Software Design Document

New York Restaurant Inspection Results

Kartik Mathur(s5309927)

Manish Shrestha ( s5308120 )

Nivethaa Elangovan ( s5298899 )

Table of Contents

[1.0 System Vision 3](#_Toc46748622)

[1.1 Problem Background 3](#_Toc46748623)

[1.2 System Overview 3](#_Toc46748624)

[1.3 Potential Benefits 3](#_Toc46748625)

[2.0 Requirements 4](#_Toc46748626)

[2.1 User Requirements 4](#_Toc46748627)

[2.2 Software Requirements 4](#_Toc46748628)

[2.3 Use Cases 4](#_Toc46748629)

[3.0 System Components and Software Design 5](#_Toc46748630)

[3.1 System Components 5](#_Toc46748631)

[3.2 Software Design 5](#_Toc46748632)

[4.0 User Interface Design 6](#_Toc46748633)

# System Vision

## Problem Background

The data was collected by the Department of Health in NYC which collected the inspection report for permitted food establishments and graded them on a scale of A-F. Records for each restaurant collected contain the inspection date, type, action, address, cuisine description, violation code and description from January 1, 2010, to August 29, 2017. (NYC Restaurant Inspections, 2023) The dataset provides a vast range of information which when viewed cannot give an idea or provide the user with any knowledge. The dataset needs to be visualized in order to get interesting insights and finding it is very hard without data visualization. So for the Department of Health, it is very easy to get a summary of the data if their dataset is visualized in such a way that it provides a dashboard summary and gives them a data report. Another problem is that the user cannot search by any type or view all of the results after searching. By exhibiting all listings' information, offering a visitation chart, and fetching all records that contain a specific keyword that may be tied to the user input, the built application will enable the delivery of data reports at the selected start and end date.

## System Overview

The new system would have the following features:

* The user will be able to enter the start and end date.
* The system displays all the details for inspection reports between the dates specified.
* The inspection details between the dates and keyword are displayed to the user in a tabular format.
* It creates a dashboard of the charts displaying the inspection reports.
* It gives a pictorial summary of the violation distribution of suburbs.
* It gives a pictorial summary of Violation counts per cuisine
* It gives a pictorial summary of Violation related to animals
* It gives the option to export the dataset into Excel or a spreadsheet.

## Potential Benefits

The new system gives the user the following advantages:

* The information provided by the system helps the Department of Health NYC to analyse the closing ratio of the restaurants.
* The information provides what violations are prominent in each suburb.
* Gives an overview of competitors of chain restaurants vs. independent restaurants.
* Gives people an idea of the health safety of having different cuisines.
* Helps the restaurants to improve their quality by analysing what violations exist and avoiding them.
* Can help the health department do routine inspection to ensure food quality in the restaurants that have already recorded violations.

# Requirements

## User Requirements

The user of the system will be able to get the required information on violations by restaurants through the system. To handle the large dataset of information, the system will have an input interface where users enter keywords and date range which displays the results with visualization and summary in a user-friendly dashboard. The users of the system are mainly the Department of Health and can be food critics or food enthusiasts.

To enable this system the following features need to be created:

* The Home page with necessary fields, columns and attributes.
* The Dashboard page with necessary fields, columns and attributes.
* The system should contain a date picker field for the Start date and end date
* The system should have a text field to enter a keyword.
* The program processes the query to display the results based on the query
* The dashboard page displays the summary results and charts.
* The data and results are exported as a spreadsheet or Excel.

There a two main pages which the user can access:

Dashboard page:

* The user logs into the system to open a dashboard tab which navigates the user to a query page. Textbox, date picker field and button are used in the dashboard where the user of the system filters the dataset according to their needs. The system processes the query and displays the results that are visualized in the form of charts such as bar graphs, histograms and line graphs.

Home page:

* The home page contains a search textbox and date picker field which displays data in a tabular format in reverse chronological order.

## Software Requirements

R1. The system shall have two pages: the home page and the dashboard page.

R2. The system shall display all the data in a tabular format on the home page.

R3. The system shall display the necessary data filters on the dashboard and home page.

R4. The system shall allow the user to select a date range from the date picker in the filter section.

