Proceedings of Student-Faculty Research Day, CSIS, Pace University, May 5th, 2017

Automated Team Assignment System using Greedy Algorithm for Efficient Student Group Formation

Jimmy Patel *and* Md M H K Mujahedi
Seidenberg School of CSIS, Pace University, New York, NY

***Abstract* – Capstone courses are increasingly used in educational institutions as part of graduation requirement. Students come from diverse backgrounds and with different skill sets or differing levels of specific skill sets. Different combination of skills in individual groups benefits each member. It is difficult for a faculty to form student groups with perfect combination of skills for individual projects. This study proposes a system that will automate the process of group formation based on student skills, project preferences, and individual project requirements.**

***Index Terms*: Capstone course, Preference, Greedy Algorithm, Best Match, Fit Score, Optimize.**

I. INTRODUCTION

Some educational institutions require capstone projects to be completed by a team of students [2]. Large projects especially capstone projects require use of all aspect of the learning to implement in projects. Successful team formation is a rigorous process, as it requires knowledge of project requirements (different kinds of skills), as well as student skill set.
In many cases, the projects are multi-disciplinary and hence different types of students need to participate in such projects with varied skill sets.

An appropriate team formation is crucial from two different aspects. First, students have their skills and preferences. Usually students prefer to select projects for which they are capable of doing the tasks and having appropriate skill levels. If students are assigned to a project that matches their skills, then the outcome of the projects are better [3]. Some capstone projects have clients from a company or organization, and they sometimes provide sponsorship (in the form of scholarship). And success in such project ensures continuation of sponsorship.
Secondly, the students gain more knowledge from the project. Appropriate combination of skills helps them learn different skills from each other.

Different approaches, however, are used by different Institutions for group assignment [1].

a) Team members are selected by the faculty manually based on inspection of project skill requirements and student skills.
b) Allowing students to form their teams. This approach may not be best way to form teams all the time because instead of choosing themselves based on their skill match they could form themselves with personal preference (friendships).
c) Using an algorithmic process to assign groups that is a best match for all students.

Selecting group members using the first method as described above is a tedious process. The faculty has to engage and spend significant time for inspection and team formation. There could be big imbalance in different teams in terms of skills. Also mismatches could occur between student skills and project required skills. The main objective of collaborative learning –complementing different skills from different students to generate maximum output – may not be met.

The second method may not give best results either. Although students select projects based on their skills and interests, sometimes students select team mates based on their friendship or personal suitability. In some cases, students become unassigned to any group – no one showed interests to make them team members or they did not try themselves to involve with others to form groups. Sometimes some minority students form groups with each other only, because they themselves become leftovers and only choice left for them is to form groups by themselves [13].

To avoid such problems, it is best way to use an automated system to assign students to different teams based on the skills and project requirements. The faculty could also overwrite (make minor adjustments) to assignment using his/her own judgment.

II. BACKGROUND

Approaches to Team Assignment

There are different approaches to team assignment – fully automated process using all possible criteria, e.g. student skills, skill level, student's project preferences, and student's diversity, and semi-automated process using student's skills and project's required skills. In a fully-automated process, allocation is formulated as an optimization problem and use a program to search for an optimal solution, where optimal means that the solution satisfies all constraints (e.g. the number of students on each project) while minimizing a cost, like violations of student preferences, or maximizing a complementary benefit.

There are some limitations to the approach [3]:

- It requires all parameters to be stored for processing by the program. Such approach could be easy with capturing hard data, like student skills as entered by the students. But it would be difficult to assess other information, like student's personalities and attitudes.
- It requires all kinds of parameters to be captured and in such way that it could be calculated numerically. This results in achieving some of the goals like assigning students to their preferred projects, but becomes very difficult to form teams with perfect combination of skills.
- There could be conflicting goals, and it requires to balance among those goals to achieve optimal result. Unfortunately, there is no general method or guideline for such balancing procedure. It is possible to achieve the goal by selecting a specific criterion unanimously only.
- It also requires to be trusted by all participant. Everyone should accept the fact the result of such system generates optimal output. It is only possible if the results of such system match participant's expected outputs.

