Senior Design Server/Client Development for Project Matching

Design Document

Team 03

Dr. Nicholas Fila	Client and Faculty Advisor
Sierra Jones	Team Leader and Organizer
Ally Finger	Client Interaction
Brady Loew	Database/Server Manager
Cheyenne Carlson	Head of QA/Testing
Connor LaFerle	UI/UX Lead
Logan Christianson	Algorithm/Research Lead
Jade Yang	Team Coordinator & Scribe
Team Email	sdmay22-03@iastate.edu
Team Website	sdmay22-03.sd.ece.iastate.edu

Table of Contents

1 Requirements

1.1 Problem Statement

Senior Design professors need a program or web application to generate project assignments and teammate assignments for future semesters of Senior Design.

The current method for determining project preferences in senior design is not ideal for neither clients, students, nor professors. Some pertinent problems include:

- Non-automated team selections
- Low percentage of student/client preferences met
- Non-user-friendly project submittal
- Minimal project preference information gathered
- Non-user-friendly project reviewing
- User/project information isn't stored in centralized location/database

1.2 Requirements & Constraints

Functional requirements:

- Must consider the following criteria for team assignments:
    - Teammate preferences
    - Project preferences
    - Project skill set requirements
    - Meeting type preferences
    - Student-proposed projects automatically get assigned that student
    - Best practices for collaborative learning
- Web application that allows both students and professors to list their preferences and receive their teams/projects

Resource Requirements

- Database to store project and student information
- Server that connects database to assignment algorithm
- Web application to collect user information for assignment algorithm and displays the results

UI requirements qualitative aesthetics requirements:

- Easy for users to understand and to enter their information
- Make all options easy to see and understand what they are
- Make it look aesthetically pleasing
- Web application should be accessible on pc and mobile devices

Database requirements

- Have the database be accessible to the team assignment algorithm when needed

Constraints

- Time
    - End of Spring semester (May 2022)
- Technology
    - MySQL - Database Management System
    - Laravel - Website framework
- Legal
    - FERPA
- Cost
    - Free - No direct cost to Senior Design
- ABET Accreditation Requirements

1.3 Engineering Standards

- IEEE 1008-1987 - Software Unit Testing
    - This describes that the process of creating software is determined by different phases, activities, and tasks, which will be upheld by us while creating our project.
- WCAG 2.1 - Web Content Accessibility Guidelines
    - This is a web application, therefore it is necessary to provide equal opportunity and access for all individuals who may use it.
- IEEE 12207-1996 - Software Life Cycle Process
    - This lays out a common framework for developing and managing software projects.
    - https://standards.ieee.org/standard/12207-1996.html
- IEEE 16085-2020 - Risk Management
    - Provides guidance on risk management associated with software projects.
    - https://standards.ieee.org/standard/16085-2020.html

1.4 Intended Users And Uses

The application is intended to benefit all parties involved in Senior Design, being the professors, students, and clients.

Senior Design Instructors will be able to…

- mass-upload student information as a csv file
- add, edit, and remove student information and client projects
- approve/refuse proposed projects
- run the team formation algorithm

Students will be able to…

- view all approved projects and associated information
- submit project/team preferences

- choose their skill sets to get a project that closely relates to them
- save their progress in preferences form
- “Favorite” projects that they like when looking through projects

Clients will be able to…

- easily input their project description and requirements at their own convenience `
- view, edit, and remove their submitted projects online

IT Department will be able to...

- upload student resources (Git/Senior Design Website)
- access all available semesters to deal with issues (Admin access)

2 Project Plan

2.1 Project Management/Tracking Procedures

A combination of waterfall and agile management styles will be used throughout the project. We have split the project into two parts, with each part having a preferred management style.

Waterfall: Team assignments algorithm

- There is a clear start and end point to reach towards
- Once something is finished, there is no need to go back towards it
- Client doesn't need to be directly involved with the implementation and coding but should be kept in the loop of the basic design and direction of the algorithm throughout the process. This is to ensure the algorithm is meeting the needs/desires of the client.

Agile: Web and database development

- Unlike the team assignments algorithm, there is no well defined start and end.
- Client gets the opportunity to give feedback on the changes and improvements that they want to the application
- Each part of the application can be built up over time

Current methods to track progress/communicate:

- Discord: Communication, Time management/Planner
- Git: Managing the project, coordinating work
- Trello: Gantt chart. Easy to track completed tasks and things that still need to get done

2.2 Task Decomposition

Our project will be broken down into three sub-projects and one general research stage. The three sub-projects are Frontend, Backend, and Algorithm. Each sub-project has its own research and development stage.

1. All Team Members:
    a. Survey clients about their experience with creating a project for Senior Design. (What could go better?)
    b. Survey students about their experience with selecting team assignments.
    c. Survey faculty about what their experience has been like working with students in Senior Design. (for grading, being an advisor, etc)

    d. Decide what architecture we want to use: MVC, Event Driven, Microservices, etc?

2. Frontend:
    a. Frontend Research: decide …
        i. What framework is going to be used (React.js, AngularJS, etc)
    b. High level design of web application
        i. Wireframe what the web application will look like
        ii. Decide on a color palette
    c. Create Basic Web Pages (and UI)
        i. Login Page -- SSO
        ii. Profile Page
        iii. Home Dashboard
        iv. Student Preferences Form
        v. Projects Page (shows all available projects)
        vi. Client Project Proposal Page
        vii. Navigation for the Web Application (like a navigation bar)
        viii. Team Assignments Page
            1. All projects and their teams, or just the user's project/team
        ix. Admin Page
            1. Professors can approve proposed projects.
            2. Professors can upload Student information as CSV.
    d. Connect with server/backend
        i. Be able to communicate successfully with server
        ii. Be able to get/post data
        iii. Write all API calls needed for each page
        iv. Test calls and fix any not working correctly
    e. Testing
        i. Will be completed intermittently during development
        ii. Final iteration, have some users use the completed web application and give their feedback/report issues or bugs
        iii. User, interface, integration, regression, acceptance, and security testing
    f. Finalize pages
        i. Adjust according to user feedback -- fix bugs, finalize design of pages
3. Backend:
    a. Backend Research: decide …
        i. What framework is going to be used (Laravel, Django, etc)
        ii. What the tables will be and what information they will contain
    b. Set up Server/Get information on server
        i. Provided by ISU IT
    c. Database

        i. Create high level design documents:
            1. Design each table and their relations
                a. Students table
                b. Clients table
                c. Faculty table
                d. Projects table
                e. Project Assignment table (relationship?)
            2. Key elements to include:
                a. TERM element: year/semester (i.e. FALL2021)
    d. Set up Backend Framework
        i. Make tables and relationships
    e. Connect MySQL database to server/framework
        i. Test via Postman
    f. Connect with Frontend and perform User Tests
    g. Incorporate Algorithm with Backend (can be moved around)
        i. Conduct tests again and adjust backend based on results
4. Algorithm:
    a. Algorithm Research: after analysis of user surveys, decide …
        i. what type of algorithm would be best (weighted, banking system, etc.)
        ii. what exactly are all the inputs for the algorithm/what weights do they have
            1. Teammate Preferences
            2. Project preferences
            3. Project skill set requirements
            4. Meeting type preferences
            5. Student-proposed projects automatically get assigned that student
    b. Design/write algorithm
    c. Test algorithm
        i. Using a dummy dataset
        ii. Testing throughout the design

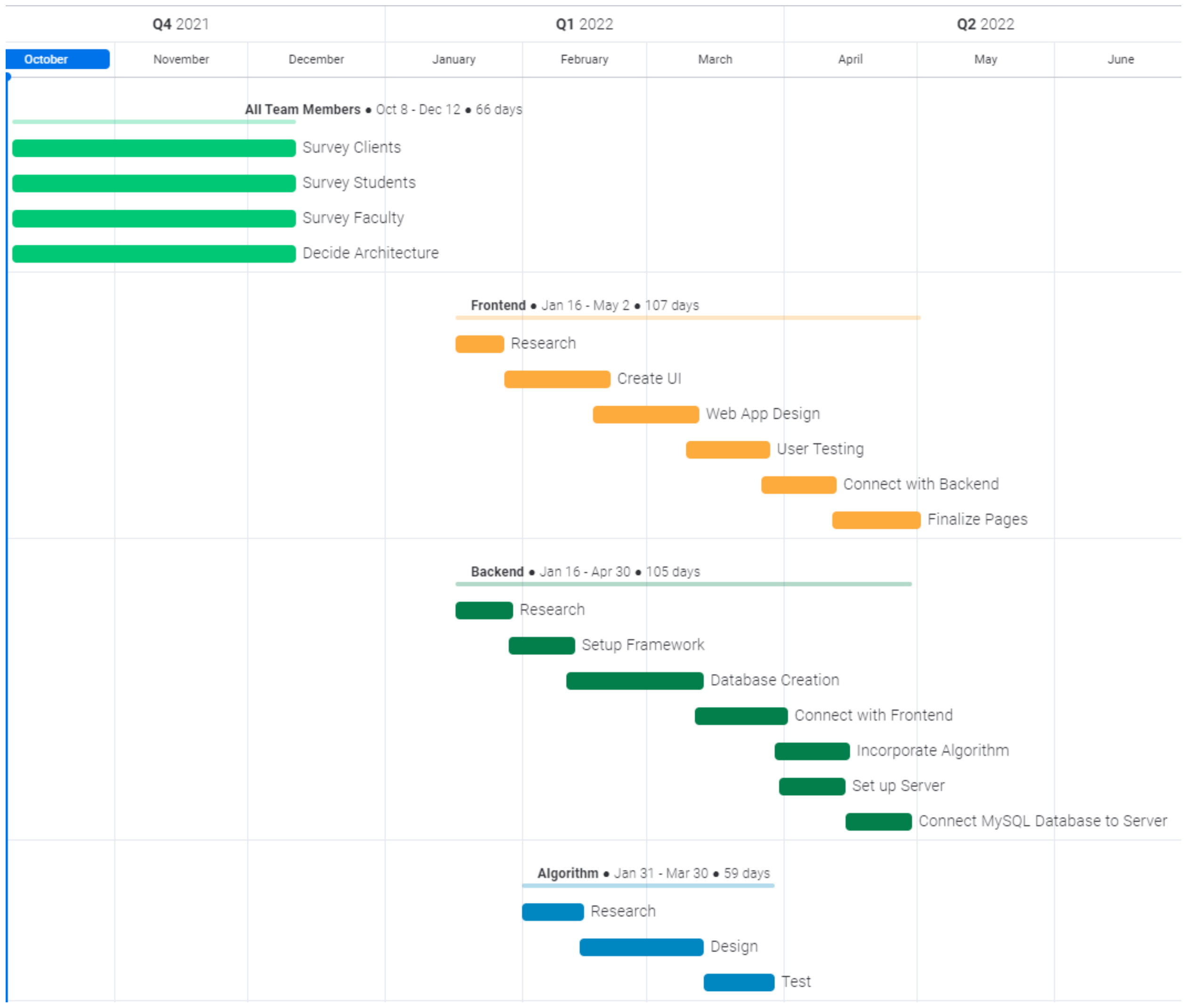
2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

