High Level Design (HDL)

Heart Disease Diagnostic-Analysis

Revision Version: 1.0

Last Date of Revision:23/10/2022

Priyanka Tanpure

Document Version Control

Date Issued	Version	Description	Author
1 Aug 2022	1.0	Abstract, Introduction	Priyanka Tanpure
14 Aug 2022	1.1	General Description	Priyanka Tanpure
21 Aug 2022	1.2	Design Detail	Priyanka Tanpure
1 Sep 2022	1.3	KPIs	Priyanka Tanpure
23 Oct 2022	1.4	Deployment	Priyanka Tanpure

Contents

Abstract

Health is real wealth in the pandemic time, and we all realized the brute effects of covid-19 on all irrespective of any status. You are required to analyse this health and medical data for better future preparation.

Heart Disease is a general term that includes many types of heart problems, and it has become major concern in India to deal with as there is around 34.3% increase seen in number of deaths due to cardiovascular problems from 2.26 million in 1990 to 4.77 million in 2020.

Thus, it has become necessary to take precautionary measures to prevent and reduce the heart diseases among people. Therefore, to make sure that the people live healthy lives, the needed research and the prevention process of heart disease can be improved by using good data driven systems for understanding the patterns.

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:

A. Present all the design aspects and define them in detail
B. Describe the user interface being implemented
C. Describe the hardware and software interfaces
D. Describe the performance requirements
E. Include design features and the architecture of the project
F. 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

Housing prices are an important reflection of the economy, and housing price ranges are of great interest for both buyers and sellers. In this project, house prices will be predicted given explanatory variables that cover many aspects of residential houses.

The objective of the project is to perform data visualization techniques to understand the insight of the data. This project aims apply various Business Intelligence tools such as Tableau or Power BI to get a visual understanding of the data.

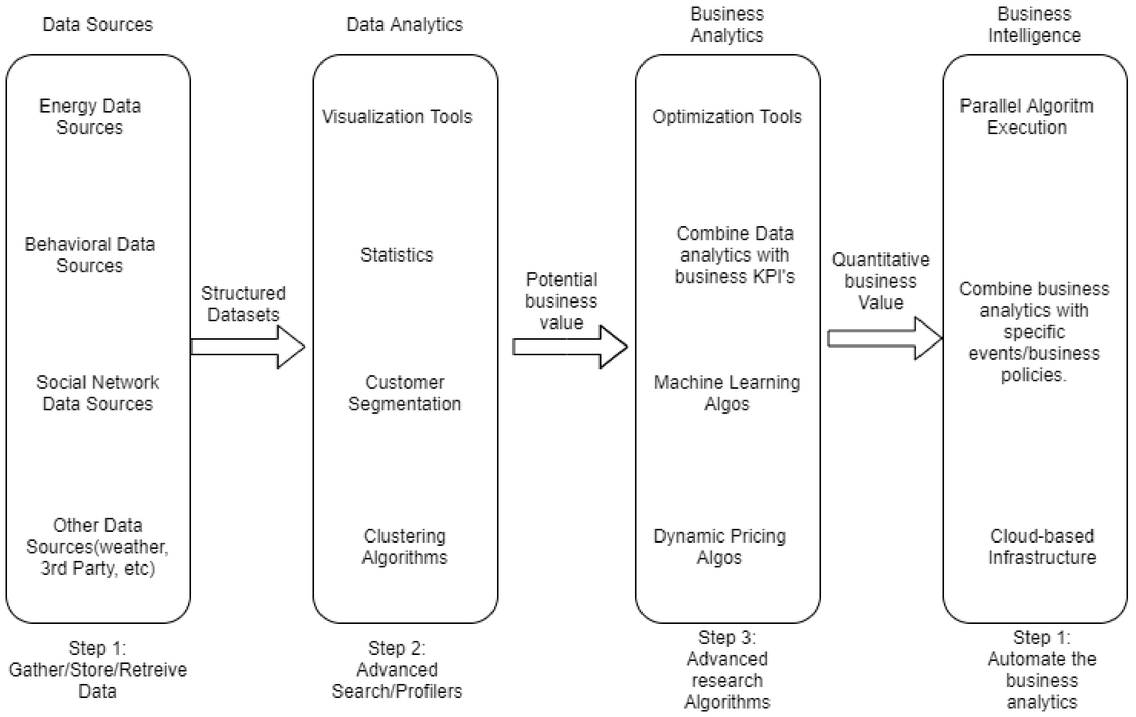
2.2 Tools used

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

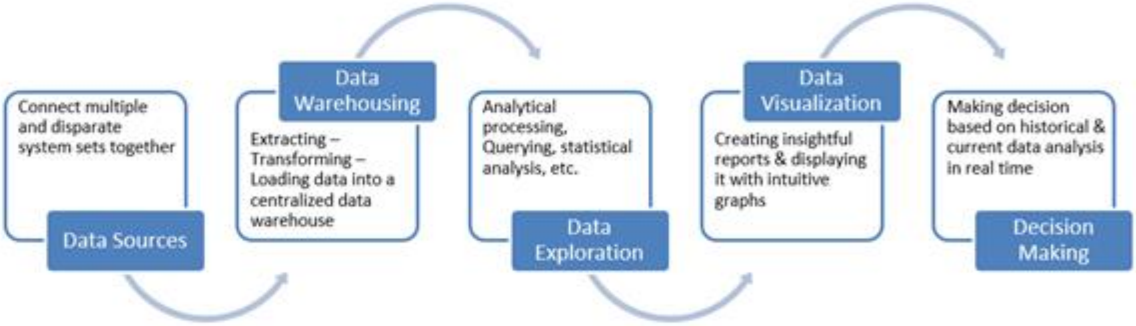
3 Design Details

3.1 Functional Architecture

Functional Architecture of Business Architecture:


