Low Level Design

# Deloitte Case Study

|  |  |
| --- | --- |
| **Written By** | Author 1, Author 2 |
| **Document Version** | 0.3 |
| **Last Revised Date** |  |

**DOCUMENT CONTROL**

## Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **AUTHOR** | **COMMENTS** |
| 0.1 | 20- April -  2024 | Author 1 | Introduction and architecture defined |
| 0.2 | 20 - April-  2024 | Author 2 | Architecture & Architecture description appended and  updated. |
|  |  |  |  |
|  |  |  |  |

**Reviews:**

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **REVIEWER** | **COMMENTS** |
| 0.2 | 22- April -  2024 | Author 1 | Unit test cases to be added |

**Approval Status:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **VERSION** | **REVIEW**  **DATE** | **REVIEWED BY** |  | **APPROVED BY** | **COMMENTS** |
|  |  |  |  |  |  |

**Contents**

1. **Introduction 04**
   1. **What is Low-Level Design Document? 04**

|  |  |  |
| --- | --- | --- |
| **1.2** | | **Scope 04** |
| **2.** | **Architecture 05** | |
| **3.** | **Architecture Description 08** | |
|  | **3.1** | **Data Description 08** |
|  | **3.2** | **Web Scrapping 08** |
|  | **3.3** | **Data Transformation 08** |
|  | **3.4** | **Data insertion into database 08** |
|  | **3.5** | **Connection with SQL server 08** |
|  | **3.5** | **Export Data from database 12** |
|  | **3.6** | **Deployment 12** |
| **4.** | **Unit** | **test cases 15** |

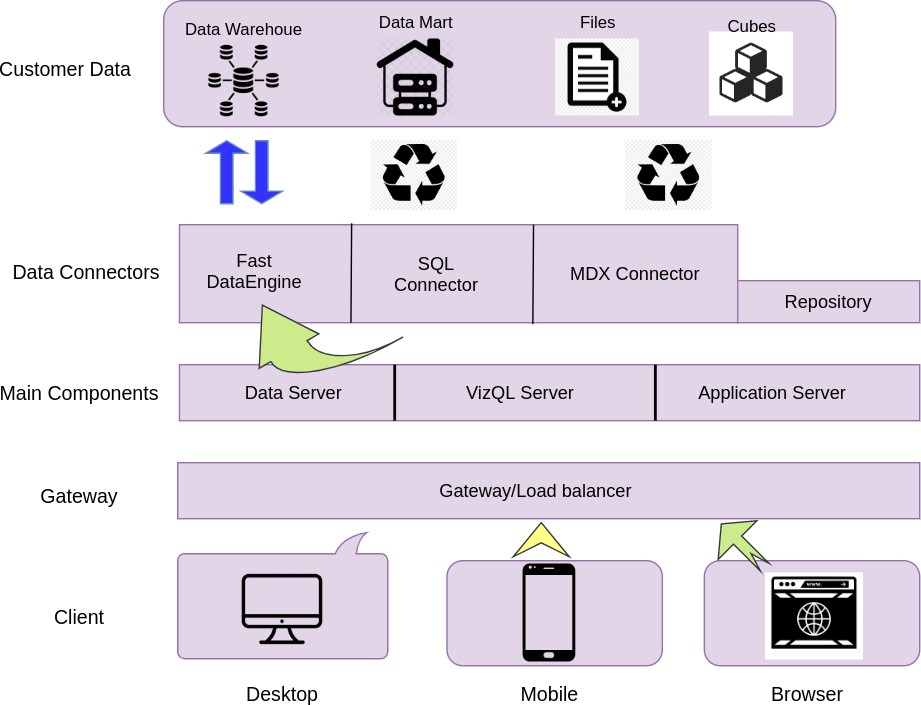
1. **Introduction**
   1. **What is Low-Level design document?**

The goal of the LDD or Low-level design document (LLDD) is to give the internal logic design of the actual program code for the House Price Prediction dashboard. LDD describes the class diagrams with the methods and relations between classes and programs 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. The 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



