High Level Design

Heart Disease Diagnostic - Analysis

Revision Number: 1.1
Last date of revision: 29/01/2022

Document Version Control

Date Issued	Version	Description	Author
16th January 2022	1.0	First Version of Complete HLD	Arushi Gogia
29th January 2002	1.1	Final Version of HLD	Arushi Gogia

Contents

Abstract

The term "heart disease" refers to several types of heart conditions. The most common type of heart disease is coronary artery disease (CAD), which affects the blood flow to the heart. Decreased blood flow can cause a heart attack. Sometimes heart disease may be "silent" and not diagnosed until a person experiences signs or symptoms of a heart attack, heart failure, or an arrhythmia.

India has one of the highest burdens of cardiovascular disease (CVD) worldwide. The annual number of deaths from CVD in India is projected to rise from 2.26 million (1990) to 4.77 million (2020). Coronary heart disease prevalence rates in India have been estimated over the past several decades and have ranged from 1.6% to 7.4% in rural populations and from 1% to 13.2% in urban populations. Statistics shows One person dies every 40 seconds in the United States from cardiovascular disease. So it becomes very important to study on this matter and take out some meaningful insights from the data for prevention and cure for Heart Diseases.

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:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application
    - Compatibility
    - 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

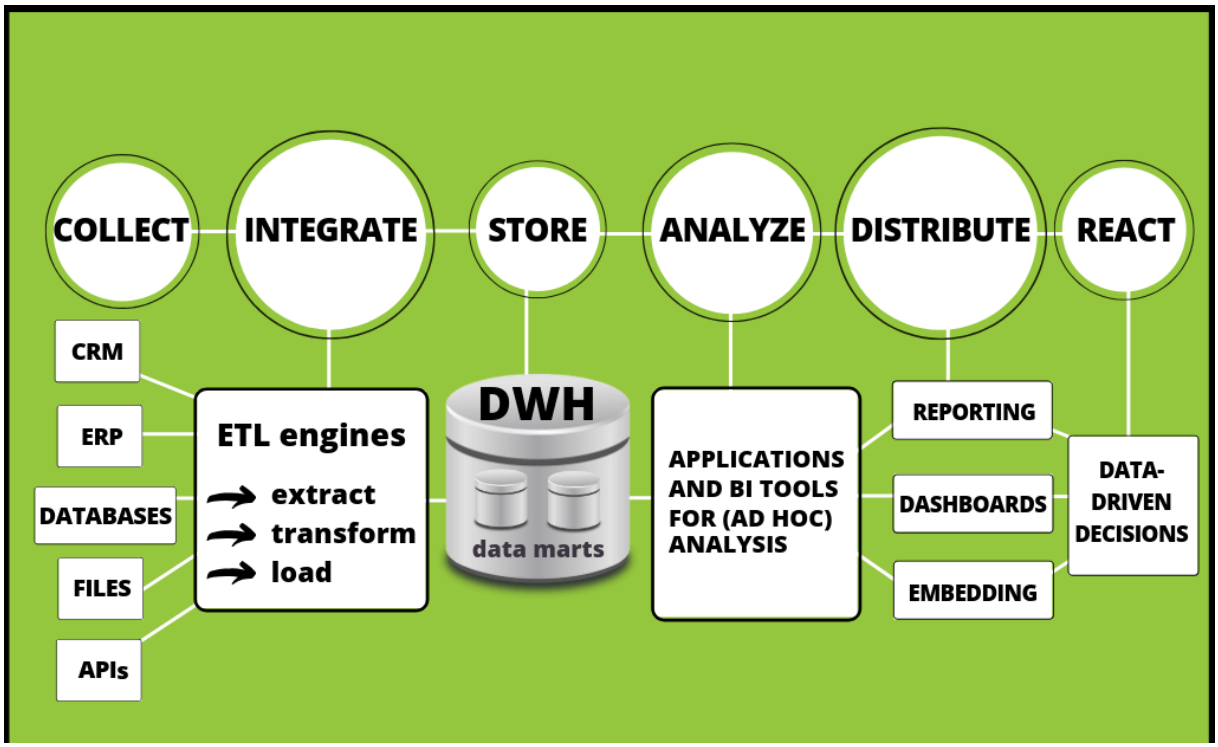
The aim of the project is to analyse and predict the probability of occurrence of Cardiovascular Diseases. This will help us prevent and cure from CVDs. To achieve this we will use dataset which contains 76 features or attributes from 303 patients; however, published studies chose only 14 features that are relevant in predicting heart disease.