Data Sources
Energy Data Sources
Behavioral Data Sources
Social Network Data Sources
Other Data Sources(weather, 3rd Party, etc)
Step 1: Gather/Store/Retreive Data
Structured Datasets
Data Analytics
Visualization Tools
Statistics
Customer Segmentation
Clustering Algorithms
Step 2: Advanced Search/Profilers
Potential business value
Business Analytics
Optimization Tools
Combine Data analytics with business KPI's
Machine Learning Algos
Dynamic Pricing Algos
Step 3: Advanced research Algorithms
Quantitative business Value
Business Intelligence
Parallel Algoritm Execution
Combine business analytics with specific events/business policies.
Cloud-based Infrastructure
Step 1: Automate the business analytics

How Business Intelligence Work:


Connect multiple and disparate system sets together
Data Sources
Data Warehousing
Extracting – Transforming – Loading data into a centralized data warehouse
Analytical processing, Querying, statistical analysis, etc.
Data Exploration
Data Visualization
Creating insightful reports & displaying it with intuitive graphs
Making decision based on historical & current data analysis in real time
Decision Making

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)

Key indicators displaying a summary of the Heart Disease Diagnostic-Analysis 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 Heart Disease Patients according to their Age
6. ST Depression Experienced by Heart Disease People w.r.t their Age

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 analyzing 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 Tableau at scale, as well as organizing, orchestrating, and unifying disparate sources of data for business users and experts alike to author and consume content.

Tableau prioritizes choice in flexibility to fit, rather than dictate, your enterprise architecture. Tableau Server and Tableau Online leverage your existing technology investments and integrate 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 Tableau to match your requirements.

Below is a comparison of the three types:

TYPE PROS CONS

Tableau Server - On Premises

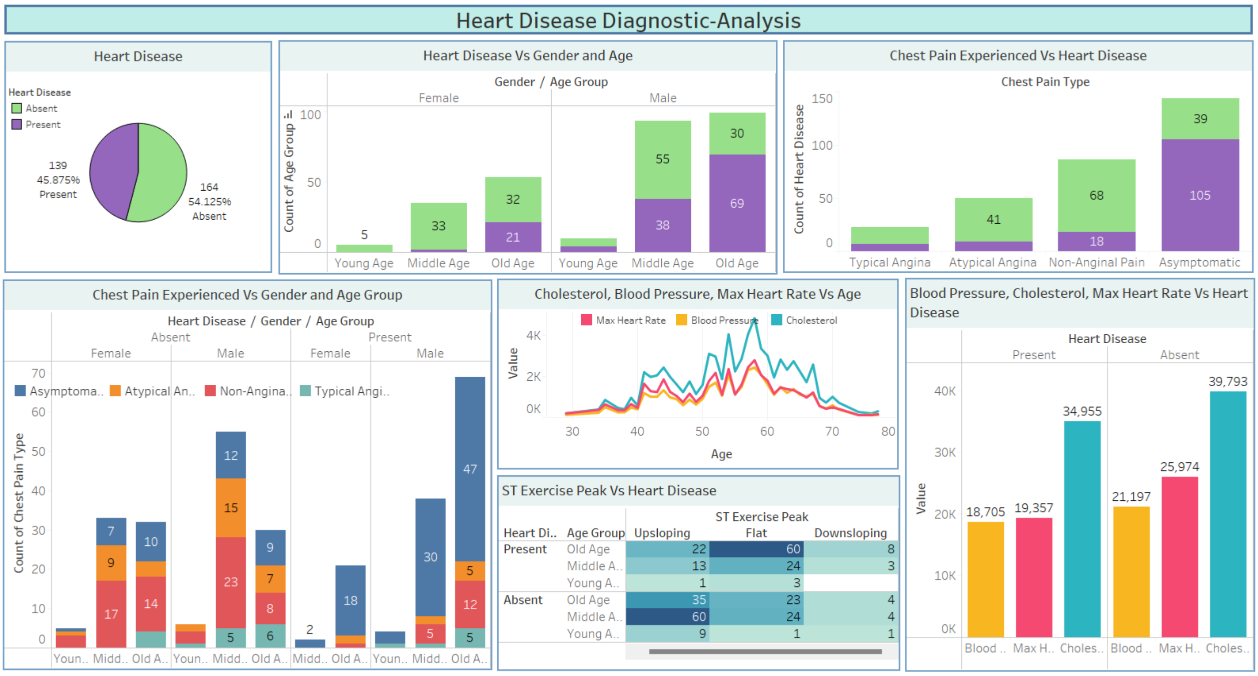
- Full control of hardware and software
- Infrastructure and data remain behind your firewall
- Need dedicated administrators to manage hardware and software
- Additional infrastructure needed to access off-network (mobile, external)

Tableau Server - Public Cloud (IaaS)

- Full control of software on managed hardware
- Puts infrastructure in same place as data (for migration to cloud)
- Flexibility to spin up/down hardware as needed
- Need dedicated administrators to manage software
- Additional infrastructure needed to access off-network (mobile, external)

Tableau Online (SaaS)

- Fully hosted solution (hardware, software upgrades)
- Fast to deploy
- Easy for external audience to access
- Single site in multi-tenant environment
- Cubes are not supported
- No guest account access



ST Exercise Peak Vs Heart Disease

		ST Exercise Peak		
Heart Di..	Age Group	Upsloping	Flat	Downsloping
Present	Old Age	22	60	8
	Middle A..	13	24	3
	Young A..	1	3	
Absent	Old Age	35	23	4
	Middle A..	60	24	4
	Young A..	9	1	1

Heart Disease Diagnostic-Analysis