LOW LEVEL DESIGN (LLD)

|  |  |
| --- | --- |
| **Written By** | **Saud Qureshi** |
| **Document Version** | **1.0** |
| **Last Revised Data** | **18-Dec-2024** |

Document Version Control:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 1.0 | 18-Dec-2024 | Saud Qureshi |  |
|  |  |  |  |

Contents

[1. Introduction 3](#_Toc188357848)

[1.1 What is Low-Level design document? 3](#_Toc188357849)

[1.2 Scope 3](#_Toc188357850)

[2. Architecture 3](#_Toc188357851)

[3. Architecture Description 4](#_Toc188357852)

[3.1 Data Ingestion 4](#_Toc188357853)

[3.2 Pre-processing 4](#_Toc188357854)

[3.3 Transformation 4](#_Toc188357855)

[3.4 Display 4](#_Toc188357856)

[3.5 Dashboard 4](#_Toc188357857)

[4. Test Cases 5](#_Toc188357858)

# Introduction

## What is Low-Level design document?

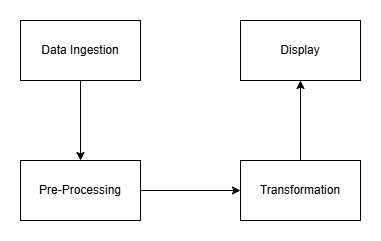
The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for flight fare estimation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

## Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

# Architecture

Below is architecture of this project.



# Architecture Description

## 3.1 Data Ingestion

The database contains two files, a .parquet and a .csv file. The files are loaded from a hdfs file path. The data is downloaded from Kaggle. The dataset contains information such as the city, and the amount of prescriptions and the number of prescripts, etc

## 3.2 Pre-processing

The next step is data pre-processing in which the data is cleaned and new columns are added wherever required. The nulls values are taken care of by deletions and imputations.

## 3.3 Transformation

After cleaning and doing pre-processing steps on the data. The transformation steps are carried out. I have used pyspark to carry out the operations.

## 3.4 Display

The final result is stored in a dataframe and displayed.

## 3.5 Dashboard

The final dataframe is displayed on a dashboard on a web page using streamlit.

# Test Cases

Test cases are given below

|  |  |  |
| --- | --- | --- |
| **Test Case Description** | **Pre-Requisite** | **Expected Result** |
| Verify whether the Front end is visible to the user |    Streamlit command should be ran | Yes, it is visible. |
| Verify the logs |    Logs file is visible | Yes, it is visible |
|  |

# 