Teamerizer!

Bringing Groups Together.

Team 1: Ninad Anil Chemburkar, Vijay Sunder Gatike, Robert Harkov,
John Madias, Jimen Moy, John Pelliccia

Seidenberg School of Computer Science and Information Systems
Pace University New York, USA

Abstract – With the growth in capstone course across many graduate curriculums the task of selecting groups has grown as a manual process for students and professors. Keeping track of all the groups, students' skills, interests, prior experience and current groups has become a daunting task for all parties involved.

Although there are proven manual methods that work the question is how these methods can become more efficient and shifted to being part of an all in one solution that can handle all the steps of group creation and selection. This paper discusses a replacement to current methods and the design of such an application.

1. Introduction

Capstone courses are becoming a fundamental part of secondary degrees, "The aim of a capstone course in computing is to familiarize students with how their trade is applied in organizations, so that the program of study delivers the practice part of the promised theory and practice" [4]. In the computing industry work or applications are completed in teams which is synonymous with capstone courses involving groups of students, working together to meet course deadlines in order to achieve an optimal grade and experience. Students at this point in their education may not be familiar with the concept of team projects when it comes to real world applications and find it difficult to create groups in a classroom of students whom they may have never interacted with before. The idea for Teamerizer stemmed from the idea that students do not know the students around them including their skills in programming, which is extremely import when it comes to completing a successful group work. If a team is formed with students who all have the same skills or inadequate skills certain components of the project will be completed inefficiently. Teamerizer when coupled with curriculums that have capstone projects allow students to build their academic profile throughout the duration of their degree within the application. This includes students entered skills, but also academically verified skills in order to prove the entered skills and levels are valid. By the time students reach their capstone course, their profile will be comprised of all the skills they have learned throughout their degree.

Teamerizer can then be utilized on your first day of your capstone course when a student must create groups. The group creator select's the necessary skills and then through a matching algorithm they are presented with students whose profiles best match the skills needed for the group and can then invite the student to be part of their group. By using Teamerizer the groups skills will be diverse rather than a group that has all the same skills or a group of students with inadequate skills required for the group.

2. Literature Review

Stephan Laguette wrote that a "successful student team should include enthusiastic, motived and engaged students" [3] in order to have a successful team. A team should include members with "satisfactory skills, technical or academic expertise required for a project" [3]. Laguette argues that not enough attention is given to the team selection process and that many methods of team selection end in random self-selection or professor selection two of the common methods of team selection. Aller, Lyth and Mallak alternatively present a study by (Brickell, Porter, Reynolds, & Cosgrove, 1994) that shared interest and motivation are a big contributing factor to team performance, these interests are discovered using the mingling method for group creation.[1] Both present a question of how students are able to determine what there project should be and who should be in their group. How do they determine the interests and motivation of other students and in turn determine the student's skill level in those areas? We can implement the suggestions laid out by Aller, Lyth and Mallak including the "Mingling Method" or we can implement Laguette's which includes students providing their top three project preferences as well as filling out a self-assessment form on their skills for the

professor to later assign members to a team based on both data sets.

An analogous question becomes apparent with these authors suggestions. Is there a more efficient way to implement group selection, proving your skills and presenting them in a meaningful manor so a group can be created comprised of individuals with the correct skillset and mindset to complete their capstone course. The solution, Teamerizer. Teamerizer provides a platform where students can input their verified skills and in the future their interests and then either be selected for a group by a group leader or ask to join a group that has availability. In fact, Laguette performed a study at the University of California, Santa Barbara where students were assigned to groups based on their skill set and interests, this way student groups were formed so that each group had a mix of skill sets which in turn translated to higher performance across all group members.[3]. This supports the claim that if students had a platform where they could not only find groups that interest them, but also know that they would be joining a team of individuals whose skills fit the need of the group. Instead of spending time guessing who should be in your group, selection will become more of an enjoyable process engaging with students who are motived to creating a high performing team.

3. Current Methods of Group Selection

In most cases as presented in this section most groups are formed by specific methods as deemed fit by the university typically avoiding random selection which should be avoided. In every case there is a manual process of compiling students' skills, interests and then reviewing the information in forming groups. Stephen Laguette's solution at the university of California, Santa Barbara is described as "simple and effective" [3]. Laguette's proposed method begins with a description being written for each group which provides students with an understanding of what the project will entail and the expected outcomes of the project. Students are asked to review the list of projects prior to the semester and submit their top three choices. Students also provide their skills, experience, and interests. All the information that students' input is available to be reviewed by other students and students are encouraged to propose teams based on what other students post. The selection process then shifts to the professor who manually reviews all the provided student information and picks which students should be in each group based on their skills, so the teams have a balanced set of skills and interests.

