High Level Design (HLD)

Heart Disease Diagnostic Analysis

Revision Number - 1.2
Last Date of Revision – 15/03/2023

Abeen Backer Palliyal

Document Version Control

Date Issued	Version	Description	Author
10/03/2023	1.0	Abstract, Introduction, General Description	Abeen Backer Palliyal
14/03/2023	1.1	Design Detail, KPI, Deployment	Abeen Backer Palliyal
15/03/2023	1.2	Final Revision	Abeen Backer Palliyal

Contents

Abstract

Heart disease is a term covering any disorder of the heart. Heart diseases have become a major concern to deal with as studies show that the number of deaths due to heart diseases have increased significantly over the past few decades in India it has become the leading cause of death in India. A study shows that from 1990 to 2016 the death rate due to heart diseases have increased around 34% from 155.7 to 209.1 deaths per 1 lakh population in India.

Thus, preventing heart diseases has become more than necessary. It is estimated that up to 90% of CVD (Cardio Vascular Disease) may be preventable. Good data-driven systems for predicting heart diseases can improve the entire research and prevention process, making sure that more people can live healthy lives. Detection of a CVD at an early stage leads to prevention of more than 80% of potential related deaths.

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

The goal of this project is to analyse to predict the probability of heart disease occurrence, based on a combination of features that describes the disease. Based on the prediction we develop a BI report depicting the features affecting the occurrence of the disease and the magnitude of the affect each feature has. To achieve the goal, we used a well-defined data set that is formed by taking into consideration some of the information of 303 individuals. The problem is based on the given information about each individual we have to calculate that whether that individual will suffer from heart disease or not.

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.

• Jupyter Notebook is used as IDE.

• Python is the Programming Language used.

• EDA is done using Numpy & Pandas.

• Visualizations were done using seaborn.
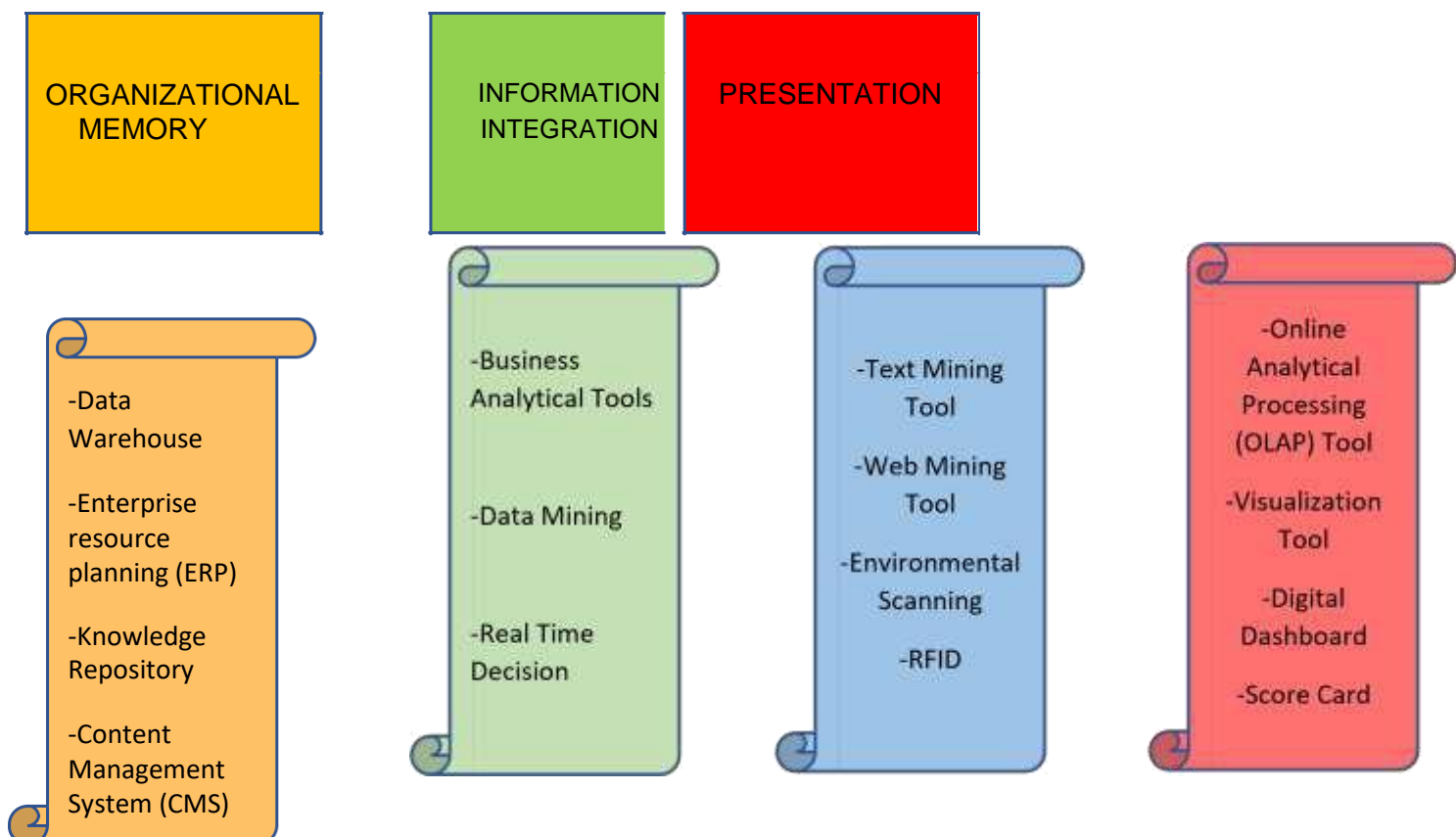
• Power BI is used for dashboard creation.

