Software Design Document

<Project Name>

Liangxian Zhao

Rafael Alexander s5277157

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

## System Overview

## Potential Benefits

# Requirements

## User Requirements

In this section you detail how a user is supposed to interact with or use your program. What do they ***need*** to be able to do? This should all be from the end users perspective. Can be a combination of narrative text and listing of needs.

**Assignment note: You have not been given a client/user, so you can make one up. Who do you think would be using your software?**

* User should have a working computer, a monitor, a mouse and a keyboard to access the python application
* User should have OS that can accommodate the python application
* User should know relevant keywords to the "NYC Inspection Results" dataset to input

## Software Requirements

In this section you detail what the requirements for the software are. What functionality will it provide? This is usually a formal listing, with requirements often using the word ‘Shall’. IE:

R1.1 The program shall accept multiple file names as arguments from the command line.

R1.2 Each file name can be a simple file name or include the full path of the file with one or more levels.

etc …

Can be primarily functional requirements, though you may include other types if you think of them.

* The program shall give the user 5 options for each feature correlating to data visualisation of the dataset
* The program shall give the user a text field to input keyword
* The program shall give the user date format to choose from
* The program shall give the user a list of options for animals and boroughs
* The program shall give the user the chance to reset their customisations of the results they want to see
* The program shall give the user the option to continue to another session

## Use Cases & Use Case Diagrams

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

# Software Design and System Components

## Software Design

A block diagram/flowchart of how your software might work

1. User opens app
2. User chooses function
3. (If needed) User enters time period/keyword/etc
4. App displays analyzation and visualization of “NYC Restaurant Inspection” dataset with chosen function and customization by user
5. User chooses to close app or to do another session

## System Components

### Functions

Preliminary list of all functions in the software. For each function in the list the following information is provided:

* a brief description of what it does (1 or 2 sentences);
* a list of the input parameters, and their data types, and what they are used for;
* a list of any side effects caused by the function (ie change global or member variables, changes data passed by reference from calling function etc)
* a description of the function’s return value
* F1: inspectionDetailsTime
  + This functions retrieve inspection details (restaurant name, date, —) from “NYC Restaurant Inspection” dataset within the time period selected by the user.
  + Parameters:
    - Time period, format: String
  + Change in global variables
    - Time variable
  + Return value
    - SQL query
    - Dictionaries of restaurants as the key and respective inspection details as lists as the value
* F2: violationDistribution
  + This functions retrieves and plots the distribution of violations over different suburbs of NYC from the “NYC Restaurant Inspection” dataset within the time period selected by the user.
  + Parameters:
    - Time period, format: String
  + Change in global variables
    - Time variable
  + Return value
    - SQL query
    - Plot of violation distribution over different suburbs of NYC
* F3: violationKeyword
  + This functions retrieves violations from the “NYC Restaurant Inspection” dataset that contains the keyword entered by the user.
  + Parameters:
    - Keyword, format: String
  + Change in global variables
    - Keyword variable
  + Return value
    - SQL query
    - List of violations that contains keyword entered by the user
* F4: animalRelCases
  + This functions cases from the “NYC Restaurant Inspection” dataset that is related to animals, their trend over time and distribution over suburbs
  + Parameters:
    - animalType, format: String
  + Change in global variables
    - animalType variable
  + Return value
    - SQL query
    - Dictionary of cases related to animals as the key and their trend overtime and distribution over suburbs as lists as the value
* F5: inspectionDetailsBorough
  + This functions retrieve inspection details (restaurant name, date, —) from “NYC Restaurant Inspection” dataset within the borough area selected by the user.
  + Parameters:
    - Borough, format: String
  + Change in global variables
    - Borough variable
  + Return value
    - SQL query
    - Dictionaries of restaurants as the key and respective inspection details as lists as the value

### 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
* D1: NYC Restaurant Inspection
  + Type: CSV
  + This is the main data source/dataset used in this app to retrieve all the data that is going to be analyzed and visualized by the app for the user
  + Data member: Key: Restaurant name, Value: List of Details (Suburbs, zipcode, cuisine, inspection date, violations)
  + Functions: F1, F2, F3, F4, F5
* D2: Restaurant Names
  + Type: List of Strings
  + This is the main data source/dataset used in this app to retrieve all the data that is going to be analyzed and visualized by the app for the user
  + Data member: Names
  + Functions: F1, F2, F3, F4, F5
* D2: DateRange
  + Type: List of 2 datetime objects
  + This is the data structure that records the date range specified by the user
  + Data member:
  + Functions: F1, F2, F4
* D3: Keyword
  + Type: string
  + This is the data structure that records the keyword specified by the user
  + Data member:
  + Functions: F3
* D4: animalType
  + Type: string
  + This is the data structure that records the animal type specified by the user
  + Data member:
  + Functions: F4
* D5: Borough
  + Type: string
  + This is the data structure that records the borough specified by the user
  + Data member:
  + Functions: F5

### Detailed Design

Pseudocode for all non-standard / non-trivial algorithms that operate on data structures

# User Interface Design

This is your initial interface design. Describe the tools you used for this design stage and any key findings that informed your design. This introduction is descriptive and should explain what you have completed for the actual design work you will present in the sub-sections below.

## 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.

## Visual Design

Detail your visual design: Layout, visual elements, icons, graphics, style, colour, fonts general screen designs. This can be sketches, wireframes, mockups etc, supported by a discussion, explanation, and justification of your choices.