Senior Design Server/Client Development for Project Matching [Phase 3]

Team 02

Client-Advisors	Dr. Akhilesh Tyagi and Jacob Grundmeier
Backend Design	Noah Nelson, Max Kueller
Frontend Design	Evan Brummer
Algorithm Design	Devin Tigges, Robert Holeman

Project Matching - Introduction

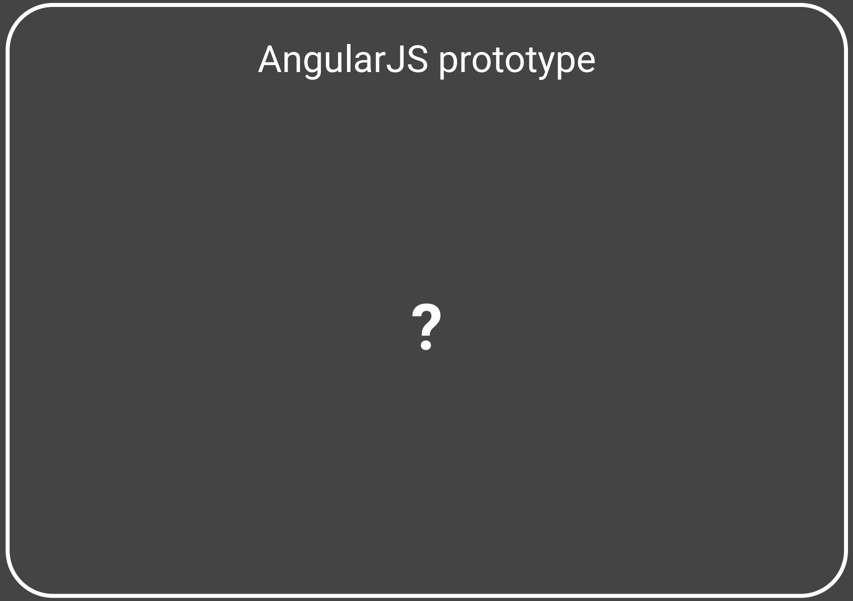
Our project is the third phase of an ECpE web application that will replace the 3rd party tools and processes currently used to manage senior design projects.

- Students will be able to configure preferences for projects and teammates.
- Clients will be able to submit projects via web forms.
- Faculty will be able to manage and assign projects using a specialized algorithm to match students to their preferences.

- It’s not quite done yet, one more phase!

