

Bilkent University

CS-491

Senior Design Project



Fall 2023/2024

Analysis&Requirements Report

Group "PrioVar" – T2309

Supervisor: Ercüment Çiçek

Innovation Expert: Can Alkan

Jury Members: Atakan Erdem & Mert Bıçakçı

<u>Member Name</u>	<u>Student ID</u>
Korhan Kemal Kaya	21903357
Halil Alperen Gözetten	21902464
Kaan Tek	21901946
Erkin Aydın	22002956
Safa Eren Kuday	21902416

1. Introduction	2
2. Current System	2
3. Proposed System	3
3.1 Overview	3
3.2 Functional Requirements	5
3.3 Non-Functional Requirements	5
3.3.1 Usability	5
3.3.2 Reliability	6
3.3.3 Performance	6
3.3.4 Supportability	6
3.3.5 Scalability	7
3.3.6 Portability	7
3.3.7 Security	7
3.4 Pseudo Requirements	7
3.5 System Models	8
3.5.1 Scenarios	8
3.5.2 Use Case Model	14
3.5.3 Object and Class Model	15
3.5.4 Dynamic Models	15
3.5.4.1 Activity Diagrams	15
3.5.4.1 State Diagrams	16
3.5.4.1 Sequence Diagrams	17
3.5.5 High-Level System Architecture & Components of Proposed Solution	19
3.5.5 User Interface - Navigational Paths and Screen Mock-ups	20
4. Other Analysis Elements	27
4.1. Consideration of Various Factors in Engineering Design	27
4.1.1. Public Health Considerations	27
4.1.2. Public Safety Considerations	27
4.1.3. Public Welfare Considerations	27
4.1.4. Global Considerations	28
4.1.5. Cultural Considerations	28
4.1.6. Social Considerations	28
4.1.7. Environmental Considerations	28
4.1.8. Economic Considerations	29
4.1.9 Table of the Aforementioned Considerations	29
4.1.10 Constraints	30
4.1.10.1. Economic Constraints	30
4.1.10.2. Privacy and Security Constraints	30
4.1.10.3. User Experience and Usability Constraints	30
4.1.11. Engineering Standards	30
4.1.11.1. IEEE-International Standard Systems and Software Engineering Software Life Cycle Processes	31
4.1.11.2. Unified Modelling Language (UML):	31
4.1.11.3. IEEE-Recommended Practice for Software Requirements Specifications	

4.2. Risks and Alternatives	31
Table 2: Risks and Alternatives	32
4.3. Project Plan	33
4.4. Ensuring Proper Teamwork	39
4.5. Ethics and Professional Responsibilities	39
4.6. Planning for New Knowledge and Learning Strategies	40
5. Glossary	40
6. References	41

1. Introduction

Many of the known diseases are genetically caused, and it is highly important to comprehend their underlying reasons by examining genomic data, as this knowledge can lead to better prevention and early detection of the diseases, which will significantly impact public health. The interpretation of genetic variants, which are obtained by DNA sequencing, is crucial for diagnosing these genetic diseases, as it enables healthcare providers to distinguish disease-causing mutations from benign variations. One way to achieve this is through variant prioritization, which is the process of ranking and selecting genetic variations for further investigation based on their potential relevance to being pathogenic. However, variant prioritization techniques merely involving genomic data may be unsatisfactory, as the combination of genetic and phenotypic information allows for a more comprehensive understanding of an individual's health, enabling more precise diagnoses and treatment strategies. After prioritizing the variants, clinicians should further investigate the connection of the selected variants with known diseases to be able to make a diagnosis by looking at the variant frequency, gene location, etc. In this case, the accuracy of the final diagnosis is highly dependent on the clinicians' knowledge, yet it can be enhanced through providing disease-related information to the clinicians. Thus, this interaction in the final phase might be a critical factor affecting the accuracy of the overall treatment process of patients.

Hence, establishing a common platform for health clinics to exchange and access variant and disease information under the anonymity of patient data will facilitate the whole process. This will allow clinicians to tap into a collective pool of knowledge, improving the process from prioritizing genetic variants to connecting them with diseases. Clinicians will make diagnoses more effectively and accurately by benefiting from the aggregated data, sharing expertise, and leveraging known cases, which will ultimately increase the quality of patient care.

2. Current System

In the market, there are companies that either focus on or use variant prioritization, such as Genomize (Turkish origin), Engenome, and Centogene [1], [2],

[3]. Nearly all of the companies in the current market have a claim about their successes on a specific metric, such as top-k scores or real clinical performances. Also, some of the companies like Centogene are more oriented towards building a large dataset of mutations with the patient data like CentoMD® data repository. As they claim that it is the world's largest curated mutation database with more than 7.3 million variants and a significant number of unpublished variants [3]. On the other hand, some companies like Engenome are focused on utilizing the inheritance patterns by analyzing the family line.

Our app PrioVar will be grounded in similar techniques by incorporating both observed phenotype and genotype information of the patients with the above companies. However, we also plan to provide our customers the ability to search into their own populations and discover insights from them by both querying and creating statistics. Moreover, our other goal is building a connected app for the health centers to exchange valuable variant and diagnosis information. With this, we expect an important decrease in the average diagnosis times and waste of resources as the analysis of repeated cases will be avoided.

3. Proposed System

3.1 Overview

Through our application PrioVar, we will facilitate the process of rare disease diagnosis, and the health clinics will become interconnected for a collective impact. Our project is planned to be a tool for health clinics by integrating phenotype data into the decision-making process. This innovative approach of integrating phenotype information is a valuable technique that allows us to make the most of the available patient data. Clinics can prioritize genetic variants in order to make more accurate predictions about the causes of rare diseases.

The clinician personnel of a health center will upload the genetic variants of the patients in VCF (Variant-Call Format) file format [4]. Each row of this file contains related attributes of a variant, such as chromosome location, reference allele, etc. An example of this file is shown below:

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	NA000001	NA000002	NA000003
20	14370	rs6054257	G	A	29	PASS	NS=3;DP=14;AF=0.5;DB;H2	GT:GQ:DP:HQ	0 0:48:1:51,51	1 0:48:8:51,51	1/1:43:5:.,.
20	17330	.	T	A	3	q10	NS=3;DP=11;AF=0.017	GT:GQ:DP:HQ	0 0:49:3:58,50	0 1:3:5:65,3	0/0:41:3
20	1110696	rs6040355	A	G,T	67	PASS	NS=2;DP=10;AF=0.333,0.667;AA=T;DB	GT:GQ:DP:HQ	1 2:21:6:23,27	2 1:2:0:18,2	2/2:35:4
20	1230237	.	T	.	47	PASS	NS=3;DP=13;AA=T	GT:GQ:DP:HQ	0 0:54:7:56,60	0 0:48:4:51,51	0/0:61:2
20	1234567	microsat1	GTCT	G,GTACT	50	PASS	NS=3;DP=9;AA=G	GT:GQ:DP	0/1:35:4	0/2:17:2	1/1:40:3

Figure 1: Example VCF File

Together with this, the clinicians also need to select the associated phenotypic features of the patient by searching through our app. These selected phenotypic features will be the HPO (Human Phenotype Ontology) terms [5]. Around 15000 HPO terms exist, which are arranged in a directed acyclic graph and are connected by edges such that a term represents a more specific of its parent [6]. Some examples of HPO terms are given below:

```
[Term]
id: HP:0000002
name: Abnormality of body height
is_a: HP:0001507 ! Growth abnormality

[Term]
id: HP:0000010
name: Recurrent urinary tract infections
is_a: HP:0002719 ! Recurrent infections
is_a: HP:0011277 ! Abnormality of the urinary system
physiology
```

Figure 2: Example HPO Terms

We will combine phenotype and genotype information provided by the clinicians and feed them into the inference models. The results for the pathogenicity scores of the variants, together with other annotations, such as related diseases and known population frequencies, will be provided to the clinicians on a separate page. In some of these known annotations, we will utilize third-party APIs to facilitate the implementation process of the application. We will also provide an insights page for a selected variant where we will display the known annotations in a more detailed manner. This will support the final diagnosis phase, where the clinician selects a disease that is related to the case.

Besides variant prioritization, health clinics will be able to exchange valuable variant frequency information anonymously via our interface and gain access to anonymous data from individuals with similar phenotypes/genotypes with known diseases. This will be done in two different ways. Firstly, we plan to provide health clinics with deep-dive metrics and statistics of specific variant frequencies, providing them with insights into their patient populations. In a broader scope, we also plan to maintain a central analysis of variant frequencies, facilitating a countrywide perspective on rare genetic conditions. Additionally, we aim to provide visual statistics of variant and phenotype distributions to enhance the overall understanding of the population patterns.

Furthermore, we plan to provide support in unresolved patient cases by allowing medical centers to search for similar patients in the populations of other centers. The search will be done either by phenotypic or genotypic similarity. However, in order to use the data in case of a match, we will require the consent of this second medical center. Besides that, we will also allow the medical centers to query their own population data by selecting from different filters to extract relevant information. This feature provides clinicians with a customizable approach to analyzing patient data within their own centers, contributing to more informed decision-making in clinical scenarios.

3.2 Functional Requirements

- Clinicians should be able to upload patient genetic data in VCF file format using PrioVar's data upload feature.
- PrioVar should provide a search interface for clinicians to select the patient's phenotypic features using HPO terms.
- The system should be able to match patients with similar genetic/phenotypic traits efficiently when clinicians ask and send notifications to matching institutions about their match, the matching institution can choose to exchange information for both patients.
- The system should integrate phenotype and genotype data from the clinicians and feed them into inference models.
- Inference models should be able to identify causative genetic variants with high accuracy.
- PrioVar should present the results of the pathogenicity scores of genetic variants on a results page accessible to clinicians, and it retrieves and demonstrates related theses for each variant.
- Clinicians should be presented with statistics and information on the associated disease functions of the prioritized variants to support the diagnosis phase.
- Health clinics should be able to exchange variant frequency information anonymously.
- Health Centers can retrieve the list of patients with specific traits, such as a list of phenotypes, genetic variants, sex, and age interval.
- The app should provide sophisticated metrics of specific variant frequencies to health clinics to provide information about their patient populations, and the system should analyze variant frequencies countrywide.
- Clinicians should be able to add new patients and edit/delete existing patient information.
- The system should only keep essential patient data following anonymity rules and privacy rules.
- Health clinics should be able to authorize access to their clinicians.

3.3 Non-Functional Requirements

3.3.1 Usability

The user interface will focus on creating a user-friendly experience characterized by minimalistic design principles. The interface should be designed to ensure that users can access key features with the least number of clicks possible; it should not be more than 4 clicks from the main dashboard. Additionally, the clarity and size of the text should ensure that all

writings and statistics are easily readable and visually appealing, and in order to do that, all writings should be at least size 12. It also should have clear and concise labels and explanations on buttons and other interactive elements, guiding clinicians smoothly. The application should be responsive and interact with the user when something is done successfully, or something goes wrong.

