

# High-Level Design (HLD)

## Analyzing Google Apps Store dataset in terms of App downloads and Rating

RAHUL PANWAR

### Document Version Control

Date Issued	Version	Description	Author
10/01/2023	1.0	Abstract, Introduction, General Description	Rahul Panwar
10/01/2023	1.1	Design Detail, KPI, Deployment	Rahul Panwar

## Contents

Document Version Control .....	2
Abstract .....	3
1. Introduction .....	4
1.1 Why this High-Level Design Document? .....	4
1.2 Scope .....	4
2. General Description .....	5
2.1 Product Perspective & Problem Statement .....	5
2.2 Tools Used .....	5
3. Design Detail .....	6
3.1 Functional Architecture .....	6
3.2 Optimization .....	7
4. KPI .....	8
4.1 KPIs (Key Performance Indicators) .....	8
5. Deployment .....	8

## **ABSTRACT**

Organizations under the Technical industry seek to attain core competence by creating and sustaining a unique process to collect personal information about users and their purchasing trends. The report critically evaluates how a service-based organization – Google apps store analysis information systems as a vibrant tool in attaining competitive advantage through efficient management and acquisition of information. As in today's market without any market analysis, it's very hard to predict how the business is running and how it will be in future. Many companies with proper analysis the market have shown better growth as they already know which item they must focus on, which product needs some improvement etc. Also, Competition is increasing day by day as many new companies are coming with better management systems and giving tough competition due to that it is now very important to have a proper analysis to run any business and to compete with these companies.

# **1 Introduction**

## **1.1 Why this High-Level Design Document?**

The purpose of this High-Level Design (HLD) document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions before coding and can be used as a reference manual for how the modules interact at a high level.

### **The HLD will:**

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project • List and describe the non-functional attributes like:
  - Security
  - Reliability
  - Maintainability
  - Portability
  - Reusability
  - Application compatibility
  - Resource utilization
  - Serviceability

## **1.2 Scope**

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

## 2 General Description

### 2.1 Product Perspective & Problem Statement

Technology is the increasing need nowadays and used everywhere. One of the features of Technology is android. Which we all use in our daily life. Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets

### 2.2 Tools used

Business Intelligence tools and libraries works such as NumPy, Pandas, Seaborn, Matplotlib, MS-Excel, MS-Power BI, Jupyter Notebook and Python Programming Language are used to build the whole framework.



## 3 Design Details

### 3.1 Functional Architecture

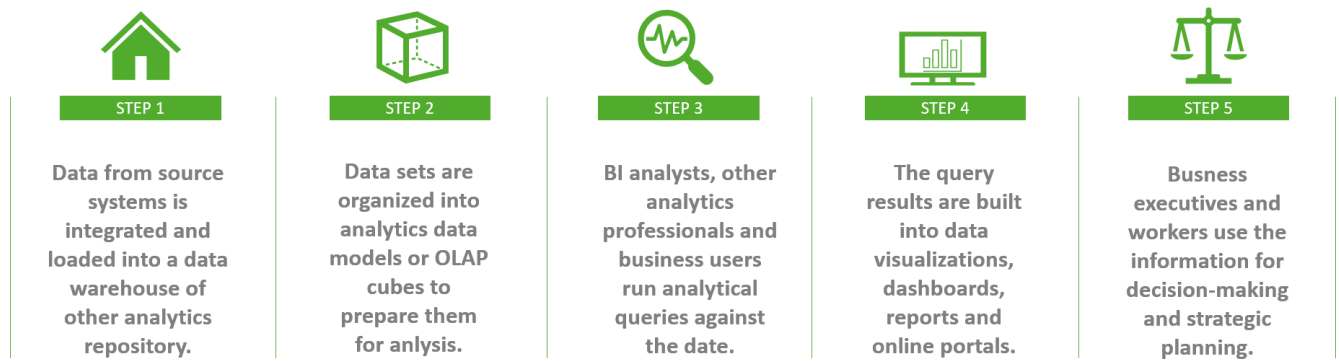
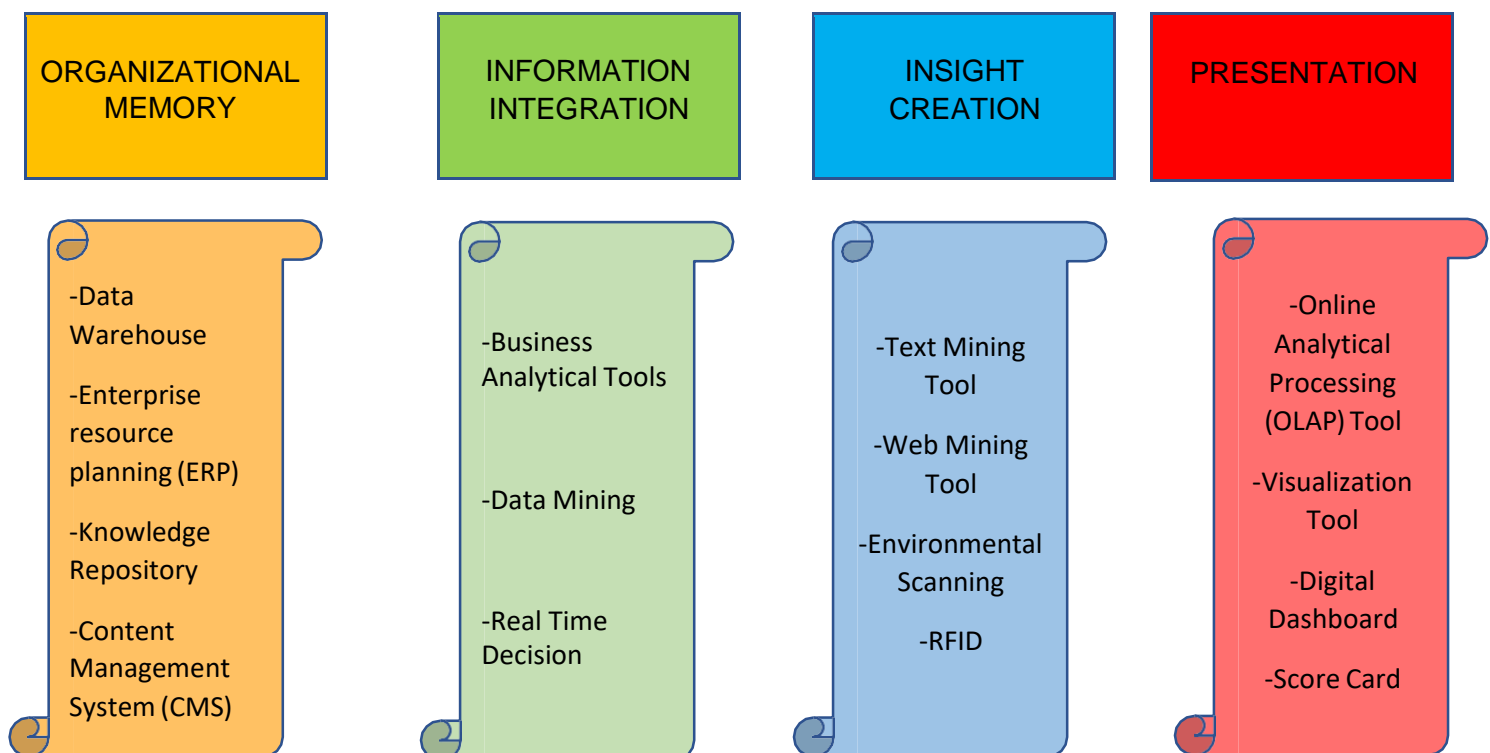