Project Matching - Progress

- Component Diagram: Phase 1

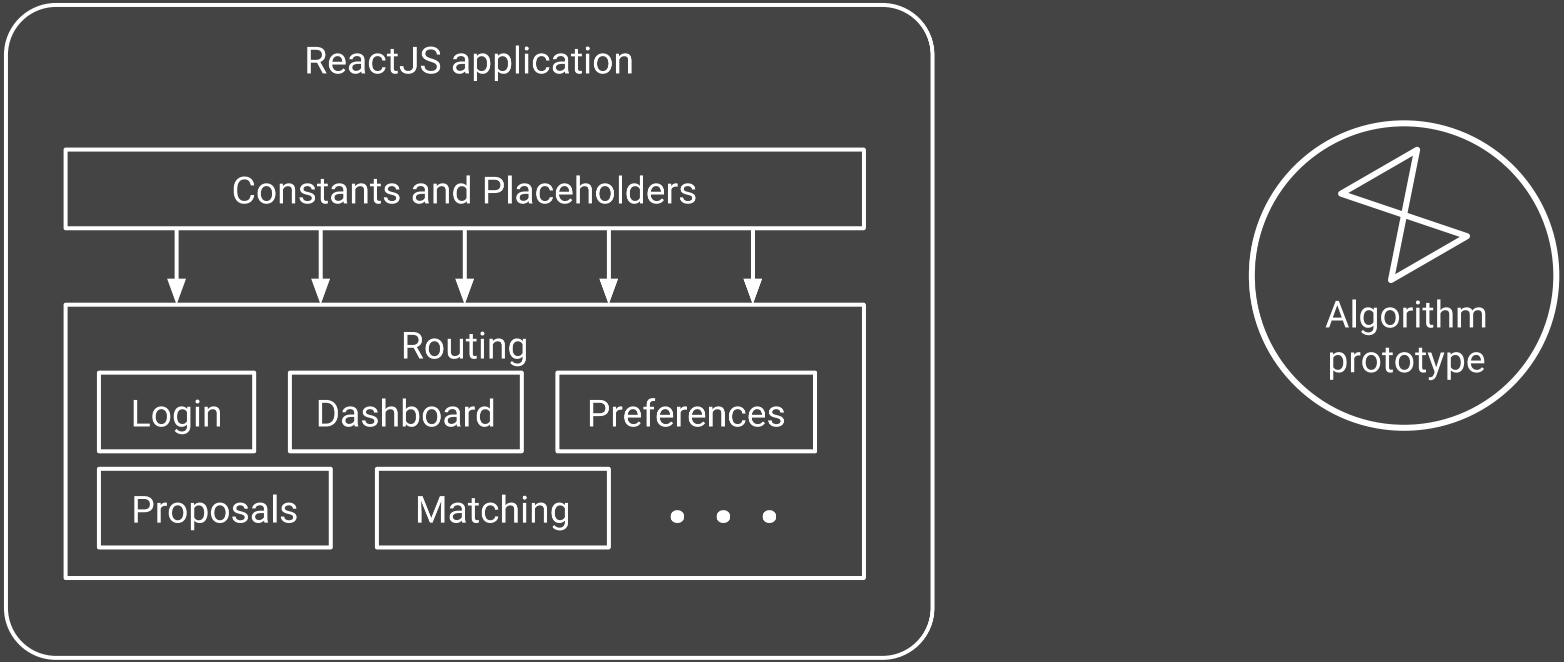


AngularJS prototype

?

