Software Requirements Specification (SRS)

Team: DFS_Phodenge

Members: 1. Nikhil Agrawal (2020102021)
              2. Mitul Garg (2020102026)
              3. Atharv Sujjlegaonkar (2020102025)

Introduction

1.1 Purpose

The purpose of this Requirement Specification Document is to define the requirements for a healthcare knowledge platform that provides patients with access to their layered healthcare data.

1.2 Scope

The scope of this project includes the requirements for the platform's functionality, performance, and user interface.

1.3 Background

The healthcare knowledge platform is intended to provide patients with access to their healthcare data, including basic data, medical observations, blood reports, genome sequence data, and more.

1.4 Glossary of Terms

A list of terms used in this SRS, along with their definitions:

Basic Patient Data: Demographic information, medical history, and contact information for a patient.

Medical Observations: Vital signs, physical examination findings, and diagnostic imaging results.

Blood Reports: Complete blood counts, blood chemistries, and other blood test results.

Genome Sequence Data: Whole genome sequences, exome sequences, and targeted gene panels.

Healthcare Providers: Doctors, nurses, and other healthcare professionals who provide care to patients.

Authorized Individuals: Individuals who have been authorized by the patient to access their healthcare data.

Functional Requirements

2.1 Patient Data Management

The platform must be able to store and manage basic patient data, including demographic information, medical history, and contact information.

2.2 Medical Observations

The platform must be able to store and manage medical observations, including vital signs, physical examination findings, and diagnostic imaging results.

2.3 Blood Reports

The platform must be able to store and manage blood reports, including complete blood counts, blood chemistries, and other blood test results.

2.4 Genome Sequence Data

The platform must be able to store and manage genome sequence data, including whole genome sequences, exome sequences, and targeted gene panels.

2.5 Data Access and Sharing

Patients must be able to access and share their healthcare data with authorized healthcare providers and other authorized individuals.

Non-Functional Requirements

3.1 Performance

The platform must be able to support a large number of concurrent users and must have a response time of fewer than 2 seconds.

3.2 Security

The platform must have strong security measures in place to protect the confidentiality and privacy of patient data.

3.3 Usability

The platform must have a user-friendly interface that is easy for patients to use.

3.4 Interoperability

The platform must be able to interface with other healthcare systems and data sources to ensure that patient data is up-to-date and accurate.

User Interface Requirements

4.1 Screen Layout

The platform must have a clean and intuitive screen layout, with a navigation menu and clear information hierarchy.

4.2 Color Scheme

The platform must use a color scheme that is easy on the eyes and consistent throughout the platform.

4.3 Fonts

The platform must use clear and legible fonts that are consistent throughout the platform.

Design Document

High-level approach
Overview: The healthcare knowledge graph will be built using a semantic web approach, utilizing knowledge representation languages like RDF and OWL. The graph will be built on top of a graph database, allowing for efficient and flexible querying.

