High Level Design (HLD)

Heart Disease Diagnostic Analysis

Revision Number: 1.1

Last date of revision: 06/09/2021

Shrey Shah

Document Version Control

Date Issued	**Version**	**Description**	**Author**
21st August, 2021	1.0	Initial Draft of HLD	Shrey Shah
6th September, 2021	1.1	Added the Variables under Analysis content	Shrey Shah

Contents

Abstract

As it is rightly said, ‘Health is Wealth’. We have realized this fact in the pandemic time after witnessing the brute effects of Covid-19 on people of all age groups. Apart from this, another major contributor to the death rate is heart related diseases.

Located in the chest region of the body, the heart beats at around 80 times per minute. Even though it is just the size of an average human fist, it is the strongest muscle which continuously to pump blood to the body organs, even at rest.

As a layman, we may feel that the common factors for heart related diseases are cardiac arrest or blockages. But the dataset under analysis describes multiple different medical parameters associated with the heart and their typical values. We will be analyzing the relationships between them and study the implications of changes in those parameters.

1 Introduction

1.1 Purpose

The purpose of this High-Level Design (HLD) Document is to describe the design of the software product in terms of the architecture. Customer requirements are initially converted into HLD. It provides the high – level functional design of the software being developed. This document describes the functioning of the various components involved in the entire system and their interrelationship. This document will form as a basis for all the queries that will be encountered in the future and also serve as an input for the Low Level Document (LLD).

The HLD will cover the following aspects:

1. Present the design aspects and define them in detail.
2. Describe the user interface being implemented.
3. Describe the performance requirements.
4. Include design features and the architecture of the project.
5. List and describe the non-functional attributes like:
   - Performance
   - Security
   - Maintainability
   - Portability

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.

1.3 Intended Audience

The HLD document can be used as a reference by the following categories of people.

1. Management Team
2. Programs Team
3. Solutions Team
4. Quality Assurance Team

1.4 Acronyms and Definitions

This sub - section includes the definitions of all acronyms required to interpret the HLD properly

Sr. No.	Acronym	Definition
1.	HLD	High Level Design
2.	.csv	Comma separated value – a file format in which the fields are separated using comma – (,) as a delimiter
3.	BI	Business Intelligence
4.	LOD	Level of Detail
5.	UI	User Interface
6.	NFR	Non Functional Requirements

2 General Description

2.1 Problem Statement

Heart diseases have been known to take a major toll of people’s lives. In this project, we will analyze the heart disease diagnostic dataset for different medical terms associated with the heart. Based on these parameters, we will try to find key metrics and derive meaningful relationships between attributes along with visualizations.

The objective of the project is to perform different analysis techniques on the heart disease dataset and derive inferences based on the output of the visualizations. In this project, we will be incorporating most trending and powerful BI tool namely Tableau.

2.2 Tools used

Business Intelligence tools and libraries works include Numpy, Pandas, Excel, R, Tableau, Power BI can be used to build the whole framework.
In our analysis, we will be using Numpy, Pandas and Tableau Public Desktop and Tableau Public Server as the tools.

