ABSTRACT
Given the importance of teams in project-based learning, many instructors hope to create well-balanced teams, based on individual student characteristics. However, there is difficulty in creating dozens of teams while accounting for multiple factors such as students’ schedules. Existing approaches for solving constraint satisfaction problems can be complicated, time-consuming, and rigid. Instructors often abandon the principled approach to team composition in favor of randomly formed or self-formed teams. This paper introduces a novel tool called Teammatic, a mixed initiative interface for instructors to work in conjunction with an algorithm that facilitates the team formation process. Teammatic accounts for specified constraints to create effective student teams. To test our tool, we interviewed instructors about the challenges and reasons for teams in their courses. In our study, 14 teaching assistants used Teammatic to produce teams that satisfied various constraints in significantly less time with higher compatibility scores than manually created teams.

Author Keywords
Team formation; mixed initiative; constraint handling

ACM Classification Keywords
H.5.2. User Interfaces: User-centered design

INTRODUCTION
As teamwork has increasingly become part of standard practice in industry and education, more research has focused on what makes teams most effective [15,24,30]. Each individual person on a team holds their own unique set of characteristics and with the right combination of individuals, a team can work more effectively together [9]. Previous research has shown that students learn proper teamwork skills by being placed on effective teams [23]. There are two common approaches to team formation in the classroom setting: randomly-formed teams and self-formed teams. Although it is simple to create teams with these two approaches, the results do not consistently produce teams that successfully achieve the goal of building teamwork skills [9, 20].

Existing tools for team formation leave room for improvement. For example, the CATME Team-Maker algorithm utilizes student data to create teams according to criteria specified by the instructor, but the tool’s poor usability make it difficult and time consuming [20, 28]. The CATME team formation tool gives different combinations of teammates as alternative solutions, but does not allow for shuffling people across teams. This kind of flexibility would be useful for applying tacit knowledge about particular combinations of people. For example, an instructor may speculate that two students have a bad social dynamic and may want to seamlessly separate them during the formation process. Existing tools do not explicitly support a dynamic relationship between automated constraint solvers and human facilitators.

This paper introduces Teammatic, a mixed initiative user interface that allows instructors and other facilitators to create teams with specific constraints in mind. With Teammatic, instructors can input constraints that will be used to form teams through a constraint solver, and maintain the freedom to move students across teams as they see fit. The mixed initiative interface allows instructors to work with an algorithm to solve for constraints in conjunction with their own expert knowledge for refinements of the teams.

This paper makes the following contributions. First, we conducted an interview study with 8 instructors to investigate current team formation practices and needs. We learned that instructors tend to avoid manually forming teams in part due to the amount of time it would take to complete this task. Second, we developed a mixed initiative interface for team formation called Teammatic that allows instructors to choose which constraints they want to consider when creating teams. Third, to evaluate the team formation process for Teammatic, we conducted a study with 14 teaching assistants where participants were given a dataset of students to create teams that satisfy given a list of constraints. We found that with Teammatic, participants could create teams in significantly less time with higher compatibility scores, and also feel more confident in the assembled teams. Participants also found it helpful to get suggestions for students to swap from one team to another.
RELATE WORK
In this paper, we draw on literature that focuses on factors that affect team performance and current practices for team formation in the classroom setting. We also review recent research on mixed initiative approaches to solving complex constraint problems.

Key Factors for Effective Teams
With the increased demand of teamwork required in industry, skills learned within the classroom setting become significantly more important. Group projects allow students to learn key skills, such as teamwork, problem-solving, communication, and leadership [12]. There are several different educational benefits students can gain from working with one another in a team. Collaboration allows students to gain a better understanding of the subject material, as well as encourage cognitive growth [9]. Gaining proper teamwork skills involves proper student interaction with other team members.

However, research shows that placing students in ineffective teams, characterized by conflict, unclear goals, or mismanagement, creates problems that are not beneficial to developing skills for the students involved [12].

Individuals gain proper teamwork skills when they are placed in an effective team rather than one filled with conflicts and disagreements [13]. Research finds that the diverse characteristics of individuals, such as their leadership qualities and technical skills, influence the team’s effectiveness indirectly through the nature of interdependent activity among team members [19]. Past work has focused on the performance of the team, given its specific composition of unique individuals [5].

Each member of a team has the potential to not only influence the overall team, but also each of the other team members [19]. While teams are affected by their members, past research has shown that there are other factors that can affect the team’s performance, including the complexity of the task at hand and overall group cohesion [11,27]. Studies have shown how conflict among team members is proven to be disruptive within a team [10]. This stresses the importance of creating effective teams for students to learn and develop essential skills. We will review some of the key factors hypothesized to effect team success.