Project Matching - Progress

- Component Diagram: Phase 2



ReactJS application

Constants and Placeholders

Routing

Login

Dashboard

Preferences

Proposals

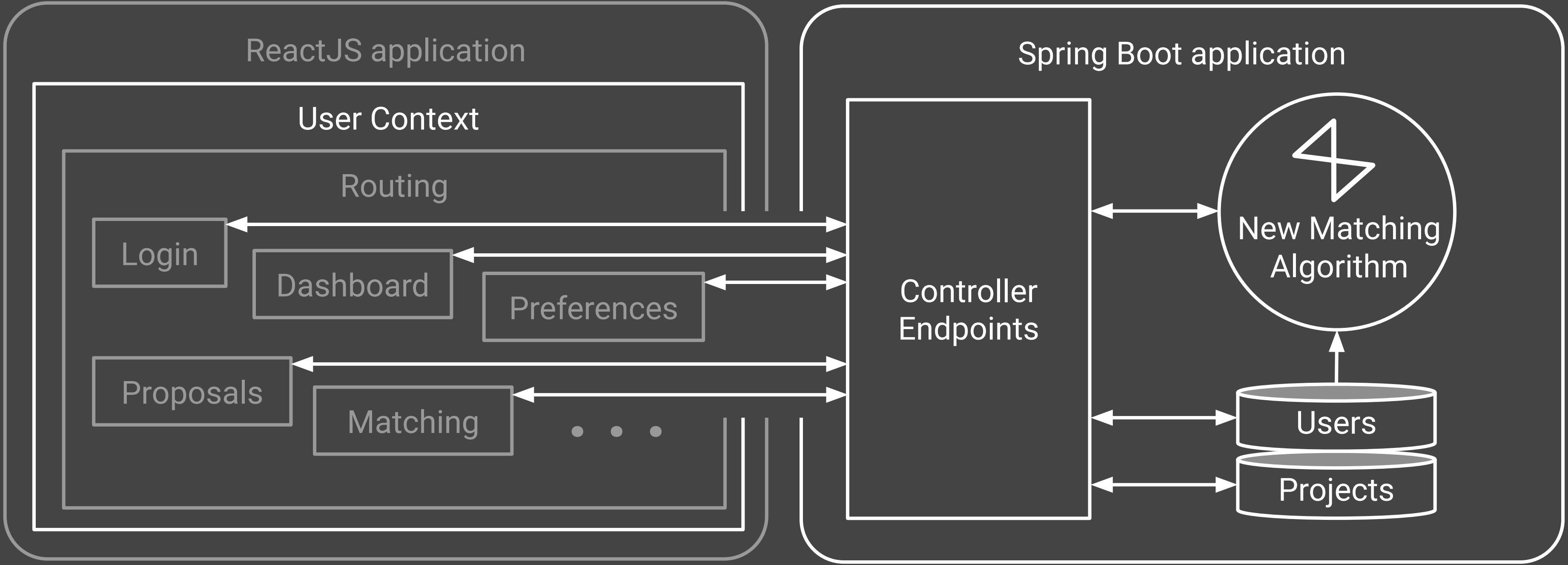
Matching

. . .

Algorithm prototype

Project Matching - Progress

- Component Diagram: Phase 3



Project Matching - Progress

Functional requirements: First Semester

1) The matching algorithm should accurately match students to a project based on their interests and preferences.

2) The web application should have separate functions based on the user type: proposal submission for clients, proposal selection for students, and proposal approval and assignment for administrators.

3) The system should handle a minimum of 100 concurrent users without a significant increase in response time.

4) UI should be expanded to allow Clients and Professors to enter projects.

5) After project matching occurs, the mass emailing process should be automated.

6) The database's design/structure should be improved significantly.

7) The algorithm should be proven to work as expected in testing environments with large data.

Project Matching - Progress

Functional requirements: Application

- The application must support 5 types of users that will use the app throughout a SD semester: Instructor, Advisor, Student, Client, and Board (members).
- The application must be able to intelligently parse CSV lists of user info (email, name, type), uploaded by Instructors to register users.
- Instructors must be able to **manage rosters** and user info at any scale, both large CSV imports and individual users.
- All users are required to **login** with an email that has been added/whitelisted by an Instructor and assigned at least one role (user type).
- The **login** page must accommodate both logging in via the Iowa State Single-Sign-On (SSO) and with a standard email and password, because some Clients may not have a registered NetID.
- **Users** must be able to have multiple roles (user types) assigned, with the option to switch between them and view the corresponding dashboard.
- **Users** (typically instructors) must be able to open multiple sessions on multiple user types for convenience when setting up for the semester.
- The **matching algorithm** should find an outcome that assigns students to their most preferred project(s) as much as possible. “Maximize the number of student bids satisfied.”

