# High Level Design (HLD)

# Heart Disease Diagnostic Analysis

**Revision Number - 1.0**

**Last Date of Revision - 09/08/2024**

**AKHIL K N**

**TARISHA SHARMA**

# Document Version Control

| Date Issued | Version | Description | Author |
|---|---|---|---|
| 09/08/2024 | 1.0 | Abstract, Introduction, General Description, Design Detail, KPI, Deployment | Akhil Tarisha |
| | | | |
| | | | |

# Contents

# Abstract

Heart disease is a leading cause of mortality in India, contributing significantly to the country's health burden. The prevalence of cardiovascular diseases (CVDs) has risen sharply due to lifestyle changes, urbanization, and aging populations. Key risk factors include hypertension, diabetes, high cholesterol, smoking, obesity, and physical inactivity. Rapid economic growth and associated lifestyle shifts have exacerbated these issues. Access to healthcare varies widely across urban and rural areas, often limiting timely diagnosis and treatment.

Public health initiatives aim to address these challenges through awareness campaigns, improved healthcare infrastructure, and preventive measures. Despite these efforts, the burden of heart disease remains high, requiring continued focus on both prevention and treatment strategies.

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 High-Level Design (HLD) Document aims to provide comprehensive details to the current project description, forming a suitable foundation for the coding phase. It is also intended to identify potential contradictions before coding begins and serve as a reference for how modules interact at a high level.

**The HLD will:**

- Outline all design aspects and provide detailed definitions
- Describe the user interface being developed
- Detail hardware and software interfaces
- Specify performance requirements
- Include design features and project architecture
- Enumerate and describe non-functional attributes such as:

  -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 objective of this project is to predict the likelihood of heart disease occurrence by analyzing a set of features related to the condition. To accomplish this, we utilized a dataset comprising information from 303 individuals. The task is to determine, based on the provided details about each person, whether they are at risk of developing heart disease. Additionally, we aim to identify other factors that may influence the individual's overall health.

## 2.2 Tools used

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

# 3 Design Details

## 3.1 Functional Architecture

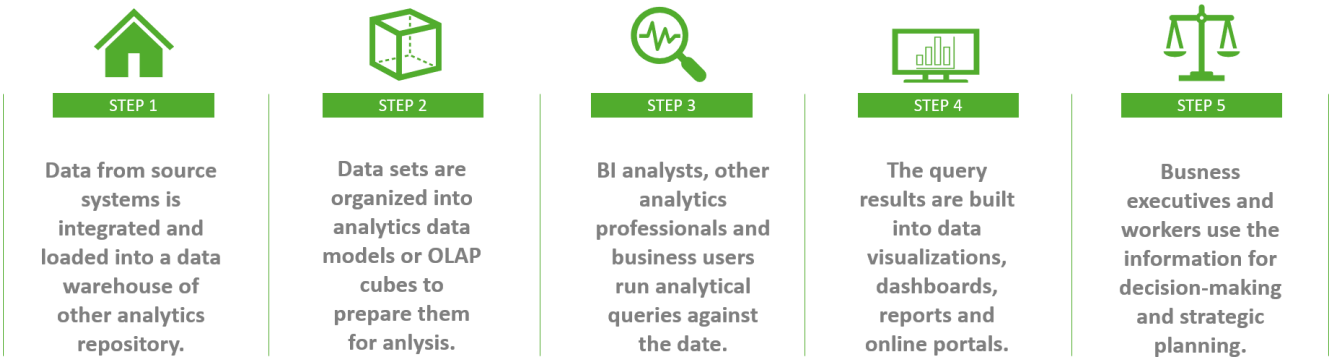
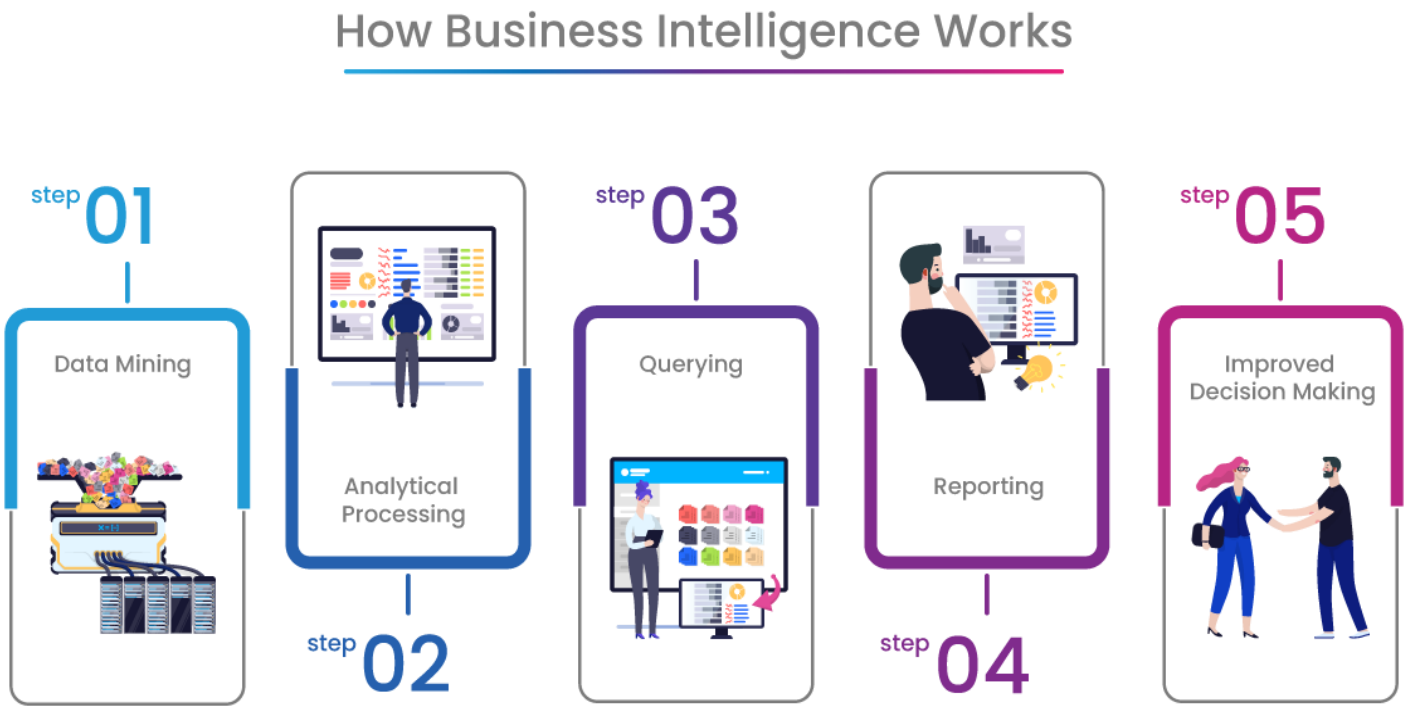



