CSC-271 Final Project: Database Concept App

Tu Nguyen

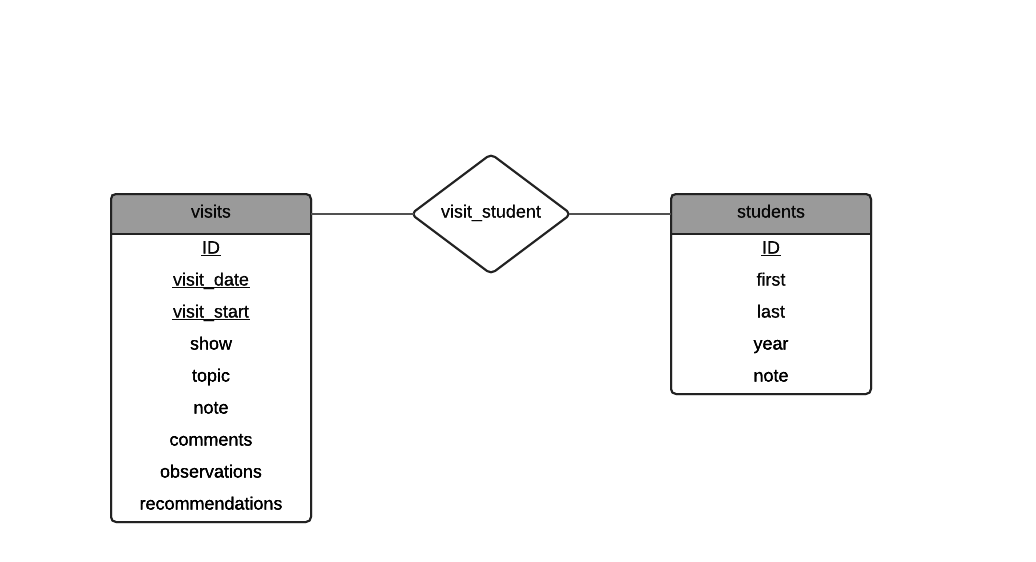
Ngoc Tran

Ashton Faramelli

Thad Castro

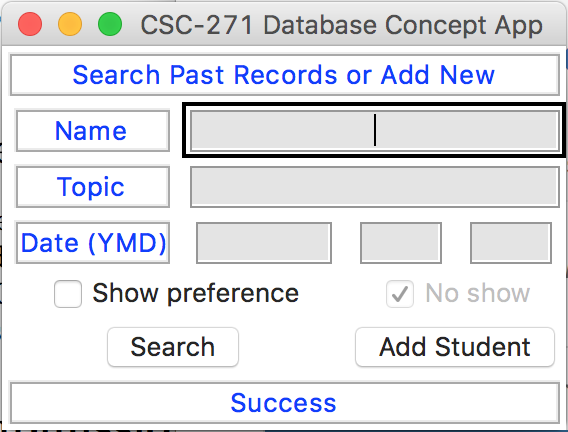
Jonah Woods

We create the Database Concept App using both Python and SQLite code in a text editor app Atom. The file contains all of our code and a test database that was provided by Dr. Koppelmann. The following is the E-R diagram for our project, which includes the two relations: Visits and Students. The relations are combined with the relationship set visit\_student.



