CS491 Project Specification Document

Team ID: T2506
Project Name: Hipograf

Team Members:
Salih Furkan Göktaş, 22202620
Orhun Güder, 22202471
Şükrü Eren Gökırmak, 22203746
Ramiz Arda Ünal, 22202554
Artun Berke Gül, 22203316

Table of Contents

1. Introduction

1.1 Purpose

Medical data, in its raw form, is hard to read and understand at a glance. If there are no charts to visualize this data, or this data is presented all at once, it is easy to miss crucial information useful for a diagnosis.

Hipograf is a new age medical visualization and graphing application that bridges the gap between humans and medical data across Türkiye. The purpose of Hipograf is to help doctors make informed decisions regarding their patients by providing them with easy to read medical history visualizations. It is also intended as a tool for patients to be more informed about their own conditions and health.

1.2 Scope

Our project, Hipograf, is a web application. It will visualize the medical data and history of the user, with various data visualization tools like charts, graphs, timelines, etc. Hipograf is intended to be useful to doctors when making a medical decision regarding their patient, by presenting them important information in a clear, intuitive way. The most important benefits of Hipograf are seeing the most important information in legible charts and facilitated filtering of desired information, which will result in more informed decision-making. Hipograf is intended for Turkish users, with a goal of being integrated with the Turkish medical system E-Nabız. Since the medical data format is different outside Türkiye, the app will be unavailable outside of Türkiye.

1.3 Product Overview

Hipograf aims to help both doctors and patients across Türkiye using data visualization. Anyone in Türkiye with an E-Nabız history will be able to use it to view their medical data. More specifically, they will be able to look through blood work data, medications, hospital visits, diagnoses, medical operations and more. All of this data will be presented to the user in an easy to understand, charted format. Users will be able to filter and select this data using techniques not normally available to them.

1.3.1 Product Perspective

Hipograf will only feature digital user interfaces. Users will be able to interact with the application by opening the Hipograf webpage. Users opening the Hipograf webpage will be presented with a login screen prompting them to log in with their Hipograf account. This account is used only to hold various preferences of the user, and no other information. After logging in, they will be presented with a screen containing various medical information. This screen will be explained in more detail in Section 2.1. Users will be able to sort and select the data they see, using tools like a search bar, a date filter, and selecting appropriate medical tags such as “Prescriptions” or “Blood Work”. Users will also be able to fetch more information about a specific data type, such as selecting blood work from a specific date, which will narrow down the displayed information. The data display preferences of the user can optionally be saved on their Hipograf account for convenience.

1.3.2 Software Interfaces

Hipograf will make use of a variety of software products and packages. First and foremost, Hipograf will be connected to the E-Nabız system. After the user logs into their E-Nabız account and grants Hipograf access to their medical information, their relevant medical data will be retrieved from the E-Nabız API.

As a web application, Hipograf is also reliant on the various software and packages used to build and host the website itself. In particular, it uses React TypeScript and Tailwind CSS.

Hipograf will also be connected to a small database for storing basic user information such as login credentials and data display preferences. This database will be hosted with MongoDB. The backend will link the database to the frontend. The backend of Hipograf will be built using the Flask web framework and will be written in Python.

1.4 High Level System Architecture of Proposed Solution

Hipograf follows a layered architecture consisting of three main layers. These are the Data Layer, the Presentation Layer and the Application Layer.

The Data Layer is split into two distinct parts. The first part will not have a direct representation on the server ecosystem that houses Hipograf's internals, since no medical data is stored on-site. It will be represented by read-only API calls to fetch the relevant data when necessary, and will also be represented with an abstract Facade class that provides temporary data during development. More details can be found in Section 2.3.1.2. The second part will contain a local key-value store for storage of a very limited subset of the total data that is available to Hipograf. It will consist of the log-in credentials to be used when directly logging into the Hipograf ecosystem, alongside information about data display specifications without storing the data itself (for instance, the information that graphs should use a bimonthly timeframe).

