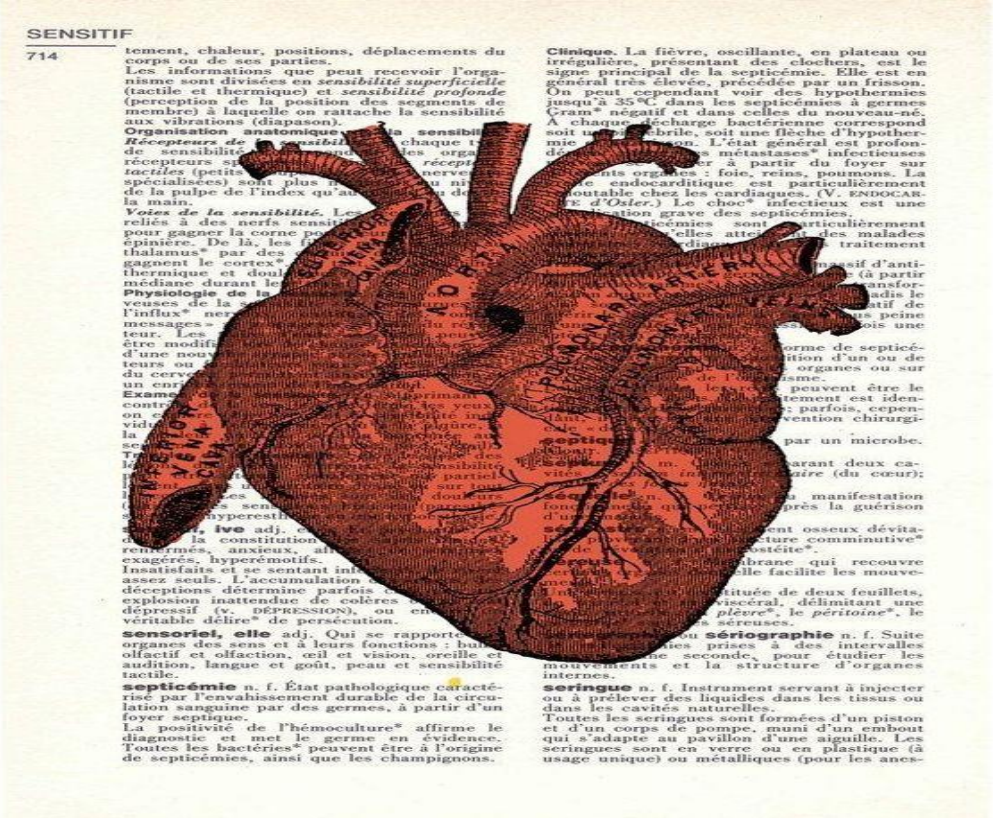
High Level Design (HLD)

Heart Disease Diagnostic Analysis



Revision Number - 1.1

Last Date of Revision - 22/06/2024

SAI MANEESHA GUDIMETLA

Document Version Control:

Page No	Date Issued	Description	Author
3-5	18/06/2024	Abstract, Introduction, General Description	Sai Maneesha Gudimetla
6-9	20/06/2024	Design Detail, KPI, Deployment	Sai Maneesha Gudimetla
1-9	22/06/2024	Final Revision	Sai Maneesha Gudimetla

Abstract:

Cardiac disease is a term that encompasses any disease of the heart. Cardiac disease has become a major issue in India with studies showing that the number of deaths due to cardiac disease has increased significantly in the past few decades. It is the leading cause of death in India. Studies show that the death rate due to cardiac disease has increased by about 34% from 155.7 to 209.1 per 100,000 population in India between 1990 and 2016.

Hence, prevention of cardiac disease has become more important than it needs to be. A good data-driven cardiac disease prediction system can improve the entire process of research and prevention and enable more people to lead a healthy life.

1 Introduction

1.1 Why use 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. To achieve the goal, we used a 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.

