Software Design Document

New York Restaurant Analysis

Pranav Patel: S2961443

Table of Contents

[Table of Contents 1](#_Toc446545257)

[System Vision 3](#_Toc681391171)

[1.1 Problem Background 4](#_Toc1894901642)

[1.2 System Overview 4](#_Toc2078550454)

[1.3 Potential Benefits 4](#_Toc163830365)

[Requirements 5](#_Toc2268208)

[1.4 User Requirements 6](#_Toc1096410030)

[1.5 Software Requirements 6](#_Toc675372061)

[1.6 Use Cases & Use Case Diagrams 6](#_Toc472161525)

[3.0 Software Design and System Components 11](#_Toc1166474536)

[3.1 Software Design 12](#_Toc1762399836)

[3.2 System Components 13](#_Toc1318983062)

[3.2.1Functions 13](#_Toc165527473)

[3.2.2 Data Structures / Data Sources 16](#_Toc1264845340)

[2.0 Detailed Design 17](#_Toc1953166448)

[User Interface Design 21](#_Toc700196127)

[2.1 Structural Design 22](#_Toc1965437238)

[4.1.1 navigational design 22](#_Toc287715328)

[4.1.2 structural design implementations 22](#_Toc551064840)

[2.2 Visual Design 23](#_Toc1256720520)

[4.2.1 Design choices 23](#_Toc316107721)

[4.2.2 Layout and wireframes 24](#_Toc1258150731)

# System Vision

## Problem Background

In order to provide business owners in the New York area with a better understanding of the health rating of their establishment and any violations it may have warranted, a software analyst program is being created to help address and alleviate the hassle of searching for and understanding the data presented within the New York Health Inspections dataset.

The program needs to ensure that it efficiently enables business owners and clients to see and compare the data surrounding the New York Restaurants sectors' food and health inspections in a thorough and business useful manner.

It needs to have to ability to search, contrast and compare the data from different establishments. Different was to view and compare data, and an instruction guide on how to use the software. All within the confines of one easy to use and accessible program.

## System Overview

Local businesses are constantly looking for information that will help them understand how they stack up against their rivals and what areas of their company need to be improved in order to ensure their business is always generating value.

The program should contain all the data regarding the grade of the business, the types of violations that they may inherit, and the overall score given to the business by the inspection analyst should all be included and able to be obtained and compared in an easy and knowledge enriching manner.

The program should also be designed so that even ordinary customers can utilize the program to make a more informed choice about where they should dine. This will help them identify the types of businesses that might have health and safety issues, keeping them safe from potential food related harm.

## Potential Benefits

The overall goal of the final program is to enable business owners and the general public, a means to view the dataset of the New York Restaurant Inspection results in one simple to use and understandable program. Businesses can use this application to compare themselves to other regional or national competitors, allowing for the identification of areas that could use improvement; Driving up value and improving the health safety of their business.

Whilst this program’s main purpose is to be used as that of a tool for analysis and comparison, we hope the program we develop also assists business owners helping creating value generation and raising awareness of the importance of workplace health, safety, and cleanliness. From a business standpoint, this enhances both customer confidence and sales. Whilst with customers, a means to better help people decide where to eat and which establishments to avoid based on the safety guidelines provided by certified professionals.

# Requirements

## User Requirements

The main overall goal of the program is to provide simplicity and effectiveness to the user when they want to search for specific data contained within the New York Restaurant dataset. On start up, we assume that the end user is someone that may not be familiar with technology, so we introduce the program with an introduction page, to ensure that the user understands what the program, what it is used for and how to go about using it. A Start button should reside below to affirm that the user understands how the software works before starting.

From the main page, the user is given the ability to input a start and end date to which they can query the program to output data within that time span. Along with timespan, the user can also input a violation keyword which will display the all the data associated with recorded violations in the allotted timespan within the dataset.

The program needs buttons to help further present aspects of the overall data as well as specified user inputted queries on the data. To do this, buttons such as a graph making button will help further visualise the content displayed in the main array of displayed data. (An example would be data such as violation data will have graphs to explain the frequency of specific types of violations.)

The user should also be able to easily clear data and restart their search/make a new query with the program, so a “Clear” button should simple and defined, so that the user understands how to clear the info on screen just from seeing the button. As well as multiple other buttons that give easy to digest and read overviews of different data collected using the program, such as Animal Violations and the Executive Report.