The Application Layer will contain the backend logic. Both the medical data fetched from API calls and other data fetched from the local database calls in the Data Layer will be processed here before being sent to the Presentation Layer for display. Only the second part of the Data Layer can be written to, since the first part contains read-only medical data.

The Presentation Layer will contain the application's frontend, and be the main web-facing component that is accessible to Hipograf end-users. The data that is received from the Application Layer will be formatted and filtered to be shown according to their preferences. As mentioned previously, they can optionally choose to save certain data display measurements in the system, which will be relayed back to the Application Layer.

In Figure 1, a simple diagram is shown to demonstrate the high level architecture diagrammatically.

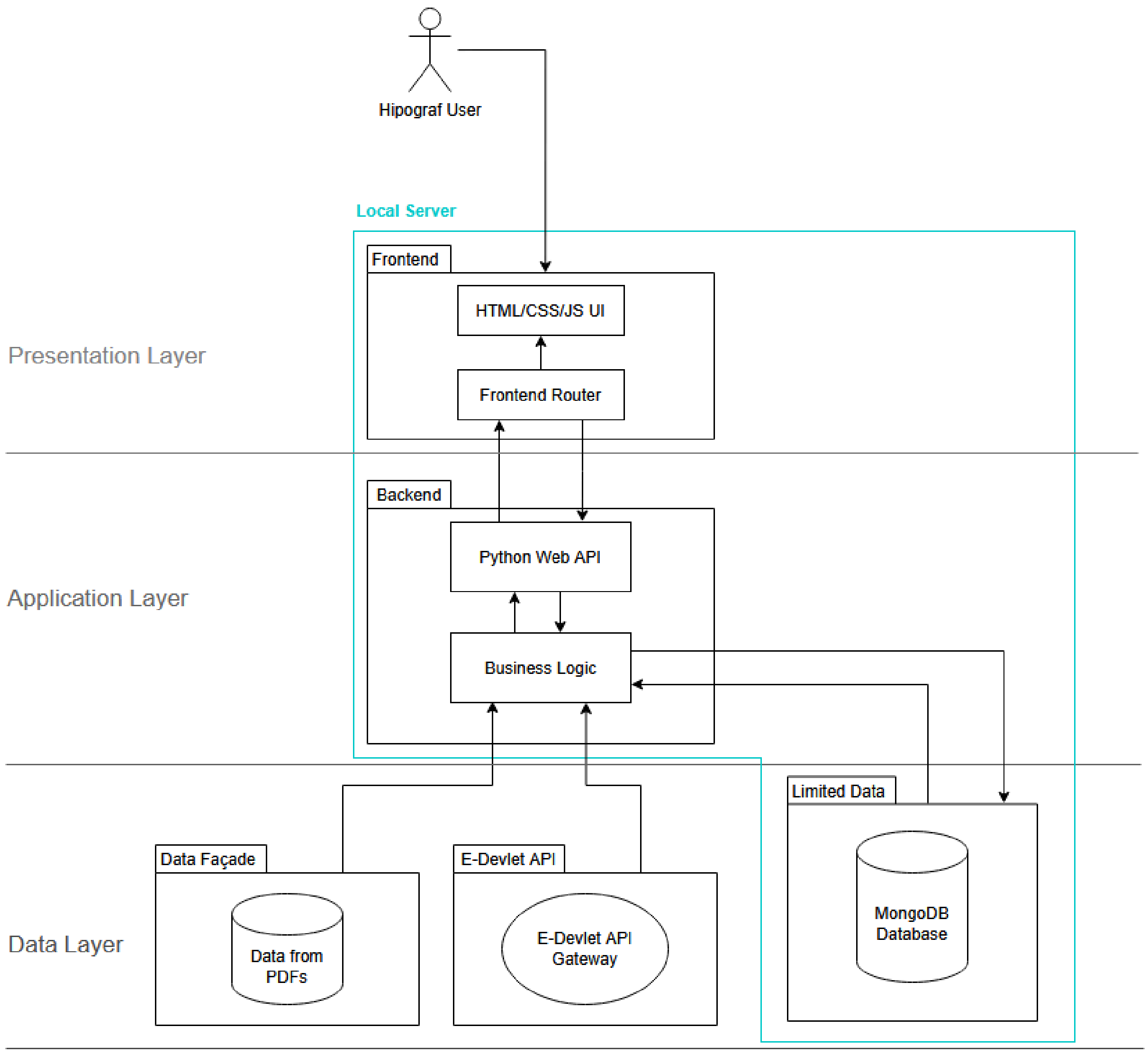

Figure 1 - Diagrammatic representation of high-level system architecture. Created using draw.io

1.5 Professional and Ethical Issues

Medical data is very sensitive, and considered to be within 'special categories of personal data' (özel nitelikli kişisel veri) by Law 6698 denoting the details of the Personal Data Protection Law (Kişisel Verilerin Korunması Kanunu) of Türkiye [1].

Full transparency must be provided to users before they continue using the program, with them consenting to letting their medical data be read and processed by Hipograf.

Hipograf’s UI must avoid misrepresenting medical information since the change in perspective brought on by detailed visualization can influence the user’s views of an issue, and thus also influence their diagnosis. Our aim is to optimize and accelerate the communication between patients and medical professionals as much as possible while avoiding misdiagnoses.

1.6 Standardization

Throughout the course of the project, we are planning on following various standards in multiple different categories. These include engineering standards, legal standards and medical standards. Engineering standards specifically concern both the planning/design and the implementation of the project. Legal standards concern data privacy. Medical standards concern industry conventions for the labelling of various medical terms.

1.6.1 Engineering Standards

1.6.1.1 Requirements Engineering and Documentation