3 Design Description

3.1 The Business Intelligence Framework

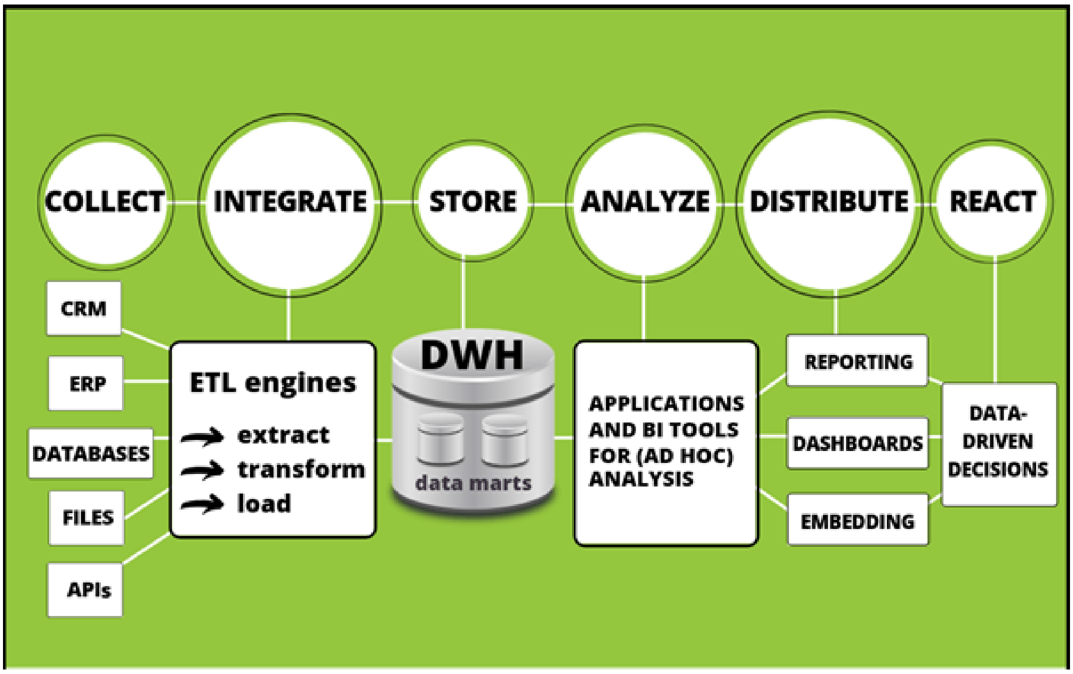


Fig. 1 – Typical Business Intelligence Framework

Fig. 1 shows the process of a a typical Business Intelligence Framework. It has 6 major phases with multiple components involved described as below:

1. **Collect**: This is the initial phase and begins with the extraction of the data relevant to the problem statement from multiple sources – Customer Relationship Management (CRM) data, Enterprise Resource Planning (ERP), internal and external databases, file systems and openly exposed Application Programming Interfaces (APIs).
2. **Integrate**: The data collected in the previous stage is fed as input to ETL engines where the data undergoes extraction from sources, gets transformed into manageable data by applying suitable transformations and is loaded into storage units for further usage.
3. **Store**: The transformed data is stored in large data warehouses and drilled down to smaller usable chunks called data marts.
4. **Analyse**: The data is analysed to derive inferences and get in-depth understanding of the data and analyse the existence of various patterns.
5. **Distribute**: The derived conclusions are then packaged into lucid and interactive dashboards and reports and presented to the stakeholders.
6. **React**: Based on the reports and dashboards, the stakeholders make appropriate decisions corresponding to the problem statement taking into consideration factors like marketing, sales, profits, expenses etc.