1. **Web Application Skeleton**
    a. (Frontend) Homepage
    b. (Frontend) Student, client, and professor login
    c. (Backend) Web server setup
    d. (Backend) Database setup
    e. (Backend) Web server to Database communication
    f. (Frontend/Backend) Client to Web server communication
2. **Project Submittal - Client, Manual**

    a. (Frontend) Client manual form submission
    b. (Frontend) Client form view (edit their uploads)
    c. (Backend) Parse client input into database
    d. (Backend) Retrieve requested project information
3. **Project Selection - Student**
    a. (Frontend) Student form view (no edit, only view)
    b. (Frontend) Student preference selection
    c. (Backend) Parse student preference selection into database
4. **Project Review - Professor**
    a. (Frontend) Professor form view (edit all uploads)
5. **Project Team Generation - Algorithm**
    a. Research best placement practices
    b. Design algorithm
    c. Test implementation

2.4 Project Timeline/Schedule

Our project has two phases for each semester - the first semester is mostly research and surveying the different user groups while the second semester is where development will start. The Frontend and Backend teams will start development simultaneously while the Algorithm team will start a few weeks later.



2.5 Risks And Risk Management/Mitigation

Risk	Description	Category	Indicating Occurrence	Impact Rating	Occurrence Rating	Risk Exposure Rating	Response Strategy
Hardware Failure	Team Members are not able to finish tasks for the project due to hardware issues (i.e. broken pc) or the server fails due to external issues.	Technical Risk	Team member notifies the team that they are experiencing hardware issues. Team members cannot connect to the server.	High	40%	High - 60	Contact ISU Library and check out a laptop while troubleshooting issues. Contact IT for assistance with the server.
Issues with perfecting the algorithm	Conflicting wants/needs/specs from different users, optimization challenges; new variables emerging that require reformulation, etc.	Technical Risk	The Algorithm is not producing desired results and/or the client notifies issues they might have with Alg design.	High	60%	High	Rework/modify the algorithm design and assign additional team members to the algorithm implementation team.
Inaccurate Work Estimations	Tasks take longer to complete than initially estimated.	Scheduling Risk	Tasks are not completed as per the schedule.	Medium	60%	Moderate - 40	Team meeting to rework the schedule.
Change of Project Scope	Work involved with project changes from what was originally planned.	Scheduling Risk	Client or team decide on major change(s) in the project.	High	50%	High - 60	Adjust schedule accordingly to account for the new changes
Confined Schedules	There are very few times where the team can meet and work on major points.	Scheduling Risk	Team members notify the team they are unable to make it to major meetings.	Medium	60%	Moderate - 40	Team meeting to reschedule the major meetings with as many people as possible.
Unproductive Team Members	Members not contributing as much as they should.	People Risk	Members not completing their assigned tasks.	Medium	40%	Low - 30	Follow the procedure decided in the team contract.
Inexperienced Team Members	Members lacking knowledge/experience in certain aspects of project.	People Risk	Members share they don't have the knowledge to complete their task.	Low	70%	Low - 20	Resources- learning is a skill

2.6 Personnel Effort Requirements

TEAM 03			
Task #	**Name**	**Task's Time Approximation**	**Explanation**
1	**All Team Members/General Research**		
1.a	Survey clients	1 hour	Gain an understanding from those who have taken senior design in the past (i.e. What could go better?).
1.b	Survey students	3 hours	Get the student's viewpoint on selecting teams and projects.
1.c	Survey faculty	2 hours	Get the faculty's viewpoint on what can be done to make things easier (grading, advisors).
1.d	Determine Resources/architecture	1 hour	Decide which architectures would be best suited for the situation.
2	**Frontend**		
2.a	Frontend Research	5 hours	Determine the best framework for the frontend.
2.b	High level design of web app	2 hours	Wireframe the web app and make design decisions.
2.c	Create Basic Web Pages	2 hours	Create all the basic web pages required for the web app.
2.d	Connect with server/backend	10 hours	Ensure connection between the frontend and the backend.
2.e	Testing	13 hours	Have some users use the web application and give their feedback. Different testing throughout development.
2.f	Finalize pages	8 hours	Adjust the web pages according to the user feedback.
3	**Backend**		
3.a	Backend Research	3 hours	Determine the best framework for the backend.

3.b	Set up Server/Get information on server	2 hours	Coordinate with ISU IT to get a server running.
3.c	Database	3 hours	Create high level design documents for the database.
3.d	Set up Backend Framework	10 hours	Make the tables and classes based on architecture and database design.
3.e	Connect MySQL database to server/framework	3 hours	Incorporate MySQL connectivity into the server/framework.
3.f	Connect with Frontend and perform User Tests	20 hours	Make sure the Frontend is able to push data and request data from the Backend.
3.g	Incorporate Algorithm with Backend	5 hours	Conduct tests and adjust backend to ensure the algorithm data is put into the database.
4	**Algorithm**		
4.a	Algorithm Research	20 hours	Determine what type of algorithm would work best, all the inputs to be used and their weights.
4.b	Design/write algorithm	5 hours	Create the algorithm based on the information gathered in research.
4.c	Test algorithm	15 hours	Ensure that the algorithm is efficient enough to handle all of the team assignments in a reasonable time.

2.7 Other Resource Requirements

- Personal Computers, or some other machine, capable of running an Integrated Development Environment and connecting to the internet for each member of the group.
- Server capable of hosting a web application and its associated database management system
- Mock project information retrieved directly from the client in precisely the format they want to upload information.

3 Design

3.1 Design Context

3.1.1 Broader Context

The goal of this project is to create a web application for Senior Design 491 that will give students, instructors, and clients (businesses and faculty members) a user-friendly and positive experience. This project will give the students in Senior Design a higher chance of being assigned to the projects they chose with the students they wish to work with. It will also give the Senior Design Instructors a platform that allows them to manage Senior Design team assignments. Clients of Senior Design will have a platform to submit their project proposals and their projects will be assigned to the best fitting team of students.

This project will address the need for a better Iowa State University Senior Design project management system and team assignment algorithm. Currently, Senior Design projects and team assignments are done manually by the Senior Design instructors. The Senior Design Project Matching Client will resolve current issues with the team assignment algorithm and streamline the process.

Area	**Description**	**Examples**
Public health, safety, and welfare	How does your project affect the general well-being of various stakeholder groups? These groups may be direct users or may be indirectly affected (e.g., solution is implemented in their communities)	- Putting students on teams in which they are comfortable, valued, and challenged will allow them to feel safe - Creating teams that have similar scheduling availability will help reduce academic/time-management related stress

Global, cultural, and social	How well does your project reflect the values, practices, and aims of the cultural groups it affects? Groups may include but are not limited to specific communities, nations, professions, workplaces, and ethnic cultures.	- Our program will aim to produce diverse teams which will enable those teams to create products, processes, and programs that reflect the values and practices of a wide variety of people groups
Environmental	What environmental impact might your project have? This can include indirect effects, such as deforestation or unsustainable practices related to materials manufacture or procurement.	- Efficient code
Economic	What economic impact might your project have? This can include the financial viability of your product within your team or company, cost to consumers, or broader economic effects on communities, markets, nations, and other groups.	- Relatively cheap to maintain. Needs to last for future senior design classes.

3.1.2 User Needs

Students

- Students need to be able to easily access information on every available project proposal so they can make informed decisions on which project they wish to work on.
- Students need a way to submit their project and team preferences for Senior Design so they can be assigned to the best fitting team.
- Students need a way to efficiently access all Senior Design resources because they are currently not accessible in one place.

Clients

- Clients need to be able to submit their project proposals to the Senior Design Instructor so that it can be approved and assigned to a team.

- Clients need a way to see information about how Senior Design project proposals operate because it will improve the success of the project being approved and chosen by students.

Faculty

- Faculty need to be able to view and grade a team's work.
- Faculty need a way to communicate their schedules with students because the faculty panel will require sign up by students.

Project Advisors

- Project Advisors need a way to effectively communicate with their teams so that they can provide guidance on projects.
- Project Advisors need a way to track how many projects they are advising so they are not overworked and can provide quality feedback.

ISU IT Department

- ISU IT Department needs a platform to post information about any Senior Design resource they create because it centralizes the flow of information.

Senior Design Instructors

- Senior Design Instructors need a way to create the Senior Design teams with a wide range of requirements because they wish to exercise team formation best practices.
- Senior Design Instructors need a way to see all project proposals in order to approve or deny them because not all projects are feasible as Senior Design projects.

3.1.3 Prior Work/Solutions

From our research on team formation, the only existing product with comparable features to the Senior Design Project Matching Client is the Comprehensive Assessment of Team Member Effectiveness (CATME) SMARTER Teamwork system.

CATME is a web-based tool that has three main functions: team formation, peer evaluation, and rater practice. The feature that is similar to the design of the Senior Design Project Matching Client is the team formation tool, Team-Maker. Team-Maker allows the user to form teams based on selected criteria, some examples being: Schedule, Gender, Race/Ethnicity, Grade-Point Average (GPA), Prerequisite courses, Software skills, Discipline, etc. Users are also able to write their own questions and criteria. Team-Maker will then create teams from the student information and show how well each team fulfills the criteria.

The main feature of the Senior Design Project Matching Client that CATME's Team-Maker does not support is it allows students to request a preferred team member and team project. Team-Maker does not account for these requests and rather just forms the teams based on the required skill sets and user criteria. The Senior Design Project Matching Client will implement this additional feature.

