Low Level Design

AMAZON SALES ANALYSIS

WRITTEN BY	HARSH DIXIT
DOCUMENT VERSION	1.0
LAST REVISED DATE	11/10/24

Contents

1. Introduction

The purpose of this document is to provide a detailed overview of the Low-Level Design (LLD) for the Aamazon Sales Analysis project. This document outlines the architecture, data flow, components, and design considerations for building and maintaining the Power BI dashboard for sales analysis. It will serve as a blueprint for the development team to understand the technical aspects of web scraping, data transformation, data storage, and interaction with the database, as well as the deployment of the final solution.

The dashboard will allow users to track and analyze sales trends, including sales by region, product categories, time periods, and other relevant business metrics. The document also includes unit test cases to validate the functionality and performance of the dashboard.

1.1 What is Low-Level Design Document?

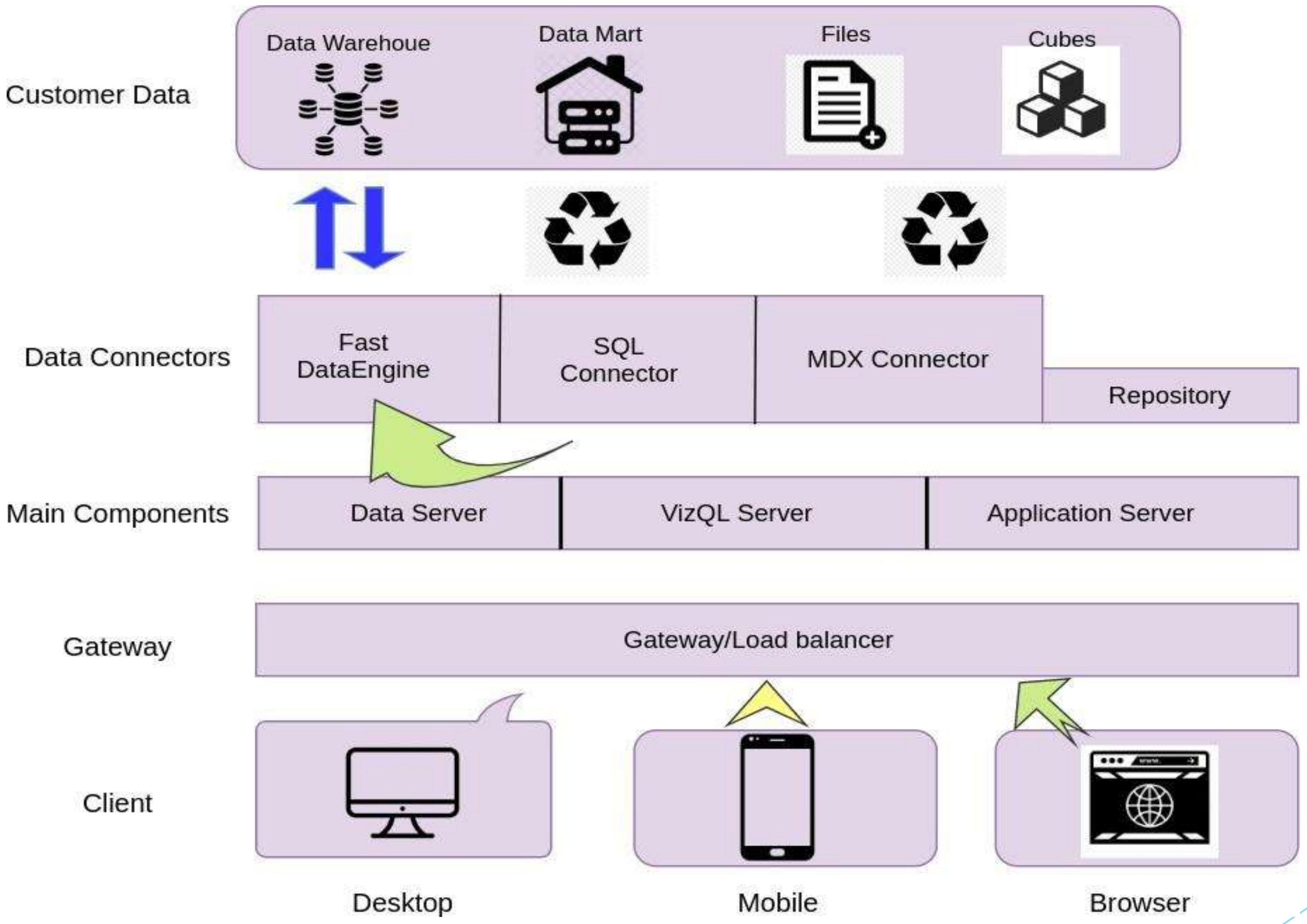
A Low-Level Design (LLD) document provides an in-depth technical explanation of the system components and their interactions. It describes how individual modules work, the technologies used, data flows, and specific implementation details. This document focuses on the design of the components for extracting, transforming, and loading (ETL) data, connecting with the SQL database, and creating visualizations in Power BI.

