**PROJECT DEFINITION**

**INTRODUCTION**

The purpose of this document is to outline a Project Definition for this capstone project, the circumstances of the problem currently, the approach towards coming up with a solution to this problem, and procedures and practices of the project team. The goal is to clearly establish a direction and goals for the project overall and lay out standard and best practices for the team members. The project itself is a computer program to quickly and effectively assign electrical and computer engineering students to industry company capstone projects based on student and company preferences.

**PROBLEM STATEMENT**

This project seeks to solve the problem of needing to assign University of Texas electrical and computer engineering students to industry capstone projects based on the preferences and expertise of the students and the requirements and preferences of the industry companies. The project consists of, first, designing an interface to create what questions and metrics to ask the students and companies and, second, use csv files generated based on those questions and metrics to create group assignments for those students paired with industry companies.

**Design Problem Description**

The design problem is the designing of a computer program to effectively assign groups to teams. This solution is important because it expedites the process of the creation of capstone groups so that the students and industry companies can more quickly get started with the projects. It also creates groups that fit very well with putting students in the fields that fit their preferences and expertise while also giving industry companies students that are qualified to work on that project. The main challenge with this project is creating a metric that measures how good the teams are in line with preferences and other restrictions such as preventing bad outlier teams or accounting for willingness to work with a non-disclosure agreement (NDA) or an intellectual property (IP) agreement, and then creating those teams based on those metrics to make optimal team configurations.

***Background Information***

At the University of Texas at Austin, in the Chandra Department of Electrical and Computer Engineering, Electrical and Computer Engineering students complete a capstone design project to concretely implement what they have learned in university [1]. These students have three options in the formation of their groups, forming their own teams and proposing their own projects, forming their own teams and approaching faculty members who already have projects, and filling out a skills, preferences, and expertise form and being assigned to industry projects. This final option, the assignment of the residual students to industry projects is the subject of this project. In the current system, the technical professor of the senior design course manually looks through the form responses and manually forms the distribution of students to capstone design groups, a process that is tedious and time consuming. This project seeks to create a software program that, given the input of the preferences and expertise of the students, and the expertise requirements from the company, quickly creates an optimized distribution of students to groups.

***Design Functionality***

The program will intake inputs of students with preference and companies with preferences. The program will be designed to be modular and be able to be changed to add or take away questions related to fields of interest. After those get inputted, the program will make an assignment of capstone groups that best fits a model or requirements we specify. This will be in the form of an executable file that runs as a web-application locally on a computer. The program is not going to be easily scaled and changed to be able to be used for non-ECE capstone project assignment. The program is not going to be available online on the web.

***Relevant Standards***

Relevant Standards include standardizing largely with the current system. For instance, being able to use the csv file that the survey software currently outputs for student preferences would be a currently existing standard to keep in mind. There are also coding standards that should be adhered to during the process of making the program.

***Ethical Considerations***

There are potential ethical considerations in the area of privacy in this specific design problem. The development of the design project virtually requires using previous data from previous years. When the form for group preferences was filled out, the students filling out the form were informed that only the professor would be able to see their answers. In regular use of the program, only the professor would be able to see their answers, but in developing this project, if previous years’ data is used, that would be a privacy violation. The data would need to be anonymized to prevent this. Another ethical consideration is not leaving students behind in their groups. For instance, an algorithm that wants to make the best possible overall rankings by making one group that is not workable like putting all students with the lowest GPAs into one group could present an ethical issue of intentionally setting up those students to fail. Another ethical consideration is ensuring that students are not mismatched into groups where they need to sign NDA or IP agreements when they stated that they are not willing to sign such agreements.

***Project Deliverables***

At the conclusion of the project, we will deliver an downloadable executable file that, through receiving the correct inputs, will provide a corresponding output of the distribution of students to capstone groups according to the input and output specifications. We will also deliver along with this file all the code that the program uses and code documentation.

**Requirements Specification**

Outlining the necessary specifications for requirements allows for the tracking of progress and clearly outlines the specific goals for the project for future reference.

