Bilkent University
Department of Computer Engineering

Senior Design Project
T2506
Hipograf

Analysis and Requirement Report

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

Supervisor: Prof. Dr. Uğur Doğrusöz
Instructor: Mert Bıçakçı
Instructor: İlker Burak Kurt

18.12.2025

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

Analysis and Requirement Report

Project Hipograf: Medical Data Visualization

1 Introduction

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.

1.1 Purpose

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.

2 Current Systems

2.1 General Data Management and Visualization Solutions

Two very prominent data visualization tools used in the industry are Microsoft Power BI and SAS Visual Analytics [1, 2, 3]. 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 for the purposes of medical data visualization.

2.2 Domestic Medical Information Visualization Solutions

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 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 [4]. 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.

2.3 International Medical Information Visualization Solutions

In the international ecosystem of medical information visualization tools, strong tools like Medihis, KeepTrackMed, CareClinic are available [5, 6, 7]. 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 Proposed System

Hipograf aims to help both doctors and patients across Türkiye using data visualization. Anyone in Türkiye with an E-Nabız account 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 format. Users will be able to filter and select this data using techniques familiar and intuitive to them.

3.1 Overview

Hipograf will 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 3.6.5. 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.

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 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.

3.2 Functional Requirements

3.2.1 Types of Users

Hipograf has two 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 can 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.

3.2.2 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.

3.2.2.1 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.

3.2.2.2 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.

3.2.2.3 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, 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.

3.2.3 Patients

Citizens who are mindful of their health (and how it progresses over time) but are not themselves 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.

3.3 Non-functional Requirements

3.3.1 Reliability Requirements

Hipograf is designed to be used in the medical sector, where access should be guaranteed for as long periods as possible. Utmost care should be taken to avoid presenting the user 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.

3.3.2 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, will be 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.

3.3.3 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 256-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
- 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.

3.3.4 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) [8].

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 [8].

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 [8].

There should be a color contrast of at least 4.5:1 between text and its background for optimal readability [8].

3.4 Pseudo Requirements

In addition to the functional and non-functional requirements, we impose additional requirements regarding the project.

- The frontend of Hipograf will be developed using React Typescript. Tailwind CSS and additional UI libraries that can be found in the NPM package registry.

- The backend of Hipograf shall be developed using Python, using Flask as the web framework. Additional packages shall be available from the PyPI repository.
- All software used in the development of Hipograf shall be under permissive software licenses such as MIT license, Apache 2.0 license, etc [9, 10].

3.5 System Models

3.5.1 Scenarios

These two scenarios denote two high-level examples of what a user might expect from a standard session of using Hipograf for several minutes. Below these example scenarios, we have written User Stories that provide detailed accounts of what a user might do in every facet of the application.

A medical practitioner registers for the Hipograf, creating an account. They then proceed to log-in using the account they just created. They access their dashboard and select a patient from their total list of patients, giving them a page that gives a summary of everything about the patient. They switch to the timeline page, seeing an overview of everything that this particular patient has been a part of medically for the default duration: Their blood works, diagnosis, hospital visits, radiological images, operation, vaccinations, prescriptions and vital signs. They filter out only the blood tests as they do not wish to see any other listings, and shift the timeline left and right to observe different blood tests before settling on a 6 month timeframe. They click on a particular blood test and observe some details about it before selecting another blood test, thus having selected the range of all blood tests between the two. They then proceed to select a parameter from that list (e.g, Iron) which proceeds to create a graph on the very same page, showing the fluctuations of iron in measurements across the selected timeframe.

A patient logs into Hipograf using their previously created account. They are brought to the dashboard, from where they go to the lists page because they'd like to look at their wearable data, tracking their daily steps over the last 3 weeks. They export this to the graph page, where they create another graph of their medical operations over the same time period. Evidently they would like to observe whether there is any pattern of how the medical operations affect their routine walks. Observing these two graphs side-by-side, they suddenly realize that they want to view them on the same graph. They drag one graph over the other to achieve this end, and observe the resulting graph. They then proceed to change the timeframe of the graph, which shows both parameters over the last 2 years. Wishing to see this graph in the future, they save this combined graph alongside

3.5.2 User Stories

3.5.2.1 General Management

User Story: As a medical practitioner, I want to register to Hipograf so that I can securely authenticate myself to the system in the future.

Acceptance Criteria:

- The user will provide a username and a password.
- The password will be checked to ensure it complies with the accepted standards.
- If successful, the user will be told that their registration was successful.
- The user will be redirected to the login page.

User Story: As a medical practitioner, I want to log in to Hipograf so that I can use the application.

Acceptance Criteria:

- The user will provide a name and password.
- The username and the corresponding hash of the password will be checked to make sure it is equal to the one in the database.
- If the credentials are correct, the user will be logged into their account, and redirected to their home page.
- If the credentials don't match the ones in the database, the user will be shown an incorrect credentials message and prompted to try again.

User Story: As a medical practitioner, I want to log out of Hipograf so that I can close my session.

Acceptance Criteria:

- The user will click the "log out" button.
- If successful, the session of the user will end, and they will be logged out.
- If the server shuts down or experiences another issue, the valid cookies issued by the server will reset.

User Story: As a medical practitioner, I want to be able to delete my account so that Hipograf servers will wipe my data.

Acceptance Criteria:

- The user will click the "Delete Account" button.

- On success, the information of the user, such as their login credentials and data viewing preferences, will be wiped from the Hipograf servers.
- On failure, a failure message will be shown, prompting the user to try again.

User Story: As a medical practitioner, I want to be able to change my password so that I can change my login credentials.

Acceptance Criteria:

- While logged in, the user will click the “Change Password” button, and be prompted to enter their new password, and confirm it.
- On success, the password of the user will be changed, and the user will be logged out.
- On failure, a failure message will be shown, prompting the user to try again. Failure can happen due to network connectivity or an unsuitable password.

User Story: As a medical practitioner, I want to be able to reset my password in the login page, so that I can change it when I forget my password.

Acceptance Criteria:

- While logged out, the user will click the “Forgot password” button, and will be prompted to enter their registered email. After, a link will be sent to the user’s registered e-mail, where they can reset their password.
- On success, the user’s password will change and the user will be redirected back to the login page.
- On failure, a failure message will be shown, prompting the user to try again. Failure can happen due to the absence of an account related to the given email, or connectivity issues.

User Story: As a medical practitioner, I want to be able to clear my data preferences so that I can reset the information displayed in my dashboard to its default state.

Acceptance Criteria:

- While logged in, the user will click the “Clear preferences” button.
- On success, the data visualization preferences will be reset.
- On failure, a failure message will be shown.

3.5.2.2 Patient Selection System

User Story: As a medical practitioner, I want to be able to list all patients appointed to me so that I can select a patient and access their medical history.

Acceptance Criteria:

- The system shall display a list containing the names of all patients who have granted me access to their medical records.
- Each patient name in the list must be selectable/clickable.

User Story: As a medical practitioner, I want to access a comprehensive, single page for a selected patient, so that I can quickly review their entire medical history and current status.

Acceptance Criteria:

- Upon selecting a patient from the list, I shall be navigated to a detailed patient profile page.
- This page must clearly display Basic Personal Information (Name, Birthdate, City of Residence, etc.).
- This page must display a unified, filterable Medical Timeline compiling all major events.
- The page must include separate sections or widgets for Recent Blood Work, Medications, Hospital Visits, Diagnoses, Operations, Radiology Images, and Vital Signs data from wearables.

3.5.2.3 List Management System

3.5.2.3.1 Blood Work Subsystem

User Story: As a medical practitioner, I want to view and analyze all a patient's medical test results (blood work and radiology), so that I can monitor specific physiological changes and health trends over time.

Acceptance Criteria:

- I shall be able to view a list of all blood work instances and all radiology images.
- I must be able to filter blood work instances by criteria such as the presence of a specific parameter and the time/date the test was performed.
- I must be able to select a single blood work instance to view a detailed page showing the test date and all measured parameters.
- From the detailed blood work view, I must be able to click on a specific parameter to display a graph showing all historical measurements of that parameter across all blood work instances.

3.5.2.3.2 Diagnosis Subsystem

User Story: As a medical practitioner, I want to view and filter a patient's past diagnoses, so that I can analyze their complete health profile and history of conditions.

Acceptance Criteria:

- I shall be able to view a list of all past and current diagnoses.
- I must be able to filter the diagnoses list using criteria such as whether the diagnosis is chronic or acute, the diagnosis's name/category, or the time frame when the diagnosis was given.

3.5.2.3.3 Hospital Visit Subsystem

User Story: As a Medical Practitioner, I want to list all of a patient's hospital visits and apply filters, so that I can review past acute care episodes and dates of admission.

Acceptance Criteria:

- I shall be able to view a list of all documented hospital visits.
- I must be able to filter the list of hospital visits by criteria, such as the date of admission.

3.5.2.3.4 Radiological Image Subsystem

User Story: As a medical practitioner, I want to view, filter, and access detailed information about a patient's radiological images, so that I can review the findings from medical imaging procedures.

Acceptance Criteria:

- I shall be able to view a list of all radiological images (X-rays, CTs, MRIs, etc.).
- I must be able to filter this list by the type of imaging (e.g., "CT Scan") and the date it was taken.
- Selecting an image from the list must navigate me to a page with detailed information about that specific radiological exam (e.g., full report, findings).

3.5.2.3.5 Operation Subsystem

User Story: As a medical practitioner, I want to list all of a patient's medical operations and apply various filters, so that I can understand their surgical history.

Acceptance Criteria:

- I shall be able to view a list of all recorded medical operations/surgeries.

- I must be able to filter the list of operations using various relevant criteria (e.g., date of operation, type of operation).

3.5.2.3.6 Vaccination Subsystem

User Story: As a medical practitioner, I want to list all of a patient's past vaccinations so that I can be informed and avoid repeat doses of vaccines.

Acceptance Criteria:

- I shall be able to view a list of all past vaccinations.
- I must be able to filter the list of vaccines using the date of the vaccine administered, and the contents of the vaccine.

3.5.2.3.7 Prescription Subsystem

User Story: As a medical practitioner, I want to list all of a patient's past and current prescriptions so that I can be informed and avoid repeat doses or conflicting prescriptions.

Acceptance Criteria:

- I shall be able to view a list of all past and current prescriptions.
- I must be able to filter the list of prescriptions by date, and the active chemical within the prescriptions.

3.5.2.3.8 Wearable Data Subsystem

User Story: As a medical practitioner, I want to list all of a patient's wearable data, if they have opted in to list them, so that I can be more informed about my patient's medical status.

Acceptance Criteria:

- I shall be able to view a patient's past wearable data.
- I must be able to list the wearable data by date, and relevant wearable data class.

3.5.2.4 Timeline Management

User Story: As a medical practitioner or a patient, I want to be able to choose the categories of kinds of events shown so that I can focus on events that are currently important to me.

Acceptance Criteria:

- The user will have access to a menu containing the currently available event categories on the same page as the timeline.
- The user will be able to toggle on or off these categories through the menu.
- The toggled off categories will be removed from the timeline.

User Story: As a medical practitioner or a patient, I want to be able to choose a timeframe for the timeline so that I can focus on events that took place in that time window.

Acceptance Criteria:

- The user will be able to input a start date and end date for the timeline.
- These dates will decide the earliest and latest dates that will be shown on the timeline, respectively.

User Story: As a medical practitioner or a patient, I want to be able to use the dedicated resize buttons or key combinations to change the length of the timeframe of the timeline so that I can prevent cluttering that can be caused by an accumulation of a high number of events in a timeframe.

Acceptance Criteria:

- The user will be able to zoom in and zoom out of the timeline.
- If the user zooms in, the earliest date shown on the screen will increase and the latest date shown on the screen will decrease, showing a narrower date interval on the screen.
- If the user zooms out, the earliest date shown on the screen will decrease and the latest date shown on the screen will increase, showing a wider date interval on the screen.

User Story: As a medical practitioner or a patient, I want to be able to use the scroll wheel or a key combination to change the earliest and the latest dates shown on the timeline while keeping the length of the timeframe constant so that I can reach all events on the timeline even if I resized the timeline.

Acceptance Criteria:

- The user will be able to scroll the timeline horizontally towards either direction.
- If the timeline is scrolled to the right, both the earliest and the latest date shown on the screen will increase.
- If the timeline is scrolled to the left, both the earliest and the latest date shown on the screen will decrease.
- In both cases, the period of time shown on screen will be kept constant.

User Story: As a medical practitioner or a patient, I want to be able to click on events shown on the timeline so that I can get detailed information about that specific event.

Acceptance Criteria:

- The user will click on a specific event stationed on the timeline.
- An info box connected to the chosen event will appear above or below depending on the circumstances of the timeline.
- This info box will contain detailed information related to the chosen event.

User Story: As a practitioner or a patient, I want to be able to save the currently applied filtering and timeline viewing options as a preference so that I can retrieve it later and apply it instantly, preventing unnecessary burden.

- The user save will click to save their current preferences.
- They will be asked to give the preset a name.
- The currently applied customization choices will be saved under the given name.

User Story: As a medical practitioner or a patient, I want to be able to load a preference set from a list I have previously saved before so that I would not need to reconfigure every option according to my needs everytime I login to the system.

Acceptance Criteria:

- The user opens the preset menu that lists all their presets.
- The user chooses the preset they would like to apply
- The timeline is adjusted according to the details of the chosen preset.

User Story: As a medical practitioner or a patient, I want to be able to transfer/export the choices I made on the timeline page to the graphs page so that I can examine the corresponding graphs in more detail if I deem it necessary.

- The user chooses the export to graphs option.
- The user is transferred to the graphs page where their timeline choices have already been applied to the graphs.

3.5.2.5 Graph Management

User Story: As a medical practitioner or a patient, I want to be able to view my medical data in a singular graphical form so that I can access my data

Acceptance Criteria:

- The user should be able to create a graph for a set of data values of their preference, seeing both the data and the threshold values.
- The user should be able to change the timescale that the data is graphed on.
- The user should be able to change the data type being graphed to obtain a different plot on the same graph.

User Story: As a medical practitioner or a patient, I want to be able to create individual graphs that represent multiple different sets of data on a single graph so that I can compare and contrast their change relative to one another over time.

Acceptance Criteria:

- The user should be able to select multiple datasets when the graph is being created.
- The user should be able to select a timescale that fits for both of them on the same graph.
- The compounded data should be automatically normalized so that I can view the graphs with relative ease.

User Story: As a medical practitioner or a patient, I want to be able to edit, scale and move the graphs that represent my data so that I can gain a holistic understanding of the interaction of datasets.

Acceptance Criteria:

- The user should be able to edit graphs to change the timescale setting or the data being graphed.
- The user should be able to resize the graphs around the screen.
- The user should be able to move the graphs around on the screen without being limited to specific spots on the screen.

User Story: As a medical practitioner or a patient, I want to be able to combine and split graphs to combine and separate the data they are representing in a united space.

Acceptance Criteria:

- The user should be able to combine existing graphs to result in graphs that graph the data simultaneously.
- The user should see an overview that automatically normalizes graph data.
- The user should be able to split up these graphs again to once again display them separately.

User Story: As a medical practitioner or a patient, I want to be able to save my preferences for a graph so that I can refer to them at a later date.

Acceptance Criteria:

- The user should be able to save their preferences for time or data type, across single or multivariate graphs, in the local database.
- The user should not have the option of saving any medical data.

User Story: As a medical practitioner or a patient, I want to be able to restore previously saved graph preferences so that I can receive a familiar overview without having to consistently reconfigure it.

Acceptance Criteria:

- The user should be able to reproduce the previously saved data onto the graphing interface.
- The user should be able to do this consistently, as many times as is required.
- The user should have the option of editing and deleting preferences in addition to restoring them.

3.6.2 Use-Case Model

Actors have been color-coded separately for clarity and readability in the diagrams.

3.6.2.1 Level 0 Use Case Diagrams

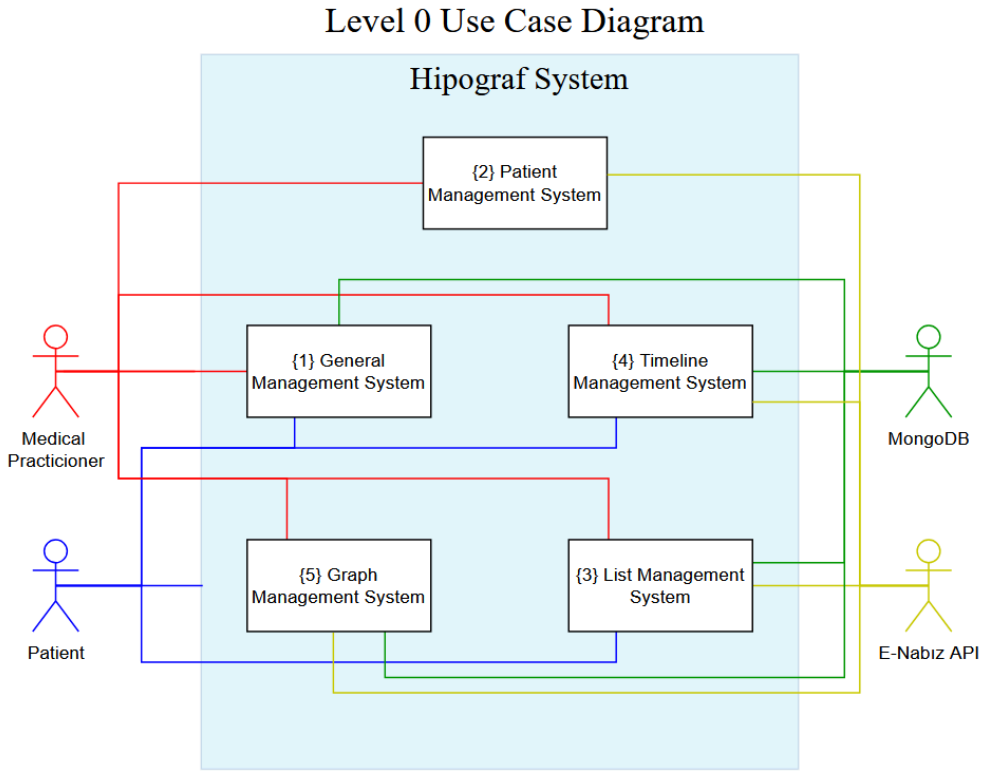

Fig. 1. Level 0 Use Case Diagram of the whole system.

