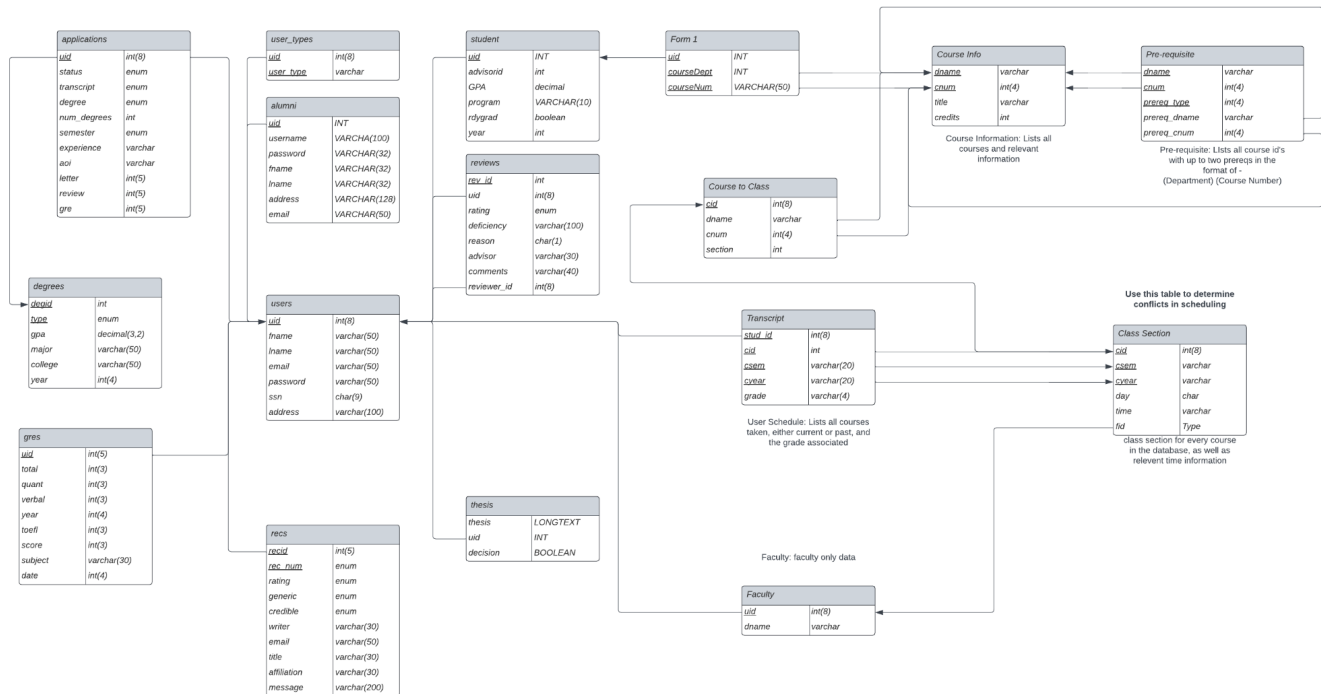


Phase 2 Report

(Ben Chapman, Joshua Kweon, Son Nguyen)

I. DB Design:

Phase 2 ER Diagram



In order to ensure the integration is done without issues, we tried to minimize any changes made to each of our modules. As a result, much of our databases have elements that are in 0NF, while some tables such as users, course info, user types, prerequisites are in 3NF. In these particular tables, all fields are atomic, there is no transitive dependency nor partial dependency. Although most of our tables are in 3NF.

One major change that we made to our database is the separation of the user types into a separate table where the primary key is the composite of the uid and the type. This is because our previous individual systems would not have supported multiple user roles, and thus this change was made to accommodate for the integration.