Key Features: The knowledge graph will allow patients to view their healthcare data in a layered manner, allowing for deeper insights and analysis
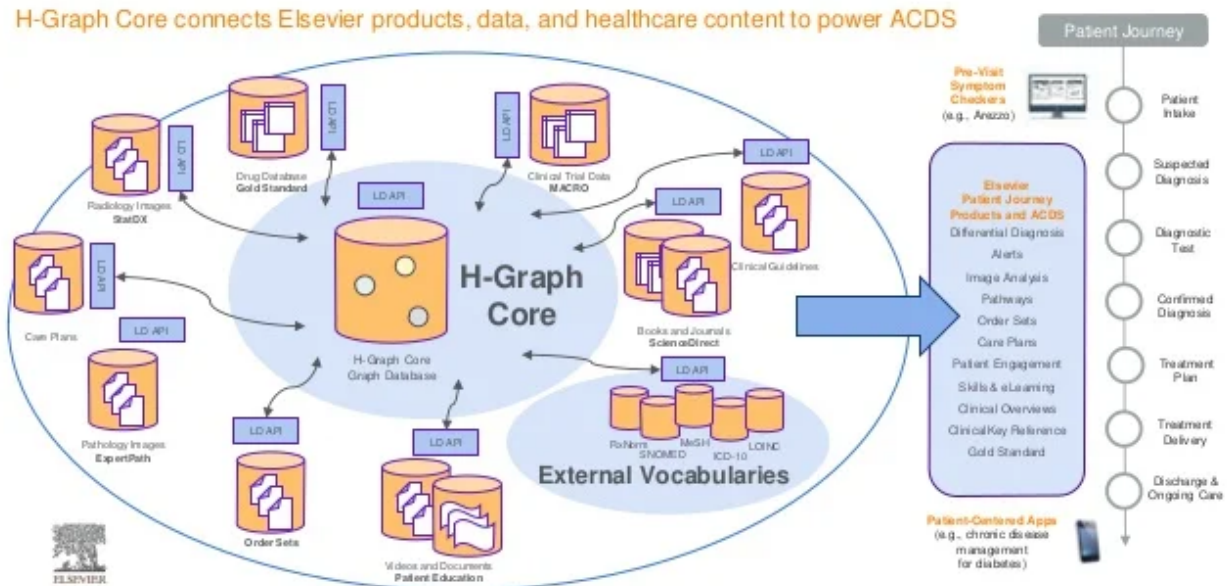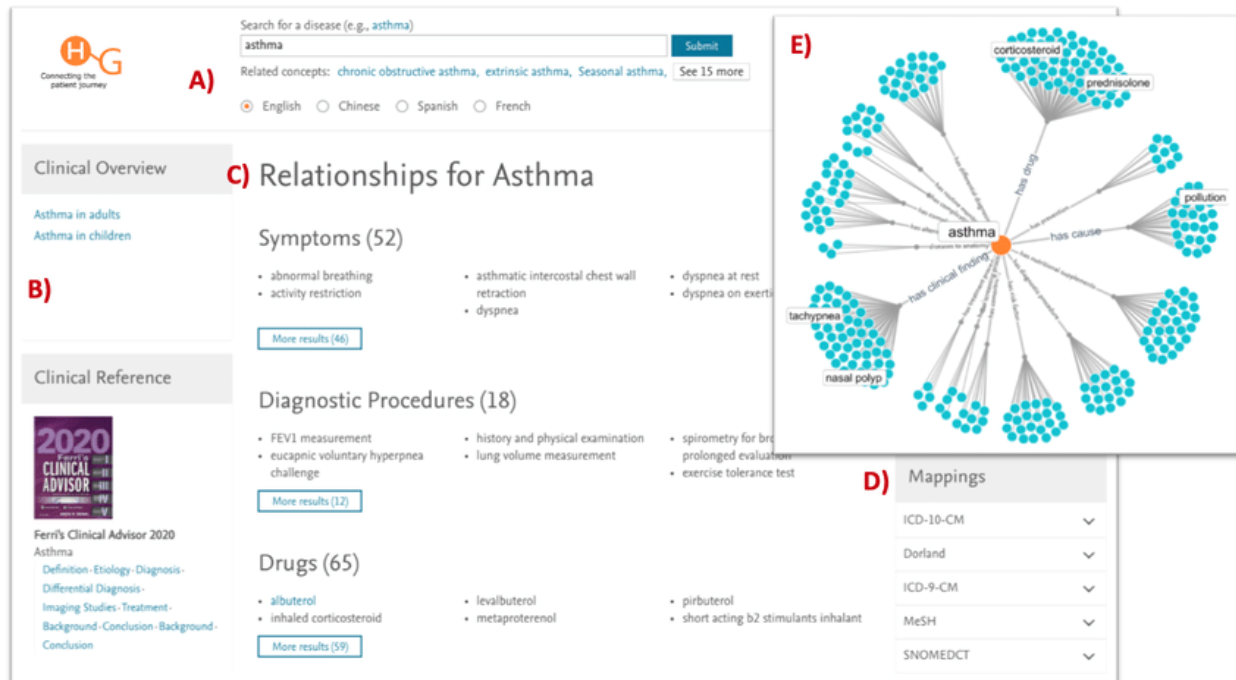
UX Design

Overview: The user experience of the knowledge graph will be designed to be intuitive and user-friendly, with a focus on accessibility and ease of use.

Key Features: The knowledge graph will include features like search, filtering, and visualization tools to help patients understand their healthcare data.

H-Graph: The Elsevier Health Knowledge Graph

High-level API

Overview: The knowledge graph will be accessible via a REST API, allowing external applications to interact with the data.

Key Features: The API will support basic CRUD operations, as well as advanced querying capabilities using SPARQL.

Libraries and models

Overview: The knowledge graph will be built using a combination of existing healthcare ontologies and custom-built models to represent patient data.

Key Features: The ontologies and models used will be chosen based on their compatibility with the semantic web approach and their ability to accurately represent healthcare data.

Libraries: The major APIs to be used are wikidediaapi, pandas, concurrent, tqdm, spacy, requests, neptune, networkx, matplotlib etc.

Conclusion

Overview: This SRS document outlines the high-level requirements for the development of a healthcare knowledge graph for patients.

Next Steps: The next steps will be to gather more detailed requirements and begin the development of the knowledge graph.