For requirements engineering, we have sought standardization documents jointly published by the International Organization for Standardization (ISO), International Electrotechnical Commission (IEC) and Institute of Electrical and Electronics Engineers (IEEE). These documents include ISO/IEC/IEEE 29148:2018 and ISO/IEC/IEEE 12207:2017. The first document gives an overview of requirements engineering for multiple different project types, including Software Requirement Standards (SRS) which is the one that we are concerned with. The second document expands on this, also being cited in the first document, providing more detail on how various fields are expected to be filled out in the software case [2, 3].

1.6.1.2 Design Modelling

We refer to Unified Modelling Language (UML) 2.5.1 modelling standards published by the Object Management Group (OMG) for the system’s design, encapsulated with the suite of structural and behavioural diagrams that are defined with UML including use case diagrams, activity diagrams, state diagrams and class diagrams [4].

1.6.1.3 Web Accessibility

In order to ensure that Hipograf remains accessible to as many people as possible, we intend to follow the Web Content Accessibility Guidelines (WCAG) 2.1, which are detailed web accessibility standards published by the World Wide Web Consortium (W3C) [5].

1.6.2 Legal Standards

As mentioned previously, medical data is sensitive data. In order to ensure that it is being accessed and processed within the guidelines of the law, we have referred to relevant Turkish Law on the privacy of sensitive data.

1.6.2.1 Privacy of Sensitive Data

In order to ensure that we are complying with national laws when dealing with patient data, we are referring to Türkiye's Personal Data Protection Law, named 'Kişisel Verilerin Korunması Kanunu' (KVKK) on guidelines in dealing with sensitive data in a legally acceptable manner [1].

1.6.3 Medical Standards

We need to follow specifications and standards that are used within the global and national medical community to ensure parity with existing data and systems. This section outlines which ones we need to follow.

1.6.3.1 Disease Classification

The International Classification of Diseases (ICD) is a detailed compendium of medical ailments and afflictions that is maintained by the World Health Organization. Each disease or condition is provided with its own unique alphanumeric identifier code. Similarly classed or otherwise related conditions are given similar identifiers [6].