3.6.2.2 Level 1 Use Case Diagrams

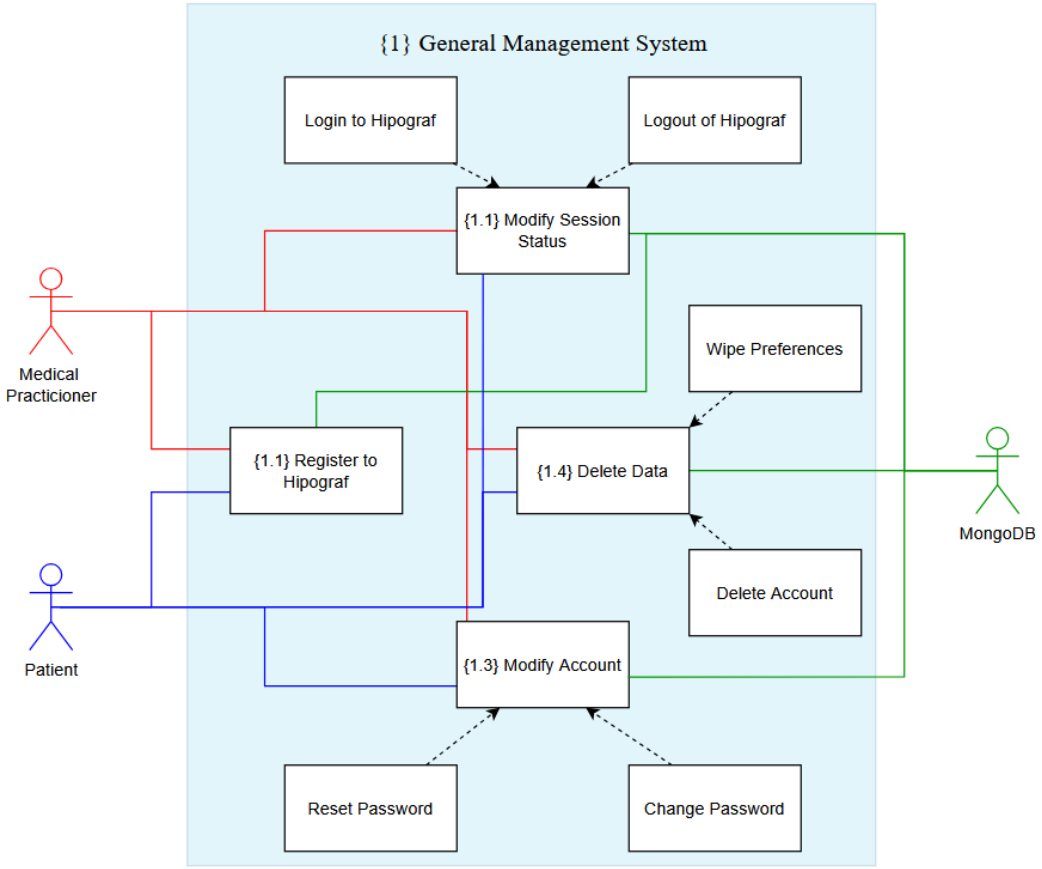

Fig. 2. Level 1 Use Case Diagram of the General Management System.

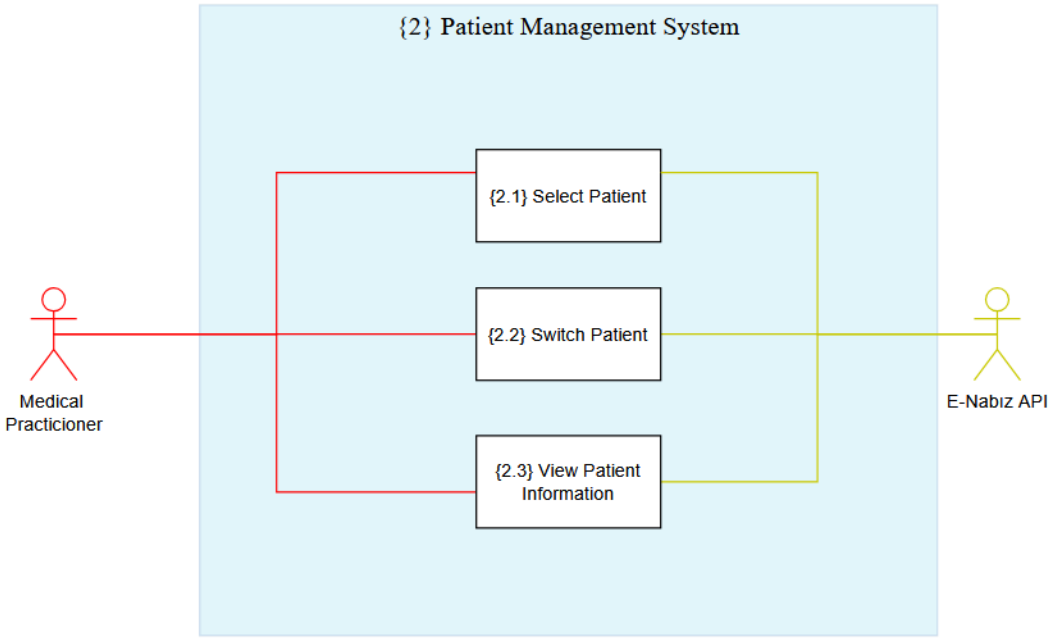

Fig. 3. Level 1 Use Case Diagram of the Patient Management System.

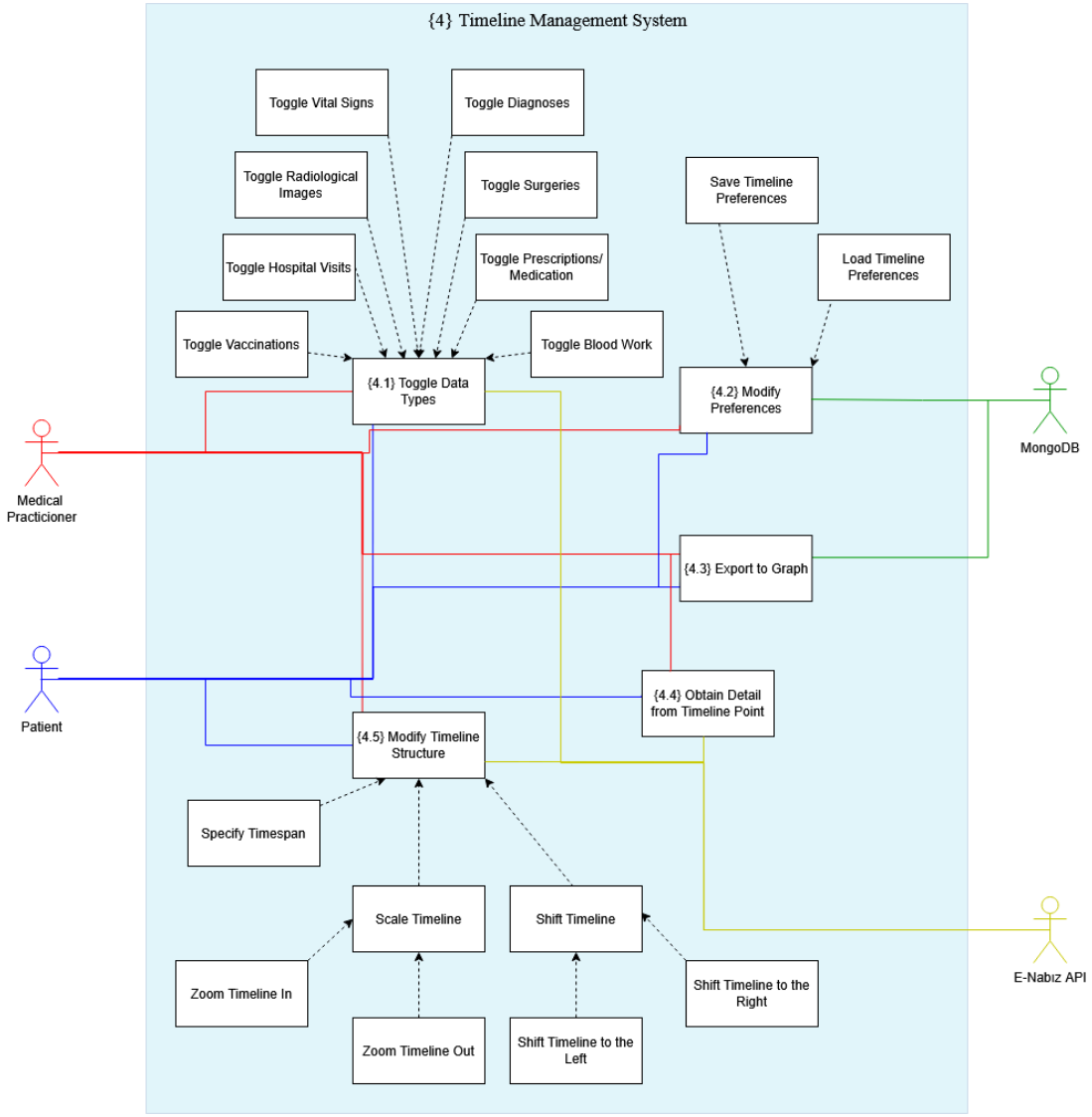

Fig. 4. Level 1 Use Case Diagram of the Timeline Management System.

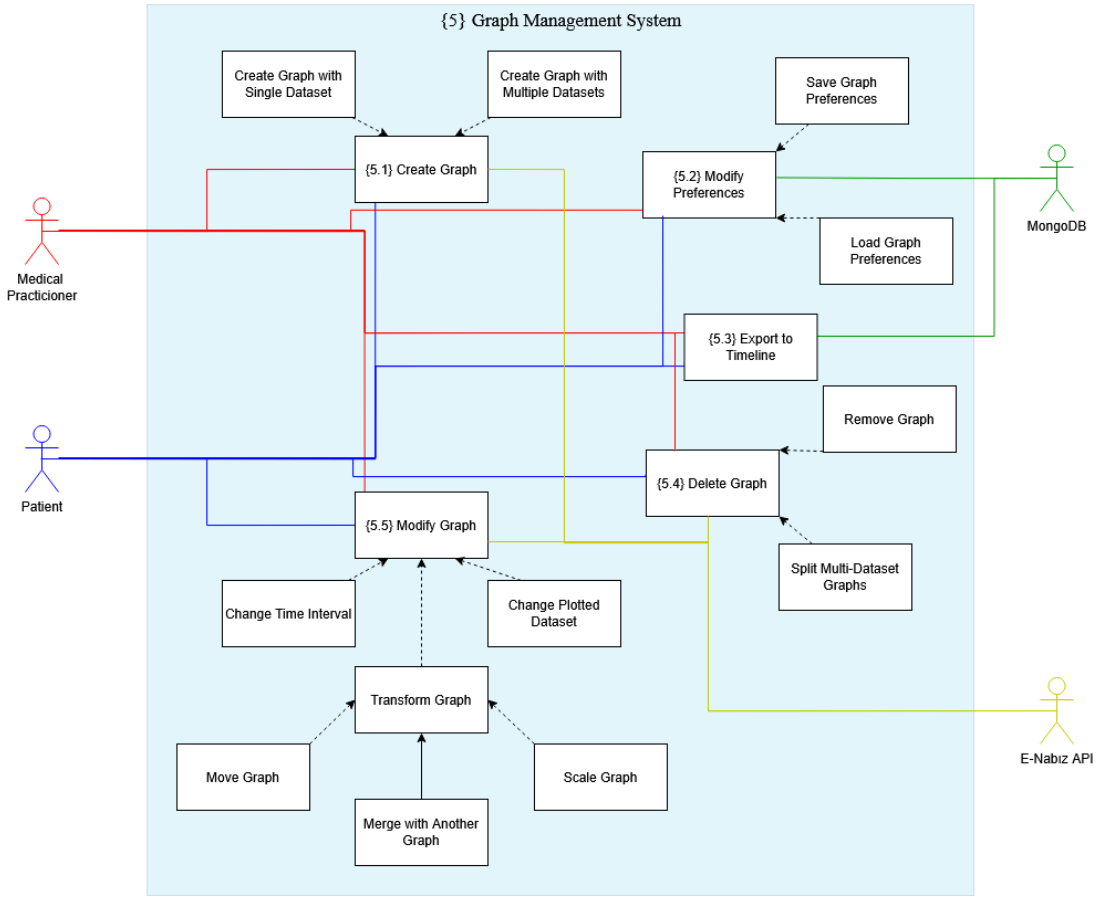

Fig. 5. Level 1 Use Case Diagram of the Graph Management System.

3.6.3 Object and Class Model

3.6.3.1 Object Diagram

The diagram shown in Figure 6 describes how the medical data will be stored in the Hipograf database.

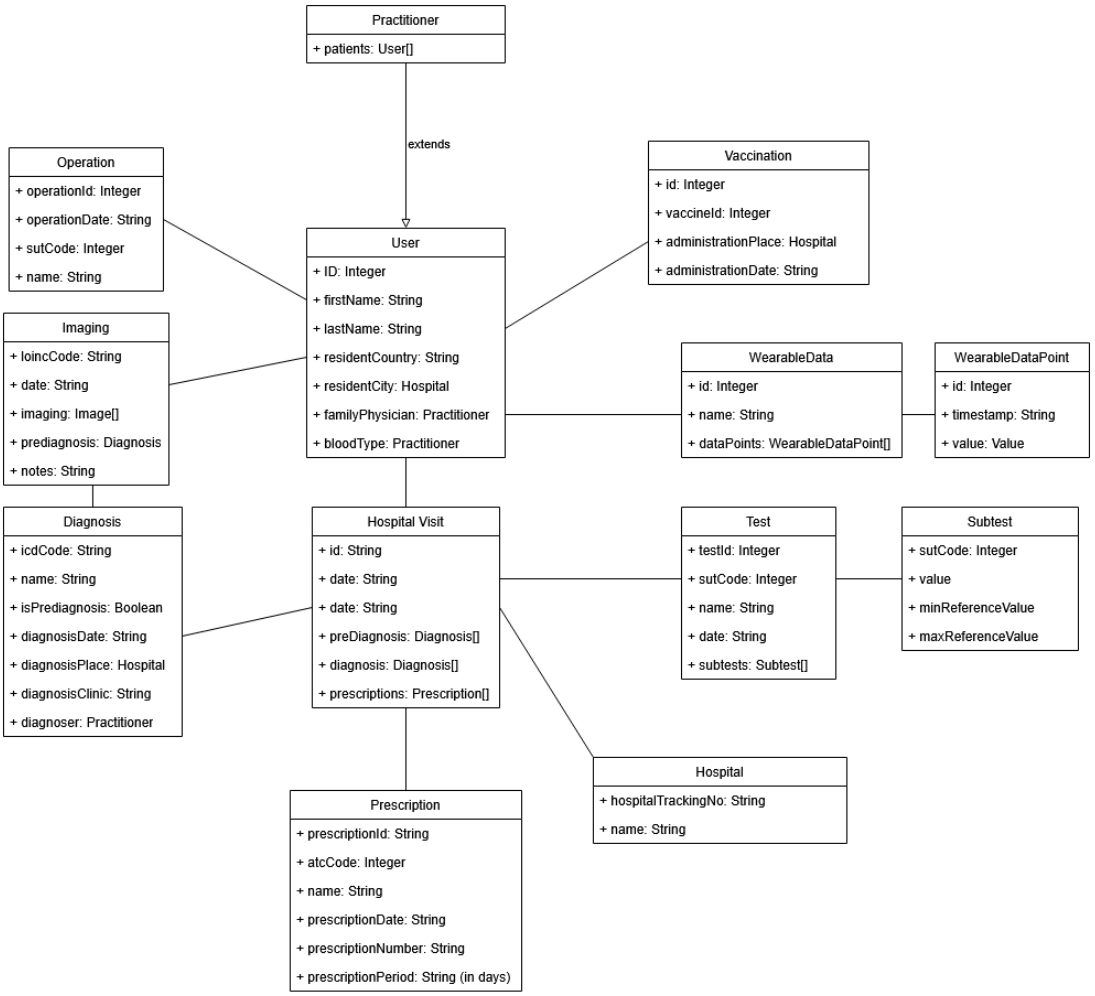

Figure 6. Object Diagram describing the database model of the project.

3.6.3.2 Class Diagram

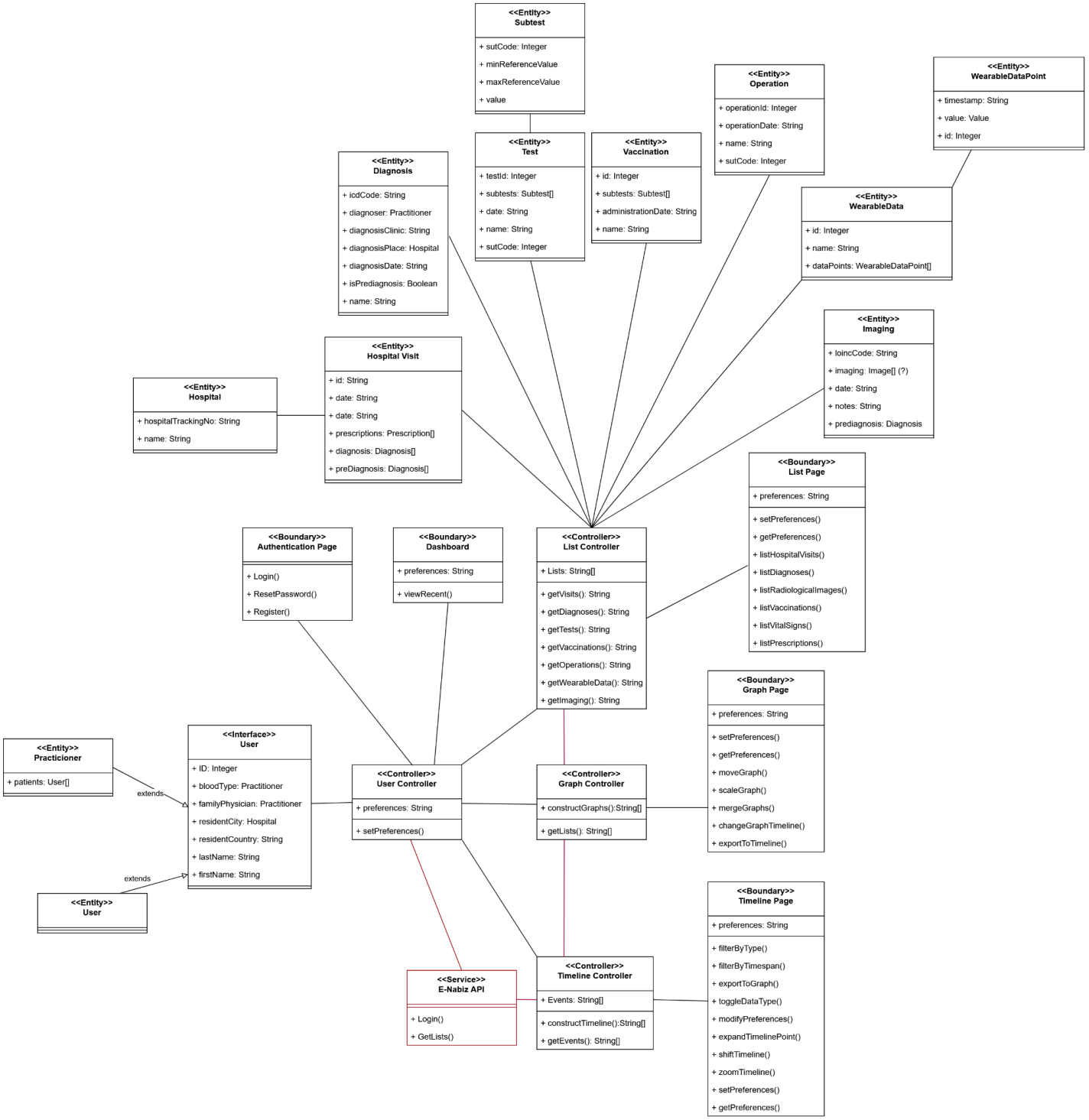

Fig. 7. Class Diagram for Hipograf.