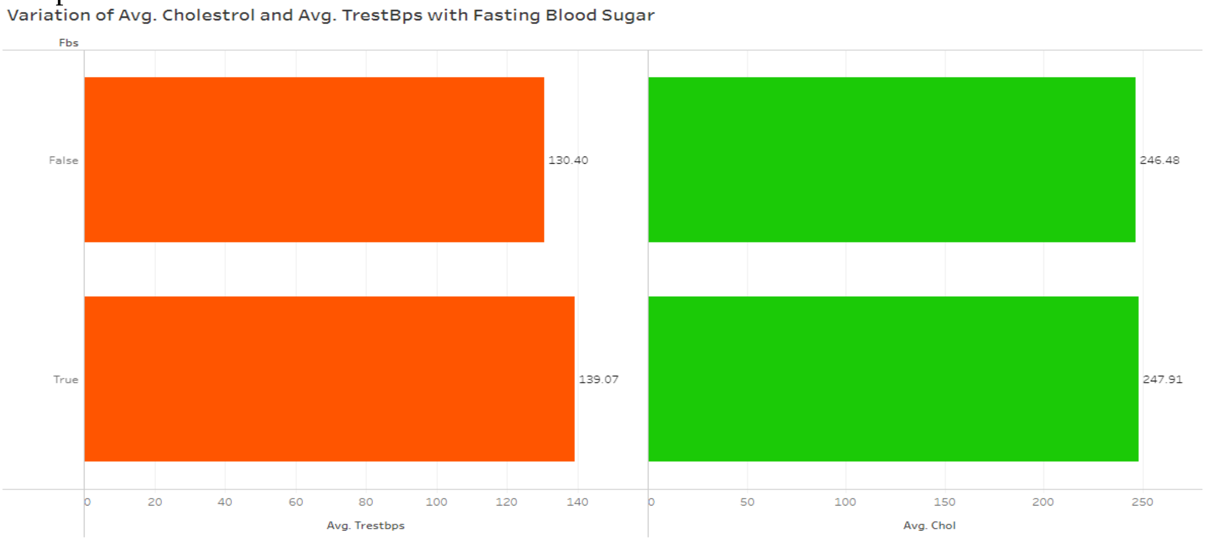
3.2 System Design

A) User Interface:

I will be using Tableau as the primary BI and data visualization tool.
As a part of the User Interface, the user will be presented with certain Tableau sheets and Dashboards.

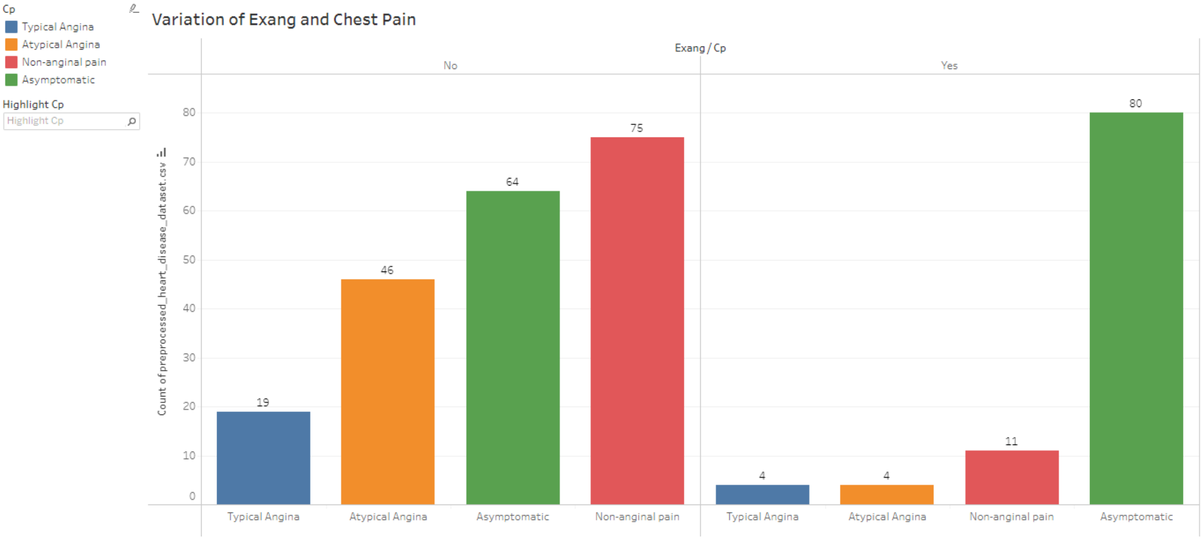
i) There will be some visualisations of the different key parameters which will be of static nature.

Sample Static UI



ii) While others may be dynamic in which the user may provide some inputs or may click / hover different parameters to see the corresponding changes in graphs.

Sample Dynamic UI



B) Data Details

The dataset used for analysis is the Heart Disease dataset provided by the UCI Repository. It actually contains 76 attributes out of which only 14 are used. We will be using the Cleveland dataset.

Dataset source: https://archive.ics.uci.edu/ml/datasets/Heart+Disease

The input dataset will be in the form of a .csv file which will undergo the following steps:

- Data Import (in Python)
- Data Preprocessing
- Data Export (to .csv file)

4 Dataset Variables under Analysis

Below are the different relationships that will be a part of the analysis of the Heart Disease Diagnostic dataset.