1.6.3.2 Disease Classification - Turkish Standard

The Turkish Ministry of Health keeps an online and publically accessible reference database for medical encodings and codes used in Türkiye, known as the 'Sağlık Kodlama Referans Sunucusu' (SKRS). One of these includes a translated and reformatted set of the aforementioned ICD-10 codes [7].

1.6.3.3 Treatment/Medication/Equipment Standards

The SKRS also contains other standards, including the 'Sağlık Uygulama Tebliği' (SUT) codes. This standard defines similar alphanumeric codes for various treatments, medication and equipment stationed at medical institutions around Türkiye [7].

2. Design Requirements

2.1 Product Functions and Functional Requirements

2.1.1 Types of Users

Hipograf has two primary types of users as its target audience: Medical practitioners and patients. Medical practitioners are expected to use the application to analyze their patients' medical information to make more informed decisions regarding their decision in treatment. Patients have the option to use Hipograf to share their information, and view their own medical data for any reason, especially if they themselves are a particularly informed patient who would benefit from being able to access detailed visual representations of this data. Medical practitioners and their patients are somewhat interchangeable roles depending on the context, since one medical practitioner can be the patient of another.

2.1.2 Medical Practitioners

Medical practitioners are the primary audience of Hipograf. They are expected to use the system to easily gather information about their patients and make better informed decisions about their conditions. There are many functionalities that would be useful to a medical practitioner (referred to as "the practitioner" in the following subsection). These are in the form of different kinds of visualization tools, each useful for particular types of medical data.

2.1.2.1 List Visualizations

Many kinds of information are useful when shown in a list view. A medical practitioner should be able to do the following actions.

1. List all patients that have shared their medical information with the practitioner.
2. Select a patient from their appointed patients to enter a more detailed page about this patient. This main page displays basic personal information (name, birthdate, city of residence, etc.), along with medical information. This medical information includes recent blood work, medications prescribed, hospital visits, diagnoses, medical operations, radiology images, vital signs gathered from wearable technologies, and a timeline that puts all this information on a unified timeline.
3. View a patient's medical tests, such as blood work and radiology images. Additionally, they should be able to filter shown blood work instances by certain criteria (e.g. whether they include a certain parameter, the time the test was done, etc.). Finally, they should be able to select a single blood work instance to view it in its entirety. This view shows when the test was done and all parameters measured in the test. The practitioner can click on a parameter to show all instances of that parameter showing up in blood work in a graph view.
4. List medications prescribed, and filter them by time of prescription. Select a prescribed medication to get brief information about the medication.
5. List vaccinations of a patient and filter them by time range.
6. List all the diagnoses, and apply filters (e.g. whether the diagnosis is chronic, name of the diagnosis, the time frames the diagnosis was given in, etc.) to view more specific groups of diagnoses.
7. List all the hospital visits and be able to filter them through by certain criteria (e.g. date of admission).
8. List all the radiological images of a patient, filtered by the type of imaging it was and the date that it was taken. Selecting a radiological image from the list takes the practitioner to a page that gives more detailed information.
9. List all the medical operations and apply various filters.

2.1.2.2 Timeline Visualization

Hipograf will include a separate page for a timeline. This timeline will include all medical events the patient has partaken in. These medical events consist of information obtainable through E-Nabız that have dates attached to them and can be things like doctor visits, blood tests, etc. The timeline will be scrollable and resizable to prevent cluttering that can be caused by a period of frequent medical events and to allow user freedom in how they view their information.

Magnification of the timeline will change the earliest and latest dates that are shown on the screen while keeping the size of the time interval constant. On the other hand, resizing the timeline will change the size of the time interval that is shown to the user. The timeline can be further modified by the user's filtering choices. The user will be able to filter out which medical events are shown by choosing to include or exclude various categories of events. In this case, the chosen event types will remain on the timeline while the rest will be removed. Further filtering can be done through choosing specific dates. This will cause all events falling out of the specified time interval to be left out of the timeline. Finally, any medical event located on the timeline can be individually selected to obtain further details about that medical event.

