Project Design

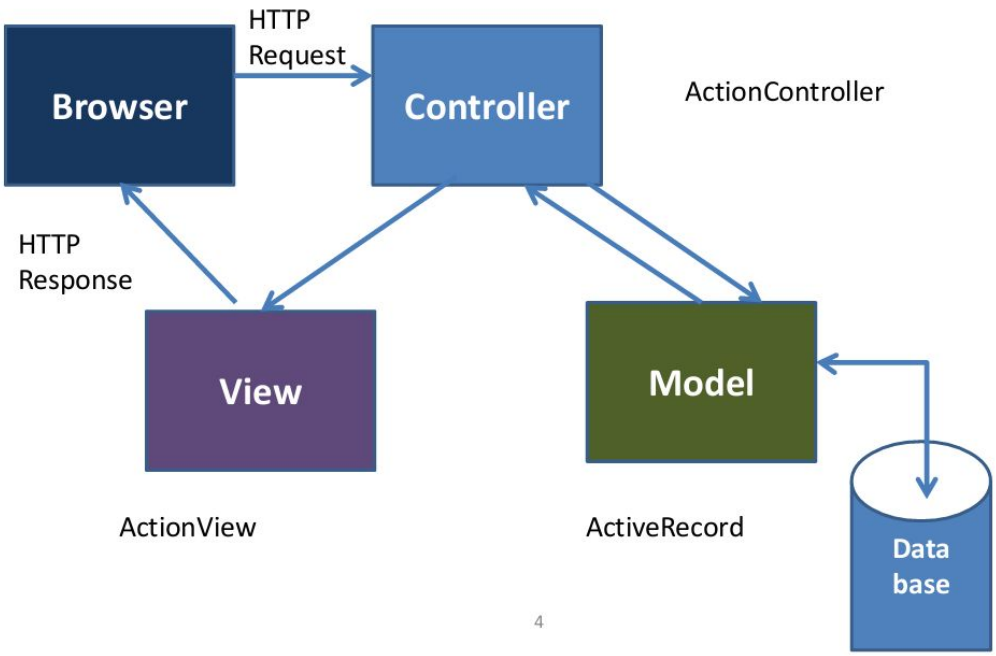
Client: United Way of Lehigh Valley

Team SQ(L)uad
Agathe Benichou, Frankie Klerer, Nick Turney, Liam Mungovan

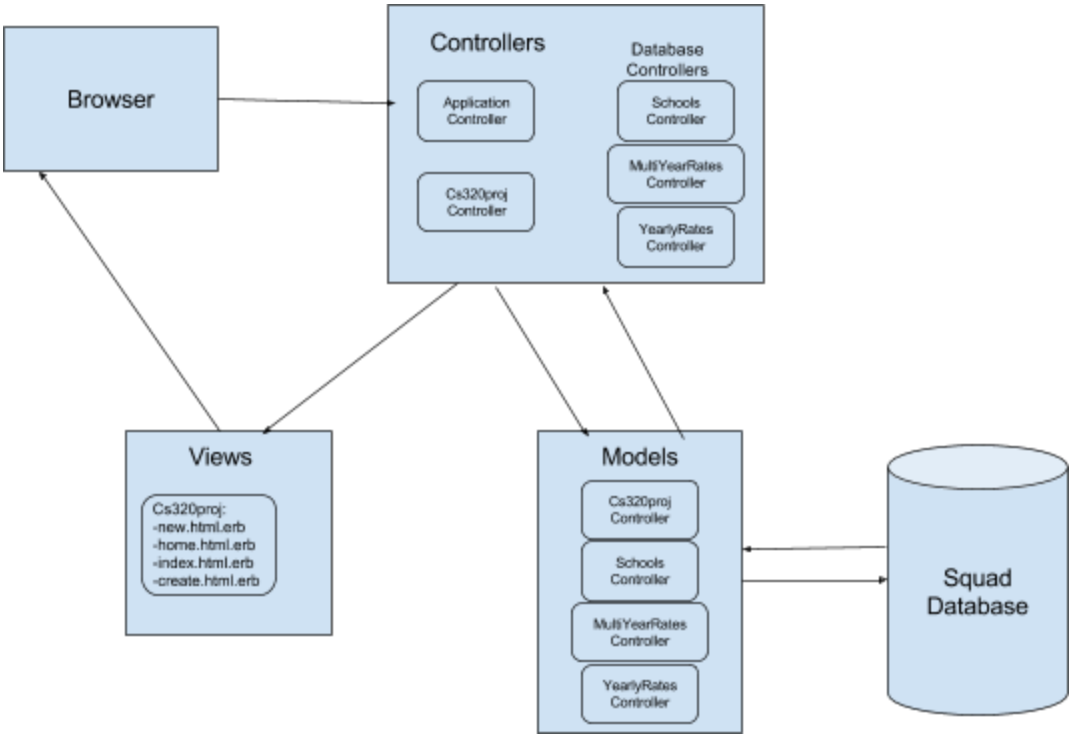
December 12th 2016

Software Organization

The image below shows the Ruby on Rails Application Framework for the MVC (Model View Controller Architecture).