R5. The system shall allow the user to search for violations based on keywords entered in the search field in the filter section.

R6. The system shall allow the user to view violations over different suburbs as a plot/chart.

R7. The system shall allow the user to view violations related to animals over a period as a line chart.

R8. The system shall allow the user to view violations related to animals over different suburbs as a bar chart or a histogram.

R9. The system shall allow the user to view the violation count based on cuisines as a clustered column chart.

R10. The system shall allow the user to export the data as an Excel or spreadsheet.

## Use Cases & Use Case Diagrams

|  |  |
| --- | --- |
| Use Case ID | 1 |
| Use Case Name | Load Data |
| Actor | Admin |
| Description | The data is loaded for the user to view. |
| Pre-condition | None |
| Post-condition | All data is valid and in the right format. |
| Success scenario | The data is loaded for any operations. |

|  |  |
| --- | --- |
| Use Case ID | 2 |
| Use Case Name | View Data |
| Primary Actor | Client |
| Secondary Actor | Admin |
| Description | The data is displayed in tabular format for the user. |
| Pre-condition | None |
| Post-condition | All data is valid and in the right format. |
| Success scenario | The data is loaded and displayed in tabular format. |

|  |  |
| --- | --- |
| Use Case ID | 3 |
| Use Case Name | Filter Data |
| Primary Actor | Client |
| Secondary Actor | Admin |
| Description | The data is filtered based on date and keyword entered by client |
| Pre-condition | There is some data displayed. |
| Post-condition | The date range selected is a valid value and data exists for the filter selected. |
| Success scenario | The data is filtered based on date and search parameters and displayed. |

|  |  |
| --- | --- |
| Use Case ID | 4 |
| Use Case Name | Visualise Data |
| Primary Actor | Client |
| Secondary Actor | Admin |
| Description | The data is visualised in charts. |
| Pre-condition | The data entered contains all the necessary fields. |
| Post-condition | None |
| Success scenario | All the charts have valid values to be displayed. |

|  |  |
| --- | --- |
| Use Case ID | 5 |
| Use Case Name | Export Data |
| Actor | Client |
| Description | The data is exported as excel. |
| Pre-condition | The data has valid rows and columns to be exported. |
| Post-condition | None |
| Success scenario | The data is exported in the excel format. |