The alternative approach proposed by Aller, Lyth and Mallak is coined the "Mingling Method". Like Laguette the projects are decided by faculty advisors who present the projects to the students on the first day of class in form of a problem, vision on design and expected results. After this is complete students then discuss the projects in class with the capstone course professor, asking any question and raising their interest. This allows other students to see who is interested in which project and with who to follow up with. The next step is students introduce themselves and speak towards their past experience in groups, skills and interests. Next is the "Mingling Exercise"[1]. During this stage students walk around and speak with potential partners asking about the skills they have, previously presented projects and finding students with similar interests. Through speaking with students directly connections can be made, and students can judge who they will work best with. Students are then asked to pick their top three choice projects and write them on the board. The next step is for students to formalize their choices by submitting a written bid for their top three choices, this bid, like Laguette method, includes their prior experience, teamwork skills, interest in each choice and their technical skills. This is all reviewed by the professor who assists with the team assignments in the case the groups are over the limits.

Based on these findings you will note that both methods are a manual and involved process, but both have been successful in the authors studies. Each process has the potential to be improved with an application such as Teamerizer where students can input all their skills, past experience, interest's and in the future faculty can then review this data if the project demand's faculty group creation. Students can then view available groups, those groups members and decide to be in a group or be selected to be in a group based on their profile information. This in the end solves one of the main problems of manual group creation by bringing everything group related all into one place.

In the remaining sections we will discuss the goals, design principles, application architecture data model and conclusion, which all support our main goal of making group selection simple yet innovative.

4. Architectural Goals

Teamerizer's architectural aim is to provide students a way to make conscious decisions of who they should work with for group projects based on a clear and concise user interface. The following are several key features of the application.

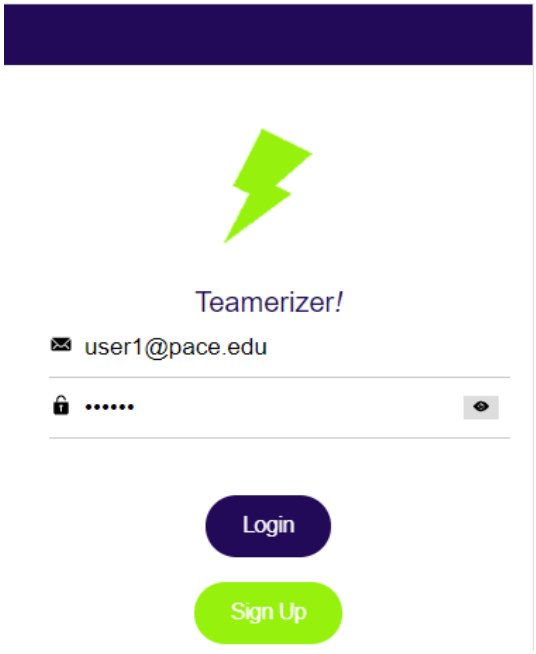


Fig. 1 User Log-in

- *User Log-in* – The application is designed around students, so users must first log in to gain access to the application. This allows for future development with validating students and assigning to class groups based on email domains.

Fig. 2 Member Details

- *Team Creation* – Users will be able to create a team including a description of the group and the skills required for the team.
- *Matched Members* – After creating the team the team creator will be presented with users who match on the skill sets. They can then add the member to the team. In future iterations of the application we will create a request functionality where users can accept or reject the team invitation.
- *Member Details* – Member details will present information to the user about the member that is selected anywhere in the application. This includes their name, skills and other details about them to be added in the future such as current groups or past groups.