Figure 1: Functional Architecture of Business Intelligence



## 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
- Remove Blanks, Null values

### 2. Limit your filters by number and type

- Minimize the number of filters used: Having too many filters in a view can complicate the query and slow down result retrieval. Review your filters carefully and eliminate any that are not essential.

- Include filters are more efficient than exclude filters because they do not require loading the entire domain of a dimension. This makes include filters faster, especially for dimensions with many members.

- Continuous date filters, such as relative and range-of-date filters, leverage indexing in your database, making them faster than discrete date filters.

- Computers process Boolean (true/false) and numeric values more quickly than string values

- These can help reduce the query load and work across multiple data sources, improving performance

### 3. Optimize and materialize your calculations

- Execute calculations within the database to leverage its processing power and efficiency
- Reduce the complexity by limiting the number of nested calculations, as they can significantly impact performance
- Finer granularity in calculations can lead to slower performance.

I. LODs (Level of Detail Calculations): Consider the number of unique dimension members involved in the calculation.

II. Table Calculations: The more data points (marks) in the view, the longer it will take to perform the calculations.

- Calculating the average (AVG) is more resource-intensive than using MIN or MAX. Often, rows may be duplicated, and MIN or MAX can provide the same result more efficiently.

- Like include filters, calculated groups only load named members of the domain, while Tableau's group function processes the entire domain.

# 4 KPI

Dashboards will be created to showcase and highlight key performance indicators (KPIs) and relevant metrics related to the disease.


As the system begins to collect historical and periodic data for each user, the dashboards will be updated to present charts that track progress over time, illustrating changes in various indicators and factors.

## 4.1 KPIs (Key Performance Indicators)

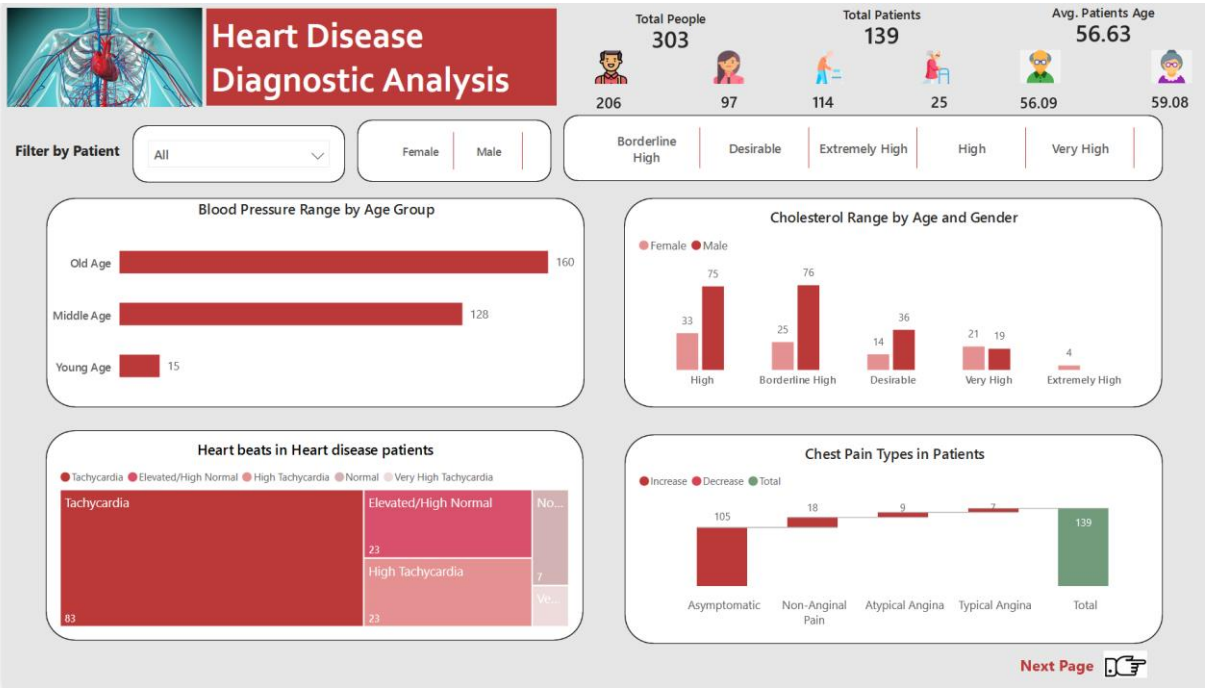
Key indicators displaying a summary of the Heart Disease Diagnostic Analysis and its relationship with different metrics