3.3.2 Reliability

The PrioVar system will utilize Spring Boot and Neo4j to ensure reliable data management, with Spring Data Neo4j streamlining data access and transactions. TorchServe will enable dependable interactions with machine learning models via REST APIs, ensuring prompt and consistent system responses. The infrastructure is designed to prevent data loss and maintain data consistency across user sessions, with robust error handling and recovery protocols in place to support a stable and secure user experience. Data anonymity is preserved throughout to maintain user privacy.

3.3.3 Performance

For the PrioVar system, performance is key in providing timely health care support. The application will be optimized to ensure response times of under 2 seconds on a stable 10Mbps connection for user actions. Database queries, powered by the Neo4j graph database, are designed to be completed in under a second for efficiency. In cases where any loading process exceeds this duration, a loading icon will be displayed to inform the user. The machine learning model inferences, deployed through TorchServe are expected to be quick, contributing to the overall speed of the system. This ensures health care providers can upload and interact with data promptly, maintaining the flow of their work without unnecessary delays.

3.3.4 Supportability

The PrioVar system will be designed employing an object-oriented approach to adapt and improve continuously through user feedback on the web interface, ensuring a consistently intuitive user experience. As case numbers grow, the platform will automatically augment its database, ensuring that the data remains extensive and robust. It is vital to ensure the keep database structure is the same during the application life-cycle for supportability in further versions. For this purpose, Neo4j documentation and conventions will be utilized. Concurrently, the accuracy and efficiency of the predictive models will be refined through the incorporation of state-of-the-art techniques in bioinformatics and computational genomics.

3.3.5 Scalability

The PrioVar system should scale to support simultaneous use by more than 100 hospitals and 2,000 individual users. The PrioVar system should be designed to handle a growing number of users and an increasing amount of data efficiently. It can be expected to accommodate at least a 30% increase in users and a 40% increase in data annually. This growth must be managed without sacrificing the speed or functionality that users rely on, ensuring the system remains fast and reliable as it expands. The application can be containerized to facilitate easy deployment across multiple servers, ensuring load distribution and service availability even as demand fluctuates.

3.3.6 Portability

The PrioVar application needs to be multiplatform, supporting devices with Android, iOS, Windows, Linux, and MacOS operating systems with pre-installed web browsers.

3.3.7 Security

First-time users should be able to sign up with a confirmation mail sent to their mail addresses. User passwords will be hashed and then stored. All patient data should be stored anonymously. In other words, any variant, phenotype, and disease should be stored without storing the personal or private information of that patient. The chatbot (LLM) should have no access to user data.

3.4 Pseudo Requirements

- PrioVar will target users across cross-platforms with a web browser on their devices (including phones and computers).
- Git and GitHub will be used for version control purposes [7], [8].
- Jira will be used for issue tracking and overall management of the project [9].
- Java and Spring Boot will be used to maintain the backend of the project, and all the business logic will be implemented through them [10], [11].
- Python will be the main development language for machine learning and inference purposes [12]. The Flask Framework of Python will be utilized to provide communication with the Spring server.
- PyTorch and PyG will be used for training graph neural networks [13], [14].
- The Neo4j graph database will be used to manage the application data, including patient phenotypic and genomic data, and it will also be used to query and extract relevant insights from the population data [6].
- A web domain name, priovar.com, will be utilized as the project hub online.
- React.js will be used in the front end to provide an interface for users on the PrioVar website [15].

- External APIs will be utilized for the initial annotation of the variants, including the known population frequencies of the variant, known related diseases, and some known pathogenicity scores.
- Slack and WhatsApp will be utilized as the asynchronous communication tool, while Zoom and Discord will be used for synchronous project meetings.
- During the development of the product, three-tier architecture will be used for the separation of concerns by dividing the organization into presentation (user interface), application (business logic), and data (persistence) layers.

3.5 System Models

3.5.1 Scenarios

Scenario 1: Health Center Onboarding Process

Actor: Health Center Owner

Entry Condition:

- Opening the registry page.

Exit Condition:

- The actor successfully creates a Health Center account.

Flow of Events:

1. The actor selects sign-up as a Health Center option.
2. The actor fills in the required fields of the Health Center, such as name, address, size, and email.
3. The actor enters the verification code that has been sent to the Health Center's email address to complete the process.

Scenario 2: Clinician Onboarding Process

Actor: Health Center Owner

Entry Condition:

- The actor needs to have a valid Health Center account
- The current number of clinicians associated with the Health Center should be less than the maximum allowed in the current subscription of the Health Center.

Exit Condition:

- An account for the Clinician has been created and linked with the Health Center
- The number of clinicians in the Health Center increases by 1.

Actor: Health Center Owner

Flow of Events:

1. The actor selects to add a co-worker option.
2. The actor fills in the necessary information, such as the name, surname, email, and password of the clinician.

3. The actor enters the verification code that has been sent to the Health Center's email address to complete the process.

Scenario 3: Admin Onboarding Process

Actor: Admin

Entry Condition:

- The system administrator needs to have valid access credentials to the administrative console.

Exit Condition:

- A new administrative account has been successfully created, and access to system settings and configurations has been granted.

Flow of Events:

1. The actor logs into the administrative console.
2. The actor selects the option to add a new admin account.
3. The actor enters the required information, such as username, password, and contact details.
4. The actor enters the verification code sent to the email address to validate the account creation.
5. Upon successful verification, the new admin account is added to the system with access permissions.

Scenario 4: Login

Actor: Clinician, Health Center Owner

Entry Condition:

- Opening the application page and having a valid account.

Exit Condition:

- The actor successfully logs in.

Flow of Events:

1. The actor enters a username
2. The actor enters a password
3. The information inputted by the actor will be checked, and the actor will be logged in to the application.

Scenario 5: Select A Subscription

Actor: Health Center Owner

Entry Condition:

- Having a valid Health Center account.
- The actor's health center shouldn't have an ongoing subscription.

Exit Condition:

- The Health Center has a new subscription.

Flow of Events:

1. The actor selects the subscribe option.
2. The actor makes a selection from a list of subscription options.
3. The actor is re-directed to the payment page.
4. The subscription has been added to the Health Center account.

Scenario 6: Analyzing New Patient Case

Actor: Health Center Owner, Clinician

Entry Condition:

- The number of patients analyzed should be less than the maximum allowed in the current subscription of the Health Center.

Exit Condition:

- The number of patients analyzed in the Health Center increases by 1.

Flow of Events:

1. The actor selects to add a new patient case option
2. The actor uploads the VCF file containing the short-variants of the patient.
3. The actor selects the set of phenotype terms that are related to the patient by selecting from the drop-down.
4. The actor clicks the start annotation processing option.
5. When the processing is finished, the health center (and clinician, if it's the actor) receives a related notification.

Scenario 7: Displaying The Results of Annotated and Prioritized Variants

Actor: Clinician, Health Center Owner

Entry Condition:

- There must be a patient case that has already been processed and annotated belonging to the Health Center.

Exit Condition:

- The list of sorted variants and their annotations, together with pathogenicity scores, are displayed to the actor.

Flow of Events:

1. The actor selects a patient case to analyze from the list of cases.
2. The actor interacts with the list elements, which contain information related to the variants and their annotations, including pathogenicity.

Scenario 8: Listing The Patient Cases Belonging To Health Center

Actor: Clinician, Health Center Owner

Entry Condition:

- There must be a patient case that has already been processed and annotated belonging to the Health Center.

Exit Condition:

- The list of patient cases belonging to the actor's health center is displayed to him/her.

Flow of Events:

1. The actor selects to list the patients belonging to the Health Center.
2. The actor interacts with the list elements, which contain information related to the patient case regarding its number, status, processing date, etc.

Scenario 9: Listing The Patient Cases Belonging To Clinician

Actor: Clinician

Entry Condition:

- There must be at least one patient case that has already been processed and annotated belonging to the Clinician and Health Center.

Exit Condition:

- The list of patient cases belonging to the actor is displayed to him/her.

Flow of Events:

1. The actor selects to only list patients belonging to him/her.
2. The actor interacts with the list elements, which contain information related to the patient case regarding its number, status, processing date, etc.

Scenario 10: Displaying Insights For A Variant

Actor: Clinician, Health Center Owner

Entry Condition:

- There must be at least one patient case that has already been processed and annotated belonging to the Clinician and Health Center.
- The actor must open the list of variants page of a patient

Exit Condition:

- The details of the variant are displayed to the actor.

Flow of Events:

1. The actor selects a variant to analyze from the list of variants of a patient.

2. Detailed information about the selected variant, including its genomic coordinates, type, and associated annotations, is displayed for the actor to interact with.
3. The actor may add notes or comments related to the variant for future reference.

Scenario 11: Querying The Health Center Population Data

Actor: Clinician, Health Center Owner

Entry Condition:

- There must be at least one patient case that has already been processed and annotated belonging to the Clinician and Health Center.

Exit Condition:

- The resulting list of patients and their attributes are displayed to the actor.

Flow of Events:

1. The actor selects the query health center population option.
2. The actor makes a selection from a pre-determined list of filters and also decides on the ranges for numeric filters.
3. The actor clicks on the search option.
4. The actor interacts with the list elements, which contain information related to the patient cases regarding their number, status, processing date, etc.

Scenario 12: Searching For Similar Patients

Actor: Clinician, Health Center Owner

Entry Condition:

- There must be at least one patient case that has already been processed and annotated belonging to the Clinician and Health Center.

Exit Condition:

- The set of patients that are most similar to the selected patient is displayed to the actor, and the relations are formed between these patients to be displayed later.

Flow of Events:

1. The actor selects a patient belonging to the Health Center.
2. The actor selects either phenotype or genotype similarity search among the population data.
3. The search results are displayed to the actor together with their similarity scores.
4. The actor selects one of the patients to access its complete data.

5. A request has been sent to the health center of the resulting patient.
6. After the request has been confirmed by this health center, the actor displays the full patient case and its diagnosis, if exists.

Scenario 13: Creating Population Statistics

Actor: Clinician, Health Center Owner

Entry Condition:

- There must be at least one patient case that has already been processed and annotated belonging to the Health Center of the actor.

Exit Condition:

- The statistics explaining the health center population data are displayed to the actor.

Flow of Events:

1. The actor selects the explore population option.
2. The actor selects the gene, phenotype, and other attributes to calculate the statistics and then clicks on the submitting option.
3. The resulting patients are displayed to the actor together with the frequency of the selection inside the population.