A diagram of a data flow

Description automatically generated

# Software Design and System Components

## Software Design

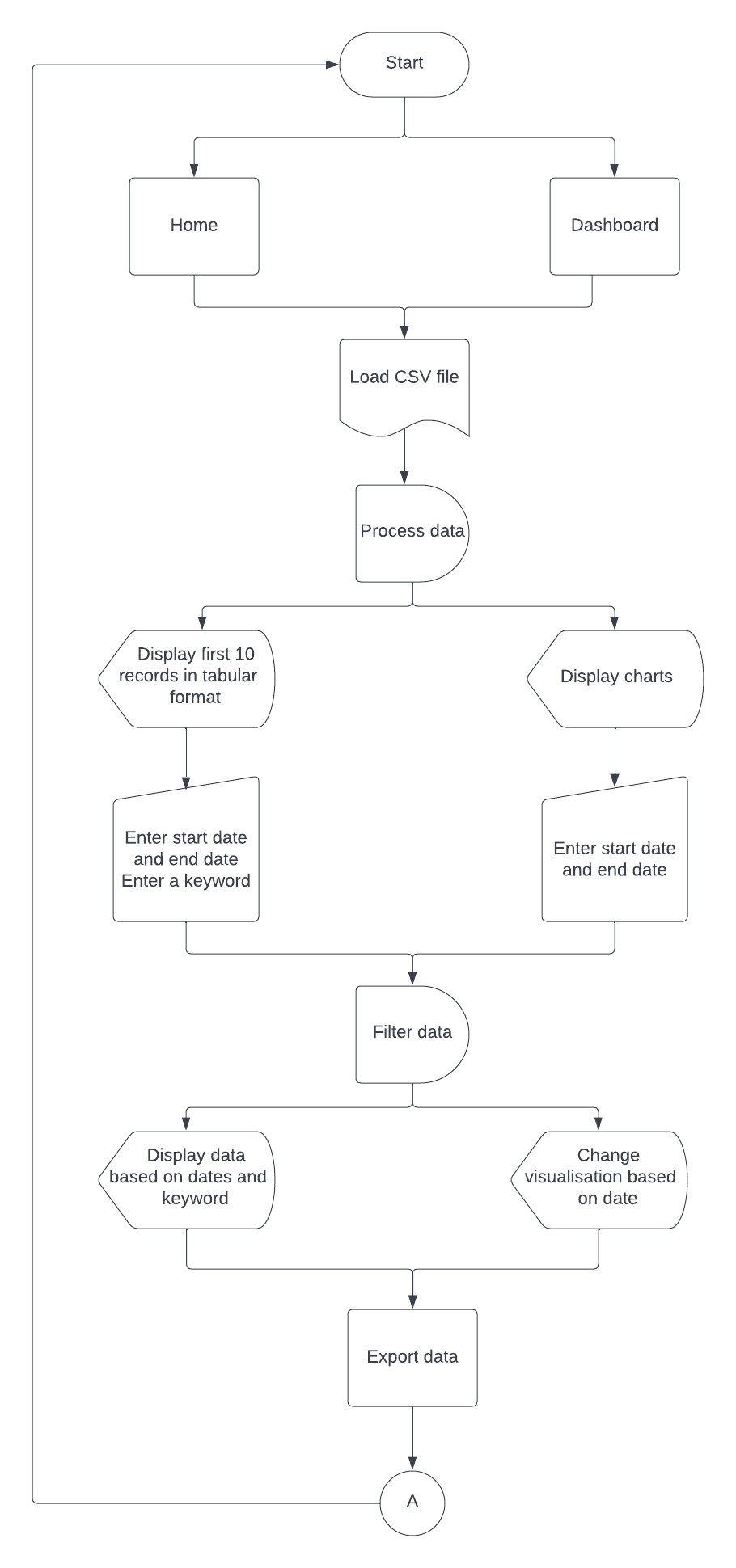


Figure 3a Flowchart

Figure 3a represents the workflow of the software.

## System Components

### Functions

loadData()

* This function is used to load the csv file that contains all data for processing

GridTableBase()

* This is a class used to construct the table which will display records

GetNumberRows()

* This function is used inside the GridTableBase() class
* It is used to set the number of rows for the table
* Return the number of rows for the table

GetNumberCols()

* This function is used inside the GridTableBase() class
* It is used to set the number of columns for the table
* Return the number of columns for the table

SetValue(row, col, value)

Parameters:

* row(int): Indicate the row number
* col(int): Indicate the column number
* value(string): text string that has to be displayed
* This function is used inside the GridTableBase() class
* It is used to set the value of cells as text inside the table
* Does not have a return value

DeleteRows(pos, numrows)

Parameters:

* + pos(int): The position of the first row to be deleted
  + numrows(int): The number of rows to be deleted
* This function is used inside the GridTableBase() class
* It is used to delete rows inside the table
* Return type of this function is boolean

InsertRows(pos, numrows)

Parameters:

* + pos(int): The position of first row to be deleted
  + numrows(int): The number of rows to be deleted
* This function is used inside the GridTableBase() class
* It is used to inset rows inside the table
* Return type of this function is boolean

DatePickerCtrl()

* This class is used for the date selectors in the home and dashboard page

Create(parent, id, dt, pos, size, style, validator, name)

Parameters

* + parent- The parent of the control window
  + id – The id of the control window
  + pos(int) – the initial position
  + size(int) – the initial size of the control
  + style(string) – the style of the control window
  + validator – any additional validator checks for the date
  + name(string) – name of the control
* This function is inside the DatePikcerCtrl() class
* It is used to create a control window for selecting a date
* Return type will be a boolean
* It is True if the control was successfully created

SetRange(date1, date2)

Parameters

* + date1, date2(datetime object) – The earliest possible date that can be valid
* This function is used inside the DatePickerCtrl() class
* It is used to set a valid date range for selecting a date in the datepicker

GetRange()

* This function is used inside the DatePikcerCtrl() class
* It is used to get a valid date range if it was set previously using the SetRange() function
* Returns a tuple of the date range

GetValue()

