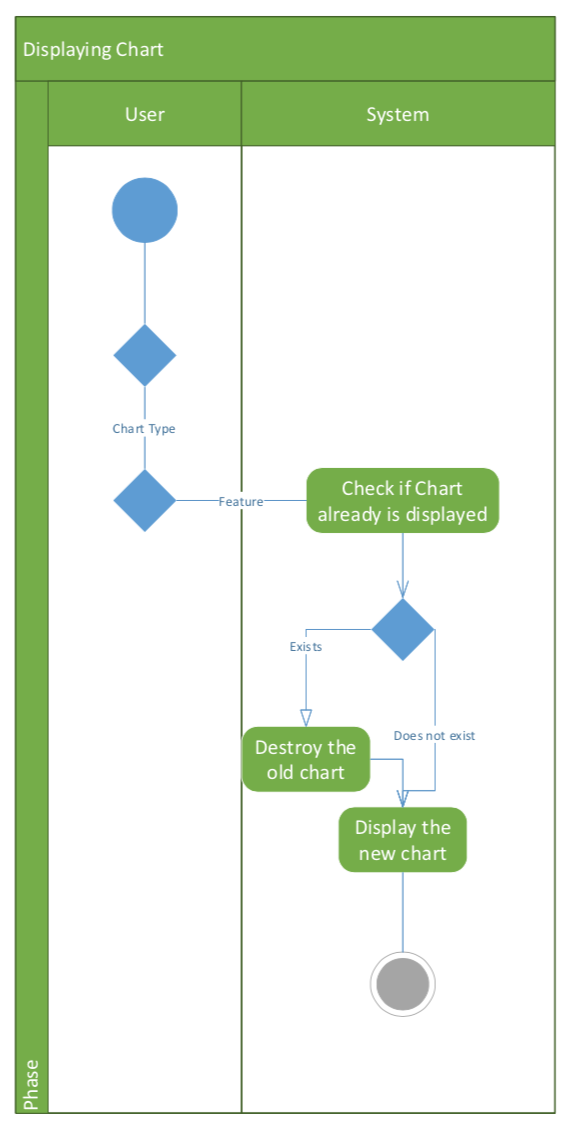
|  |  |  |
| --- | --- | --- |
| Use Case Name: | Selecting the chart to be displayed | |
| Triggering Event: | User selecting the chart type for visual representation of the data | |
| Brief Description: | The user selects the type of chart to be displayed. The user then chooses a feature or column to be used as the data for the chart. When the user click the ‘Display Chart’ button, their graph will appear below. | |
| Primary Actors: | User | |
| Related Use Cases: |  | |
| Stakeholders: | User | |
| Preconditions: | * The data set must be loaded * At least one feature should be populated in the table * There should exist at least 1 row in the table | |
| Post conditions: | Data set visualized using the selected chart type | |
| Flow of events: | Actor | System |
| * User selects chart type from the dropdown menu * User selects the specific feature from the dropdown menu * User clicks on the Display Chart button | * Checks and removes any previous charts * New chart is displayed |
| Exception Conditions | * The dataset is corrupt or in the wrong format thus failing to populate the feature list in the dropdown menu | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Calculate Statistics of the data | |
| Triggering Event: | User selecting the dataset for load | |
| Brief Description: | The user selects the dataset to be loaded. Once the dataset gets loaded, the data is displayed in the tabular format along with statistics of number of complaints, and count of different types of complaints. | |
| Primary Actors: | User | |
| Related Use Cases: | Row Filtering | |
| Stakeholders: | User | |
| Preconditions: | User must have access to the Complaints dataset  Dataset should be in the correct format - csv | |
| Post conditions: | Statistics visible in a section of the dashboard | |
| Flow of events: | Actor | System |
| * The user selects the dataset | * Verify the dataset format * Loads the dataset * Displays the dataset in the tabular format   Shows statistics with number of complaints |
| Exception Conditions |  | |

1. Activity Diagrams











1. System Sequence Diagrams











1. Sequence Diagrams









1. Analysis Model Class Diagram



1. Design Model Class Diagram



1. Design Patterns

Observer Pattern:

* Dropdown Menu for row filtering contains each values of the column it belongs to.
* This data should be dynamically updated based on the data displayed in data table.
* We use the DrawEventListener Interface of DataTable to receive the draw event update to update the menu content of the drop-down filter.
* This works according to the observer pattern (publisher subscriber) where DataTable would publish the changes to its table data, and dataset being the subscriber of the event, receives notification and refresh the menu content of the dropdown filter dynamically.



Façade Pattern:

* When the dataset class receives the table draw notification, it has to perform, activities on the drop-down Filter menu to remove the old items and create new menu with updated data items and associate handlers for each of the item click event.
* Clearly this means that it performs multiple operations on filter selection and filter menu.
* This interface is simplified using the façade pattern by implementing the intermediate class – DropDownFilter which handles all the background actions required.

