**LOW LEVEL DESIGN (LLD) DOCUMENT**

**APP STORE ANALYSIS**

|  |  |
| --- | --- |
| **Version:** | **1.0** |
| **Last Revised Date:** | **03-11-2021** |
| **Written By:** | **Balaji Mummidi** |

**Document Version Control:**

|  |  |  |
| --- | --- | --- |
| Date Issued | Version | Author |
| 3rd November,2021 | 1.0 | Balaji Mummidi |

**Contents**

|  |  |  |
| --- | --- | --- |
| S. No | Topic | Page No |
| 1 | Introduction | 4 |
| 1.1 | What is Low Level Design Document | 4 |
| 1.2 | Scope | 4 |
| 2 | Architecture | 5 |
| 2.1 | Tableau Architecture | 6 |
| 3 | Architecture Description | 7 |
| 3.1 | Data Description | 7 |
| 3.2 | Web Scraping | 7 |
| 3.3 | Data Transformation | 8 |
| 3.4 | Creating relationships between Parameters | 8 |
| 3.5 | Deployment | 9 |
| 4 | Unit Test Cases | 9 |

1. **Introduction**

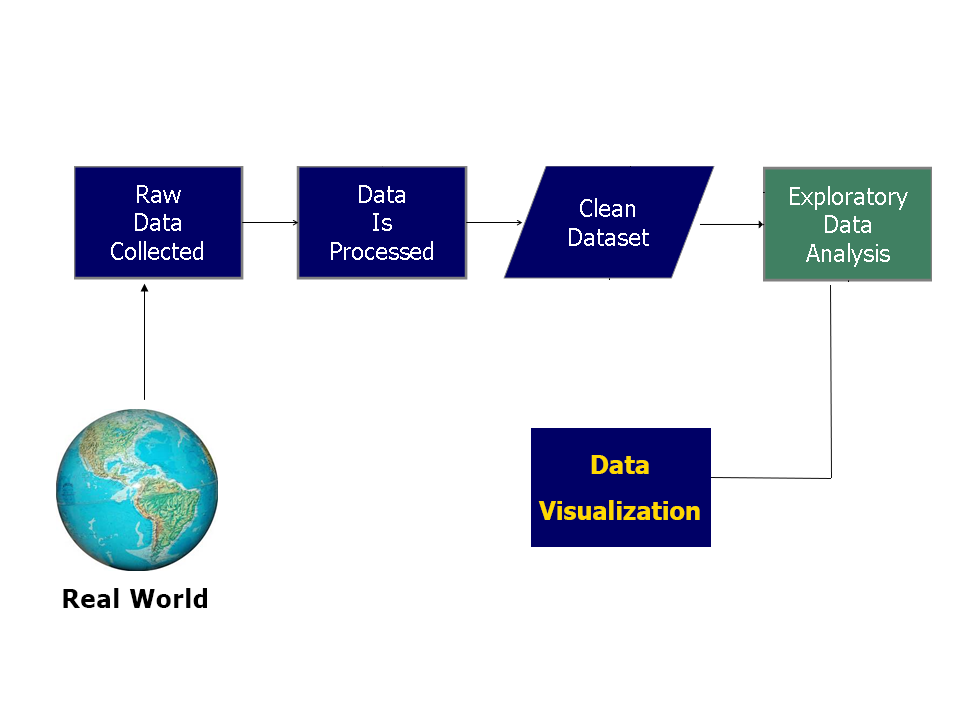
# What is Low Level Design Document

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.

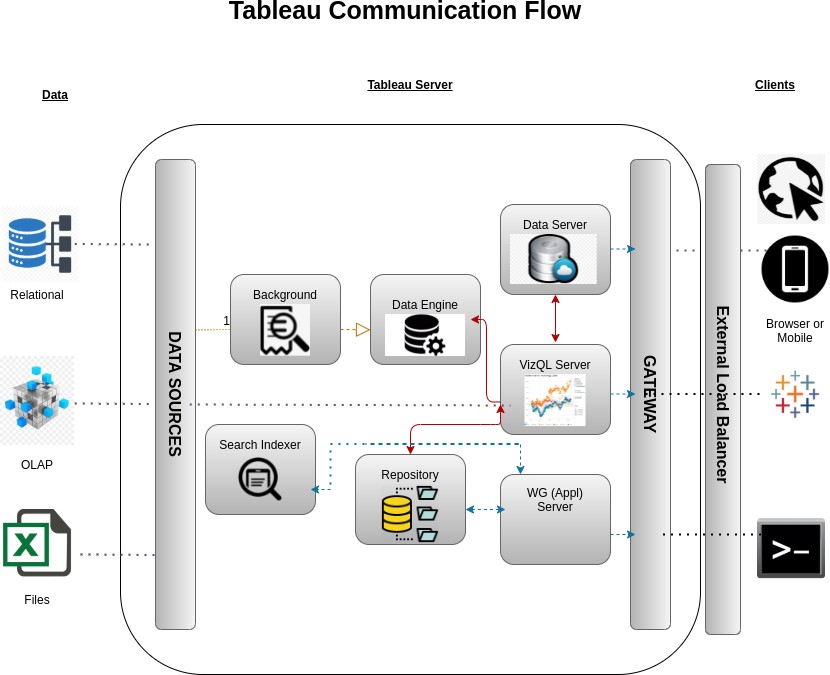
# Scope

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.

