High Level Design (HLD)

Heart Disease Diagnostics Analysis

23 Sept 2023

Sanskar Baluni

Document Version Control

Date Issued	Version	Description	Author
12/09/2022	1.0	Abstract, Introduction, Description	Sanskar
13/09/2022	1.1	Design Detail, KPI, Deployment	Sanskar
21/07/2022	1.2	Final Revision	Sanskar

TABLE OF CONTENTS | PAGE NO.

Abstract

Any abnormality of the heart is referred to as heart disease. Studies show that during the past few decades in India, deaths from heart disease have climbed dramatically, making it the country's leading cause of death. As a result, heart disease has grown to be a serious problem to be dealt with. According to a study, India's death rate from heart disease increased by about 34% between 1990 and 2016, from 155.7 to 209.1 fatalities per 1 lakh people.

Therefore, it is now more important than ever to prevent heart disease. In order to ensure that more individuals may live healthy lives, effective data-driven methods for predicting heart disease can enhance the entire research and prevention process.

1.0 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.

2.0 Description

2.1 Product Perspective & Problem Statement

The goal of this project is 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.
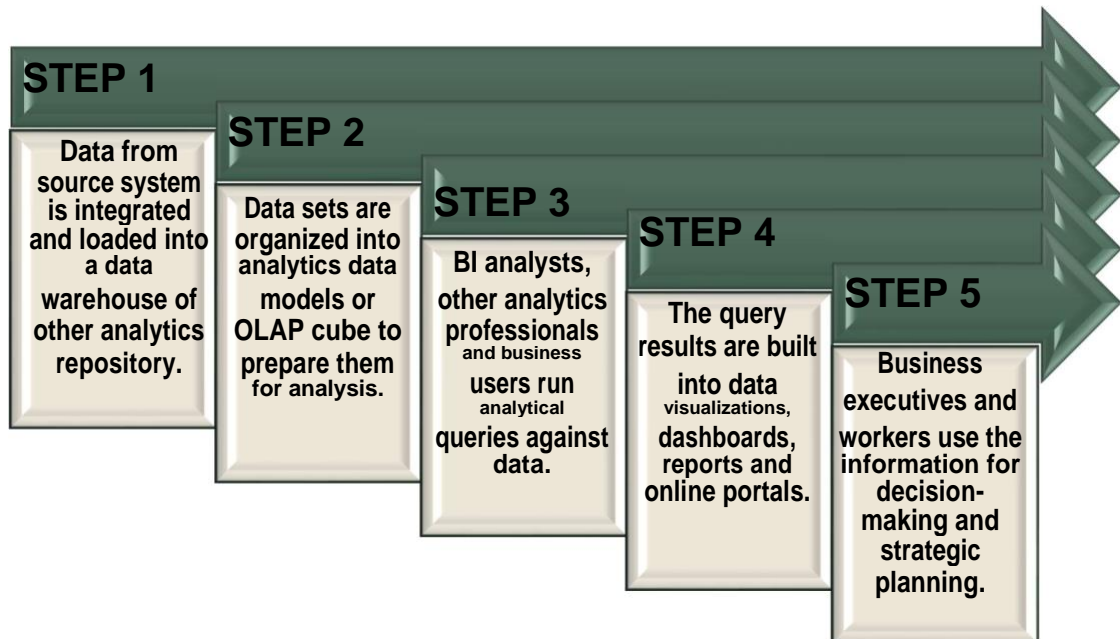
2.2 Tools used

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

3.0 Design Details

3.1 Functional Architecture

STEP 1

Data from source system is integrated and loaded into a data warehouse of other analytics repository.

STEP 2

Data sets are organized into analytics data models or OLAP cube to prepare them for analysis.

STEP 3

BI analysts, other analytics professionals and business users run analytical queries against data.

STEP 4

The query results are built into data visualizations, dashboards, reports and online portals.

STEP 5

Business executives and workers use the information for decision-making and strategic planning.

3.2 Optimization

3.2.1 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.

