\*Student Directory\*

Team 09

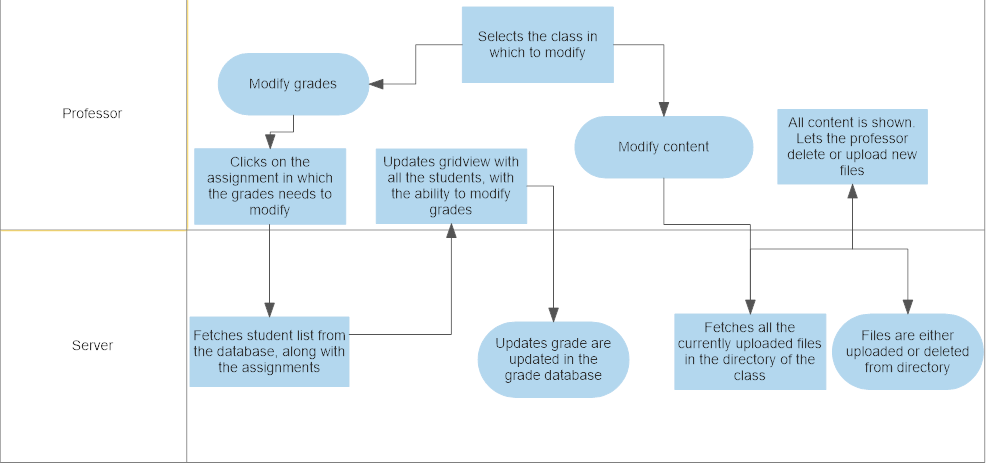
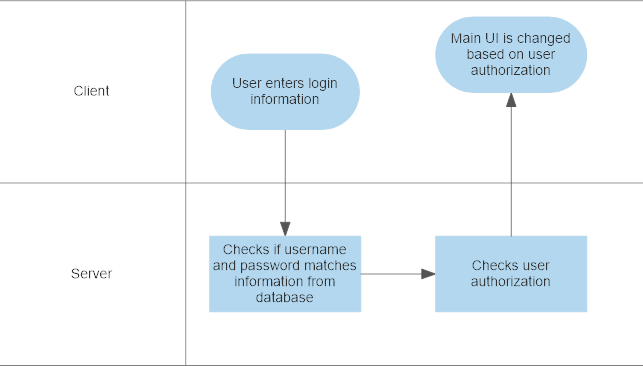
Project Lead: Gregory Billings

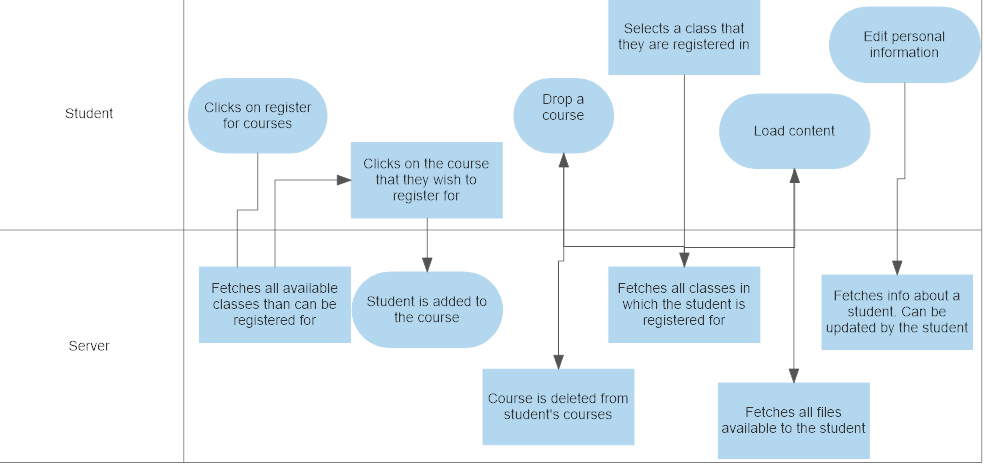
Team Members: Gregory Billings, Francis Poirier