II. Visual Overview:

Personal Info

Student Info

View Applicants

View Advisee

Current Students

Graduating Students

Alumni

View Form 1

Student Info

First Name

Last Name

User ID

Search

Ringo Starr2

University Info

University ID: 3

Program: PhD

Ready to Graduate: No

GPA: 4.00

Personal Info

Email: ringo2@gmail.com

Full Name: Ringo Starr2

Address: 123 abc st

Advisor: Gabe Parmer

Input Advisor's ID

Assign Advisor

Class Info

Fall 2023

Course Department	Course Number	Course Name	Credits	Day	Time	Grade
CSCI	6221	SW Paradigms	3	M	15:00-17:30	A <div>Grade A</div> <div>Submit</div>
CSCI	6461	Computer Architecture	3	T	15:00-17:30	A <div>Grade A</div> <div>Submit</div>
CSCI	6212	Algorithms	3	W	15:00-17:30	A <div>Grade A</div> <div>Submit</div>
CSCI	6232	Networks 1	3	M	18:00-20:30	A

Course Title

Search

Classes Offered

Class ID: 1

Semester: Fall

Year: 2023

day: M

Class Time: 15:00-17:30

Instructor: narahari narahariname

Department: CSCI

Course Number: 6221

section: 1

Title: SW Paradigms

Credits: 3

Add

Class ID: 2

Semester: Fall

Year: 2023

day: T

Class Time: 15:00-17:30

Instructor: narahari narahariname

Department: CSCI

Course Number: 6461

section: 1

Title: Computer Architecture

Current Registration

Classes Enrolled

Class ID	Course Department	Course Number	Course Name	Credits	Day	Time	Grade	Remove
1	CSCI	6221	SW Paradigms	3	M	15:00-17:30	A	Drop Class
2	CSCI	6461	Computer Architecture	3	T	15:00-17:30	A	Drop Class
3	CSCI	6212	Algorithms	3	W	15:00-17:30	A	Drop Class
4	CSCI	6232	Networks 1	3	M	18:00-20:30	A	Drop Class
5	CSCI	6233	Networks 2	3	T	18:00-20:30	A	Drop Class
6	CSCI	6241	Database 1	3	W	18:00-20:30	A	Drop Class
7	CSCI	6242	Database 2	3	R	18:00-20:30	A	Drop Class
8	CSCI	6246	Compilers	3	T	15:00-17:30	A	Drop Class
9	CSCI	6251	Cloud Computing	3	M	18:00-20:30	A	Drop Class
11	CSCI	6260	Multimedia	3	R	18:00-20:30	A	Drop Class
12	CSCI	6262	Graphics 1	3	W	18:00-20:30	A	Drop Class

Class Info

GPA: 4.00

Upcoming Semester: (Fall 2023)

Course Department	Course Number	Course Name	Credits	Day	Time	Grade	Drop
CSCI	6221	SW Paradigms	3	M	15:00-17:30	A	Drop
CSCI	6461	Computer Architecture	3	T	15:00-17:30	A	Drop
CSCI	6212	Algorithms	3	W	15:00-17:30	A	Drop
CSCI	6232	Networks 1	3	M	18:00-20:30	A	Drop
CSCI	6233	Networks 2	3	T	18:00-20:30	A	Drop
CSCI	6241	Database 1	3	W	18:00-20:30	A	Drop
CSCI	6242	Database 2	3	R	18:00-20:30	A	Drop
CSCI	6246	Compilers	3	T	15:00-17:30	A	Drop
CSCI	6251	Cloud Computing	3	M	18:00-20:30	A	Drop
CSCI	6260	Multimedia	3	R	18:00-20:30	A	Drop
CSCI	6262	Graphics 1	3	W	18:00-20:30	A	Drop

Personal Info

Student Info

View Applicants

View Advisee

Current Students

Graduating Students

Alumni

View Form 1

Alumni

Graduation Semester

Degree

Graduation Year

Search

Student Name	Student Email	Graduation Year	Admitted Year	Degree
Paul McCartney	pm@gmail.com	Spring 2023	None	Master's
Eric Clapton	ec@gmail.com	Spring 2014	2010	Master's
Alumni2 Lname	alumni2@gmail.com	Fall 2023	2019	Master's
Alumni Lname	alumni@gmail.com	Fall 2022	2018	Master's