-This function is used inside the DatePickerCtrl() class

- It is used to get the current date entered in the date field

- Returns a python datetime object

SearchCtrl()

* This class is used to implement the search operation in the home page

GetValue()

* This function is used inside the SearchCtrl() class
* It is used to get the valut of string inside the seach field
* Returns a string

Clear()

* This function is used to clear the text in control

Plot()

* This class is used to plot the graphs

plotWhisker(data)

Parameters

* + data(tuple) – values to be plotted
* This function is used inside the Plot() class
* It will be used to plot the whisker and box plot

plotColumnChart(data)

Parameters

* + data(tuple) – values to be plotted
* This function is used inside the Plot() class
* It will be used to plot the clustered column chart

plotLine(data)

Parameters

* + data(tuple) – values to be plotted
* This function is used inside the Plot() class
* It will be used to plot the line chart

plotBar(data)

Parameters

* + data(tuple) – values to be plotted
* This function is used inside the Plot() class
* It will be used to plot the bar chart

exportData()

* This function is used to export the data to a xlsx format

filterData()

* This function is used to filter the data based on date and search keyword
* Returns an array of values

(*wx.grid.GridTableBase — wxPython Phoenix 4.2.1 Documentation*, n.d.)

### Data Structures / Data Sources

Tuple

* Tuple is used to obtain daterange
* Used to pass values to plotting functions

Function:

* GetRange()

Array

* Array data type is used at places where data is filtered from the csv file and is returned to the another function for display

Function:

* filterData()

### Detailed Design

Pseudocode for Tuple:

FUNCTION TUPLELOOP(tuple):

FOR value in TUPLE FROM INDEX -> 0 TO LENGTH(tuple):

IF CONDITION:

Append or assign to new variable

ENDIF

END LOOP

END FUNCTION

RETURN value

Pseudocode for Array:

FUNCTION ARRAYLOOP(array):

FOR value in ARRAY FROM INDEX -> 0 TO LENGTH(array):

IF CONDITION:

Append or assign to new variable

ENDIF

END LOOP

END FUNCTION

# User Interface Design

In this section, we will discuss in detail the User Interface Design of the proposed software which is used to visualize the New York Restaurant Inspection Results dataset. The actual design is created using Canva, a graphic design tooL and Microsoft Visio is used to design a low-fidelity wireframe and structure of the product. This section comprises two sub-sections: 1) Structural Design, which will elaborate on the workflow of the product and a detailed analysis of the design and 2) Visual Design, which focuses only on the visual elements of the product.

## Structural Design

The product is designed in such a way that it is self-explanatory and is user-friendly. It consists of two main pages: 1) Home and 2) Dashboard. We will discuss in detail about each page in this section.