2.1.2.3 Graph Visualizations

The application will include a robust graphing system, where the user will have the option of choosing which graphs to create based on which data they would like displayed. These graphs will display data as a function of time. The data itself would be periodic data that is gathered on various discrete instances over time, such as the values acquired from medical blood tests. Data obtained from wearable technology that the patient has linked appropriately can also be graphed.

The user is able to create graphs on an interactive screen where one or more data types can be specified from a searchable dropdown menu, alongside a time range, to graph. They can resize the graph and move it around on the screen. Other graphs can also be created with similar input parameters and placed on the screen.

When one more than one graph has been created, the user can choose to either view them separately on the screen or combine them to produce graphs plotting several data points at once. This action automatically normalizes the data so that the combined graph can display the result meaningfully, and multiple y-axes will appear to accommodate the multiple data types. These will be appropriately identified with differing colors and markings, identified in the graph's legend. A combined graph can then also be split up back into the individual constituent graphs. Ultimately, absolute freedom will be given to the user on which datasets to combine on which graph according to what combinations they feel would be useful to observe.

Users will also have the option to save presets for the graphs that are being displayed in order to recreate them at a later date. This specifically only includes the ways in which the data is displayed, not the data itself. For instance, the fact that a graph has been created over a certain time period that charts two different variables together might be saved, which can then be brought back at any point.

2.1.3 Patients

Citizens who are mindful of their health (and how it progresses over time) but are not themselves practicing medical practitioners will also be able to access Hipograf. Their access will be limited to their own personal data, but otherwise they will be able to perform all the functions listed under Section 2.1.2 which are not related to viewing the details of other patients.

2.2 Non-Functional Requirements

2.2.1 Reliability Requirements

Hipograf is designed to be used in the medical sector, where access should be guaranteed for as long periods as possible. Care should be taken to avoid the user being presented with incomplete information. Hipograf should notify the user that its knowledge base is incomplete when there is an issue with the amount of information it is able to access.

Hipograf will have an uptime of at least 95% on a daily basis, translating to at least 347 days a year of full availability. This includes weekends and holiday periods. Only the holiday periods of Türkiye will be considered in this definition, since the application is intended for use in Türkiye.

Any scheduled maintenance for server upkeep should not result in any outages exceeding 3 hours, with off-peak hours being preferred as possible, especially the early morning range from 3.00 AM to 6.00 AM. Once again, since service is only provided within Türkiye, no consideration needs to be made for whether these hours would coincide with a different region's peak usage hours.

Any inability to fetch data from an API or otherwise, whether a complete or partial failure, must be immediately communicated to the Hipograf user in the form of an alert that blocks further input until acknowledged. This precaution will ensure that these users do not gain an unfound misunderstanding of an arbitrary patient's medical history with the limited available data.

2.2.2 Performance Requirements

One of Hipograf's main objectives is to increase efficiency, not take away from it; being designed to be a system that, once deployed, is used by medical facilities across all of Türkiye. This means the system should be able to work under heavy load, letting users quickly access whatever representation of data they require.

Hipograf's login page should take less than 2 seconds (wall-clock time) to connect to a client with an internet connection (in other words, any potential slowdowns should not have Hipograf's server infrastructure as the common denominator. Naturally, we cannot account for subpar connection speeds that a user is experiencing when connecting to the internet).

After logging in, Hipograf's main overview dashboard should take less than 5 seconds to load.

Subsequent operations conducted to view medical data (such as moving/scaling/combining graphs, filtering selections, specifying affliction types to narrow data input, etc.) should feel responsive and fluid, with no input mouse/keyboard lag more than 200ms present on hardware comparable to the average work computer available at medical institutions in Türkiye.

Hipograf should still be fully accessible and operational when 500 users are simultaneously connected to it, without encountering any slowdowns that would violate the time constraints given in the previous paragraph.

2.2.3 Security Requirements

Secure handling of personal and private data is critical to Hipograf's success. No medical data will be kept permanently on site, and all data that is handled during the duration of a log-in session will be deleted immediately upon session termination.