3.1.4 Technical Complexity

- Interdependent Variables
  - Algorithm since it's going to be based on so many different requirements.
  - Server/Database/Algorithm/Web-Application Communication and Interaction
- Scripting Languages
  - We are not familiar with Laravel (php) and would need to learn it.
- Technical Interfaces
  - Okta SSO integration
  - Canvas compatibility or integration
- Several complex use-cases
  - Project proposals
    - Professors can approve/deny
  - Students surveys/forms
  - Team assignment
  - Faculty functionality
    - Panels
    - Grading

3.2 Design Exploration

3.2.1 Design Decisions

We will need to:

- Frontend - decide on a framework → Vue.js
- Backend - decide on a framework, server, database → Laravel
- Backend - decide on a software architecture
- Both - backend and frontend communication

3.2.2 Ideation

Among the key design decisions, we agreed that finding a viable framework for our project is an important and complex enough decision to warrant listing potential options:

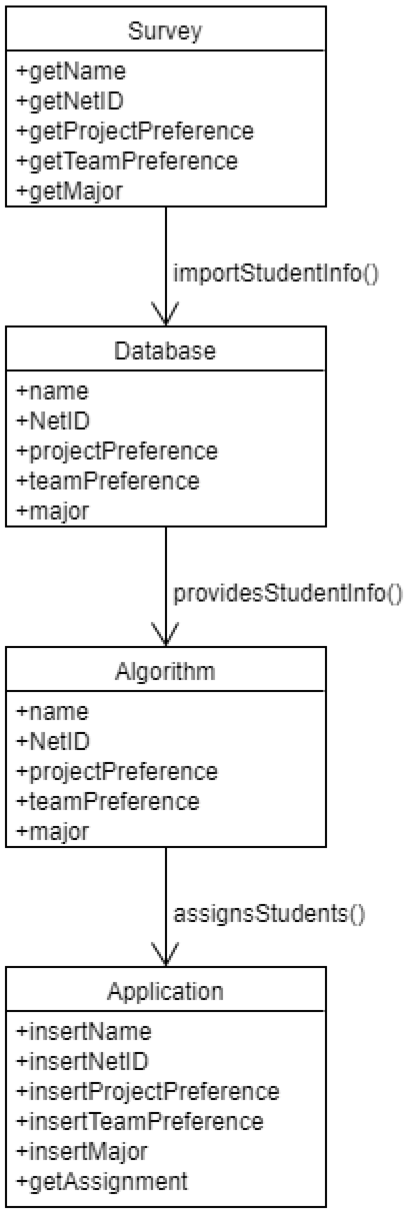
We used a pros/cons table (see below) to eventually decide on Vue.js as our frontend framework. Some factors that went into this decision were its compatibility with Laravel and its popularity with other developers (which will hopefully enable us to receive support from the developer community). Some alternative frameworks we considered were AngularJS, jQuery, EmberJS, React, Vue.js, Foundation, Svelte, and Semantic-UI.

3.2.3 Decision-Making and Trade-Off

Framework	Pros	Cons
Vue.js	● Compatible with Laravel ● Currently popular among developers (source) ● Easy to learn ● Flexible ● Simple syntax (similar to javascript) ● Thorough documentation	● Relatively new framework so fewer resources ● Not very stable with components ● Most plugins are difficult to use (language barrier)
EmberJS	● Extremely fast ● Good documentation	● Can be difficult to learn ● Heavy for small applications
React	● Efficient performance due to Virtual DOM ● Easy to learn ● Reusability of components ● Can handle high traffic on web page	● Frequent updates make the learning curve higher ● Hard to understand JSX when learning React ● Poor documentation ● Struggles with more complex applications
Angular	● Two-way binding built into the framework ● Components are very easy to manage and reuse ● Only framework based on Typescript ● Tons of support and resources	● Can be difficult to learn ● Optimization issues can occur with more complex applications
Svelte	● Faster than Angular and React ● Uses existing javascript libraries ● Scalable framework	● Currently unpopular ● Newest framework, so lacking in resources

3.3 Proposed Design

3.3.1 Design Visual and Description



Our project will first start off by surveying the students in the senior design class. Once we have all of their information, including their name, netID, preferences, and major, we can import that information into our database. After we have all the information in the database that we need, we can retrieve that information and use it for our algorithm. Our algorithm will then decide the assignments of the students based on their project and team preferences. The application will then be able to insert the information of the students and receive the assignments.

This application is supposed to work by inputting the students' name, netID, preferences, and major and then outputting the team assignment based on the students' preferences.

The requirements are well answered since the program will take in the preferences of the students and use them to determine what project the students are assigned to and what team members they have.

3.3.3 Areas of Concern and Development

The main problem with this would be revolving around the assignment algorithm. The algorithm is supposed to assign students to groups based on their preferences, but that's not always possible. Sometimes the group may be already filled up, or the student lacks the experience that is recommended by the project. So it will be nearly impossible to try to fit every student's wants for the project, and there is a good chance that some students won't get any of their preferences fulfilled. We would like to assign students to their preferred project or with their preferred group, so if that's not possible, we need to have a way to assign them to another project.

There are multiple ways to address the solution. One is that we would just randomly assign the student to an open group that requires another team member. The problem with this, is the student may not be happy with their group or project at all, so another viable solution would be to assign the student to a similar project or one that they have experience in.

3.4 Technology Considerations

This section highlights the strengths, weaknesses, and trade-offs made in technology available.

3.4.1 Ubuntu Server vs. Microsoft Server

Ubuntu Linux has built in integrations with many open source software options. Lastly, Ubuntu Linux is a lightweight operating system, requiring fewer resources than other solutions.

3.4.2 Laravel

It is the number one PHP framework, meaning that Laravel is already a framework with which most web developers are familiar with. It also has a range of built-in templates which makes development simpler and easier for developers. Other key strong points are its ability to handle traffic, security, and flexibility.

3.5 Design Analysis

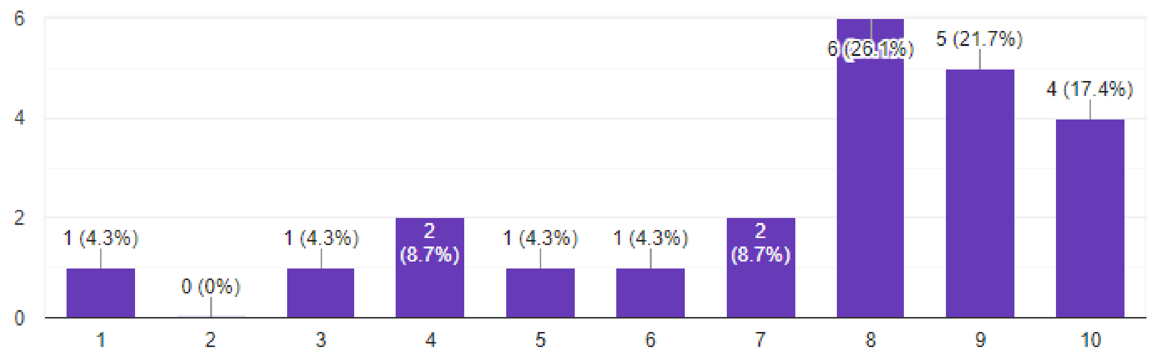
So far, our proposed design has been working as expected. We have received numerous surveys from students, professors, and clients, and have already organized the information for when we start working on the algorithm.

Below are the current results from our survey (Appendix A-2, pg 46). The ongoing statistics will be updated at https://sdmay22-03.sd.ece.iastate.edu/docs.html

Student Responses

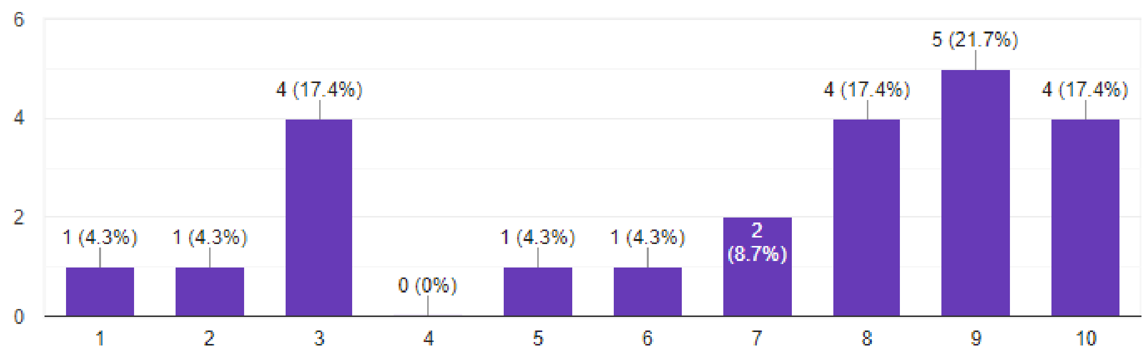
How satisfied are you with your team assignment? Rate your experience 1 - 10:

23 responses

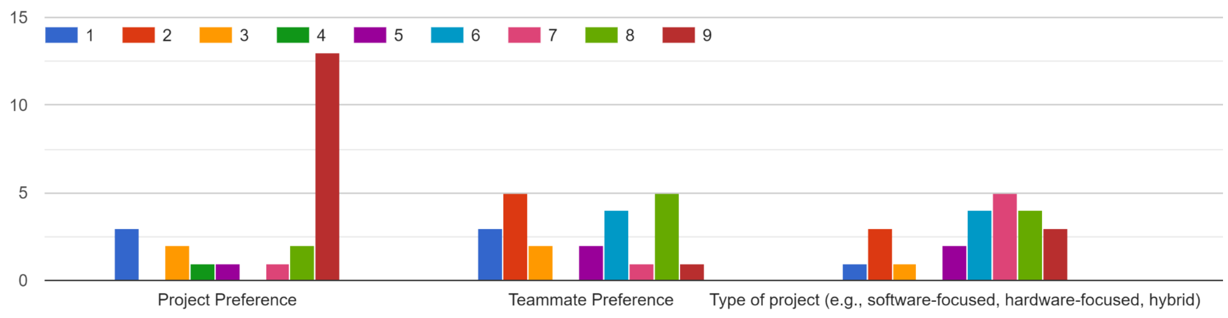


How satisfied are you with your project assignment?

