**Research Assignment 2 – Practical Project Part 2 Planning**

**Prepared by :** Vaishali Jaiswal

**Course Name & Code:** Programming Language

Research Proj.

CST8002\_040

**Professor’s Name:** Tyler DeLay  
**Due Date:** 2nd February, 2025

1. **Development Environment Changes**

There are no changes to my development environment. I am using the same setup as Practical project part 1.

1. **Research and Learning Resources**

For Practical Project Part 2, I researched N-Layered Architecture and Model-View-Controller (MVC) Architecture to determine the best approach for my project. After reviewing multiple sources, I decided to use N-Layered Architecture because it provides a structured way to separate concerns and aligns well with my console-based Python project.

**N-Layered Architecture vs. Model-View-Controller (MVC)**

|  |  |
| --- | --- |
| **N-Layered Architecture** | **MVC Architecture** |
| **Presentation Layer:** Handles user input and output | **Model:** Represents the data and logic |
| **Business Layer:** Manages data processing, logic, and calculations | **View:** Handles the user interface |
| **Persistence Layer:** Handles file storage, database interactions, and data retrieval | **Controller:** Manages user input and updates the Model and View |

Since my project is console-based and does not involve GUI elements, N-Layered Architecture is the better choice because it allows me to separate data storage, logic, and user interaction effectively.

**References:**

1. R. Sanderson, “Structuring Python Applications with Layered Architecture,” Medium, 2021. Available: <https://medium.com>

2. GeeksforGeeks, “Difference between N-Tier Architecture and MVC Architecture,” GeeksforGeeks, 2020. Available: <https://www.geeksforgeeks.org>

3. Python Software Foundation, "Python File Handling," Python Docs, 2023. Available: <https://docs.python.org/3/library/io.html>

4. Python Documentation, “Working with CSV Files,” Python.org, 2023. [Online]. Available: <https://docs.python.org/3/library/csv.html>

1. **WBS and Gantt Chart for Practical Project Part 2**

* **Work Breakdown Structure (WBS)**

1. Repository & Version Control Setup

1.1 Create a new private GitHub repository for Part 2

1.2 Add professor as a collaborator (if not already added)

1.3 Copy initial code from Practical Project Part 1

1.4 Create subfolders for N-Layered Architecture

1.5 Commit initial project setup & tag as v2.0

2. Implement N-Layered Architecture

2.1 Model Layer (Data Entity)

2.1.1 Modify or reuse TrafficDataRecord class

2.1.2 Implement accessors/mutators for data fields

2.1.3 Commit changes

2.2 Persistence Layer (File I/O)

2.2.1 Read CSV dataset & initialize 100 records

2.2.2 Implement function to write records to file using UUID

2.2.3 Handle missing/corrupt file exceptions

2.2.4 Commit changes

2.3 Business Layer (Data Processing)

2.3.1 Implement in-memory data storage (List/Array)

2.3.2 Add search, filter, and sorting functionality

2.3.3 Commit changes

2.4 Presentation Layer (User Interaction)

2.4.1 Implement menu system for user options

2.4.2 Ensure user’s full name appears in the program output

2.4.3 Commit changes

3. Implement Core Functionalities

3.1 Load and process 100 records from the dataset

3.2 Implement CRUD operations (Create, Read, Update, Delete)

3.3 Save updated records to disk using a UUID-based filename

3.4 Implement exception handling for CRUD operations

3.5 Commit changes for each functionality

4. Unit Testing (Single Test)

4.1 Write one unit test for one core functionality

4.2 Run the unit test & verify results

4.3 Commit unit test implementation

5. Documentation & Final Submission

5.1 Update comments and docstrings for all files

5.2 Take screenshots of working program

5.3 Commit & push final version to GitHub

5.4 Submit final project report & Gantt Chart

Milestone: “Practical Project Part 2 Delivered” (set with due date)

* **Gantt Chart**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**