It is very difficult to satisfy everyone (faculty and students) with the optimal solution generated by the system. Everyone involved – faculty and students – might have nonmatching goals even sometimes with conflicting goals and values.

It is very difficult to come up with a solution that satisfy everyone. An ideal system would be that which satisfy everyone in such a way that it would be perceived by everyone as the best possible match with minimal compromise in either side. Everyone would have a satisfaction because the system gave best possible output.

Previous works

Dyer and Mulvey [8] worked on group assignment problem which may be considered one of the first studies on the topic. They used faculty and student preference in a balanced way, although heavier preference was given on faculty's choice. The study mainly focused on a faculty's capability of conducting courses under certain constraints such as available time slots and number of courses to be taught.

The other approach to assign students to groups focused on student diversity, each group with student composition of maximum possible diversity, and also minimize difference among groups. The two approaches, however, produces same output [11].

Beheshtian-Ardekani and Mahmood used a method to balance student experience by making total scores of the groups as equal as possible. They used greedy heuristic algorithm to implement that. The results were, instead of being efficient, gave not so optimal results [7]. Weitz and Lakshminarayan [9] also worked on creating groups with student's maximized difference in the groups. They also used heuristic approach to attack the problem. They used VLSI design problem. Their results show that the group totals were ensured but the balance in individual groups did not. In one group students were of equal skills with total that is equal to another group total having very low and very high skilled student combinations. So, pedagogical goals could not be achieved.

Baker and Powell [11] came up with some optimization functions to ensure maximum diverse groups. They used optimization being used in the industry to use in student assignment problem. The end result is that the group diversity were ensured in those methods but not optimized in pedagogical sense.

Very little software implementation has been seen regarding this kind of team formation problem, there was a system, however, developed and used at Rutgers-Camden taking care of diverse students having diverse schedules [15]. A PASCAL based software was developed and used over four years. Although the software used mainly student time schedules, relevant experience and preference were given less priority.

III. METHODOLOGY

The objective of the system is to develop a process to optimize group assignment of students. The system will also reduce time for team assignment work of the faculty. The system will take care of the common issues of time consuming team assignment, improper combination of skill sets in individual projects.

The new system is intended to be more automated version than collecting data via google forms and putting into excel sheets. The Web based interface, after deployment on the server, will facilitate capturing student survey data through web browsing and registering into the system. The faculty will also be able to enter project data and client data into the system.

Students will provide data along with self-assed skill level (0 to 10), and project preference. The following are the sample data that the students will provide.

a) Domain skills – Requirements Specification/Business Analysis, Web Design, Web Development, Database design/Development, System design, Network engineering, AI/Pattern recognition, Data Science, Quality Assurance/Testing, and Academic research/writing.

b) Technical skills - Java, C#, PHP, Python, Swift, HTML, CSS, JavaScript, ASP.NET, SQL, MySQL, SQL Server, and Mobile development (Android and iPhone).

c) Student preference for five projects.

Individual projects will also have required skill sets and skill levels. Faculty will give input to those data.

The skills data are not fixed, it will be varied semester to semester. When faculty first enters project information for a specific semester for a specific course, the list of skill set is entered in the system (saved in database). Only those skill sets will be shown on the student survey page.

Utilizing the project and student data, the system will make group member assignment by utilizing greedy algorithm approach. The system will make scores for individual student's suitability for different projects by matching the skills with each project's required skill sets (fit score) as well as his/her project preference. The algorithm will use weighted fit score [2] for each skills required by the project by using multipliers as shown in the following table. The skill level will be multiplied with the preference multiplier to find the fit score.

Preference	Multiplier
1st Choice	1.25
2nd Choice	1.20
3rd Choice	1.15
4th Choice	1.10
5th Choice	1.05

The percentage array will be applied to the skill level as input by the student in survey data to generate weighted value to apply matching skill level for each project.

