

EMR Skills Development

Taylor Colby, Joe Marcinuk, Cameron Oeze, Alon Varshaver, Department of Computer Science and Information Technology, University of New Hampshire

Project Sponsor
Clinical Assistant Professor Clarissa Michalak



Introduction

An EMR is an electronic record that shows details from a patient's file such as their previous medical history, medication, general information, as well as being a place for nurses to write notes on their patients

Currently, it is not feasible for students to train with these EMRs in a classroom setting due to either cost of the software or lack of availability.

The goal of this capstone is to develop a EMR web application that replicates current EMR software for students to practice on before encountering them in the workplace, while also providing a way for instructors to assess how well a student uses the application.

Requirements

- Students can select a patient from a list of patients
- Students can access different sections of a patient's file from clickable tabs
- Students can insert a note into a patient's file and view notes from other students
- Professors can create and delete patients
- Web application looks and acts similar to other EMRs

Design/Implementation

Front-end:

- A mock medical records web-application that students can log into and learn how to navigate real-world EMR software.
- The app uses HTML, CSS, and React JS to mimic other EMRs. We are mostly basing ours on a major EMR called "Epic".

Backend:

- To service our web application, we are containerizing our API application in a Docker container that will run on the University's business servers.
- The API is implemented in using a Python Library called "Flask". This is the primary "glue" between our database, and the web app itself.
- We are using the MongoDB database as our primary data service, as it's document-based data storage system is ideal for storing patient data in compartmentalized and searchable ways.
- Certain constants and formatting data are contained within configuration documents that are also stored in the MongoDB for use by the web application.