**Power BI Server Architecture**

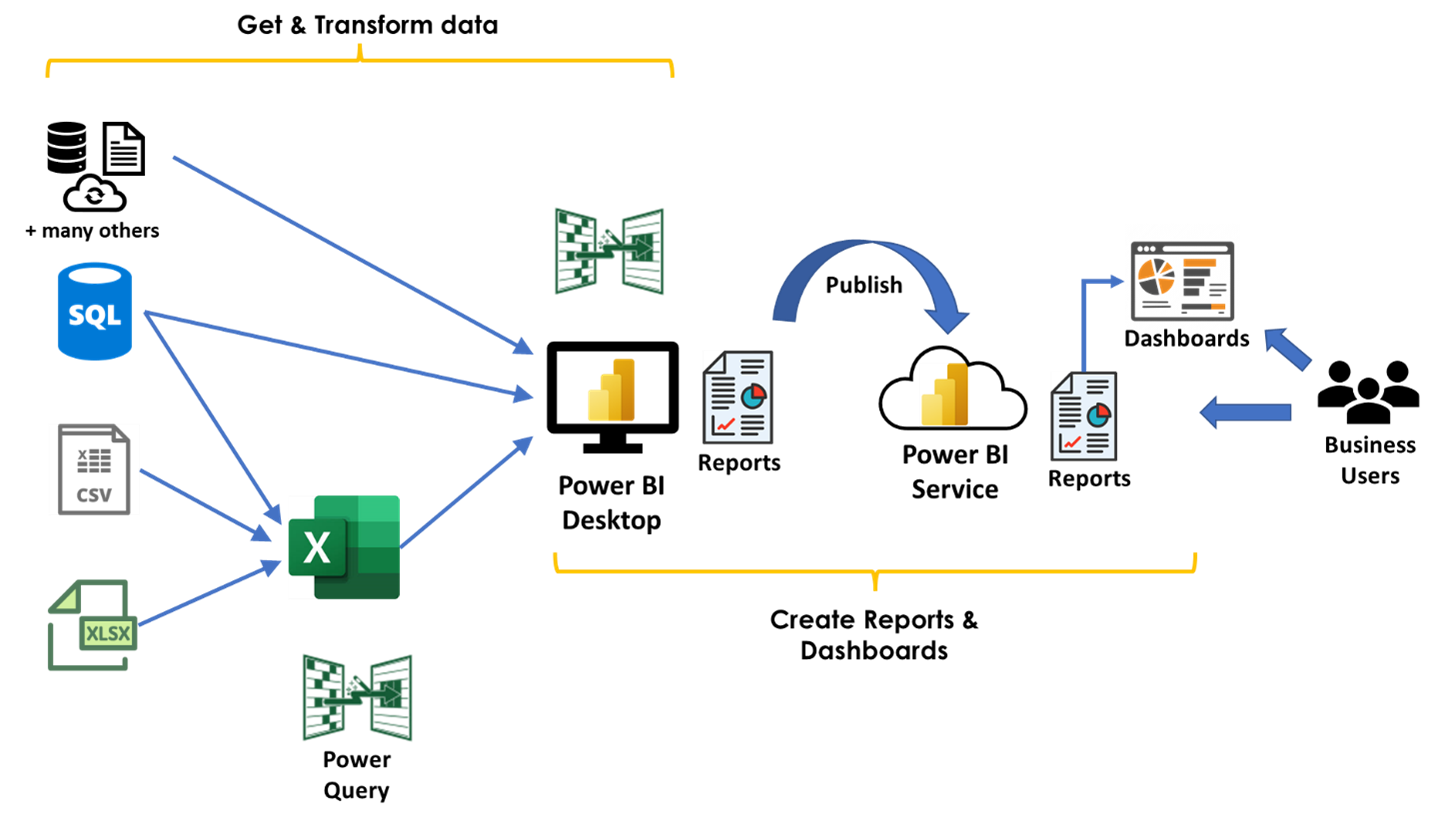
Power BI has a scalable, n-tier client-server architecture that serves mobile clients, web clients, and desktop-installed software. Power BI Service architecture supports fast and flexible deployments.

The following diagram shows Power BI ‘s architecture:

### Shape Description automatically generated with low confidenceLOW LEVEL DESIGN

**6**

### Power BI Communication Flow

****

Power BI is internally managed by the multiple server processes.

Dashboards are created by designers using the **Power BI service**(also known as Power BI online).  The Power BI service is the SaaS (*Software as a Service*) part of Power BI.  Each of the visualizations that you see on a Power BI dashboard is a *tile* that is *pinned* from a Power BI *report.* To create a dashboard, you therefore need to create the underlying reports in the **Power BI Desktop** app.

The diagram below shows a *typical* process for creating Power BI reports and dashboards.

* **Get & Transform data**.

This is where you identify the sources of data for the information you want to display in the report/dashboard.  A typical report or dashboard might use data from databases, as well as CSV files and Excel spreadsheets.  The process of getting and transforming the data can be done in Excel or Power BI Desktop.  You could, for example, import data into Excel from SQL Server, perform some transformations using *Power Query,* and then import the Excel workbook into Power BI Desktop.  Alternatively, you can work in Power BI Desktop and query/import/transform data from there.  The end result is the same – you’ll have a Power BI *dataset* that you can use to create reports and then dashboards.

