Bilkent University

CS-491

Senior Design Project

Fall 2023/2024

Project Specifications Report

Group “PrioVar” - T2309

Supervisor: Ercüment Çiçek

Innovation Expert: Can Alkan

Jury Members: Atakan Erdem & Mert Bıçakçı

Member Name	Student ID
Korhan Kemal Kaya	21903357
Halil Alperen Gözeten	21902464
Kaan Tek	21901946
Erkin Aydın	22002956
Safa Eren Kuday	21902416

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 knowledge of the clinicians, 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.

1.1 Description

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 tailored 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 [1]. 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      NA00001        NA00002        NA00003
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 [2]. There exist around 15000 HPO terms, which are arranged in a directed acyclic graph and are connected by edges such that a term represents a more specific of its parent [2]. 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 will be provided to the clinicians on a separate page, where they will be able to access the associated diseases/functions of the prioritized variants, which will support the final diagnosis phase.

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 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.

1.2 High-Level System Architecture & Components of Proposed Solution

The system that we are going to design will be a lightweight website with low consumption of the system resources and will not have a bare minimum as the hardware specification but will depend on the requirements of the browser used. Computers, smartphones, and tablets with browsers will be capable of accessing and interacting with our website. For a computer that will access our system, the most basic requirement will be a pre-installed proper-working browser.

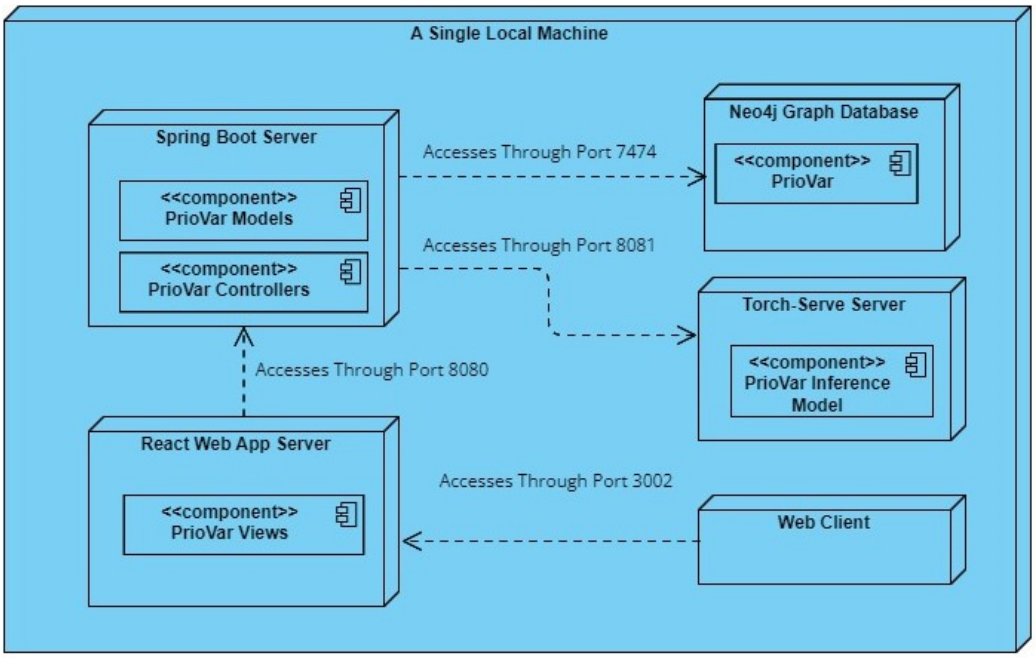

Figure 3: *High-Level System Architecture*

We've chosen to implement our front-end architecture using React as it provides performance, scalability, and excellent cross-platform support. Also, it's easy to adopt, has a small learning curve, and facilitates UI-focused designs, making it one of the best options for building an interactive user interface [3]. On the other hand, we chose Spring Boot Framework together with Java as the implementation language in order to deliver rapid and reliable implementations [4], [5]. Spring Boot helps to complete most of the job through simple annotations when implementing endpoints, and it also provides automated queries to the database [4]. Also, it has a small learning curve compared to other server-side frameworks, and the previous experiences of our team make it an even more profitable option.

For developing the backend side of our web application, we will use the Spring Boot Framework together with the Neo4j Database, a graph database [4], [6]. One of the main reasons for us to pick Neo4j as our database was its great support for connected data and its seamless connection with Spring Services [6]. To connect these two technologies, we will use the Spring Data Neo4j to provide a data access

layer [6]. This will provide an effortless implementation and improve the persistence code by only providing the repository interfaces and without writing complex SQL queries. The Spring Data Neo4j will provide automated implementations for specific queries once finder method signatures are written. Also, through the @Query annotation of Spring Data Neo4j, it's possible to write a custom query which will allow for even more fine-grained operations.