3.5.2 Use Case Model

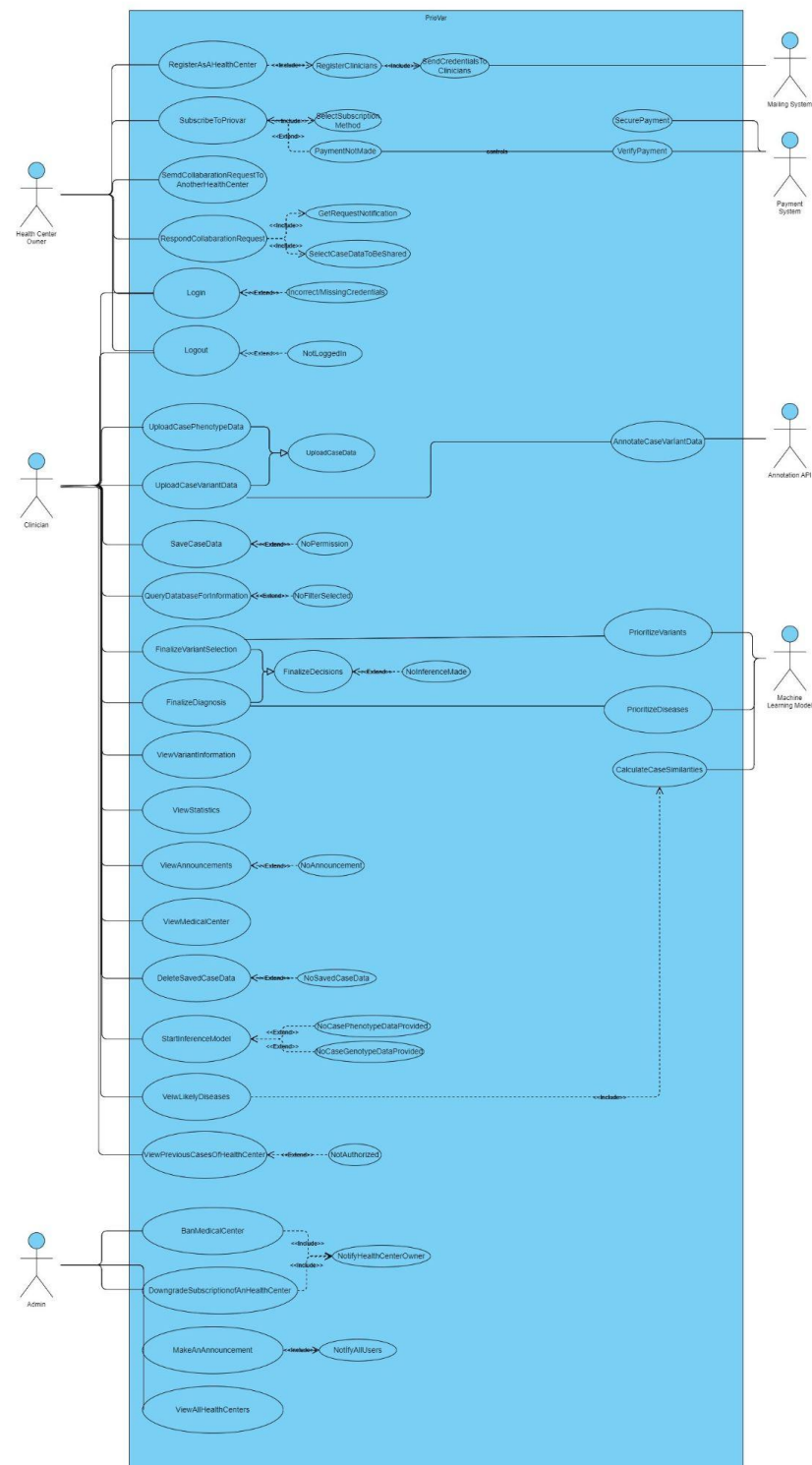


Figure 3: Use Case Diagram for PrioVar (for a Better Image Resolution: <https://imgur.com/a/qJx5tap>)

