



1. A description of the application and its features. State as clearly as possible what you want to do. This can be extracted from the Checkpoint 1 write-up.
  - a. Our final project will be a **web-based application** in the **US college domain**. The context of our domain will be that of college tuition data, which includes the tuition costs themselves, along with fees by colleges/universities for 2018-2019. Additionally, we would also like to visualize the diversity of each college/university, the average net cost by income bracket, historical tuition, and the potential for salaries of each student.

This is important because college tuition data is somewhat difficult to find. As college students, we think it is important to create a more user-friendly web application for students and parents to figure out the trending costs of college tuition by geographic area, and/or salary.

2. The ER diagram of your application domain, plus English description.
  - a. For our ER Diagram we have 5 main entities, These are Tuition income, Tuition Cost, Historical Tuition, Diversity makeup, and salary potential. We picked these because they can be important identifiers when choosing a college. We also wanted to focus on Tuition cost vs. salary potential because that can be an important statistic when picking a university.
  - b. As for the attributes we have many of them to cover in our database. Some of the more important ones to discuss would be attributes such as enrollment, tuition cost, career pay, as well as costs such as room and board and other fees we included. We thought it was important to show these because they can also be relevant indicators when looking at colleges
  - c. Lastly we included a couple relationships into our model. This would be the pays for relationship, which indicated the relationship between tuition cost vs. tuition income. We also included the represents relationship. This is a salary potential vs. tuition cost relationship as it's important to look at salary potential when choosing a tuition type.
3. A description of the functionalities that you plan to offer. At the minimum, you must handle inserting, deleting, and modifying tuples in the database, as well as handling queries that involve joins (all joins), projections, selections, and aggregations (e.g., sum, average, max, min, etc.). Note: notions such as aggregations will be covered shortly in the class.
  - a. Besides the minimum functionalities (inserting, deleting, modifying tuples, handling queries, etc.), we plan to provide visualization functionalities that will show the entities selected by the user. The idea is similar to how a user would look for apartments, or food places nearby, where the user would have a "Search" bar where they would like to place their keyword queries, except we would be showing tuition costs (because when selecting institutions, cost of attendance is one of the biggest decision making points). We plan to visualize data that shows each entity, as well as, comparisons against other colleges. We would also like to create some sort of search parameter for the user. So for example, the user wants to look at colleges with tuition costs <\$10,000, then the app would ask the user whether this is for out-of-state or in-state, and it will show which schools fit that criteria. Then we plan to have a "shopping cart" system where the user selects schools, put them in the "shopping cart", then compare the tuition of all the schools selected.

4. Specify the kind of queries that will be commonly asked in your application in English and how they match with the functionalities that you plan to offer. For example: if you are developing a database to manage movies and want to provide two features: search, and review. For the search feature, you will provide the functionalities to search by directors, by genres, by star rating and date. For the review feature, you will provide the functionalities to display the average rating for a movie, or to enable users to enter a review for a specific movie.
  - a. The queries that will be commonly asked in our search bar will be about cost of tuition. Whether that be about showing the colleges that have under a user specified price or within a range of user specified prices. Another one would be return on investment, they could find which schools have the best ROI based on tuition and salary potential, whether that be mid career or early career pay.
5. Discuss the planned user interface to your database. At the minimum, this user interface must be a text-based one. It is preferred to be a Web-based or mobile app.
  - a. The user interface for our application will be web based and built using react. The UI at minimum would have a search bar and a sign in/create account feature. We would also like to add something that would let you compare different schools and to filter based on any of the entities. We haven't spoken about the actual layout of these features on the web page but it will probably have the basic look of search bar and sign in, on the top right of the screen, then have a hamburger button on the top left to go to other features of our application. We will continue to adjust the layout as we see fit.
6. A description of software platforms/languages that you plan to use.
  - a. Seeing we are going for a web based application, We were planning on using Java for the backend as that is a language we are all familiar with as well as ReactJS for the front end as it should give us the tools necessary to complete the front end.