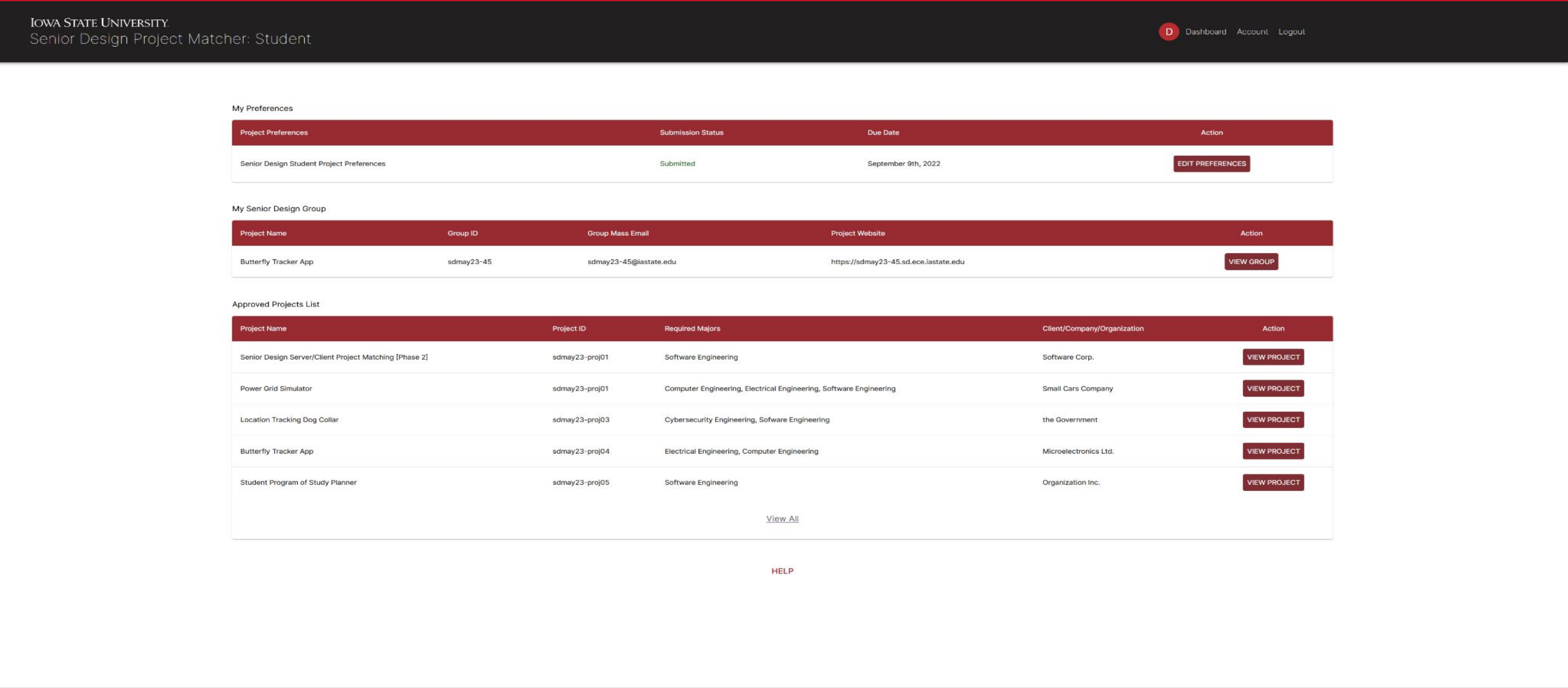
Project Matching - Progress

Functional Requirements: User Stories

- **Clients** must be able login, create proposal forms and submit them for review by an Instructor. (After being registered)
  - Proposals should contain the client name and email, project title, etc.
- **Instructors** must be able to approve and reject project proposals.
- **Students** must be able to declare their top [5] project preferences from the list of approved projects, and update their preferences at any point before the matching deadline.
- **Instructors** must be able to configure and send automated emails from inside the application. This is ideal for prompting clients to submit proposals, prompting students to enter project preferences, and other general notifications.
- **Instructors** must be able to execute a project matching algorithm to automatically assign teams of students to approved projects based on their preferences.
- **Instructors** must be able to modify the results of the matching algorithm to account for special or unexpected cases.
- **Instructors** must be able to assign Advisors to one or more generated project groups.
- **Board/Faculty members** must be able to sign up for one or more "industry review panel" timeslots near the end of the 492 semester.