## Software Requirements

The following is a list of the requirements that will be included in the final build of the program:

* Ability to display data of user selected periods.
* Ability for program to produce graphics such as pie charts and histograms that can be utilized to further understand each section of the dataset.
* Refresh button that clears all current user selected/searched data within the same program window.
* Instruction screen/Start up prompt that details to users how to use the program.
* Graphics and design/layout of the program to be consistent throughout, and to be eye-catching but not overbearing for the user to look at.
* Executive report button that displays a collection of data and diagrams found and created by the software itself.
* Button that when pressed with the right conditions will display a visual diagram for the data displayed for the data corresponding to the search bar.
* Button that provides collected analysis data on animals/pests

## Use Cases & Use Case Diagrams

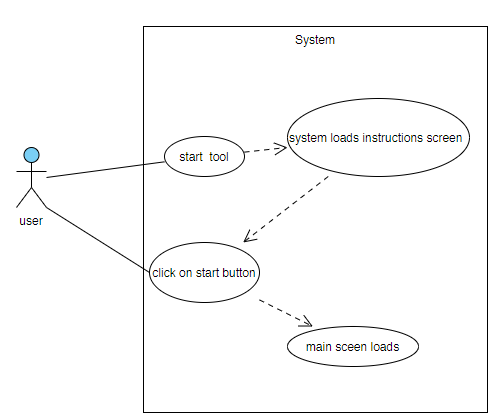
Below are 5 use cases and use case diagram of the tool.

In this section you provide some use cases showing how people may use your software.

1. Main screen of tool

|  |  |  |
| --- | --- | --- |
| Use case name | Open main screen of tool | |
| Scenario | The user wants to use the analysis tool | |
| Triggering event | Starting the program | |
| Brief Descriptions | The user must be in the main screen to use the analysis tool. Once the program starts the user must go through the instructions screen to reach the main screen. | |
| Actors | User (New York Health inspector), Analysis tool | |
| Related use cases | Open main screen of tool | |
| Stakeholders | New York Inspection department, customers | |
| Preconditions | The program should be started | |
| Postconditions | none | |
| Flow of activities | User | System |
| 1. User opens the instruction screen by starting the tool. 2. User clicks on the start button on the instructions screen . | 1. The system will load the instructions screen on start of the software. 2. When start button is clicked the main screen will load. |
| Exception condition |  | |

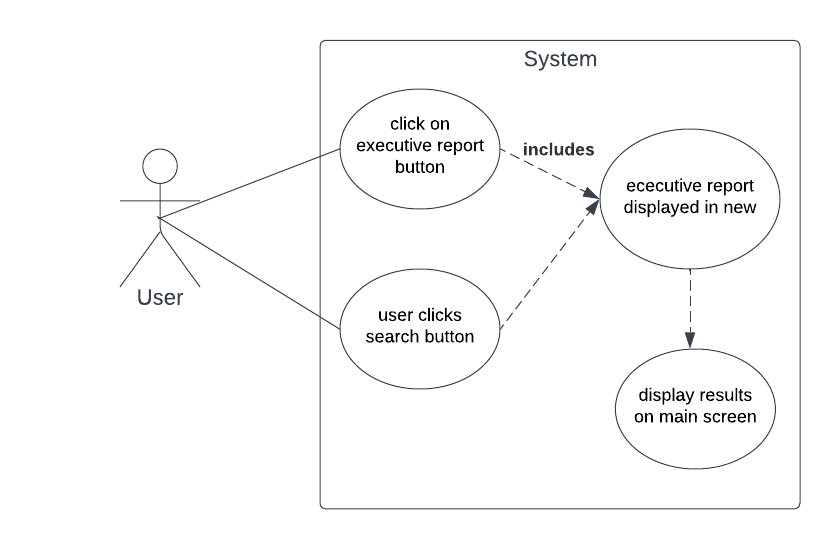
Below is the use case diagram of the use case above.