3.5.3 Object and Class Model

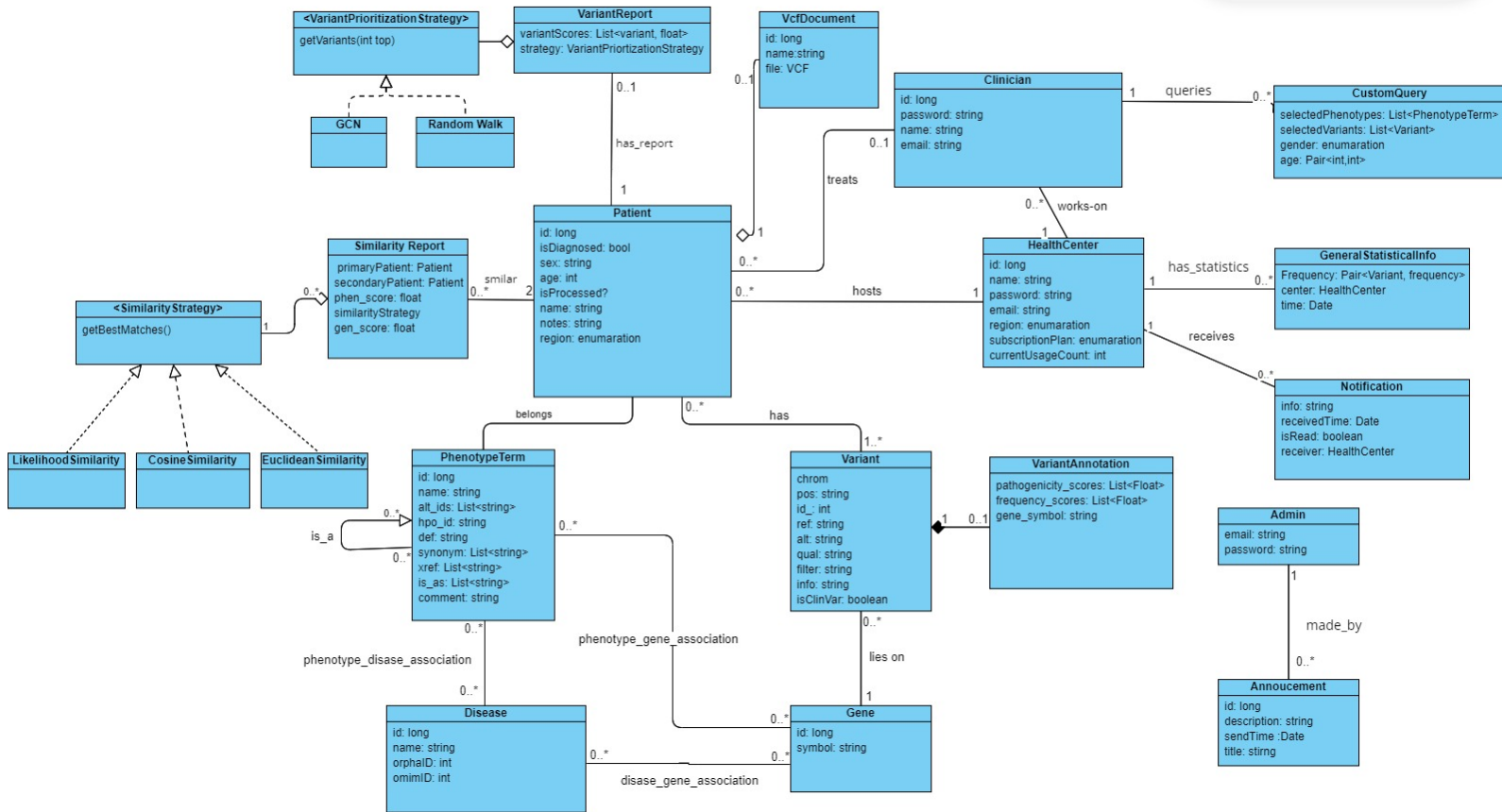


Figure 4: Class Diagram of the Design of PrioVar

3.5.4 Dynamic Models

3.5.4.1 Activity Diagrams

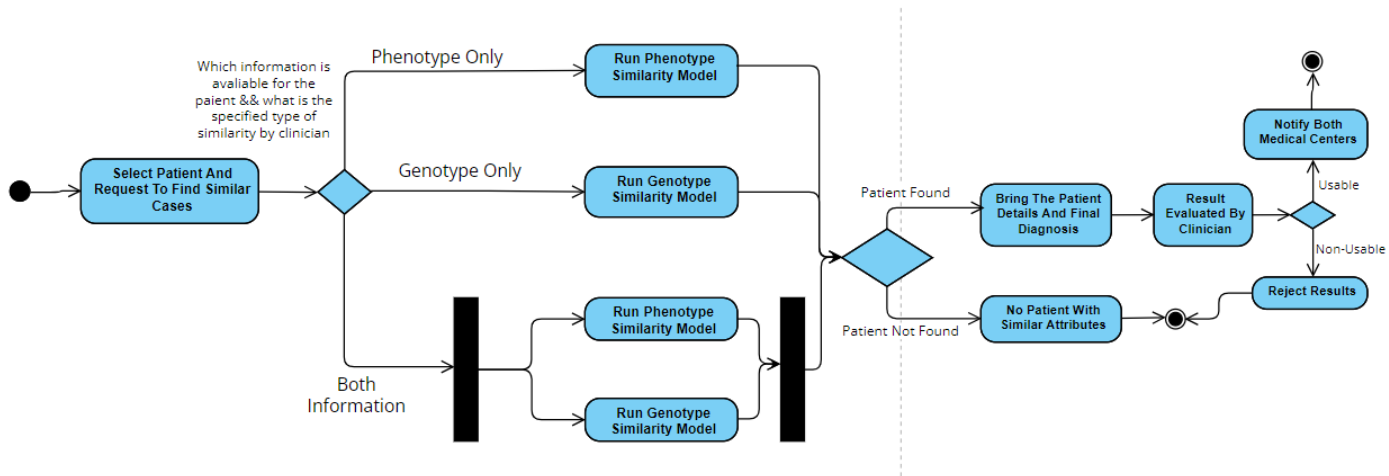


Figure 5: Activity Diagram of Searching Similar Patients

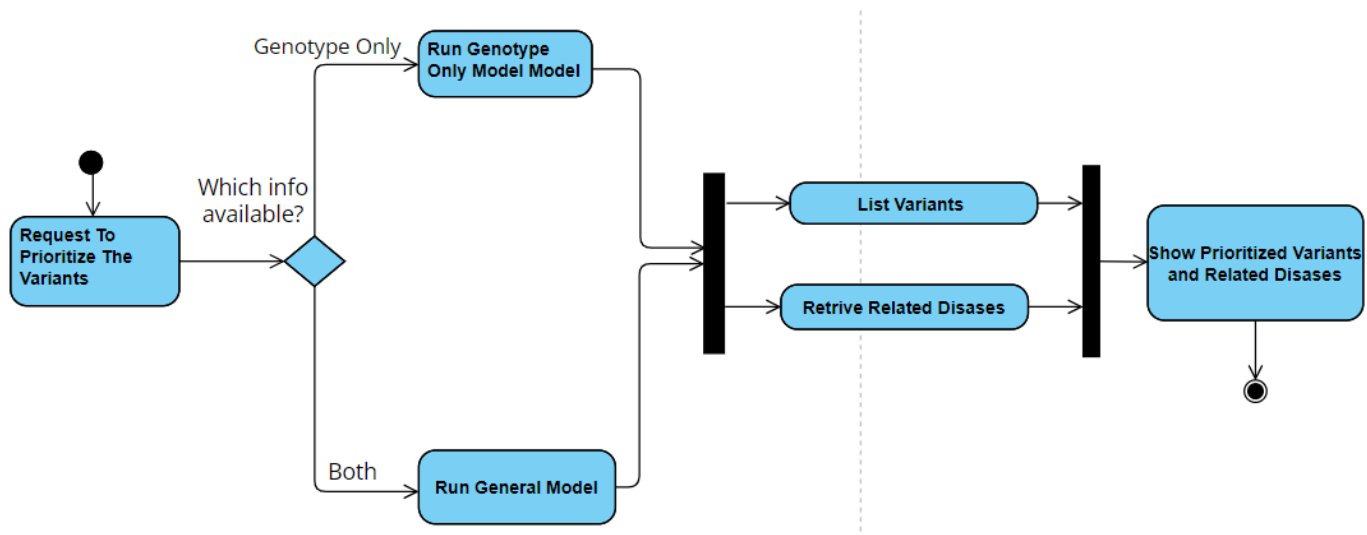


Figure 6: Activity Diagram of Priotizing Variants

3.5.4.1 State Diagrams

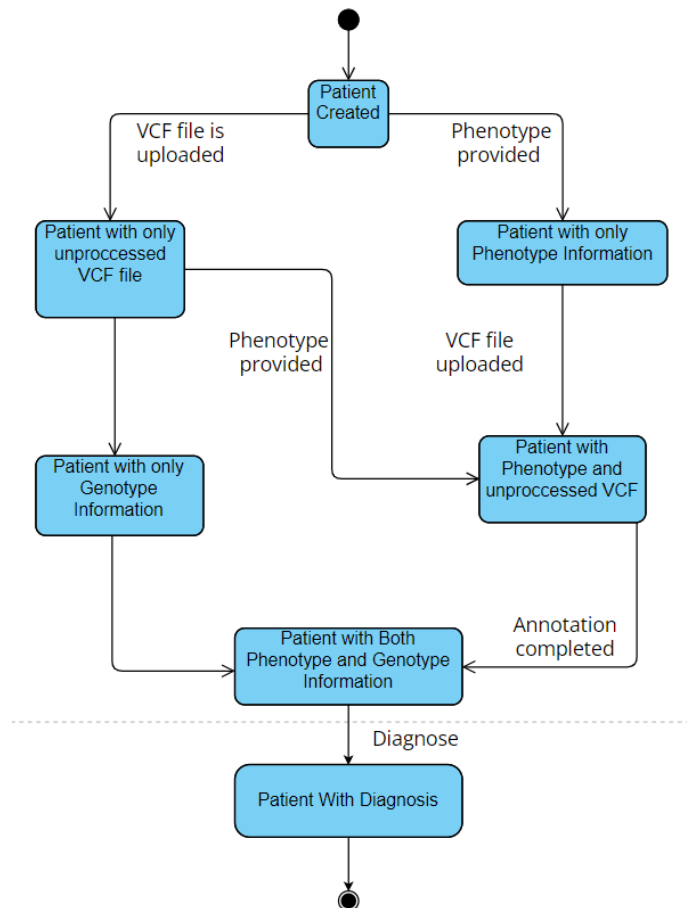


Figure 7: State Diagram of Information Status of a Patient

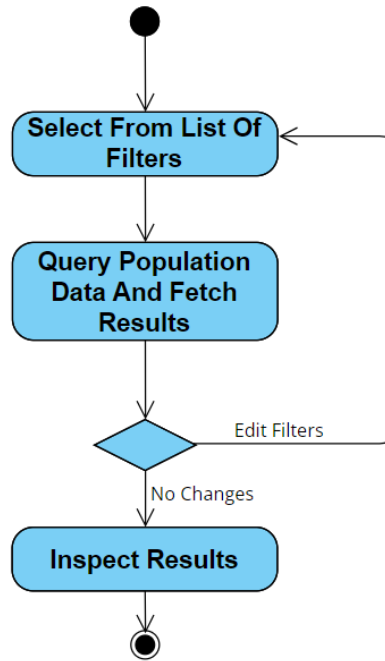


Figure 8: Example HPO Terms
3.5.4.1 Sequence Diagrams

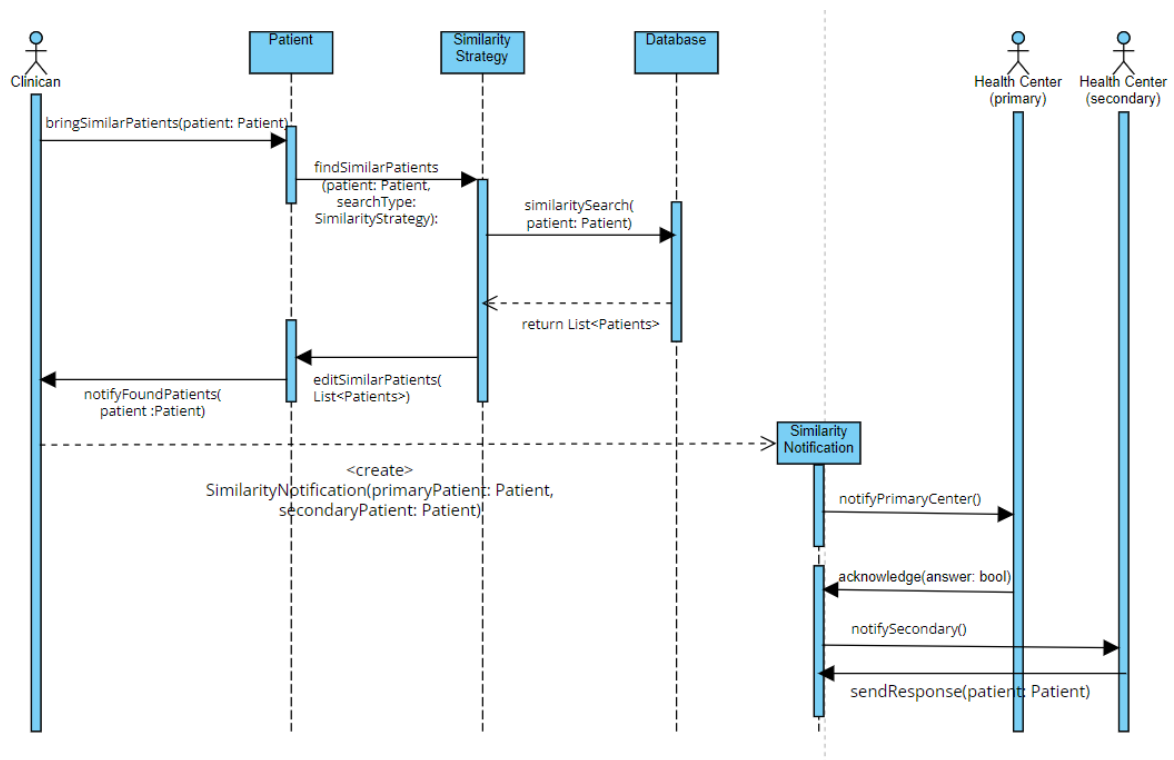


Figure 9: Sequence Diagram of Similar Patient Match

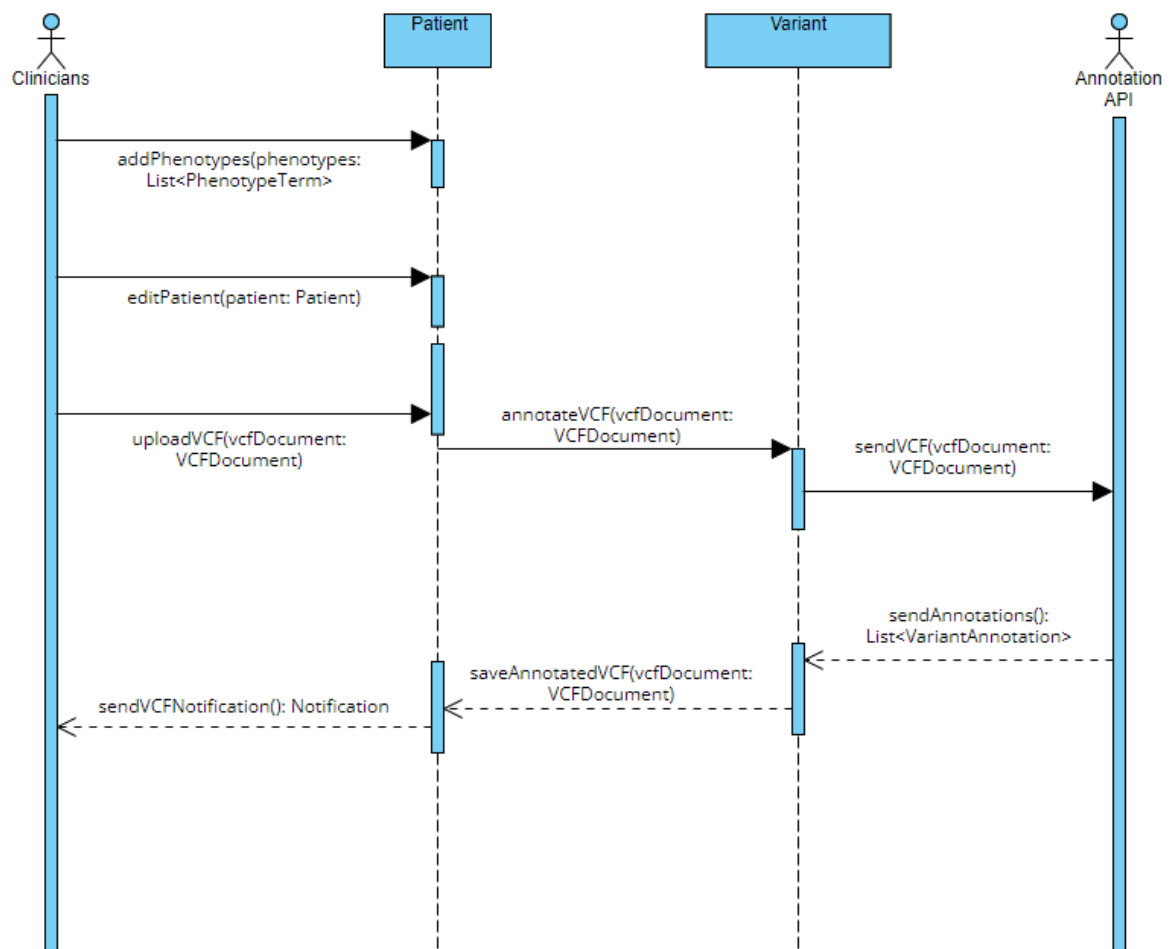


Figure 10: Sequence Diagram of VCF File Annotation

3.5.5 High-Level System Architecture & Components of Proposed Solution

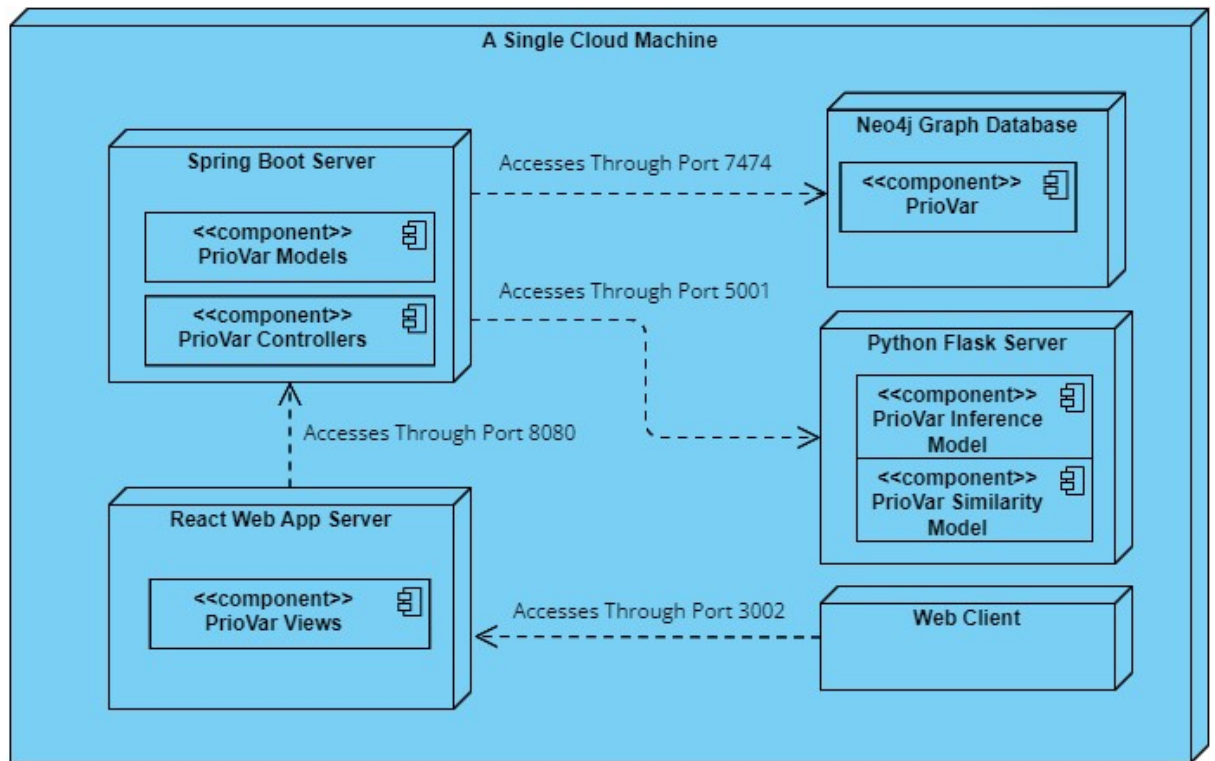


Figure 11: High-Level System Architecture

We've chosen to implement our front-end architecture using React as it provides performance, scalability, and excellent cross-platform support [15]. On the other hand, we chose Spring Boot Framework together with Java as the implementation language [10], [11]. To develop the backend side of our web application, we will use the Spring Boot Framework together with the Neo4j Graph Database [10], [6]. To connect these two technologies, we will use the Spring Data Neo4j to provide a data access layer [6]. To provide a platform that is able to interact with the machine learning model, we plan to use Python Flask Web Server [16]. The results of these inferences will then be communicated back to the Spring Server, enabling real-time responses.