1. Variation of the ‘age’ with ‘chol’ (Cholesterol) and ‘sex’
2. Variation of ‘cp’ (Chest Pain type) with ‘sex’
3. Variation of ‘thal’ (Thalassemia type) with ‘sex’
4. Variation of ‘thalach’ (Maximum heart rate) with ‘age’
5. Variation of ‘chol’ (Cholesterol) , ‘trestbps’ (Resting blood pressure) with ‘fbs’ (Fasting Blood Sugar)
6. Variation of ‘restecg’ (Resting electrocardiograph results) with ‘sex’
7. Variation of ‘exang’ (Exercise induced angina) with ‘cp’ (Chest Pain type)
8. Variation of ‘slope’ (Slope of the peak exercise ST segment), ‘restecg’ (Resting Electrocardiograph results) and ‘oldpeak’ (ST depression induced by exercise relative to rest)
9. Distribution of ‘ca’ (Number of major vessels)
10. Variation of ‘num’ (Angiographic disease status) with ‘sex’

5 Non Functional Requirements

Non Functional Requirements (NFRs) are those requirements which aim at bringing and building ‘quality’ into the application. They are the constraints that apply to a set of functional requirements and allow you to judge the attributes of a solution rather than its functional behaviours. These include availability, maintainability, performance, reliability, scalability, security, and usability and so on.

NFRs can be incorporated in different ways into the Software Engineering Paradigm. The most common ways of doing this are with an explicit backlog item, as Acceptance Criteria, or as part of the team’s Definition of Done.

The NFR can be created as an independent Backlog Item like a Technical Story for the component. Another way, is to consider them as an Acceptance Criteria which are the conditions which must be met for that component to be accepted. Else if NFRs are applicable globally for the project they can be laid down in the Definition of Done documentation.

For our analysis on the Heart Disease Diagnostic project we will be looking into the below NFRs with respect to Tableau Public Desktop and Tableau Public Server products:

1. **Performance:**

   - Tableau Public Desktop works on the dataset and hence using the Live or Extract connection depends on the data. In case the data does not change frequently and is small in size, it is better to use the Extract connection. But in case of frequently changing dataset Live Option is preferable.
   - Try displaying your data in different types of views.
   - Reduce the number of filters in use. Excessive filters on a view will create a more complex query, which takes longer to return results.
   - Use parameters and action filters. These reduce the query load.
   - Perform major calculations in the database rather than using complex calculated fields.
   - Reduce the granularity of LOD or table calculations - the more granular the calculation, the longer it takes.

2. **Security:**

   - Tableau provides several ways to control which users can see which data. For data sources that connect to live databases, you can also control whether users are prompted to provide database credentials when they click a published view.
   - The administrator can also set the access and privileges of the user depending on Viewer, Explore, Administrator roles which prevents misuse of data.
   - Since we will be using Tableau Public Server for our analysis, it does not offer strong security measures as the data you publish is visible to anyone. We need to be more mindful that we do not publish any confidential details.

3. **Maintainability:**

- Maintaining content in Tableau Public Server is easy as it does not involve much complex controls.
- If you wish to modify the existing dashboards and stories, you can easily connect to the Tableau Public Server and login via personal credentials and perform the changes.

4. **Portability**

- Tableau Desktop Public is an application and it needs to be installed in the personal system. It is available for multiple Operating Systems and is platform – independent.
- Tableau Public Server is portable as it does not require hardware setup. We just need to login to the website via the personal credentials and we will be able to use it.

6 Deployment

Prioritizing data and analytics could not come at a better time. Companies are 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 growth of enterprise data, database technologies, and the demand for analytical skills, most IT organizations have shifted their focus to deploying and operating Tableau at scale, and organizing and unifying disparate sources of data for business users and experts alike to author and consume content.

Tableau prioritizes choice in flexibility to fit to the 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 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:

Types:

A. 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)

B. 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)

C. 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

Depending on your organizational roles and responsibilities, Tableau Server should be installed by a systems administrator and the designated Tableau Server Administrator in coordination with the appropriate IT roles. For Tableau Online, you will integrate with your existing technology and configure the site settings. The Data & Analytics Survey, completed by business teams, identifies and prioritizes data use cases, audience size, and users. You use the information collected to plan your deployment strategy, including sizing, installation, and configuration of your Tableau Server or integration and configuration of Tableau Online. In addition to this, administrators need to plan for the client software installation of Tableau Prep Builder, Tableau Desktop, Tableau Mobile, and Tableau Bridge for Tableau Online where applicable.