Project Plan

New York Restaurant Inspection Results

Student Names

Kartik Mathur (s5309927)

Manish Shrestha (s5308120)

Nivethaa Elangovan ( s5298899 )

Table of Contents

[1.0 Introduction 3](#_Toc145020794)

[1.1 Background 3](#_Toc145020795)

[1.2 Scope 3](#_Toc145020796)

[1.3 Document contents 3](#_Toc145020797)

[2.0 Work Breakdown Structure 4](#_Toc145020798)

[3.0 Activity Definition & Estimation 5](#_Toc145020799)

[4.0 Gantt Chart 7](#_Toc145020800)

# Introduction

## Background

NYC Restaurant inspection dataset has been taken from Kaggle which contains a lot of Restaurant inspection data collected by the Department of Health in NYC. The data covers all of NYC from 2010 over a period of 7 years. The data gives us details of the violations done by each restaurant with their record details. The main attributes recorded for each restaurant are suburb, cuisine, inspection date, inspection type, Violation code, description, score and grade(A-F). (NYC Restaurant Inspections, 2023)

The motivation for collecting this data is to analyse how many restaurants have been closed in a particular time frame or can close, what types of violations are common in certain suburbs, which cuisines are more likely, and differences in franchises and self-made restaurants. (NYC Restaurant Inspections, 2023)

The client requires a system to find interesting insights with the dataset provided with the following features:

* List the information for the specified date range when the user of the system enters the start date and end date.
* When navigating to the dashboard page displays different charts which give a summary of data for different time frames. (Violations over different suburbs, violation related to animals over a period, violation related to animals over different suburbs and violation count based on cuisines)
* Allow users to retrieve data with particular keywords.
* Allow users to export data to store the information generated.

## Scope

The scope of the project plan is to provide information about how the program or system will be created with proper planning and time allocated. The work breakdown structure gives an idea of the phases of the project. The activity definition and estimation shows what is to be done during the phase like resources and period. The time frame details for tracking progress for completing each activity are shown in the Gantt chart. This document helps the project manager to update the development progress to respective stakeholders and makes it easy to track the process of development.

## Document contents

The document consists of three major contents to enable proper project planning. They are Work breakdown structure, Activity definition and Gantt chart. The work breakdown structure divides each tasks or phases into multiple steps and sets a guideline or flow chart for the entire project. The Activity definition & estimation provide a more detailed approach to how each step is accomplished and resources assigned. Gantt chart is used to schedule the project activities into time frames and marked by different phases or development. This make it easy to keep track of the dates from start to finish.

# Work Breakdown Structure

A diagram of a company

Description automatically generated

# Activity Definition & Estimation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity ID** | **Activity** | **Activity Description** | **Duration** | **Resources** | **Periods** |
|  | **Planning** | | | | |
| 1 | Meeting with client | Meet the client and eliciting the requirements. | 2 hours | 1 | 1 |
| 2 | Understand client requirements | Discuss the requirements given by the client with the team, and make sure everyone understands them. | 3 Hours | 3 | 1 |
| 3 | Define objectives and scope | Define all the objectives clearly with the scope of those objectives. | 5 Hours | 3 | 2 |
| 4 | Create project plan | Create time periods of work for the project and break the project into subsystems for easier workflow. | 2 Hours | 1 | 2 |
| 5 | Allocate resources | Allocate the resources to the different tasks according to based on the requirements. | 2 hours | 2 | 2 |
| 6 | Schedule milestones and timelines | Decide on the timeline of the whole project and assign milestones and dates to all subsystems in the project. | 4 hours | 2 | 3 |
| 7 | Develop use cases | Develop use case diagrams, and descriptions for all subsystems. | 6 hours | 3 | 3 |
|  | **Analysis** | | | | |
| 8 | Requirement analysis | Analyse if the requirements are all in line with scope and objectives. | 2 Hours | 1 | 3 |
| 9 | Analyse past and present data | Analyse all the past data from various sources and compare it to the current data. | 4 Hours | 3 | 4 |
| 10 | Evaluate scope | Evaluate the scope for all subsystems. | 1 hour | 3 | 4 |
| 11 | Document the requirements | Document the requirements and keep track of any changes. | 5 hours | 3 | 4 |
|  | **Design** | | | | |
| 12 | System software and components design | Designing the software architecture and various system components and main system software logic. | 4 hours | 3 | 5 |
| 13 | Interface design | Designing an interface for the software as elicited by the client. | 5 hours | 2 | 5 |
| 14 | Design documentation | Making a software design document to record the design choices and functionalities. | 4 hours | 2 | 6 |
| 15 | Obtain client approval | Showing the design to client and get their views on the whole design of the software. Get approval from them to finalise the design. | 2 Hours | 2 | 6 |
|  | **Implementation** | | | | |
| 16 | Create user interface | Implementing the design finalised with client’s approval into the software. | 16 hours | 2 | 7, 8 |
| 17 | Develop backend | Develop the backend according to the system design and the interface. | 48 hours | 3 | 9 - 14 |
| 18 | System Integration | Integrate all the components with each other to finalise the software as a whole. | 8 hours | 2 | 15 |
| 19 | Document the development process | Document and record the whole process, everything that was decided, challenges faced and solutions. | 8 hours | 2 | 16 |
|  | **Testing** | | | | |
| 20 | Analyse requirements and identify loopholes | Analyse all the requirements and look for any loopholes that would cause problems with the software. | 5 hours | 2 | 17 |
| 21 | Prepare test plan | Prepare a testing plan that covers unit and coverage testing. | 3 hours | 2 | 17 |
| 22 | Develop test cases | Develop enough test cases to cover every possible output including the ones that could throw errors. | 5 hours | 3 | 18 |
| 23 | Run test cases | Run all test cases, and record all the outputs, expected or unexpected. | 8 hours | 3 | 18, 19 |
| 24 | UAT | Do real-world User acceptance testing, and record all the outcomes from that. | 8 hours | 2 | 19, 20 |
| 25 | Prepare test closure report | Prepare a report by combining the outcomes of all the tests with additional comments. Check if further development is required or if the software can be deployed. | 4 hours | 1 | 20 |
|  | **Deployment and Maintenance** | | | | |
| 26 | Documentation | Document the process, from requirements to testing and compile all the reports. | 4 hours | 2 | 21 |
| 27 | System installation and configuration | Configure the system to be installed in line with the requirements provided by the client, and make an installation guide. | 5 hours | 2 | 21 |
| 28 | Deployment | Deploy the software in accordance with client requirements. | 16 hours | 3 | 22, 23 |
| 29 | Maintenance | Make a maintenance plan, explain it to client and perform maintenance whenever required. | 2 hours | 1 | 24 |

# Gantt Chart