* **Create Reports and Dashboards.**

A Power BI *report* provides views into a data model, with visualizations that represent different findings and insights from that data model. Reports normally have a detailed operational focus, whereas *dashboards* provide a higher level ‘quick glance’ overview of a business area.  Reports are created in Power BI Desktop by adding visualizations and filters to the canvas based on the dataset(s) created in the previous step.  Once reports have been created, they are published to the Power BI Service where they can be shared with end-users and (optionally) dashboards created from them.  Dashboards are simply containers that display highlighted visualizations from a collection of reports.

1. **Architecture Description**
   1. **Data Description**

The Dataset contains Three file Cpi, Exchange Rate and Export Merchandise of different country

Annually,Monthly

## Web Scrapping

Web scraping is a technique to automatically extract content and data from websites using bots. It is also known as web data extraction or web harvesting. Web scrapping is made simple now days, many tools are used for web scrapping. Some of python libraries used for web scrapping are Beautiful Soup, Scrapy, Selenium, etc.

## Data Transformation

In the Transformation Process, we will convert our original datasets with other necessary attributes format. And will merge it with the Scrapped dataset.

## Data Insertion into Database

1. Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.
2. Table creation in the database.
3. Insertion of files in the table

## Make the SQL connection and set up the data source

**Step 1: Configuring Tableau**

To configure a connection to SQL Server in Power BI, follow these steps:

1. **Launch Power BI Desktop**: Open Power BI Desktop on your workstation.
2. **Get Data**:
   * Click on "Home" in the Power BI ribbon.
   * Select "Get Data" from the options.
3. **Choose SQL Server**:
   * In the "Get Data" window, select "SQL Server" from the list of available data sources.
   * Click "Connect."
4. **Provide Connection Details**:
   * In the "SQL Server database" dialog box, enter the server name where your SQL Server instance is hosted. If you're using a port other than the default, include the port number after the server name, separated by a comma. For example: **my\_server,8051**.
   * Choose your authentication method:
     + **Windows Authentication**: This uses your current Windows credentials to connect to the SQL Server. Click "Connect" to proceed.
     + **SQL Server Authentication**: Enter a username and password to connect. Check the "Use the following User Name and Password" box and enter your credentials, then click "Connect."
5. **Select Database and Objects**:

After connecting, you will see a navigator pane listing the available databases on the SQL Server. Select the database you want to connect to and choose the tables or views you want to import into Power BI.

Click "Load" to load the selected data into Power BI or "Transform Data" to edit and transform the data before loading.

**Data Import**:

Power BI will import the selected data into the Power BI Desktop. You can then start building reports and visualizations using this data.

**Step 2: Configuring Data Source**

Configuring Power BI

After successfully connecting to your data source in Power BI, follow these steps to configure your data:

1. **Naming the Data Source**:
   * In the data source page, select the data source name option.
   * Provide a unique name for the database you are using. Using a unique name helps users easily identify the database from which data is being fetched.
2. **Selecting the Schema and Table**:
   * Use the schema drop-down list from the column on the left to select the desired schema.
   * You can also use the search functionality to find the schema or table you want.
   * Once you've selected the desired schema, find and select the table you want to analyze.
   * Drag the selected table onto the canvas.
3. **Analysis in Power BI**:
   * Now, click on the "Data" or "Report" tab to start your analysis.
4. **Using Custom SQL**:
   * If you want to focus on specific SQL statements rather than querying the entire database, you can use the Custom SQL feature.
   * Click on the "Advanced options" or "Transform data" to access the Custom SQL option.
   * A new dialog box will open up where you can input the SQL query you want to execute.

By following these steps, you can easily configure your SQL Server data source in Power BI and start your analysis.

## Export Data from Database

Data Export from Database - The data in a stored database is exported as a CSV file to be used for Data Pre-processing.

Sure! If you're working with Power BI instead of Tableau Public, here's how you would save and publish your dashboard:

1. **Server**:
   * In Power BI Desktop, go to the "Home" tab on the ribbon.
   * Click on "Publish" to upload your dashboard to the Power BI service.
2. **Power BI Service**:
   * Once published, your dashboard will be available on the Power BI service.
   * You can access it by logging into the Power BI service at https://app.powerbi.com/.
3. **Save to Power BI Service**:
   * After making changes to your dashboard in Power BI Desktop, save your file.
   * Then, click on "Publish" to save the changes to the Power BI service.
4. **Change Accordingly**:
   * If it's your first time publishing, you'll need to log in using your Microsoft account associated with Power BI.
   * Follow the on-screen instructions to complete the publishing process.

Remember, to share your dashboard with others, you'll need to manage permissions within the Power BI service. You can set who can view or edit the dashboard based on their email addresses or groups.

### Shape Description automatically generated with low confidence

**12**

Shape

Description automatically generated with low confidence