3.6.4 Dynamic Models

3.6.4.1 Sequence Diagrams

Register

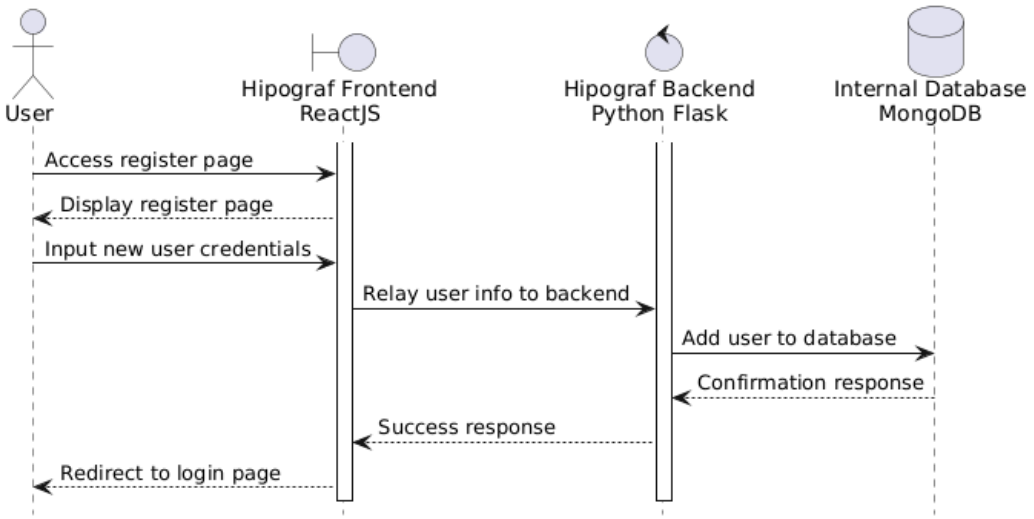

Fig. 8. Sequence Diagram for User Registration.

Login

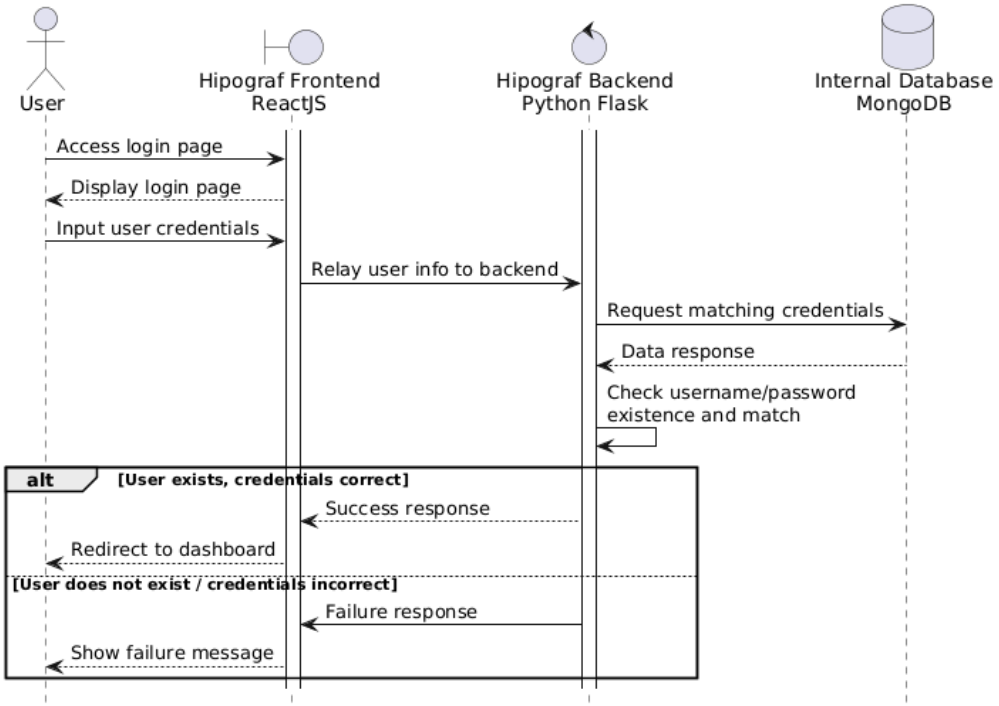

Fig. 9. Sequence Diagram for User Login.

Logout

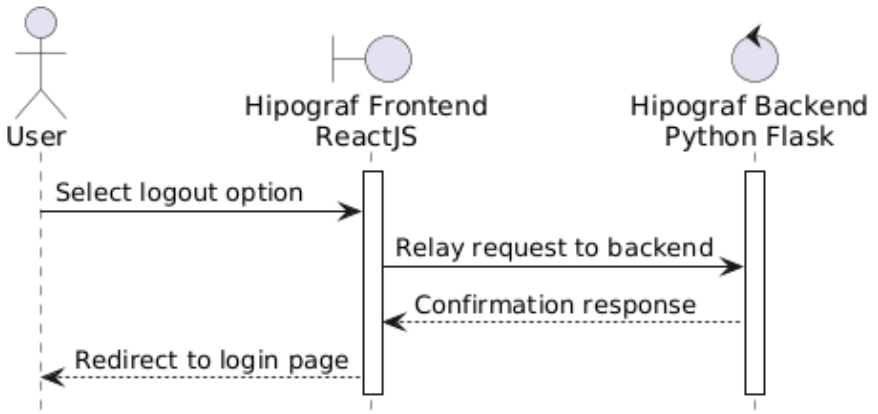

Fig. 10. Sequence Diagram for User Logout.

List Page Access

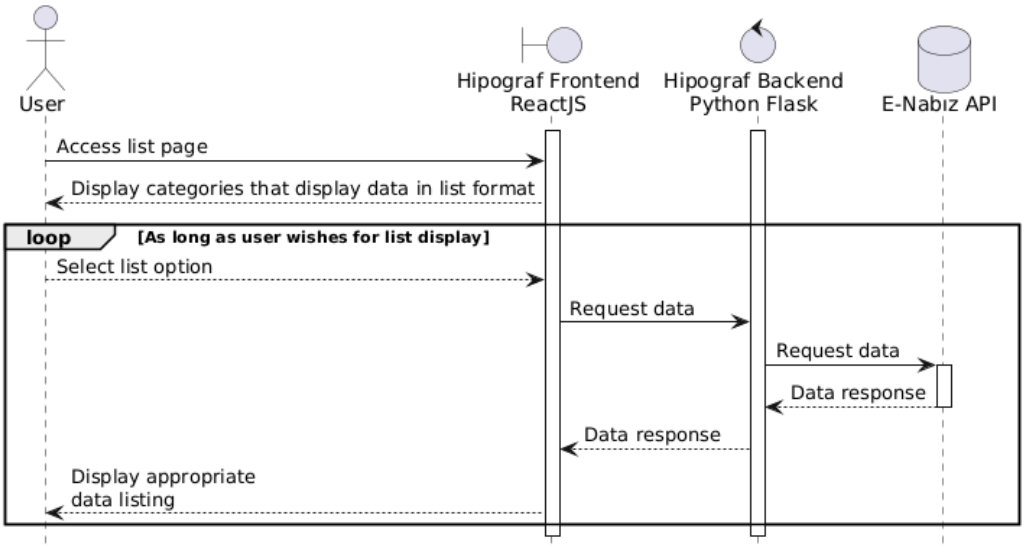

Fig. 11. Sequence Diagram for Initial Access of a List Page.

List Filtering

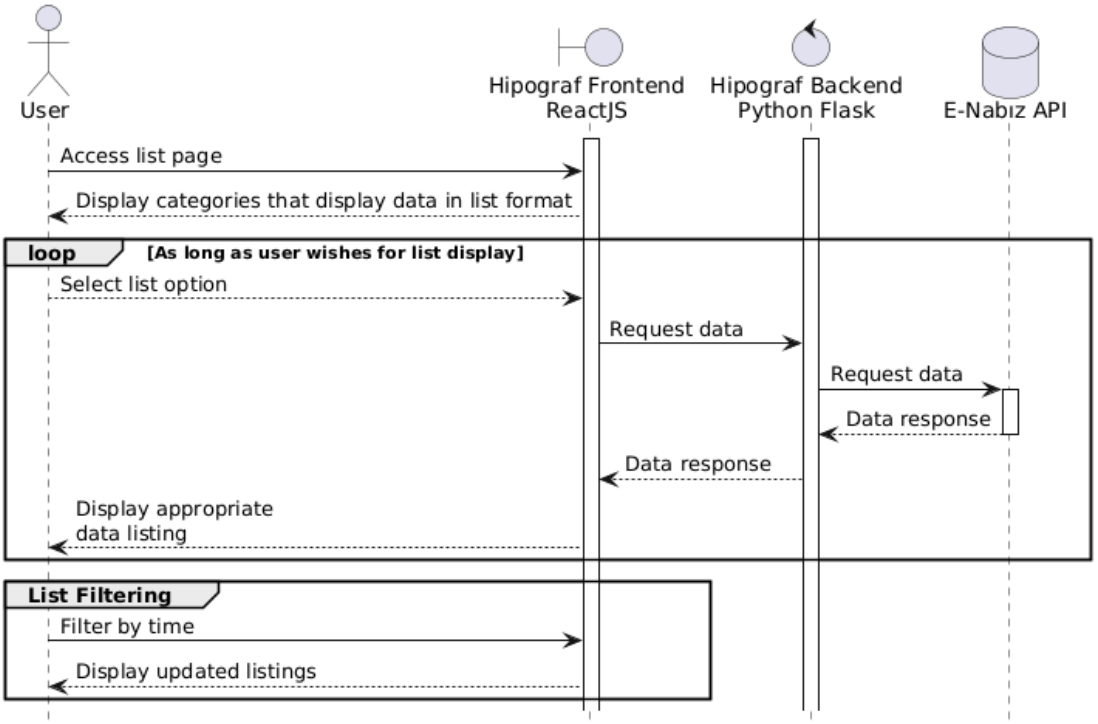

Fig. 12. Sequence Diagram for Data Filtering on the List Page.

Timeline Page Access

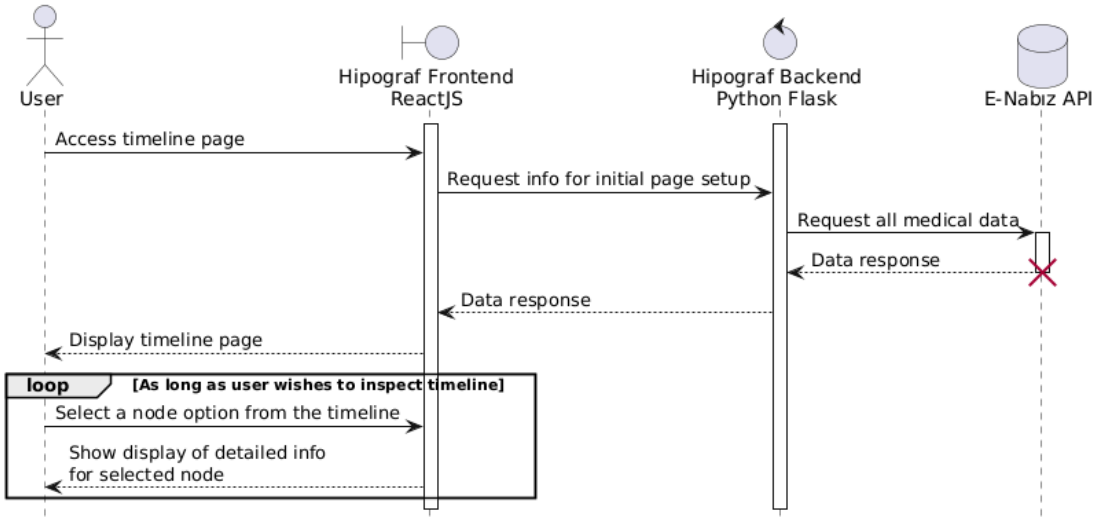

Fig. 13. Sequence Diagram for Initial Access of a Timeline Page.

Timeline-to-Graph

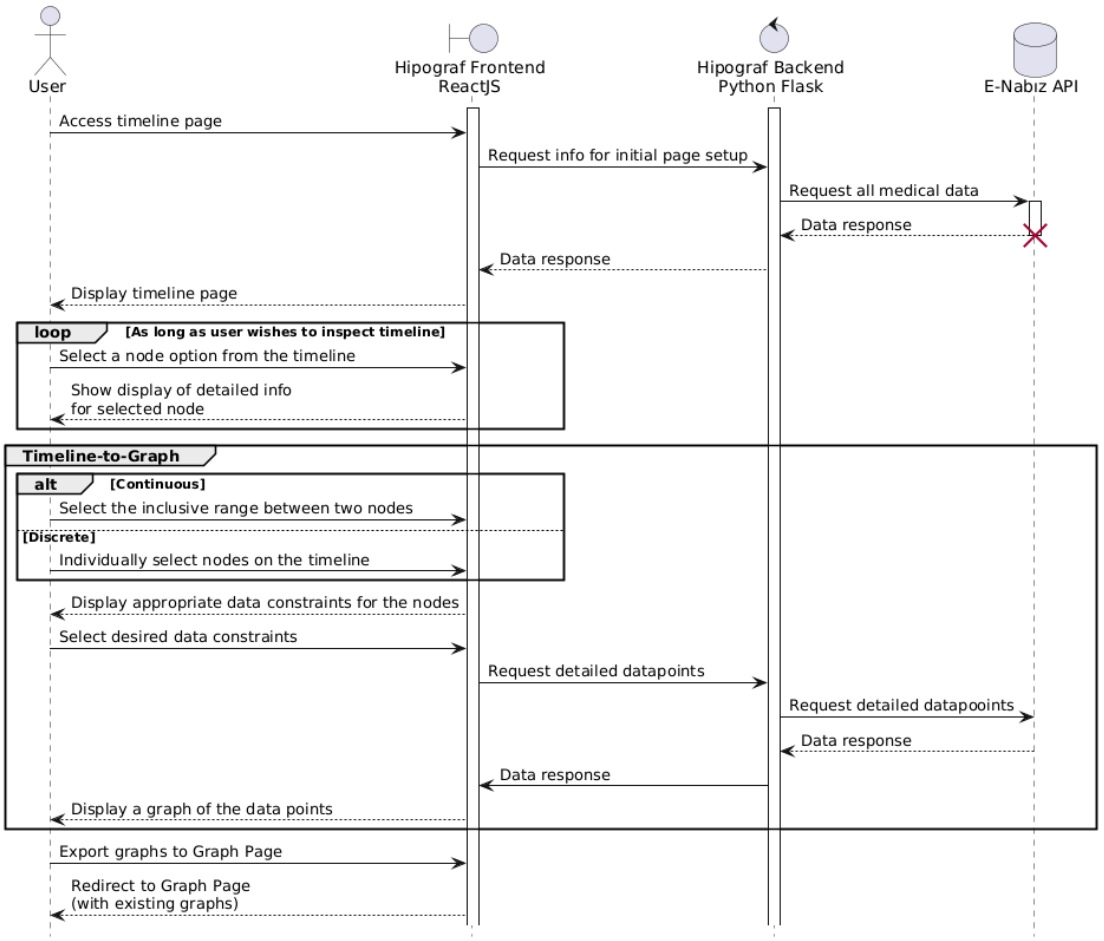

Fig. 14. Sequence Diagram for Graph Creation on the Timeline Page.

Timeline Transformations

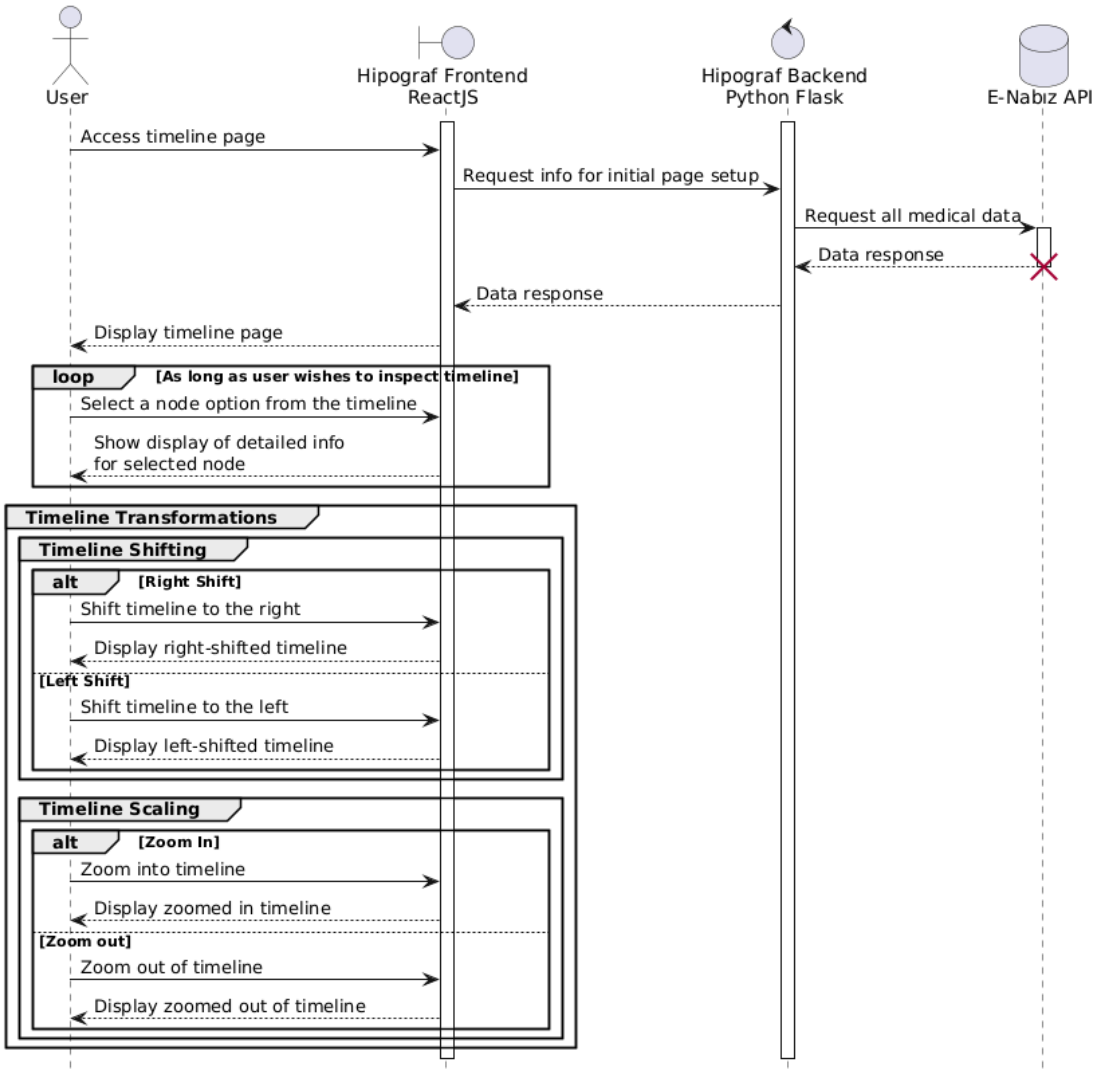

Fig. 15. Sequence Diagram for Transformations on the Timeline Page.