Project Matching: Web Application

Student Dashboard



My Preferences

Project Preferences	Submission Status	Due Date	Action
Senior Design Student Project Preferences	Submitted	September 9th, 2022	EDIT PREFERENCES

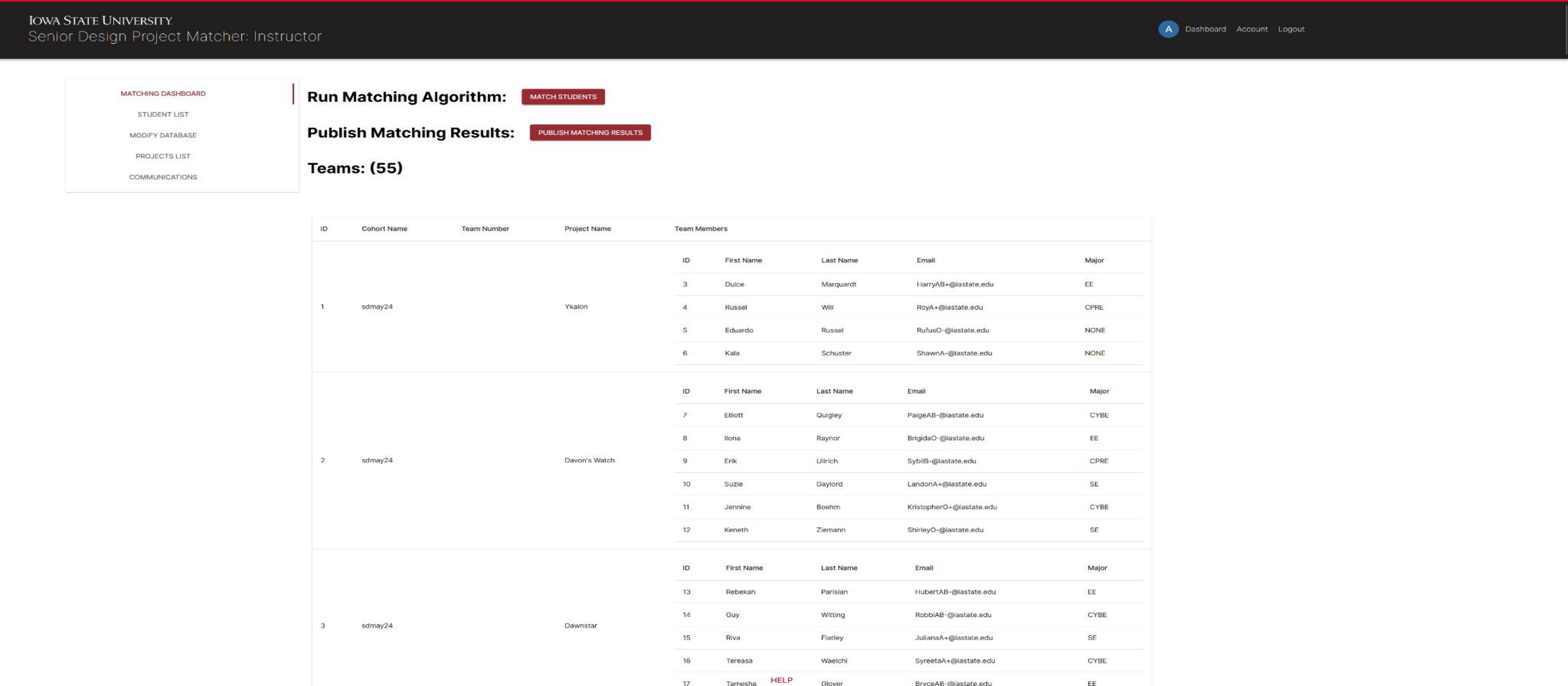
My Senior Design Group

Project Name	Group ID	Group Mass Email	Project Website	Action
Butterfly Tracker App	sdmay23-45	sdmay23-45@iastate.edu	https://sdmay23-45.sd.ece.iastate.edu	VIEW GROUP

