High Level Design (LLD)

Heart Disease Diagnostic Analysis

Revision Number – 1.2

Last Date of Revision – 2nd May, 2022

Taniya Nath

Document Control

Date	Version	Description	Author
26/04/22	1.0	Abstract, Introduction and General Description	Taniya Nath
29/04/22	1.1	Design Detail, KPI and Deployment	Taniya Nath
02/05/22	1.2	Final Revision	Taniya Nath

Contents

Abstract

Heart disease is a general term that means that the heart is not working normally. This is a type of disease that affects the heart or blood vessels. Heart disease have become a major concern to deal with as studies show that the number of deaths due to heart disease have increased in India. We all have realised the brute effects of Covid-19 on all irrespective of any status. Early diagnosis of heart defects are very important for medical treatment.

Thus, preventing heart diseases has become more than necessary. 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

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 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

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 describe the disease. To achieve the goal, we used a dataset 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.

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

2.2 Tools Used

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

3. Design Details

3.1 Functional Architecture

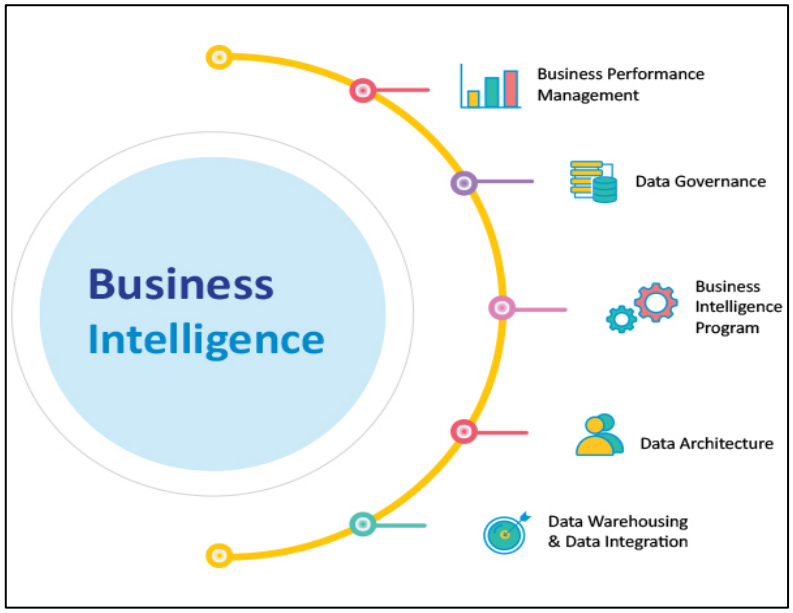
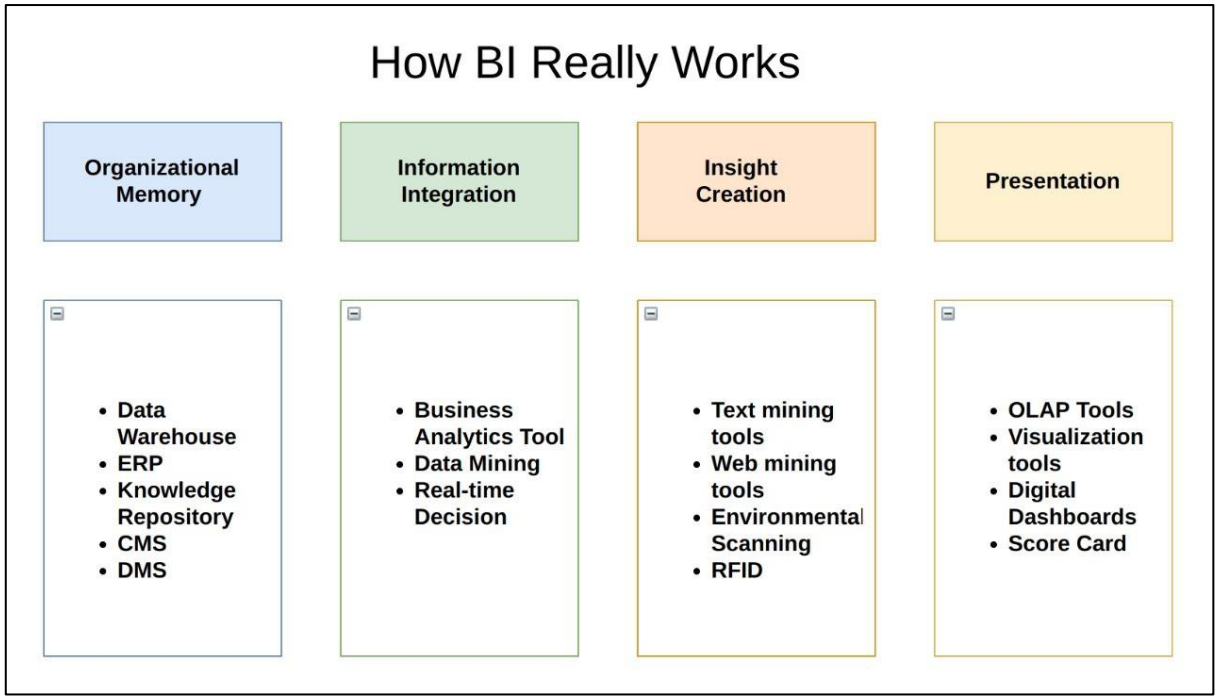


