High Level Design (HLD)

Analysing Google Apps Store

Revision Number: 1.0

Last date of revision: 06/09/2021

Swati Pandey

[AUTHOR NAME]

# **Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| **06th  Sep 2021** | 1.0 | First Version of Complete HLD | Swati |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[AUTHOR NAME]

Contents

[**Document Version Control** 2](#_Toc81834898)

[**Abstract** 4](#_Toc81834899)

[**1)** **Introduction** 5](#_Toc81834900)

[**i)** **Why this High-Level Design Document?** 5](#_Toc81834901)

[**ii)** **Scope** 5](#_Toc81834902)

[**2)** **Product Perspective** 5](#_Toc81834903)

[**i)** **Problem Statement** 5](#_Toc81834904)

[**ii)** **Tools used** 5](#_Toc81834905)

[**3)** **Design Details** 7](#_Toc81834906)

[**i)** **Functional Architecture** 7](#_Toc81834907)

[**4)** **Optimization** 8](#_Toc81834908)

[**i)** **Data Optimization through Python/Tableau** 8](#_Toc81834909)

[**ii)** **Use of Context Filters in case of filtering huge percentage of data** 8](#_Toc81834910)

[**iii)** **Calculations:** 8](#_Toc81834911)

[**iv)** **Visualization and layout:** 8](#_Toc81834912)

[**v)** **Limit your filters by number and type** 8](#_Toc81834913)

[**vi)** **Optimize and materialize your calculations** 8](#_Toc81834914)

[**5)** **KPIs** 9](#_Toc81834915)

[**i)** **Dashboard:** 9](#_Toc81834916)

[**ii)** **KPIs (Key Performance Indicators)** 9](#_Toc81834917)

[**6)** **Deployment** 9](#_Toc81834918)

# 

# **Abstract**

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. Good research is very much important to get best out of any business model. This project holds the thorough research on the available apps based on certain metrics.

[AUTHOR NAME]

# **Introduction**

### **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 prior to 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:

o Security

o Reliability

o Maintainability

o Portability

o Reusability

o Application compatibility

o Resource utilization

o Serviceability

### **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.

# **Product Perspective**

## **Problem Statement**

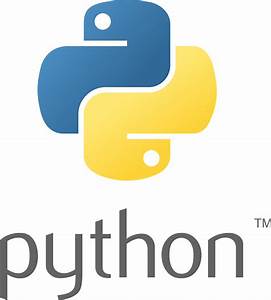
It is very important to understand how the apps have performed and what all areas of improvement can be looked upon to ensure best out of Google Play Store business model. Also, the research can help the administrators to take future decisions.

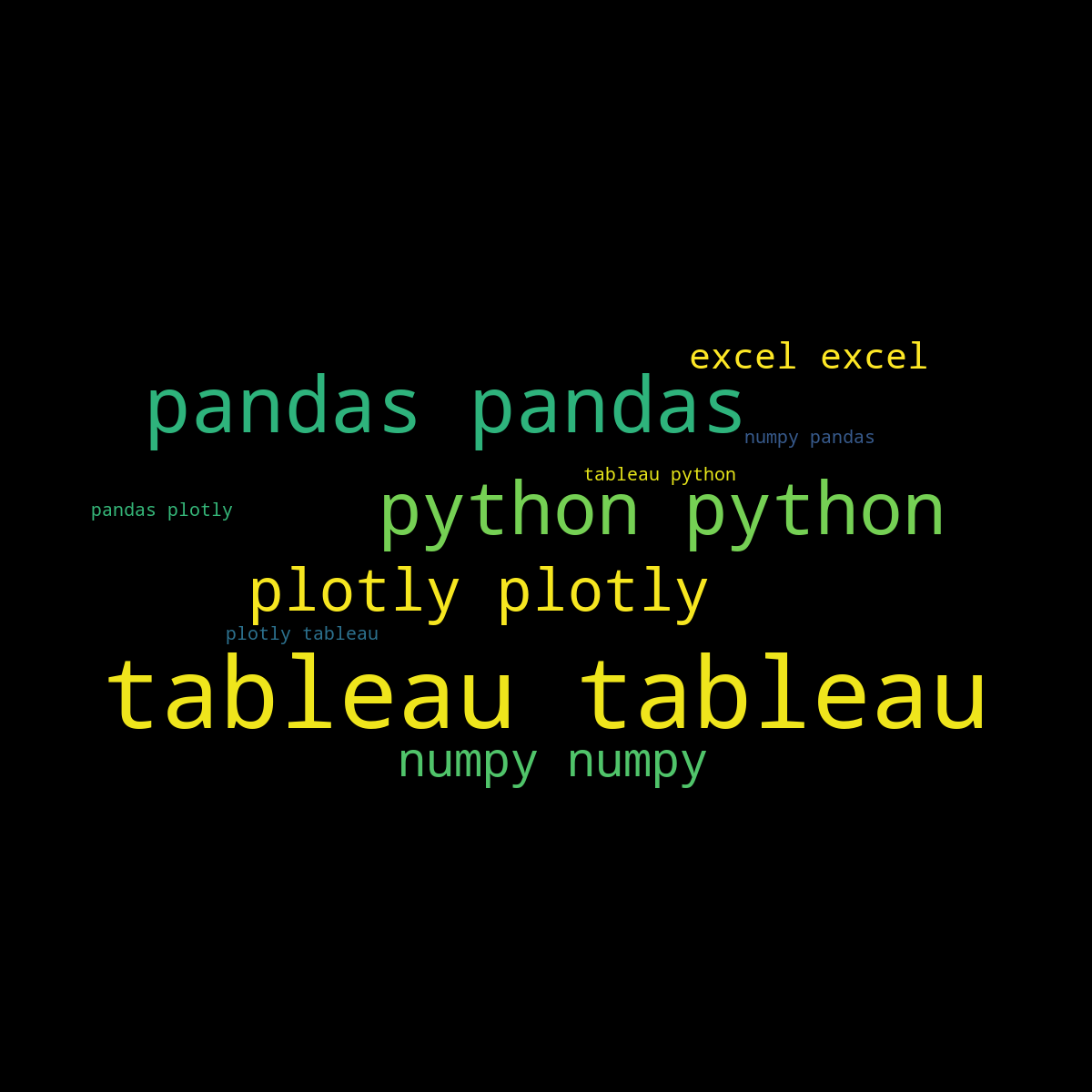
The objective of the project is to perform data visualization techniques to understand the insight of the data. This project aims apply various Business Intelligence tools such as Tableau or Power BI to get a visual understanding of the data.