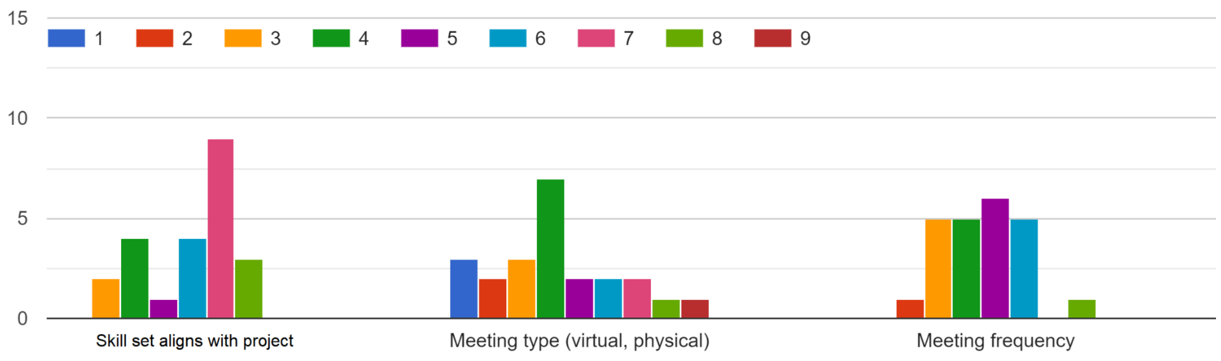
23 responses



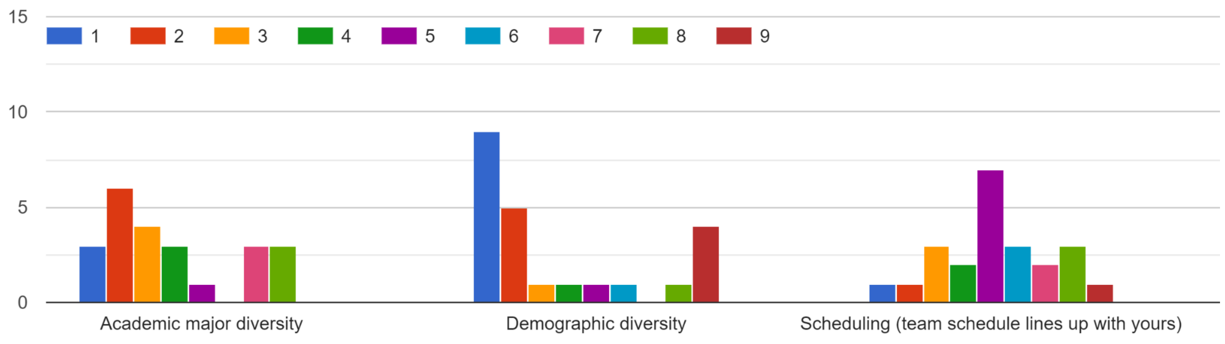
Rank the following preferences/attributes of team assignment according to their importance to you.



Rank the following preferences/attributes of team assignment according to their importance to you.

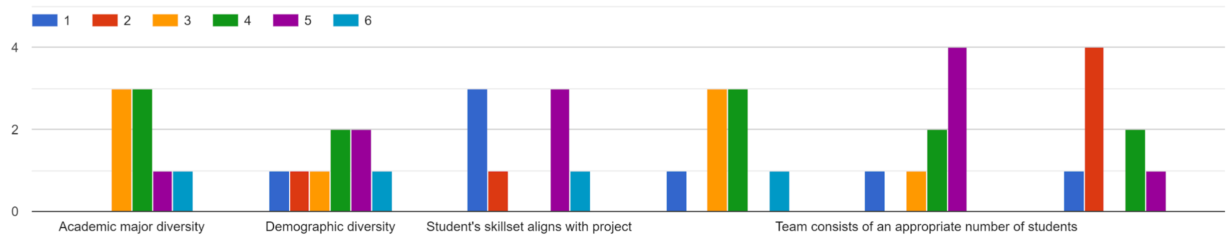


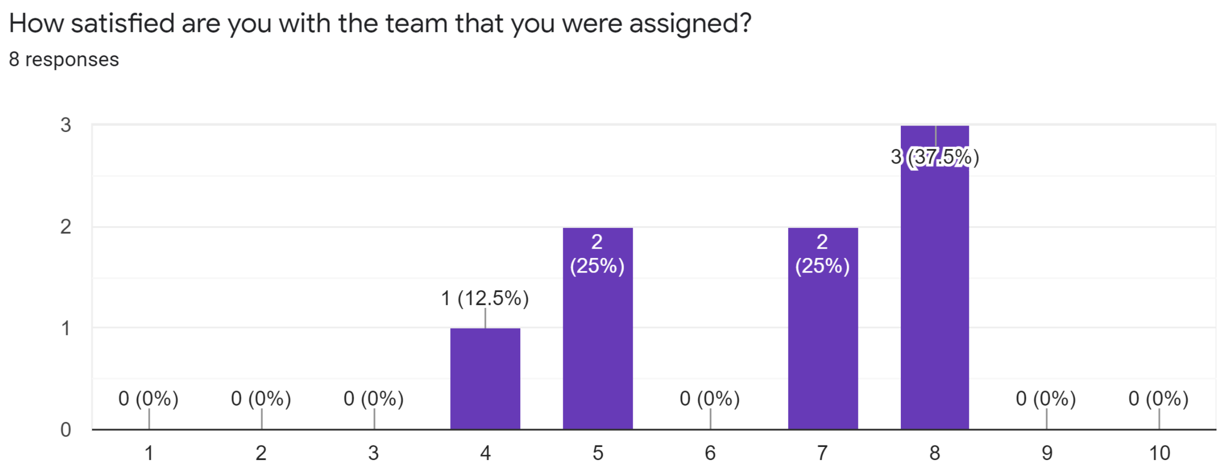
Rank the following preferences/attributes of team assignment according to their importance to you.



Faculty Responses

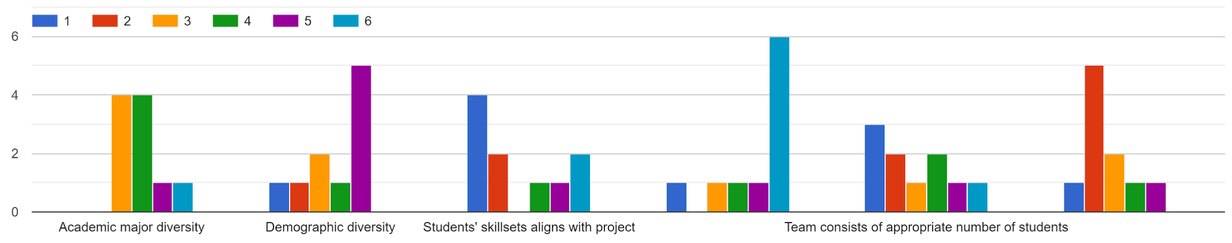
Rank the following preferences/attributes of team assignment according to their importance to you. 1 being the most important, 6 being the least important. (Please put in ranked order, i.e. each attribute receives a different score/ranking).





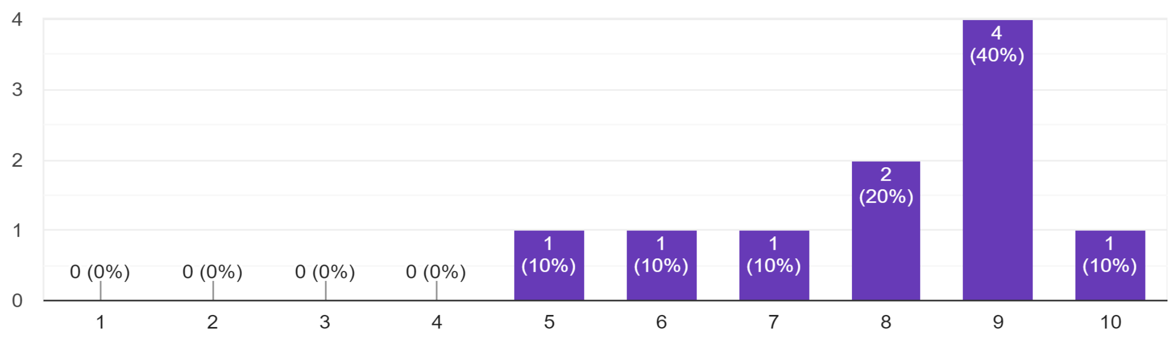
Client Responses

Rank the following preferences/attributes of team assignment according to their importance to you. 1 being the most important, 6 being the least important. (Please put in ranked order, i.e. each attribute receives a different score/ranking).



How satisfied are you with the team that you were assigned?

10 responses

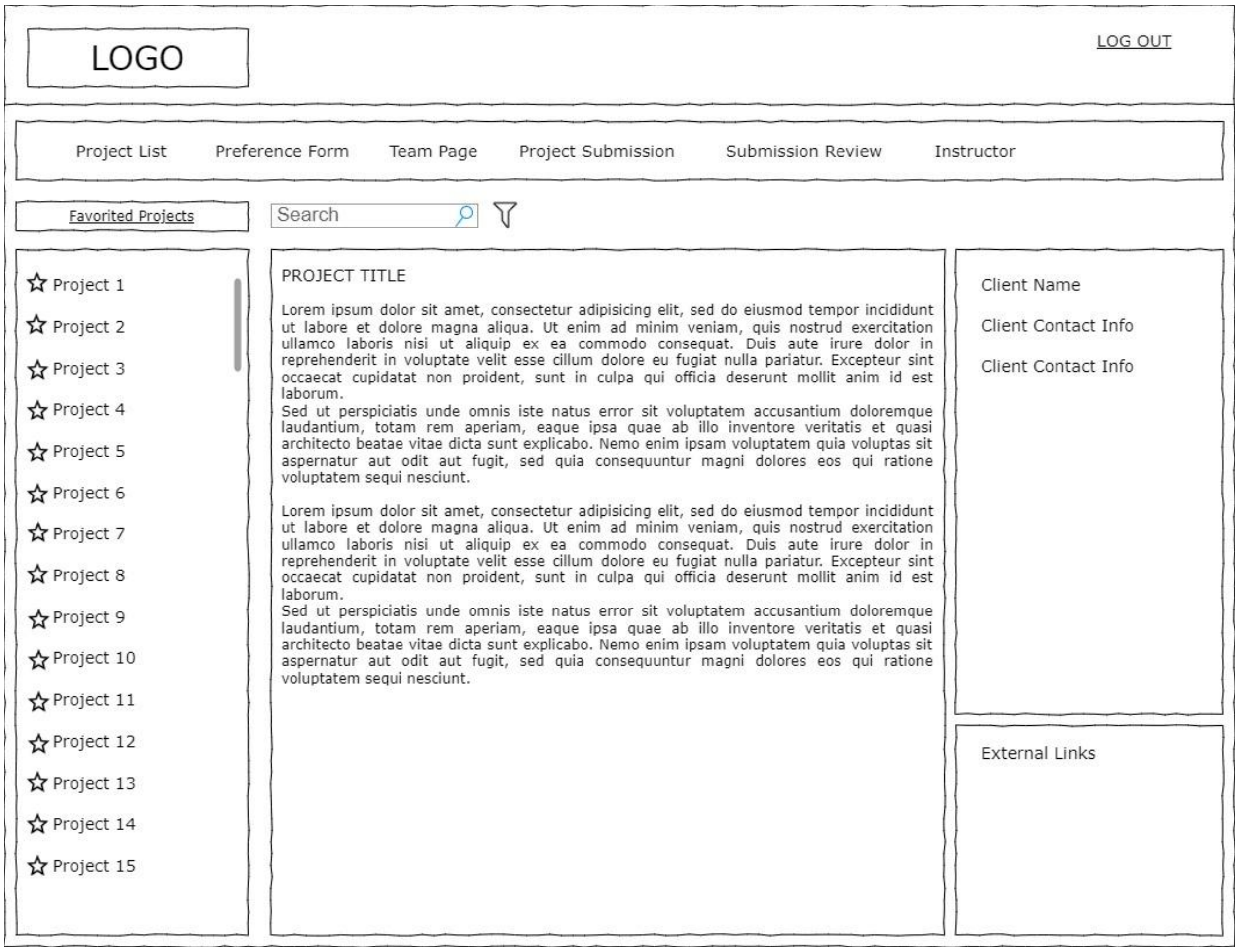


3.6 Design Plan

Our design plan consists of three sections: frontend, backend, algorithm. Below are the design diagrams that outline each of the three project sections.