Leadership
Leadership within a team has the potential to both positively influence the team learning and knowledge applications within a group [30]. Team leaders are essential within a group setting since they play a pivotal role in both maintaining the environment and coordinating tasks within a team [22]. The proper distribution of leaders is important to creating an effective team environment [32]. Adequate leadership on a team can provide guidance to other members, maintain organization, and foster collaboration with one another.

Demographics
Inclusion is important within the educational setting. For individuals belonging to a minority group, there is a high risk that these students may drop out of the given major or school all together [23]. Studies have indicated that when these individuals are isolated within a project team, they do not gain the benefits of working within a team because they are assigned the passive roles or choose to take on these roles [23]. Heller & Hollabaugh document the positive effects of heterogeneous groups, where students of the minority group feel more comfortable contributing [15].

Scheduling
Effective communication can make or break a team. When communication is limited, it becomes difficult to coordinate tasks among members, resolve conflicts, and develop interpersonal relations [12]. Studies have proven that both computer-mediated communication and face-to-face communication are an effective means of working together in a group [17]. However, it has been shown that individuals prefer to interact in a face-to-face manner over virtual communication when working with others [24].

It is important to place students in groups that allow for scheduled meeting times. By asking students information regarding their schedule, we are able to incorporate this constraint when we provide suggestions on team formation. Instructors have the ability to determine the minimum number of common time slots they want each team to have.

Strategies for Team Formation
Smartcrowd is an existing framework that focuses on improving collaborative crowdsourcing [29]. This framework takes into consideration various traits of the workers and the tasks assigned. Several different approaches were taken when implementing such an algorithm, including a greedy and an approximation approach. Our problem is similar to that of collaborative crowdsourcing in that we also want to find the most efficient way to group together individuals for the purpose of completing different tasks. However, these problems differ since crowdsourcing involves individuals who do not necessarily need to work together as a team, and can be completely unknown to one another [29].

Prior work has been conducted on forming teams online through social networks [3]. The algorithm proposed assigns various tasks for teams to deal with, while maintaining a fair distribution of work among the team and finding the minimum coordination cost. The work conducted by Anagnostopoulos et al. involves a static group of individuals who are connected within a social network. Conversely, our set of individuals differs in that many may not know one another and their set of skills can vary drastically.

Diebel focuses on team formation for in-class group work [9]. She describes the educational benefits that students gain from learning from one another in a group setting. She stresses the importance of student interaction and how formation of the teams can have a major influence on the quality of collaborative learning. Our work differs in that the teams we form can be extended for a longer period of time, in addition to the shorter term teams, where students will need to meet together outside of the classroom [9].

Existing Team Formation Platforms
CATME proposes a tool called Team-Maker that collects information from students, and utilizes the data gained to assign students to teams according to criteria specified by the instructor [20]. Creating teams through a system like CATME has shown success within various classrooms [21]. More instructors are looking to algorithmic tools for team formation [31]. The Team-Maker tool utilized in creating teams demonstrated results of a more effective composition than teams created manually [28]. When comparing the results between teams created by the Team-Maker tool and teams created manually, teams created by the tool had a higher compatibility score average, but scored no worse than the teams created by hand [8]. This compatibility score was determined by the constraints specified by the instructor.

Our work builds on this current approach by allowing instructors to have the power to move students around within the teams. While an algorithm can be utilized to form the initial teams and present suggested students to swap, instructors will be the ones to make the final decisions. In addition, we strove to improve the user interface to create a tool that is intuitive for our users to utilize.

Mixed Initiative Interfaces
Mixed initiative interfaces are a flexible interaction strategy where users of the interface work in conjunction with the automated system to contribute to the task [1]. Prior mixed initiative solutions have been utilized to solve several complex problems that involve scheduling, simulations, and management issues [7, 14]. Cobi’s scheduling tool is a mixed initiative interface that integrates preferences and constraints in order to build a schedule for a large conference [18]. Moreover, a mixed initiative solution was utilized to help complement a supervisor managing a team of robots [14]. Results have also indicated that users prefer the mixed initiative support in a study, where mixed initiative was utilized to help customize an interface [6].

Our proposed solution will allow professors to work in conjunction with Teammatic’s algorithms by applying their own knowledge and expertise to create teams for their students. We will adopt Cobi’s idea of offering suggestions for swaps to ease the process of swapping students from one team to another. In addition, we will integrate the idea of an adaptable interface by allowing professors to easily shuffle students from one team to another.