Graph Page Access

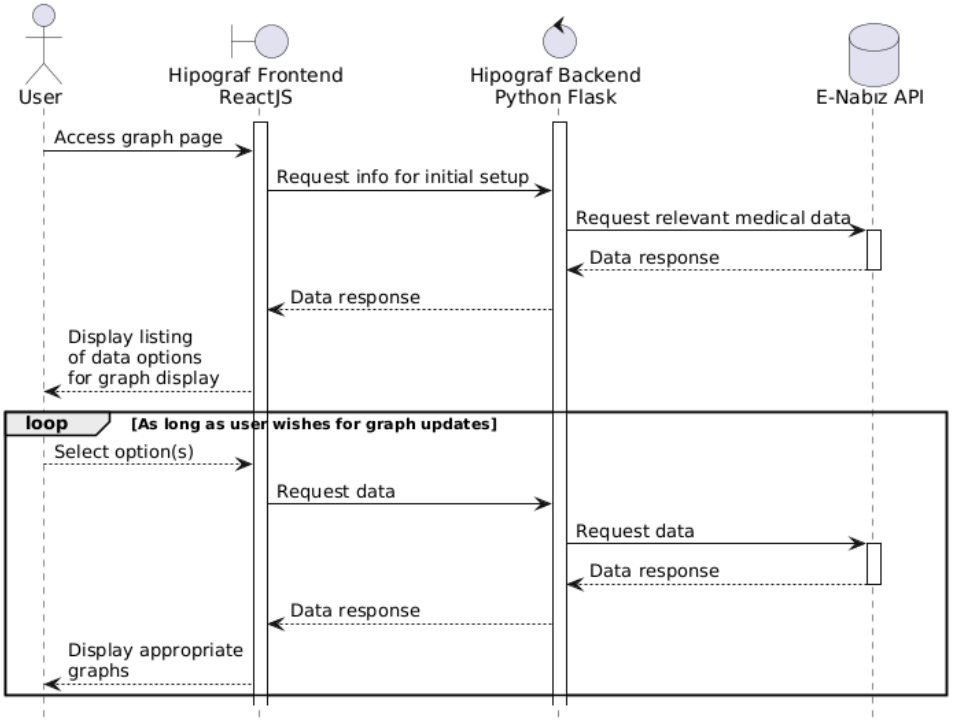

Fig. 16. Sequence Diagram for Initial Access of a Graph Page.

Graph Transformations

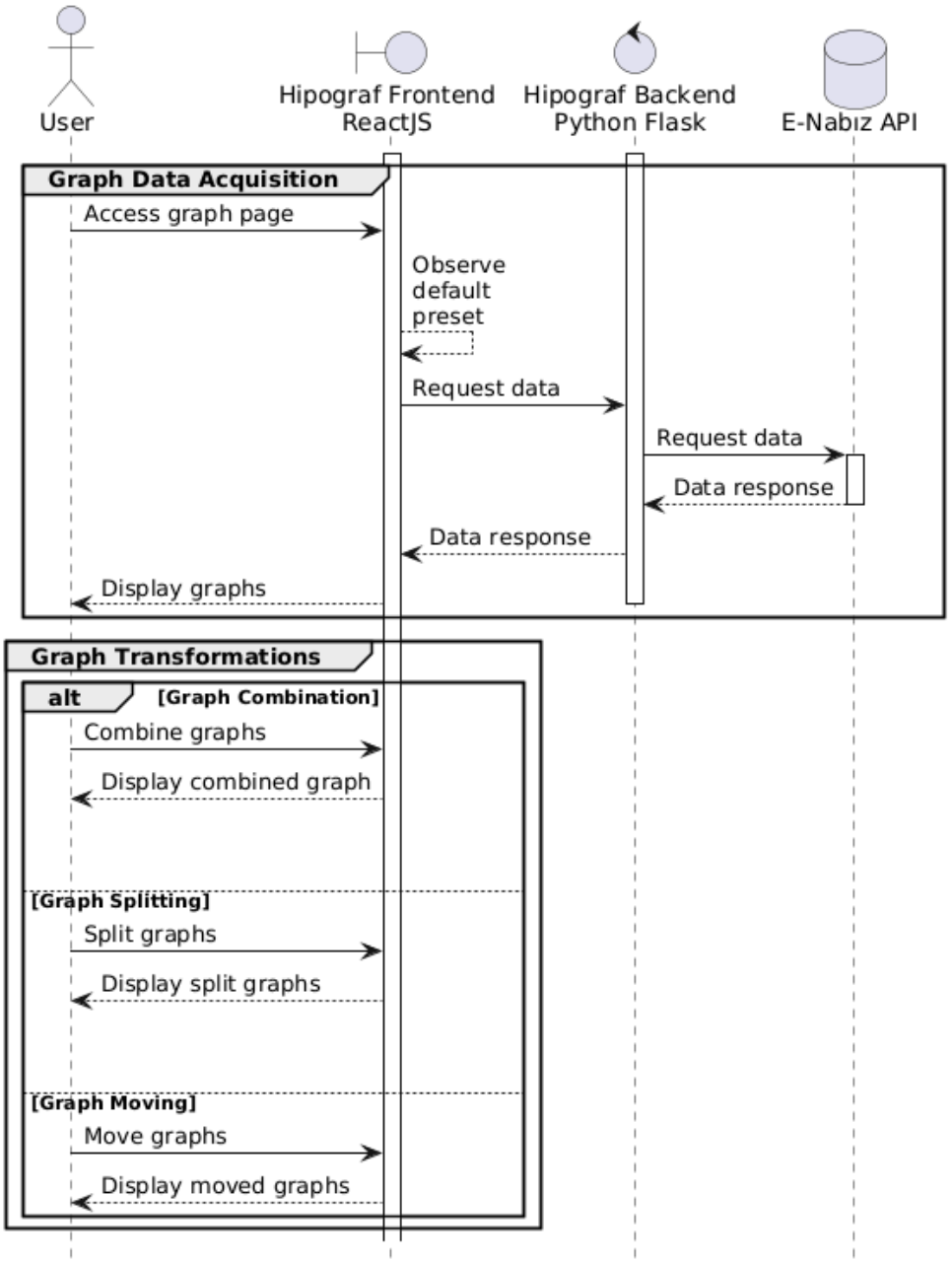

Fig. 17. Sequence Diagram Transformations on the Graph Page.

Graph Preset Saving

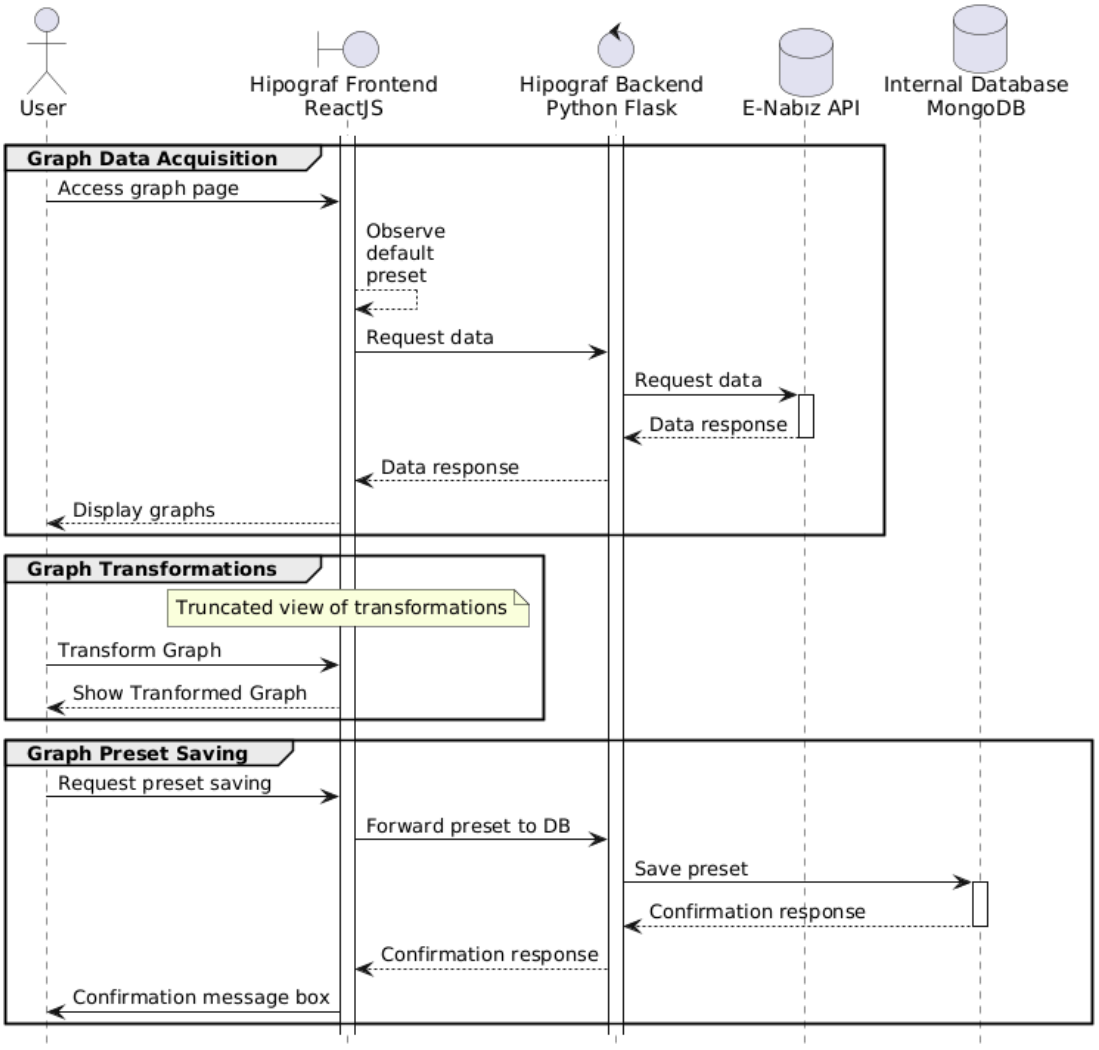

Fig. 18. Sequence Diagram Transformations Graph Preset Saving.