1.2 Scope

The scope of this LLD document is to cover all technical details related to the development of a Power BI dashboard for Amazon Sales Analysis, including:

- *Data extraction using web scraping techniques.*
- *Data transformation and cleaning.*
- *Insertion and retrieval of data from a SQL Server database.*
- *Deployment and testing procedures.*
- *Integration with other systems (if any).*

2. Architecture



This section provides an overview of the architecture for the system. The architecture will follow a modular approach where data extraction, transformation, loading, and visualization are separated into distinct components, ensuring scalability and maintainability.

- ***Data Source***: *Web scraping of Amazon sales data from specific endpoints or APIs.*
- ***ETL Process***: *Data will be transformed into a structured format suitable for analysis.*
- ***Database***: *A SQL Server will store the cleaned and transformed data.*
- ***Visualization***: *Power BI will be used for generating the sales dashboard.*

3. Architecture Description

This section will describe each component of the architecture in detail.

3.1 Data Description

The system will handle various types of data, including:

- *Product information (name, category, price, etc.).*
- *Sales data (units sold, revenue, etc.).*
- *Time-based data (daily, weekly, monthly sales).*
- *Regional data (sales by geographic locations).*

The data will be collected via web scraping and transformed into a structured format before being inserted into the database.

3.2 Web Scraping

Web scraping will be used to extract sales data from relevant sources (e.g., Amazon webpages or APIs). The scraping tool will be designed to handle dynamic pages, pagination, and data parsing.

- ***Tools Used**: (e.g., Python with Beautiful Soup, Selenium, or a similar scraping framework).*
- ***Frequency**: Data scraping will be scheduled daily to keep the dataset updated.*

3.3 Data Transformation

Once the raw data is scraped, it will undergo transformation to clean and normalize it. This will involve:

- *Removing duplicates.*
- *Filling in missing data.*
- *Normalizing categorical fields.*
- *Aggregating data (e.g., total sales per product, region, or time period).*

3.4 Data Insertion into Database

After the data is cleaned and transformed, it will be inserted into a SQL Server database. The database schema will include tables such as:

- *Products: Storing product information.*
- *Sales: Storing sales records for each product.*
- *Regions: Storing geographic data related to sales.*

This will allow for efficient querying and reporting in Power BI.

3.5 Connection with SQL Server

Power BI will connect to the SQL Server database using secure connections to fetch the necessary data for visualization. Direct queries or import methods will be used, depending on the performance and latency requirements.

3.6 Export Data from Database

For reporting or archival purposes, data may be exported from the SQL Server database to external formats such as CSV, Excel, or other file formats. The system will provide scheduled exports or on-demand exports.

3.7 Deployment

The deployment process will involve:

- *Setting up the web scraping scripts on a cloud server or local machine.*
- *Scheduling regular scraping and ETL jobs.*
- *Deploying the Power BI dashboard on Power BI Service or embedding it within another platform.*

Monitoring tools will be used to ensure that scraping, transformation, and dashboard updates happen without issues.

4. Unit Test Cases for this Pinned Power BI Dashboard

To ensure the reliability of the dashboard and its components, unit tests will be created for the following:

- ***Data Scraping****: Verify that the correct data is scraped and formatted as expected.*
- ***Data Transformation****: Validate that data cleaning, deduplication, and normalization work correctly.*
- ***Database Insertion****: Check that data is correctly inserted into the SQL Server database with no missing records.*
- ***Dashboard Visualizations****: Ensure that the visuals in Power BI reflect accurate data and update as expected.*
- ***Connection Tests****: Test the database connections and ensure that data queries return the expected results.*