The system will run recursively, and in the first run system will assign best matches to individual projects. Hence, in the first run, the system will assign minimum number of students in each project. The process will continue with the remaining students until all students are assign to projects. In the second and subsequent runs, student with lesser skills (fit score) will be chosen. A student could only belong to only one project for a specific course.

The algorithm will run with project information, skills information, and student information.

Step 1 -Select minimum team members for each project:

```
Loop through all students
  For each project, select one
  student for each project from
  student's skill data and
  preference data based on best match
  (multiplication of skill and
   Preference multiplier)
End Loop
```

Step 2 – Select one student having soft skills for each project:

```
Loop through all students
  For each project, check if already
  selected students have soft
  skills. If found, do nothing.
  Else,
  select students from students
  table based on soft skill plus
  Preference (each project with one
  student having communication
  skill)
END Loop
```

Step 3 – Fill the teams with remaining students:

```
Loop through all student
  If all the students are exhausted
  in one run, then end.
  Else, run the process again for
  remaining best matches. Continue
  until all best matches are assigned
  to projects with maximum limit of
  members for every project.
End Loop
```

System Architecture

The system is a web-based application based on ASP.NET MVC framework (Figure 1). The reason for selecting this architecture is to develop a stable web app which would be easily deployed and leaves option for future scalability. The web based applications allow user to interact with the application using browsers on their machines. The advantage of web based application is its availability. It can be accessed by any one from anywhere with internet connection. The web application has been designed as 3-tier architecture - web client, web application server, and back-end database system. The project is intended to be a user-friendly web-based application that automates the student group assignment.

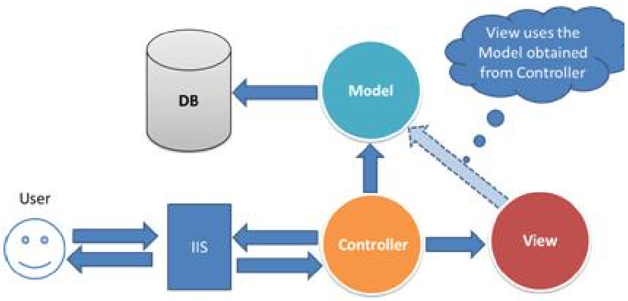


Figure 1: ASP.NET MVC Architecture

The system will be build step by step. In the first version, the only focus was to develop basic data entry facility and come up with main algorithm for assigning students to teams.

The front end of the application is based on HTML, CSS, and JavaScript. The back-end classes for data access and algorithm implementation is developed with C# language.

The general architecture of the system consists of three tiers – presentation tier (rendering user interfaces), business tier (implementation of business logic), and Data tier (for data access components). The following figure shows the general architecture of the web application:

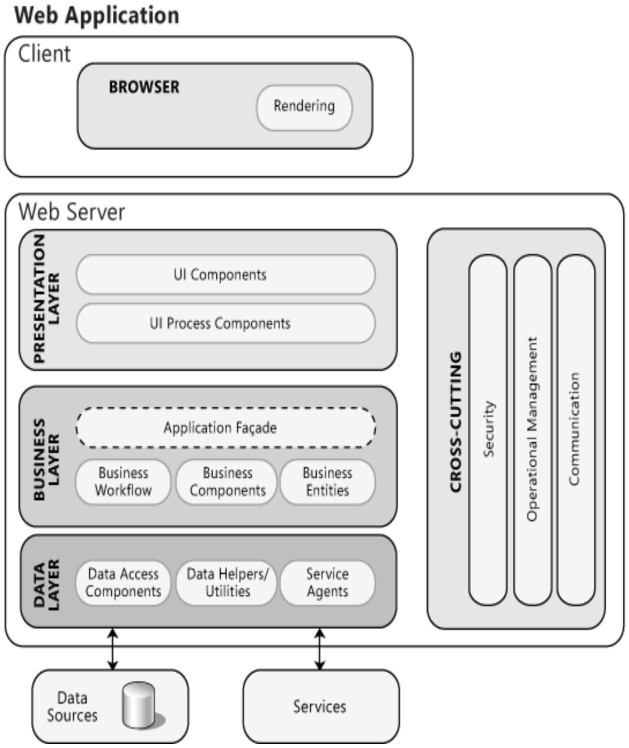


Figure 2: Web Application architecture [16]

The first version of the system has been built with simple interface for enabling easy data entry and producing team assignment report.

