Project Proposal

Intelligent Review System v2.1

Group Members and Skills

- Jessica Bishop - 3rd year CS major
    - Java, Python, React/JavaScript, html, css, C, SQL
- 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. Right now, however, there is a significant lag for getting the data to the front-end and there is room for improvement in the analysis we provide. In addition, the current UI overloads the user with information and it is difficult to understand the individual points of data.

Solution

We believe that we can increase the speed of the application by querying for less data at a time and caching the data. We would like the change the web application for more targeted summary statistics (including tabs for both TAs and teachers, filtering by classes, and filtering by chapters). This would make it easier for the user to understand and would also decrease the amount of data we need to query, speeding up the application. We would also like to update the UI in order to provide a better experience to the user and to allow the user an in-depth and clear look into the data with less data to look at and a cleaner, larger graph. Some more detailed solutions are listed below:

Front-end:

- Improvements to our UI
    - Including a retractable side panel to adjust filters
    - Adding tabs for students, TAs, professors, and administrators
    - General improvement of aesthetic

Our student tab will have:

- Overview of general student performance for the course, and see performances of past semesters
- The ability to compare your performance with the rest of the class
  - Can filter the graph down to a specific assignment, or view the entire course at once
  - Box plots (like canvas does) so students can see their percentile score
- The ability to compare your performance with previous semesters

Our Teacher/TA tab will have:

- Have a graph on total time spent on questions with filters for particular periods of time or specific assignments (TA/Teacher tab)
- General filtering per section and per assignment (specific to classes that TAs are assigned to)
- Displays of average statistics (time spent per assignment, score per assignment, how long before the due date that students begin working, etc.)

Out Administrator tab will have:

- Database query speeds (broken down to see which part of the queries are taking too long)
- How often teachers and students are using this application

Backend:

- Use SQL indexing for student/question information instead of searching the whole database for a specific piece of information
- Preprocessing data to increase speed
  - Make views (not actual tables) that have the columns we need to eliminate joining tables when actually querying
  - Cache data to limit the number of queries to the database
- Profile database so we can see which queries are taking the most time

Potential Issues/ Problems we might encounter

- Clear communication between front-end and back-end
- Figuring out how to speed up our database queries and what mechanisms we are going to add to the IRS in order to increase performance
- Designing UI that is modular and supports querying multiple times on the same page

Minimum Viable Product

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

- A noticeable increase in time to receive the data from the back-end by implementing the researched solutions
- A modular, more user-friendly UI with different tabs for TAs, students, and administrators
- At least one graph for each tab that contains meaningful, easy to understand data visualization

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**
Week 5-6 **(Sukhmai and Prem on back end, Jessica and Mike on front end)**	**1. Finalize project proposal** **2. Planning/research phase** **3. Determine exact graphs and filters that we will use**	**Front End: Improve the design of front end, making it more user-friendly. Also look at potential design and graphing libraries** **Back End: Research ways to increase speed from database → API → front end**
Week 7-8	**Setup** **Milestone 1:** **Front-End: Basic design planned with UI libraries chosen. Also include a loading icon. Create the three different tabs.**	**Front End: Start fixing front end UI to include new design and graphing libraries** **Back End: Start experimenting with different queries for speed**
Week 9-10 **(might switch roles to gain better exposure)**	**Milestone 2:** **Front End: Add 1-2 graphs to the dashboard** **Back End: Have some progress with speed of queries**	**Front End: Figure out new analysis to add and start implementing new graphs** **Back End: Start coordinating queries based on the particular graphs being used**

Week 11-12	**Milestone 3: Front End: Complete the graphs and add filters for the graphs Back End: Have a significant speed increase in pulling data from queries and analysis calculations Goal: Complete MVP**	**Front End: Keep adding new analysis, figure out if there's any other front-end features we want to add Back End: Continue speeding up query calls, API, and analysis**
Week 13-14	**Milestone 4: Finish up any extraneous problems. Solve any bugs or issues. Complete Final Presentation.**	**Divide work as appropriate**
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
- Fall 2019 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