High Level Design (HLD)

Heart Disease Diagnostic Analysis

Revision Number - 1.0
Last Date of Revision - 21/10/2024

MAINAK MUKHERJEE

Document Version Control

Date Issued	Version	Description	Author
14/10/2024	1.0	First version of Complete HLD	Mainak Mukherjee

Contents

Abstract

Heart disease refers specifically to conditions that involve the heart and its structure or function. Heart disease is one of the leading causes of death in the world, It has become a great challenge to health. Heart disease mortality has escalated over the past couple of decades manifesting a plethora of cases. That means that better prevention is desperately needed. This is where data-grounded analysis and insights can step in to pinpoint the contributing factors, offer early warning signs of risk factor burden, as well as inform public health initiatives.

Through the use of data analytics, healthcare providers can manage and relieve much of that impact to keep a healthier population living longer.

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 mild-technical terms which should be understandable to the administrators of the system.

2. General Description:

2.1 Product Perspective and Problem Statement:

The purpose of this study is to analyze the data to predict the possibility of developing heart disease based on a variety of risk factors. The dataset used consists of 303 individuals, each of whom is described by many characteristics such as age, cholesterol levels, and other heart-related measurements. The goal is to use these characteristics to examine whether or not a person will acquire cardiovascular disease. This investigation seeks to give significant insights that can help healthcare practitioners identify high-risk individuals and implement preventative strategies, thereby improving patient outcomes and lowering the total effect of heart disease.

2.2 Tools used:

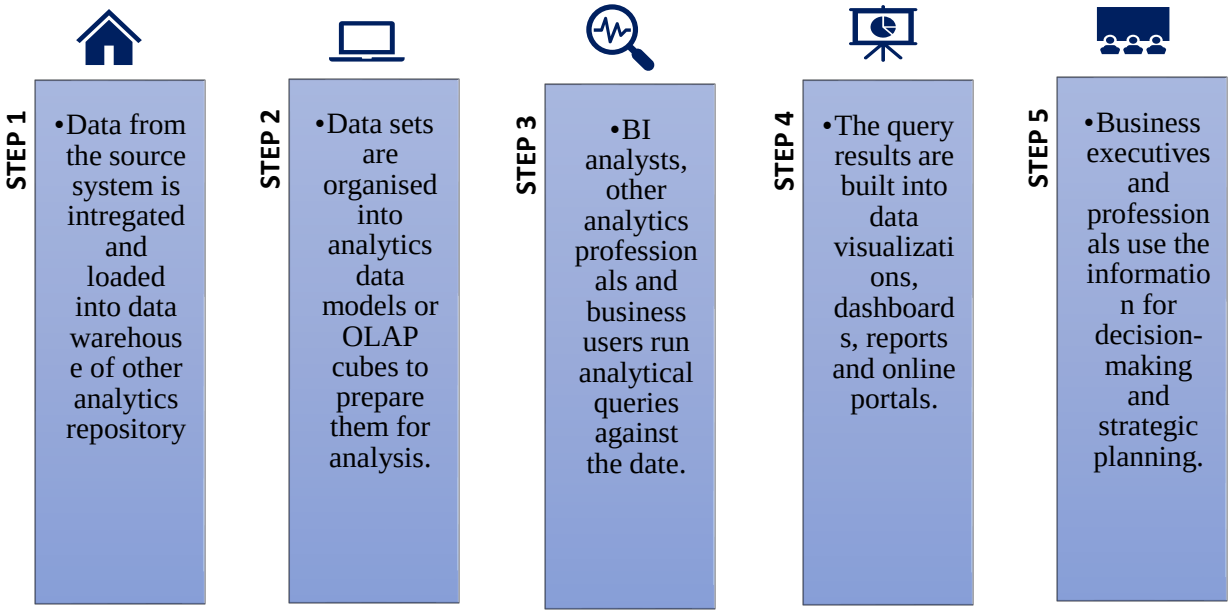
This project used a variety of strong tools and libraries for data processing and visualization.