**Basic Structure**

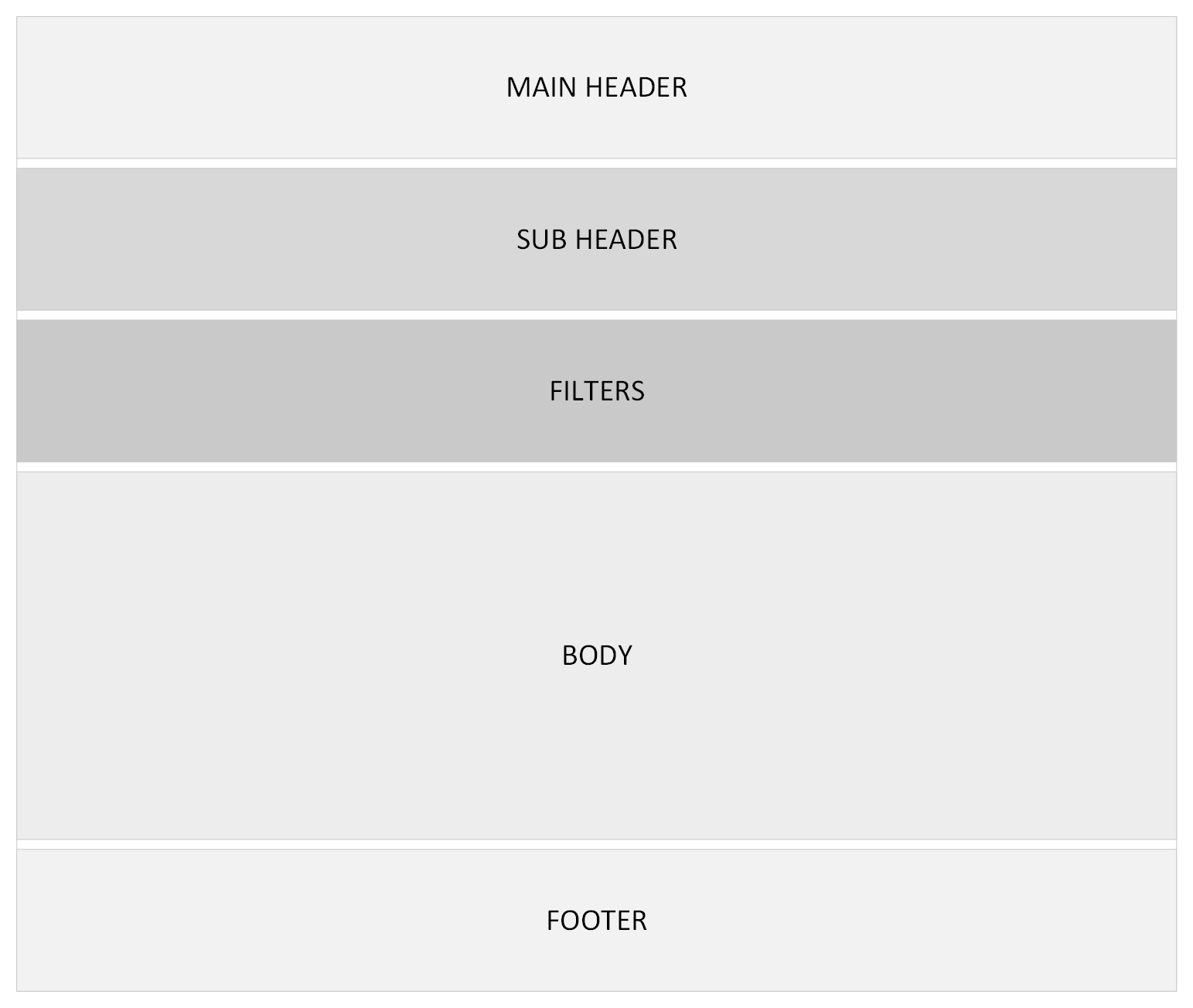


Figure 4a Basic structure of User Interface

Each page is divided into five sections as shown in figure 4a which are explained as follows:

* + Main Header: This section of the page contains the name of the product, the loggedin username and a dropdown-menu for the user ( The dropdown-menu will not be developed in this project as there is no requirement for a user login)
  + Sub header: This part contains the navigation tabs to toggle between pages. It displays the title of the dataset used.
  + Filters: This section contains any filtering options used to filter the dataset as per the user requirements. In addition, it also contains the export option to export the filtered dataset.
  + Body: This section contains the dataset displayed in the form of table or charts.
  + Footer: It contains copyright information and any other additional contact information.

**Detailed Structure**

**Home Page**

****

Figure 4b Structure of Home Page

Figure 4b shows the detailed structure of the main or home page of the product that the user will see initially. The first section will contain the logo and/or the product name on the top left corner and the user icon will be displayed on the right for the loggedin user. Below that is the sub header section, which will comprise of two tabs: Home and Dashboard to toggle between the two pages. On the right side of this section, the title of the dataset will be displayed.

The thrid section will feature any filters that are applied on the dataset. On the left side, there are two date fields which will enable the user to select a start date and an end date from a datepicker, based on which the dataset will be filtered. A view button is provided next to the date fields to initiate the search action. On the right side of the same section, an input box is provided. The dataset will be filtered and displayed based on keyword entered in this section. It also contains an export button which can be used to download the filtered dataset.

The fourth section contains the actual data that is displayed as a table. The table comprises a header ( column headings ), body ( values ) and footer ( pagination ). The table will display only important information related to any search to make it presentable and avoid data congestion that may occur with long text data. Since an export option is provided the user will be able to view all columns by exporting the data. The fifth section will display copyright information and any other contact information.