3.6.1 Frontend

UI Diagram



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

The above diagram is a wireframe of the Senior Design Team Matching web application. It is currently on the *Project List* page.

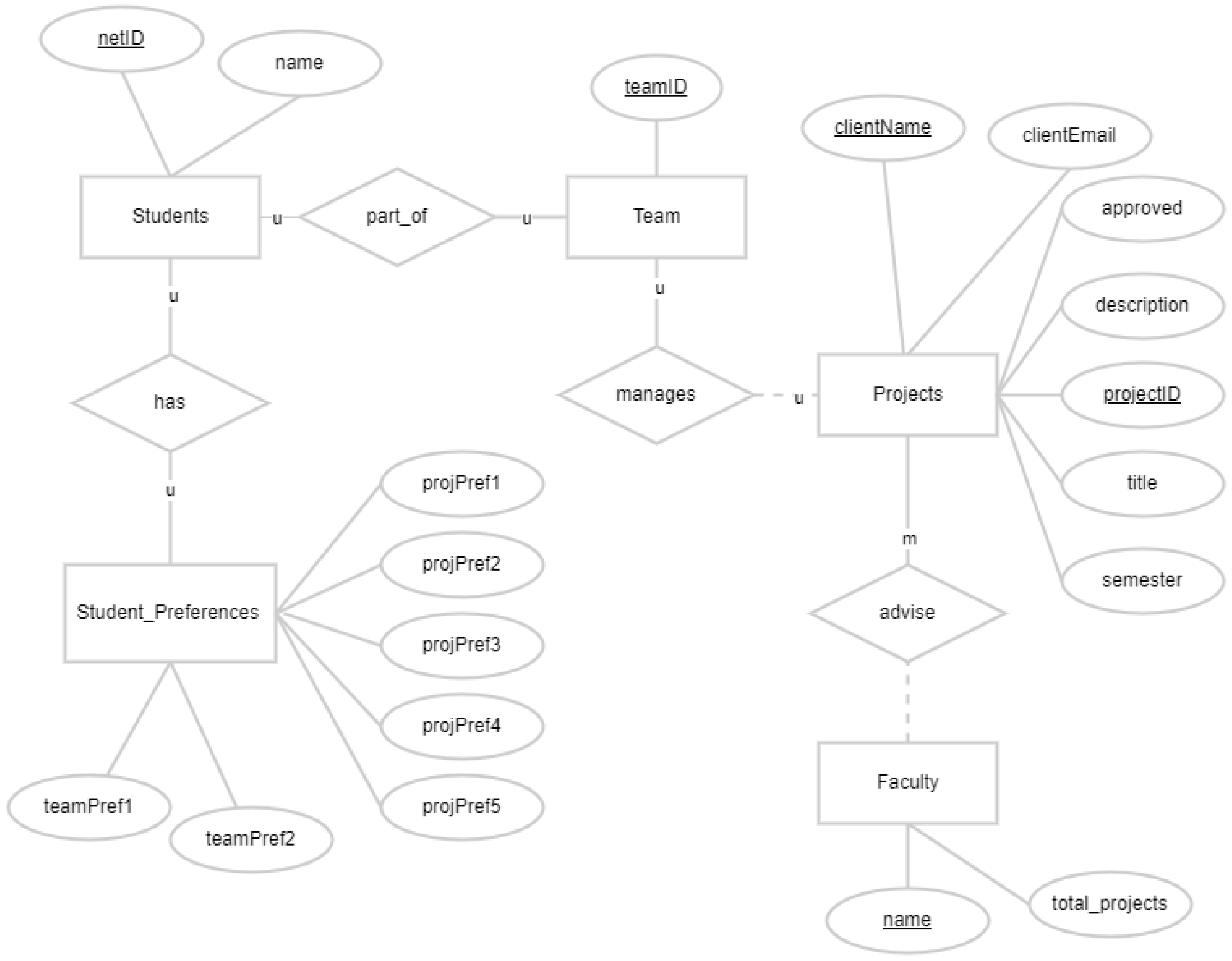
In order to keep the web application simple but functional, all users will see the same view except for the Instructors who are able to see more pages and Clients who will see a limited number of pages. Specifically, only Instructors will be able to see the *Submission Review* and *Instructor* pages. Only Clients will be able to access the Project Submission page, and they will not be able to access the Preference Form or the Instructor pages.

Overview of each webpage:

- Project List - It shows the list of approved projects for the current semester. Each project will contain the details submitted by the client (including a description, expertise needed/recommended, and the majors required) and the client's contact information. Users are able to favorite projects and search/filter for projects.
- Preference Form - Contains the team and project preferences form for each student to fill out and submit.
- Team Page - Will display each students' team and project assignment after the algorithm has been run. Will also contain the links for each team's web page and GitHub.
- Project Submission - Will display the project submission form for each client to fill out and submit.
- Submission Review - Will display each submitted project for the instructors to review and approve or deny.
- Instructor - Will contain the settings and mechanism for running the algorithm and display the results after running.

3.6.2 Backend

Entity Relationship (ER) Diagram



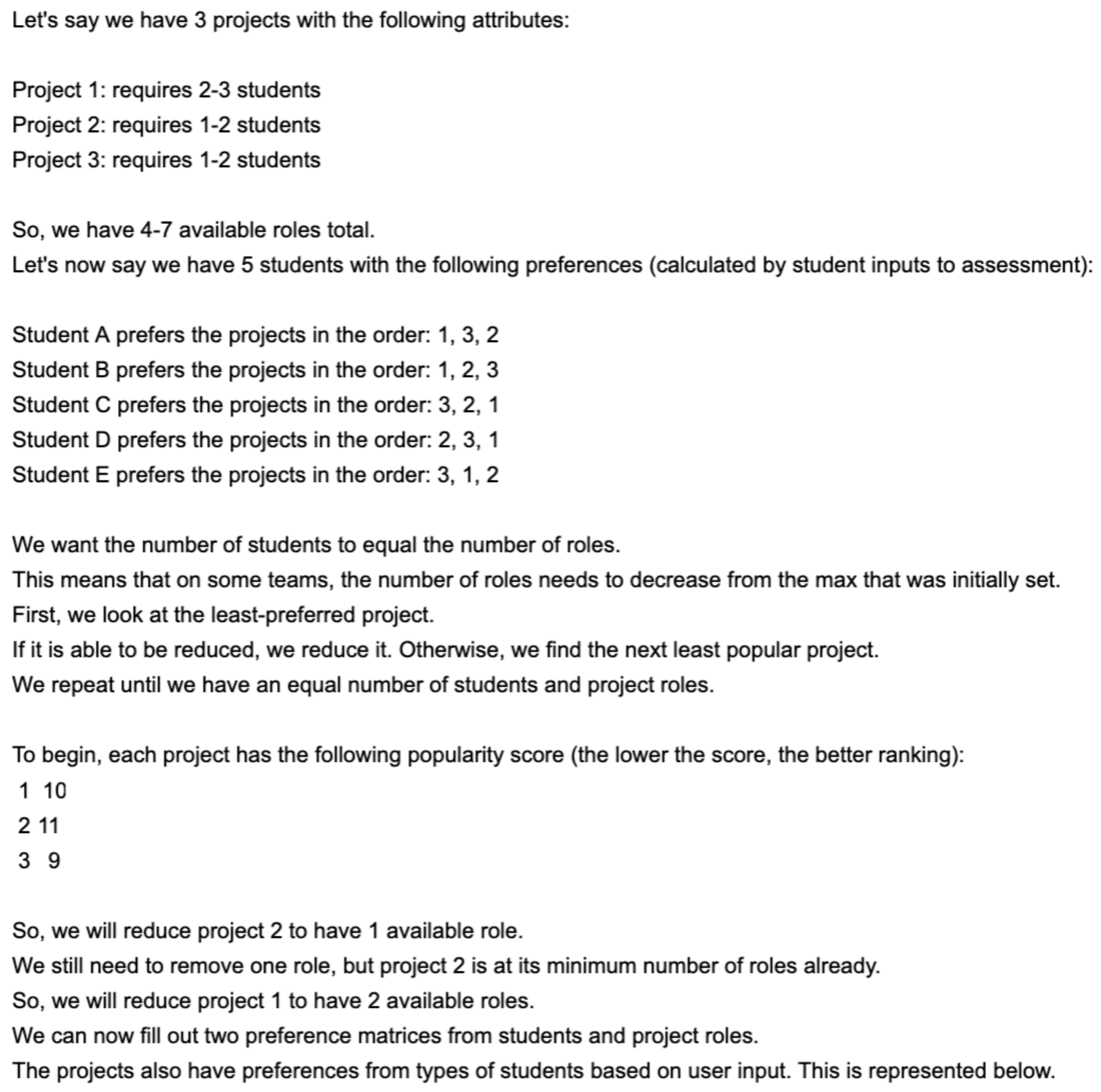
The above figure is an ER diagram of the database. The database will consist of four entities: Students, Projects, Faculty, Team and Student Preferences. Each Student will be identified by their Net ID and will have a one-to-one relationship with Student Preferences. Student Preferences will contain the results of the student preferences form (project preferences, teammate preferences, etc). Each student will be a part of a Team which manages a Project. Each Project will have a Faculty as an advisor.

3.6.3 Algorithm

The basic strategy for student/project assignment will be to:

1. Produce ranked project role preferences (based on each student's input)
2. Produce ranked student preferences (based on each project/client's input)
3. Adjust roles so that # of students = # of total roles seeking a student to fill it
4. Perform the Gale-Shapley algorithm on these preference matrices in order to produce stable matches

A basic example is outlined below:



Let's say we have 3 projects with the following attributes:

Project 1: requires 2-3 students
Project 2: requires 1-2 students
Project 3: requires 1-2 students

So, we have 4-7 available roles total.
Let's now say we have 5 students with the following preferences (calculated by student inputs to assessment):

Student A prefers the projects in the order: 1, 3, 2
Student B prefers the projects in the order: 1, 2, 3
Student C prefers the projects in the order: 3, 2, 1
Student D prefers the projects in the order: 2, 3, 1
Student E prefers the projects in the order: 3, 1, 2

We want the number of students to equal the number of roles.
This means that on some teams, the number of roles needs to decrease from the max that was initially set.
First, we look at the least-preferred project.
If it is able to be reduced, we reduce it. Otherwise, we find the next least popular project.
We repeat until we have an equal number of students and project roles.