To provide a platform that is able to make interactions with the machine learning model, we plan to use TorchServe as a serving library that facilitates the deployment of models [7]. It simplifies the process of serving machine learning models by providing a communication platform through REST API endpoints. The results of these inferences will then be communicated back to the Spring, enabling real-time responses.

1.3 Constraints

1.3.1. Implementation Constraints

- Git and GitHub will be used for version control purposes [8], [9].
- Jira will be used for issue tracking and overall management of the project [10].
- Java and Spring Boot will be used to maintain the backend of the project, and all the business logic will be implemented through them [4], [5].
- Python will be the main development language for machine learning and inference purposes [11].
- PyTorch and PyG will be used for training graph neural networks [12], [13].
- The Neo4j graph database will be used for managing the patient phenotypic and genomic data, and it will also be used for querying and extracting relevant insights from the population data [6].
- TorchServe will be used for deploying the machine learning [7].
- 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 [3].

1.3.2. 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.

1.3.3. 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.

1.3.4. User Experience and Usability Constraints

- Because of the nature of our product, the users are expected to have familiarity 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.

1.4 Professional and Ethical Issues

User anonymity and data privacy are one of our main priorities while designing the application. As mentioned earlier, all of the genetic data, symptoms, and diseases of the patients will be stored anonymously. Hospitals will only be able to use this information to have an idea of what kind of diseases are common for what genetic/phenotypic information, or share information with other hospitals/clinics without any personal information included, etc.

Besides, since there will be a trained model at the heart of our application, and we are doing sensitive work that is directly related to people's health, it is our duty to come up with a model that yields correct results as much as possible. Before releasing our product, we will optimize all the parameters, do plenty of tests, and compare its performance with other similar products in order to make sure our final model performs satisfactorily well.

2. Design Requirements

2.1 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 phenotypic features of the patient 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 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.
- 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.
- 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 an existing patient.
- 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.

2.2 Non-Functional Requirements

2.2.1 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.

2.2.2 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.

2.2.3 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.

2.2.4 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 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.

2.2.5 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. Feasibility Discussions

We evaluate the feasibility of PrioVar in two aspects: Its feasibility from the market perspective and its feasibility from the academic perspective. The previous implementations of variant prioritization methods in these fields will be examined.

3.1 Market & Competitive Analysis

In the market, there are companies that either focus on or use variant prioritization, such as Genomize, Engenome, and Centogene [14], [15], [16]. Here, we will go over these companies and measure their success to determine the feasibility of PrioVar on the market.

3.1.1. Genomize

Genomize is the developer and the owner of the SEQ Platform [14]. They claim that the pipeline the SEQ Platform is using has a 97% successful variant prioritization rate. Their variant prioritization pipeline uses a 5-tier variant prioritization classification, which utilizes the patient's clinical information, such as observed phenotypes [17], as we also try to do with PrioVar.

3.1.2. Engenome

Engenome is the owner and the developer company of the eVai platform that automates ACMG guidelines and prioritizes variants to highlight candidate diagnosis [15]. They assign a pathogenicity score to each variant for this purpose. They also provide various services, such as family analysis for up to 7 members, in which eVai automatically infers the "possible inheritance patterns according to a fully penetrant disease[18]".

3.1.3. Centogene

Centogene defines itself as “the rare disease company,” founded in 2006, which focuses on the diagnosis of rare diseases. Variant prioritization is one of the techniques which they seriously focus on. For this purpose, they created the CentoMD® data repository, and they claim that it is the world’s largest curated mutation database for rare diseases. CentoMD® data repository has more than 7.3 million variants and a significant number of unpublished variants [16].

As can be seen, there are companies who work on both genotype and phenotype-driven variant prioritization in the market, and therefore, our app PrioVar seems quite feasible, considering that it will also be grounded in similar techniques.

3.2. Academic Analysis

There are various methods and papers published in variant prioritization. Xrare and Exomiser are two of them. Here, we analyze these methods to figure out the feasibility of PrioVar.

3.2.1. Exomiser

Exomiser has been developed as a variant prioritization pipeline and includes four genotype and phenotype-driven variant prioritization methods: PhenIX, PHIVE, hiPHIVE, and ExomeWalker. While the first two use the patient's genome and phenotypic anomalies as input, ExomeWalker uses only the genetic information [19]. As for analysis, we briefly explain what hiPHIVE and ExomeWalker do, as they are closely related to our plans for PrioVar.

