1. NetIDs of all team members:

| **NetID** |  | **Name** | **Email** | **Role** |
| --- | --- | --- | --- | --- |
| connorb8 |  | Connor Brown | connorb8@illinois.edu |  |
| frankns2 |  | Frank Salamone | frankns2@illinois.edu | Captain |
| jt38 |  | Jonathan Trusheim | jt38@illinois.edu |  |
| raylb2 |  | Ray Butler | raylb2@illinois.edu |  |
| xhuang84 |  | Xinlei Huang | xhuang84@illinois.edu |  |

2. Topic overview

a. Our free topic is a medical chart search engine.

b. Medical professionals must select, interpret, and act on the information contained in a large number of documents for each clinical interaction. When seeing a patient for the first time, a clinician may need access to only a small subset of the documents in the patient’s EMR chart. The makeup of this subset of documents would depend on the type of clinician and type of visit. An endocrinologist seeing a hypothyroid patient for the first time may want to see all CT scans, lab tests, and clinical notes related to “thyroid disease”. Finding relevant documents in a chart is particularly difficult because there are many documents related to billing, compliance, and other non-clinically relevant information contained in the chart.

c. Our team will create a search engine to aid in this document retrieval problem. As the use of real medical data is problematic, we will create a dataset consisting of several hundred artificial charts. Our system will be designed to be scalable so that it may be applied to a larger dataset in a real world application. These charts will be stored in a data storage system such as GFS. These charts will be presented to the user of our system via a simple interface organized by patients.

One of the patients from the patient list can be selected using the interface. The user may look in each patient record at the various documents. In addition to being able to select individual documents, the user will be presented a search bar. Entering a query into the search bar will retrieve relevant documents from the patient’s chart.

d. Rapid discovery of relevant documents will aid in patient care by decreasing the paperwork overhead for clinicians. Using a search engine, it would be much more likely that all relevant documents for a particular clinical encounter will be found. This may decrease medical errors. Many EMR systems do not have search engine functionality.

e. We will make an inverted index of documents in each patient’s chart, and then use standard search engine techniques to find relevant documents.

f. We will index all medical records with ElasticSearch, build a backend search engine with Python, and front-end interface with modern frameworks like ReactJS or Angular. Everything will be dockerized for easy setup. Patient charts will be constructed using web scraped data.

g. We will present a toy example of an EMR with search capability. Our search engine will rapidly and precisely find relevant documents based on the clinician’s query.

h. We will use explicit evaluation of search results by a clinician. This feedback will hopefully increase search relevance.

3. Language used: Python, modern framework

4. Time Breakdown:

Chart and database construction 20 hrs

Search engine construction 30 hrs

User interface 30 hrs

Evaluation of system 10 hrs

Documentation and presentation 10 hrs