1. Restaurant inspection details

|  |  |  |
| --- | --- | --- |
| Use case name | Retrieve Restaurant inspection details of specific date range | |
| Scenario | The user wants to retrieve details of inspections in a specific date range. | |
| Triggering event | the user enters end date and start date and click on search | |
| Brief Descriptions | The tool will search through the data files for inspections on from the user specified start date to the end date and will display the results in the main screen | |
| Actors | User (New York Health inspector), Analysis tool | |
| Related use cases | Open main screen of tool | |
| Stakeholders | New York Inspection department, customers | |
| Preconditions | The start date and end date must be entered. | |
| Postconditions | none | |
| Flow of activities | User | System |
| 1. The user inputs start date and end date 2. The user clicks the search button | 1. When search button is clicked, system takes input of start date and end date. 2. The system collects data of all inspections from start date to end date from data files. 3. System displays all results on main screen in a specific order. |
| Exception condition | No keyword entered | |

Below is the use case diagram of the use case above.



heloo

1. Executive report

|  |  |  |
| --- | --- | --- |
| Use case name | Get executive report | |
| Scenario | The user needs to see the executive report made of the given all the data | |
| Triggering event | The user clicks on Executive report button | |
| Brief Descriptions | When the user clicks on the executive report button a screen with the executive report made by the team of the given data will be shown. | |
| Actors | User (New York Health inspector), Analysis tool | |
|  |  | |
| Related use cases | Open main screen of tool | |
| Stakeholders | New York Inspection department, customers | |
| Preconditions | User is on main screen | |
| Postconditions | Executive report screen is open | |
| Flow of activities | User | System |
| 1. The user must click on the executive report button | 1. The system will open a new screen with the executive report image once the button is clicked |
| Exception condition |  | |

4.Cleanest restaurant list

|  |  |  |
| --- | --- | --- |
| Use case name | Cleanest restaurant | |
| Scenario | User wants the list of cleanest restaurants so clicks on cleanest restaurant button. | |
| Triggering event | Clicking on cleanest restaurant button | |
| Brief Descriptions | The user wants to see the cleanest restaurant list so they can click on the cleanest restaurant button which will open the cleanest restaurant screen. | |
| Actors | User (New York Health inspector), Analysis tool | |
| Related use cases | Open main screen of tool | |
| Stakeholders | New York Inspection department, customers | |
| Preconditions | Main screen should be active | |
| Postconditions | Cleanest restaurant screen should be open | |
| Flow of activities | User | System |
| 1. User clicks on cleanest restaurant button | 1. System retrieves data of cleanest restaurant based on violations 2. System shows data in cleanest restaurant screen |
| Exception condition |  | |

