

Karthik Palaniappan (karth295@cs) and Keith Yeung (kyeungk@cs)

## Project Proposal: iDoc

### Vision

We envision creating a simple application to house patient medical information. Doctors will be able to view and update basic medical information efficiently, and patients will be able view their medical history online. These online records are called Electronic Health Records (EHRs).

A major problem for healthcare providers is working with a large volume of files. Doctors need to store and search for possibly hundreds of patients' files. When a patient is referred to a specialist or changes doctors, the hospital must fax the patient's entire file. Even still, specialists often do redundant or unnecessary tests. This is unnecessary work. We envision a system where there is only one copy of the patient's data, and each of the patient's current doctors can update this information.

At the same time, without EHRs, there is no way for clients to easily access their medical information. They must call their healthcare providers and request a copy. We envision a system in which clients can easily login to a website and view their entire medical history. In addition, this will reduce the need for patients to fill in redundant forms about their basic information (e.g. address, telephone number, insurance, etc.) and medical history at each doctor or specialist.

There are currently several systems that manage EHRs—GE Healthcare, eClinicalWorks, Aprima, etc. However, these systems are not widely adopted because they are complex and difficult to use. Doctors reason that the productivity *decrease* from having to navigate the UI—even after spending the time to learn it—is a net loss for them. We envision a simple, easy-to-use interface that will be as efficient as writing notes and updating information on paper.



An example of a complicated, confusing EHR

## Software Architecture

We plan to make this a web application so it is easily accessible from a variety of devices (e.g. computer, tablet, mobile, etc). For the backend, we plan to use a SQL database such as MySQL to store the patient data. The backend will likely be in JavaScript using a framework like JavascriptMVC to more easily communicate with the client.

A large portion of the project will be the security apparatus, as the data needs to be very secure. Only patients and their doctors who have been granted access should be able to view and modify their data. We plan to encrypt the data in the database on both the backend, and the front-end cache with AES.

For the front-end, we plan to make simple UI that can be easily accessible from a variety of devices—computer, tablet, mobile, etc. This will be a website, so we will use HTML5/CSS and JavaScript.

To integrate the two, the “controller” will need to request the model for the client information, and send updates to the model as well. It will also perform the authorization and permission checks.

## Challenges and Risks

A major challenge will be researching the information doctors record on paper forms so we can accurately create a schema for the database. Since different doctors likely record different information, we will have to talk to different doctors (e.g. doctors at the UW Medical Center) to understand how to design our system.

The doctors will also give us an idea of what basic information is “important” in a standard visit, and how they would like to see the information represented. Other EHR systems often show too much patient information on a single screen, which makes them confusing and crowded. We plan to show them our sketches and prototypes.

The other major problem will be privacy. Both patients and doctors need to be confident that their data is stored securely. As mentioned before, beyond a secure login/authorization system, we will encrypt data both on the backend and frontend as an added security measure. We will need to be careful in complying both with US privacy law with regards to health information, as well as laws regarding EHRs. Beyond that, we must talk to both doctors and patients to further understand their expectations of security before using they will use the system.

This will be an ambitious, but doable product in 8-10 weeks. In order keep within the allotted time frame, we will need to constrain the scope of the project to our vision above.