

ALL-INCLUSIVE HEALTH AND PRESENT ILLNESS REPORT

Ron Rounsifer and Daniel Schroeder and Eric Schoenborn
School of Computing and Information Systems, Grand Valley State University

What we made

The end product is a web game created to help medical students reinforce the teaching of the book that Dr. John Sheagren and Dr. Adam Killian are writing.

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 of them 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 more of 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 efficiently as possible. The focus of this capstone project is to provide software that will bundle with the book that Dr. John Sheagren and Dr. Adam Killian are writing for students of medicine about how to quickly, and accurately, generate an AIHPI that will help the patients to receive the best care possible.

Backend

Our back end was created using Python3 and Django. Our database is a SQLite database and has multiple tables. For our many to many relationships we have a new table which consist of each entities ID. This is to reduce the size of storage in the long run. We also used Django’s built in user authentication to verify users login attempts and verify valid email addresses.

When we compare the the users solution to our solution we have we use a version of cosine similarity. We take both strings and change them to vectors by each word. We then remove filler words and compare the vectors. The result we get from the cosine similarity is then reported as the score for the given problem.

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 questions the player will be asked.

The second setting is 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 the data provided.

The question will begin hidden from the player and will only appear when they click the ready button. Once the player begins they will be prompted with one of four types of questions (true/false, fill in the blank, multiple choice, or short answer). This will also start a timer that will begin to count down and penalize the player 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 that allows the user to quit their session at any point. 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 pop-up 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