Figure 1: Functional Architecture of Business Intelligence

## How BI Works



### 3.2 Optimization

#### i. Your data strategy drives performance

- Minimize the number of fields□
- Minimize the number of records□
- Optimize extracts to speed up future queries by materializing calculations, removing columns and the use of accelerated views□

#### ii. Reduce the marks (data points) in your view

- Practice guided analytics. There's no need to fit everything you plan to show in a single view. Compile related views and connect them with action filters to travel from overview to highly granular views at the speed of thought.□
- Remove unneeded dimensions from the detail shelf.□
- Explore. Try displaying your data in different types of views.□

### iii. Limit your filters by number and type

- Reduce the number of filters in use. Excessive filters on a view will create a more complex query, which takes longer to return results. Double-check your filters and remove any that aren't necessary.□
- Use an include filter. Exclude filters load the entire domain of a dimension while including filters do not. An include filter runs much faster than an exclude filter, especially for dimensions with many members.□
- Use a continuous date filter. Continuous date filters (relative and range-of- date filters) can take advantage of the indexing properties in your database and are faster than discrete data filters.□
- Use Boolean or numeric filters. Computers process integers and Booleans (t/f) much faster than strings.□
- Use parameters and action filters. These reduce the query load (and work across data sources).□

### iv. Optimize and materialize your calculations

- Perform calculations in the database□
- Reduce the number of nested calculations.□
- Reduce the granularity of LOD or table calculations in the view. The more granular the calculation, the longer it takes.□
- ✓ LODs - Look at the number of unique dimension members in the calculation.
- ✓ Table Calculations - the more marks in the view, the longer it will take to calculate.
- Where possible, use MIN or MAX instead of AVG. AVG requires more processing than MIN or MAX. Often rows will be duplicated and display the same result with MIN, MAX, or AVG.
- Make groups with calculations. Like include filters, calculated groups load only named members of the domain, whereas Tableau's group function loads the entire domain.
- Use Booleans or numeric calculations instead of string calculations. Computers can process integers and Booleans (t/f) much faster than strings.

Boolean>Int>Float>Date>DateTime>String.

## 4 KPI

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the disease.

As and when the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors.

## 4.1 KPIs (Key Performance Indicators)

Key indicators displaying a summary of Apps Data and its relationships with different metrics.

- i. Yearly, Monthly Ups and Downs in install.
- ii. Content rating That Generated Highest revenue, install etc.
- iii. Top 5 category performance.

## 5 Deployment

Prioritizing data and analytics couldn't come at a better time. Your company, no matter what size, is already collecting data and most likely analyzing just a portion of it to solve business problems, gain competitive advantages, and drive enterprise transformation.

With the explosive growth of enterprise data, database technologies, and the high demand for analytical skills, today's most effective IT organizations have shifted their focus to enabling self-service by deploying and operating Power BI at scale, as well as organizing, orchestrating, and unifying disparate sources of data for business users and experts alike to author and consume content.

Power BI prioritizes choice in flexibility to fit, rather than dictate, your enterprise architecture. Power BI Desktop and Power BI Service leverage your existing technology investments and integrate them into your IT infrastructure to provide a self-service, modern analytics platform for your users. With on-premises, cloud, and hosted options, there is a version of Power BI to match your requirements.