The system consists of following components:

- Web Interface with components for – a) sign up and log in to the system (to be developed in the next version), b) Web-based survey form for entering student information, c) project information screen to enter project information (project names, required skills, and required number of students)

d) Client information screen to enter project client information

- Back-end classes for processing survey data and project data and saving to database table, processing team assignment algorithm and save in database table as well as displaying results on request of the user
- Web-based interface showing the results of team formation
- Database with candidate team member information, project information, client information, and generated team information.

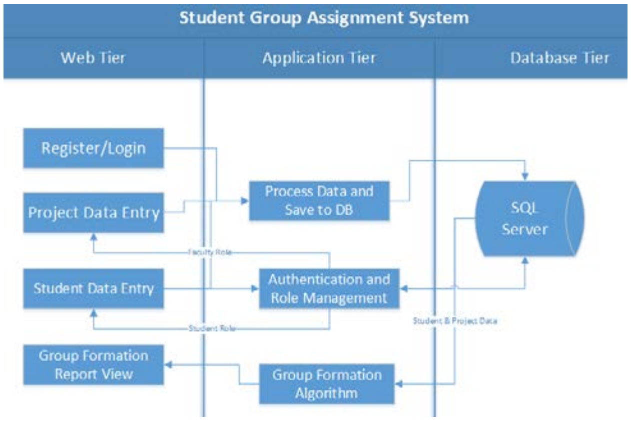

Student Group Assignment System
Web Tier
Application Tier
Database Tier
Register/Login
Project Data Entry
Process Data and Save to DB
SQL Server
Faculty Role
Student Data Entry
Authentication and Role Management
Student Role
Student & Project Data
Group Formation Report View
Group Formation Algorithm

Figure 3: Flow chart diagram of the system

Web Interface

The web interface will collect team member input having three parameters (technical skills, soft skills, and project preference) and display project teams formed.

The register and log in pages will manage users and roles (student, and faculty) and give appropriate permission and links to pages.

The web based system will provide authorization to specific user (professor) to run the different processes. System will make calculations with the data from the database and generate result as a report on the web page.

The web interface is based on HTML, CSS, and JavaScript. The system will be accessible using any web browser – Internet Explorer, Chrome, or Firefox.

The site navigation is as follows:

The index page:

This is where the users will first visit (students or faculty). The main page will show log in or register option. A new user will be asked for sign up or register. After registration (with user name/email address and password). For students, the role will be assigned to student role, the faculty user will be assigned to faculty role with appropriate authorization. Faculty will be given special invitation for log in with initial password by the admin. The log in data and role data will be saved in the access database. (The registration option for the new users will be available in the next version of the system).

Home Page:

Based on the role, the user will be shown appropriate pages. The home page will show student survey data entry form for students, and faculty will be shown Project and Client information entry page. (Role based authorization to specific function will also be available in the next version of the project).

Student Page:

Student page consists of following sections -

- Student survey – students will enter semester and course information, then students will enter skills along with skill level from the list shown (the list of skills will be available as faculty enters project information data with required skills). Students will also enter project preference data for 5 projects from the list shown.
- Update survey – students will be able to modify the survey data they entered.
- View teams – students will be able to see the teams with students assigned. They can identify their own team from the report.

Faculty Pages:

Faculty page will consist of Manage Project, Manage Team, Manage Client, and Manage Students links.

Manage Project:

- Enter Projects – faculty will enter project names, skills needed for each project. The list of skills will be saved in the database and will show up in the list of skills in student's page.
- Update projects – option for updating previously entered project data
- Delete projects – options for deleting projects wrongly entered
- View projects – to view all the projects entered.

