Development Plan
Software Engineering

Team 14, Team Name
Aamina Hussain
David Morontini
Anika Peer
Deep Raj
Alan Scott

Table 1: Revision History

Date	Developer(s)	Change
2023-09-22	David	Added first draft for tech and coding standard
2023-09-23	Deep	Added first draft for proof of concept plan and project schedule
Date3	Name(s)	Description of changes
...	...	...

[Put your introductory blurb here. —SS]

1 Team Meeting Plan

2 Team Communication Plan

3 Team Member Roles

4 Workflow Plan

- How will you be using git, including branches, pull request, etc.?
- How will you be managing issues, including template issues, issue classificaiton, etc.?

5 Proof of Concept Demonstration Plan

The main risks for the success of this project are; none of the members of the team have deployed a web app from scratch before, and that the data provided by the clinical is often in natural language. Since none of the team members have deployed a web app from scratch this may be a risk as we are not certain of all the steps which must be taken to publish a web app. The data provided by the clinical trials being in natural language also is a risk as we will have to research and use a natural language processor to extract the relevant data points.

Therfore to demonstate that we can overcome these risks our proof of concept demonstations will have to both be deployed and able to be ascessed from a device that is not running the source code, and the application must be able to extract the requirements for the clinical trial from the data source. Also the application should be able to display the retreived requiremnts in a GUI of some fashion.

6 Technology

As a result of our system being a pretty complex full-stack application, it will be built using several different tools and technologies, across both the frontend and the backend.

Frontend

- Programming language - Typescript
- Unit testing framework - Jest
- Frontend web framework - React

Backend

- Programming language - Python
- Linter - Flake8, Mypy, autoflake
- Unit testing framework - pytest
- Backend web framework(s) - Django, FastAPI

In addition to the technologies mentioned above, we will be using a few tools that will help us develop our app more efficiently and effectively. First, we will containerize our app by using docker, which will make it easier to test and deploy our application (also use of docker desktop for running/testing our application locally). Furthermore, for our continous integration pipeline, we will be using

github actions to run automated tests and linting for both the frontend and backend services/code, and to build our docker images. Finally, we plan to use firebase as our cloud environment, which is where our application will actually run.

There are also a few tools/technologies that still need to be decided upon. We will need to have some database to store client information. It is likely that this will be a relational database, and will be one of Postgres, MySQL, or SQLite. Additionally, it is possible that we will need to use some existing library/tool for parsing trial eligibility criteria, however this will become more clear in the following weeks (i.e., an advanced parsing library, or existing NLP designed for healthcare systems).

7 Coding Standard

For all of our python code/services, we will be following the Pep8 coding standard, and we will be enforcing this with the help of the flake8 linter. We will also be using mypy, which will enforce strict typing (which is usually not present/mandatory in python), meaning all of our function parameters, return types, class variables, etc.. will need to have its type defined. Finally, we will have docstrings present in each module, to give a brief description of what the main purpose of the module is.

For the frontend code we will not be following any formal coding standard, but we will be following best practices when coding in typescript and when using React. Some of these best practices include -

- use functional components
- don't use inline-styles
- maintain a proper import structure (third-party imports first –¿ internal imports below)

8 Project Scheduling

- September 25 - Problem Statement, Proof of Concept Plan, Development Plan
- October 4 - Requirements Document
- October 20 - Hazard Analysis
- November 3 - Verification and Validation Plan
- November 13-24 - Proof of Concept Demo

- January 17 - Design Documentation
- February 5-16 - Revision 0 Presentation
- March 6 - Verification and Validation Report
- March 18-29 - Final Demonstation
- April 4 - Final Documentation
- April 9 - Capstone Expo