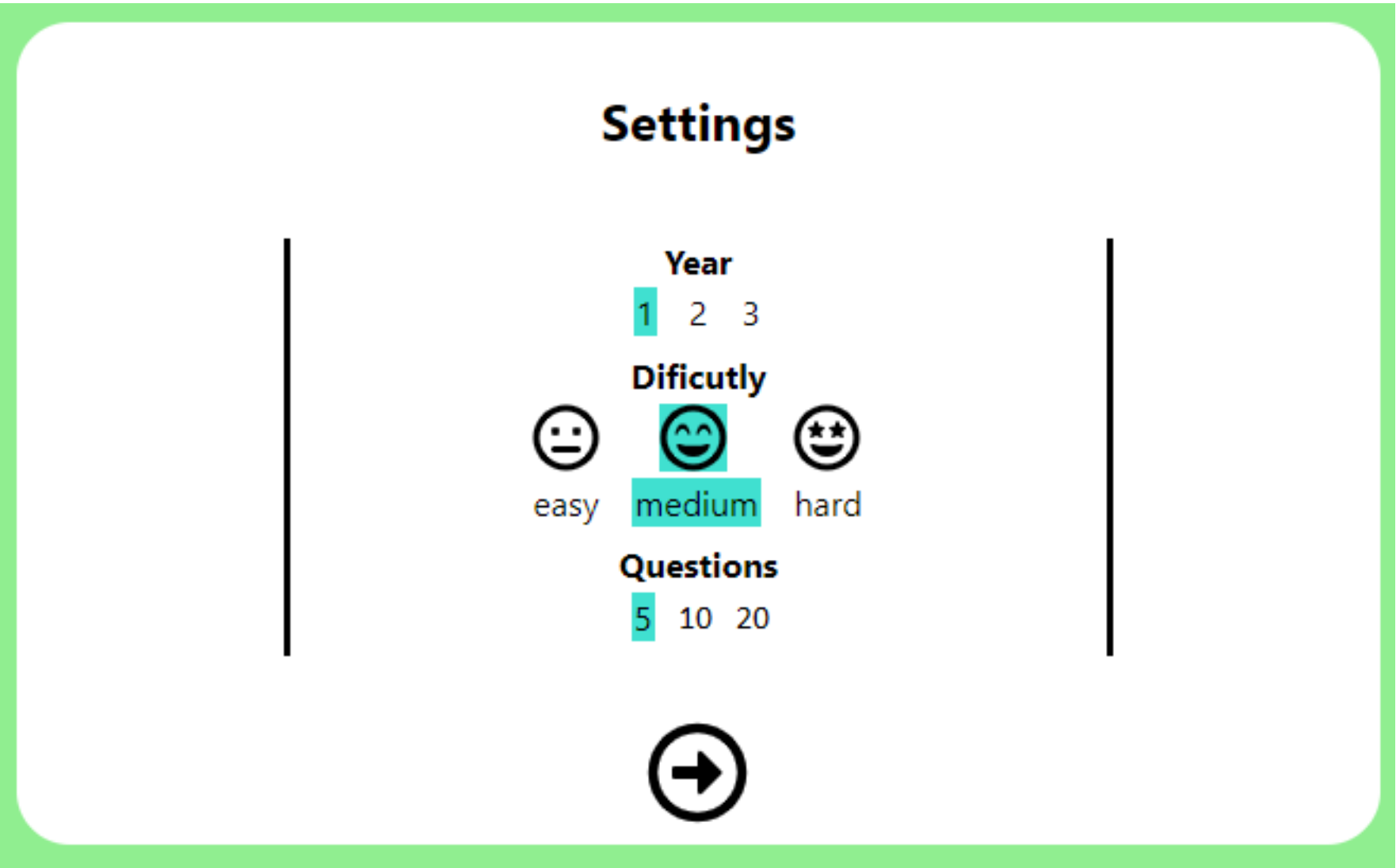


Fig. 1: Settings page.

Web App Continued

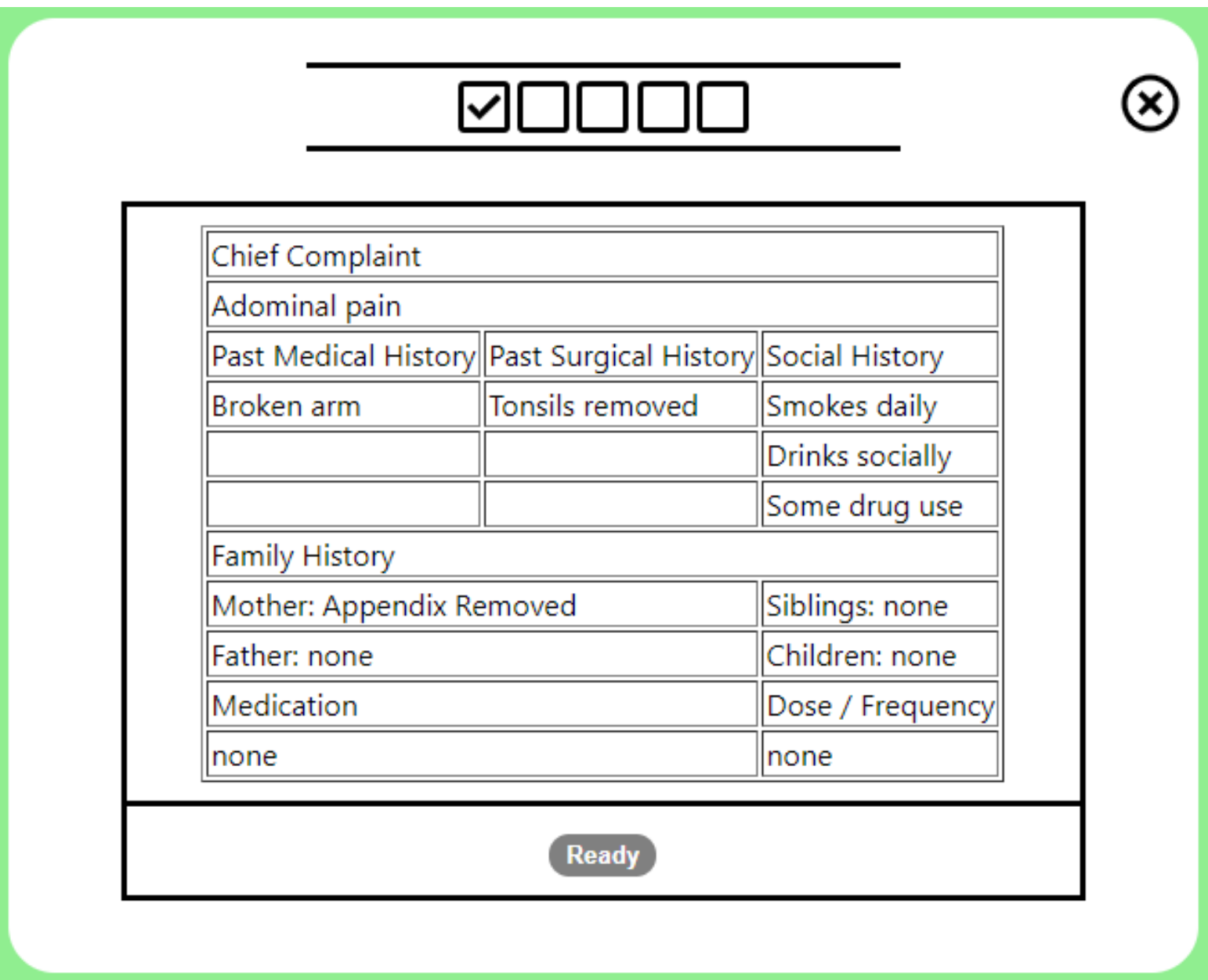


Fig. 2: Game play with question hidden.