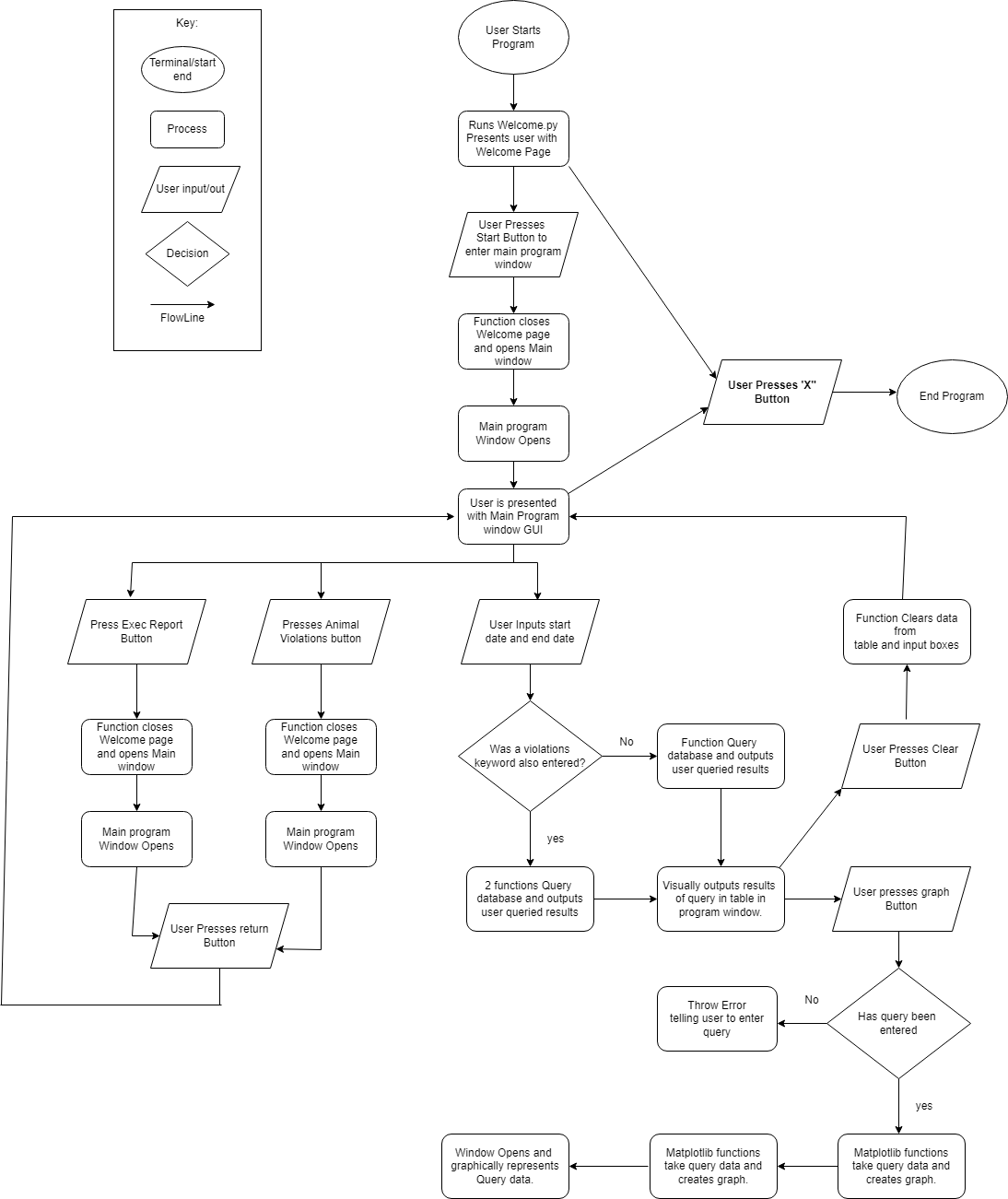
1. animal violations data use case

|  |  |  |
| --- | --- | --- |
| Use case name | Animal violations data | |
| Scenario | User wants to see animal violations data | |
| Triggering event | Click on animal violations button | |
| Brief Descriptions | The user clicks on the animal violations button to see the animal violations data. The tool retrieves data regarding animal violations and show them in a new screen | |
| Actors | User (New York Health inspector), Analysis tool | |
| Related use cases | Open main screen of tool | |
| Stakeholders | New York Inspection department, customers | |
| Preconditions | The main screen of the tool should be open | |
| Postconditions | none | |
| Flow of activities | User | System |
| 1. The user clicks on the animal violations button | 1. The system retrieved information regarding animal violations. 2. System creates graph with retrieved data 3. The system displays graph and data in new window called animal violations |
| Exception condition |  | |

# 3.0 Software Design and System Components

## 3.1 Software Design

The following diagram is a flowchart of how the program operates during an average user experience:



## 3.2 System Components

### 3.2.1 Functions

Def Welcome\_Page():

First function that runs when the user starts the program. Opens a window that displays a screen with an introduction and a set of usage instructions on how to operate the program.

Has Bold text with the with name of the software as well as text below that describes to the user how to operate the software successfully.

Contains a page transition function that as described below that is used with the start button.

A GUI start button that when interacted with, sends the user to the main window of the program.

Def “Page\_Transitioner”():

Called when User presses on a button that intends to send the user to a new window of the program

Small function that grabs imports function from file Main\_Window.py

Destroys the running of the current function which causes the main window to close.

Runs imported function which causes the main window to open/run.

Def Main\_Window():

When called, runs main window of program. Contains all visual and user input elements of the main file including text input boxes, drop down menus, Buttons and data array. All the visuals for this function will be created using Python

Buttons and their places of the following Items. 3 Input boxes to insert Start Date, End Date and Violations Keyword. Uses search function to query database. Search Button to the right. GUI e.g size colour, shape defined in Tkinter. Contains Tkinter function Command = Search. Which initilzes search function.

Throws Error if user input data is Incorrectly entered.

Buttons below Input bars include Animal Violations button and Executive report button which sends the user to a new window when pressured using page transition function and imported destination function. (Referenced Below)

Table that sits at bottom of window and displays a table of all the New York Restaurant data. Table is finalized using Create\_Table Function and is updated using the search button.