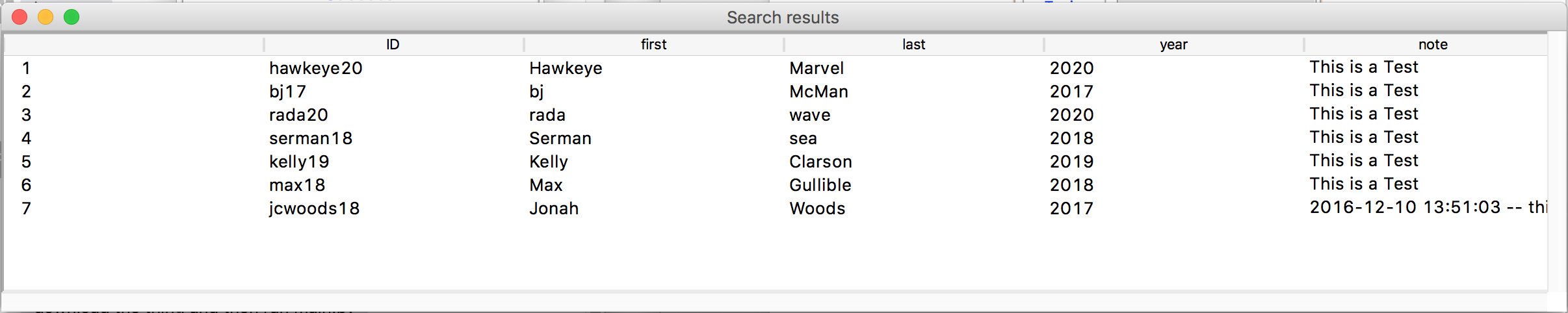
Each member was able to access the files and work in conjunction with each other using the version control system Git and its repository GitHub. We could test the program originally using IDLE, a python development environment. We then created the actual app that can be opened without working in python code at all. With these tools, we were able to create the Database Concept App.

The application is a simple, offline set of graphical user interfaces that links each student to his visits. To open the app, simply click on the icon. The first page that appears looks as follows:

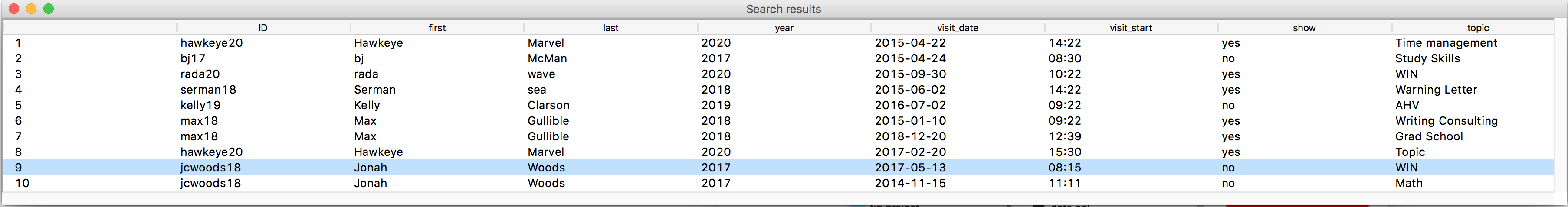


As stated at the top, this page is used to either search past records or add new ones. The fields are mostly self-explanatory. The user can type in a name and search the student relation. By typing in a topic and/or date, he can search the visits relation for specific visits. All three fields can be searched individually, or in any combination together. In addition, the user can click Show preference, which will display students based on if they showed up to a visit or not. One hidden feature is a wildcard input in topic. The asterisk (\*) can be input into the topic field and all visits will be shown when search button is clicked. At the bottom of the page we have a status bar that displays errors in the system.

Following a search of the name field, the table below opens.



This table displays the results of the query. Double-clicking any name will bring up the selected student’s page.