## **Tools used**

Business Intelligence tools and libraries works such as NumPy, Pandas, Polly, Excel, Python, Tableau are used to build the whole framework.



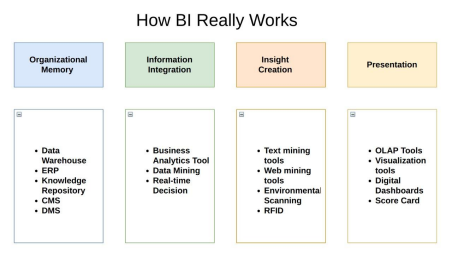




# **Design Details**

## **Functional Architecture**

Figure 1: Functional Architecture of Business Intelligence

[AUTHOR NAME]  

# **Optimization**

## **Data Optimization through Python/Tableau**

* Minimize the number of Columns which are not adding any information to the research to speed up future queries by materializing calculations.
* Reduce the number of records in extract by applying the filters

## **Use of Context Filters in case of filtering huge percentage of data**

* Context filters can be faster than extract. In this case, we still leverage the speed, power, and optimization of the database instead of relying on the client machine.
* There is also no need to re-extract to get the current data

## **Calculations:**

* If we have only if-else condition in calculations, it is recommended to use Boolean calculations.
* Use, Case function if we are using groups in visualization. This would return the same output but would be much faster as compared to groups.
* Use ELSEIF instead of ELSE-IF, as in case of else-if condition the if is treated as the second if statement whereas for ELSEIF treats it as part of the first statement.

## **Visualization and layout:**

* Selecting proper visualization is also one way of optimizing the dashboard performance.
* Excess worksheets on a dashboard can impact performance. Better to hide or delete unused sheets in the dashboard.

## **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 include 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 date 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).

## **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.
* 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.
* 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>Date Time>String

# **KPIs**

## **Dashboard:**

Dashboards will be implemented to display and indicate certain KPIs and relevant information w.r.t Business model of Google App Store.

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

## **KPIs (Key Performance Indicators)**

Key indicators displaying a summary of the Housing Price and its relationship with different metrics

1. Impact of Application Size on Rating
2. Impact of Application Size on Number of Installations
3. Impact of Type on Reviews, ratings and no of Installations
4. Impact of Application Size on Reviews
5. Impact of category on Reviews, ratings and no of Installations
6. Impact of Android Version on Reviews, ratings and no of Installations

# **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 analysing 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 Tableau at scale, as well as organizing, orchestrating, and unifying disparate sources of data for business users and experts alike to author and consume content. Tableau prioritizes choice in flexibility to fit, rather than dictate, your enterprise architecture.  Tableau Server and Tableau Online leverage your existing technology investments and integrate 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 Tableau to match your requirements. Below is a comparison of the three types:

TYPE PROS CONS

**Tableau Server - On Premises**

• Full control of hardware and software

• Infrastructure and data remain behind your firewall

• Need dedicated administrators to manage hardware and software

• Additional infrastructure needed to access off-network (mobile, external)

**Tableau Server - Public Cloud (IaaS)**

• Full control of software on managed hardware

• Puts infrastructure in same place as data (for migration to cloud)

• Flexibility to spin up/down hardware as needed

• Need dedicated administrators to manage software

• Additional infrastructure needed to access off-network (mobile, external)

**Tableau Online (SaaS)**

• Fully hosted solution (hardware, software upgrades)

• Fast to deploy

• Easy for external audience to access

• Single-site in multi-tenant environment

• Cubes are not supported

• No guest account access

Depending on your organizational roles and responsibilities, Tableau Server should be installed by a systems administrator and the designated Tableau Server Administrator in coordination with the appropriate IT roles. For Tableau Online, you will integrate with your existing technology and configure the site settings. The Data & Analytics Survey, completed by business teams, identifies and prioritizes data use cases, audience size, and users. You will use the information collected in both surveys to plan your deployment strategy, including sizing, installation, and configuration of your Tableau Server or integration and configuration of Tableau Online. In addition to installing Tableau Server or configuring Tableau Online, administrators will also need to plan for the client software installation of Tableau Prep Builder, Tableau Desktop, Tableau Mobile, and Tableau Bridge for Tableau Online where applicable.