3.6.4.2 Activity Diagrams

Saving preference

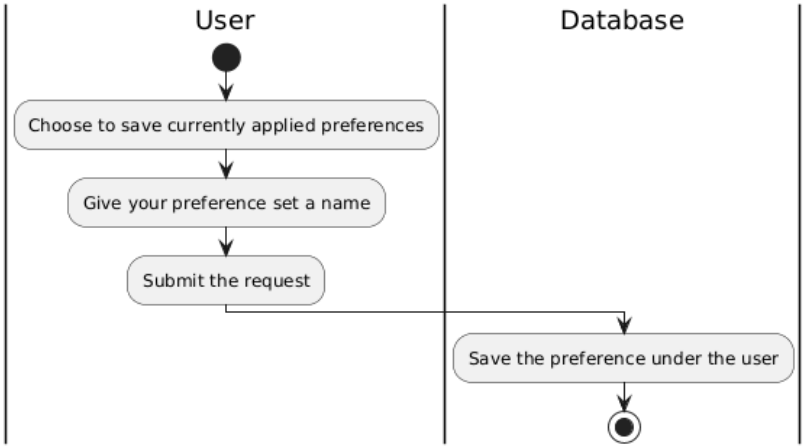

Fig. 19. Activity Diagram for Preference Saving

Loading preference

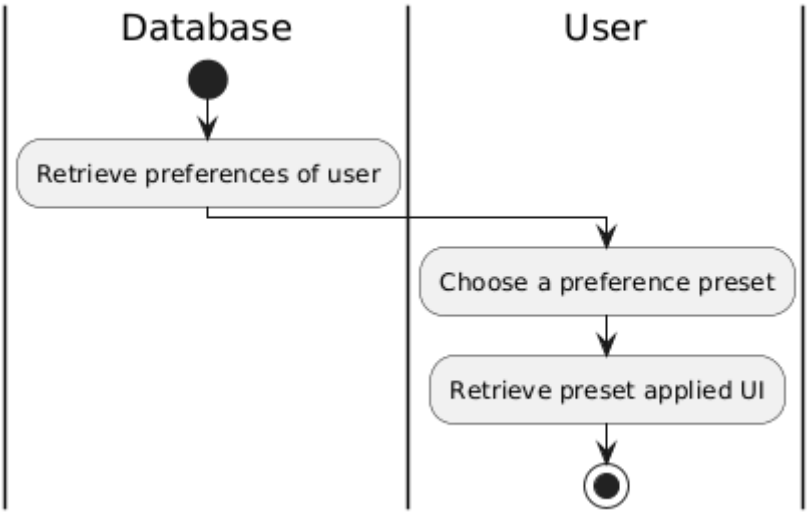

Fig. 20. Activity Diagram for Preference Loading

Login

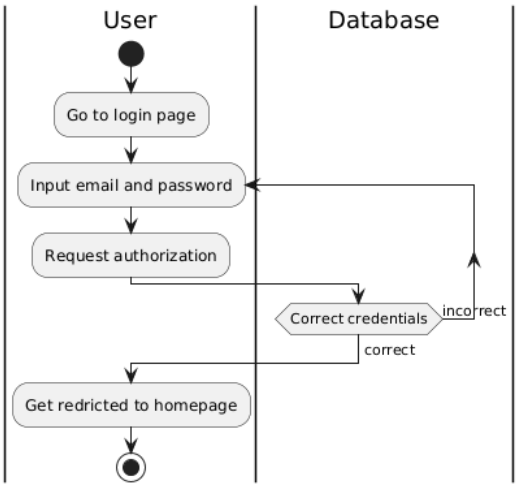

Fig. 21. Activity Diagram for Hipograf Login

Reset password

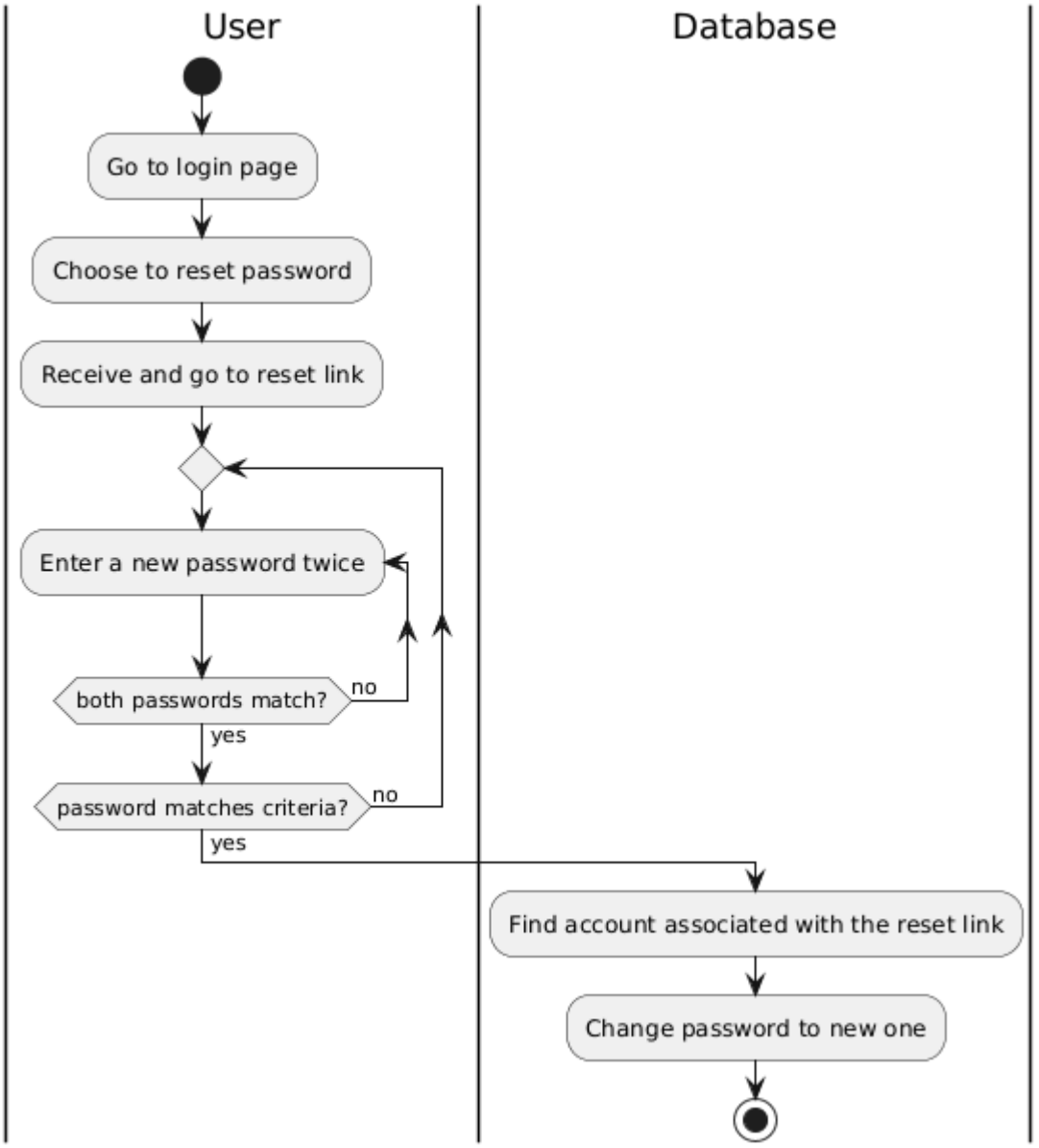

Fig. 22. Activity Diagram for Password Reset

Register

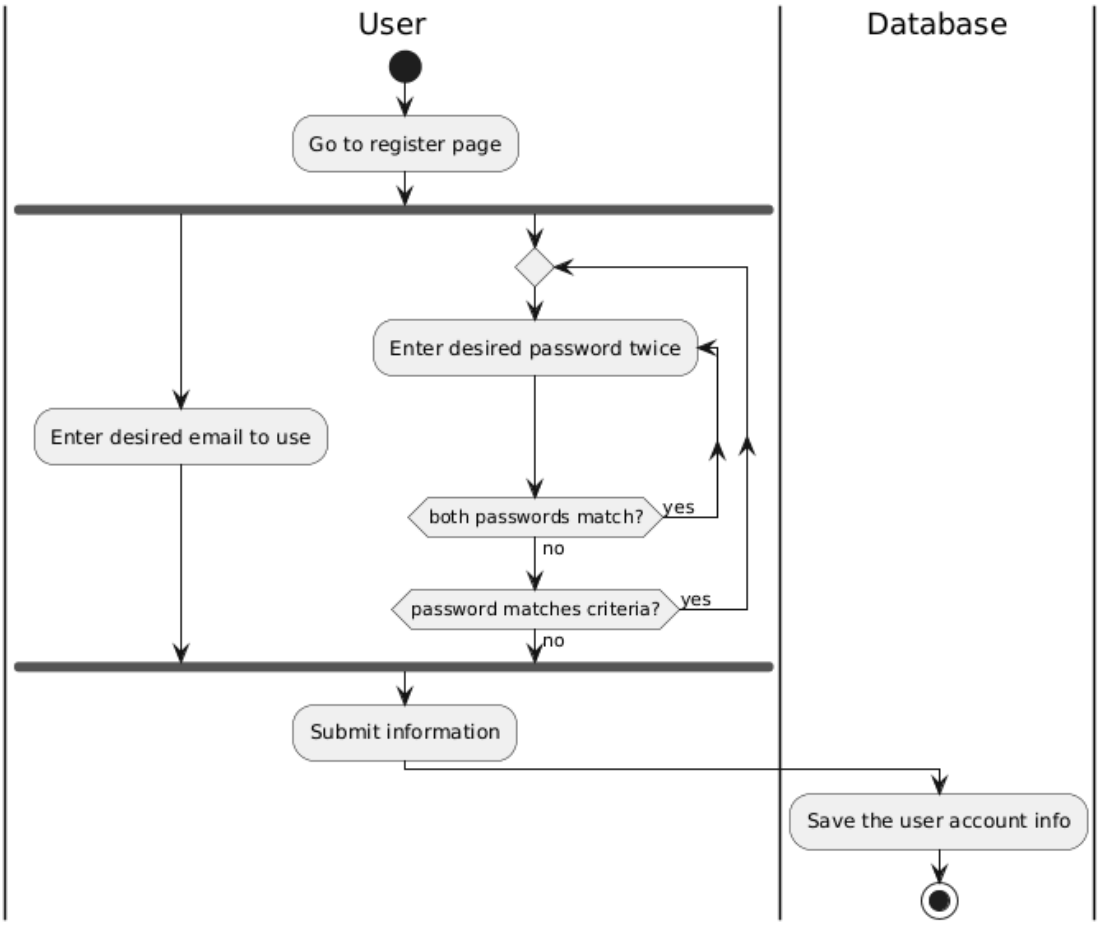

Fig. 23. Activity Diagram for Registration

Doctor choosing a patient

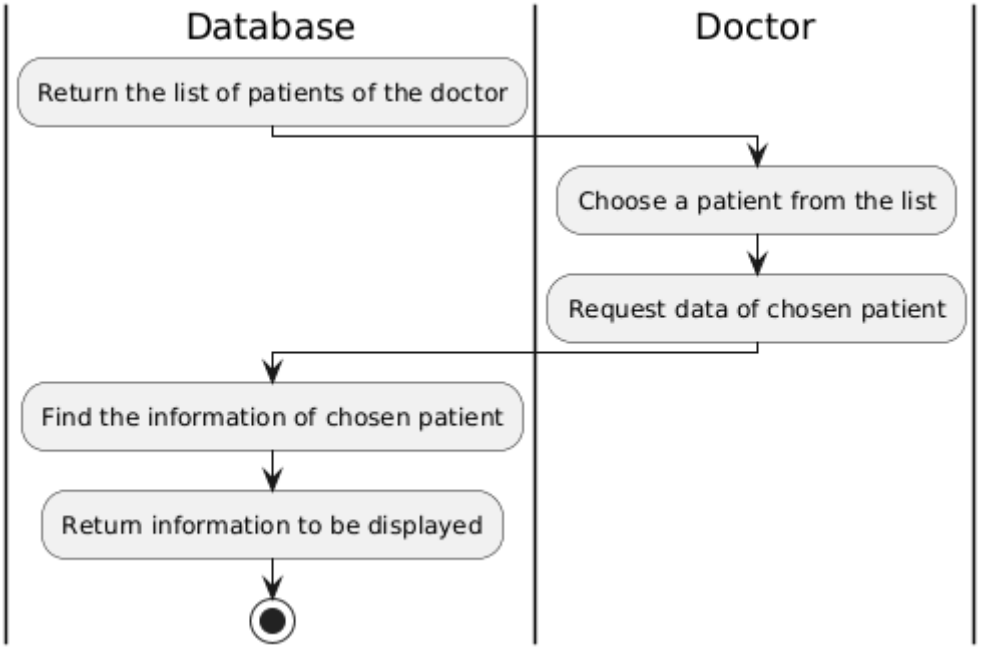

Fig. 24. Activity Diagram for Patient Selection

Filter timeline

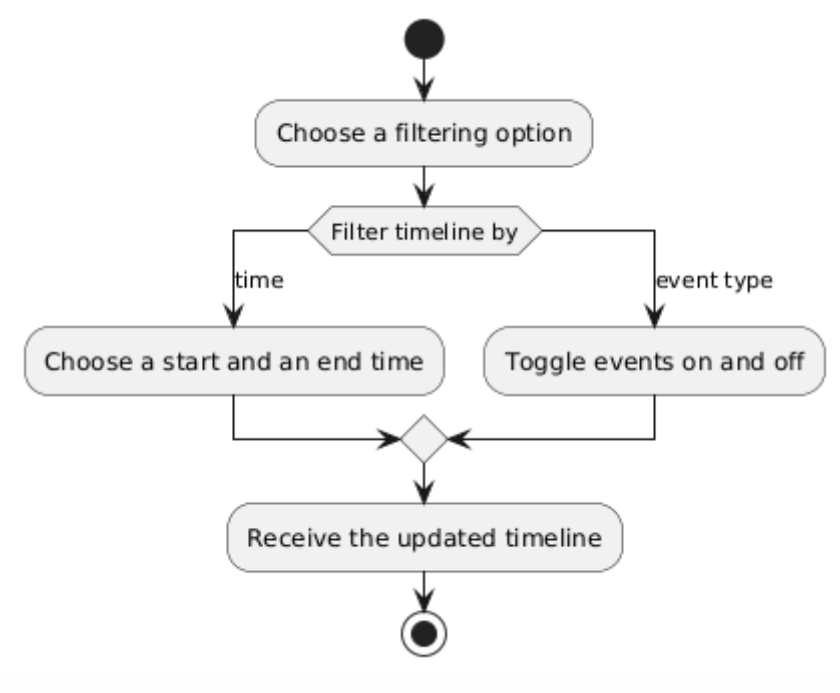

Fig. 25. Activity Diagram for Timeline Filtering

Visual formatting of timeline

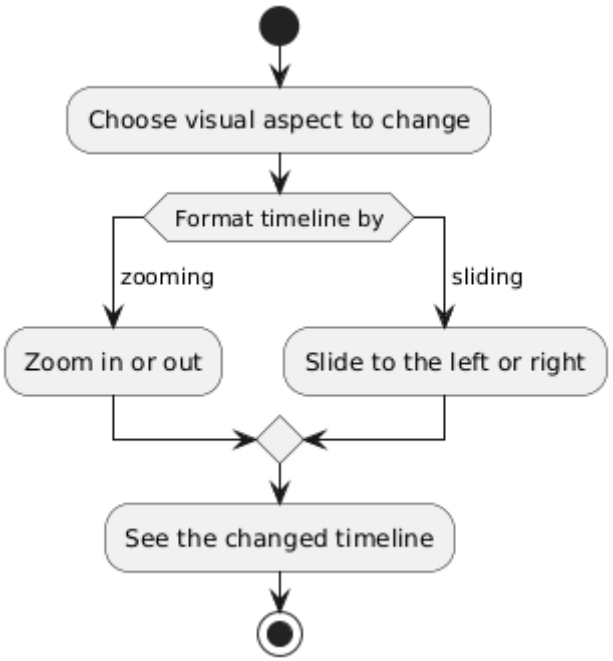

Fig. 26. Activity Diagram for Timeline Visual Formatting Timeline

Timeline graph creation

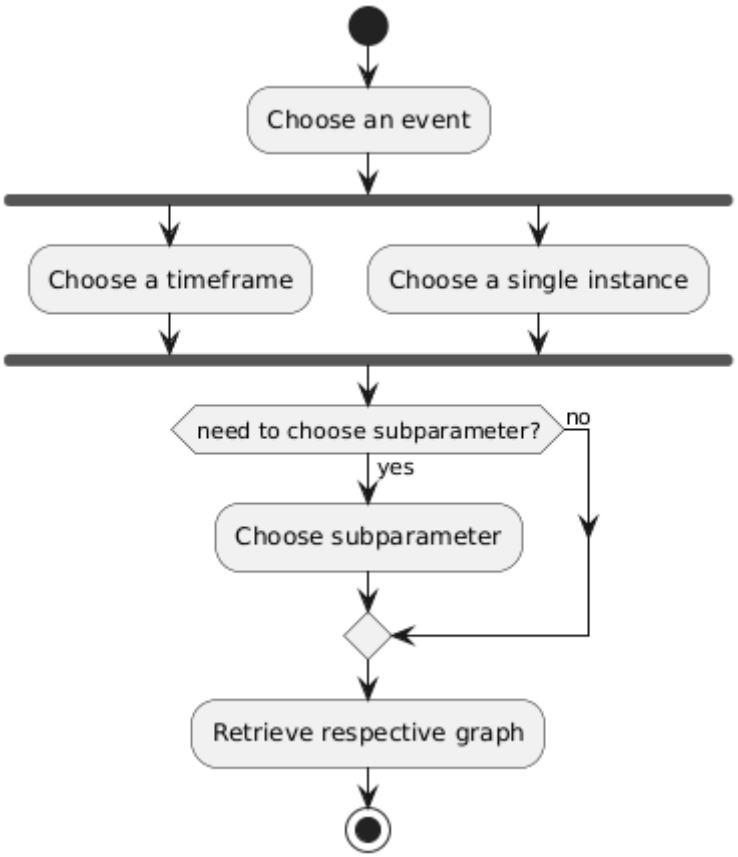

Fig. 27. Activity Diagram for Timeline Graph Creation

Graph creation

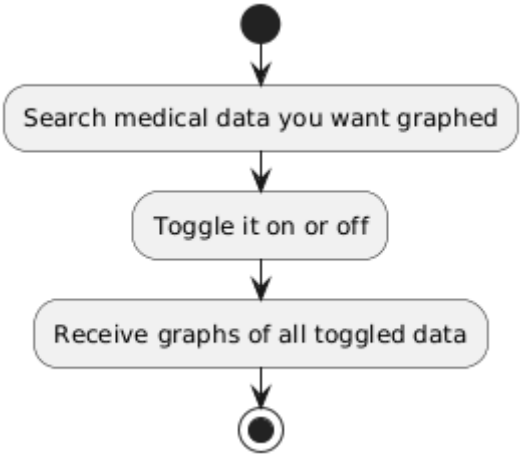

Fig. 28. Activity Diagram for Graph Creation

Graph customization

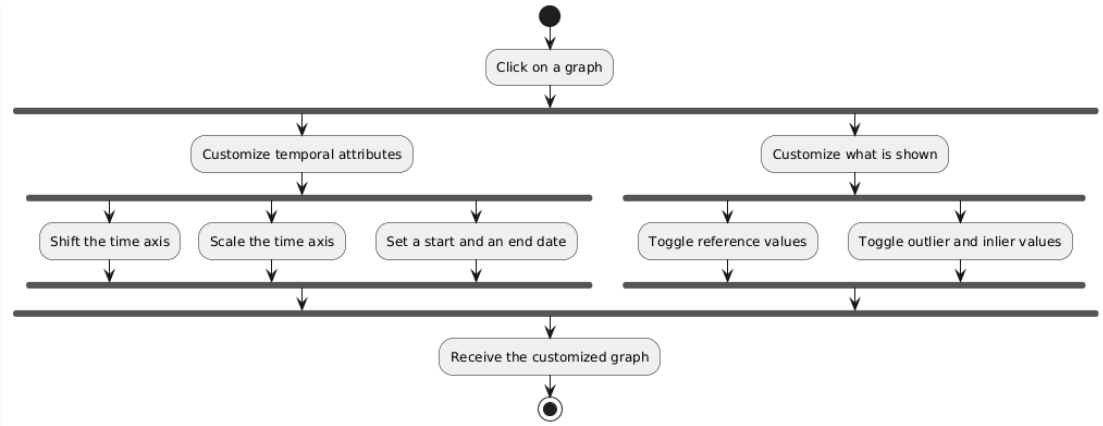

Fig. 29. Activity Diagram for Graph Customization

Graph merging and splitting

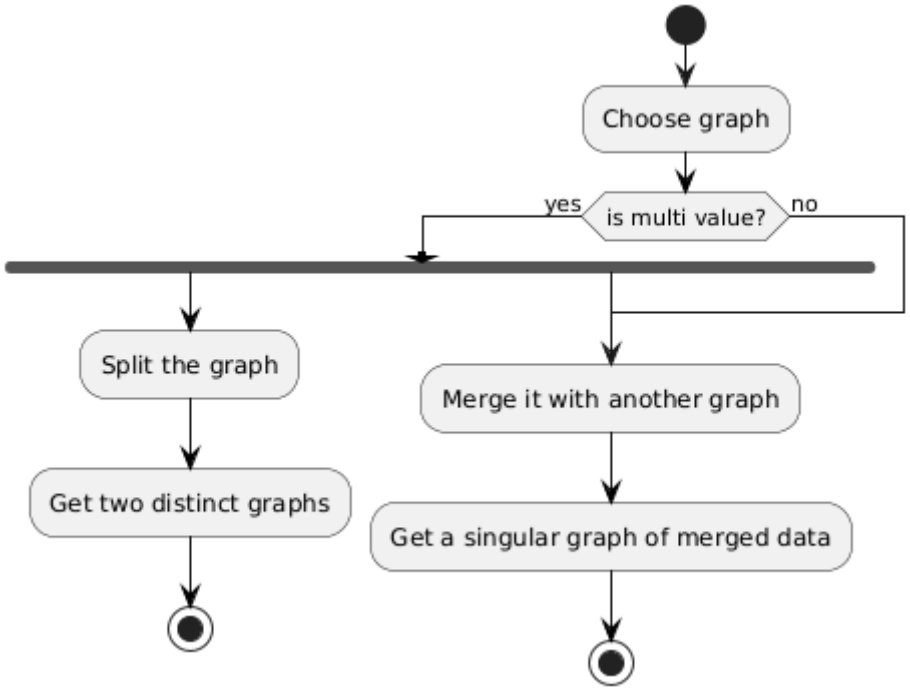

Fig. 30. Activity Diagram for Merging/Splitting

Export timeline to graphs page

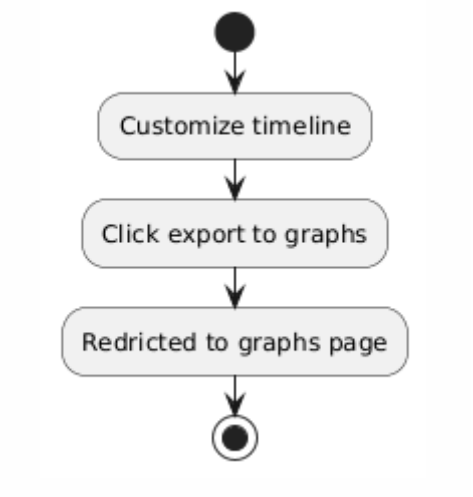

Fig. 31. Activity Diagram for Exporting Timelines to the Graph Page

Export graphs to timeline page

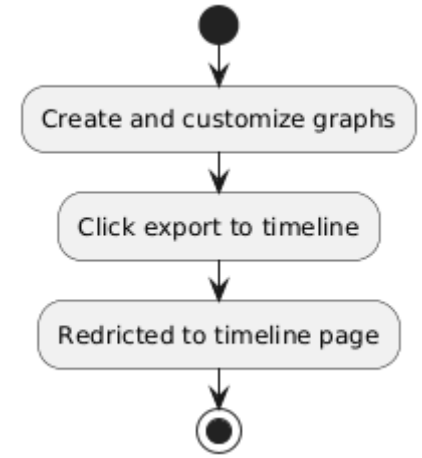

Fig. 32. Activity Diagram for Exporting Graphs to Timeline Page

Generic list page usage

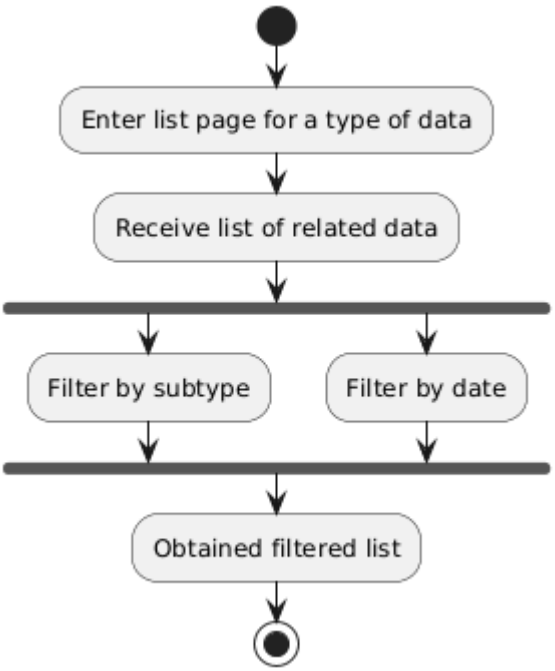

Fig. 33. Activity Diagram for List Page Usage

3.6.5 User Interface and Mockups

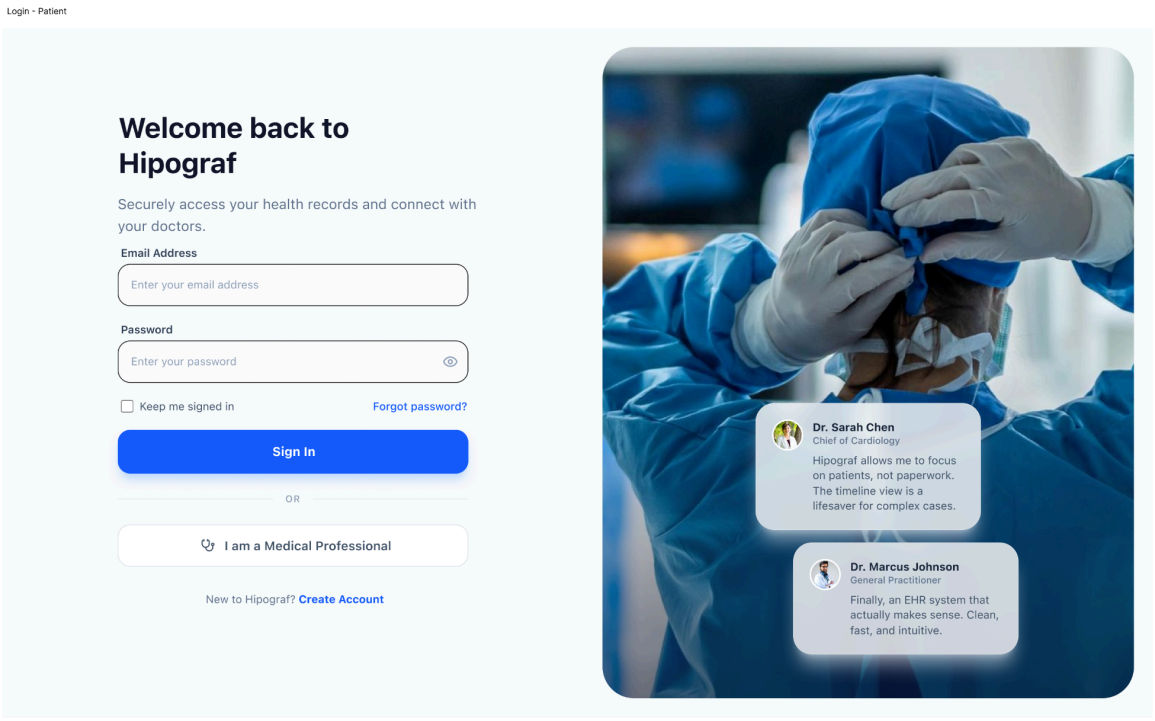

Fig. 34. User Interface for Login

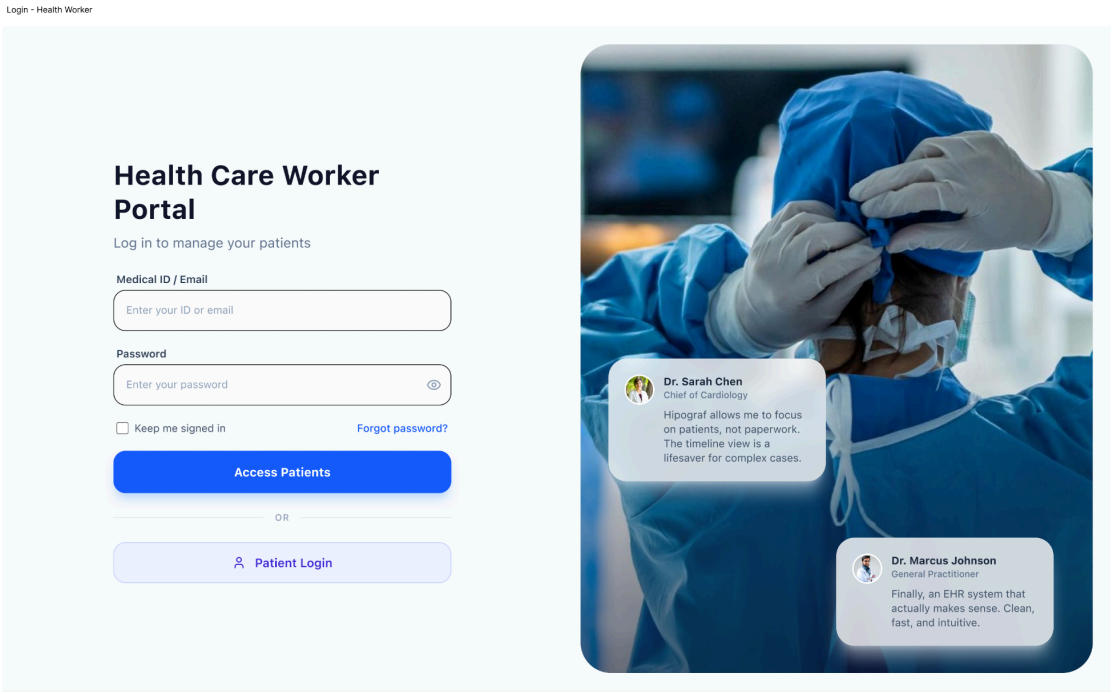

Fig. 35. Alternative User Interface for Login

Forgot Password

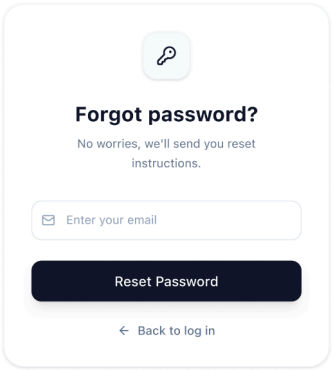

Fig. 36. User Interface for Password Reset

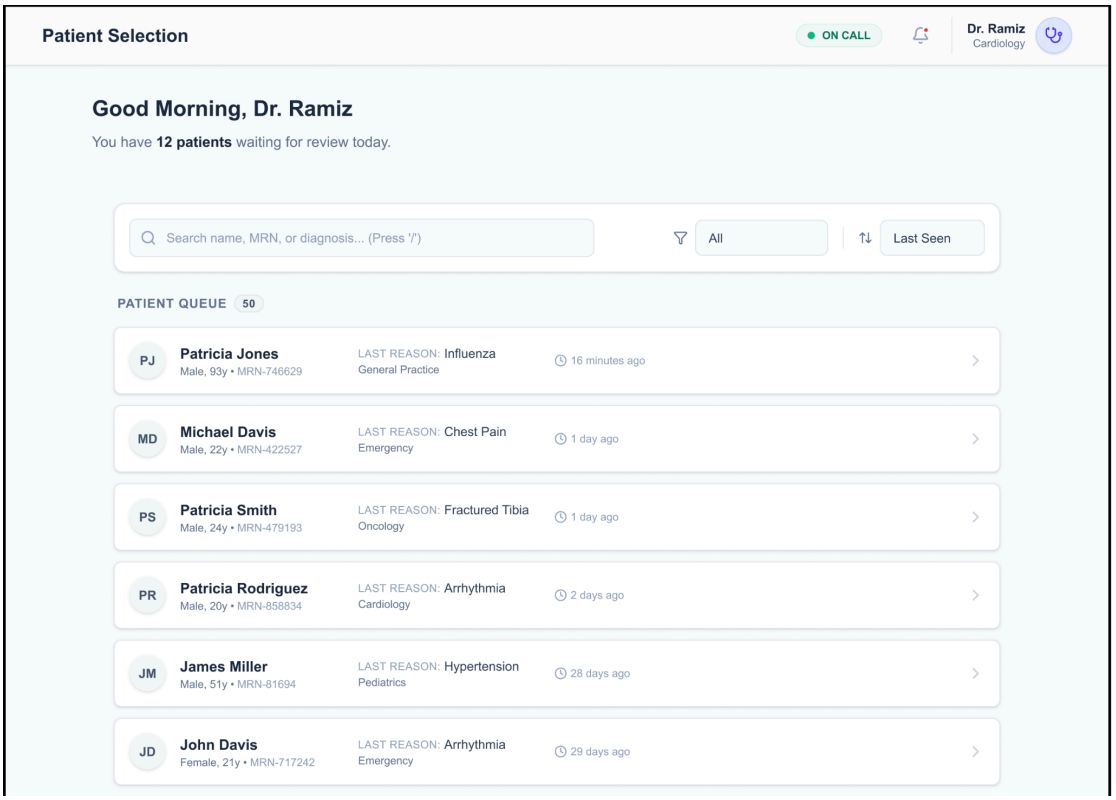

Fig. 37. User Interface for Patient Selection

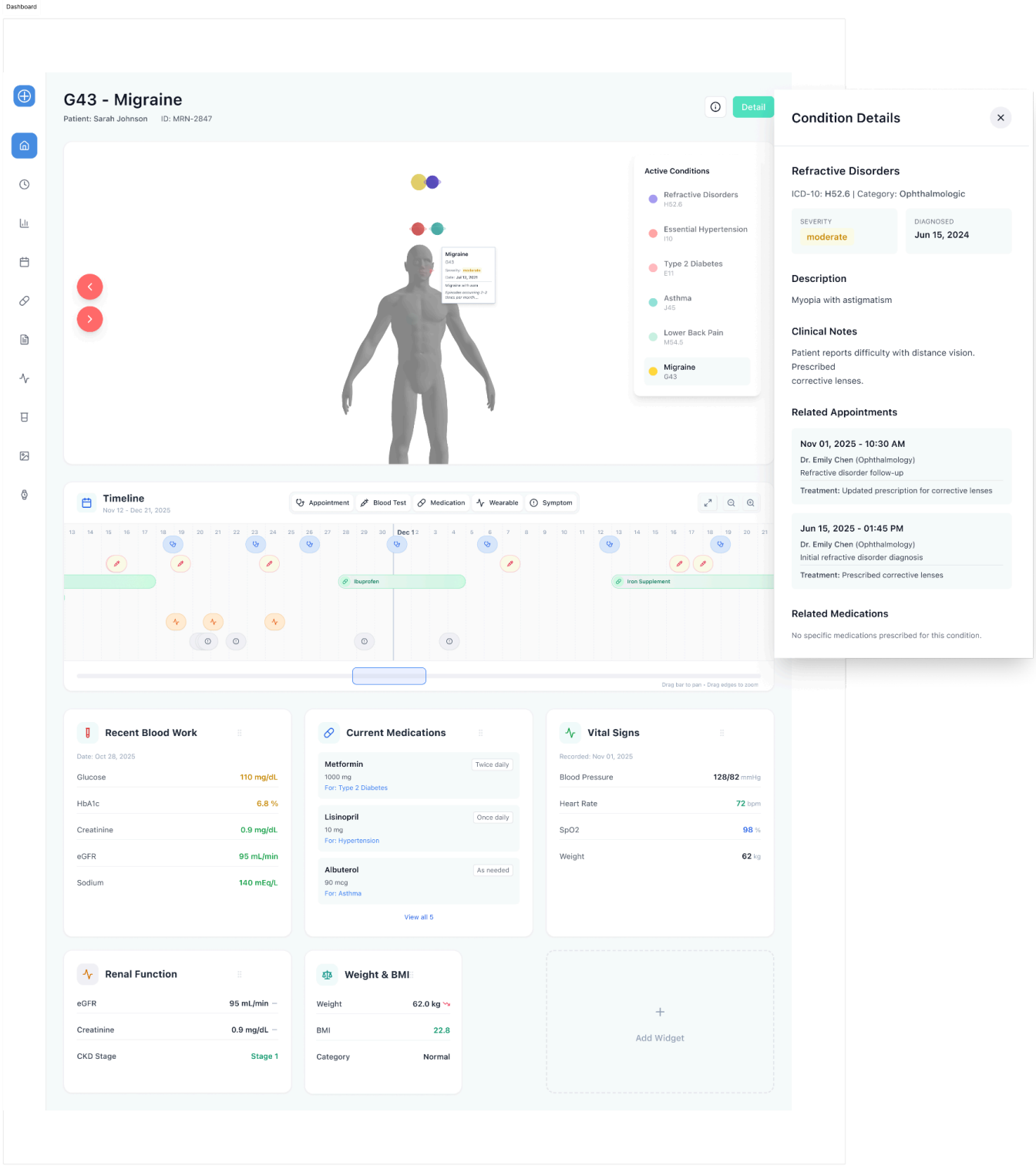

Fig. 38. User Interface for Dashboard Page

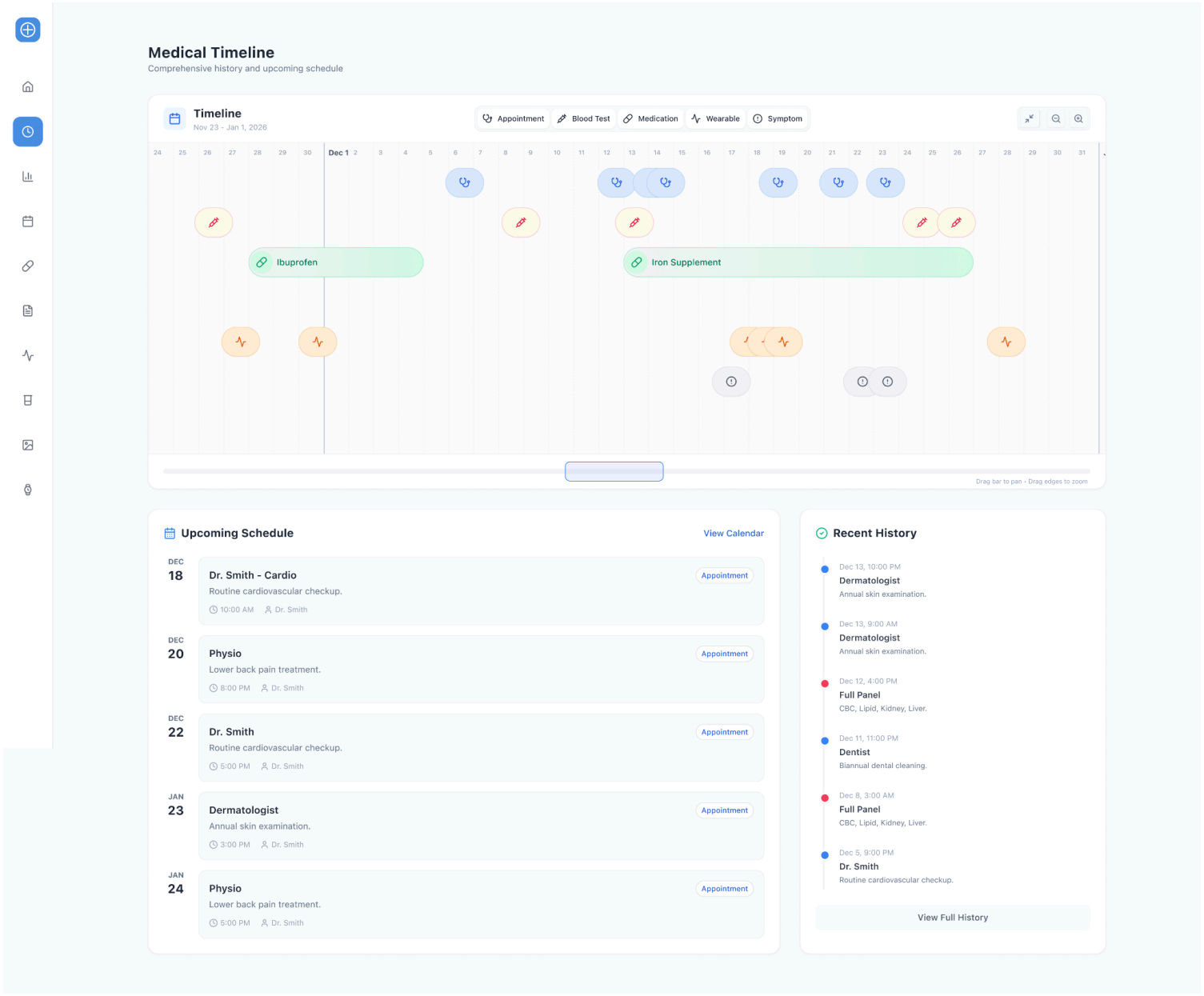

Fig. 39. User Interface for Timeline Page

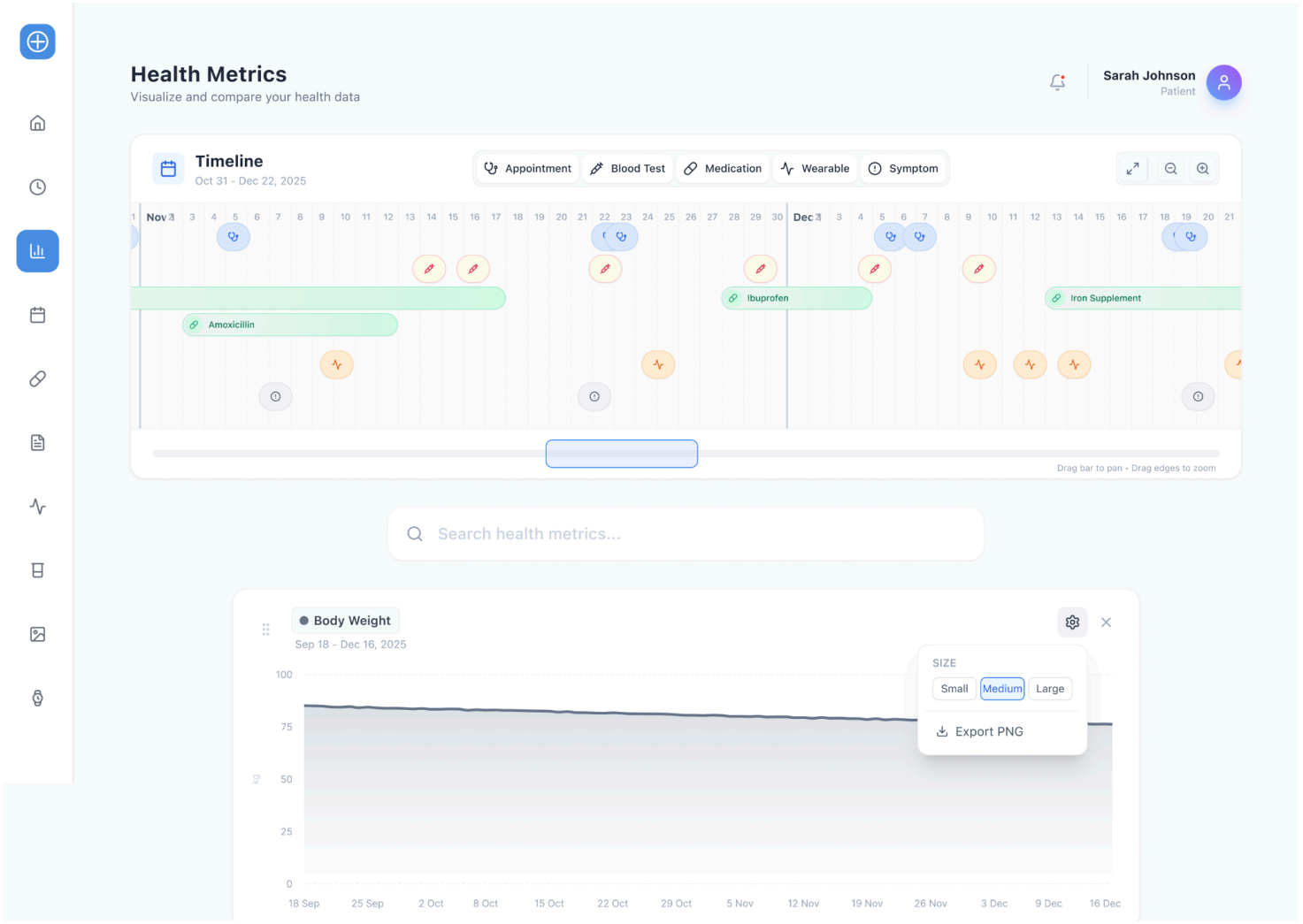

Fig. 40. User Interface for Timeline Page, Graph Creation

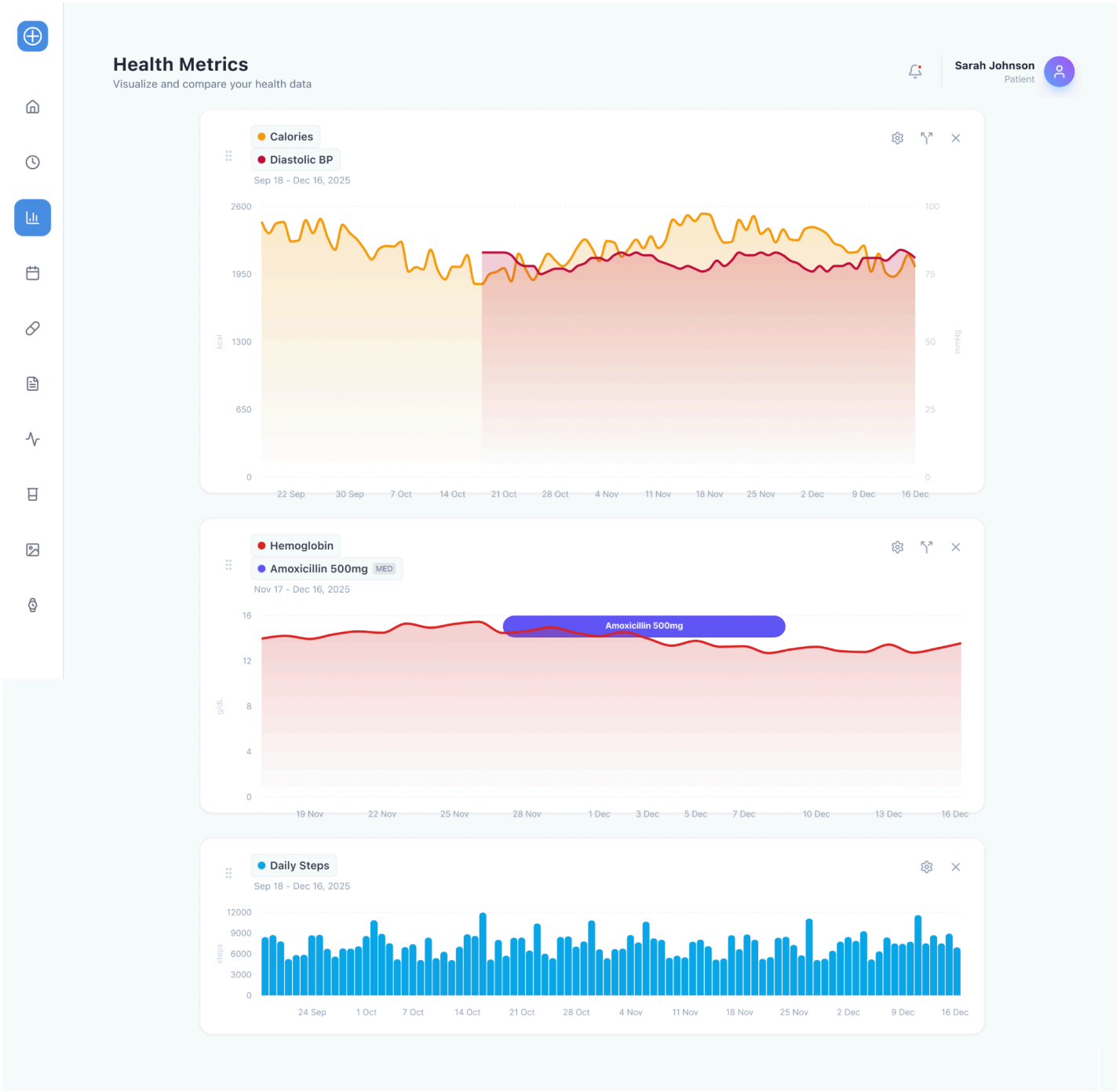

Fig. 41. User Interface for Graph Page

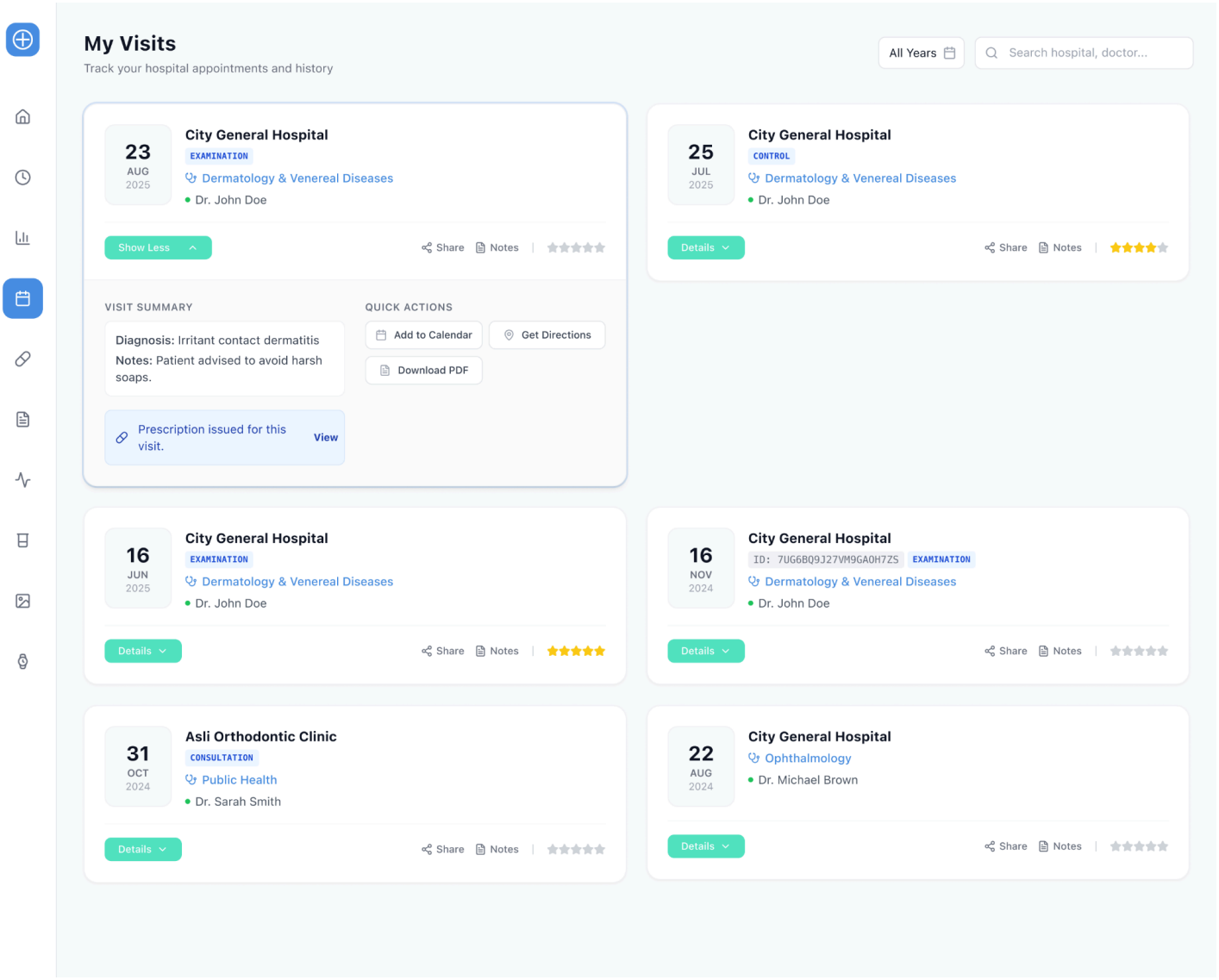

Fig. 41. User Interface for List Page, Hospital Visit Listing

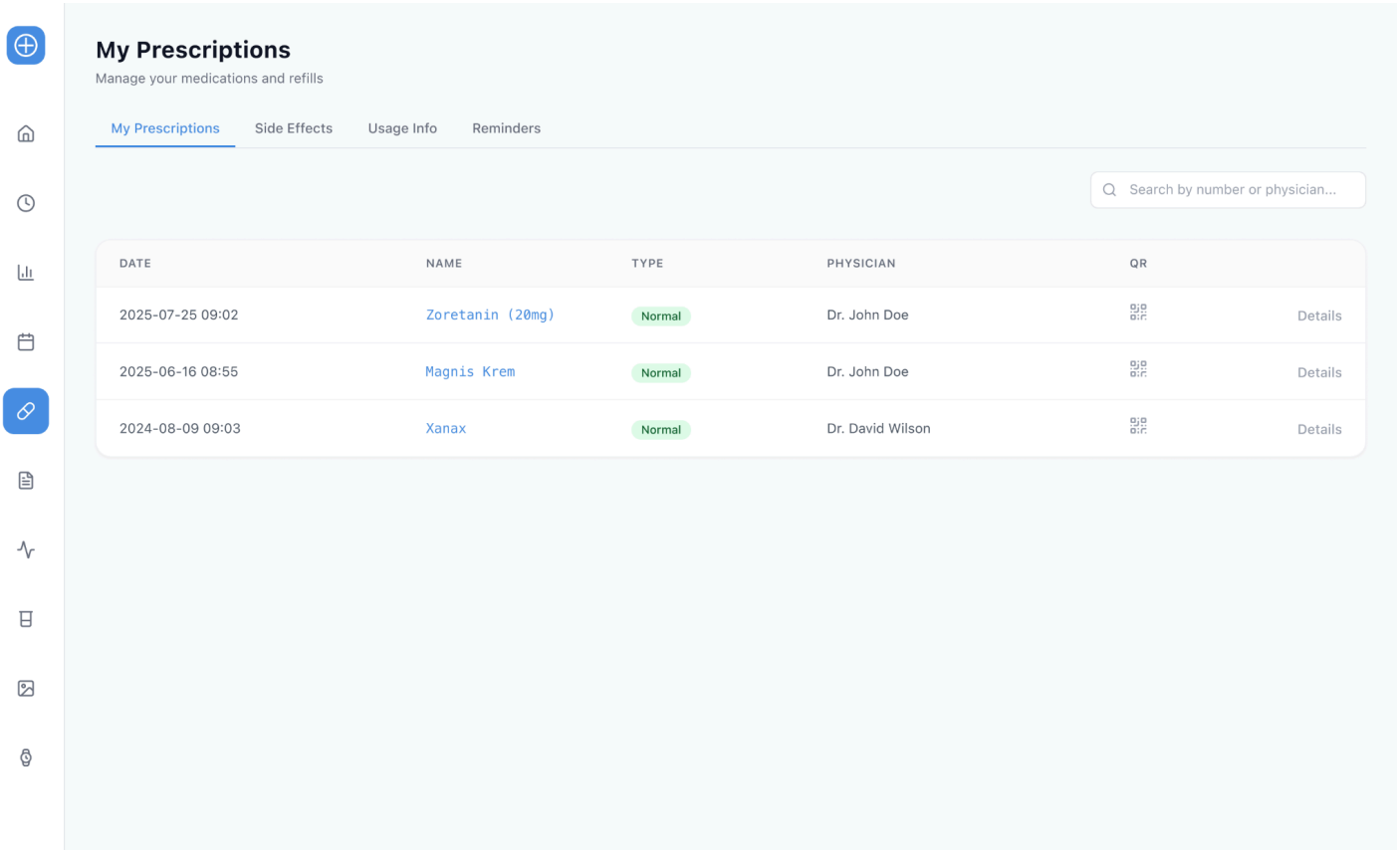

Fig. 42. User Interface for List Page, Prescription Listing

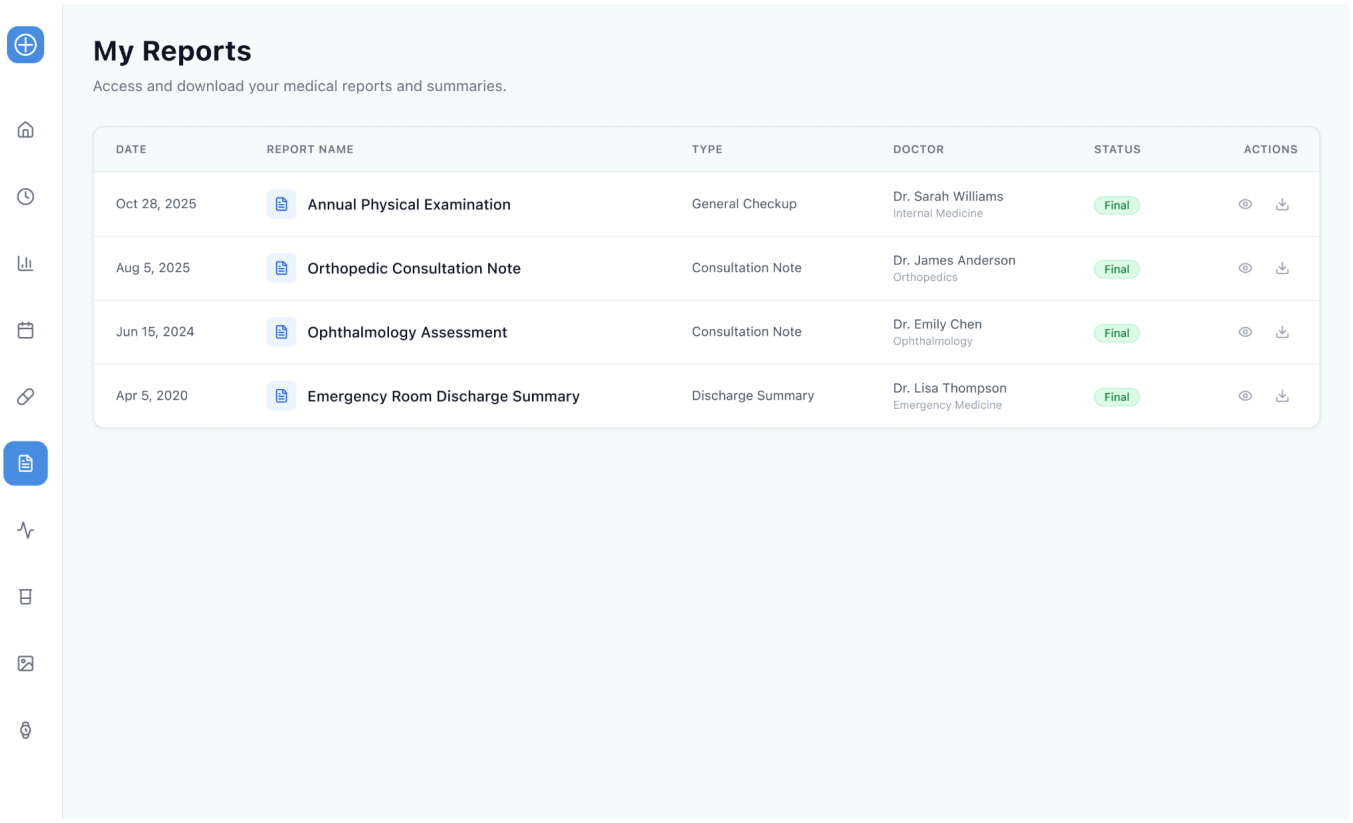

Fig. 43. User Interface for List Page, Report Listing

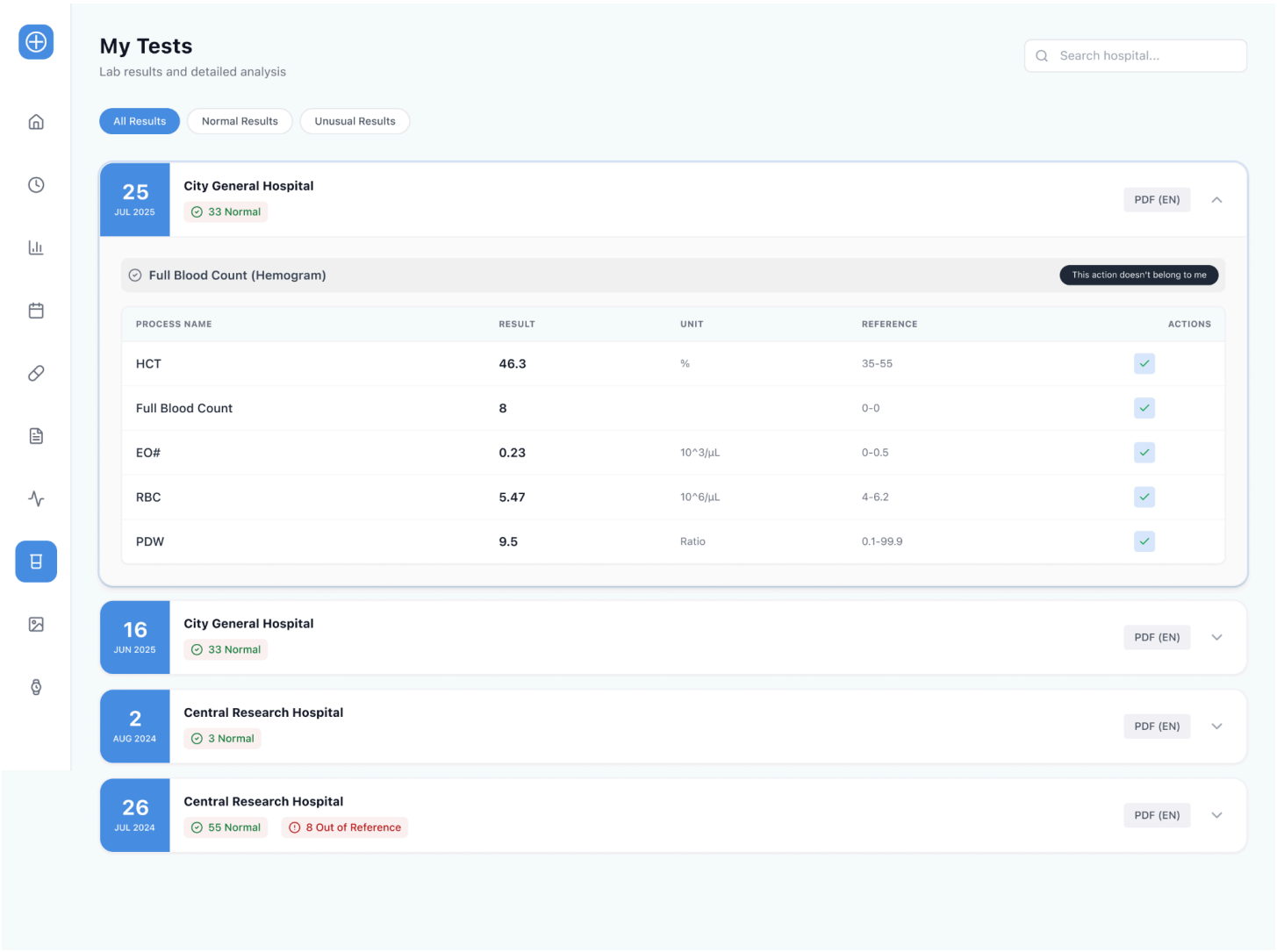

Fig. 44. User Interface for List Page, Blood Test Listing

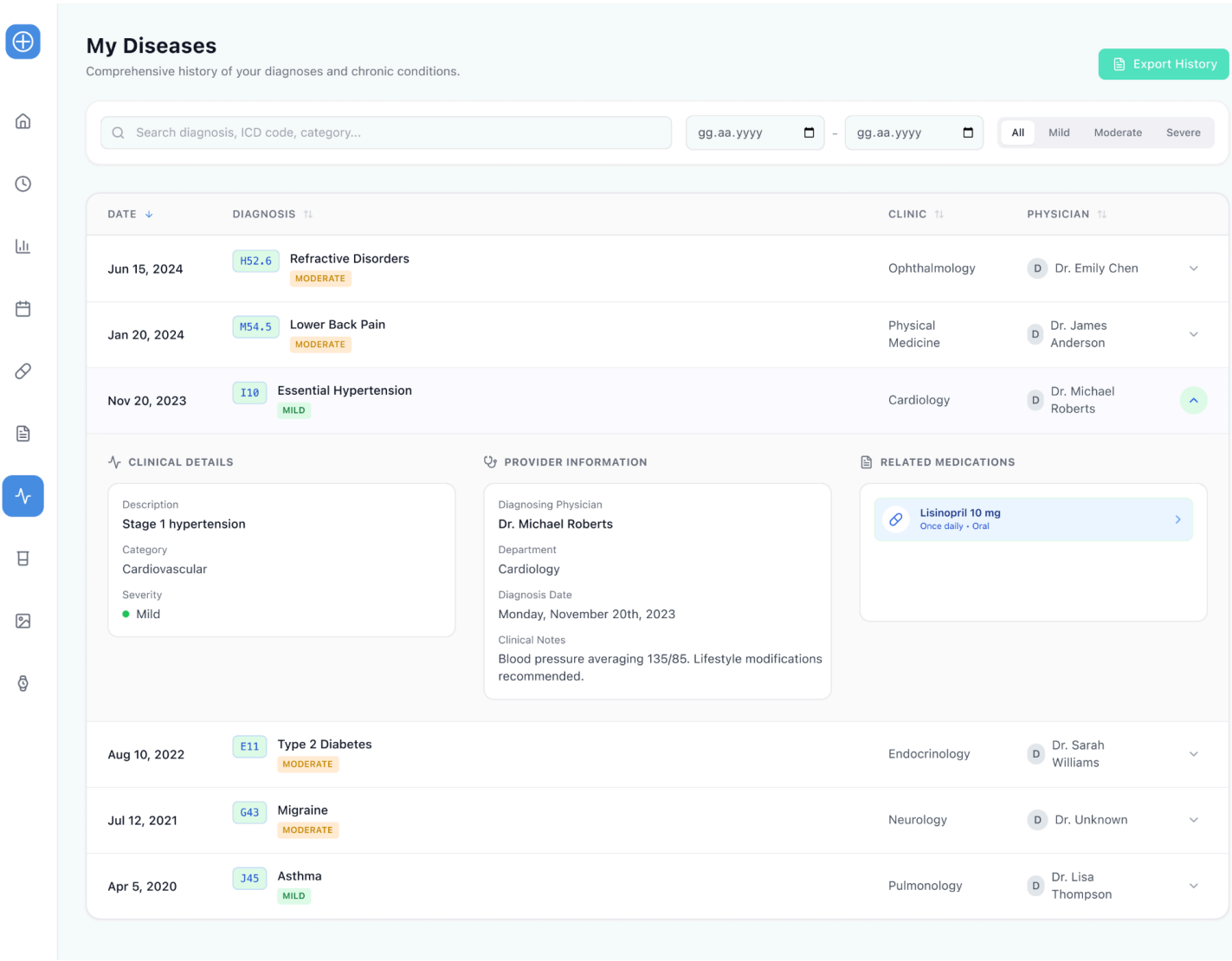

Fig. 45. User Interface for List Page, Diagnosis Listing

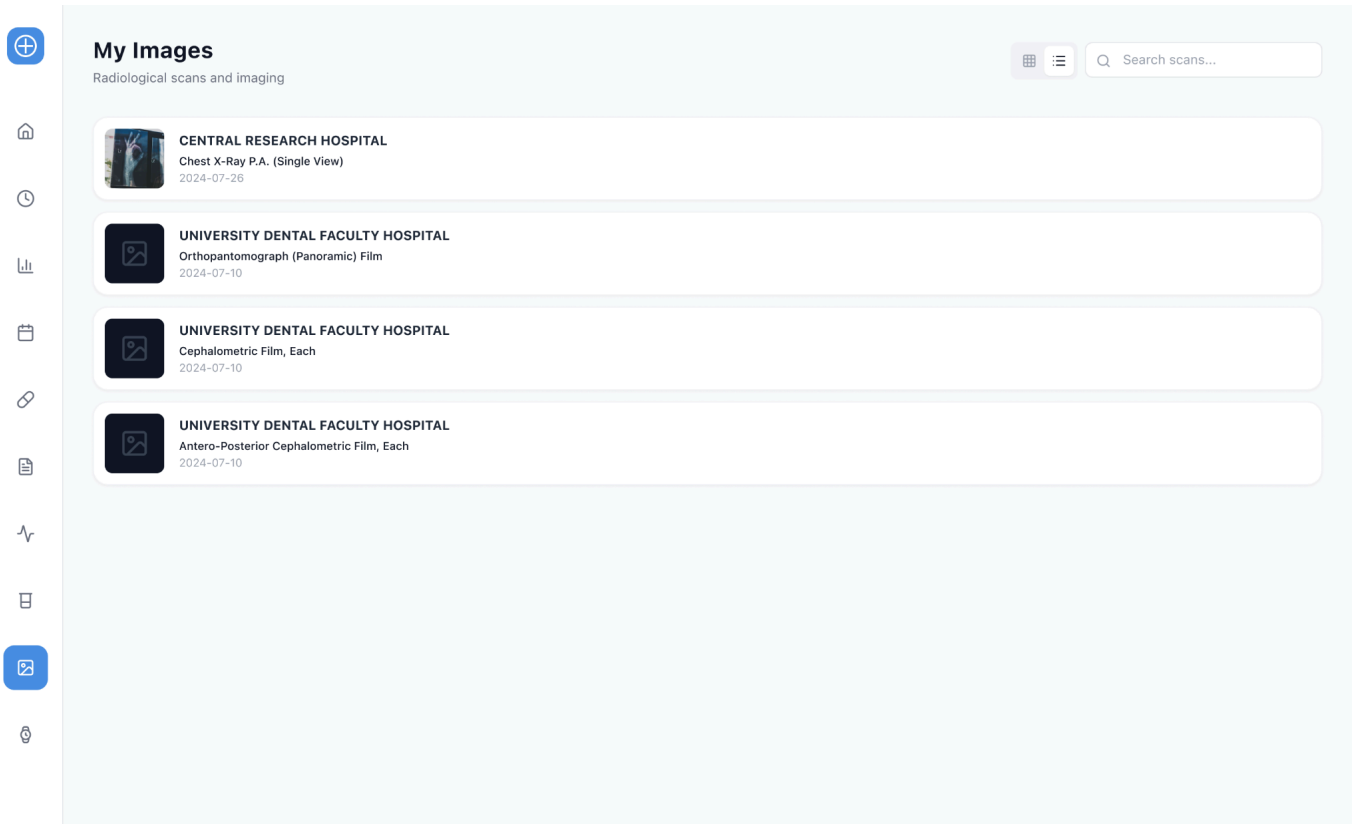

Fig. 46. User Interface for List Page, Radiological Image Listing

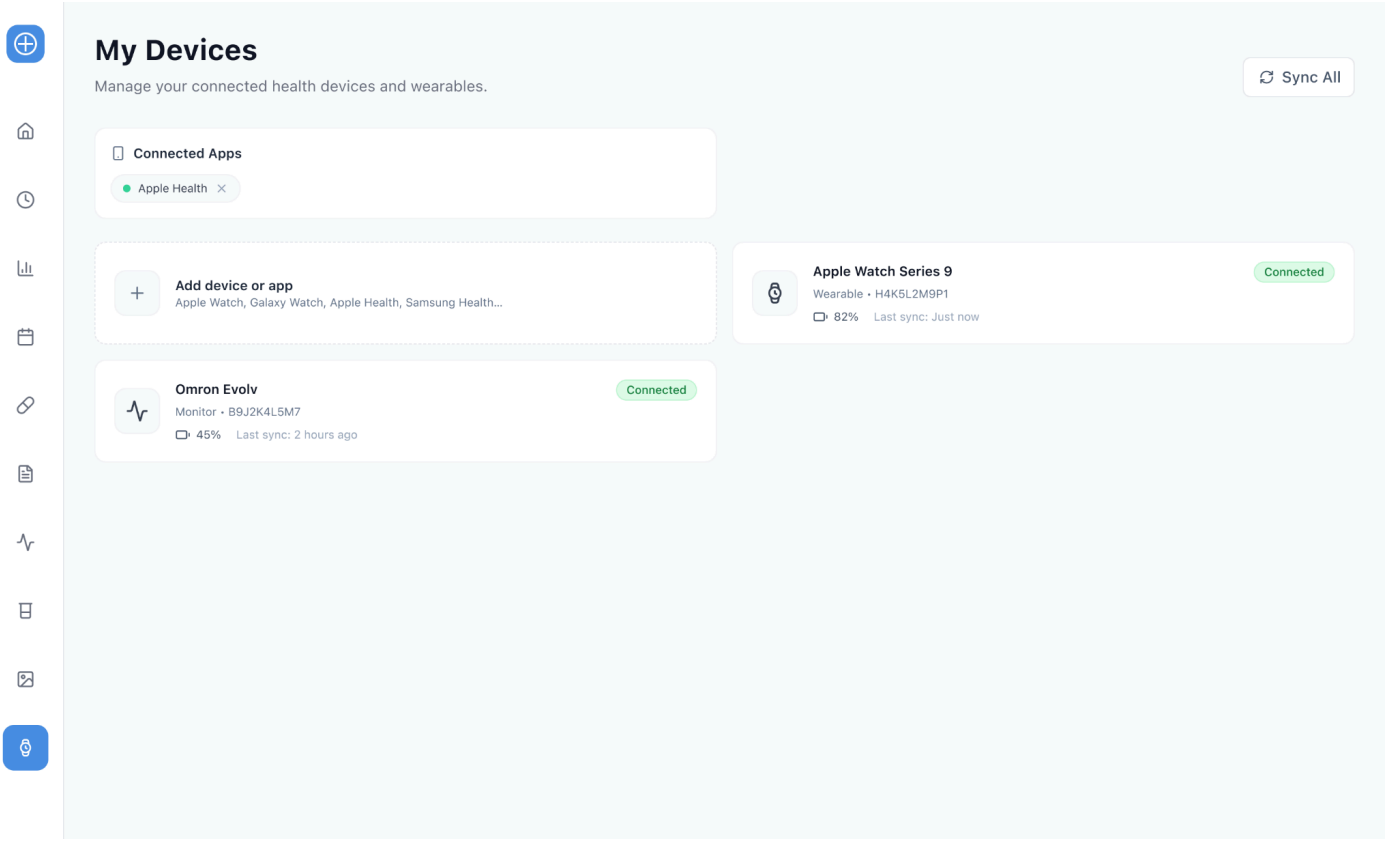

Fig. 47. User Interface for List Page, Wearable Listing

4 Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

4.1.1 Constraints

4.1.1.1 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.

4.1.1.2 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 afterward. 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.

4.1.1.3 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.

4.1.2 Standards

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.

4.1.2.1 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 [11, 12].

4.1.2.2 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 behavioral diagrams that are defined with UML including use case diagrams, activity diagrams, state diagrams and class diagrams [13].

4.1.2.3 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) [8].

4.1.2.4 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 [14].

4.1.2.5 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 [15].