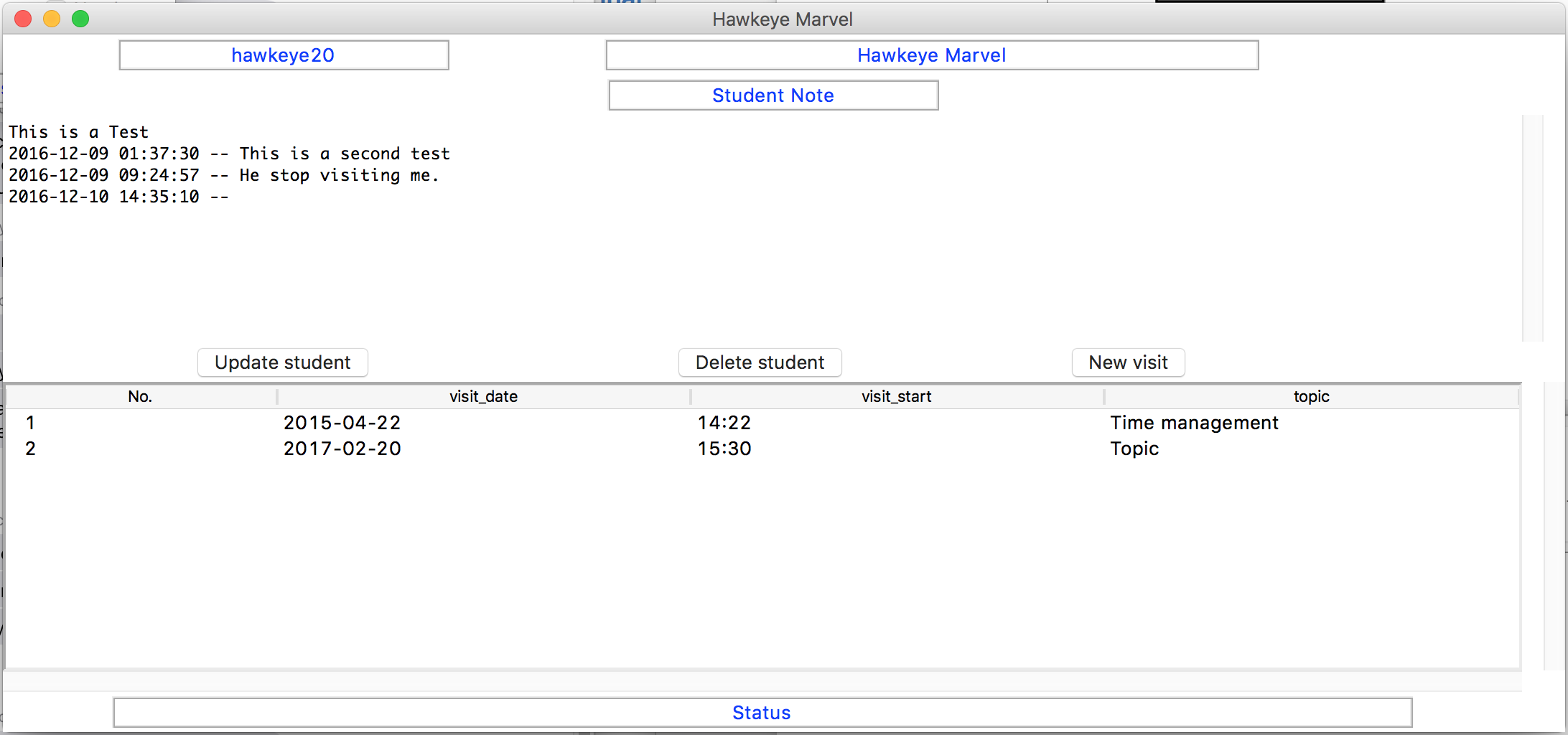
The results of a topic or date query differ only slightly.

The user sees a few more fields to the right of the student’s name. These fields display information directly linked to a specific visit. We placed a hidden feature in this table as well. If the user double clicks the first or last name of a student, that student’s page is opened. Double clicking any other field in the table will bring up that specific visit.

To reemphasize, the user can reach a student’s page by one of the following:  
1) In a query of the name field – double click any result