Graph Button is large than most buttons and when pressed with the right conditions, starts a function which grabs the array of violations data and queries the distribution plot of violations over the different suburbs. This matplotlib takes these results and displays it in a new window.

In right corner is button that when pressed, clears data from main table below. Function called, deletes array content for each input box, as well as deleting the table data and running the original function to recreate the initial table data.

The Cleanest Restaurant is located next to graph bar. Pressing itself queries the table and presents the results of which restaurant has the most positive results. This is done by using a function updates the table with an auto queried question that presents the user with the results of the restaurants with the most positive review scores.

Def Animal\_Violations():

Called when user presses input button on Program\_Main\_Window

Runs window that has contains text and images about collected data using the program on Animals Violations related to the New York Restaurant Inspections Dataset.

Has a “PageTransitioner” function same as described above, that allows the user to transfer back to the Main window of the program.

Has a button created using the Tkinter framework that has a “Command” function that when run, sends the user back to the main program window.

Def Clear\_Screen():

User Presses GUI button on main program page. Function clears content/arrays and textboxes of user inputted data and query output data.

Deletes current data presented in table and calls function to place initial data back into table.

Def DateSearch():

Takes values from input boxes Start date – End Date. If one of the values is empty, program throws up error. If both are valid and entered correctly. Queries database/linked arrays to present user with results of entered inputs.

Def KeywordSearch():

Called when user has pressed search button on main menu and has inputted a keyword along with start date and start time. Function is quries database and outputs results in table. Throws error if missing values or no results are found.

Def Cleanest\_Restaurant():

on main program page. Calls functions to clear of content/arrays and textboxes of user inputted data and query output data.

Searches for key words in the array that that contains the keywords that correlate to the violations column.

Presents data in the table in the main program window that corresponds to the inputted dates

These keys words are inputted by the user and grabbed

Def Create\_Graph():

Called when user presses input button on Program\_Main\_Window

Runs window that has contains text and images about collected data using the program on Animals Violations related to the New York Restaurant Inspections Dataset.

Has a “PageTransitioner” function same as described above, that allows the user to transfer back to the Main window of the program.

Has a button created using the Tkinter framework that has a “Command” function that when run, sends the user back to the main program window.

### 3.2.2 Data Structures / Data Sources

List of all data structures in the software (eg linked lists, trees, arrays etc) or eternal data sources. For each data structure in the list the following information is provided:

* Type of structure (tree, list etc),
* Description of where and how it is used
* List of data members, and what each one is for do
* List of functions that use it

Data is grabbed from the file and put into arrays

These arrays are then sorted into Linked lists

The linked list is given values that represent each section of the visual table. These values are sorted into their corresponding data categories such as GRADE, VIOLATIONS and SCORE.

The following is the complete list of Arrays that will be included in the Linked List:

**CAMIS:** Represents the ID given to each establishment.

**DBA:** The name given to the associated Restaurant

**BORO:** The District of Each Restaurant

**BUILDING:** Building number of the corresponding establishment

**STREET:** Street name of the Restaurant.

**PHONE:** Phone number of the restaurant

**CRITICAL FLAG:** Boolean of true or false that signals whether the violations of a restaurant were deemed to be of critical importance or not.

**SCORE:** Score given to the restaurant after Inspection/grading. Lowest score being the best result. On a scale of 1-100

**GRADE:** Grade given to a restaurant after Inspection. Scales from Z-D. Z being best. D being worst.  
GRADE DATE: Date the grade was presented to the establishment.

**VIOLATION DESCRIPTION:** Text description of violation given to business by inspectors.

**RECORD DATE:** Date of the recorded Inspection.

Using 11 different arrays. Each will be placed into a dictionary and given a label. These labels will be used to segment each data value into a linked list which will be used to contain and separate the data so that it can be queried by the user and displayed within the table in the main program window.

Functions that use theses arrays include:

Def Clear\_Page():

Def DateSearch():

Def KeywordSearch():

Def Cleanest\_Restaurant():

Def Create\_Graph():

### Detailed Design

Frameworks being used to create project:

Tkinter – Used to create all the GUI elements of the program.

MatPlotLib – Being used to create Graphs for the collected and queried data.

Pandas – Grants us easier and more effective methods of sorting and allocating the New York Restaurant Database data.

