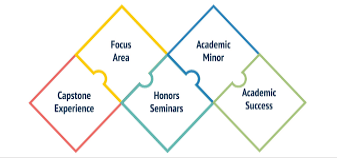
Engineering Honors Program Database Optimization

ENGINEERING HONORS PROGRAM
UNIVERSITY OF MICHIGAN

Students: Joseph Charboneau (Honors Capstone), Katie Christiansen (Honors Capstone)
Advisors: Dr. X. Jessie Yang, PhD students Cheoljoon Jeong & JinYong Kim

Introduction

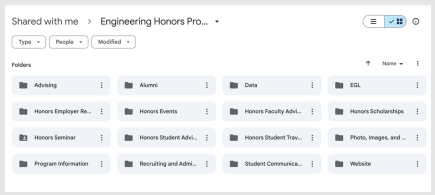
The College of Engineering Honors Program was established in 2012 for high-achieving engineering students to develop their academic rigor while enhancing their leadership skills. Working closely with faculty and peers, students complete a capstone project, elect a minor, develop skills through Honors seminars, and partake in community service projects.



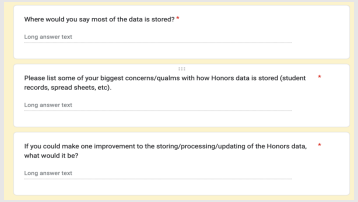
As the Honors program grows, consequently its student and program data increases. **Honors staff and faculty are struggling to find a way to mediate data changes and additions while maintaining data organization**. Data is stored in a variety of locations, making it difficult to work timely and to keep updates of changes to students' academic plans consistent.

Objective & Data Analysis

The goal of this project is to consolidate and optimize the systems which store Honors student information. Working with Honors staff, we aim to make it easier to insert, update, and delete students records from where the data is being held, and allow current data to communicate with each other.



Engineering Honors Program Google Drive Folder Main Page



Google Form we created to survey Honors staff on their perspective of current data organization

- The program currently houses their data across several different online storage spaces
- Primarily, student data is located in a Google Drive called Engineering Honors Program
- We focused on five main spreadsheets storing student data such as name and demographics, major, minor, year, capstone focus area, and Honors progress
- We found that this data is repeated in multiple spreadsheets, and these spreadsheets do not communicate with each other. Any edits are currently done manually.

The biggest issue besides limited organization is the data's inability to communicate with each other. Thus, we have proposed three solutions for Honors staff to restructure their data: **(1) a new database built using MySQL, (2) implement LOOKUP solutions within the spreadsheets for data to match, and (3) a set of rules for staff to update student data**.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	LAST	FIRST	PREF I	UMID	CAMPUS ID	EMAIL	FOCUS	BSE MAJOR	MINOR	FOCUS ARE	CAPSTONE	NOTES	Status Check N	Design Expc
2	Charboneau	Joseph	Joseph	[redacted]	CHARBO	charbo@umich.edu	Design	Computer Science (EECS)	Mathematics	#N/A	Approved	430 FA23 - working w/ accurate		
3	Christiansen	Katherine	Katie	[redacted]	KACHRIS	kachris@umich.edu	Design	Industrial and Operations Engine	Mathematics	#N/A	Approved	430 FA23 - working w/ accurate		

Example student data from "CoE Honors Capstone Assignments" spreadsheet; some data is missing, inaccurate, or improperly formulated

Proposed Solution (Optimized Database)

New Organization Explained

The previous organization of the CoE Honors data was **redundant**, **inconsistent**, and **prone to inaccuracies**. We felt that before building a relational database, we needed to sift through the currently existing google sheets and find the most efficient way to organize the data. We then changed the organization of the data to consist of 9 new tables: **AdmissionsIndividualChecklist, AdmissionsMassChecklist, CapstoneChecklist, CurrentStudents, EGLToHonors, GraduationChecklist, HonorsFocusAreaProposals, HonorsToEGL, and WithdrawalChecklist**. Much of the redundancy and inconsistency of information will be eliminated with this model, and many of the previously tedious processes will be automated by triggers. We have also included a **list of data standards**, **description of tables**, and a **description of triggers**.

Data Standards

- Capitalize the first letter of each word in a table name.
- Column names are all lowercase, with words separated by underscores.
- Create descriptions for all columns, so there is no ambiguity as to what columns mean.
- Each table includes a table description.
- Keep entries consistent.
- Create a list of standard abbreviations for consistency.
- Details on what authority to grant for specific types of situations.
- A list of who can approve what types of database authorization requests.
- Table-by-table list of who has read/read-write permissions.
- Procedures for removing security from retiring, relocating, and terminated employees.