To begin, each project has the following popularity score (the lower the score, the better ranking):

1	10
2	11
3	9

So, we will reduce project 2 to have 1 available role.
We still need to remove one role, but project 2 is at its minimum number of roles already.
So, we will reduce project 1 to have 2 available roles.
We can now fill out two preference matrices from students and project roles.
The projects also have preferences from types of students based on user input. This is represented below.

Students

A	L	M	O	P	N
B	L	M	N	O	P
C	O	P	N	L	M
D	N	O	P	L	M
E	O	P	L	M	N

Project Roles

L	D	B	E	C	A
M	B	A	D	C	E
N	A	C	E	D	B
O	D	A	C	B	E
P	B	E	A	C	D

where:
L is a role on project 1
M is a role on project 1
N is a role on project 2
O is a role on project 3
P is a role on project 3

From here, we can treat the problem similarly to the stable marraige problem and
solve it using the Gale-Shapley algorithm to produce stable matches for each student/role relationship.

For this (overly simplified) example, the algorithm would produce the following matches:

A:M (project 1)
B:L (project 1)
C:O (project 3)
D:N (project 2)
E:P (project 3)

The complexity in this (and any) strategy will be in producing the proper preference matrices for both students and projects.

4 Testing

The testing strategy we will utilize for the Senior Design Project Matching Client is to test early and often. We will use a variety of testing methods such as: Unit Testing, Interface Testing, Integration Testing, System Testing, Regression Testing, Acceptance Testing, and Security Testing. By testing early and often with many testing methodologies, we can be assured that we are able to fix any bugs in advance.

A few unique challenges we have to consider are the algorithm, the virtual software testing, and the integration testing. For the team formation algorithm, we will have to consider what we are willing to accept as valid results for the algorithm and how we verify that those are satisfactory. Another challenge to our project is that we are working exclusively with software. Therefore, all of our testing will be done virtually. We also have three large parts for our project which we have to take into consideration when testing if they work together.

4.1 Unit Testing

The units being tested would be the functions/methods of any part of the code (primarily the algorithm implementation). Any helper function/method, such as how to determine preference priority, the actual group assignment, or anything else that might be helpful to have unit testing for. The idea behind this would be to test each function/method by putting in several different inputs and getting the expected result. There are many tools to help us perform unit testing. If we use Java for our algorithm, we would most likely use JUnit testing and maybe Mockito testing with it. If we use PHP, we could use PHPUnit for our unit testing as well.

4.2 Interface Testing

Our project will be using a web-based application to interact with the database and backend. As such, all aspects of this application will be thoroughly tested. The two main factors that we will consider when testing the application are functionality and aesthetics.

To test that the application code works as expected, we will use a combination of manual and automated testing. The manual testing will consist of developers testing-as-they-go when designing new features for the UI. The automated testing will be put in place so that we can quickly be sure that if we make a change to one part of the application, the login page does not break, for example. We plan to use tools like Selenium IDE for record and playback testing. We may also utilize a tool such as Perfecto Scriptless, a codeless automation testing tool.

For testing user experience, we will have trial users periodically test the app/UI throughout the development process to obtain continuous feedback. The purpose of this is twofold, as it allows us to not only make sure that UI is stable, but also test the application in a more realistic scenario. Feedback from the testers will be crucial to ensure that the application is something that users will want to use. By using tools like Git, we will be able to set up demo branches that we can give to users so they can test the current version before it is released.

4.3 Integration Testing

Our project has three critical integration paths: Database to Server Communication, Web Application to Database/Server Communication (Frontend to Backend), and Algorithm integration with Backend.

4.3.1 Database to Server Communication

This is a simple but essential step for our web application because for a complete web application the database must successfully communicate with the server. The main part of this is establishing a connection to the server and being able to save our data there. We will test this critical integration path by making a connection to the server from the database using the command line or Postman. Then, using dummy data, we will test if we have correctly set up the database and server by passing data through and saving it.

4.3.2 Web Application to Database/Server Communication (Frontend to Backend)

Our web application must be able to communicate with the backend for the project. This means all of the requests from the web application must be handled successfully in the backend. The web application will be making API calls to the server to retrieve and send data as needed. As such, this step in the development process will be made after the Database to Server communication is established and tested. We will be testing if the web application displays correct information from each request it makes. We will then test if the database and server are able to correctly handle the requests made by the web application.

4.3.3 Algorithm Integration with Backend

This will be an ongoing critical integration path for our project. The algorithm for team assignments is essential to the project but will probably be changed a lot throughout the course of development. The main objective of this integration is to provide the algorithm with the necessary information from the backend. To test this, we will verify that the correct data is being provided by the backend when it is requested by the algorithm. We will also test if the results of the algorithm are being sent to the correct location in the backend with the correct format.

4.4 System Testing

As system testing is done only after all sub-testing is completed on the project (unit testing, interface testing, and integration testing), we'll make use of those parts to complete a full test of the entire project. This testing will be completed thoroughly, and ensure each requirement outlined in the previous section of our paper is met. Information will be tester-generated dummy data. This will include:

- Project submission via Professor, Client view is input then seen in the database
- Project editing via Professor, Client view is made then seen in the database
- Project viewing via Professor, Client, Student view can be done any time and view up-to-date project information
- Project selection via Student view done and recorded in database
- Demo-run of algorithm works consistently regardless of dummy input

4.5 Regression Testing

Retest the previous functionalities using the unit, integration, and interface tests. This is to ensure the old functionalities that have already been implemented are still working correctly. All tests that were successful before the new addition need to be passed after the new code is added. This means going back and retesting code after every new feature/functionality added. If something changes on a fundamental level, we will also go back through our tests to make sure that they are compatible with our changes.

The most important part of our project that needs to work is integrating our algorithm with the database and the database with the application. If something goes wrong with the algorithm or the database, the functionality of the application will not matter. Therefore, it is imperative that our integration testing works properly, so that if there is a problem, we can immediately fix it.

4.6 Acceptance Testing

For our acceptance testing, we will base the tests off of our use cases to ensure that our functional design requirements are being met. We will mark the use case with "passed acceptance testing" when we have verified that the use case requirements have been met.

Some examples of the use cases we will be testing:

Senior Design Instructors will be able to…
- mass-upload student information as a csv file
- add, edit, and remove student information and client projects
- approve/refuse proposed projects

Students will be able to…

- view all approved projects and associated information
- submit project/team preferences

At each of our major milestones in the project, our client will be brought in for acceptance testing. We will then take their feedback and change the project accordingly.

4.7 Security Testing

For our project, security testing will be essential to ensuring the project is secure and no privacy laws are violated. The main form of security testing we will do is web page security testing. We will test the integrity of the web application using various testing methods to make sure that no malicious users can interfere with the web application, steal data, or access data they are not authorized to see. Below are the security vulnerabilities we will be addressing with our testing.

4.7.1 SQL Injection & Cross-Site Scripting (XSS)

SQL injections are attacks upon the system from the user entering SQL statements as their input; which the web application then executes. XSS are attacks that use HTML or script (e.g. JavaScript, PHP, etc) to steal data from the web application or other malicious actions.

To mitigate this potential security breach, we will program tests for SQL statements in the user input to make sure only acceptable inputs are allowed. Additionally, no HTML or scripts will be accepted as user input.

We plan to test SQL injection by manually entering SQL statements in the web application or using external testing software such as Acunetix or Netsparker. An example of an SQL statement we could inject is: “select uid, phone_number from students where id=1 or 1=1” (which would always return true since 1=1 is true). We plan to test XSS in a similar fashion - testing simple script or HTML statements as user inputs.

4.7.2 Brute Force Attacks

Brute Force Attacks are when a malicious software attempts to get into an account by cycling through every possible password. This can also be when the malicious software attempts to break through any encryption used for sensitive data by cycling through all possible algorithms.

To mitigate this potential security risk, we will implement some form of account suspension after a number of login attempts by the same user (e.g. after 3 failed login attempts, ‘lock’ the account for 30 mins).

This can be tested manually by the team by using an incorrect password multiple times for an account.

4.7.3 URL Manipulation & Role Management

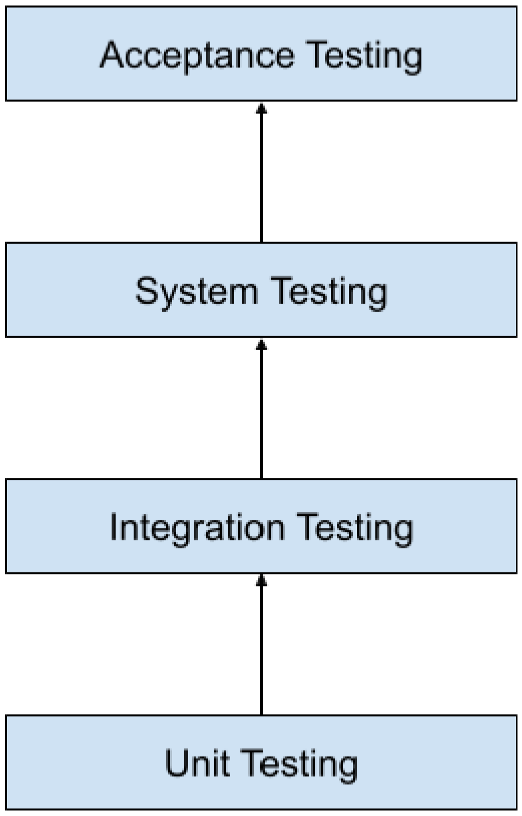
By modifying certain parts of a URL, a hacker may be able to access web pages they are not supposed to have access to. The hacker could access protected pages or force errors that reveal important information. URL Management is also an important part of testing Role Management which details what web pages users are able to access.

To prevent this security risk from occurring, we will perform user permission checks before displaying each page. If the current user does not have access to the page they are attempting to view, the application will display an error page instead.

We plan to test this by logging in as different types of users and manually editing the URLs to mimic an attempt to access unauthorized pages.

4.8 Results

As of now, we don't have any results from testing, but when we do, we will ensure those results are compliant with the requirements by manually going through each test and figuring out what the results should be based on the requirements. All of the testing will be connected together as this diagram shows:



So, the testing should be done in specific order as shown above to make sure all of the code is correct.