Manage teams:

- Create teams – faculty will create teams for the specific semester and course to be filled with students with the algorithm
- Modify teams – for modifying team information. Faculty could also redistribute students among teams after the automated algorithm populates the teams. (The option will be available in the next version of the project).
- View teams – for viewing the teams created and modified by the faculty.

Manage Students:

- View student survey data – for viewing all the data for the students filled out so far
- View student list – for viewing all the students registered so far (including those not completed survey yet).

Manage clients:
(Client information entry will be available in the next version of the system).

- Enter client - for entering client data and the projects they are client of
- Update client – for updating client information previously entered
- Delete client – for deleting client data wrongly entered or not available for a project
- View client – for viewing all the client information as a report.

Following figure shows web page for collecting student information (to be filled by each student):

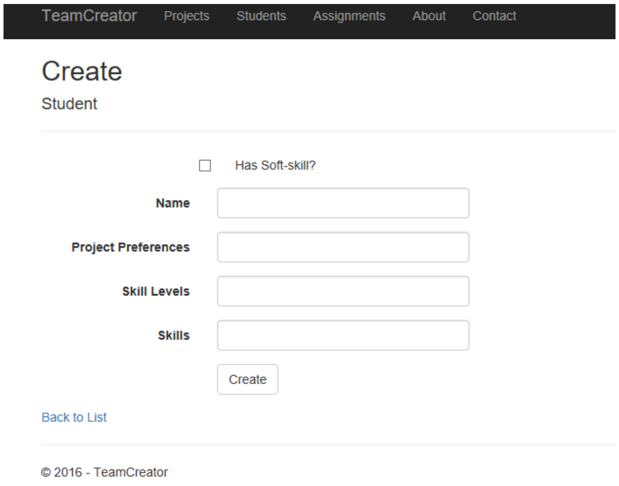


Figure 4: Project and skill data entry screens

Figure below shows the sample screen for other skills

Other Skills
Select your skill level for the following between 0 and 10, 0 being no skill and 10 being exert.

Skill Name	0	1	2	3	4	5	6	7	8	9	10
Communication											
Presentation											
System Analysis											
System Design											
Academic Writing											
Leadership											

Figure 5: Soft skills entry page

Figure below shows web page for entering project information:

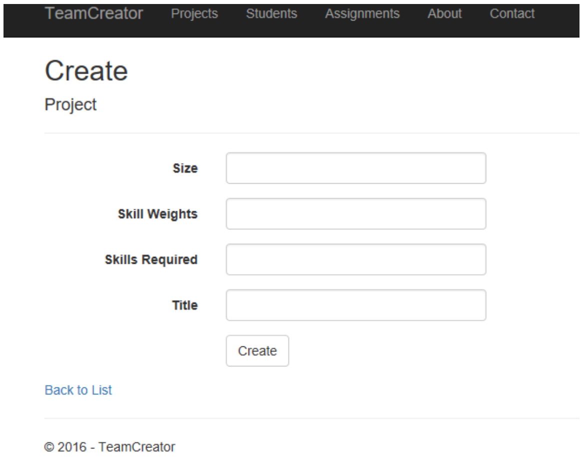


Figure 6: Project information entry screen

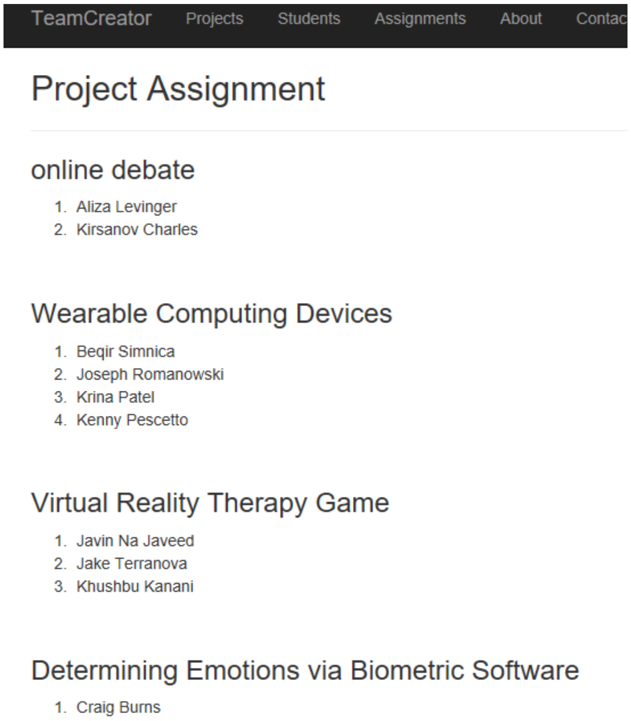
Figure below shows sample screen for entering client information (to be implemented in future version of the system)

Client Information: Enter client information

Client Name	Project Name	Project Description	Email	Address

Figure 7: Client information entry screen

Figure below show generated group report



Project Assignment

online debate

1. Aliza Levinger
2. Kirsanov Charles

Wearable Computing Devices

1. Beqir Simnica
2. Joseph Romanowski
3. Krina Patel
4. Kenny Pescetto

Virtual Reality Therapy Game

1. Javin Na Javeed
2. Jake Terranova
3. Khushbu Kanani

Determining Emotions via Biometric Software

1. Craig Burns

Figure 8: Team Assignment Report

Data Model:

Data model focuses on what information should be stored in the database for the functioning of the system. So data modeling is performed as the first stage of the database development. During the requirement analysis phase of the system, all required data is identified. An ideal data model accommodates enough data to support current system as well as future scalability of the system.

MS SQL Server database will be used to store project, client, and student information. The generated student groups for projects will also be stored in the database. The users and passwords will be stored in the database table.

List of tables in the database:

Users

Name	Data Type	Constraint
UserId	Int	Primary Key
Password	Varchar	
Role	varchar	

Students

Name	Data Type	Constraint
StudentId	Int	Primary Key
firstName	varchar	
LastName	varchar	
EmailId	varchar	
Seemster	int	

Projects

Name	Data Type	Constraint
ProjectId	Int	Primary Key
Title	varchar	
CourseId	Int	
NumberOfStudent	Int	
ClientId	Int	Foreign Key

Skills

Name	Data Type	Constraint
SkillId	Int	Primary Key
StudentId	Int	Foreign Key
ProjectId	Int	Int
SkillDetails	Varchar	
SkillLevel	Int	int

Clients

Name	Data Type	Constraint
ClientId	Int	Primary Key
ProjectId	Int	Foreign Key
ClientName	Varchar	
ClientMail	Varchar	
ClientAddress	varchar	

Groups

Name	Data Type	Constraint
GrpupId	Int	Primary Key
ProjectId	Int	Foreign Key
ProjectName	Varchar	
StudentName	varchar	

Preferences

Name	Data Type	Constraint
StudentId	Int	
ProjectId	Int	
PreferenceNumber	Int	

Following figure shows (proposed) final version of Entity – Relationship diagram for the system.

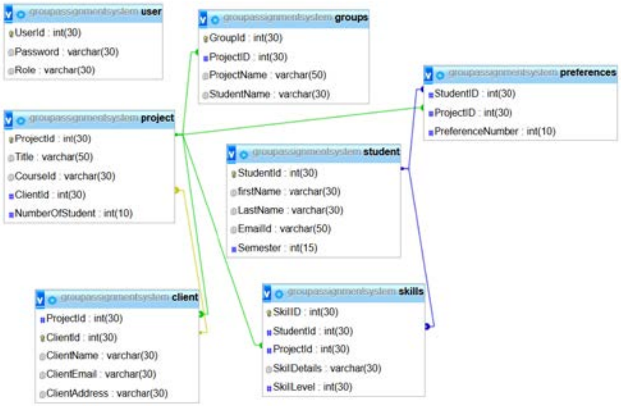
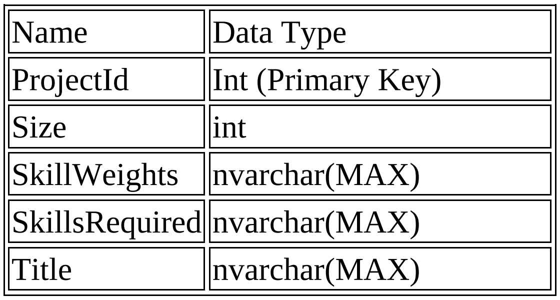