The image below shows our projects MVC Architecture.



As you can see from the diagram, the cs320proj is a part of our project. We have a controller for cs320proj and a controller for every relation in our database: schools, multiyearrates, and yearlyrates. The only views we use in our browser are the views inside cs320proj. We have a model for cs320proj and a model for every relation in our database: schools, multiyearrates and yearlyrates.

The software we used included PSQL, Ruby on Rails, JQuery, CSS, D3, and Bootstrap. We received the data in excel files which then converted to csv files which we uploaded to the PSQL database. The procedure for uploading data can be found in the design document. After we uploaded the data, we then integrated Ruby on Rails and PSQL to have dynamic use of our database on a web server. From here we developed our website using Bootstrap to make the website responsive, JQuery to implement dynamic elements, CSS for style aspects and D3 is the graph creating software we used.

Database Location

Our team database is called squad: it can be accessed by ssh-ing to compute217 and typing "psql squad" to pull up the psql. From there, the command "\dt" can be used to view all of the relations. These relations belong to the squad database and can be accessed by any user part of it. In our submission file, there is a file called "squad.sql" which is where our database has been dumped to using the command "pg_dump squad > squad.sql". This file contains the psql commands used to create, alter and add constraints to our relations as well as a comprehensive listing of all of the data in the database.

The source data that our team received is located inside the "Data Source Files" folder. Within this folder, there are 2 additional folders named "Cohort Graduation Rates" and "Graduates Public By School" since our team received two kinds of data. Our data comes from a single source, the Pennsylvania Department of Education which records and reviews statistical information regarding schools and student performance across the state. The data we are analyzing are the Cohort Graduation Rates and the Graduate Data and Statistics. The Cohort Graduation Rate data "compiles statistical information covering the 4-Year, 5-Year, and 6-Year Graduation Rates for Pennsylvania public schools. The cohort graduation rates are a calculation of the percentage of students who have graduated with a regular high school diploma within a designated number of years since the student first entered high school." The Graduate Data "reports provide information on intended post-high school activity of graduates, including college bound students" (PA Dep. of Edu.).

Bug and Limitations

Present Bugs/Unexpected Function:

- Missing/Null data points cause psql query failure, returns '#' + failure number + 'entryUnfound' onto axis of graph without data entry
- Home Page drop down selection contains schools without data for selectable queries
- Selecting same school more than once returns '#' + failure number + 'entryUnfound' onto axis of graph without data entry
- Long school names cut off by space limitation to the left of axis
- Graph title not capitalized

Limitations:

- No selection for Time Period of Cohort Rates
- No access to Graduate Rates dataset
- No ability to compare more than one rate
- No variation in quantity of schools compared, always three
- No ability to compare over time (select multiple years, or multiple Time Periods)
- No ability to compare across datasets (Cohort rates and Graduate Rates)
- No access to school information past school name

Group Self Evaluation

Accomplishments

In the end, we were successful in both our database creation and the implementation of the database within our application. The design of our database was cohesive and well thought through. The design of the database allowed us to upload all of the information from the excel sheets we were given while limiting the amount of repeated data and wasted space. The database design we had also allowed us to perform queries across the two different sets of data, which was another major accomplishment. Overall, we were successful in the creation of our application as well. Although we did not implement all the functionality we had originally proposed, we were able to create a dynamic, user controlled visualization within the website that gave the user the option to explore the attributes across multiple schools.

Things Learned

As a group and as individuals, we learned a lot in the process of creating this project. First, we learned the process and strategies in designing and implementing a database. We worked together as a group to discuss what the final design of our database would be. We learned to use SQL to create the framework of our database and to upload the data into it. We also learned to use SQL to perform complex queries on our data set. In the process of creating our application we learned to use the various tools (Ruby on Rails, JQuery, CSS, JavaScript(D3), and Bootstrap). Each of these tools were essential parts of our final application. Most of us in the team had little to no experience with these tools, so the project provided a great opportunity for us to combine them to create a single functional web application. Another interesting and important thing we learned was the implementation of a website using a server. Lastly, we discovered the power behind the various group project development tools, mainly GitHub. We had not all used this tool before to work in a team coding project.

Future Work

With the data provided, in the future, this database can be extended to correlate our rates geographically, possibly using heat-maps, to reveal possible connections between location and demographic graduation rate. It can also extend multi-year rates to compare rates of change overtime for each statistic between schools, also possibly geographically, to compare progress/change and reveal locations in need of the most help. In addition, we would like to have implemented different types of visualizations, including line graphs, scatter plots, tables, and pie charts.