3.5.5 User Interface - Navigational Paths and Screen Mock-ups

PrioVar

Hi, Welcome Back!

Sign in to PrioVar

Clinician Portal Health Center Portal Admin Portal

Login Portal for Clinicians

Email address

Password

Login

Figure 12: Login Screen for Clinicians

PrioVar

Hi, Welcome Back!

Sign in to PrioVar

Clinician Portal Health Center Portal Admin Portal


Login Portal for Clinicians

Email address


Password

Login

Figure 13: Login Screen for Health Centers

**PrioVar**

Hi, Welcome Back!



Sign in to PrioVar

Clinician PortalHealth Center PortalAdmin Portal


Login Portal for Clinicians


Email address

Password


Login

Figure 14: Login Screen for Admins

**PrioVar**




Upload a VCF File



Drop or Select a File
Drop files or click to [browse](#) through your machine

Phenotype Term Selection for Selected File

No file has been selected yet!



Files

Delete

Uploaded At

Completed

Notes

Filename

Sample

Status

Go

Currently, there are no files uploaded!

Figure 15: VCF File Pre-Upload Page

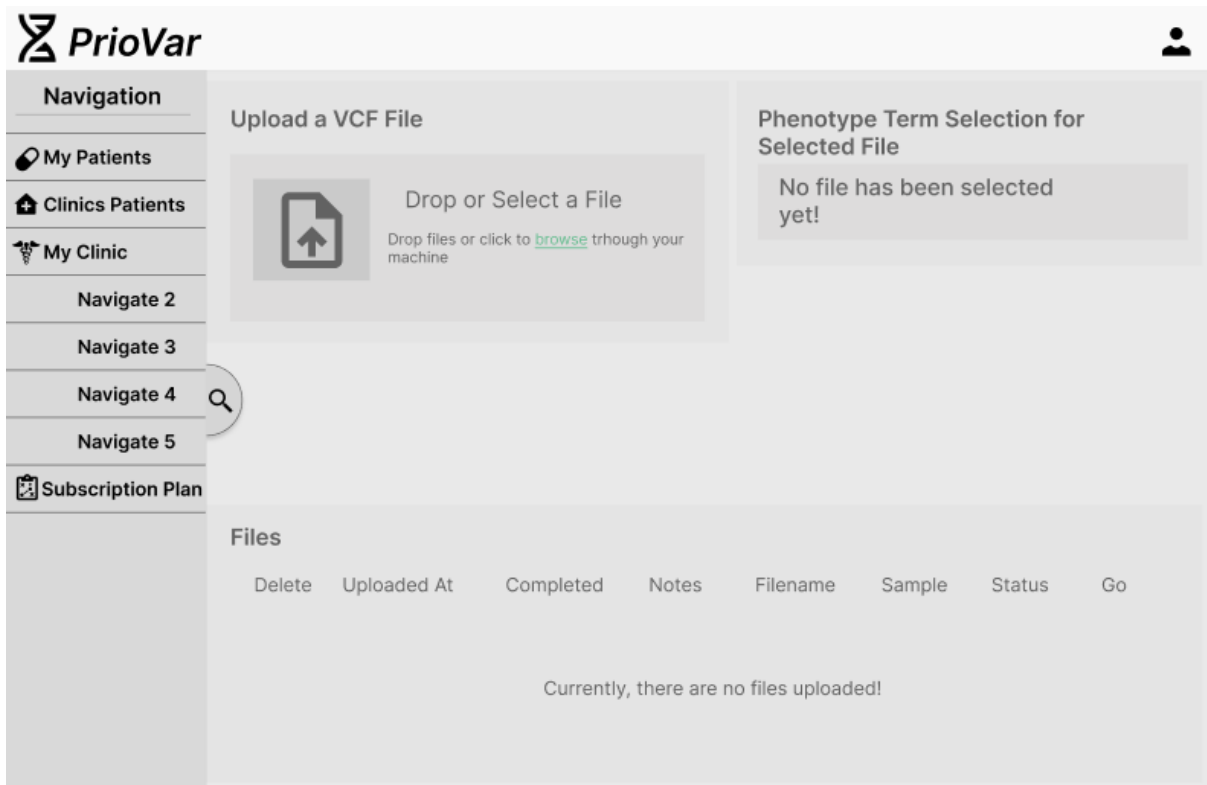


Figure 16: Navigation Bar

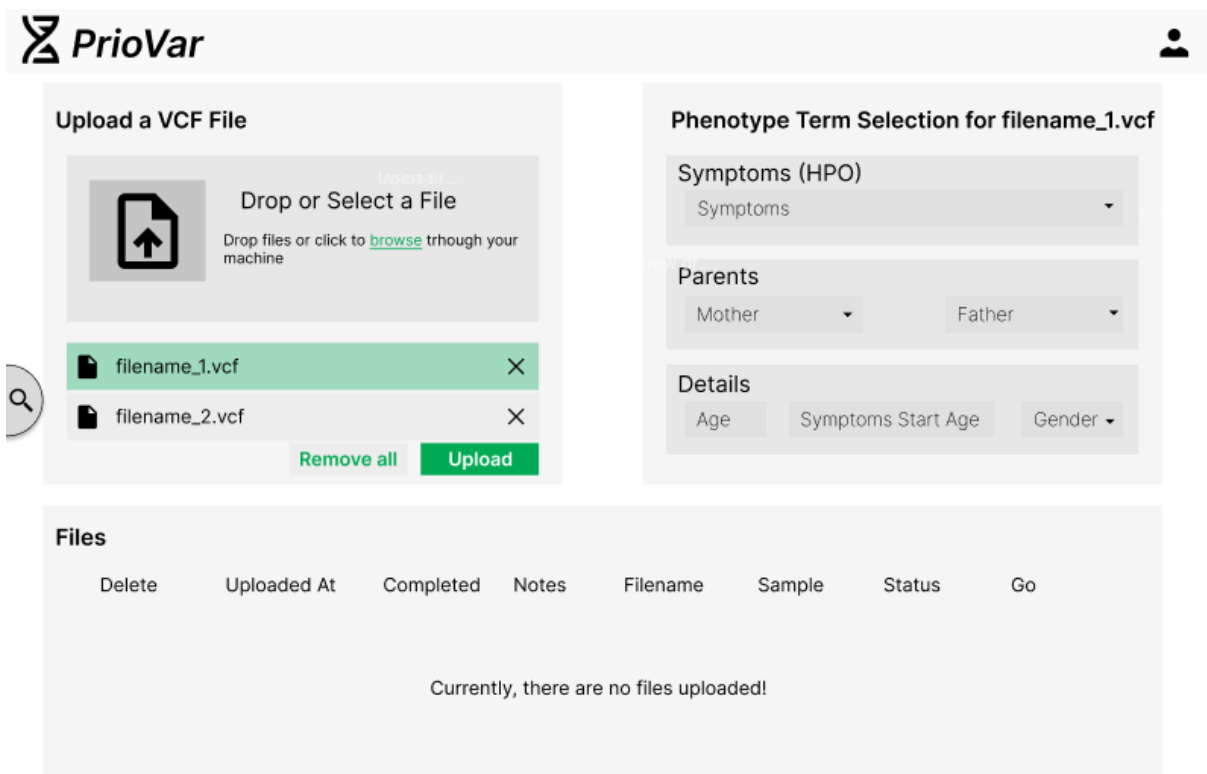





Figure 17: VCF File Upload Process

Upload a VCF File



Drop or Select a File

Drop files or click to [browse](#) through your machine

Phenotype Term Selection for Selected File

No file has been selected yet!