STUDY 1: CURRENT TEAM FORMATION EXPERIENCES
An initial study was conducted in order to understand the experiences instructors have had with team formation. We asked 8 professors, lecturers, and teaching assistants to describe the importance of teams in their classes. All recruits were interviewed from a west-coast university.

Method
Professors, lecturers, and teaching assistants were asked a series of questions in order to evaluate current procedures and tools utilized in forming teams. We targeted instructors who utilize teams within the courses they teach. The majority of courses were within the computer science field, but also consisted of cognitive science, business, and other engineering courses. Interviews typically lasted approximately 30-45 minutes. The following is a list of the initial interview questions we asked participants.

- What is the importance of teams in the classes you teach?
- How do you currently handle team formation?
- What are the current challenges in creating teams?
- What causes problems within teams?
- How do you resolve these problems?
- What can you do to prevent these problems?
- What traits of a team are the ones that are most successful?
- What do you think is important in team formation?
- What are the factors that matter?
- How would you form teams if you had a good tool for it?

Findings
Importance of Teams Within a Given Course
Teams present a number of functions for students within the classroom setting. One of the main reasons instructors choose to utilize teams within their curriculum is to prepare students for their careers in industry. Teamwork helps students build upon the skills that would be necessary for working with others. When we asked instructors why teams are important in the courses they teach, the following are examples of some of the responses we received.

“Teamwork is really important in software engineering and engineering in general. We want students to have that experience.” - Software Engineering Professor

“To prepare students for the workforce. These days in almost every industry you’re going to work as part of a team.” - Business Lecturer

Allowing students to work within teams provides the opportunity for them to learn from one another. Students will get the chance to ask each other questions and solve problems with their peers rather than directly going to the instructor for assistance. Teams enable students to engage in a collaborative learning environment. The instructors highlight:

“We want students to be learning from each other.” - Software Engineering Professor

“A lot of learning happens in a much better way when people collaborate in teams. A team is an opportunity to transfer skills from student to another.” - Cognitive Science Professor

Current Methods for Team Formation
Instructors apply various methods for handling team formation. While some may manually create assigned teams for their students, most allow students to choose their own teams. The reason most instructors do not selectively place students together is because of the amount of time it would take to parse through the data and determine how to form teams.

“Instructor-formed teams have the problem of taking up instructor time.” - Cognitive Science Professor

Each course can range from having around 40 students to 200+ students, and given the amount of time professors have, it is not feasible to manually go through the information and
group students together. If teams were assigned, most would randomly assign students together.

“Groups are formed randomly.” - Computer Science Teaching Assistant

However, allowing students to create their own teams or randomly assigning teams may not produce the best results or provide the best learning experience for those in the class. Selectively placing students together, based off of data gathered, creates an improved team environment where students can develop necessary teamwork skills.

Challenges and Problems with Team Formation

Given the interview responses regarding their past experiences, it was found that instructors face similar challenges. Teams typically failed when there was not an even distribution of work for the given project and when members had conflicting goals. It is the instructors’ assessment that this boils down to the amount of commitment each member of the team wants to devote to the course. Instructors noticed that when members of the team have the same level of commitment to a course, the team does not encounter as many problems.

Challenges also occur when students do not interact well on an interpersonal level. The behavior and personality of each individual has an effect on the overall relationship between members of the team. From the instructors’ past experiences, the problem arises when students do not address the issue early on in the project and only bring up the conflict towards the end of the project. We received the following responses when asked what instructors believe make a successful team.

“Successful teams are a combination of having the same level of commitment to success in the class and being able to manage the personality complex.” - Computer Science Professor

“Successful teams are the ones that have a shared goal and have a shared process.” - Cognitive Science Professor

Desired Components of a Team Formation Tool

Flexibility was a key component that instructors look for in a team formation tool. Instructors say they would utilize a tool where they can easily gather data from their students and form teams given the instructors’ own constraints. However, each constraint varied between each class and customization was key. Given the idea that instructors should form teams, many stated that they would consider this option only if shown the value of this process over their current process. Instructors would want to understand the reasoning behind why a given constraint is important within a team rather than blindly trusting a tool.

TEAMMATIC: A MIXED INITIATIVE APPROACH TO FORMING TEAMS

The proposed solution is to create a tool that will make it easy for instructors to form teams in the classroom setting. This tool will be a mixed initiative interface where instructors will have the ability to work in conjunction with the Teammatic algorithm to form teams. Our user is the instructor and the system will be the interface used to form the teams. Having the user work with the system will allow a collaboration that will help to refine the results of the teams formed [16].