Diagrams

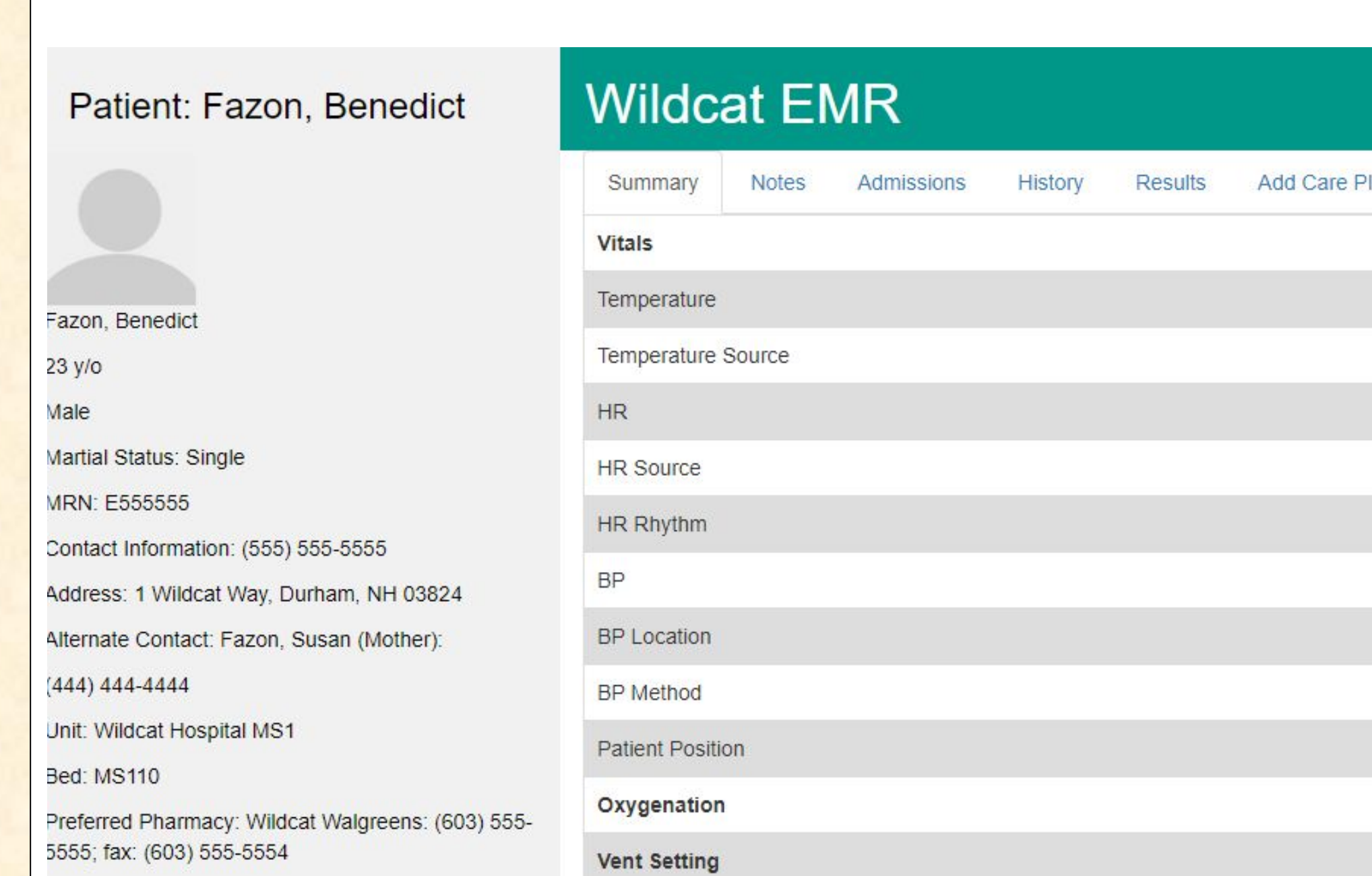


Figure 1: Current iteration of the EMR Web Application

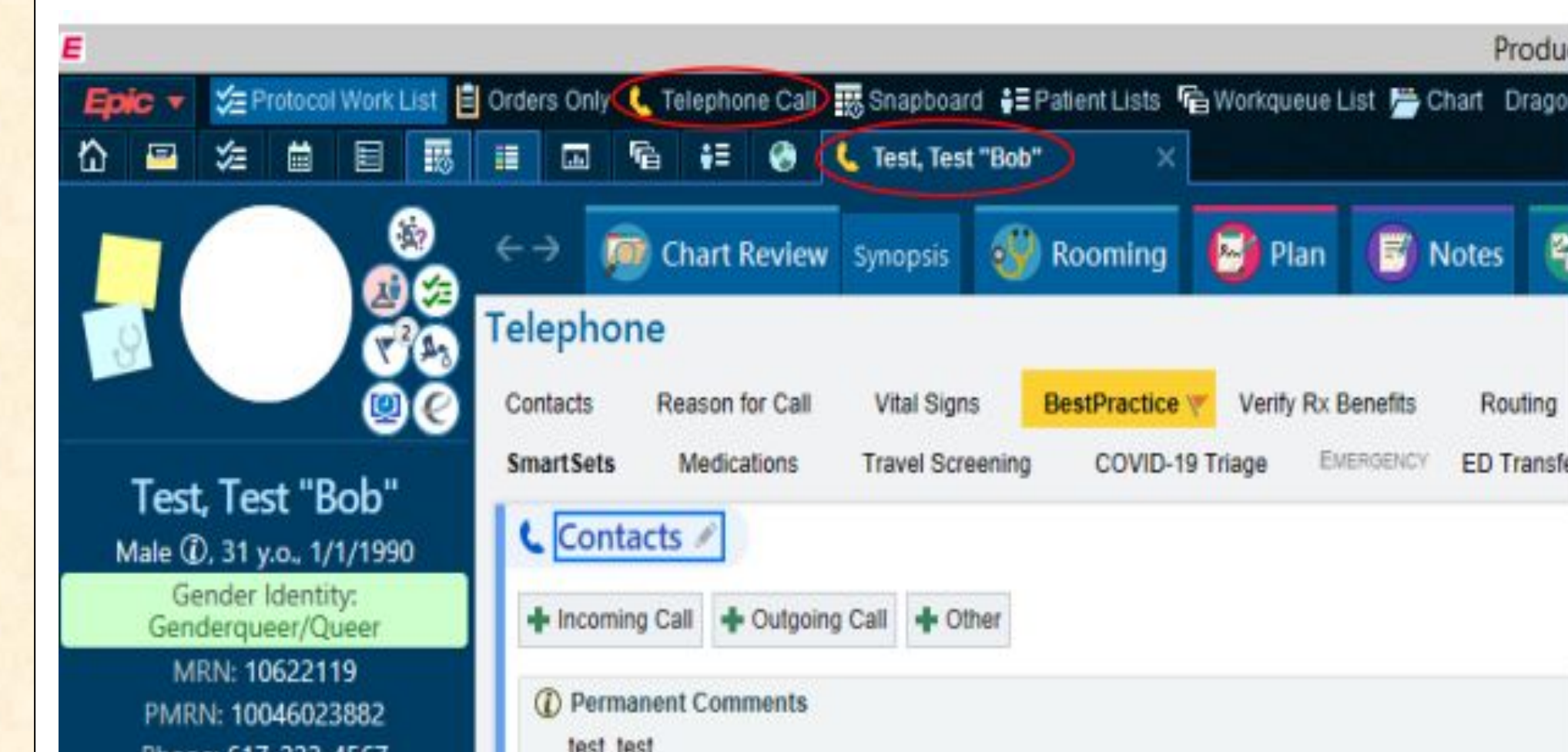


Figure 2: Example image of Epic EMR

Testing

Front-end:

- GUI testing, patient tab functionality testing, and usability testing of different features such as note creation.