**“PageValue”.Mainloop()**

Mainloop is a Tkinter function that is called at the beginning of every page to ensure that a window opens to run the program. It’s essentially an infinite loop that allows the program to draw and redraw parts of the application on user interaction.

Def Page\_Start():

Start = Tk() #Initilizes the use of Tkinter

Start.Background(white) #Background colour of program

Start.geomerty( Size of program window e.g. “Width = 1280, Height = 920”)

Start.Title = (“Title on top of window”)

SmallBlackFrame/Border = Frame( Start bg=”black”)

PlacingFrame.Place( “Dimensions go here” e.g Width = 1280, Height = 920)

Most of the code above will be repeated on each page of the program as it both initializes the use of built-in python framework Tkinter, and keeps a design consistency across each window of the program . Imported from title.py

Def “Function to go to new Window”():

“CurrentWindow”.Destroy() # Destroys current program window

From “NewWindow e.g Main\_Page.py” import “Imported Function e.g Main\_Page”

RunImportedFunction() #Runs newly imported function used in separate file.

Basic function that will be used many times across the software to transition to each window of the program, by destroying the current window and running the imported function of the selected window. E.g from the Introduction page to the main page and vice versa.

Def Introduction\_Page():

Page\_Start()

Text - “Name of program” Bold and striking.

Text – Information on how to use program.

Function as described above to GotoMainwindow()

Button – Start button that begins the program with tkinter Command = GoToMainWindow

Introduction\_Page()

Def Program\_Main\_Window():

Page\_Start()

Sqlite3.connect(datebase\_name) #Connects application to database so data can be read and manipulated by program.

Multiple input buttons. Run.

3 input boxes that are placed around vertically next to each other.

Def Clear\_Screen()

Clear\_Screen() # Function either place in another file and imported or just constructed into this file

Button that is placed in right corner that has command=”Clear\_Screen” to clear user screen

Def GotoWindowAnimalViolations() #Clone of “Function go to new Window” as described above

Button that is gray, 350, 200 in width and height and placed in below date input boxes. has command=”GotoAnimalViolations” To go to Animal Violations Window

Def GotoExecReportWindow() #Clone of “Function go to new Window” as described above

Button that is gray, 350, 200 in width and height and placed in below date input boxes. has command=”GotoAnimalViolations” To go to Animal Violations Window

Program\_Main\_Window()

Def Animal\_Violations():

Page\_Start()

Text - Describing what information was discovered about that animal violation data.

Picture – AddImage(Height = 500 width = 500)

Picture.place(X and Y Co-ordiantes of Pictures)

Done Multiple times depending on the amount of information wanting to be presented

Def “Function to go to new Window”(): #Function described previously that when run destroys current window and loads function to open new window

Button – imput button with height, width, colour properties. Command = ProgramMainWindow

Animal\_Violations\_Page.Mainloop()

Animal\_Violations()

Def Clear\_Screen():

DateSearch.delete(all)

DateSearchEnd.delete(all)

SeachBox.Delete(all)

Feachall() fetch all main data.

Maintree.delete()

For loop that inserts initial values back into tree

#Deletes all data inside input boxes. Fetches data from database and deletes info in table before replacing it with the default start data using the same method to originally create it.

Def Search():  
 cursor.fetchall()

Def Create\_Graph():

Def Cleanest Restaurants():

Def Exec\_Report():

Page\_Start()

Text - Describing what information was discovered about that animal violation data.

Picture – AddImage(Height = 500 width = 500)

Picture.place(X and Y Co-ordiantes of Pictures)

Done Multiple times depending on the amount of information wanting to be presented

Def “Function to go to new Window”(): #Function described previously that when run destroys current window and loads function to open new window

Button – imput button with height, width, colour properties. Command = ProgramMainWindow

Animal\_Violations\_Page.Mainloop()

Animal\_Violations()

# User Interface Design

When deciding how to go about designing our GUI, the team looked at similar analysis software tools such as Power BI and Excel, to gain an understanding of what functionality and design rules we should be incorporating into our dataset program, to ensures it achieves its intended task as effectively as possible.

What we took away from our research was the professional tools had buttons and layout designs that both stuck out and were easy on the eye to observe and look at. Each element was displayed in the program in ways to make sure that important functions were given a larger amount of space and more priority on screen, emphasizes their importance/usage rate.