1. **Architecture**

****

**2.1 Tableau Server Architecture**



**1). Gateway/Load Balancer**

It acts as an Entry gate to the Tableau Server and also balances the load to the Server if multiple Processes are configured.

**2) Application Server:**

Application Server processes (wgserver.exe) handle browsing and permissions for the Tableau Server web and mobile interfaces. When a user opens a view in a client device, that user starts a session on Tableau Server. This means that an Application Server thread starts and checks the permissions for that user and that view.

1. **Repository:**

Tableau Server Repository is a PostgreSQL database that stores server data. This data includes information about Tableau Server users, groups and group assignments, permissions, projects, data sources, and extract metadata and refresh information.

1. **VIZQL Server:**

Once a view is opened, the client sends a request to the VizQL process (vizqlserver.exe). The VizQL process then sends queries directly to the data source, returning a result set that is rendered as images and presented to the user. Each VizQL Server has its own cache that can be shared across multiple users

1. **Data Engine:**

It Stores data extracts and answers queries.

1. **Data Server:**

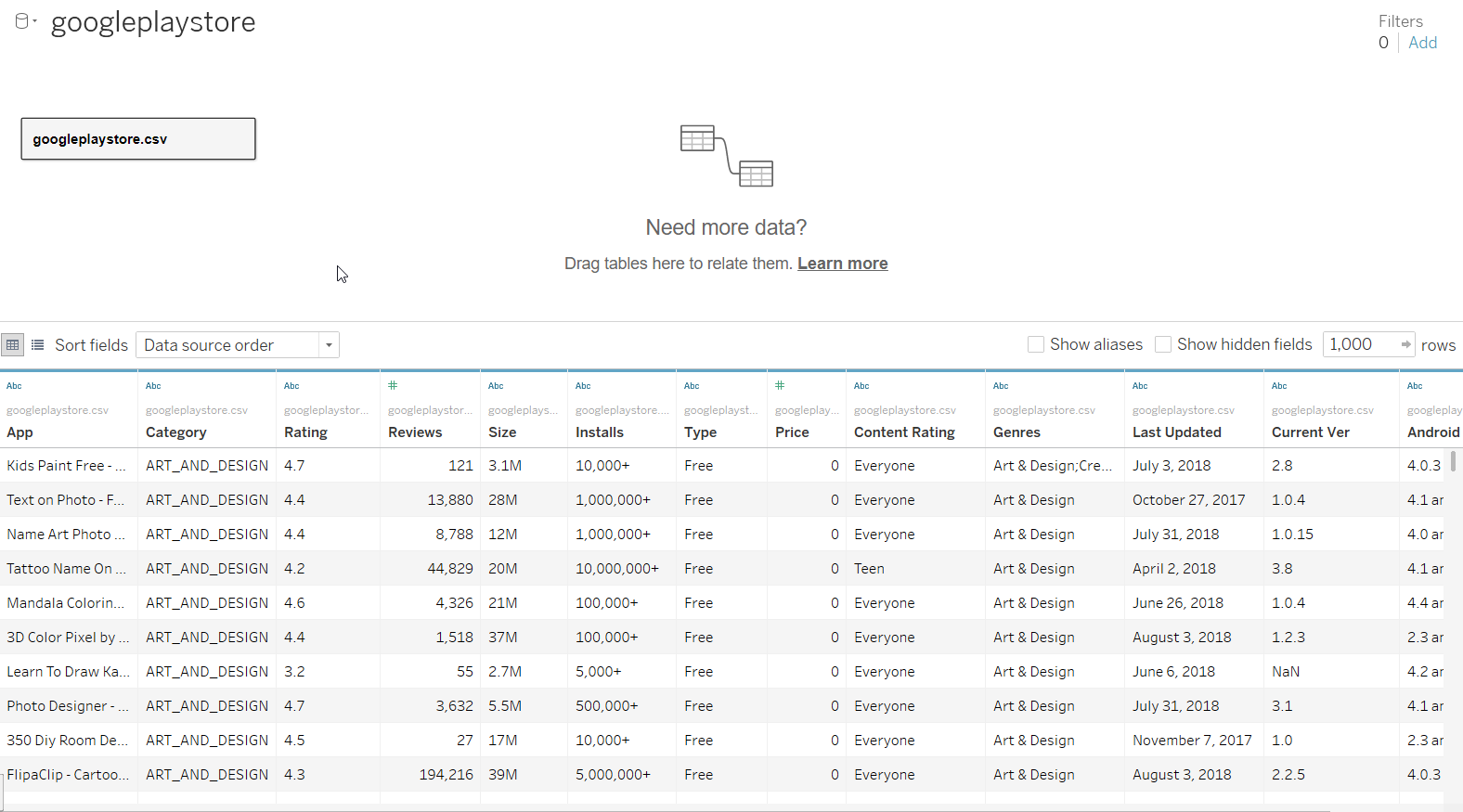
Data Server Manages connections to Tableau Server data sources

It also maintains metadata from Tableau Desktop, such as calculations, definitions, and groups.