3.2.2 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.

3.2.3 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).

3.2.4 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.

4.0 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. **Heart Disease based on Age and 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.**

5.0 Deployment
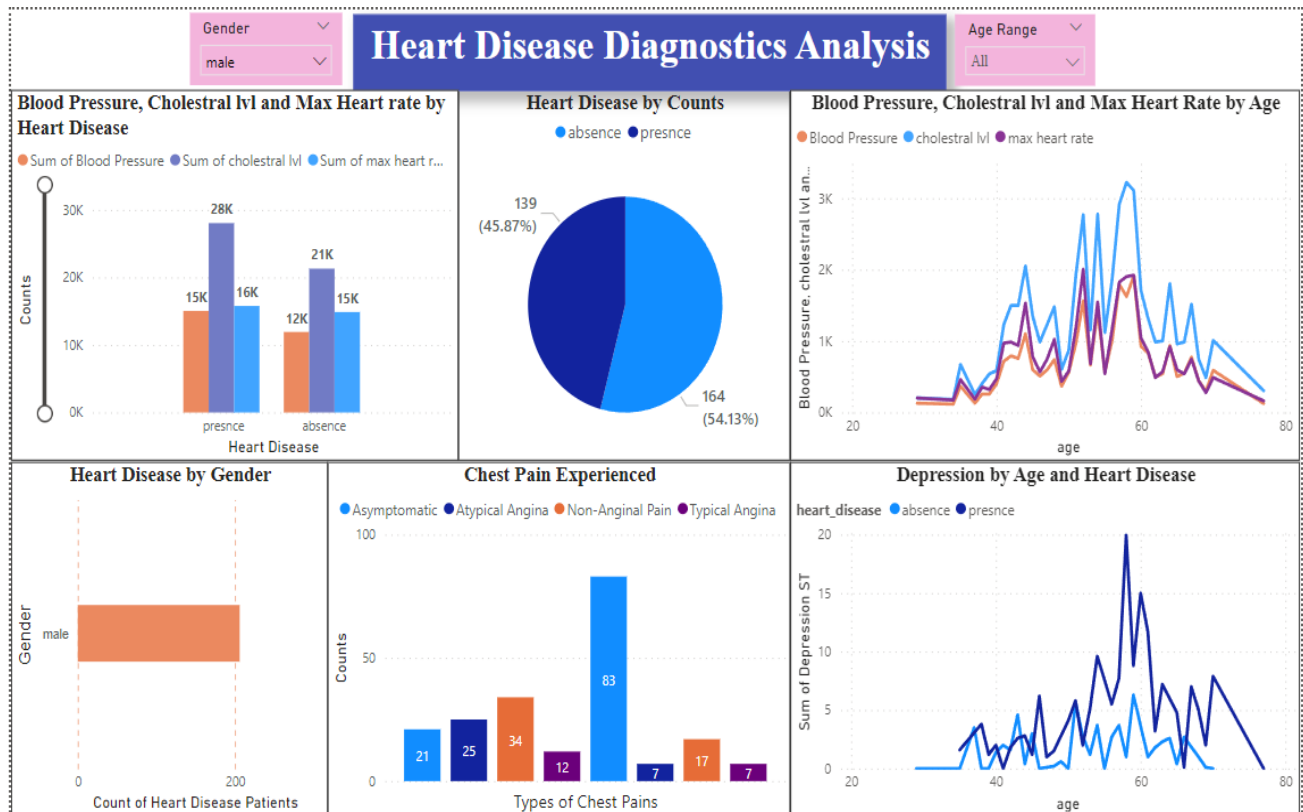
Prioritizing data and analytics couldn't come at a better time. Your company, no matter what size, is already collecting data and most likely analysing just a portion of it to solve business problems, gain competitive advantages, and drive enterprise transformation. With the explosive growth of enterprise 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 Power BI at scale, as well as organizing, orchestrating, and unifying disparate sources of data for business users and experts alike to author and consume content.

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.