Triggers

students_to_admissions
Upon insertion of a new student into the CurrentStudents table, that student will also be added to the AdmissionsIndividualChecklist table

withdrawal_student
When a student is inserted into the withdrawal_student table, they are deleted from the CurrentStudents table, creating a cascade effect, deleting them from other effective tables

new_capstone_student
When a student has admissions_type = Honors, a row for that student will also be added to the CapstoneChecklist table

transfer_capstone_student
If a student's admissions_type is updated from 'EGL' to 'Honors' in the CurrentStudents table, a row for that student will also be added to the CapstoneChecklist table

egl_delete_capstone
If a student's admissions_type is updated from 'Honors' to 'EGL' in the CurrentStudents table, delete that student's row from the CapstoneChecklist table

add_future_graduate
When a new student is added to the CurrentStudents table, they are automatically added to the GraduationChecklist table

capstone_completion
When all the capstone checklist fields are true in CapstoneChecklist table, set capstone_complete = true for that student in the GraduationChecklist table

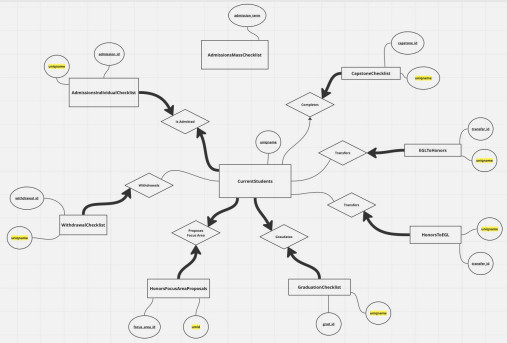


Table Descriptions

CurrentStudents
This table holds all the students who are currently enrolled in the CoE Honors Program, undergraduates and graduates. This is the central table that holds most of a student's information. It is also where most of the updates will be made to a student's personal information.

AdmissionsIndividualChecklist
This table holds the checklist that needs to be completed for each new admission. When a new student is added to the CurrentStudents table, they are automatically added to this table.

AdmissionsMassChecklist
This table holds the checklist that needs to be completed for each new admission term. These are the tasks that need to be completed for all students together: updating Honors roster, removing students from email lists, etc.

HonorsFocusAreaProposals
This table holds all students' focus area proposals. This table will be connected to a google form through the SAS Zapier.

HonorsToEGL
This table holds all students who have transferred from Honors to EGL. When a student's "admissions_type" column is updated from 'Honors' to 'EGL' in the CurrentStudents table, they will be automatically added to this table.

EGLToHonors
This table holds all students who have transferred from EGL to Honors. When a student's "admissions_type" column is updated from 'EGL' to 'Honors' in the CurrentStudents table, they will be automatically added to this table.

CapstoneChecklist
This table holds the checklist for all the capstone projects of students of admissions_type of 'Honors'. When a student of admissions_type 'Honors' is added to the CurrentStudents table, they are automatically added to this table. They will be deleted if their admissions_type is changed to 'EGL'.

GraduationChecklist
This table contains the graduation checklist for all current students. They are automatically added to this table when they are added to the CurrentStudents table.

WithdrawalChecklist
This table holds all students who have withdrawn from the Honors program. When a student is added to this table, they are automatically deleted from the CurrentStudents table.

Discussion & Alternatives

Through data analysis, data experimentation, and optimization research, **we found that developing a database will serve as the most efficient solution for the Honors Program's data organization problems**. A database would allow all data to be in one area, updated frequently, and nonrepetitive.

However, a true database takes large technical skill to develop. In addition, users would need to understand the database and learn SQL to query search results. As a result, we have also developed **a list of rules and regulations on how to update and access the current data, if nothing were to change.** This list assigns staff members to a specific table where they can update information. While manual input is still required, each staff member has their own job, allowing the current setup to function more efficiently and staff members to communicate with each other.

This list is an immediately implementable solution that Honors staff can utilize until the technical solution is better understood.

Conclusions & Future Work

Optimized Database

The details provided in the proposed solution are a launch point for the College of Engineering Honors Program to migrate their student data to a more sophisticated database. The program could either enlist the University of Michigan IT department to build this for them, or possibly hire a student intern with sufficient knowledge to take care of the implementation. The implementer will need to take additional steps to create the fully functioning database, including:

- Choosing a hosting provider with cloud capabilities (AWS, Microsoft Azure, etc.)
- Create a virtual server on the hosting provider to host MySQL
- Install MySQL on said server
- Create the tables and triggers listed here on MySQL
- Test implementation using the tests listed here
- Set up user access (create Role Based Access Controls)
- Implement collaborative features (should be included in the host's toolbox)
- Define protocols for backup and maintenance of the database

Alternative Future Solutions

- If the database is too complex, utilize LOOKUP functions to connect Google Sheets
- Utilize LOOKUP functions in Excel and use Microsoft VBA to build a simple database
- Many students seek on-campus internships; Honors would benefit from hiring a student intern to develop a technical backend for the program's data
- Consolidate data to ONE cloud storage location (IE, all in Google Drive or OneDrive instead of multiple units)