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

The Google Play Store Contains following columns.

* + - App Name: It describes the name of the app.
    - Category: It describes different category of the app.
    - Rating: Rating for the app .
    - Size: Describes the size of the app.
    - Type: Describes whether the app is free or paid app.
    - Price: Describes the price of the app.
    - Last Updated: Describes when the app is last updated



* 1. **Web Scraping:**

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.

* 1. **Data Transformation:**

In the Data Transformation Process, we will convert our original datasets with other necessary attributes format. For the given Data Set names of the Columns have been changed and Null Values have been removed from the Data Set.

# Creating relations between Parameters.

# In this Project we had used many types of Visualizations like:

# Bar Charts

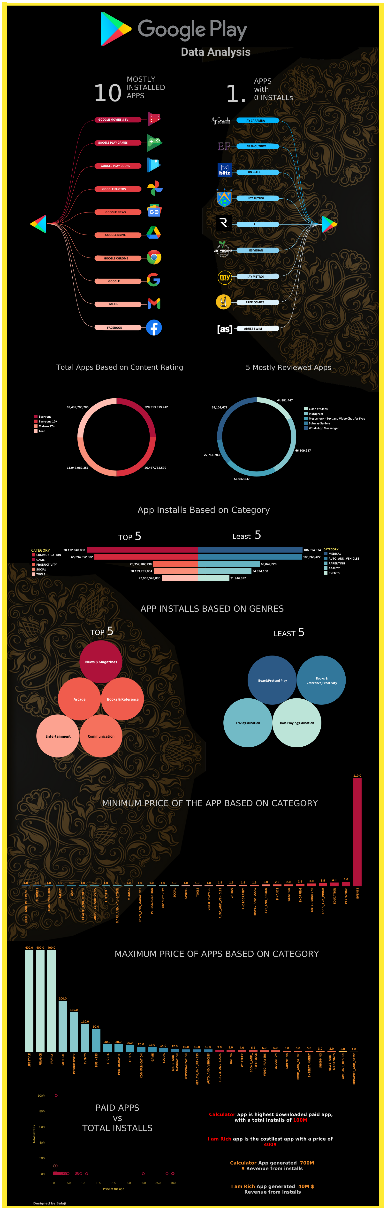
# Pie Charts

# Horizontal Bar Charts

# Count Chart

# Butterfly Charts

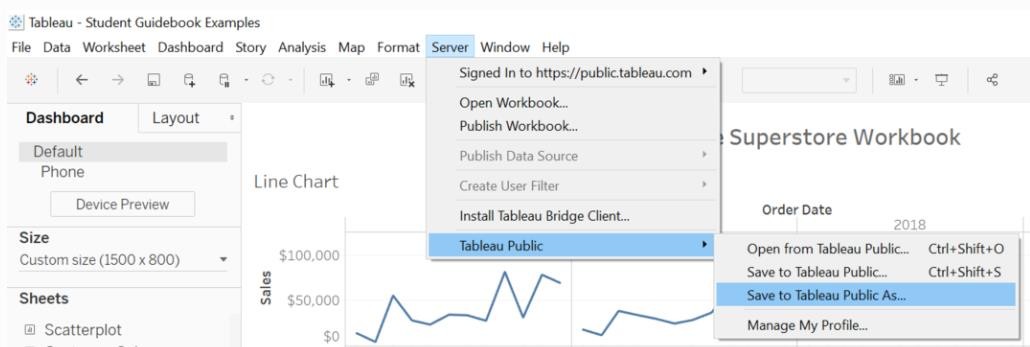
With the use of all the available parameters we had plotted visualizations.



* 1. **Deployment**

Once you’ve completed your dashboard, follow these steps: **Server, Tableau Public, Save to Tableau Public As**

You may be prompted to log into your Tableau Public profile first if this is your first time publishing.



.

1. **Unit Test Cases**

|  |  |  |
| --- | --- | --- |
| S. No | Test Case Description | Expected Results |
| 1 | Total number of apps based on category | Count of apps are shown by count plot based on category. |
| 2 | Top 10 apps with highest installs | Sorting the data based on total installs in descending order gives the highest installs. |
| 3 | 10 hotels with least installs | Sorting the data based on installs in ascending order gives the apps with least installs. |
| 4 | Paid app vs Total installs | Scatter plot is drawn for paid apps and total installs |
| 5 | Mostly installed paid apps and less installed paid apps. | By filtering data & presenting in bar chart we get the required data |