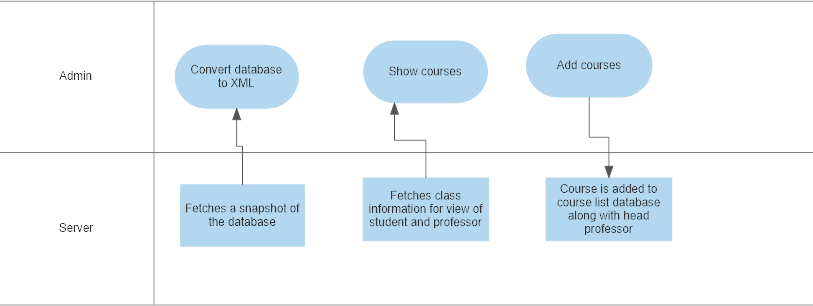
# Project Description:

Software as a service student directory package beta of the standard package, this would be of interest to any tutoring service, school district or post-secondary education. The project at completion will have a basic GUI setup that will allow for the addition of classes, removal of classes, registration of classes and grading system all interacting with a backend database. Produced as a byproduct of main development will be our product prototypes, our design reviews that we will receive in class and the database infrastructure. For the completion of the project, the key requirements are as follows: developing database infrastructure, designing an intuitive GUI, proper testing of the application and proper process development.

# Process Model:

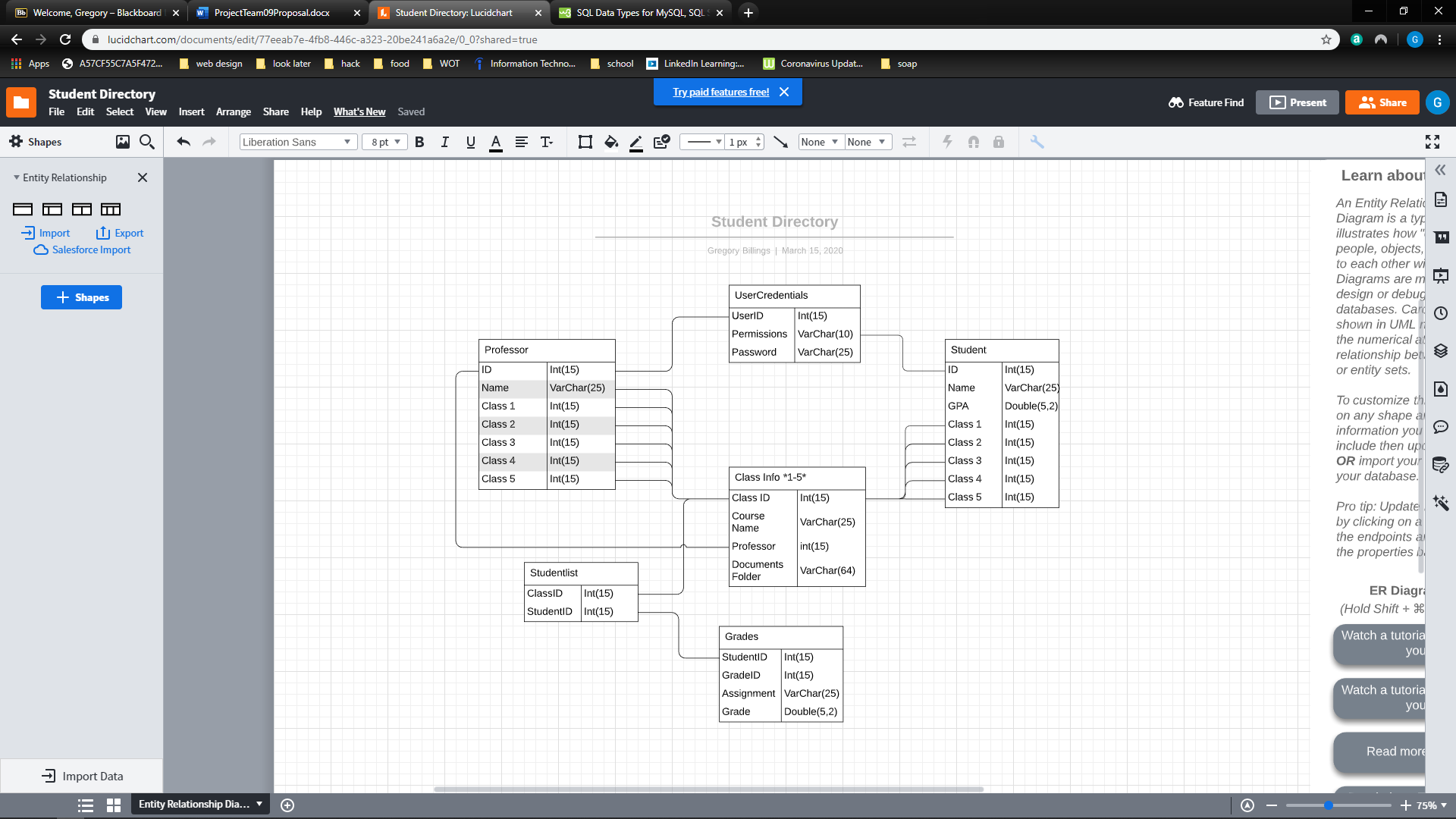






# Database Model:

<https://www.lucidchart.com/invitations/accept/c242a653-1e07-41f3-8311-0bb3a399eebd>