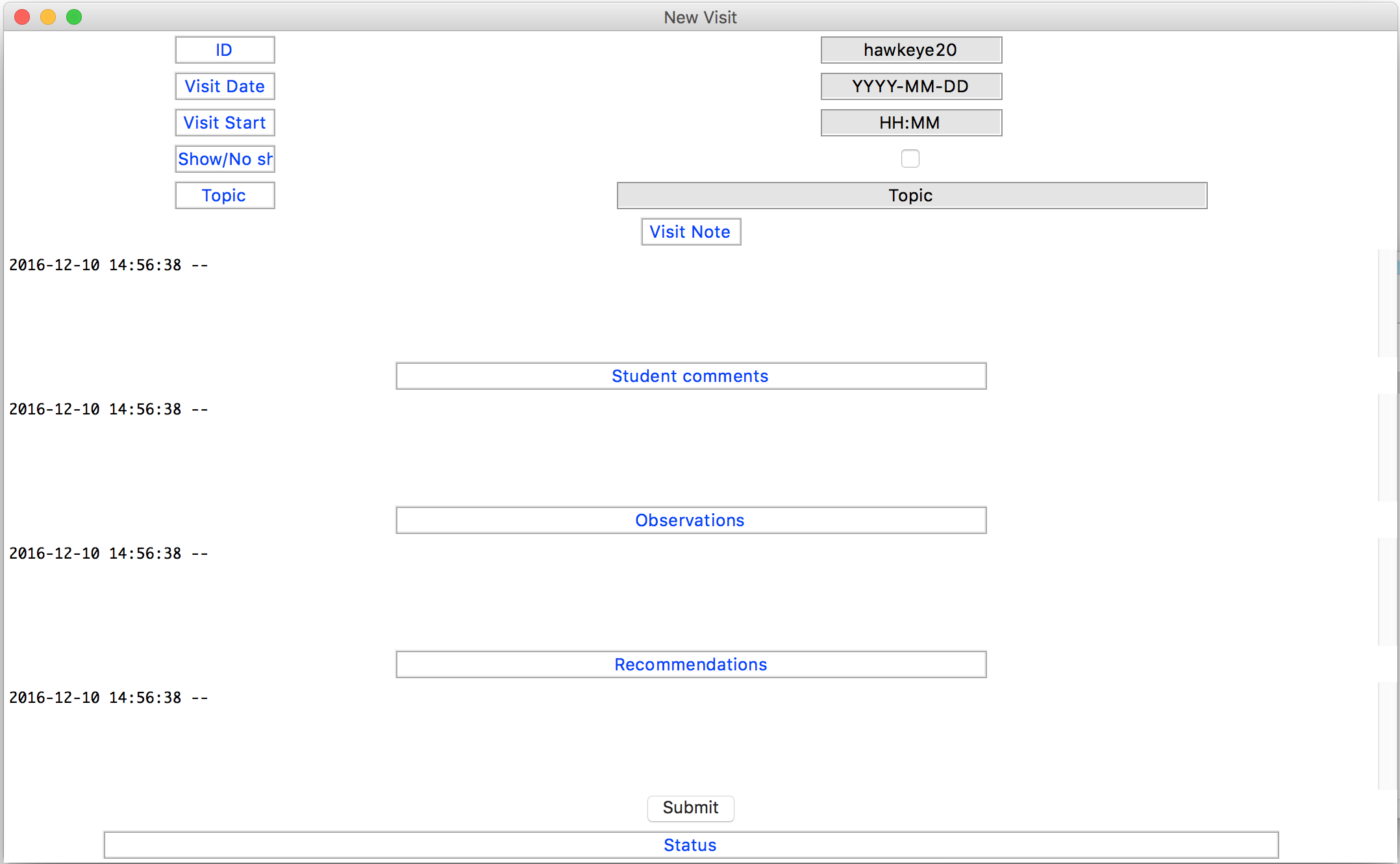
2) In a query of the topic/date field(s) – double click in the first or last name column

An example of a student’s page is shown below.



Once again, we tried to make the pages as self-explanatory as possible. The top displays key information for the student. Below that is an editable note. Three buttons sit on top of the table displaying each individual visit. These visits can be accessed and edited by double clicking their row. (Deleting a visit will be discussed later). The update student button updates the student note directly above it. The delete student will delete the student and cascade delete all visits connected to this student. The new visit will create a new visit for this student.