The use of one primary colour such as Green for Excel or Gray for Power BI made it so that these elements were easy on the eyes to view and made the overall layout clean to look at.

From our research we found that often the ones we found the most intuitive and effective at their job were the ones that We then used Draw.io to come up with example designs which are detailed and explained in the below sections.

## Structural Design

Structural design refers to the navigational and information structure of your product – the structure that supports the interface layout. How will you structure your product? How will you group your information? How will you navigate through your product? Why? This can take the form of a diagram showing structure and hierarchy, supported by a discussion and justification of your choices. Why have you made these design choices? Describe and outline the structure of your interface and of your information.

### 4.1.1 navigational design

The navigational interface of this design has been finalized keeping general design conventions and simplicity in mind. The final interface was decided upon a few team meetings and lots of tweaks depending on how the feature were to be implemented. Below in Figure 1 is the diagram of how the interface is implemented. The overall structure is likely to remain the same, but the design of some screens might change depending on the analysis.

### 4.1.2 structural design implementations

First as the user opens the tool an instructions page will open with instructions of how to use the tool, this is to make the tool quick and easier to use. After that the user can click on start to go to the main page which can be seen in figure 3.

In the main screen the user has 4 different paths to analyse the data.

* The first path is to retrieve relevant inspection detail for a selected period. To do this the user has to enter a start date and end date and then click on search for the inspection details or the user can also use a keyword to retrieve inspection details of a specific violation type. The inspection details will be seen under the selection options in the menu bar as can be seen in figure 3. this implementation has been decided after research of a few different analysis's tools. After that the user has the option to see the graph of the data, this can be done by clicking on graph and a screen of the graph will appear.
* For the second path the user can click on animal violation to see an analysis of the animal violations according to the suburbs. The team has decided to show this analysis in graph form because it will make it easier to understand the data. This will look like figure 5. To exit from this the user can click on the x on the top right corner of the page.
* The third path is to click on executive summary which will display a screen with executive summary of the entire data. This will look like figure 6. To exit from this the user can click on the x on the top right corner of the page.
* The fourth path is to click on cleanest restaurants, which will lead to our analysis of the cleanest restaurants from our analysis of their violations. This is the extra feature our team has decided to implement. The reason to include this feature is that a healthy competition can be promoted between the restaurants and this data can also be used to promote the restaurants. To exit from this the user can click on the x on the top right corner of the page.

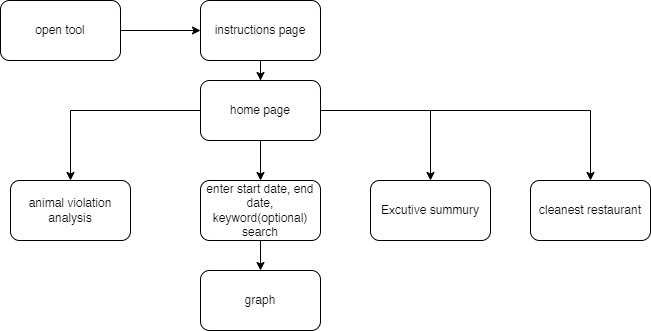


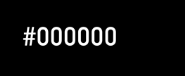
Figure 1

## Visual Design

### 4.2.1 Design choices

The design of this tool has been kept as simple as possible and been designed in a way that follows general design conventions. This has been done so the user can easily figure out how to navigate around the tool and thus decreases the learning time. The close button for certain the graph is a X symbol and placed in the top left corner as per convention.

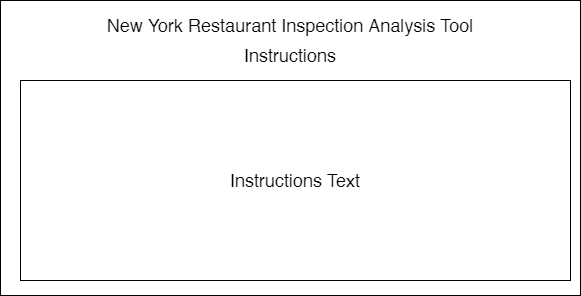
The user group for this toll is assumed to be normal industrial client and customers, as there are no usability requirements mentioned by the client. But the tool design and colours are chosen so it is usable the highest number of users. These are some basic design elements of the tool

* Font size of will be 16px or 18px depending on the element as this is not a text heavy tool. These font sizes are to make the tool easily readable.
* The font used will be San serif because it is easily readable and free.
* The colours used in the tool will be
  + Black 
  + Blue 
  + White 
  + Green 
* The buttons of the tool will be square shaped

### 4.2.2 Layout and wireframes

The tool will start with the instructions screen then go to the main screen, and from the main screen there are three 4 paths that can be taken the wire frames below show the various screens.