Web-based access to Hipograf will be over an HTTPS connection using Transport Layer Security version 1.3, which itself will use the Advanced Encryption Standard in Galois Counter Mode with 128-bit keys.

Direct log-in to Hipograf will be enabled through user defined passwords that must meet the following requirements:

- At least eight (8) alphanumeric characters in all
- At least one (1) uppercase letter
- At least one (1) lowercase letter
- At least one (1) numeric digit
- At least one (1) special character

These passwords will be hashed using the Argon2 password hashing algorithm and stored in a local database that will not be directly web-facing.

After a user logs out of their current Hipograf session, no medical data will be allowed to persist on any layer of Hipograf's internal architecture.

2.2.4 Usability Requirements

As Hipograf's main purpose is to be an aid to medical information visualization, usability is one of the most important aspects of the system. The user interface should be easily understandable and usable, along with being accessible enough to keep the number of people that could use it effectively at a maximum.

All operations will be possible by use of the keyboard, and care will be taken to avoid the presence of any keyboard traps (UI elements that keyboard input is unable to "escape" as focus cannot be redirected back to previously accessible page components) [5].

All visual content should have an explanatory textual description that is shown within the HTML tags and also when a visual component fails to load [5].

When navigating through to different components and pages of the site, the current position tab should be identified on the screen through a navigation bar [5].

There should be a color contrast of at least 4.5:1 between text and its background for optimal readability [5].

2.3 Constraints and Limitations

This subsection discusses both limitations that we have placed on the application and constraints out of our immediate control.

2.3.1 Implementation Constraints

2.3.1.1 Rationale For Web Application

Hipograf will be implemented as a web application that can be easily viewed from a desktop computer. This constraint is put into place because a desktop computer is the most likely device to be used for medical information visualization inside a hospital. A desktop application would make the application impossible to use from a mobile device. Mobile devices should still be able to view this application, however, even if there is reduced ease of use because of the use of a touchscreen.

2.3.1.2 Data Impermanence

As Hipograf is only supposed to be a visualization solution, it should not modify any medical record from any system it is integrated with. It should only fetch data that is required to create the desired visualizations and safely discard them afterwards. This data impermanence policy does not apply to all information. For instance, it would not apply to a medical practitioner's preferences regarding the user interface (like graph view presets). These do not constitute private information, and thus it is acceptable to store them for future use by the practitioner.

Due to the fact that the application development must be done before integration into e-Nabız, the application will have to store some medical data to construct the minimal viable product. This data will come from either the developer team's own medical information per their consent, or be generated data that does not belong to a real patient. After proper integration, the application would stop this storage of medical data.

2.3.1.3 Security

Hipograf is bound to operate on private information that must be unreachable to adversarial actors. This requires that the system has proper authentication and authorization in place.

2.3.2 Ethical Constraints

Hipograf will operate on medical information of its users. Medical information is deeply personal and private. This makes it absolutely necessary that this data should be kept from malicious actors. Furthermore, the operation of the application must be compliant with KVKK [1].

Another point to consider is that this system will be used by medical practitioners to diagnose patients. This means that this application is partially responsible for all the diagnoses it's used in, including the errors in judgment that result from Hipograf's errors. For this reason, Hipograf must be implemented in a way where the information shown is ensured to be correct, and in the event of an error, the medical practitioner should be explicitly and clearly warned.

