CMD Degree Guide - Project Plan

Ethan Cha, Peyton Elebash, Haley Figone, Yaya Yao

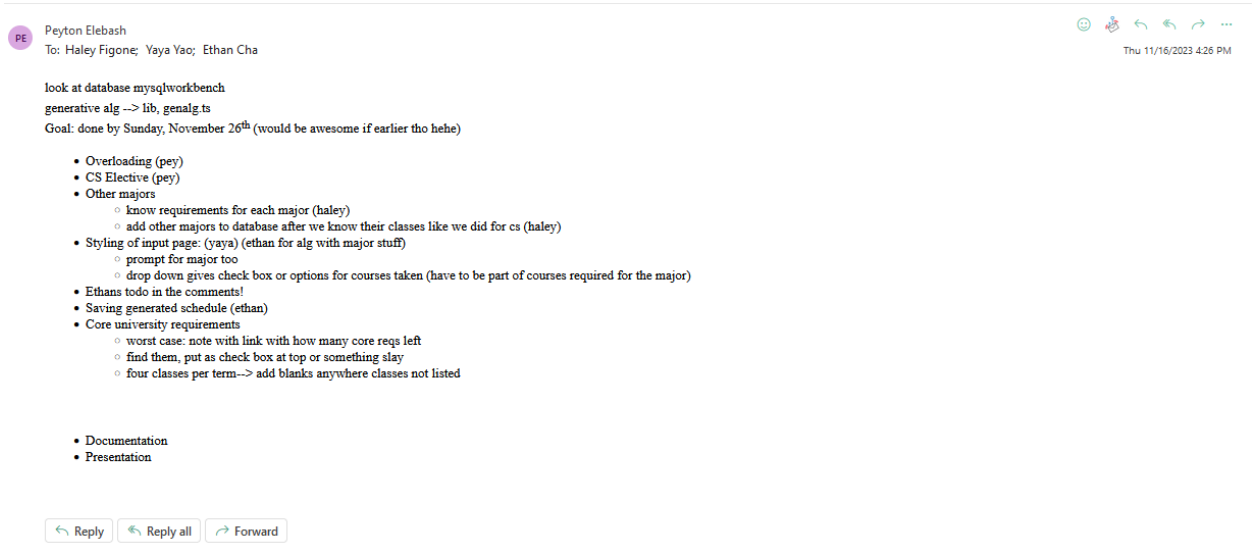
Table of Contents

Management Plan 1
Build Plan 2
Projected Milestones 2
Rationale 3
Risk Management 4
Work Log 4

Management Plan

We divided our team into a front end team and a back end team. Team members Yaya Yao and Haley Figone are on front end, having the most prior experience working in JavaScript and HTML/CSS respectively. Haley's responsibility was focused on HTML/CSS, in particular focusing on standards compliance, accessibility, and responsive design, as well as maintaining compliance with the architectural design. Scripting logic for pages such as the Course Selection page and the Degree Guide Display was delegated to Yaya. Team members Peyton Elebash and Ethan Cha were in charge of the back end, having more practical experience with Python and databases. Peyton developed the main algorithm for degree guide generation. Ethan created and maintained the database for storing user and course data. Ethan also flexed heavily between the back and front end, focusing on implementing on the logic bridging the gap between the two (namely database queries and integrating the generative algorithm with the display).

Our approach to team meeting, communication and decision-making was loosely based on the Scrum framework, adjusted according to the smaller time frame and scope of the project. Meetings happened weekly, in-person, with occasional on-demand meetings for any specific and urgent problems requiring discussion. During each meeting, we developed a weekly agenda, distributed to all members via email. Minor issues or questions were dealt with over text, but major difficulties or decisions about the project were reserved for the in-person meetings due to the need for clear and easy communication. We used a shared GitHub repository to maintain version control.



Peyton Elebash
To: Haley Figone; Yaya Yao; Ethan Cha
Thu 11/16/2023 4:26 PM

look at database mysqlworkbench
generative alg --> lib, genalg.ts
Goal: done by Sunday, November 26th (would be awesome if earlier tho hehe)

- Overloading (pey)
- CS Elective (pey)
- Other majors
  - know requirements for each major (haley)
  - add other majors to database after we know their classes like we did for cs (haley)
- Styling of input page: (yaya) (ethan for alg with major stuff)
  - prompt for major too
  - drop down gives check box or options for courses taken (have to be part of courses required for the major)
- Ethans todo in the comments!
- Saving generated schedule (ethan)
- Core university requirements
  - worst case: note with link with how many core reqs left
  - find them, put as check box at top or something slay
  - four classes per term--> add blanks anywhere classes not listed

- Documentation
- Presentation

Reply Reply all Forward

Example meeting agenda, Week 8

Build Plan

The front end teams and back end teams began work individually and simultaneously, working inward from each side and eventually meeting in the middle. After each module was mostly complete, the team regrouped and formed a collaborative effort to fully integrate the front and back end. The front end is a website; the main degree guide tab is populated by data provided and sorted by the database and sorting algorithm on the back end. The modularity of our architecture allowed the two sides to be developed largely independently, though regular progress meetings ensured that both teams were working towards the same set of specifications, and that they maintained compatible interfaces. The implementation of the front end was simpler than the back end, which was expected and planned for from the outset. After finishing their work, front end team members were expected to flex roles to help finish the back end and start to integrate, as well as completing and polishing documentation. This also allowed for some degree of flexibility in the event of scheduling issues or other difficulties.