Figure 1: Functional Architecture of Business Intelligence



3.2 Optimization

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

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.

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 data 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.
    - O LODs - Look at the number of unique dimension members in the calculation.
    - O 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. KPIs (Key Performance Indicator)

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 Indicator)

Key indicators displaying a summary of the Heart Disease Diagnostic and its relationship with different metrics :

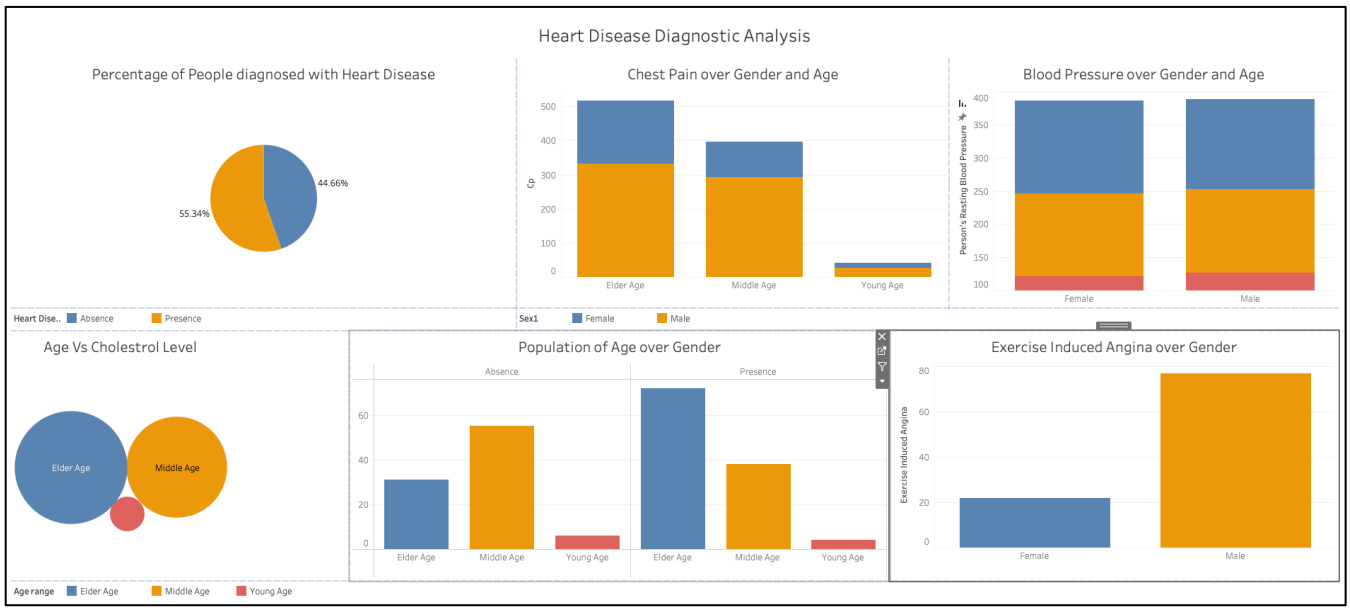
1. Percentage of Gender diagnosed with Heart Disease
2. Percentage of total population over the Heart Disease
3. Chest pain based on Gender
4. Age Category over the Population
5. Population of Age over Gender
6. Blood Pressure Vs Gender
7. Blood Pressure Vs Age
8. Cholesterol Vs Age
9. Cholesterol Vs Gender
10. Fasting Blood Sugar Vs Gender
11. Resting electrocardiographic results Vs Age
12. Thalassemia Vs Number
13. Exercise Induced Angina VS Gender
14. Population Vs Ca (number of major vessels)
15. ST Depression VS Age

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 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.

TableauURL:
https://public.tableau.com/app/profile/taniya.nath/viz/HeartDiseaseDiagnosticAnalysis_16515004907410/Dashboard1



6. Conclusion

This concludes the *High Level Documentation* of the **Heart Disease Diagnostic Analysis**.