**Dashboard Page**

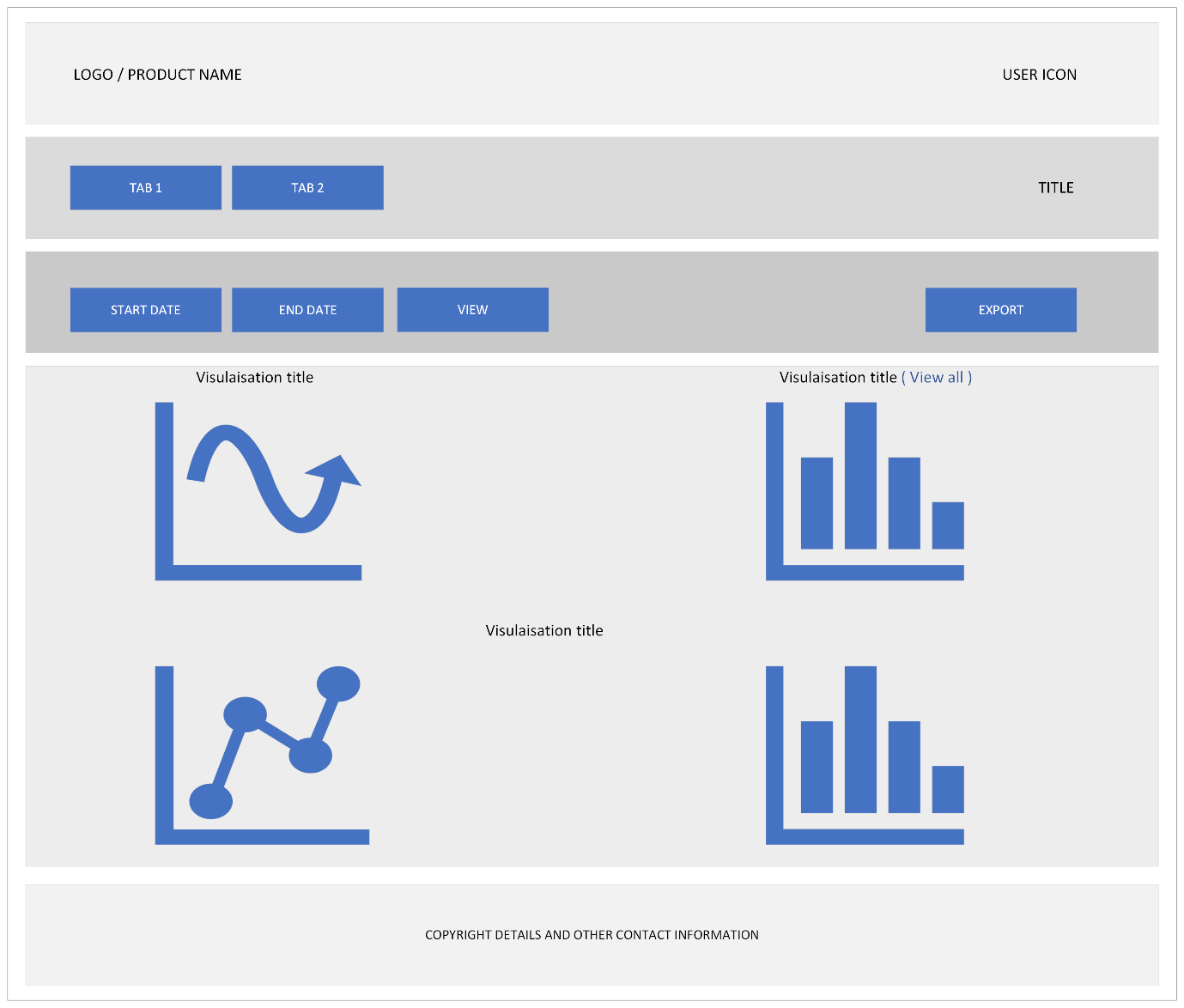


Figure 4c Structure of Dashboard Page

Figure 4c shows the dashboard page which will be displayed when the user selects tab 2. This page will feature all visualizations for the dataset in the body section of the page. The same header, subheader, filter and footer sections are retained. But this page does not have the keyword search option since we have 4 different visualizations and the parameters for each one differs.