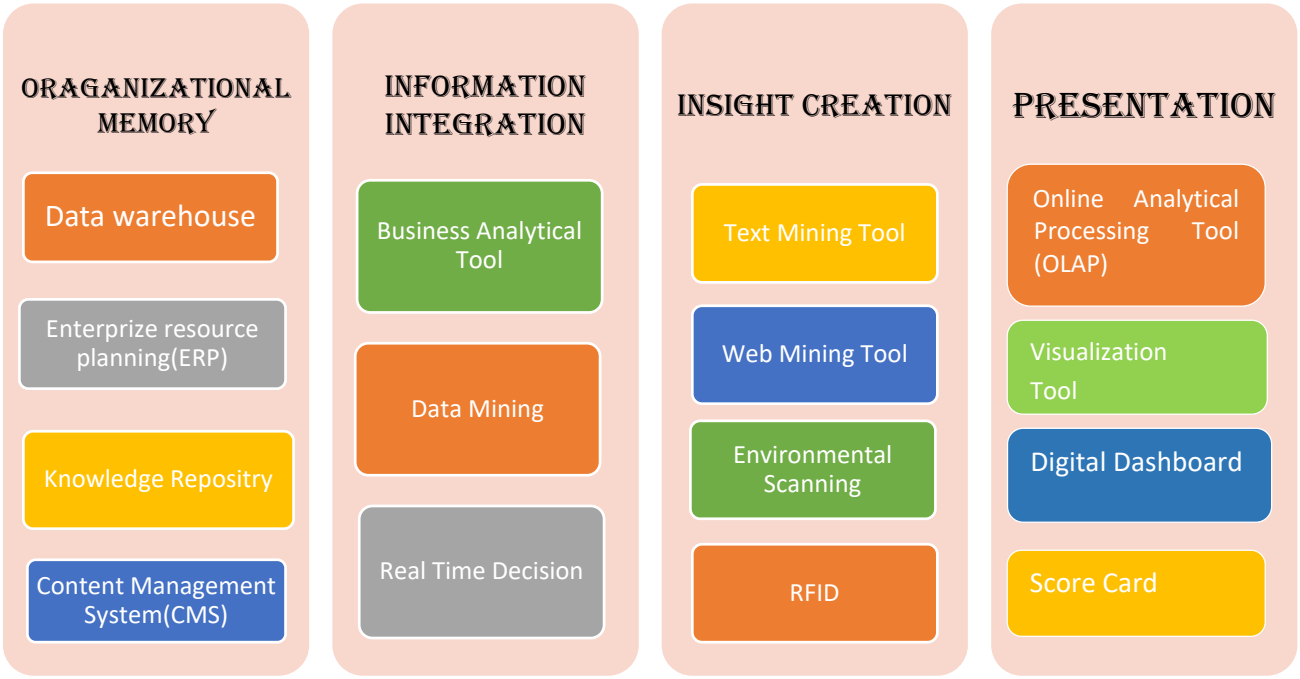
2.2 Tools

Business Tools such and libraries such as Power Bi , Numpy , Seaborn, Matplotplib, Plotly, Pandas and Python Programming language are used to build the whole framework

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 historic/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 Heart Disease Diagnostic Analysis and its relationship with different metrics

- ✓ Percentage of People Suffering from Heart Disease
- ✓ Gender Distribution
- ✓ Age Distribution plays an important role
- ✓ Chest Pain Experienced by people suffering from disease
- ✓ Blood Pressure, Maximum Heart Rate Achieved ,Cholesterol acted as key performance Indicators

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 desktop and Service leverage existing technology and integrates them into IT infrastructure to provide self-service modern analytics platform.

.