The front end was implemented using the React framework, which it was thought would best support the drag-and-drop functionality we initially wanted for the final product. The database was implemented using MySQL. The backend algorithm was originally developed in Python by Peyton, then ported to JavaScript by Ethan in order to provide the most convenient interface with the website.

Projected Milestones

Week 1:

- Project concept

Week 2:

- Project plan

Week 3:

- Project proposal and architectural design

Week 4:

- Basic "skeleton" implementation — empty "shells" of programs and webpages which illustrate basic app structure.

Week 5:

- Usable, fully-rendered webpages without backend integration
- Database setup

Week 7:

- Generative algorithm completed

Week 8:

- Degree guide generates and populates

Week 9:

- User state information saves to database

Week 10:

- Degree guide fully interactive
- All majors functional — courses present in database, degree guide generates according to selected set of requirements.

Rationale

Our plan was designed to accommodate a large degree of flexibility. Most of us were working with technologies we weren't totally familiar with, such as setting up a MySQL database or developing a website using React. We also have very little practical experience, at this point, with implementing many types of algorithms and solutions. This contributed to a lot of uncertainty about how the project would proceed, so we opted for a development framework which would allow us to make lots of changes on the fly. We assigned roles to our members which played to each member's unique strengths and levels of experience, and kept the milestones schedule loose in order to account for future potential difficulties.

To accommodate for this, we tried to make the functionality as modular as possible. The functionality for each major in the degree guide was implemented one-by-one, prioritizing the Computer Science major first. This allowed us to suspend implementation of majors as necessary in order to meet time constraints, as well as being able to utilize a single major as a smaller, easier-to-debug testing ground. The idea of the structure was to make the code easy to update in the future, so that adding more courses or majors would be relatively trivial.

Risk Management

The biggest risk source of risk for this project was that most of us were working with at least one development tool, framework, or language that we were unfamiliar with at the time of starting. In particular, most of our group members were either unfamiliar or only somewhat familiar with JavaScript. We attempted to mitigate this risk as much as possible by assigning responsibilities to team members according to what they already had experience with. However, since most of the final product would use JavaScript, and since it was necessary for members to have a sufficient understanding of how the parts they worked on would interact with the other components, there was only so much we could do about this. In the end, we decided that React was the best choice of tool for our purposes, in spite of these drawbacks. We deemed the risk of unfamiliar technology a necessary trade-off for achieving our end goal.

Work Log

Task	Assigned To
WEEK 3	
SDS	Team
SRS	Peyton, Team
Create Intro Presentation	Yaya, Team
Initial Project Plan	Haley, Team
Architectural diagram	Haley
Configure Git repository	Ethan
WEEK 4	
Set up MYSQL database via Google Cloud Compute	Ethan
Set up production server to host web app via Vercel	Ethan
Build data structures for course and student info (DAG)	Peyton
HTML skeletons for landing/login pages	Haley
Homepage visdev	Yaya
HTML skeletons, Navbar	Yaya
WEEK 5	
Structure database	Ethan
Populate database	Ethan, Peyton
Design topological sorting algorithm	Peyton
Login/signup layout, styling, I/O	Haley

Degree table visdev, markdown base, styling	Yaya
Landing page layout, styling, updated navigation	Haley
Update/document changes to architectural design; new architectural diagram	Haley, Team
Update SRS/SDS	Team
WEEK 6	
Adjusted algorithm to sort courses into terms based on topologically sorted path	Peyton
Login/signup basic functionality	Haley
Integrate database into main repository	Ethan
Update documentation	Yaya
Login/signup final functionality	Ethan
WEEK 7	
Course, major selection, and schedule generator output basic functionality	Ethan
Add other majors to DAG	Peyton
Address course overloading problem in degree guide generation	Peyton
Course and major selection page	Yaya
About and Contact pages	Haley
Store user account information in database	Ethan
WEEK 8	
Populate degree guide table with database contents	Ethan
Make classes taken input a checklist that changes based on selected major	Ethan
Added CS upper division electives to the CS major DAG	Peyton
Added generative algorithm support for Data Science and Math (pure)	Peyton
Styling cleanup	Yaya
Update documentation	Team
WEEK 9	
Clean up more CSS, fix visual and interaction bugs, improve UX	Ethan, Haley
Save degree guide in database alongside user account info	Ethan
Add database fetching to generative algorithm	Ethan
Fix/finalize Navbar	Haley
Finalize layout/styling	Yaya, Haley
Course selection checklist	Yaya

Links implementations	Yaya
Input page layout and design	Yaya
Fix and check for generative algorithm bugs	Peyton
Finalize documentation	Team
Final presentation prep	Team