* Visualisation 1: Violation Distribution over Suburbs

This plot will visualise the violations across different suburbs. The suburbs will be listed along the x-axis with the numerical values listed along the y-axis. This will be a box and whisker plot.

* Visualisation 2: Violation count per cuisine ( Additional visualization / Inisight )

This plot will visualise the violations across different suburbs based on the cuisine. The data will be plotted as a clustered column chart. The violation will be plotted along the bottom x-axis and the cuisines will be clustered along the upper x-axis. The y-axis will denote numerical values or count of violations. There are numerous violation codes so the initial visualization will feature only a few. The “View all” link next to the visualization title will redirect to a new page where all the violation code will be plotted.

* Visualisation 3: Violation related to animals

There are two visualisations in this part. Firstly, the violation cases related to animals and their trend over time is plotted as a line chart. Each line will indicate an animal. The x-axis will denote time period ( months/years ) and the y-axis will denote numerical values to represent count of cases over time. Secondly, the violation cases related to animals over different suburbs will be displayed as a histogram with suburbs listed along the x-axis and the numerical values denoted along the y-axis.

**Conceptual Model**

Figure 4d represents the workflow of the user interface.

A diagram of a process

Description automatically generated with medium confidence

Figure 4d Workflow Diagram

## Visual Design

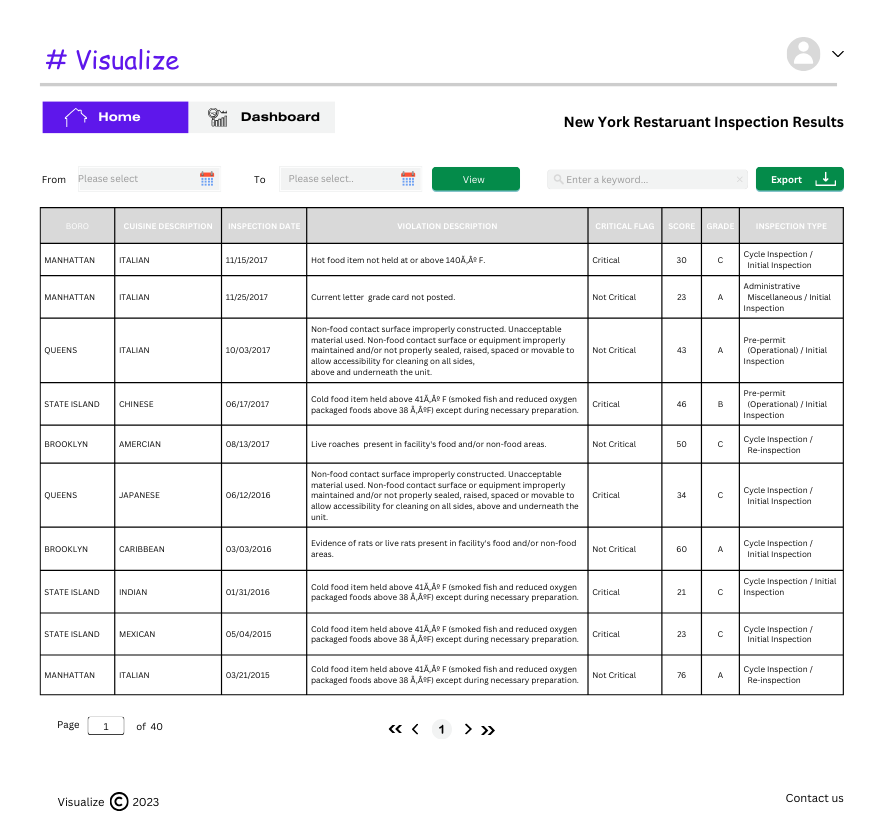


Figure 4e UI Design of Home Page

Figure 4e shows the design of home page. It replicates the structural design as described in figure 4b. It shows 10 records by default arranged in a tabular form with only the most important columns sorted in reverse chronological order.

A screenshot of a computer

Description automatically generated

Figure 4f UI Design of Dashboard Page

Figure 4f follows the basic strutural design of the product. This page will shows 4 insights for the New York Restaurant Inspection Results. The color codes, fonts and icons used for the home page and dashboard page are listed below.