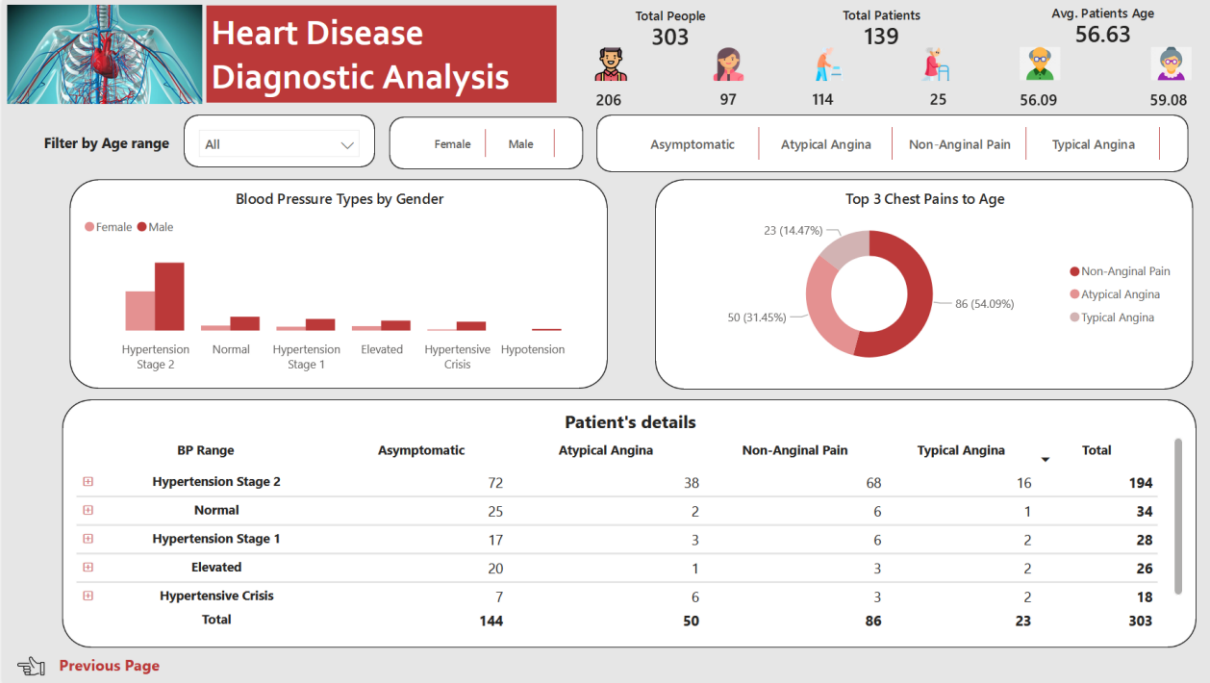
1. Total People
2. Total Patients and Average Patients
3. Blood Pressure Range by Age Group
4. Cholesterol Range by Age and Gender
5. Heart beats in Heart disease patients
6. Top 3 Chest pain types in Patients
7. Blood Pressure Types by Gender
8. Patient's details.

# 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 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.






| BP Range | Asymptomatic | Atypical Angina | Non-Anginal Pain | Typical Angina | Total |
|---|---|---|---|---|---|
| Hypertension Stage 2 | 72 | 38 | 68 | 16 | **194** |
| Normal | 25 | 2 | 6 | 1 | **34** |
| Hypertension Stage 1 | 17 | 3 | 6 | 2 | **28** |
| Elevated | 20 | 1 | 3 | 2 | **26** |
| Hypertensive Crisis | 7 | 6 | 3 | 2 | **18** |
| **Total** | **144** | **50** | **86** | **23** | **303** |