4.1.2.6 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 [16].

4.1.2.7 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 [16].

4.2 Risks and Alternatives

Table I
Factors that can affect analysis and design

	Effect level	Effect
Public health	High	As it is concerned with the health of the public, any changes to the general health of the national public has a great probability of affecting Hipograf.
Public safety	None	Hipograf is not influenced by the situation of public safety.
Public welfare	Low	If people’s buying power was to be affected, it is possible that Hipograf would have to adapt to suit the new user base’s computer specifications.
Global factors	None	Hipograf is only concerned with the E-Nabız system in Türkiye.
Cultural factors	None	Hipograf is not influenced by cultural factors, as it is designed to be used by medical entities that are themselves tied to state operations.
Social factors	None	Hipograf is not subject to cultural factors.
Environmental factors	None	Hipograf has no issue with environmental factors being subject to change.
Economic factors	None	Hipograf will be a free-to-use and free-to-distribute software application.

Table II
Application Level Risk and Alternatives

Risk	Likelihood	Effect on the project	B Plan Summary
No Integration With E-Nabız	Medium	Patients can not automatically use their medical data stored in E-Nabız.	Release the software as a standalone application, allowing users to import their medical data semi-automatically.
Data Misrepresentation	Low	While a list is plain and unambiguous, visual representation gives a user much stronger intuition for pattern detection. There exists a risk that a mistake in how a graph plots creates confusion in the representation of the data.	Re-evaluate the way in which the existing data is represented through the visual components and modify these components accordingly.
Data Leak	Low	The preferences of the users can be leaked, leading to reduced trust in the user base.	Create an incident report detailing the events that led to the database leakage, and improve the security based on the results of the report.

4.3 Project Plan

Table III
Work Packages For Future Project Planning

WP#	Work package title	Leader	Members involved
WP1	Frontend Implementation	Ramiz Arda Ünal	Ramiz Arda Ünal Salih Furkan Göktaş Orhun Güder Şükrü Eren Gökırmak Artun Berke Gül
WP2	Backend Implementation	Şükrü Eren Gökırmak	Salih Furkan Göktaş Orhun Güder Artun Berke Gül Şükrü Eren Gökırmak
WP3	Testing	Orhun Güder	Orhun Güder Salih Furkan Göktaş Şükrü Eren Gökırmak
WP4	Non-Required Feature Implementation	Salih Furkan Göktaş	Salih Furkan Göktaş Orhun Güder Artun Berke Gül Ramiz Arda Ünal Şükrü Eren Gökırmak
WP5	E-Nabız Integration	Artun Berke Gül	Artun Berke Gül Ramiz Arda Ünal Orhun Güder

Table IV
Work Package Descriptions

WP 1: *Frontend Implementation*			
Start date: *15.01.2026* **End date:** *25.03.2026*			
Leader:	Ramiz Arda Ünal	**Members involved:**	*Salih Furkan Göktaş* *Orhun Güder* *Ramiz Arda Ünal* *Artun Berke Gül* *Şükrü Eren Gökırmak*
Objectives: Implement the existing and future UI mockups into Hipograf. Improve the existing frontend systems.			
Tasks: ***Task 1.1 Implement mockups into Hipograf*:** Convert the existing mockups designed with the team into functional frontend code. Ensure the Hipograf systems function as similar as possible to the mockups. ***Task 1.2 Improve existing frontend* :** Augment the implemented frontend pages. This can be done with simple visual improvements, such as button			

<table>
<tr><td colspan="4">hover effects and navigation transitions. The existing frontend can also be changed upon discovery of a better UI flow.</td></tr>
<tr><td colspan="4">Deliverables
D1.1: Implementation of My Prescriptions page
D1.2: Implementation of Wearables page</td></tr>
<tr><td colspan="4">WP 2: Backend Implementation</td></tr>
<tr><td colspan="4">Start date: 30.01.2026 End date: 31.03.2026</td></tr>
<tr><td>Leader:</td><td>Şükrü Eren Gökırmak</td><td>Members involved:</td><td>Salih Furkan Göktaş