# User Interface:

1. Login screen – Username and password authentication. Will lead to main screen based on user permissions.
2. Students -
   1. Register for a course – May select through an available course list from a grid view. If the student has more than 5 courses, then they will not be allowed to register for more courses. Once a course has been selected, a register for course button will appear. This will add the student to the class list, adding a row to the grades table for the class. If the student has less than 5 classes, the remaining classes will be left null.
   2. Drop a course – Will load current registered courses in a gridview with the option to delete a course.
   3. Show grades – Student selects a registered course from a gridview. This will load all the assignments and midterms on the clicked course that has been publicly shown to student. May look at the assignment/midterm breakdown if it has multiple sections (example: midterm – multiple choice, essay).
   4. Show content – Student select a registered course from a gridview. This will load all the current publicly available files of the clicked course (from file directory linked to professor)
   5. Edit personal information – Will load the profile of a student to edit phone number, email, etc.
3. Professors -
   1. Course Information – Controls are a data grid view, drop-down list and two buttons. All controls are always displayed, the data grid view is an updated display of course information based on the last selected item in the drop-down list. The first button on this UI has the professor go to the grade's modification screen, and the second button brings the professor to a content modification screen.
   2. Modify Grades – Allows the professor to select a specific assignment. This will load a student list for that class in which grades can be modified and saves for the specific assignment.
   3. Modify content – Lists current content uploaded to the directory. This gives an option to either delete already uploaded content and upload new content.
4. Admin (extra flourishes and tech support can be added to this if extra time/ if it's something worked upon after end of course)–
   1. Main screen – The main screen for Administrator will have three Buttons, a TextBox and a RadioButton. The first Button will backup the database to XML. The purpose of the rest of the controls is combined to bring the user to the Show course page, the RadioButton will select whether the perspective will be from teacher or student and the Textbox will be where the user inputs the course ID they need to view, the second Button initiates the search. The third Button goes to the Add Courses screen.
   2. Show course – Shows exactly the searched class from the perspective of the student or teacher with full permissions to edit from both perspectives (logging will be a stretch goal to make sure this power isn't abused with no accountability.)
   3. Add Courses- This screen will show multiple TextBox entries and a Button that allow the user to add new courses to the database. The TextBoxes will be an entry for Course ID, Course Name and Professor of class. If the user tries to add a course without all of these filled an error is thrown.(stretch goal turn this into a modify courses, so that only course ID and course name are required and professor can be changed later, along with being able to delete unneeded courses.)
   4. (Stretch goal: user permissions updater)

//this needs to be changed. Show course is now show user. The stretch goal is now just a user add function.

# Project Plan:

1. User Interface (Week 1: person 1):
   1. By this proposal is handed in we will already have an idea of what we want the user to see, from there it is very simple to implement and is not reliant on any other portion of the project, as such this can be implemented immediately.
2. Database (Week 1: person 1):
   1. If this step is not implemented early on it would become onerous to continue working on other stuff without it, therefore it needs to be done as the first dependency.
3. Data Access (Week 1: person 2):
   1. This is the last step we would have before being able to build a prototype, this needs to be completed so we can start our iterative prototyping.
4. First Rough Prototype (Week 2: Combined):
   1. Forming the first prototype of the full program to see if there are any glaring technical or usability issues to work through or if it needs scrapping and a redesign.
5. Forming Second Prototype or Redesign (week 3: Combined):
   1. Fixing the bugs and modifying what needs to be changed from the first prototype if no redesign is required.
   2. If a redesign is required, we need to establish what was the problems with the Project to create a new design.
6. Apply Polish Week 3: person 1):
   1. Making sure everything looks nice and that all the code is formatted correctly.
7. Final Testing (Week 3: person 2):
   1. We test the final iteration of the project and check the backend data to confirm the results are as expected, fuzzing to make sure that no user input errors could happen and that it is secure.