Files							
Delete	Uploaded At	Completed	Notes	Filename	Sample	Status	Go
	05 Dec 2023 21:05	<input type="checkbox"/>		filename_2.vcf	220924	Annotation Required	
	05 Dec 2023 18:05	<input type="checkbox"/>		filename_1.vcf	220923	Analyzed	

Figure 18: VCF File Post-Upload Page



[← Files](#)
[Variant Dashboard](#)
[Selected Variants](#)
[HPO](#)
[Filters](#)
[+ Create a New Table](#)



All Detected Variants of Patient 1 in filename_1.vcf

Pathogenicity	Gene Symbol	Related Diseases	Frequency



Figure 19: Patient Detected Variants Page

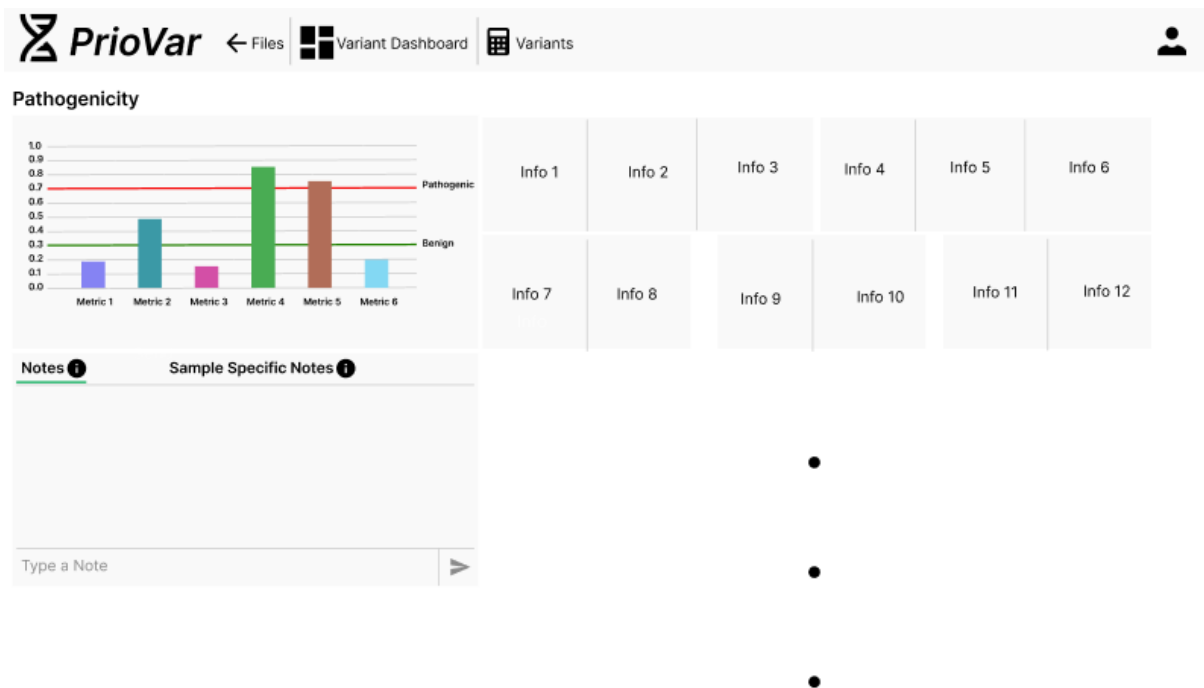


Figure 20: Variant Information Page

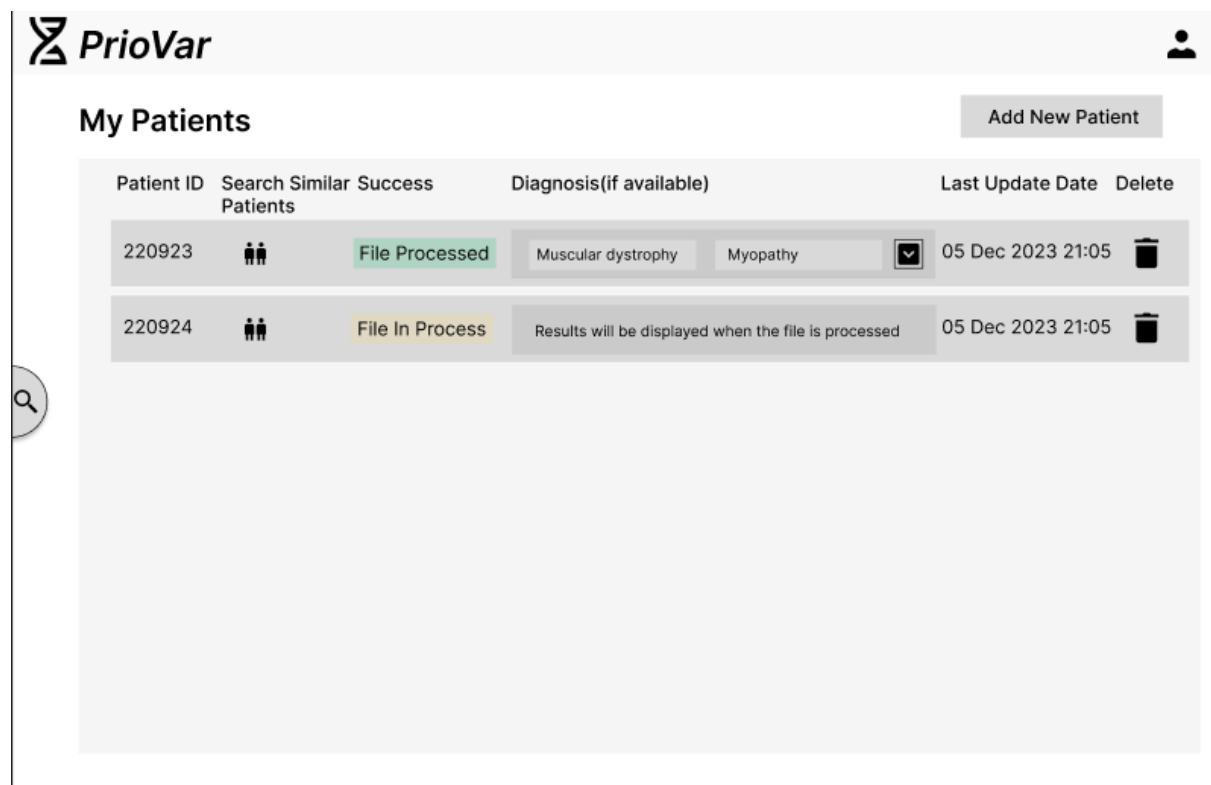



Figure 21: My Patients Page for Clinicians

Clinics Patients

My Patients

Patient ID	Last Update Date	View Details
220923	05 Dec 2023 21:05	
220924	05 Dec 2023 21:05	



Other Patients



Patient ID	Last Update Date	View Details
210927	03 Dec 2023 19:45	
240981	28 Nov 2023 14:23	
230285	26 Nov 2023 08:18	

Figure 24: Clinics Patients Page

Subscription Plans

Junior Packet

Properties included in the plan

- Property 1
- Property 2
- Property 3

109.99\$ / month

Bioinformatician

Properties included in the plan

509.99\$ / month

DEVOURER OF THE GENES

Properties included in the plan

1009.99\$ / month

Figure 25: Subscription Page

4. Other Analysis Elements

4.1. Consideration of Various Factors in Engineering Design

4.1.1. Public Health Considerations

PrioVar is an application that aims to facilitate current health practices in analyzing genetic variants of patients and detecting rare genetic diseases through this process. Consequently, PrioVar considers health to be a highly critical point, as any wrong diagnosis and treatment might have adverse effects on the patients. The algorithms utilized in the platform will be thoroughly tested across various real clinical cases, ensuring their reliability in diverse scenarios. To increase the accuracy of the process, PrioVar will put a strong emphasis on utilizing the most up-to-date data both in the algorithms and the knowledge base of the platform for variants, diseases, and genes. This is especially critical considering the dynamic nature of genomic research; thus, PrioVar will incorporate the data from the latest research findings.

4.1.2. Public Safety Considerations

PrioVar considers public safety to be one of its top priorities during development. The protection of personal data is one of those issues. PrioVar will not permit sharing personal data, including names and passwords, with third parties or publicizing them. Internally, the data will be stored and encrypted while also preventing unauthorized access to it. Since the variants, symptoms, and diagnoses of the patients will be managed in the application, it is sensitive that we protect those data to ensure public safety and instill confidence in the users of the PrioVar platform. The medical centers will only be able to use statistical information without any personal information from other centers. This will also be executed only under the explicit consent of both medical centers, and we will facilitate this process.

4.1.3. Public Welfare Considerations

While PrioVar may not have a direct impact on public welfare, it is essential to recognize the indirect and long-term contributions it can make to the broader well-being of society. By enhancing the accuracy of genetic analysis, PrioVar indirectly supports public welfare through improved healthcare outcomes. It's important to facilitate accurate diagnoses through the PrioVar platform, as it can lead to more targeted and effective treatments, reducing the overall burden on healthcare systems.

4.1.4. Global Considerations

PrioVar will aim to find a marketplace only in Turkey at first, yet the primary language of the app will be English to ensure accessibility to audiences all around the globe. The persisted data in our application will also be in English. Despite the initial focus, PrioVar aims to attract global customers, and as time passes, we believe more languages will be added to the production, together with collaborations with international healthcare centers and experts.

4.1.5. Cultural Considerations

As PrioVar, we target all medical centers from all regions of Turkey. Given the relatively uniform nature of genetical analysis and related medical treatment across cultures within Turkey, the application will not emphasize cultural aspects in the design. Yet, PrioVar recognizes the importance of accessibility and inclusivity in healthcare. PrioVar will ensure that the benefits of the platform reach diverse populations, addressing the disparities through discounts or promotions for accessing the platform.

4.1.6. Social Considerations

PrioVar handles a diverse set of sensitive patient data that comes with social and ethical concerns, so to protect privacy, encrypted and secure storage solutions are needed. The stored data of our application will not contain unique identifiers related to patients, such as name, identification number, etc, ensuring an added layer of anonymity and protection. Secondly, PrioVar won't use our existing patient data to train our machine-learning models and will not employ hidden data collection mechanisms in the platform. Besides, the medical data needed for the model training and constructing a knowledge base will be collected from reputable institutions with proper permissions by emphasizing transparency and adherence to ethical standards.