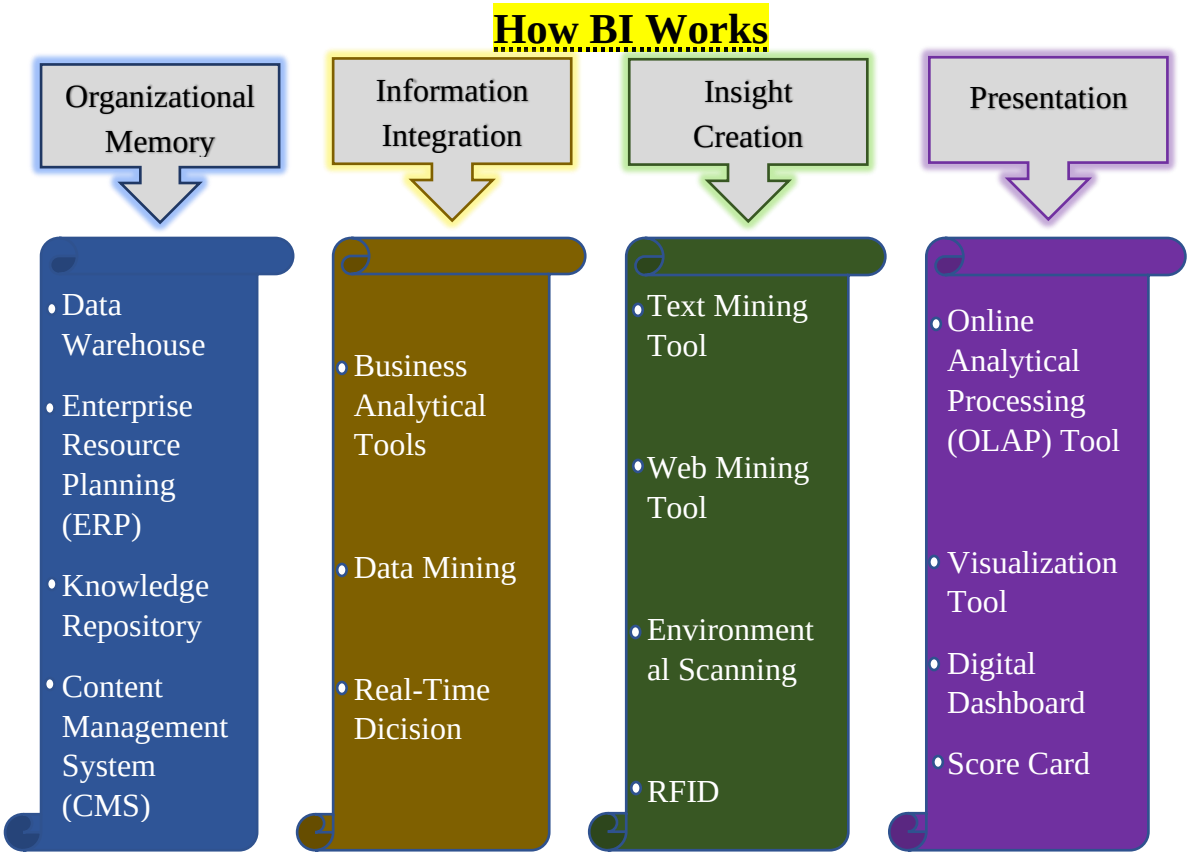
- **NumPy** and **Pandas** are used as the essential tools for efficient data manipulation.
- **Seaborn** and **Matplotlib** are used for comprehensive data visualization.
- **Python** programming language is used for developing and executing the analytical framework.
- Business intelligence tools such as **MS Excel** is used for basic understanding of the datased and **MS Power BI** is used to construct intuitive and dynamic dashboards.
- The entire project was created and conducted using **Jupyter Notebook**, which provides an interactive environment for seamless analysis.

Together, these technologies are used to form a strong foundation for extracting valuable insights from data.

3. Design Details:

3.1 Functional Architecture:





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 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.
    - 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. KPIs:

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)

A look at the primary indicators of heart disease and their relationships with other relevant factors.

- Percentage of individuals diagnosed with heart disease.
- Average age of heart disease patients.
- Proportion of individuals with and without heart disease.
- Gender-specific impact of heart disease.
- Impact of heart disease across different age groups.
- Distribution of chest pain types in patients with and without heart disease.
- Gender wise age distribution
- ST Depression: heart disease vs. healthy individuals controls by age.
- Blood pressure, Cholestrol and Maximum heart rate achieved by age.
- Blood pressure, Cholestrol and Maximum heart rate achieved by people with and without heart disease.

5. Deployment:

Businesses of all sizes can benefit greatly from data and analytics. Most businesses are already collecting and analyzing data to address challenges, gain a competitive advantage, or drive organizational transformation. As data volumes and technology options expand, effective IT teams are increasingly adopting self-service tools like Power BI to empower a wider range of users. Power BI facilitates data collaboration among both business users and traditional BI experts.

Power BI is a flexible and adaptable tool that can be integrated into existing IT infrastructures. It offers deployment options including on-premises, cloud-based, and hosted, allowing businesses to choose the best fit for their needs. By leveraging existing technology investments, Power BI provides a modern analytics platform that empowers users to explore and analyze data effectively.

Power BI Dashboard