Approved Projects List

Project Name	Project ID	Required Majors	Client/Company/Organization	Action
Senior Design Server/Client Project Matching [Phase 2]	sdmay23-proj01	Software Engineering	Software Corp.	VIEW PROJECT
Power Grid Simulator	sdmay23-proj01	Computer Engineering, Electrical Engineering, Software Engineering	Small Cars Company	VIEW PROJECT
Location Tracking Dog Collar	sdmay23-proj03	Cybersecurity Engineering, Sofware Engineering	the Government	VIEW PROJECT
Butterfly Tracker App	sdmay23-proj04	Electrical Engineering, Computer Engineering	Microelectronics Ltd.	VIEW PROJECT
Student Program of Study Planner	sdmay23-proj05	Software Engineering	Organization Inc.	VIEW PROJECT

Instructor Dashboard



Run Matching Algorithm:

Publish Matching Results:

Teams: (55)

ID	Cohort Name	Team Number	Project Name	Team Members				
				ID	First Name	Last Name	Email	Major
1	sdmay24		Ykalon	3	Dulce	Marquardt	HarryAB+@iastate.edu	EE
				4	Russel	Will	RoyA+@iastate.edu	CPRE
				5	Eduardo	Russel	RufusO-@iastate.edu	NONE
				6	Kala	Schuster	ShawnA-@iastate.edu	NONE
				ID	First Name	Last Name	Email	Major
2	sdmay24		Davon's Watch	7	Elliott	Quigley	PaigeAB-@iastate.edu	CYBE
				8	Ilona	Raynor	BrigidaO-@iastate.edu	EE
				9	Erik	Ullrich	SybilB-@iastate.edu	CPRE
				10	Suzie	Gaylord	LandonA+@iastate.edu	SE
				11	Jennine	Boehm	KristopherO+@iastate.edu	CYBE
				12	Keneth	Ziemann	ShirleyO-@iastate.edu	SE
				ID	First Name	Last Name	Email	Major
3	sdmay24		Dawnstar	13	Rebekah	Parisian	HubertAB-@iastate.edu	EE
				14	Guy	Witting	RobbiAB-@iastate.edu	CYBE
				15	Riva	Flatley	JulianaA+@iastate.edu	SE
				16	Tereasa	Waelchi	SyreetaA+@iastate.edu	CYBE
				17	Tamesha	Glover	BryceAB-@iastate.edu	EE

Project Matching: Backend Server

Team 02

Clients	Jacob Grundmeier and Akhilesh Tyagi
Advisor	Akhilesh Tyagi
Database Design	Noah Nelson
Frontend Design	Joshua Izumba, Noah Nelson, Evan Brummer
Algorithm Design	Robert Holeman, Devin Tigges, Max Kueler

Backend

- Java 17 and Spring Boot 3.1
- Maven
- Lombok annotation
- Hibernate (JPA) for database handling
- H2 for DB testing
- Faker for data generation
- Uses Service, Controller, Repository flow

Project Matching: Algorithm

Team 02

Clients	Jacob Grundmeier and Akhilesh Tyagi
Advisor	Akhilesh Tyagi
Database Design	Noah Nelson
Frontend Design	Joshua Izumba, Noah Nelson, Evan Brunnar
Algorithm Design	Robert Holeman, Devin Tigges, Max Kueller

Algorithm

We implemented a bidding algorithm optimized by student project and group mate preferences that gives students control over how they want project to be selected.

Algorithm - Design

Initialization: Set up the initial conditions by assigning each student to be free and marking each project and groupmate as totally unsubscribed.

Main Loop - Student Assignment: For each student in the list:

- Determine the highest-bid project and the highest-bid groupmate
- If the student has no project preference, set them in the first valid project that will take them or consider the groupmate bid if available.
- If the student already has a project, compare bids to decide whether to keep the current project or switch to a higher-bid project.
- Set the student's project to the determined highestProjectBid.

Algorithm - Design

Groupmate Handling: For the current student:

- Add groupmates unless major requirements are not met.
- If a groupmate already has a project: Check bids to decide whether to keep the current assignment or switch to another groupmate's project.
- Set the groupmate's project accordingly.