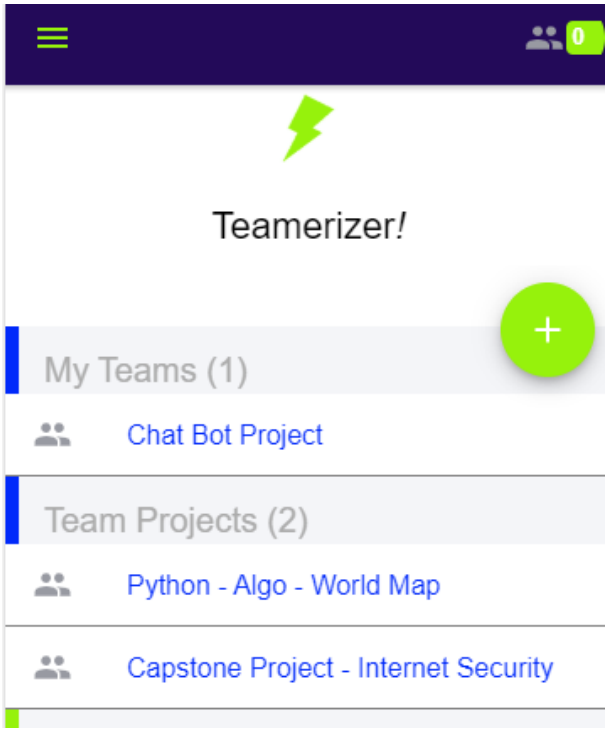


Fig. 3 Main Page

- Group View (Main Page) – The main page will display users' current groups, but also give the option for users to look through groups that still have availability using the search feature.

- Notification Page – The notification page presents two types of notifications to users. The first is notifications from group creators who invited the user to be part of their group. The second is if the user had created a group other users can request to be part of that group. In both cases the user can accept or decline the invitation.
- Profile Page – The profile page will allow for users to update important information on their profile. This includes update their picture, name, skills and resetting their password.

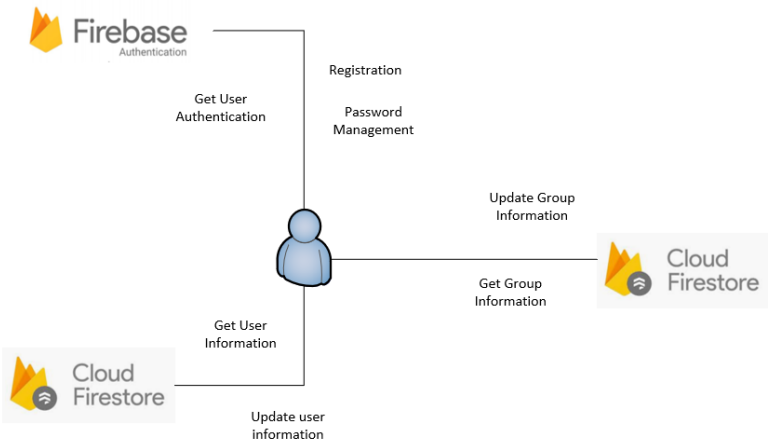
5. Design Principles

In order to provide our users with a seamless experience our application will adhere to the design principles laid out below covering two main areas

1) Application user interface
    a. Pages will have a consistent user interface for customer satisfaction and avoid confusion. This includes the layout of the menu bars and buttons so that users know where things are with minimal effort.
    b. Consistent methods of displaying user information to users.
    c. Adhering to consistent methods of user data entry.
    d. CSS will be consistent across the application such as backgrounds, layouts, fonts, spacing and ionic layout used.
2) Database data handling
    a. Data will be collected from users and written to the database in a consistent manner
    b. Confidential data will be kept confidential using google firebase API
    c. Database reads will be carried out in a consistent manner for all data

6. Topology Diagram

The following diagram displays our systems architecture in relation to the user interface and the database.



Interactions Explained

Firebase Authentication – Manages user sign-in and user registration through the firebase API, which handles encryption, storage and management of user credentials.

Cloud Firestore – Cloud Firestore is Googles noSQL cloud database that stores information in defined collections. Firestore will store our user information, group information and allow for searching on the data to find the right groups or group members.

7. Application Architecture

The application architecture of Teamerizer is critical to the use of the application as it is what bridges the gap between the database layer and the UI while providing users with the appropriate content. The Teamerizer application is built with the Ionic framework and coding is done in Angular and CSS for styling. The driving decision behind this decision is that it is platform agnostic and can be compiled into an iOS or Android application making no need to manage two separate code bases.

Presentation Layer

Represents the user interface and how the primary user interacts with the application. The presentation layer is written in HTML and uses CSS for styling to bring a clean and user-friendly UI to the users. Angular events are used throughout the presentation layer to trigger events when users interact with buttons and submit data.

Content Layer

The content layer is driven from Google Firestore. This layer presents users with their user information, group information, other member information and updates to that information.

Database access Layer

The database layer is driven by our user server and written in angular which handles all the user storage events and retrievals from the database. This layer also includes a group service which serves the same purpose, storing and retrieval group information from the database. User authentication and updates are also handled on this layer in a secure method.