III. Assumptions Made:

- Queries for graduating students are for their admit year, as all students who are graduating will be this year, making it redundant if it's by graduating year.
- Course descriptions would be stored in a file system where each file description maps to a course department and number instead of a database, hence no course descriptions are stored in the database.
- Application must have three letters of recommendation

IV. Design Justification:

Due to time constraints and the complexity of each component, we designed the new system where each of our members only requires cursory knowledge of the workings of every other member's modules. A number of design choices were chosen to facilitate this:

- The database is integrated with minimal changes except where needed.
- Each user page employs nav panes that are easily added to the html. Hence, each functionality for each user is done in a separate html, then to be integrated into the actual html after. This ensures that we do not have any conflict or error that slows down our progress.
- Certain functions are redone (such as updating user information) to be able to accommodate for all different info and user types, allowing for better code reusability.

With respect to the front-end, we utilized the same css file style throughout each html file to ensure consistency in styling. The website's appearance is also cleaned and made straight forward via the inclusion of a navigation bar at the top of our website that would present different tabs depending on which user was logged in so that they can easily access their needed routes and information. The addition of the navigation bar made integration a lot more seamless as we could initially add our html pages and functions from our phase 1 portion as a new tab on the navbar and then connect them afterwards. As the navbar utilized jinja conditional to check which route would be added, all that needed to be done for each html file was add this same navbar to them to ensure consistency in appearance.

Some other key design aspects include:

- Our system only supports 2 semesters - Spring and Fall.
- The form 1 is used as the advising form where the lack of such a submission would place an advising hold. This was agreed by the group as the more logical choice rather than creating a new separate form.
- Recommenders can log in without needing a password. It would simply check if their email, which is unique, is a recommender email.
- Students can drop classes even if they already got a grade for it so long as the class is either in current or next semester (but they can only enroll for next semester). This is intentional as we wanted students to have this capability and be able to retake the class.
- Admins are the only ones who can create accounts (except for applicants who can make their own account).
- Each class meets only once a week. This is intentional to enable a simpler and cleaner product, however the database design does support meeting multiple times a week.

V. Special Features

One additional special feature that we added was that the advisor hold corresponds to a student's ability to register so only once the hold is lifted, then they can register for classes. This hold will take effect until the student submit a form 1 (and have that form 1 be accepted).

VI. Work Breakdown:

- **Son:**

I worked on integrating all of the specific functionalities of the registration system:

- Students can register for classes and drop them as long as they fulfill the requirements (no advising hold, fulfill prerequisites, no schedule conflict).
- All users can update their own personal information.
- GS and Instructors can update student grades, but Instructors can only do so once.

- Search functions for each user page to allow them to search for others (if their role permits).

Other functionalities I worked on include:

- CSS styling, particularly the navbar, queries, and tables.
- Creating the tabs for the user page, using app code to make the addition of any new query and their display requiring only 2 lines of code.
- Integration of my teammates's html files into each user page.
- Rework update info, update grade, update GPA, and user information routes to allow for reusable code.
- Additional queries for the GS, such as the current students, graduating students, alumni, etc.
- Query for advisor to get list of advised students.

- **Ben:**

I worked on integrating the APPS functions. This included:

- Changing previous APPS code to fit the new database design.
- Requiring the applicant to have three letters of recommendations
- Allow for multiple reviews to be made for an application.
- Functionality of the CAC, Applicant, Reviewer, part of the GS and Admin
- Made some of the queries for applicant data that the GS could run
- The automatic system for an applicant to become a student when they accept their offer. And when their application had the transcript and letters of recommendation in the status automatically was complete.

- **Josh:**

I worked on integrating all of the ADS specific functions into the website (form 1, assigning an advisor to a student, graduating a student, creating alumni) while also modifying the schema so that the integration of both the REGS and APPS components can transition more seamlessly into the ADS schema. There were new features that needed to be added like a PhD student's ability to write and submit a thesis which ultimately made the check for their ability to

graduate more meticulous. I also worked on creating an advisor hold for when a student creates a form 1 so that their ability to register for classes is not available until their respective advisor approves their form 1.