* Below is the wireframe of the instructions of the analysis tool. On this screen there will be instructions on how to use the tool. To go the next screen which is the main screen form here the user must click on start.



start

Figure 2

Below is the wireframe of the main screen of the analysis tool. In this screen the top right portion has all the selections for input which are

* Start date
* End date
* Keyword

And the buttons on this screen as shown in figure 3 are

* Search
* Clear
* Graph
* Executive summary
* Animal violations
* Cleanest restaurant

The start date and end date are compulsory inputs while the keyword is optional to view the inspection details. The table in the bottom will be empty initially but once the inputs are entered and the user clicks on search the details will be displayed. If the user wishes to see a graph of these details, they can click on graph which will lead to a new screen as shown in figure 4 There is a clear button to clear the inputs.

Below the inputs is the cleanest restaurant, animal violations and executive summary which are shown on figure 7, figure5 and figure 6 respectively.

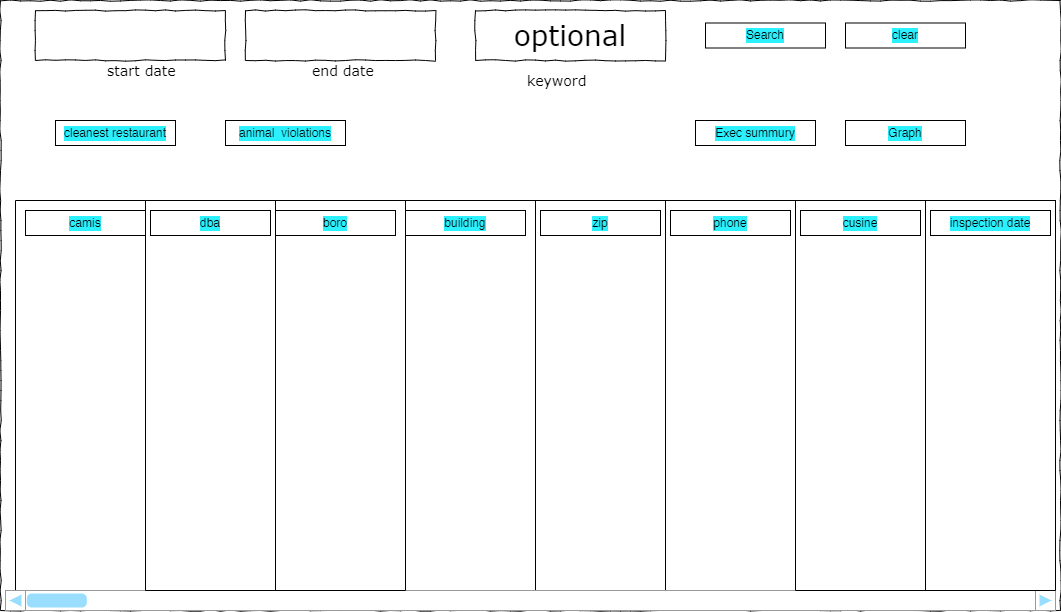


Figure 3

* Below is the wireframe of the graph screen. This screen will be opened on clicking graph and show the graph if the details that are shown in the main screen. This screen can be closed by clicking the x at the top right corner.

Figure 4

* Below is the wireframe of the animal violations analysis screen. This screen will appear when the animal violations button is clicked in the main screen. This screen will show analysis graph of the animal violation based on the suburb This screen can be closed by clicking the x at the top right corner.

Figure 5

Below is the wireframe of the wireframe of the executive report screen. This screen will appear when the Executive summary button is clicked in the main screen. This screen will show the executive report based on all the data of violations. This screen can be closed by clicking the x at the top right corner.

Figure 6

This screen will appear when the cleanest restaurant button is clicked in the main screen. This screen will show the restaurants with the least violations and be ranked. The screen has four divisions

* Ranks
* Restaurant name
* Suburb
* Cuisine

This screen can be closed by clicking the x at the top right corner.

Figure 7