3. Feasibility

3.1. Market and Competitive Analysis

3.1.1 General Data Management and Visualization Tools

Two very prominent data visualization tools used in the industry are Microsoft Power BI and SAS Visual Analytics [8, 9, 10]. Both of these tools have vast data visualization capabilities with a high capacity of customization with support for a variety of data domains. However, the magnitude of capabilities provided by these tools can be a detriment when they are not used by an industry professional. An important goal of Hipograf is for it to be intuitive and be easy to use by everyone, regardless of their background. The variety of options and ways to customize provided by these tools can overwhelm Hipograf's target audience. On top of this, taking advantage of Microsoft Power BI requires the data to be processed to be physically stored in one form or another. This is another reason why a tool like Microsoft Power BI is unfit to meet our goal. One of our clear goals is to avoid storing any private medical information about our users. The final issue with these tools is the fact that they are not designed to be compatible with the E-Nabız ecosystem. A middleware between the systems provided by the Turkish government and these tools is still required, which imposes restrictions on this pipeline's availability to every citizen of Türkiye. Hipograf does not have this issue because it is designed to be a part of the E-Nabız ecosystem and be available to everyone who is a part of the Turkish health system. With the shortcomings of these tools stated, it can be concluded that general purpose data management and visualization tools are not fit to be used under requirements imposed on Hipograf.

3.1.2 Domestic Medical Information Visualization Tools

All health related information can be accessed through E-Nabız for all citizens of Türkiye. As a direct result of this, we assume that the amount of publicly available tools are not focused on private medical information. The existence of a government backed, centralized service like E-Nabız is likely to prevent the introduction of a software ecosystem consisting of comparable applications since it serves as a monolith that uniquely provides a significant number of capabilities. New competitors cannot sprout when one entity has already deeply rooted itself with practically all of the needed functionality. However, when the existing system eventually encounters a weakness, the lack of alternatives leaves the citizens with no viable options to switch to. In the case of E-Nabız, this weakness is the visualization of the medical information. E-Nabız provides most data as text; historical information is presented as simple lists, which puts little emphasis on things like passage of time between data points [11]. Hipograf serves as the missing link between the textual data and the user's comprehension. By integrating itself with E-Nabız, Hipograf will be able to access all the necessary information to provide its users with strong visualization options for their private medical information, something currently lacking in the Turkish health application ecosystem.

3.1.3 International Medical Information Visualization Tools

In the international ecosystem of medical information visualization tools, strong tools like Medihis, KeepTrackMed, CareClinic are available [12, 13, 14]. Even though these tools provide alternative or similar visualization capabilities to what is planned to be included in Hipograf, not a single one of these applications individually provides everything Hipograf plans to provide. A major problem with these applications is that they often have their own way of keeping data, relying on people uploading their own information. This goes against two important requirements we set out to achieve: Having a tool that is available to all citizens of Türkiye and temporary possession of private medical information. As mentioned before, thanks to its planned integration into E-Nabız, Hipograf will be able to provide similar or better visualization compared to the aforementioned tools without storing users' sensitive information.

3.2. Academic and Technical Analysis

The existence of aforementioned tools that provide similar visualization capabilities to Hipograf and extremely versatile general purpose visualization libraries like D3 and Plotly clearly show the feasibility of Hipograf on the academic front [10, 15, 16]. Another important part of the application is the data used in the visualization. This data is planned to come through Hipograf's integration into the Turkish health system. However, the integration process of services into the E-Nabız ecosystem is a very arduous and bureaucratic process, requiring many legal and technical documents and tests [17]. Therefore, regardless of Hipograf's feasibility on the academic front, its biggest challenge is the lengthy and complex process of integrating it into the Turkish health system. Even though Hipograf is planned to be ready for its integration, the completion of this process may not be fully achieved in the timeframe given for the senior project.

4. Verification

Verification specifically concerns the validation of fulfillment concerning targets and goals set by the non-functional requirements of the project, as specified in ISO/IEC/IEEE 12207. This section will detail how the non-functional requirements will be verified during the course of the project's software development phases.

4.1 Reliability Verification

The most efficient way to verify reliability is to monitor the hardware that Hipograf is being hosted on. The hardware servers that hold the frontend/backend web servers would need to be monitored to determine their uptime, ensuring that it stays above 95% overall for a significant period of time measurement (such as several months). Any maintenance efforts on the source servers would need to be appropriately logged to ensure that they are conducted at the appropriate off-peak times.