Database Layer

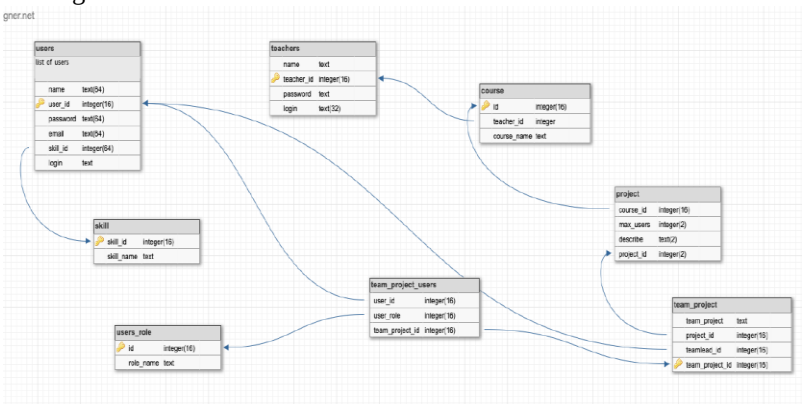
At the database layer user authentication, user information and user groups are stored. User authentication is encrypted by Google so that even admins cannot see passwords, admins can however manage users including, deactivating, password resets and deletion. Stored user information will be used for our back-end algorithms for matching members to the correct groups based on their skill sets. Groups will store group information and members that are part of those groups, this will be presented to members on the main page.

8. Database Architecture

The decision to go with an online highly available and scalable database was simple since both features are highly sought after in the application market and the application needed a reliable storage model.

Data Model

Data in a noSQL database is stored as documents, within those documents we have a collection of items. For example, users will be in a collection which will have a list of users. Below is the general scheme to our database.



Collections

A brief summary of our user collections is included below.

1. The Users collection will be used for storing all user information and have a reference id which will link it to Googles user authentication. This collection will also store reference to the groups that a user is part of, if any. Other metadata included is, name, email, skill, skill level and any future items added.
2. The Groups collection will be used for storing group information including the following: Name, description, max users, skills and a sub collection of current group members.
3. The Courses collection will have a list of courses. This collection will be used to link users and groups.
4. The Skills collection will comprise of all of the skills that are available to users to be selected in the application.

9. Conclusion

High performing teams of students with the appropriate skills are vital to capstone courses not only to achieve a decent grade, but for students to have the best group experience. For this to be achievable students and professors need an efficient way to share their skills, interests and past work in order to select or be assigned to a group. Teamerizer not only enhances the group creation experience, user selection, group availability, knowledge of student's skills and interests but also takes the burden of the professor manually compiling all students' profiles.

In future iterations of the application we will expand functionality to include professor access so that professors are able to review all the student's skills, interests and group choices so they can pick the group members instead of student self-selection. Another enhancement in the backlog to be completed is a way to test users skills so that they can become verified and this can be displayed within the app to alert others that they are not simply creating skills they may know.

In conclusion in its current iteration Teamerizer provides an invaluable platform to students for creating groups and finding group members who share the same interest and have the skills required of the group.

10. References

[1] Aller, Betsy M., et al. "Capstone Project Team Formation: Mingling Increases Performance and Motivation." *Decision Sciences Journal of Innovative Education*, vol. 6, no. 2, July 2008, pp. 503–507., doi:10.1111/j.1540-4609.2008.00190.x.

[2] Chang, Mark L, and Allen B Downey. "A Semi-Automatic Approach for Project Assignment in a Capstone Course." *Proceedings of the American Society for Engineering Education Annual Conference and Exposition 2008*, 1 June 2008, https://olin.tind.io/record/1514142/.

[3] Laguette, Stephen. "Development of High Performance Capstone Project Teams and the Selection Process." *Proceedings of the 2010 ASEE Annual Conference*, 2010, pp. 1–4., http://capstoneconf.org/resources/2010 Proceedings/Papers/Laguette.pdf.

[4] Tappert, Charles C, et al. "A Real-World-Projects Capstone Course in Computing: A 15-Year Experience." *2015 Proceedings of the EDSIG Conference*, 2015, pp. 1–10., http://www.csis.pace.edu/~ctappert/papers/2015EDSIG.pdf.

[5] Whalley, Jacqueline, et al. "Student Values and Interests in Capstone Project Selection." *Proceedings of the Nineteenth Australasian Computing Education Conference on - ACE 17*, 2017, doi:10.1145/3013499.3013508.