Orhun Güder
Artun Berke Gül
Şükrü Eren Gökırmak</td></tr>
<tr><td colspan="4">Objectives: Implement the backend containing the endpoints used by the frontend.</td></tr>
<tr><td colspan="4">Tasks:
Task 2.1 Implement Database: Convert the object diagram into a database schema to be used in the MongoDB database.
Task 2.2 Implement Endpoints: Use the database schema created in Task 2.1 to create a Flask API for the purpose of connecting the database to the frontend.</td></tr>
<tr><td colspan="4">Deliverables
D1.1: MongoDB database design
D1.2: Implementation of Flask Application</td></tr>
<tr><td colspan="4">WP 3: Testing</td></tr>
<tr><td colspan="4">Start date: 31.03.2026 End date: 10.05.2026</td></tr>
<tr><td>Leader:</td><td>Orhun Güder</td><td>Members involved:</td><td>Orhun Güder
Salih Furkan Göktaş
Şükrü Eren Gökırmak</td></tr>
<tr><td colspan="4">Objectives: A testing and verification process to ensure that all facets of the application function at maximum capacity.</td></tr>
<tr><td colspan="4">Tasks:
Task 3.1 Reliability Testing: Will be conducted by surveying the server that Hipograf is hosted on for uptime and maintenance periods.
Task 3.2 Security Testing: Detailed consideration of Linux process-user permissions and audit of cryptographic primitives being used within the application domain.
Task 3.3 Performance Testing: Tools like Google PageSpeed Insights will be utilized to ensure the website’s performance on standard hardware meets the expected threshold.
Task 3.4 Usability Testing: The guidelines in the WCAG 2.0 standard will be followed in detail to ensure that web accessibility standards are correctly met.</td></tr>
<tr><td colspan="4">Deliverables
D3.1: Reliability Tests
D3.2: Security Tests</td></tr>
</table>

D3.3: *Performance Tests* ***D3.4:*** *Usability Tests*			
WP 4: *Non-Required Feature Implementation*			
Start date: *08.03.2026* **End date:** *10.05.2026*			
Leader:	Salih Furkan Göktaş	**Members involved:**	*Salih Furkan Göktaş* *Orhun Güder* *Artun Berke Gül* *Ramiz Arda Ünal* *Şükrü Eren Gökırmak*
Objectives: *After critical system components are fully in place, time may be allotted for the implementation of additional features.*			
Tasks: **Task 4.1 Implementation of the Sandbox Module :** Sandbox would be an experimental and low-priority additional module for Hipograf that would function as an extension of the Graphs Page, where graphs, timelines and listings could all be dynamically created and linked on a page that allows one to explore with utmost freedom. **Task 4.2 OCR Blood-Test Support:** Real, printed copies of a patient's blood tests could be picked up using OCR text recognition by an additional mobile applet, sending the data to be temporarily (until the end of the session) used in Hipograf's visualization suite. **Task 4.3 Mobile Support:** An additional extra goal would be to ensure that the web-based Hipograf application works properly on smaller screen sizes that come standard on mobile phones and tablets, alongside their touch screen environments.			
Deliverables ***D4.1:*** Sandbox Module ***D4.2:*** OCR Applet ***D4.3:*** OCR Data Integration ***D4.4:*** Support for Smaller Screen Resolutions ***D4.5:*** Support for Touchscreen I/O			
WP 5: *E-Nabız Integration*			
Start date: *25.02.2026* **End date:** *31.06.2026*			
Leader:	*Artun Berke Gül*	**Members involved:**	*Artun Berke Gül* *Ramiz Arda Ünal* *Orhun Güder*
Objectives: Get approval from the government to implement Hipograf into the E-Nabız ecosystem.			
Tasks: ***Task 5.1 Apply for E-Nabız integration*:** *Fill the required documents for E-Nabız integration and apply to Turkish Health Ministry Documentation Unit*			

Task 5.2 Passive account creation in government portal**:** *A passive account in kts.saglik.gov.tr portal is created and the necessary documents are uploaded.* ***Task 5.3 Testing*****:** *Pass standardized government mandated tests on various fronts to acquire a software access code.* ***Task 5.4 Integration*****:** *Integrate Hipograf into E-Nabız using the software access code granted by passing the government mandated testing.*