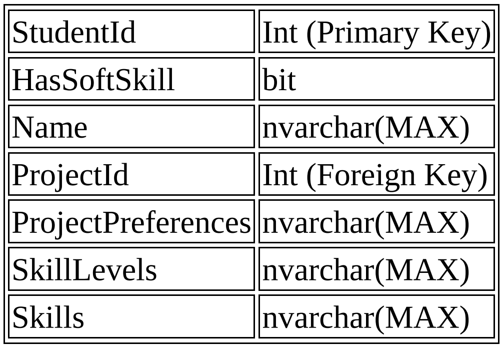
Figure 9: Entity-Relation diagram

In the current version of the system projects and students tables were used. The implementation of the tables in SQL Server is shown in the following figures.



Name	Data Type
ProjectId	Int (Primary Key)
Size	int
SkillWeights	nvarchar(MAX)
SkillsRequired	nvarchar(MAX)
Title	nvarchar(MAX)

Figure 10: Projects table



StudentId	Int (Primary Key)
HasSoftSkill	bit
Name	nvarchar(MAX)
ProjectId	Int (Foreign Key)
ProjectPreferences	nvarchar(MAX)
SkillLevels	nvarchar(MAX)
Skills	nvarchar(MAX)

Figure 11: Students table

Main Classes:

Projects.cs – to process and save projects data

Students.cs – to process and save students data

GreedyAssignmentManager.cs – greedy algorithm class to assign students to projects.

IV. RESULTS

To verify the validity of the system, forms will be distributed (in excel format) to students to make the survey, i.e. input their skills and skill levels, project preferences. The faculty will also be given an input form (in excel format) to input the project lists and required skills and skill levels for each project. The faculty could manually distribute the students among the projects to create groups for each project using the same logic as proposed. While creating student groups, the algorithm will be strictly followed instead of subjective judgments. Although faculty could have his/her own understanding of individual students, for the sake of testing the algorithm, the steps will be followed only with the algorithm. After the manual process is complete, the same data will be used in the system.

After running the process, system will generate report with the student groups for the project. If the two results match (manual distribution list and system generated) then we could be confirmed that the system works.

The student survey data has been collected through web forms of the system, and projects data for the current and previous semester were available.

For the testing of the developed system, sample data from the previous semester were input through the input screens. After running the process, the results were available through web interface. The results were compared with the teams that could be come up manually following the steps of the algorithm. The tests showed similarity of the outcome. Hence it could be concluded that the team assignment algorithm could be used in assigning students to projects in future capstone projects.

It should be noted here that although the two results give same result, still it is worthwhile to use the system for group assignment tasks, because manual assignment process is a laborious work that requires faculty's involvement of time and effort. The system will reduce the extra burden of the faculty of time and concentration for these kinds of jobs.

V. CONCLUSION

The proposed group assignment system will take project information as entered by the faculty and student information through web-based interface. The system will calculate the assignment with those data using greedy algorithm.

The automated group assignment system will reduce faculty's work and also it could be expected that student satisfaction will also be higher as student's preference is also taken into consideration.

The Team assignment system performs the job of automated group assignment fairly accurately. This system could be used for group assignment of future capstone projects along with overriding options.

VI. FUTURE WORK

The system has options to fully implement the user registration and authentication of appropriate users. Also, the student survey screen and project screen need to be developed to further validate entered data for accuracy.

The database could also be expanded so that the skill sets will not be static, rather depend on project information entry. From the entered project information, the system will identify required skills for the project for the specific semester.