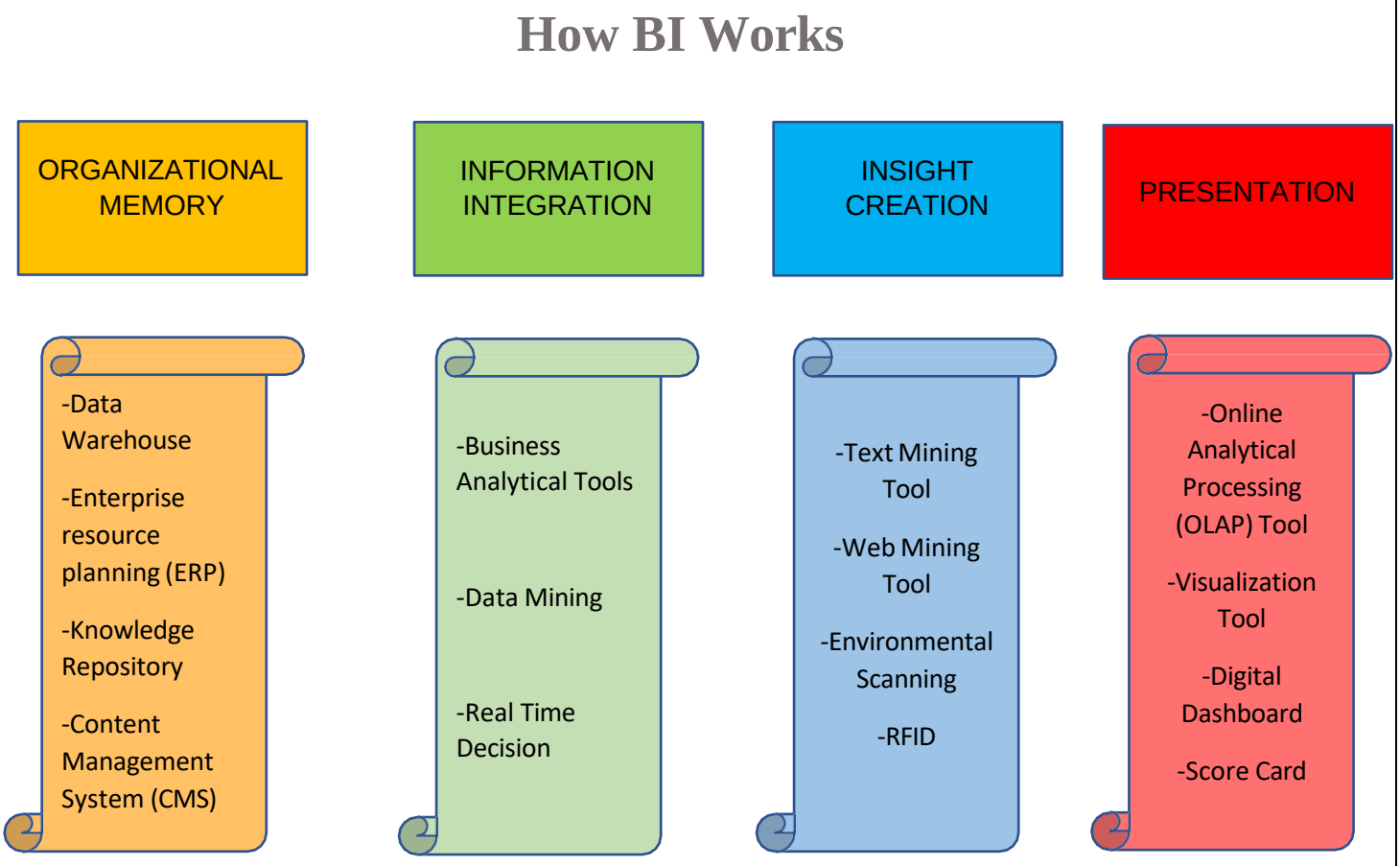
2.2 Tools used

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.

3 Design Details

3.1 Functional Architecture

Figure 1: Functional Architecture of Business Intelligence



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-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).

4. 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 the Housing Price 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
5. Blood Pressure, Cholesterol Level and Maximum Heart Rate of People According to their Age and Heart Disease Patients.
6. ST Depression Experienced by People According to their age and heart disease.

5 Deployment

There has never been a better time to prioritize data and analytics. Your company, regardless of size, is likely already collecting data and analyzing only a portion of it to solve business problems, gain competitive advantage, and drive business transformation. Given the explosive growth of enterprise data, database technologies, and high demand for analytical capabilities, today's most efficient IT organizations are shifting their focus to deploy and operate Power BI at scale to organize, orchestrate, and integrate disparate data sources to enable self-service. Business users and experts alike can create and consume content.

Power BI prioritizes choice and flexibility for customization rather than dictating enterprise architecture. Power BI Desktop and Power BI Service leverage existing technology investments and integrate into your IT infrastructure to provide users with a modern, self-service analytics platform. With on-premise, cloud and hosted options, there's a version of Power BI to fit your needs.