4.1.7. Environmental Considerations

PrioVar places a strong emphasis on environmental considerations, recognizing that any wrong diagnosis followed by a wrong treatment not only jeopardizes patients but also causes wasteful consumption of resources. PrioVar will also aim to provide clinicians with comprehensive information so that the time to diagnosis will be shorter, thus saving resources. In alignment with sustainability, PrioVar will encourage the medical centers to invest in more environmentally friendly practices during their operations, including obtaining genetic variants and the treatment after the diagnosis.

4.1.8. Economic Considerations

PrioVar offers a wide range of functionalities to its users, yet not all users have access to certain functionalities. Therefore, the system will adopt a tiered monetization structure depending on the user type. For example, the clinicians working in the medical centers will not be paying to use the platform, yet medical centers will have to pay a predetermined monthly fee to use the system. So, PrioVar will be using the subscription business model, and there will be different subscription options related to the number of patient cases to be analyzed through the platform. Also, some additional functionalities, such as chatbot through interacting with a third-party LLM, might require an additional payment. The specific subscription amounts are not determined at this stage. Furthermore, varying regional needs and the impact of inflation rates complicate the establishment of a universal subscription model.

4.1.9 Table of the Aforementioned Considerations

	Effect Degree	Comment
Public Health	9	PrioVar needs to employ best practices and the most up-to-date data both during the algorithm development and knowledge base creation phases
Public Safety	7	PrioVar needs to consider a variety of aspects in its design to ensure public safety and avoid publicization and sharing of personal data.
Public Welfare	1	little to no effect
Global Factors	5	PrioVar should take globalization into account during the development and advertisement phases.
Cultural Factors	2	little effect
Social Factors	9	PrioVar needs to consider the privacy of the patients in its design, and hidden data collection mechanisms and training on patient data will be refrained.
Environmental Factors	5	PrioVar will put emphasis on the accuracy of the processes and also encourage medical centers to use environmentally friendly practices.
Economic Factors	7	PrioVar should put emphasis on economic considerations and arrange monetization to provide inclusiveness.

Table 1: Engineering Considerations

4.1.10 Constraints

4.1.10.1. Economic Constraints

- For us developers, all frameworks and libraries that are to be used are free, and we accessed some of them through verification of being a current student. However, we had to pay ourselves to buy the web domain name. Additionally, we might need to allocate some additional budget if we decide to use some paid servers to host our application and machine learning model. We also plan to look for external funding for the project in order to help us with the budget.
- For the customers, we are planning to make most of the main features of our application free to use. However, some additional functionalities, such as chatbot through interacting with a third-party LLM, might require an additional payment.

4.1.10.2. Privacy and Security Constraints

- First-time users will be able to sign up with a confirmation mail sent to their mail addresses. User passwords will be hashed and then stored.
- All patient data that is stored will be anonymous. In other words, any variant, phenotype, and disease will be stored without storing the personal or private information of that patient.
- The chatbot will have no access to user data.

4.1.10.3. User Experience and Usability Constraints

- Because of the nature of our product, the users are expected to be familiar with technical terms related to variant phenotype data and diseases. However, since the users of this product are already experts from hospitals and clinics, this should not be an issue.
- The application will be user-friendly, and anyone without any technical knowledge will be able to use the application with ease.

4.1.11. Engineering Standards

4.1.11.1. IEEE-International Standard Systems and Software Engineering Software Life Cycle Processes

This standard outlines the necessary processes for a systematic software development life cycle, including planning, development, testing, and maintenance. In our project, we'll integrate these guidelines into the required stages, especially since maintenance will be a clearer objective in the later stages.

4.1.11.2. Unified Modelling Language (UML):

This is a standardized modeling language, which is key for visualizing software system designs. In our project, UML will be used in the design phase for creating standardized diagrams such as object-class diagrams, activity diagrams, sequence diagrams, high-level system architecture, and state diagrams. It will also help us with clear communication since it will present the common ground of our project. This use of UML aids in maintaining clarity and consistency throughout our software design documentation.

4.1.11.3. IEEE-Recommended Practice for Software Requirements Specifications

This standard is for clear, comprehensive software requirements documentation. Our project needs to ensure all functionalities and constraints are understood. We'll use this by systematically gathering and documenting requirements, and collaborating closely with stakeholders to capture every detail accurately.

4.2. Risks and Alternatives

Data Privacy and Security: Genetic and phenotypic data are inherently sensitive, and any data leakage can create severe results. Thus, it is vital to maintain data secure and private. Enhanced access controls, robust encryption, and regular security checks can be employed to establish trust in the system among users and avoid any risks, such as unauthorized access or data breaches.

Accuracy of Variant Prioritization: Depending on the project's success, inaccuracies in variant prioritization could lead to faulty conclusions. Although the final decision is up to the clinician, regular validation against real-world data and continuous refinement of inference algorithms are crucial to avoid unwanted results.

User Adoption and Training: The application's usefulness relies on how effectively the clinicians use it. In other words, although we will provide an intuitive interface, there might still be cases when the application's facilities are not fully utilized. Consequently, insufficient training or resistance to new technology could hinder effective use. To prevent this situation, comprehensive training and tutorials, user-friendly interfaces, and clear documentation can be helpful.

Regulatory Issues: Health norms and standards, which may differ between locations, must be adhered to by the application since it manages personal health data. It is essential to work with legal professionals to manage these complicated obligations and make sure that regional and global standards are followed.

Risk Category	Likelihood	Impact	B Plan Summary
Data Privacy and Security	High	High	Enhanced security protocols and regular checks
Accuracy of Variant Prioritization	Medium	High	Ongoing model validation and updates
User Adoption and Training	Low	High	Comprehensive training and support materials
Regulatory Compliance	High	Medium	Consultation with regulatory experts

Table 2: Risks and Alternatives

4.3. Project Plan

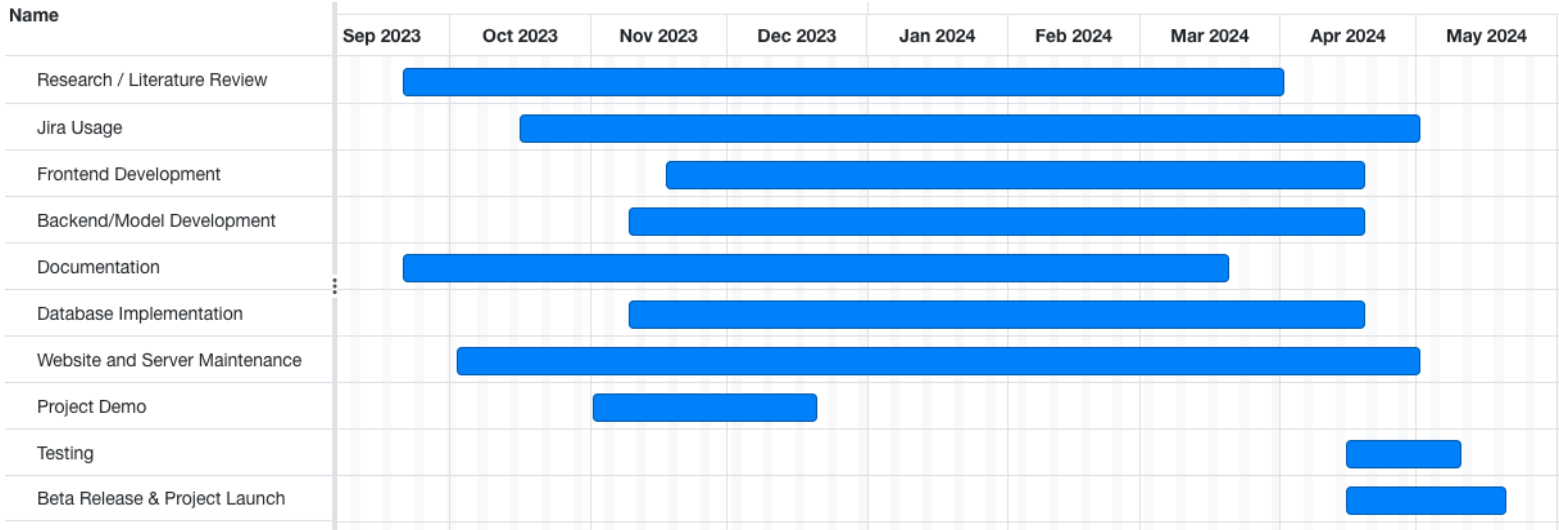


Figure 26: Project Plan Gantt Chart

Work Package	Dates	Leader	Members Involved
Research / Literature Review	20 Sep 2023 - 1 April 2024	Safa Eren Kuday	All members
Jira Usage	15 Oct 2023 - 1 May 2024	Kaan Tek	Erkin Aydın
Frontend Development	18 Nov 2023 - 20 April 2024	Erkin Aydın	Kaan Tek
Backend/Model Development	9 Nov 2023 - 20 April 2024	Halil Alperen Gözetten	Safa Eren Kuday, Korhan Kemal Kaya
Documentation	20 Sep 2023 - 1 May 2024	Kaan Tek	All members
Database Implementation	9 Nov 2023 - 20 April 2024	Korhan Kemal Kaya	Safa Eren Kuday, Halil Alperen Gözetten
Website and Server Maintenance	1 Oct 2023 - 1 May 2024	Erkin Aydın	Halil Alperen Gözetten, Kaan Tek
Project Demo	1 Nov 2023 - 20 Dec 2023	Safa Eren Kuday	All members
Testing	15 April 2024 - 10 May 2024	Korhan Kemal Kaya	All members

Beta Release & Project Launch	15 April 2024 - 20 May 2024	Halil Alperen Gözetin	All members
--	-----------------------------	-----------------------	-------------

WP 1: Research / Literature Review
Leader: Safa Eren Kудay
Members Involved: All members
Start - End Date: 20 Sep 2023 - 1 April 2024
Objectives: Read papers on similar products and models to understand which methods are used and how they perform
Tasks: Task 1.1: Reading papers of similar products Task 1.2 Preparing a presentation to our supervisor Task 1.3: Finding the dataset used by those papers/products Task 1.4: Searching for tools that we can leverage for our connected data Task 1.5: Exploring and contacting large bioinformatic data centers/projects
Deliverables: Deliverable 1.1: Literature review presentation Deliverable 1.2: Method and dataset summary presentation

WP 2: Jira Usage
Leader: Kaan Tek
Members Involved: Erkin Aydın
Start - End Date: 15 Oct 2023 - 1 May 2024
Objectives: Using Jira in order to effectively keep a record of the tasks and the progress
Tasks: Task 2.1: Open a Jira Project Task 2.2: Assign tasks to members Task 2.3: Open and track weekly sprints

Deliverables: Deliverable 2.1: Weekly sprint reports

WP 3: Frontend Development

Leader: Erkin Aydın

Members Involved: Kaan Tek

Start - End Date: 18 Nov 2023 - 20 April 2024

Objectives: Implementation of the Frontend, maximizing user experience
--

Tasks:

