All-Inclusive Health and Present Illness Report

Ron Rounsifer, Daniel Schroeder, Eric Schoenborn

School of Computing and Information Systems, Grand Valley State University

Overview

When a patient visits a health provider, whether it be from a hospital, urgent care, or a family doctor, the chief concern (CC) is the first statement made to provider, which should describe the health issues/concerns that caused the patient to seek medical attention. From this CC, the working doctor is able to generate a short list of potential reasons behind the illness, which is known as the differential diagnosis (DDx). Out of these reasons, one will be the most likely reason for the illness at hand, which is called the favored diagnostic hypothesis (FDH). One heavily used method of FDH generation is by applying a "All-Inclusive History of Present Illness" process. This process is a problem-solving tool that physicians are able to use to efficiently make the most likely diagnosis of a patient's CC. Once a FDH has been established the doctor can then begin to run calculated tests on the patient while initiating the proper treatment. Of course, it is important that this FDH be correct in order for the patient to begin their recovery process as soon as possible. The focus of this project is to generate a web application that will bundle with the book that Dr. John Sheagren and Dr. Adam Killian are writing for students of medicine. The focus of the book being about how to quickly, and accurately, generate an AIHPI.

Backend

1 Web Framework - Django

To have a working application, the front end must be able to send requests and receive responses from the code base that is hosted on the server. To make this work, the use of an application programming interface (API) is mandatory, which Django allows us to build with relative ease.

2 Hosting - Heroku

In order to publish the web app to the actual internet for others to make use of it, we chose to use the cloud platform Heroku. This allows the process of deploying, configuring, scaling, tuning, and managing applications to be made as simple as possible. Granting us the opportunity to focus more on the application itself, instead of tweaking the hosting parameters.

3 Data Storage - MySQL

Out of the box, Django offers support for the MySQL database management system. This in tandem with our prior experience using MySQL causes this to be the most attractive option to solve our data storage problems.

4 Comparing Reports - Cosine Similarity

Majority of the score will be based on the comparison an "ideal" report with the student submitted report, the cosine similarity (see below) of the two reports will be calculated. This means that each report will be turned into a vector of numbers, with each number representing a specific word. This will result in an answer between -1 and 1, meaning exact opposites and exactly the same vectors respectively.

$$similarity = cos(\theta) = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

User Interface

When the player starts the game they will be prompted with a login screen. Once successfully logged in, the player will be brought to the game settings page. The game settings page has three settings for the player to adjust to their preferences.

The first settings is the year setting. This setting will allow the user to select what year they are in the program (one, two, or three). This setting will dictate what question the player will be asked.

The second setting with the difficulty setting. Since it is important in the field to be able to come up with and present an AIHPI in a short amount of time, the difficulty setting adjust the time given for each question. Players are penalized for using too much time even if a correct answer is given.

The final setting is how many questions the player would like to answer in a single game. The options being five, ten, or twenty. Once the player clicks start, they will brought to the actual game. the game itself is divided into tow basic parts.

The data section is the first part and contains the patient health form. The player can spend as much time was they wish to review this part.

The question will begin hidden from the player and will only appear when they click the ready button. Once the player will be prompted with one of four types of questions (true/false, fill in the blank, multiple choice, or short answer). However, once the question is reviled a timer will start counting down and the player will be penalized if they take too long. There are two more items on the game page to be mentioned. First, a quit button in the top right. Second, a progress meter that represents each question as an empty box and completed questions as a box with a check mark inside. At the end of the game, after answering all questions the player will receive a popup that displays their score. The popup will show their score in both percentage form and in star icons. Each full star is worth 20 percent and each half worth 10.

The Web App

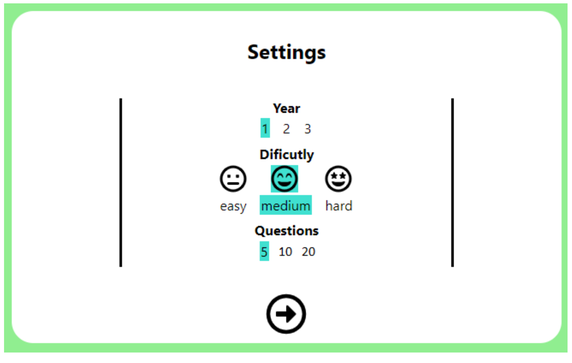

Fig. 1: Settings page.

Web App Continued

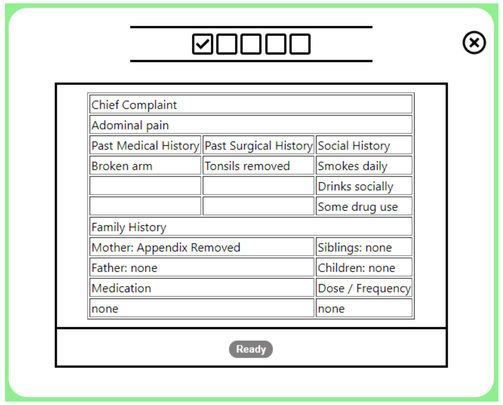

Fig. 2: Game play with question hidden.

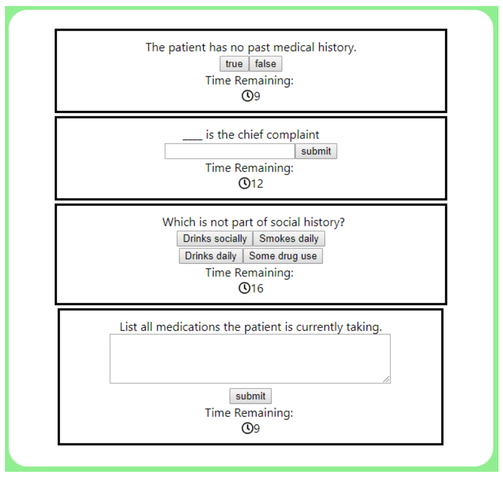

Fig. 3: Four types of questions with timer.

Fig. 4: End of game pop up with score.