“ExomeWalker” is one of the methods used by Exomiser. The ExomeWalker algorithm is designed to identify new causative genes by identifying which of the mutated genes interacts closely with previously known disease-associated genes. “The user supplies the list of implicated or suspected seed genes, and a random walk with restart algorithm is used to score how close each candidate gene is to these in a protein-protein association network.” Hence, the ExomeWalker algorithm can be run when the user can provide a set of seed-genes as input[19].

“hiPHIVE” is the last pipe of the Exomiser pipeline. Human, mouse, and zebrafish phenotype data are used to calculate phenotypic similarity. For the genes that are found to have phenotypic data, disease-gene associations can be detected with high sensitivity. For

the genes that have no phenotype data from any of the human, mouse, or zebrafish phenotype data, a random walk with restart algorithm run to score how close the candidate is in a protein-protein association network to genes with strong phenotypic similarity to the patient [19].

3.2.2. XRare

Xrare is a machine-learning approach to disease-causing variant prioritization based on a rich set of phenotypic and genetic features. Xrare uses 10 interactome graphs to create 10 predictors for phenotype-gene similarity. An XGBoost model (gradient boosting decision tree) has been trained on genes found between 2005 and 2011, and it has been used to predict the gene-phenotype similarity scores for non-seed (novel) genes. Xrare is claimed to perform even better and ranks 23% more causal variants at the top than the Exomiser hiPHIVE algorithm [20].

Exomiser and Xrare are just two of the methods developed in academia for variant prioritization. Most of these methods are open-source, and their data are available on the internet. Hence, we have direct access to previous developments in this field. Therefore, due to the availability of academic content in this field, we claim that PrioVar is feasible in this perspective.

4. References

[1] 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 Nov. 17, 2023).

[2] Human phenotype ontology, https://hpo.jax.org/app/ (accessed Nov. 17, 2023).

[3] React, https://react.dev/ (accessed Nov. 17, 2023).

[4] “Spring boot3.1.5,” Spring Boot, https://spring.io/projects/spring-boot (accessed Nov. 17, 2023).

[5] Java.com, https://www.java.com/en/ (accessed Nov. 17, 2023).

[6] “Neo4j graph database & analytics – the leader in graph databases,” Graph Database & Analytics, https://neo4j.com/ (accessed Nov. 17, 2023).

[7] “Torchserve¶,” TorchServe - PyTorch/Serve master documentation, https://pytorch.org/serve/ (accessed Nov. 17, 2023).

[8] Git, https://git-scm.com/ (accessed Nov. 17, 2023).

[9] “Let’s build from here,” GitHub, https://github.com/ (accessed Nov. 17, 2023).

[10] “Move fast, stay aligned, and build better - together,” Atlassian, https://www.atlassian.com/software/jira (accessed Nov. 17, 2023).

[11] “Welcome to Python.org,” Python.org, https://www.python.org/ (accessed Nov. 17, 2023).

[12] “Membership available,” PyTorch, https://pytorch.org/ (accessed Nov. 17, 2023).

[13] “Home,” PyG, https://pyg.org/ (accessed Nov. 17, 2023).

[14] “About Us,” Genomize, https://genomize.com/about_us/ (accessed Nov. 17, 2023).

[15] “Engenome,” engenome, https://www.engenome.com/ (accessed Nov. 17, 2023).

[16] “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 Nov. 17, 2023).

[17] “AI-driven NGS Data Analysis: A variant prioritization pipeline for precision medicine,” Genomize, https://genomize.com/ai-driven-ngs-data-analysis-a-variant-prioritization-pipeline-for-precision-medicine/ (accessed Nov. 17, 2023).

[18] G. Nicora, S. Zucca, I. Limongelli, R. Bellazzi, and P. Magni, “A machine learning approach based on ACMG/AMP Guidelines for genomic variant classification and prioritization,” *Scientific Reports*, vol. 12, no. 1, 2022. doi:10.1038/s41598-022-06547-3

[19]D. Smedley *et al.*, “Next-generation diagnostics and disease-gene discovery with the Exomiser,” *Nature Protocols*, vol. 10, no. 12, pp. 2004–2015, 2015. doi:10.1038/nprot.2015.124

[20] Q. Li, K. Zhao, C. D. Bustamante, X. Ma, and W. H. Wong, “Xrare: A machine learning method jointly modeling phenotypes and genetic evidence for rare disease diagnosis,” *Genetics in Medicine*, vol. 21, no. 9, pp. 2126–2134, 2019. doi:10.1038/s41436-019-0439-8