Design Features

Teammatic allows instructors to import data about students within their class. The current version of Teammatic handles student availability, leadership attribute, gender, role distribution, topic interest, whether or not students own a vehicle or special devices (e.g., tablets), and student preferences for possible teammates. Once the data of student information is received, instructors will have the ability to choose constraints of how they want to create teams. The algorithm used for the team suggestions will account for these various constraints. After a suggested group of teams have been formed, the instructor will have the ability to modify these teams to his or her liking.

Importing and Filtering Rosters

The student roster displays all the data gathered from the students of the course. Instructors will have the ability to sort the list of students in a variety of ways to easily find pertinent information. Figure 1 displays the interface of the student roster and the details of each student. In addition, Teammatic provides a way to filter out certain data attributes for all students in the course. In Figure 1, we see filters that are available given the constraints currently accounted for in Teammatic. Once a filter button is selected, the selected attribute will not be shown for all students displayed in the roster. Figure 1 displays an example of selecting certain attributes to hide in the roster.

Selecting and Specifying Constraints

Before deciding how to create teams, instructors will have the ability to input various constraints of how they would want the teams formed. In addition, they can determine how they want to prioritize a certain constraint over the others. Figure 2 displays the constraint modal where instructors can input the constraints they desire. Current constraints available through
Teammatic include ensuring a minimum number of overlapping time slots per team, distributing roles across teams, placing students together or apart based off of given preferences, ensuring that leaders exist in a team, and balancing genders amongst the teams.

Creating and Evaluating Teams
Once the instructor has determined the various constraints they want to account for and inputted how many students they want in each team, they can select the “Create Teams” button to form a suggested group of teams. Figure 3 displays the output after all teams have been created. Teams are displayed in two columns, and Teammatic enables instructors to drag teams around as well, as collapse teams to only show the team name.

On the upper right hand corner of each team is a compatibility score. This compatibility score is determined by how well each team satisfies the given constraints. If a constraint is violated, a yellow warning sign will appear on the team header. Figure 5 shows an example of a team that violates a constraint, and details about what constraint is violated.

In addition, each team contains a calendar that displays the overlapping time slots of all members on the team. By clicking the calendar icon on the team header, a calendar will appear to show where these time slots occur. Figure 4 displays what this calendar will look like. Each blue rectangle on the calendar represents a one-hour time slot where all members on the team are available.

Swapping Students
After the teams are formed, instructors will still have the ability to move students as they see fit. By clicking on an individual student, Teammatic will provide suggestions of whom to swap with, while maintaining the highest score average for both teams involved. Figure 5 displays an example of a suggested swap. The students highlighted in green are the ones Teammatic suggests to swap with the student highlighted in blue.

Algorithm
We began by defining a simple model for our algorithm on how to create the teams. We take as input, data that represents the students characteristics. Given the various data points obtained about each student, different constraints can be stated on how to create the teams. The objective of our algorithm is to create teams that maximize the overall compatibility score average.

Our objective with this model is to maximize the score average across all teams, while minimizing the standard deviation. We want to obtain this maximum average while satisfying all given constraints. The given characteristics and constraints that we are to consider in the above example are subject to the instructor’s discretion, and may vary depending on what the instructor is looking to have among teams in his or her class.

Defining the Score Function
For each team, we will calculate its score based off the characteristics of the students that compose the team. As previously mentioned, there are certain characteristics that contribute to creating an effective team. The team formation algorithm will select the teams that have the highest scores. Constraints can fall under one of the following categories.
Increasing the Amount of Overlap: Instructors can strive to group students by increasing the amount of overlap of a particular constraint. For example, instructors may want to create teams of students with similar schedules.

Decreasing the Amount of Overlap: Instructors may also want to add a constraint that involves decreasing the amount of overlap between the students on a team to increase diversity. For example, having a constraint to distribute roles across teams can fall under this category.

Special Rules: If a constraint does not fall into one of the above categories, instructors have the ability to create a constraint that may be more customized to their criteria. For example, a constraint may be created to ensure that an individual of a certain minority group is not isolated on a team [23]. Instructors may create a constraint to ensure that there are either 0 or 2 or more individuals of the minority group in each team.

The score function will be the sum of the chosen characteristics with each characteristic multiplied by a weight (ex: w1*schedule_overlap + w2*gender + w3*leadership + w4*role_distribution). This weight, wi, represents how much instructors will want to weight each characteristic when determining how to create teams. The score function is what will ultimately be used to evaluate the compatibility score of a team. Teammatic utilizes the compatibility score to determine which teams to select.