**Color:**

|  |  |  |
| --- | --- | --- |
| **Color Type** | **Color Code** | **Example visual representation** |
| Primary color | #5E17EB |  |
| Secondary color | #F2F3F3 |  |
| Black | #FFFFFF |  |
| White | #000000 |  |

Table 4g Product colors

Table 4g represents the colors used in the product. The product has one primary and one secondary color. In addition, one shade of black and white is used through the entire product. The colors are kept minimal to enhance the user experience.

**Fonts:**

|  |  |
| --- | --- |
| **Font Type** | **Font Name** |
| Primary font | Canva Sans |
| Secondary font | Comic Sans |
| Product font | Helios Extended |

Table 4h Product Font

Table 4h represents fonts used in the product. The product uses one primary and one secondary font. Variations of the font like bold and regular are used at relevant places.

**Icons**

|  |  |
| --- | --- |
| **Icon** | **Description** |
| A person in a circle  Description automatically generated | This is the user icon. This can also be replaced with the user’s profile picture(which is not implemented in this case) |
| A black outline of a house  Description automatically generated | This icon is used for the home page navigation tab. It is used in white color for an active tab and in black color for an inactive tab |
| A graphic of a graph and a magnifying glass  Description automatically generated | This icon is used for the dashabord page navigation tab. It is used in white color for an active tab and in black color for an inactive tab |
|  | This icon is used for the datefield |
| A black arrow pointing down  Description automatically generated | This icon is used inside the export button |
|  | This icon will be used to display a dropdown menu next to the user icon(which is not implemented in this case) |
| A black arrows on a white background  Description automatically generated | These icons are used to navigate to the previous page and next page in the table |
| A black arrows pointing to the left  Description automatically generated | These icons are used to navigate to the first and last page of the table |
|  | This icon is used for copyrighting |

Table 4i Product Icons

Table 4i represents the icons used in the product. It displays the icons with a usage description of each icon.

**Element Description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | | **Description** | **Color** | **Style** |
| Product color | | General color of the product / Brand color | #5E17EB |  |
| Background color | | Overall background color of the software | #FFFFFF |  |
| Title | | Color used for the logo or title of the software | #5E17EB |  |
| Font color | | Color used for all the text/fonts in the software | #000000 |  |
| Button color | | Color used for buttons | #048B4A |  |
| Navigation Tab | Font Style | Font style used for navigation tab | Helios Extended Bold | **Sample** |
| Active Navigation Tab | Font Color | Color of font used for active navigation tab | #FFFFFF |  |
| Background Color | Background color used for active navigation tab | #5E17EB |  |
| Inactive Navigation Tab | Font Color | Font color used for inactive navigation tab | #000000 |  |
| Background Color | Background color used for inactive navigation tab | #F2F3F3 |  |
| Date field | Background color | Color used for the background of date field | #F2F3F3 |  |
| Font Color | Color used for the font of date field | #000000 |  |
| Font Style | Color used for the font of the date field | Canva Sans Regular | Sample |
| Input field | Background color | Color used for the background of the input or search field | #D9D9D9 |  |
| Font Color | Color used for the font of the date field | #000000 |  |
| Font Style | Font style used for the input field | Canva Sans Regular | Sample |
| Table | Table Header Color | Color used for the background of table header | #D9D9D9 |  |
| Header Font Style | Font style used for table | Canva Sans Bold | **SAMPLE** |
| Table Body Color | Color used for the background of the table body | #FFFFFF |  |
| Body Font Style | Font style used for table body | Canva Sans Regular | SAMPLE |

Table 4j Product Elements and Description

Table 4j represents colours, fonts and style of each element in the home page and the dashboard page.

# Reference:

NYC restaurant inspections. (2023). Kaggle. Retrieved from:

<https://www.kaggle.com/datasets/new-york-city/nyc-inspections>

wx.grid.GridTableBase — wxPython Phoenix 4.2.1 documentation. (n.d.). Retrieved from: <https://docs.wxpython.org/wx.grid.GridTableBase.html#wx.grid.GridTableBase.SetView>