Deliverables ***D5.1:*** *Turkish Health Ministry Documentation Unit documents.* ***D5.2:*** *Documents uploaded to kts.saglik.gov.tr portal.* ***D5.3:*** *Acquisition of software access code.* ***D5.4:*** *Fully integrated Hipograf.*

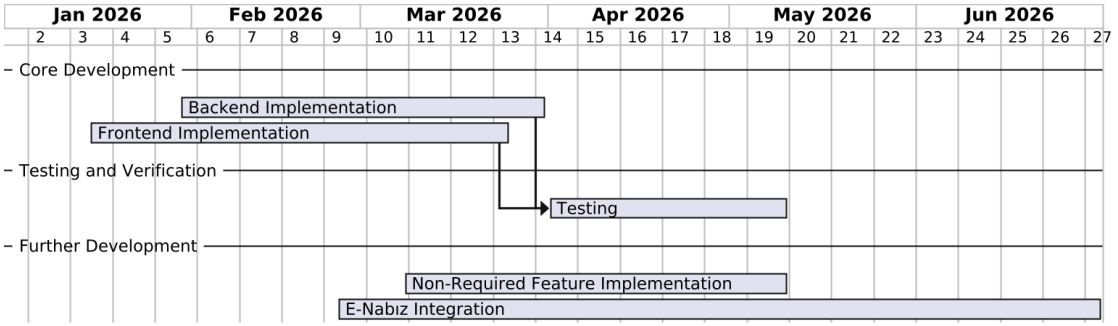

Fig. 48. Project Plan Gantt Chart

4.4 Ensuring Proper Teamwork

The most important factor boosting our team's quality of teamwork is the fact that our team has good communication. Our team has created multiple groups in voice and text communication tools that we use daily for the purpose of senior group project communications, lowering the bar of entry into the project discussions and ensuring everyone is always informed about the most recent addition and changes to the project. On top of this, we have created structures to follow for team wide decision making, writing, and technical participation to further facilitate teamwork.

4.4.1 Team Wide Decision Making

To make any decisions related to the project, each and every team member comes up with their own suggestions. Afterward, all team members convene to present their respective suggestions and vote on one of their choosing. Thanks to this process, all the team members are involved in the decision making process and every team member is always up to date regarding the project.

4.4.2 Writing

When writing reports and documents related to Hipograf, they are segmented into small sections. Then, these sections are distributed equally between the team members. However, to maximize member motivation and ensure teamwork, team members are given the freedom to pick the sections to their liking.

4.4.3 Technical Participation

To track what is done by each team member on the technical front, the issue system in Github is used since Git is already being used as our team's version control system and Github is used to share the project. Each issue represents a part of the project to be implemented or an implementation that needs to be fixed. These issues are distributed equally among the team members to ensure equal contributions from all members. On top of this, using this strategy, we are able to obtain a log of everybody's past contributions and a plan for the future.

4.4 Ethics and Professional Responsibilities

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 [14].

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.

4.5 Planning for New Knowledge and Learning Strategies

As a project spanning over multiple programming languages and frameworks, it is expected that we will have to acquire and use new knowledge. To gain this knowledge, we are planning to use various sources. The first source we plan to use is the documentations of the languages and libraries we use. Next; online courses, video tutorials, and otherwise informative content found online is marked as potentially useful for learning new knowledge. For commonly found problems, we also see websites such as Stack Overflow as valuable resources [17].

5 References

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

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

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

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

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

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

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

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

[9] “The MIT License” https://opensource.org/license/mit. [Accessed: Dec 17, 2025]

[10] “Apache License, Version 2.0”. https://www.apache.org/licenses/LICENSE-2.0. [Accessed: Dec 17, 2025]

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

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

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

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

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

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

[17] “Stack Overflow”. https://stackoverflow.com/. [Accessed: Dec 18, 2025]