Task 3.1: Setup the frontend project, libraries

Task 3.2: Start the implementation

Task 3.3: Design mock-up screens

Task 3.4: Implement requests connecting to backend endpoints
--

Deliverables:

Deliverable 3.1: Mock-up screens

Deliverable 3.2: Web user interface

WP 4: Backend/Model Development
--

Leader: Halil Alperen Gözetin

Members Involved: Safa Eren Kuday, Korhan Kemal Kaya
--

Start - End Date: 9 Nov 2023 - 20 April 2024
--

Objectives: Implementation of the Backend, and the model/algorithm
--

Tasks:

Task 4.1: Setup the backend frameworks/environments

Task 4.2: Writing the model classes

Task 4.2: Start the implementation of the endpoints/algorithms
--

Task 4.4: Creating connection with database and server
--

Deliverables:

Deliverable 4.1: Backend accessible through REST-API endpoints with all needed logic
Deliverable 4.2: Creating Postman workspace for the whole team for managing the endpoints

WP 5: Documentation

Leader: Kaan Tek

Members Involved: All members

Start - End Date: 20 Sep 2023 - 1 May 2024

Objectives: Make sure everything is documented properly to help other developers and end-users to understand the product easily

Tasks:

Task 5.1: Opening the project website

Task 5.2: Writing the Project Specifications Document

Task 5.3: Writing the Analysis and Requirement Report

Task 5.4: Writing the Detailed Design Report

Task 5.5: Writing the Final Report

Deliverables:

Deliverable 5.1: Project website

Deliverable 5.2: Project Specifications Document

Deliverable 5.3: Analysis and Requirement Report

Deliverable 5.4: Detailed Design Report

Deliverable 5.5: Final Report

WP 6: Database Implementation

Leader: Korhan Kemal Kaya

Members Involved: Safa Eren Kuday, Halil Alperen Gözetten

Start - End Date: 9 Nov 2023 - 20 April 2024

Objective: Setup the database

Tasks:

Task: 6.1: Setup Neo4j
Task 6.2: Fill the database

Deliverables:

Deliverable 6.1: A fully functioning database

Deliverable 6.2: Preparing the knowledge-base by adding HPO terms, diseases, genes, variants and their associations to the database

WP 7: Website and Server Maintenance

Leader: Erkin Aydın

Members Involved: Halil Alperen Gözetin, Kaan Tek

Start - End Date: 1 Oct 2023 - 1 May 2024

Objective: To find appropriate server to deploy our code and make sure there will be no problem related to the website after production

Tasks:

Task 7.1: Find the server

Task 7.2: Setting up environment to run our code in the server

Task 7.3: Start using the server by deploying our code

Task 7.4: Regularly check website and the potential bugs

Deliverables:

Deliverable 7.1: A fully functioning project website

Deliverable 7.2: A server successfully hosting our code and database

WP 8: Project Demo

Leader: Safa Eren Kудay

Members Involved: All members

Start - End Date: 1 Nov 2023 - 20 Dec 2023

Objective: Presentation of current status of the project to the stakeholders

Tasks:

Task 8.1: Preparing a presentation

Task 8.2: Rehearsal of the presentation
Deliverables: Deliverable 8.1: Project presentation Deliverable 8.2: Project demo

WP 9: Testing
Leader: Korhan Kemal Kaya
Members Involved: All members
Start - End Date: 15 April 2024 - 10 May 2024
Objective: To make sure the application is ready to use and bug-free
Tasks: Task 9.1: Test all the components Task 9.2: Fix bugs
Deliverables: Deliverable 9.1: Bug reports

WP 10: Beta Release & Project Launch
Leader: Halil Alperen Gözetin
Members Involved: All members
Start - End Date: 15 April 2024 - 20 May 2024
Objective: Get the project to a state that is ready to be used by the end-users
Tasks: Task 9.1: Prepare user manual Task 9.2: Launch Task 9.3: Testing on-site
Deliverables: Deliverable 9.1: User manual Deliverable 9.2: Product

4.4. Ensuring Proper Teamwork

When designing our project, one of our biggest priorities is that everyone takes on as equal a role as possible. As mentioned in the previous section, we divided the project process into work packages, but we want everyone to have done an equal amount of work in total. In such a large and comprehensive project, it may be difficult to give everyone an equal amount of work and to keep track of everyone's tasks. Unless this is considered, this can also lead to ineffective communication and, thus, information loss. In order to prevent these from happening, we have been using the following tools during the process:

GitHub: We are using GitHub as our version control tool in order for each member to effectively contribute to the project synchronously [8]. Everyone can work on their own bit without interrupting others' work. It is also useful in tracking the contributions of each member.

Jira: In order to keep track of work packages and every smaller subtask inside those packages, we decided to use Jira. It is a very useful tool to plan the process and move organized. It is also useful in tracking who is doing what [9]. We are following the Scrum methodology and using sprints to plan our tasks and assign time periods for our tasks to be completed [17]. We set our sprints to be 1 or 2 weeks long, depending on our school-related workload during that time period.

Google Docs: We are also using Google Docs to keep track of meeting notes and share our findings [18]. In every meeting that we have with our supervisor and course coordinators, we take turns writing key points and to-dos in a detailed manner. Then, we transfer these notes from Google Docs to Jira.

Postman: We are benefiting from the workspace feature of the Postman to create and work on the endpoints during the development phase, which enhances collaboration and prevents confusion [19].

4.5. Ethics and Professional Responsibilities

While designing our project, our top concerns are user anonymity and data protection. All genetic information, patient symptoms, and illnesses will be anonymously stored. Hospitals will only be able to utilize this data to exchange information with other hospitals or clinics without revealing any personal information or to get a sense of the diseases that are prevalent for certain genetic or phenotypic data.

Furthermore, as our application will be based on a trained model and we are working on sensitive issues directly related to people's health, it is our responsibility to develop a model that produces the most accurate findings possible. To ensure that our final model operates satisfactorily, we will optimize all the parameters,

conduct several tests, and compare its performance with other similar products before launching our product.

Additionally, we will make sure everyone finishes their assigned work and tasks in their given time unless an odd exception happens. It is also each of our duties to produce clean code and make the necessary tests before publishing the product.

4.6. Planning for New Knowledge and Learning Strategies

We were to learn new technologies and methodologies for our project, and at this point, we already learned most of the stuff. For the coding and more technical side of our project, some of our members had to learn new libraries and frameworks, depending on their work. For the Frontend side, our team already had some experience with the necessary frameworks, such as ReactJS, but we still watched a few online videos as a reminder and went through the documentation when needed. Regarding the backend, the learning curve was a bit higher since the technologies used here were unfamiliar to us, such as Neo4j. Our members watched online tutorials and examined similar projects, which included Neo4j to grasp the idea and improved themselves even more during the actual implementation via hands-on experience. Additionally, we also learned how to use Jira effectively in order to get the best out of this tool.

5. Glossary

Variant: The term variant or genetic variant is used to describe a subtype of a microorganism that is genetically distinct from a main strain but not sufficiently different to be termed a distinct strain. The types include SNP (Single-Nucleotide Polymorphism) and CNV (Copy-Number Variant) [20].

HPO: The Human Phenotype Ontology (HPO) provides a standardized vocabulary of phenotypic abnormalities encountered in human disease. Each term in the HPO describes a phenotypic abnormality [5].

Variant Prioritization: Variant prioritization is a procedure that is commonly used in clinical studies to reduce the number of genetic variants that need to be evaluated manually according to a defined metric [21].

VCF File Format: The Variant Call Format (VCF) is a standard text file format used in bioinformatics for storing gene sequence variations [4].

Graph Database: A graph database (GDB) is a database that uses graph structures for semantic queries with nodes, edges, and properties to represent and store data [6].

REST API: A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for

representational state transfer and was created by computer scientist Roy Fielding [22].

6. References

- [1] "About Us," Genomize, https://genomize.com/about_us/ (accessed Dec 7, 2023).
- [2] "Engenome," engenome, <https://www.engenome.com/> (accessed Dec 7, 2023).
- [3] "Home," Big Data and AI Driving Rare Disease Diagnosis: centogene.com, <https://www.centogene.com/science/whitepapers/centogenes-variant-prioritization-big-data-and-ai-driving-rare-disease-diagnosis> (accessed Dec 7, 2023).
- [4] Embl-Ebi, "Understanding VCF format," Understanding VCF format | Human genetic variation, <https://www.ebi.ac.uk/training/online/courses/human-genetic-variation-introduction/variant-identification-and-analysis/understanding-vcf-format/> (accessed Dec 7, 2023).
- [5] Human phenotype ontology, <https://hpo.jax.org/app/> (accessed Dec 7, 2023).
- [6] "Neo4j graph database & analytics – the leader in graph databases," Graph Database & Analytics, <https://neo4j.com/> (accessed Dec 7, 2023).
- [7] Git, <https://git-scm.com/> (accessed Dec 7, 2023).
- [8] "Let's build from here," GitHub, <https://github.com/> (accessed Dec 7, 2023).
- [9] "Move fast, stay aligned, and build better - together," Atlassian, <https://www.atlassian.com/software/jira> (accessed Dec 7, 2023).
- [10] Java.com, <https://www.java.com/en/> (accessed Dec 7, 2023).
- [11] "Spring boot3.1.5," Spring Boot, <https://spring.io/projects/spring-boot> (accessed Dec 7, 2023).
- [12] "Welcome to Python.org," Python.org, <https://www.python.org/> (accessed Dec 7, 2023).
- [13] "Membership available," PyTorch, <https://pytorch.org/> (accessed Dec 7, 2023).
- [14] "Home," PyG, <https://pyg.org/> (accessed Dec 7, 2023).
- [15] React, <https://react.dev/> (accessed Dec 7, 2023).

- [16] "Torchserve", TorchServe - PyTorch/Serve master documentation, <https://pytorch.org/serve/> (accessed Dec 7, 2023).
- [17] "What is Scrum?", Scrum.org, <https://www.scrum.org/resources/what-scrum-module> (accessed Dec 7, 2023).
- [18] "Build your best ideas together, in Google Docs", Google, <https://www.google.com/docs/about/> (accessed Dec 7, 2023).
- [19] "What is Postman?", Postman, <https://www.postman.com/product/what-is-postman/> (accessed Dec 7, 2023).
- [20] "Variant identification and analysis", Human genetic variation, <https://www.ebi.ac.uk/training/online/courses/human-genetic-variation-introduction/variant-identification-and-analysis/> (accessed Dec 7, 2023).
- [21] "Settling the score: variant prioritization and Mendelian disease", nature review genetics, <https://www.nature.com/articles/nrg.2017.52> (accessed Dec 7, 2023).
- [22] "What is a REST API?", RedHat, <https://www.redhat.com/en/topics/api/what-is-a-rest-api> (accessed Dec 7, 2023).