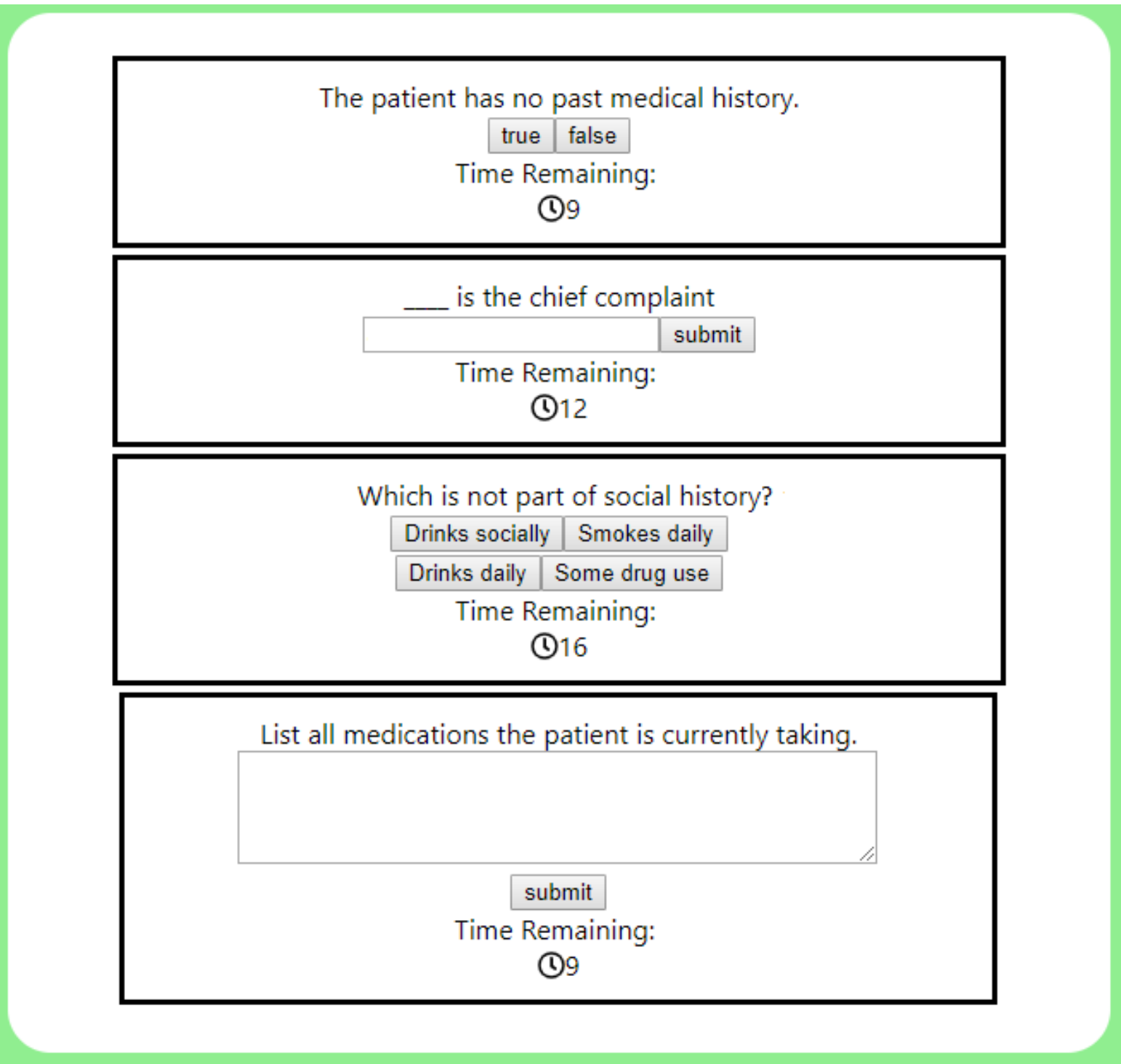


Fig. 3: Four types of questions with timer.



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