

The George Washington University

Phase II of Database Project  
Integrating APPS, ADS, and REGS

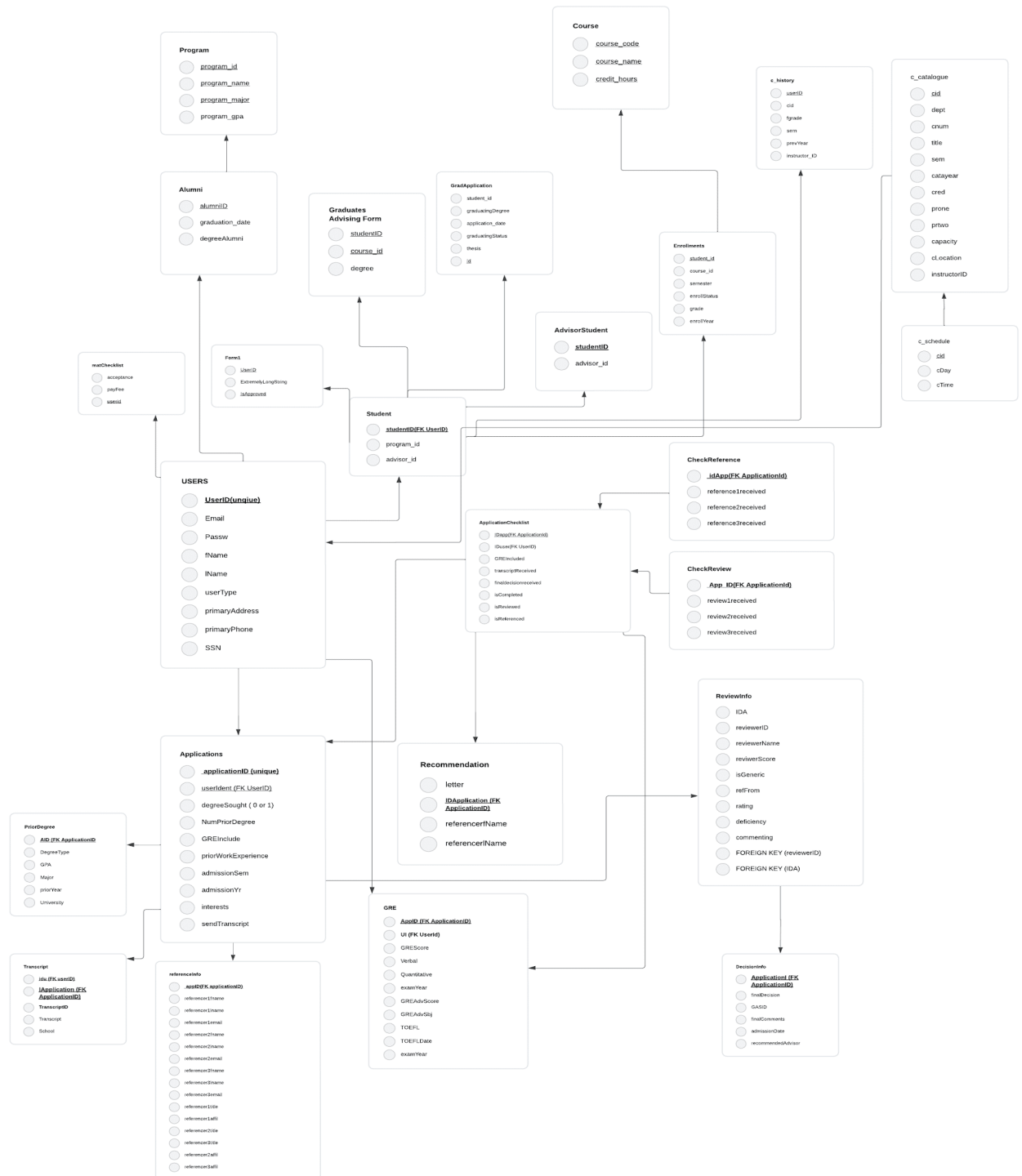
Brendan Jarmusz, Mazen Saadi, Theresa Le