Examples of successful tests would be:

- Unit tests passing all test cases (unit test must be thorough → test edge cases etc.)
- Successful integration/connection between the different parts of the project (FrontEnd, BackEnd, Alg.)
- When adding a new functionality the previous unit, interface, integration tests etc. Must be rerun and get the same passing results.
- Final system must meet all the functionality requirements listed out in the requirements document

5 Professionalism

5.1 Areas of Responsibility

Area of responsibility	Definition	NSPE Canon	ACM
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; avoid deceptive acts.	An individual who is assigned a task or function is considered the responsible person for that role.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	Most computing professionals work for employers. The employment relationship is contractual: The professional promises to work for the employer in return for a salary and benefits.
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; avoid deceptive acts.	When assessing the capabilities and risks of computer systems, the professional must be candid: The professional must report all relevant findings honestly and accurately.
Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	When designing a new computer system, the professional must consider not only the specifications of the client but also how the system might affect the quality of life of users and others.
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	The professional does not have the right to profit from independent sale or use of this intellectual property, including software developed with the employer's resources.
Sustainability	Protect the environment		

	and natural resources locally and globally.		
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	When designing a new computer system, the professional must consider not only the specifications of the client but also how the system might affect the quality of life of users and others.

5.2 Project Specific Professional Responsibility Areas

Area of responsibility	Definition	Our Project	Current Team Performance
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Applicable - We need to be working hard in a timely manner, producing products of high quality and integrity.	Medium - All of our work is high quality and we have not missed any deadline. However, we need to practice better professional competence.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Applicable - Our software will be free to use for our direct clients (Iowa State professors) and will be offered at a reasonable price to other potential users.	High - Since this is a purely software project, we don’t have a budget and will not be purchasing anything.
Communication Honesty	Report work truthfully, without deception, and understandably to stakeholders.	Applicable - We will strive to be transparent with stakeholders about our processes and decisions.	High - We are honest and maintain two-way communication with our client through all stages of development.
Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Applicable -Allows users to preference meeting type to accommodate COVID-19 concerns/comfort levels Makes senior design processes simpler, reducing stress of users and increasing user satisfaction	Medium- While it is not the main focus of the project these points should always be kept in mind.
Property Ownership	Respect property, ideas, and information of clients and others.	Applicable - Given the type of system we are designing, we need to be respectful and careful when dealing with intellectual property, personal information, etc.	Medium - We will not sell information or data acquired from any clients, students, or faculty that utilize our software.

Sustainability	Protect the environment and natural resources locally and globally.	Applicable - Ensures efficient use of resources (i.e. server)	Medium- Our software project doesn't impact the env. As much as hardware would.
Social Responsibility	Produce products and services that benefit society and communities.	Applicable - We have a responsibility to produce teams which can be the most successful at serving their communities and society at large; this includes creating teams that are diverse and unified behind a common goal themselves.	Medium - One of the main goals of our software is to benefit our community. This is a social responsibility we are addressing in our design. We have yet to implement these ideas as we are still in the design stage of this project.

5.3 Most Applicable Professional Responsibility Area

Responsibility - Communication Honesty:

- Why is this important?
    - Communication Honesty is important to our project because it is affected by multiple stakeholders who all have their own requirements for the software. We are responsible for communicating with all of them and addressing any of their concerns.
- How have we achieved this?
    - We're honest when unsure about what the client wants/is asking for
    - At each client meeting, we report what work has been completed and ask for feedback
- What are the specific impacts to the project that you have observed?
    - The project was split up into two separate teams when originally it was supposed to be one large team working together.

6 Appendix

A-1 Team Contract

Team Members: Ally Finger, Brady Loew, Cheyenne Carlson, Connor LaFerle, Jade Yang, Logan Christianson**,** Sierra Jones

Required Skill Sets for Your Project:

- Programming
- Algorithms
- Website Management
- Database Management (MySql)

Skill Sets Covered by the Team:

- Programming languages (C, C++, Java)
- Database Knowledge
- Algorithm Knowledge
- Data Structures
- Computer and Server management
- UI/UX design

Project Management Style Adopted by the Team:

Agile development: 2-week-long sprints, project milestones, weekly meetings to go over progress; may split teams up into smaller squads; use Github task board.

Team Name: **SD MAY 22-03**
Team Members:
1) Logan Christianson_______________ 2) Connor LaFerle__________________
3) Jade Yang______________________ 4) Cheyenne Carlson________________
5) Ally Finger______________________ 6) Sierra Jones_____________________
7) Brady Loew_____________________

Team Procedures

1. **Day, time, and location (face-to-face or virtual) for regular team meetings:**
    a. TA: 2:30 pm Sunday
    b. Team: 3:00 pm Sunday
    c. Client meeting: 2:30 pm Friday
    d. All virtual
2. **Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):**
    a. Discord
    b. Email
3. **Decision-making policy (e.g., consensus, majority vote):**
    a. Consensus if possible
    b. Team vote (leader gets final say in case of tie)
4. **Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):**
    a. Meeting times and notes through the discord channel

Participation Expectations

1. Attend team meetings on time, but let the team know if you can't make it and review meeting notes for meeting summary.
2. Ask for help if you need it for an assignment (i.e. time constraints, just need more help researching, etc)
3. Ask for clarification if you are confused about anything.
4. Respond to messages within 24 hours.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
    a. Team Leader and Organizer: Sierra Jones
    b. Team coordinator: Jade Yang
    c. UI/UX Lead: Connor LaFerle
    d. Database/Server Manager: Brady Loew
    e. Algorithm/Research Lead: Logan Christianson
    f. Client Interaction: Ally Finger
    g. Head of QA/Testing: Cheyenne Carlson
2. Strategies for supporting and guiding the work of all team members:

    a. Weekly meetings to keep everyone up to date on issues/progress
    b. Split the team into smaller groups where they can focus on their strengths
3. Strategies for recognizing the contributions of all team members:
    a. Give shout-outs at weekly meetings
    b. Thank members for work completed, especially when it goes above and beyond what is expected

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.
    a. Ally Finger: experience with python, java, c/c++, AWS, *some* web development, *some* algorithm work
    b. Brady Loew: Experience with C/C++, Java, SQL, backend programming, and database creations, only backend programmer for CS 309
    c. Cheyenne Carlson: Web development experience -- mostly frontend, frontend development for mobile applications,
    d. Connor LaFerle: Full-stack development for web applications, UI and UX experience
    e. Jade Yang: Backend development experience for web/mobile applications, some HTML/CSS/JavaScript experience
    f. Logan Christianson: Made game mods, experience in Lua, Unreal Engine 4 experience
    g. Sierra Jones: CPRE, FrontEnd dev, android studio and UI design, Info Security Basics
2. Strategies for encouraging and support contributions and ideas from all team members:
    a. Make sure to give every person an opportunity to share his/her opinion
    b. Ask people who are usually quieter for their input
    c. Take surveys/polls
    d. When brainstorming, there are no “bad” ideas
3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)
    a. During a weekly team meeting, have a designated time for people to share how the current team climate is affecting their ability to learn/contribute
    b. Have one on one conversation with team leader for guidance if necessary
    c. Don’t be defensive - look for solutions

Goal-Setting, Planning, and Execution

1. Team goals for this semester:
    a. Set ourselves up for easy development next semester
    b. Develop into team roles
    c. Produce detailed and succinct architecture for the project

    d. Develop strong communication with team members, client, and users
2. Strategies for planning and assigning individual and team work:
    a. Create milestones with deadlines and individual tasks within each milestone -> assign each milestone to a squad
    b. Use Github task board to assign tasks to individuals
    c. Go over progress at weekly meetings
3. Strategies for keeping on task:
    a. Assign deadlines for individual tasks
    b. Assign deadlines for squad milestones

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
    a. Discuss respectfully as a team and try to think of solutions that work for everyone
    b. Compromise where appropriate, but stick to the commitments made in this contract
2. What will your team do if the infractions continue?
    a. Team leader may have one on one conversation with person
    b. TA may be involved
    c. Professor, as a last resort, will be involved

---

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
b) *I understand that I am obligated to abide by these terms and conditions.*
c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*

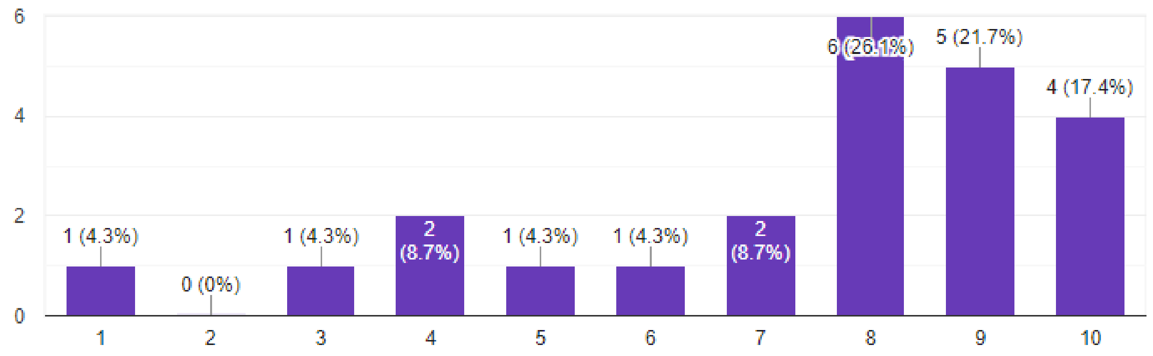
1) Ally Finger		DATE 9/22/21
2) Connor LaFerle	[redacted]	DATE 9/22/21
3) Sierra Jones	[redacted]	DATE 9/22/21
4) Brady Loew	[redacted]	DATE 9/22/21
5) Jade Yang	[redacted]	DATE 9/22/21
6) Cheyenne Carlson	[redacted]	DATE 9/22/21
7) Logan Christianson		DATE 9/22/21

A-2 Survey Responses

Student Responses

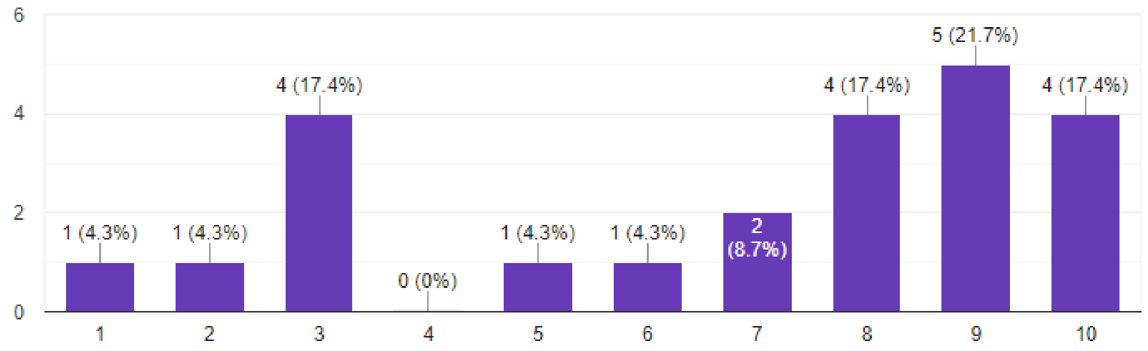
How satisfied are you with your team assignment? Rate your experience 1 - 10:

