Project Proposal

Intelligent Review System v2.1

Group Members and Skills

- Jared Raiola - 3rd year CS major
  - Java, Python, JavaScript, html, css, C++, SQL, Scala, Swift
- Sukhmai Kapur - 2nd year CS major
  - Java, React/JavaScript, Python, html, css
- Michael Keohane - 2nd year CS major
  - Java, javascript, html, css, react, sql, python
- Prem Sakala - 3rd year CS major
  - Java, SQL, Python, C, Angular/JavaScript

Problem and Solution

Problem

Visualization tools help users quickly gain an understanding of a data set. The Intelligent Review System gives these tools to teachers and TAs to help them understand the students in their class. As of now, the Intelligent Review System (IRS), connects the database full of student information to the front-end visualizations through a REST API. The front end displays graphs for this analysed data, giving the user access to different types of analysis and filters. In previous semesters there was less focus on the review of aspect of IRS, and we would like to put more focus on that area located in the Student tab to add an interactive component. Also we would like to focus on the Administrator tab and give information about our database and cached tables. Once this is hosted on a server, backend information is hard to inquire about easily. Displaying this information will be really helpful.

Solution

Our main focus this semester involves adding content to our Student and Administrator tabs, along with providing more views to the Teacher tab. More detailed information is provided below:

Front End

- Improvements to our UI
  - Implement additional charts besides the current bar graph (ex. histogram, Gaussian distribution)
  - Bug fixes
  - General improvement of aesthetic to improve usability

Our student tab is interactive (but not personalized) and informs current students on past students have done.
The student table will have:

- Question prediction (difficulty bins) based on kNN bins
- The ability to see model students' data and generate review questions based off model students

Our Teacher/TA tab will have summary statistics that weigh in about the course as a whole.

In addition to last semester, it will have:

- ○ Add Gaussian Distribution graph
    - ■ Calculate and display standard deviation with mean scores as well
- ○ Use of histograms to display data

Our Administrator tab will have:

- ○ Database query speeds
    - ■ Explains if and when the app is running very slowly
- ○ Display last time the database was updated with more data
    - ■ A way for us to tell when the script that caches tables was run

Back End:

- Add question/answer endpoint, possibly cache a new table
- Add endpoints for database speeds and results
- Dynamic fetching of model students for comparison
- Endpoint to pass model student data to frontend
- Add endpoints to push IRS-ML
- Cache KNN results (takes a long time) into a table
    - ○ Link to existing script that caches

Potential Issues/Problems we might encounter

- Clear communication between front-end and back-end
- Merging the IRS-ML project into IRS.v2 might be difficult with the kNN calculation
- Switching from python visualizations to javascript visualization could cause problems with visual clarity
- Accessing questions and their answers for possible review system
- Finding "good" data for model students and be able to cache it well
- Deploying onto GT servers could have security constraints

Minimum Viable Product

Before the final presentations we would like to have the following things implemented:

- Student tab - Model Student information for students to look at
- Student tab - Ability to classify students by KNN results
- Admin tab - Show last modified timestamps and database query speeds
- Teacher tab - gaussian distribution/histograms for better view of summary statistics

Note: Our MVP is important to demonstrate our ideas and to continue an iterative agile process. We also want to show that all aspects of our ideas are meaningful and can be achieved simply with more effort and time.

Working Plan

Time Schedule	**Task**	**Responsibility(Flexible)**
Week 5-7 (Sukhmai and Prem on back end, Jared and Mike on front end)	**Milestone 1: Transfer kNN, add new Teacher Views Front End: New graphs for teacher tab, structure admin tab Back End: kNN cache, query speed and timestamp cache**	**kNN transfer (Prem, Jared) Admin Tab Structuring/ New Graphs (Sukhmai, Mike)**
Week 8-11	**Milestone 2: Model Students Front End: Structure Student Tab Back End: Cache Model Students/Endpoints**	**Front End: Jared, Mike Back End: Prem, Sukhmai**
Week 12-13	**Milestone 3: Front End: Make student tab interactive/Look into Displaying Review Questions/Answers Back End: Look into Deployment Goal: Complete MVP**	**Front End: Mike, Jared Back End: Prem, Sukhmai**
Week 14	**Milestone 4: Complete Final Presentation.**	**All Members**
Week 15	**Milestone 5: Complete documentation, wrap up the work // merge with master**	**Push final changes to github, create documentation and merge**

Implementation Tools and Resources

Project Documentation

- GitHub: https://github.gatech.edu/VIP-ITS
- Project Documentation Notebook
- Spring 2020 Github: https://github.gatech.edu/VIP-ITS/IRS-v2

REST API Information

- https://flask-restplus.readthedocs.io/en/stable/

Backend Information

- https://github.com/PyMySQL/PyMySQL#documentation
- https://dzone.com/articles/how-to-optimize-mysql-queries-for-speed-and-perfor
- https://www.freelancer.com/articles/web-development/how-to-make-your-sql-queries-faster
- https://www.infoworld.com/article/3210905/10-essential-performance-tips-for-mysql.html

Front-end Information

- https://hackernoon.com/9-best-javascript-charting-libraries-46e7f4dc34e6
- https://material-ui.com/
- https://reactjs.org/docs/getting-started.html