CSCI 2541: Databases

Professor Taylor

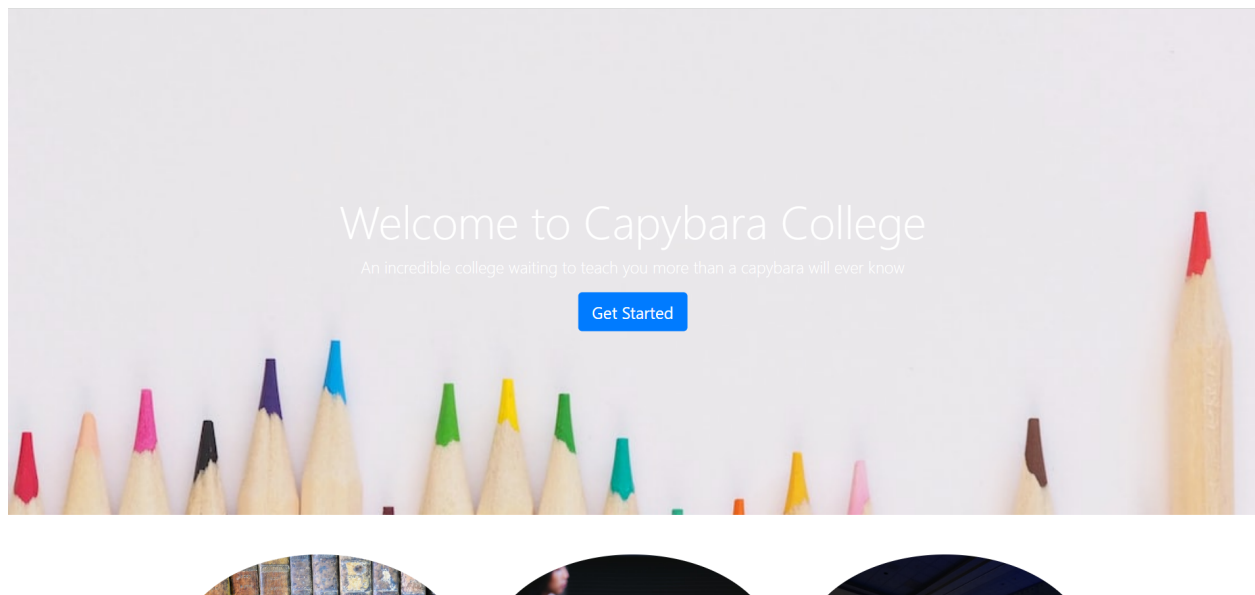
5 May, 2023

## DB Design



Based on the SQL Table Diagram, our tables do not meet a normal form. We did not meet the First Normal Form. Our attributes were atomic as in there were not multiple valued attributes within a column. Even though our attributes in the table are atomic, we do have some tables that have repeating groups. For example, in our referenceInfo table, each referencer(referencer1, referencer2, and reference3) has a column for their first name, last name, title, and affiliation. The purpose of our table is to store information about the referencer and this information is pulled up when a faculty reviewer is reviewing the recommendation letter as well as the application. In addition to the referenceInfo table, checkReference and checkReview also share the same type of data: a boolean that checks if the referencer/reviewer submits a recommendation letter or a review form.

## Visual Overview



## Login

UserID:

Password:

Login

Recommenders Login Here

[Signup](#)

## Welcome, John!

User ID: John

Logout

### Student Dashboard

View Personal Information

Course Catalogue

Registered Courses

View Transcript

Submit Form 1

Talk to College Advisor

Apply for Graduation

## Apply for Graduation

Enter your student ID:

Select your degree:

Select your major:

Enter your thesis:

## Welcome, Bhagirath Narahari!

### Search Student Transcript

User ID:

Last Name:

## Design Justification

In order for an applicant to become a student, they need to apply and be offered an admission as well as pay a fee to be matriculated. If the applicant accepts their offer, a cursor executes where it inserts into matChecklist the userID and sets payFee and acceptance to true, indicating that the applicant is matriculated. From there, the cursor executes an update where it changes the user type in the users table from applicant to student. Afterwards, the applicant becomes a student and the student can still log in with the same credentials.

Similarly, in order for a student to become an alumni, they need to apply to graduate. A student will apply to graduate and their form requires their student ID, graduation degree, major and thesis. Once they submit their application, they are approved if they fulfill their requirements and a cursor will execute an update in the users table where the user type changes from student to

alumni. From there, the new alumni can log in with their same credentials as before, but they are taken to an alumni home page.

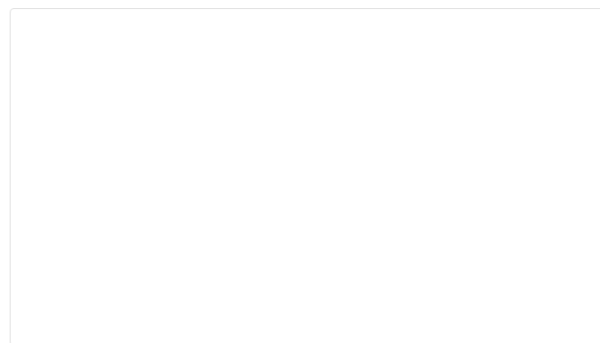
### **Assumptions**

One assumption we had is every faculty member can be an advisor, a reviewer, or both. Another assumption we had is that each applicant needs to have at least one recommendation and at least one review, but no more than three for recommendation or review. We did not have a formal process to email transcripts for an applicant, but once an applicant clicks a button that says “Email transcript”, it is assumed that the transcript has been emailed to the school and the database automatically updates that the transcript has been received. Furthermore, once a user is in the database, they have the same userID from when they were an applicant to when they become an alumni.

### **Special features**

Return

**Talk to our College Advisor, Kev!**



Type your message

Send

A special feature we included was implementation of a support chat with Kev. Students can reach out to Kev with questions regarding their courses and school overall.

### **Work breakdown**

Theresa started the ER diagram and she created the SQL script. She created functions that changed an applicant to a student. In addition, she implemented additional functions to APPS, such as creating queries for the faculty reviewer to look up applicants and functions to edit an applicant's personal information. She also added a feature that the applicant had an option to email their transcripts, which updates the database automatically once the transcript is emailed. Furthermore, she added the ability for an applicant to have several reviewers and recommendation letters, which the faculty reviewer can see as well as GS and CAC. She also added a status so that the applicant can see what stage their application is in. She also wrote the report.

Maazen modified his REGS where he had to take into consideration retrieving information for a student from when they were an applicant. He contributed to the ER design. In addition, Maazen created the sign up where it assumes the person signing up is an applicant. He implemented several queries that the GS can search. Additionally, Maazen implemented functions where faculty can edit their personal information, such as phone number, email, and address. Maazen worked on checking whether a student can register for courses that overlapped or that required prerequisite courses.

Brenden did the styling and implementing bootstrap into the html files. Additionally, he made significant contributions to the extra feature. He contributed to the ER design. He

implemented how a student becomes an alumni. He worked on the submission of form 1 and a function where a student is assigned an advisor.