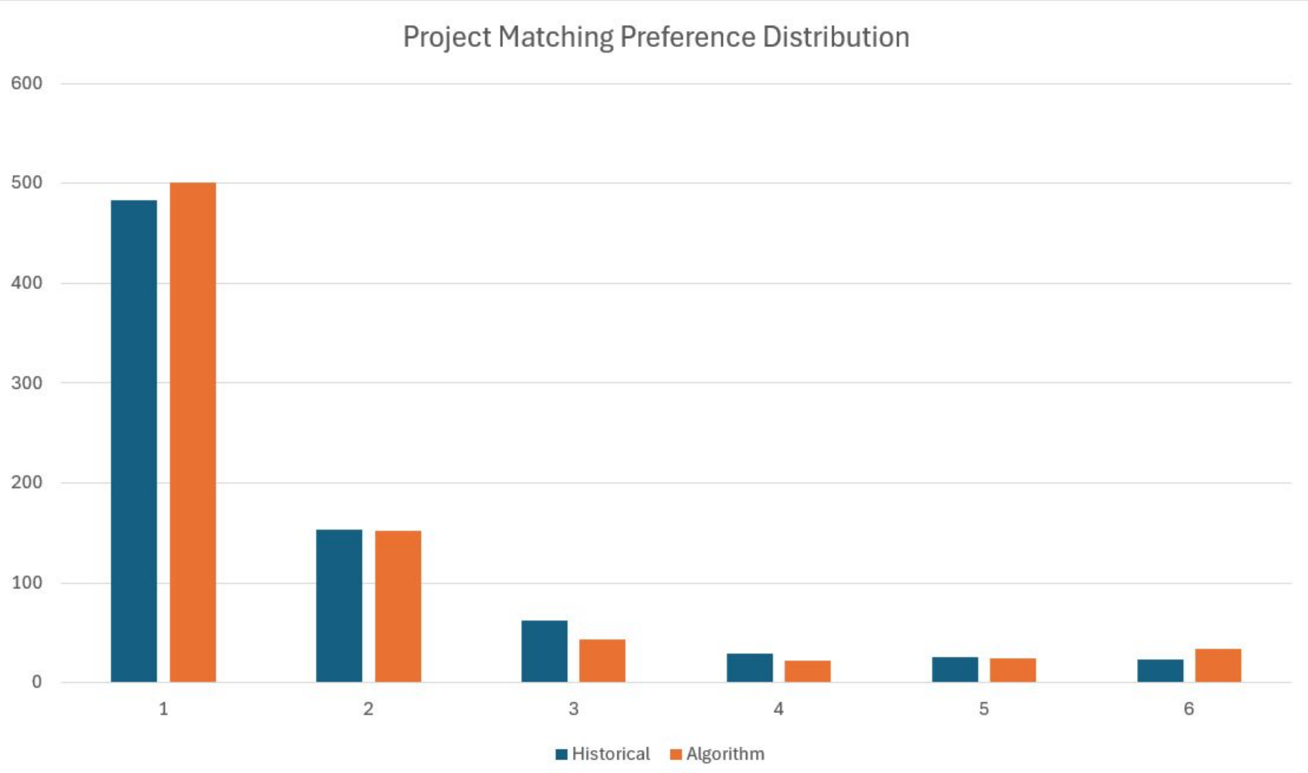
Second Pass - Assign to Open Projects:

- Go through the list of students again.
- For each student, find a project based on preferences and major.
- If the student is not initially assigned, match them to the first open project that will take them.

Result:

The algorithm returns a list of students with their assigned projects to be sent to the frontend.

Algorithm - Test Results



Demo

Team 02

Clients	Jacob Grundmeier and Akhilesh Tyagi
Advisor	Akhilesh Tyagi
Database Design	Noah Nelson
Frontend Design	Joshua Izumba, Noah Nelson, Evan Brunner
Algorithm Design	Robert Holeman, Devin Tigges, Max Kuehler