Back-end:

- On the backend, we are doing unit testing on API request, database functionality for patient and patient note creation, as well as unit tests for building various patient result tables. Unit tests are also used for validating data documents and config files

Diagrams (cont.)

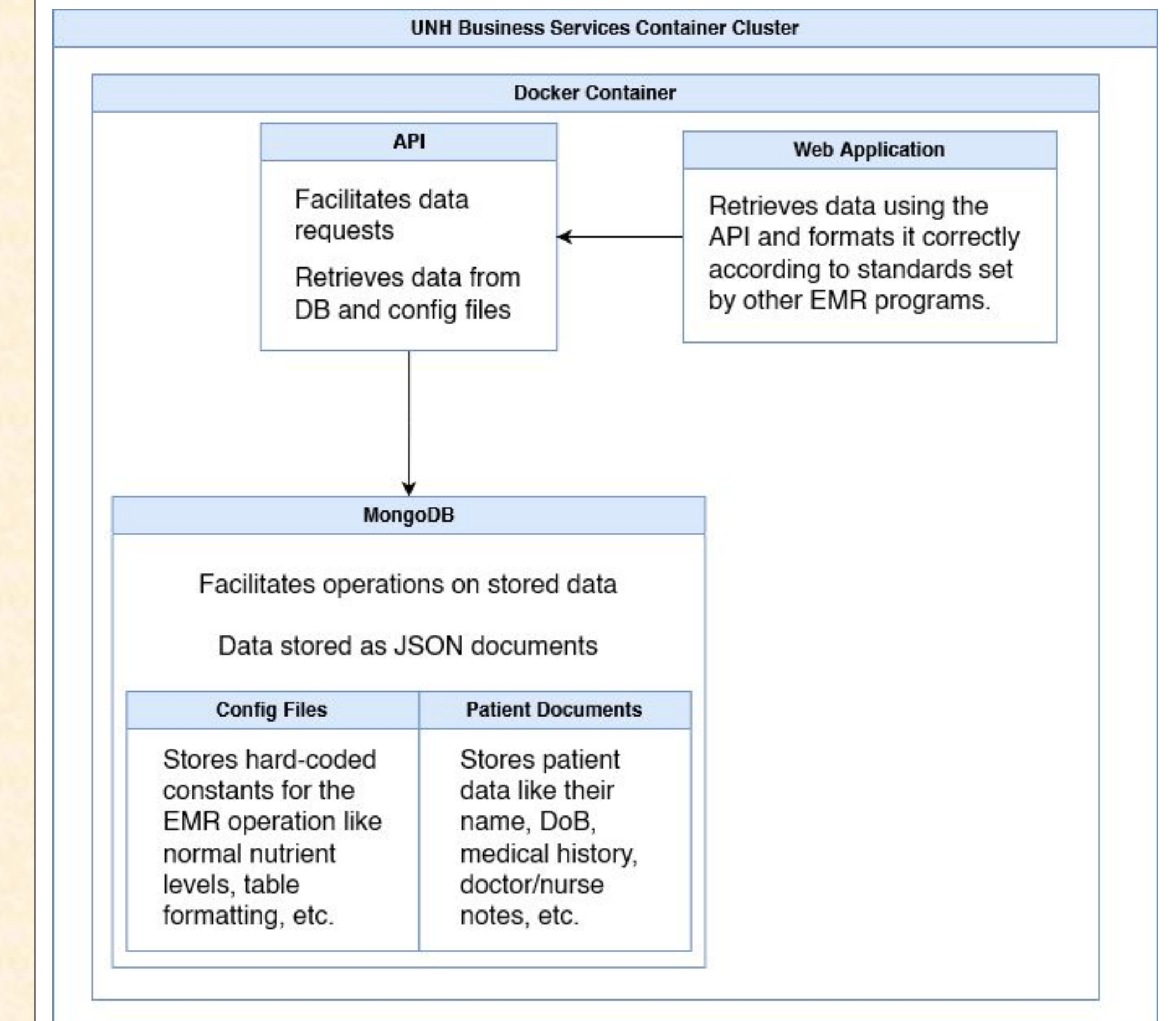


Figure 3: Design diagram of system



Results

Students can log in to the EMR site and view multiple tabs. They can switch between patients and add care plans and notes. Professors can log in and add and or delete patients. Professors can assess students based on their notes as well as a embedded quiz

Next Steps

- Add more tabs to create a more well rounded EMR.
- Add efficiency monitoring for professor users to grade students