3 Design Details

3.1 Functional Architecture

Figure 1: Functional Architecture of Business Intelligence

How BI Works

Constraints

Our analysis is done based on a limited dataset provided for a specific (15) features affecting heart disease. The analysis does not take into account any external interventions like underlying disease, type of medication used, lifestyle patterns, BMI value etc.

Assumptions

It is a task that is trivially performed by doctors, however the use of past cases and the potential cause for such cases has enabled data analysts to create a segment of symptoms/causes belonging to which significantly increases a risk of a potential heart disease in the future.

Our analysis assumes that all the data provided was true without any corruption and the features mentioned in the raw dataset are the only driving factors of a potential heart disease.

.

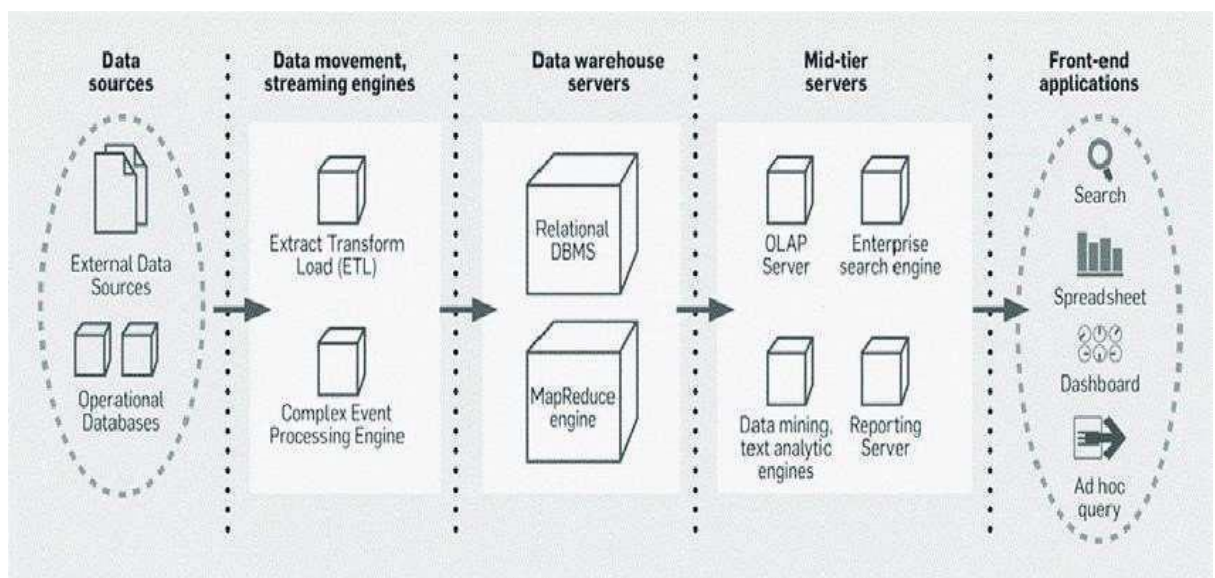# Design Details
Functional Architecture

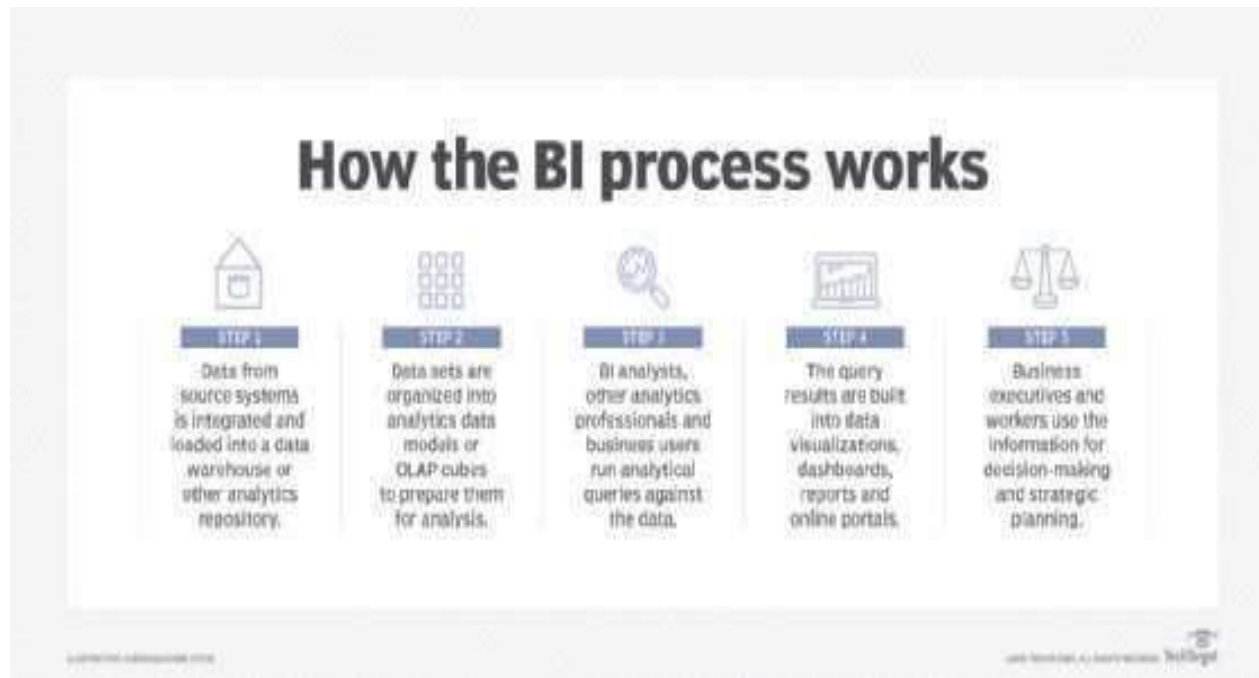Figure 1: Functional Architecture of Business Intelligence