23 responses

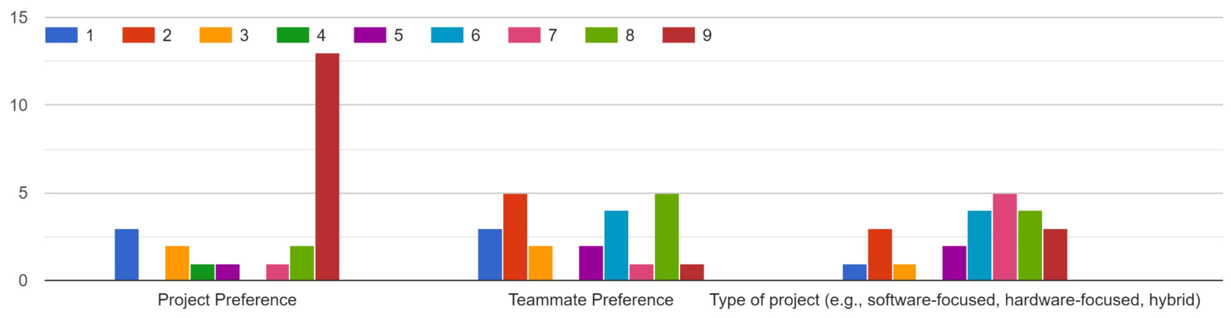


How satisfied are you with your project assignment?

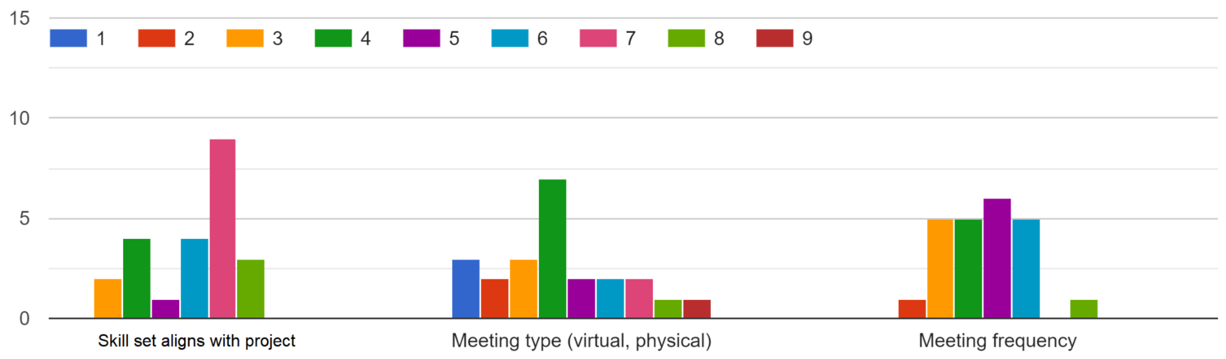
23 responses



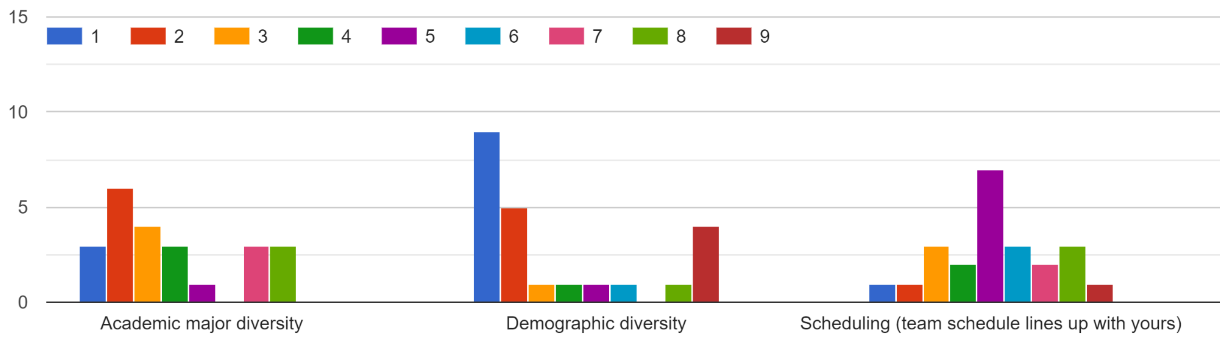
Rank the following preferences/attributes of team assignment according to their importance to you.



Rank the following preferences/attributes of team assignment according to their importance to you.

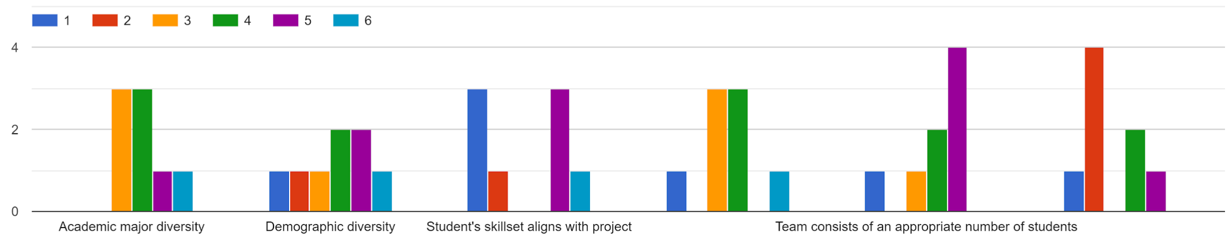


Rank the following preferences/attributes of team assignment according to their importance to you.



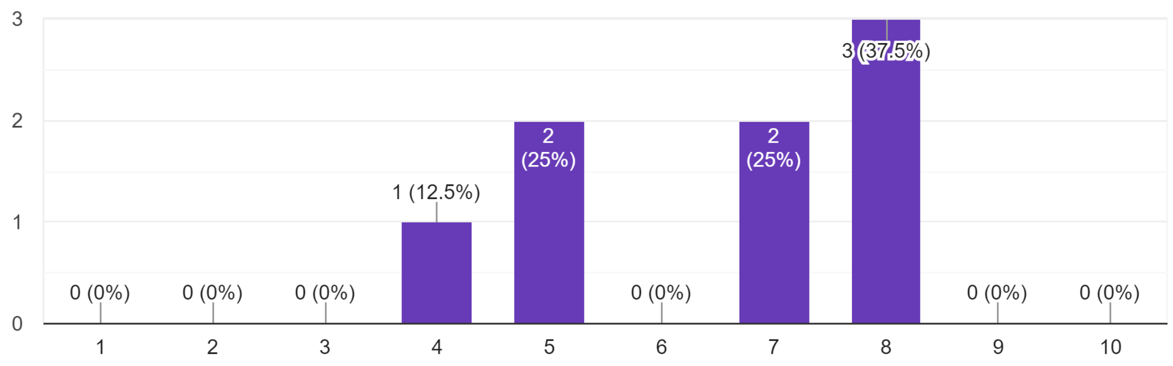
Faculty Responses

Rank the following preferences/attributes of team assignment according to their importance to you. 1 being the most important, 6 being the least important. (Please put in ranked order, i.e. each attribute receives a different score/ranking).



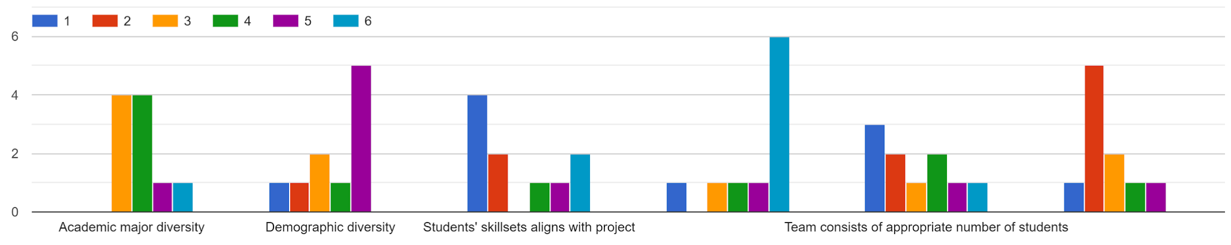
How satisfied are you with the team that you were assigned?

8 responses



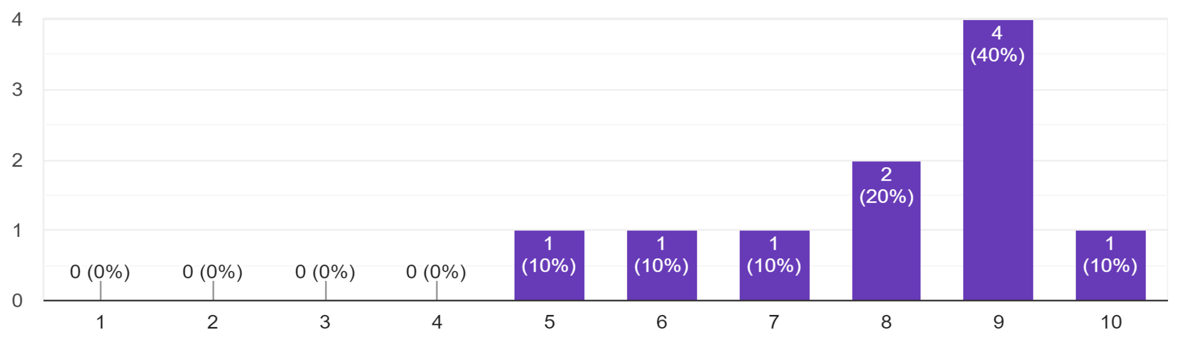
Client Responses

Rank the following preferences/attributes of team assignment according to their importance to you. 1 being the most important, 6 being the least important. (Please put in ranked order, i.e. each attribute receives a different score/ranking).



How satisfied are you with the team that you were assigned?

10 responses



4
3
2
1
0
0 (0%)
0 (0%)
0 (0%)
0 (0%)
1 (10%)
1 (10%)
1 (10%)
2 (20%)
4 (40%)
1 (10%)
1
2
3
4
5
6
7
8
9
10