Moreover, Group assignment algorithm could be applied to non-academic settings also. Group based work is common in workplace settings. The system could be extended to accommodate team assignment tasks in industry setting or administrative task assignment processing. The same algorithm with slight modification could be used in different kinds of team formation problems. Engineering projects, for industrial production system, and many operational jobs require appropriate team assignment among sections. People of diverse skills and skill levels are usually employed in such works. Use of such system may also reduce the risk of unfairness and reduce the limitations of subjective judgments for team member assignment.

REFERENCES

[1] Tappert, C.C., Cotoranu, A., and Monaco, J.V., "A Real-World-Projects Capstone Course in Computing: A 15-year Experience", *Proceedings of Conference on Information Systems and Computing Education (EDSIG)*, Publisher, Location, Nov 2015

[2] Robert D. Plumley, Charles C. Tappert. A GREEDY ALGORITHM ASSIGNMENT OF CAPSTONE COURSE STUDENTS TO TEAMS AND PROJECTS USING SKILL HEURISTICS Seidenberg School of CSIS at Pace University

[3] Mark Chang, Allen Downey. AC 2008-1961: A SEMI-AUTOMATIC APPROACH FOR PROJECT ASSIGNMENT IN A CAPSTONE COURSE

[4] Cooper, J. Group formation in cooperative learning: what the experts say. Cooperative Learning and College Teaching 7 (1996), 14-15.

[5] Cottell, P.G., and Harwood, E.,M., Background knowledge probes: an example of synergy between cooperative learning and classroom assessment techniques. Cooperative Learning and College Teaching 7 (1996), 6-8.

[6] S.L. Tanimoto, "The Squeaky Wheel Algorithm: Automatic Grouping of Students for Collaborative Projects," Proc. Workshop Personalisation in Learning Environments at Individual and Group Level in Conjunction with 11th Int'l Conf. User Modeling, pp. 79-80, 2007.

[7] M. Beheshtian-Ardekani and M.A. Mahmood, "Development and Validation of a Tool for Assigning Students to Groups for Class Projects," Decision Sciences, vol. 17, pp. 92-113, 1986.

[8] J.S. Dyer and J.M. Mulvey, "An Integrated Optimization/Information System for Academic Departmental Planning," Management Science, vol. 22, no. 12, pp. 1332-1341, 1976.

[9] R.R. Weitz and S. Lakshminarayan, "An Empirical Comparison of Heuristic Methods for Creating Maximally Diverse Groups," J. Operational Research Soc., vol. 49, pp. 635-646, 1998.

[10] D.-Y. Wang, S.S. Lin, and C.-T. Sun, "DIANA: A Computer-Supported Heterogeneous Grouping System for Teachers to Conduct Successful Small Learning Groups," Computers in Human Behavior, vol. 23, pp. 1997-2010, 2007.

[11] K.R. Baker and S.G. Powell, "Methods for Assigning Students to Groups: A Study of Alternative Objective Functions," J. Operational Research Soc., vol. 53, no. 4, pp. 397-404, 2002.

[12] A. Gogoulou, E. Gouli, G. Boas, E. Liakou, and M. Grigoriadou, "Forming Homogeneous, Heterogeneous and Mixed Groups of Learners," Proc. Workshop Personalisation in Learning Environments at Individual and Group Level in Conjunction with 11th Int'l Conf. User Modeling, pp. 33-40, 2007.

[13] Michael A. Redmond, Department of Math and Computer Science, La Salle University "A Computer Program to Aid Assignment of Student Project Groups"

[14] Roland Hubscher "Assigning Students to Groups Using General and Context-Specific Criteria" IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, VOL. 3, NO. 3, JULY-SEPTEMBER 2010

[15] Redmond, M.A. "A Computer Program to Aid Assignment of Student Project Groups", in Proceedings of the 32nd SIGCSE Technical Symposium (2001), ACM, pp134-138.p

[16] Designing Web Applications https://msdn.microsoft.com/en-us/library/ee658099.aspx