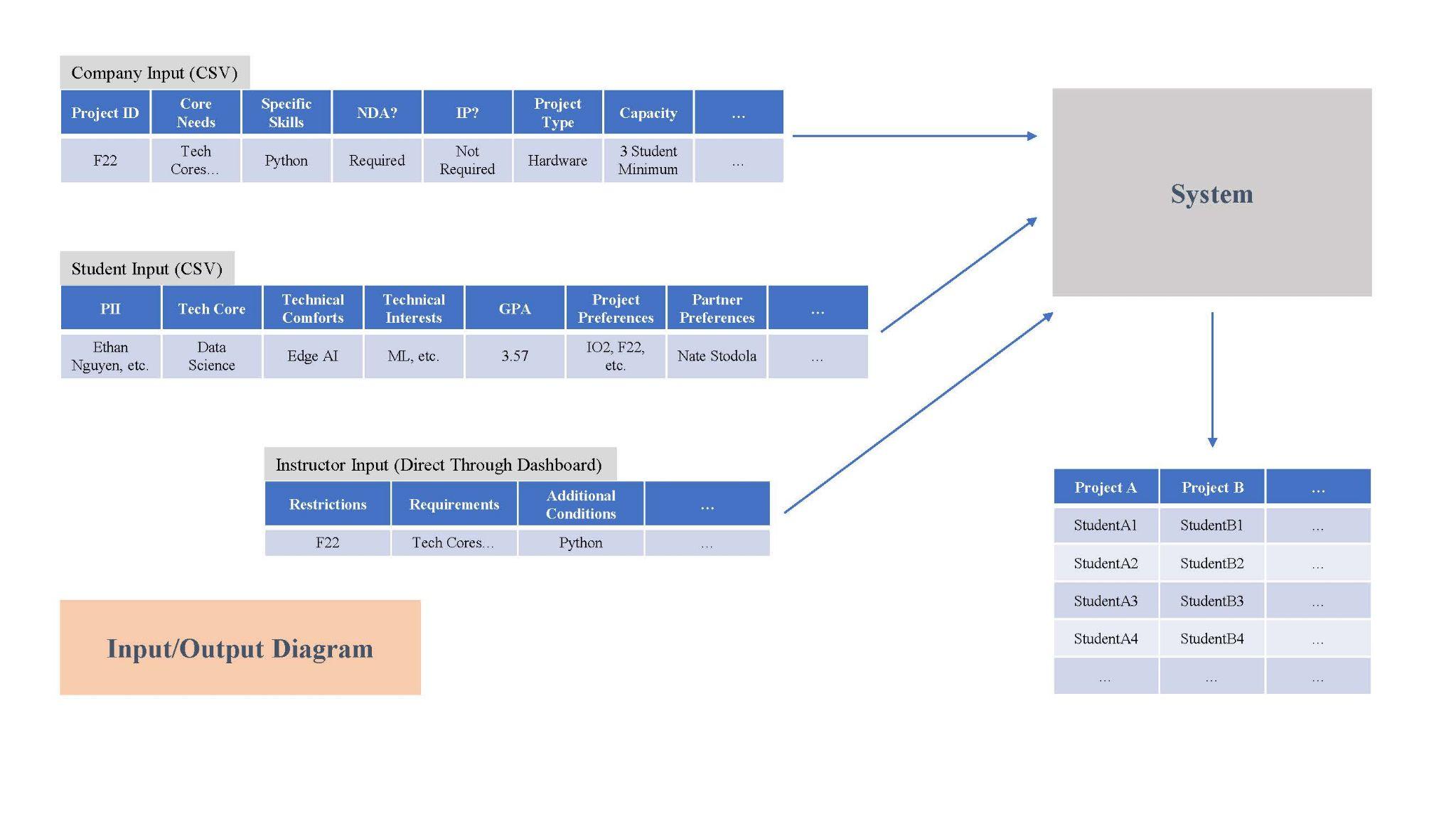
***Inputs and Output Specifications***

For this project, the designed system may receive inputs from three general categories of individuals, as seen in the Input/Output Diagram attached:

* Each company will provide input information on its particular project for which it requires a team.
* Each student will provide information on their own skills, comforts, and preferences.
* The instructor may provide additional general input, including certain universal constraints that must be met.

The system will need to process all of this information in order to properly assign students in line with boundary constraints and as optimally as possible, output corresponding team assignments for each project. As outlined in the attached table, there is a wide range of data types and data ranges for each possible element. Below is a sketched illustration and a table outlining the different inputs and output for the system:

**Figure I. Input/Output Diagram for the Program**



**Table 1. Input/Output Descriptions**

| **Input/Output Category** | **Input** | **Value Type and Range** | **Description** |
| --- | --- | --- | --- |
| Company Input | Project ID | String | Identifier for each company project |
| Company Input | Specific Skills Needed | List of Strings | Necessary unique and niche skills on the part of team members required by company to complete the project |
| Company Input | Core Needs | List of Strings | Necessary general areas of expertise and experience required by company to complete the project |
| Company Input | NDA | Boolean | Does the company require team members to sign an NDA? |
| Company Input | IP Agreement | Boolean | Does the company require team members to sign an IP agreement? |
| Company Input | Project Type | String, either "hardware", "software", or "both" | Does the project consist in developing hardware, software, or both? |
| Company Input | Min Capacity | Integer | Minimum number of team members required |
| Student Input | Student Personal Identifiable Information (PII) | Set of values, name, year, EID, etc | Key information to identify each student |
| Student Input | Tech Core | String | The technical ECE core which this particular student possesses |
| Student Input | Skill Experience | Set of Pairings (specific skill for key, comfort for value) | Of the specific skills required for each project as provided by the companies, which, if any, does the student have familiarity with? |
| Student Input | Tech Comforts | Set of Pairings (comfort area for key, comfort level for value) | Of the different core needs provided by the companies, how comfortable is the student with them each? |
| Student Input | Interests | Set of Pairings (interest area for key, interest level for value) | Of the different core areas provided by companies, which fields is the student most interested in working in? |
| Student Input | GPA | float | The student's grade point average (for preventing too low an overall GPA of a team) |
| Student Input | Project Preference | Set of Pairings (Project ID for key, preference for value) | Which projects would the student rather or rather not work on? |
| Student Input | Partner Preference | String | Does the student have a particular partner they would like to work with? |
| Student Input | IP Willingness | boolean | Whether the student is willing to sign an IP agreement |
| Student Input | NDA Willingness | boolean | Whether the student is willing to sign an NDA |
| General Input | Restrictions on particular student-assignments | Set of Strings | Any and all blocks between a student and a particular project assignment or pairing with another student |
| General Input | Requirements on particular student-assignments | Set of Strings | Any and all necessary assignments between a student and a particular project or pairing with another student |
| General Input | Other conditions | Any | Any and all additional conditions |
| Output | Project Assignments | Set of Pairings (Project ID for Key, List of team members for Value) | Every Project and its corresponding student team assignment |

***User Interface Specifications***

For the system design, the first and most important user interface will be that used by the instructor in charge of arranging student teams. The instructor will need a means to input company project specifications and student information in the form of CSVs, as well as means to input additional restrictions and requirements, weights and biases, etc. The instructor user interface specifications are listed below in Table 2. Additionally, the system could include a user interface for companies to input project information that automatically compiles the data into csv form and begins outlining the form which students will need to fill out, which itself could take the form of another user interface, though there are already existing means for collecting this data. The user interface for the instructor (and any additional user interfaces) would most ideally be accessible from a browser.

**Table 2. Instructor User Interface Specifications**

| **Specifications/features** | **Description** |
| --- | --- |
| Company Project Info Input | Takes in (at least) a CSV detailing company project info and which automatically organizes it |
| Student Info Input | Takes in (at least) a CSV detailing student info and which begins to format, filter, and organize it corresponding to relevant project data |
| Display Panel for Filtering | A panel which displays all of the project and student data and which allows the instructor to filter and view the data to get a preliminary notion of ideal matches |
| Input for Conditions | An input which allows the instructor to give additional conditions the calculated team assignments will have to match, needs to be accessible for a user with little programming knowledge |
| Restrictions and Requirements on Students Input | Allows professor to restrict or force assignment for particular students to other particular students or projects |
| Input for Adjusting Weights & Biases on Data Elements | Allows instructor to bump up or down the importance of particular points of data, even for particular projects, such as experience in core areas, acquisition of needed skills, etc |
| Panel for Analyzing & Optimizing Student Team Output | A panel which displays calculated possible team assignments to different projects based on the current selection of criteria and which outputs an "effectiveness" score for each individual team and for how optimal the entire distribution of teams is |

***Operating Environment Specifications***

Our goal will be to simplify environmental specifications and requirements as much as possible for users who have limited programming knowledge as well as to make our product available to use off-line. Our product will be able to run across any operating system with the front-end on the user’s browser. Environmental specifications are listed in Table 3 below.

**Table 3. Environmental Specifications**

| **Environmental Specifications** | **Description** |
| --- | --- |
| JRE (ideally newest) | A working Java Runtime Environment will be needed; most (if not all) computers come with this prepackaged |
| Browser (Ideally Chromium) | React is functional on most browsers, but Chromium is most ideal; a functional browser will be sufficient to run the product |
| Running OS on a Functional Computer | A working computer with an OS (Windows, Mac, or Linux) will be needed to possess as well as use the product |

***Performance Specifications***

Once our program has outputted a distribution of teams with assigned students, the optimality of individual teams as well as, in consequence, the optimality of the entire distribution by which the performance of the program will be evaluated will take into account each corresponding company’s needs in its core area of undergraduate study, for expertise on specific topics, for the project’s nature, and for its NDA/IP requirements, and will quantify how well these needs match to the project’s students’ technical comfort, interest, and willingness as well as how well the students’ own preferences were met. These measures of optimality will likely be single numbers, ranging between 0 and 1, where a score of 1 suggests that all of a company’s needs and its students’ preferences have been fully met and a score of 0 suggests that either none of the needs of the company were met, whatsoever, or some other absolutely necessary condition for any assembled team was not fulfilled, such as, for example, the team’s total GPA not passing some absolutely necessary minimum threshold.