Figure 2: Working of BI process

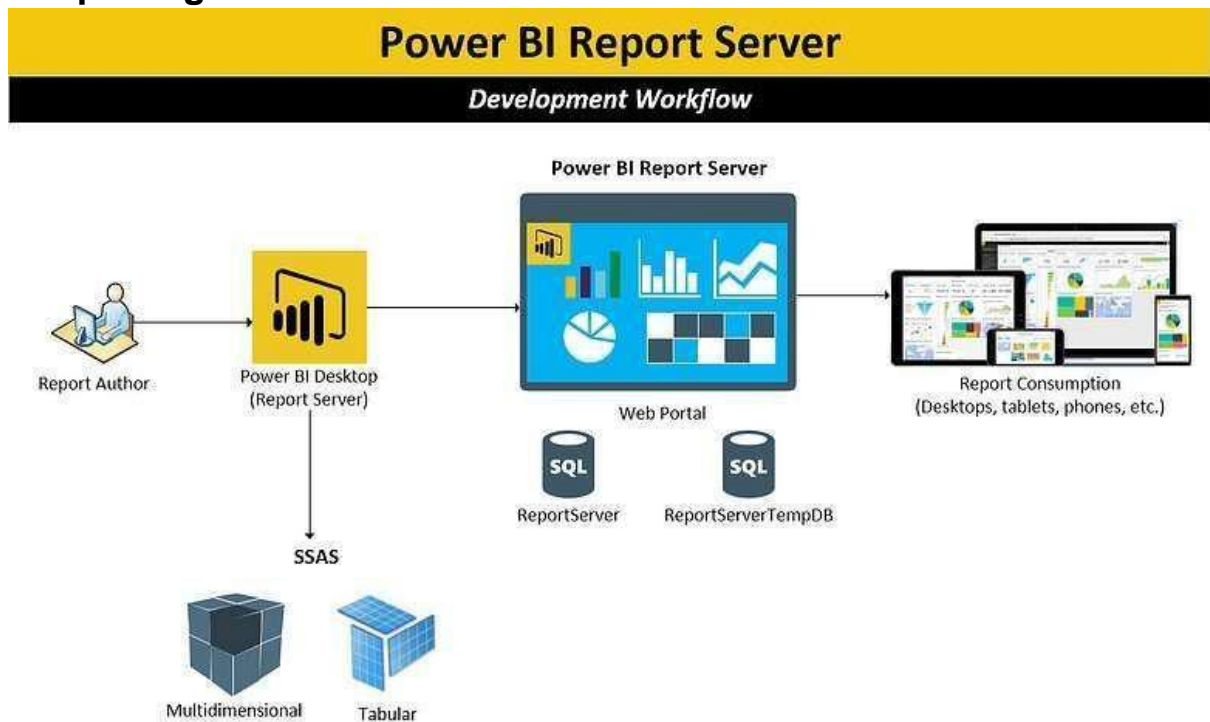## BI Reporting Architecture

Figure 2: BI Reporting architecture

3.2 Optimization

1. 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

2. 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.

3. 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-ofdate 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).

Performance

Healthcare analytics determines the presence or absence of a life threatening condition, it should be as accurate as possible. So that it will not mislead the user. Also, model retraining is very important to improve the performance.

Security

Since the Health care analysis consists of patient's data, the information should be secured.

Reusability

The code written and the components used should have the ability to be reused with no problems.

Resource utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

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 the Heart Disease Analysis and its relationship with different metrics

1. Percentage of People Having Heart Disease
2. Age Distribution including Gender
3. Gender Distribution Based on Heart Disease
4. Chest Pain Experienced by People Suffering from Heart Disease with different type of age category
5. Blood Pressure, Cholesterol Level and Maximum Heart Rate of People with Heart Disease Patients.
6. ST Depression Experienced by People According to their age and heart disease.
7. ECG measurement based on heart disease patients

5 Deployment

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.

The Dashboard is published on Power BI service and an auto-refresh mode has been set so that the dashboard keeps on updating as the real time data loads into the log file.

The Dashboard showcases the multiple insights that has been drawn from the log files as follows:

1. **Summary:** The dashboard contains the statistical summary of the dataset and the visual displaying the effect of different feature to heart disease.

2. **Critical Factors:** This dashboard picks out the most important feature resulting in the presence of a heart disease and the magnitude at which it affects the presence.

3. **Patient Profile:** This dashboard shows the profile of a patient that has a disease displaying the average age, BP level e.t.c.