4.2 Performance Verification

Performance verification can be performed using readily available online resources such as Google PageSpeed Insights, GTmetrix, or WebPageTest [18]. The planned maximum time to load the main content of Hipograf is 2 seconds, and the time input delay is planned to be less than 200ms. The average hardware specifications of a computer used in healthcare institutions across Türkiye will be used as a standard reference point when conducting these tests. The website can also be stress tested, with tools such as Locust, to assess its ability to handle high traffic and ensure that performance does not degrade when many users are simultaneously connected to it.

4.3 Security Verification

Basic security for Hipograf can be done by the project team, such as getting a TLS certificate through Let's Encrypt's Certbot which will be visible and verifiable through the browser upon connecting to Hipograf [19]. Other security features can be monitored by a third party security audit. This audit is especially convenient to conduct since Hipograf will be fully open source software. Essentially, anyone can verify security details like password hashing/storage rules or medical data impermanence by looking through the code available in the repository.

4.4 Usability Verification

The detailed metrics provided in the WCAG 2.1 provide a simple yet comprehensive verification framework for meeting the usability standards. Each metric has a "Success Criterion" that determines what needs to be accomplished in order to meet it, which can be used for a general verification of usability when it is considered for every relevant metric [5].

5. References

[1] “Mevzuat Bilgi Sistemi”. https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1&MevzuatTertip=5. [Accessed: Nov 27, 2025].

[2] “IEEE/ISO/IEC 29148-2018”. https://standards.ieee.org/ieee/29148/6937/. [Accessed: Nov 27, 2025].

[3] “IEEE/ISO/IEC 12207-2017”. https://standards.ieee.org/ieee/12207/5672/. [Accessed: Nov 27, 2025].

[4] “Unified Modelling Language, v.2.5.1”. https://www.omg.org/spec/UML/2.5.1/PDF. [Accessed: Nov 27, 2025].

[5] “Web Content Accessibility Guidelines (WCAG) 2.1”. https://www.w3.org/TR/2025/REC-WCAG21-20250506/. [Accessed: Nov 27, 2025].

[6] “ICD-10 Version:2019”. https://icd.who.int/browse10/2019/en. [Accessed: Nov 27, 2025].

[7] “Sağlık Kodlama Referans Sunucusu”. https://skrs.saglik.gov.tr/. [Accessed: Nov 27, 2025].

[8] “Power BI - Data Visualization | Microsoft Power Platform” https://www.microsoft.com/en-us/power-platform/products/power-bi. [Accessed: Nov 27, 2025].

[9] “SAS Visual Analytics | SAS” https://www.sas.com/en_us/software/visual-analytics.html. [Accessed: Nov 27, 2025].

[10] “Healthcare Data Visualization: Insights for Better Decision-Making” https://binariks.com/blog/data-visualization-in-healthcare/. [Accessed: Nov 27, 2025].

[11] “e-Nabız V.2.1 Kullanım Klavuzu 2025”. https://enabiz.gov.tr/document/User_Manual.pdf. [Accessed: Nov 27, 2025].

[12] “Medihis”. https://www.medihis.com.tr/tr-TR/anasayfa. [Accessed: Nov 27, 2025].

[13] “KeepTrackMed - Personal Health Record Tracking App”. https://www.keeptrackmed.com/. [Accessed: Nov 27, 2025].

[14] “Personal Health Tracker: Symptoms, Medications, Chronic Illness | CareClinic App”. https://careclinic.io/. [Accessed: Nov 27, 2025].

[15] “D3 by Observable”. https://d3js.org/. [Accessed: Nov 27, 2025].

[16] “Plotly javascript graphing library in JavaScript”. https://plotly.com/javascript/. [Accessed: Nov 27, 2025].

[17] “USS İşleyiş”. https://rehber.enabiz.gov.tr/Home/UssIsleyis. [Accessed: Nov 27, 2025].

[18] “About PageSpeed Insights”. https://developers.google.com/speed/docs/insights. [Accessed: Nov 27, 2025].

[19] ”About Certbot”. https://certbot.eff.org/pages/about. [Accessed: Nov 27, 2025].