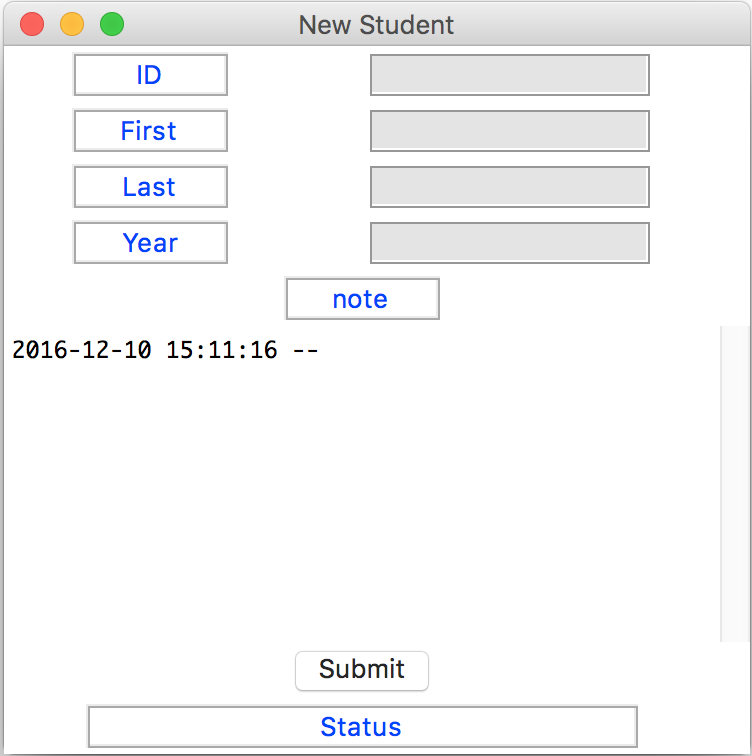
The following is an example of a new visit for the student Hawkeye Marvel.



Again, the top displays the key information regarding the specific visit. The left hand side information acts as labels for the editable fields on the right hand side. On the right, the student’s ID, the visit date, the start time of the visit and the topic are all editable. We also provided a checkbox that indicates whether or not the student showed up to the visit or not. Below that we have the Visit Note section where general notes about the visit can be entered. The three sections below that behave in the same manner as the Visit Note section, where information related to that section can be entered in. Clicking the submit button on the bottom will save the newly created visit into the database.

This form is the same as the form that would appear if a visit was double clicked from the specific student’s page. The only difference is that instead of the submit button, there are delete visit and update visit buttons.

Once the New Visit form is submitted, the app redirects back to the first page. From here, if you click the Add Student button, the following page will appear.



Similar to the New Visit page, starting at the top, the left hand side serves as labels for the editable fields on the right hand side of the page. From here, all of the information for a new student can be entered, including his ID, first and last name, and his school year. Also, we included a section where you can add any general notes about the new student for future reference. Clicking the submit button will save the newly created student into the database and redirect the user back to the first page of the application.

Problems in the program

If you do not exit a window before trying to search again, the program will crash. Random crashes (few).

Instead of a standard error message telling the user what specifically is wrong with the form, such as an empty required field, appearing in the Status box, it instead shows the message “invalid literal for int() with base 10: “ “ (Can be fixed with better error handling)

What we learned in the project

Tu: It was hard developing the GUI since I am not familiar with it. I also learnt how to interact with sqlite database using Python.

Ashton: I learned how the SQL and another language (Python in this case) talk to each other as well as how the app was actually created. Also, I learned how the different ideas of our group interacted with one another and how to utilize the advantages of different people’s skills.

Jonah: I learned a new Python interface called Treeview. As everyone else has mentioned, I also learned how to use SQLite data in Python code.

Ngoc: I learned how to use the Python debugger in addition to how to design the GUI.

Thad: I learned how to utilize Python in conjunction with SQLite3 as well as how to actually launch a local application from scratch.

What we found difficult

Designing the GUI.

Things to do in future

We wanted to improve the way the program looks and feels. Add the YYYY-MM-DD (like in a new visit window) in the opening page. When you want to add a new student named Jonah Woods, you should be able to input that on the first page and then that info is put into the add student page. Add am/pm on time of visit. When you hit submit on new visit, you should be able to go back to the specific student page instead of having to exit. Should be able to add to pc. Be able to use enter button in addition to double click. Better error handling.