**TEAM ORGANIZATION**

The key to any collaborative project is effective organization and a clear delineation of responsibilities and a structure to support the team. The team is mainly organized into two subteams along with additional project roles. In addition there are specific communication media and standards that will be used to communicate between team members and keep documents, code, and documentation organized.

**Project Roles**

Each team member is assigned into specific roles for the functioning of the project. Roles include being present on a particular subteam or having an additional role such as project lead or secretary.

**Subteams**

The team is divided into two main sub teams in order to effectively allocate responsibilities to the respective members in the respective teams. Each team will have individual team meetings along with the main meetings with all of the team members. The meetings with all team members is a regularly scheduled event. The individual subteam meetings are up to the discretion of the members of the subteam.

***Backend Team***

The backend team has Ethan and Nathan. The backend team is primarily responsible for the backend for developing the backend of the project and all the responsibilities that go along with that like using Java coding standards and writing documentation. This team will also work very closely with the other team, the frontend team. The backend team membership, along with corresponding areas of expertise and coursework for each student are outlined in Table 4 (next page.

***Frontend Team***

The frontend team has Gabriel. The Frontend Team is primarily responsible for the frontend for developing the frontend of the project and all the responsibilities that go along with that like using JavaScript coding standards and writing documentation. This team will also work very closely with the other team, the backend team. The Frontend Team membership, along with corresponding areas of expertise and coursework for each student are outlined in Table 4 (next page).

***Additional Roles***

Project Lead (Ethan): The project lead coordinates activities across the full project and is the primary point of contact with the mentor, technical TA, and client.

Secretary (Nathan): The secretary takes notes at the meetings and is primarily responsible for documenting what happens at each meeting and putting a summary in the groupchat for the meetings and what is happening next.

**Table 4. Personnel Information**

| **Name** | **Planned contributions** | **Areas of expertise** | **Related coursework** |
| --- | --- | --- | --- |
| Ethan Nguyen | Project Lead,  Member of Backend Team | Backend software Development in Java | ECE 312,  ECE 422C, |
| Nathan Stodola | Secretary,  Member of Backend Team | Backend software Development in Java, Frontend in Javascript | ECE 312,  ECE 422C,  ECE 360G,  ECE 461L |
| Gabriel Mount | Member of Frontend Team | Experience in Java, Frontend in Javascript with React | ECE 312,  ECE 422C,  ECE 461L |

**Team Policies**

Team policies are in place to have standardization of communication, establish expectations, and just generally provide for the well functioning of the team logsitically. The team policies are meant to mitigate confusion and conflict by creating a consistent framework around which the team operates.

***Project Management Tools and Platforms***

The team will use Discord for all strictly internal team communication including virtual meetings and regular text communication. The team will keep track of tasks through meeting summary documents posted by our secretary, Nathan Stodola. Scheduling of meetings, events, and due dates will occur on Google Calendar. Documentation management will occur in GoogleDocs in a shared Google Drive for organization.

***Team Expectations***

Team members are expected to attend meetings regularly and complete their tasks within the given deadlines. Team members are also expected to have open communication and be prompt in responding and communication issues and updates to the other team members. Team members are also expected to be flexible and be able to adapt to changing circumstances.

***Conflict Resolution***

Decision making is not through bare-majority democracy, rather through group consensus and all team members being on the same page. When conflicts arise on the team, a consensus approach to resolution is the preferred practice. Team members are to address disagreements calmly and constructively, with the aim of understanding each other in respect and coming to a consensus of ultimately what is best for the project.

**CONCLUSION**

The purpose of this document is to outline a Project Definition for this capstone project, the circumstances of the problem currently, the approach towards coming up with a solution to this problem, and procedures and practices of the project team. The goal is to clearly establish a direction and goals for the project overall, lay out standards and best practices for the team members, and give a preliminary plan for the architecture of the project.

**REFERENCE**

[1] “Capstone design: Texas ECE - Electrical & Computer Engineering at UT Austin,” The University of Texas at Austin Chandra Department of Electrical and Computer Engineering, https://www.ece.utexas.edu/academics/undergraduate/capstone

**APPENDIX A: APPLICABLE STANDARDS**

[1] *IEEE Reference Guide.* IEEE Standard, 2023