Selecting Constraints
We provide a way for instructors to input their own constraints, relative to their course. Ensuring that each student belongs to only one team and that each team has the correct number of students are the hard constraints that are satisfied for all teams created with Teammatic. In addition, if the instructor wants to honor student preferences, they can choose to make this an additional hard constraint.

All other constraints are flexible given the preferences of the instructor. Instructors will determine how important each constraint is when creating teams by assigning a weight to each constraint. These constraints will be incorporated into the score function to determine the compatibility of a team.

Implementation

Initial Team Formation
Once we have received the data of each individual student, we will specify which constraints to take into consideration. These constraints can include the minimum number of overlapping times within a team’s combined schedule, balancing out gender within each team, ensuring that one leader exists within each group, distributing the roles across teams, or allowing students to specify preferences for teammates. Each of the teams is given a score based off of the importance assigned to each constraint and how well they satisfy the requirements. The algorithm will compute teams on the objective of maximizing the overall score average within the entire set of teams.

To select the teams, we utilize a greedy approach. We first generate all possible combinations of teams with the given team size. Next, we filter these combinations with the hard constraints. Currently, student preferences is the only hard constraint accounted for. We take into consideration all the students and the given constraints as inputted by the instructor. The students who are most constrained will be placed on a team first. By most constrained, we mean the students who placed in the least number of teams when generating all possible combinations. We select the highest scoring team with the student who is most constrained. Once the team is selected,
we eliminate all students who are placed on the selected team. From there, we repeat the process of selecting the team for the next most constrained students with the highest possible score, until all students have been placed on a team.

Given the idea of incorporating constraints, it is possible that some students are left remaining without a team. With the remaining students, we loop through the already selected teams and place these students on a team that generates the highest score, while still maintaining the list of constraints.

Swapping Students
Once the teams have been formed through the algorithm, if instructors want to swap students from one team to another, Teammatic provides suggested swaps of students. However, instructors are not forced to swap students only based off of the suggest swaps; they are the ones who make the final decision.

The swapping algorithm takes into consideration which student the instructor chooses to swap, their current teams, and the current constraint list. It will then provide suggestions with other individuals in the course which maximizes the compatibility scores of both the teams if the swap were to occur.

Constraint Violation
Given the dataset of student information and the list of constraints chosen by the instructor, there are times where constraints may be violated when creating the teams. Once the teams are created, we check to see which constraints are violated and provide the feedback of each constraint violation for the corresponding team on the interface.

Performance Testing of Algorithm
To evaluate the effectiveness and efficiency, we ran the algorithm on several different constraints with varying team sizes. We measured the amount of time it takes for Teammatic’s algorithm to output results and the overall compatibility score average of the resulting teams.

Figure 6 displays the various performance metrics that result from the algorithm described above. Several different constraints were inputted to create teams from a data set of 52 students. Our algorithm shows the variance in times to compute teams based off how many students the instructor specifies to place on each team. The larger the groups, the longer it takes to create a list of suggested teams. The reason behind this is because our algorithm starts by computing all possible combinations of students before determining which to place on a given team.

The average scores of the teams created are not affected by the team size. The algorithm utilized selects teams that maximize the overall average score.

STUDY 2: EVALUATING THE TEAM FORMATION PROCESS
A controlled study was conducted in order to gain insight behind manual team formation processes compared to Teammatic. 14 teaching assistants from various courses were recruited to participate in the study and were contacted via email. All teaching assistants involved currently or have previously taught a team-based course.

During the study, we observed how teaching assistants interacted with Teammatic as well as methods they utilized in order to satisfy the requirements provided to them to form teams.

Method
Instructors began the study by reading instructions for a manual process for team formation (see Appendix). They were given time to manually create the teams with the Teammatic interface. The dataset that was utilized in this study derives from real student data of a course. Once the instructors felt that they were satisfied with teams they had formed, questions were asked regarding what they found difficult, their thought process of creating the teams, and their confidence level in the manual teams they created.

Next, the participants were shown the automated teams created by Teammatic. We asked how they felt these teams compared to their own and how confident they were in the teams formed automatically. Participants were then given a scenario that described two students who mutually preferred to work together, but as the instructor, they have prior knowledge about the students and know that when the two students are placed together, they tend to act dishonestly. It was then observed what the participant would do in this situation, and how they would change the automated teams.

We concluded the study by asking the participant to evaluate their experience and results with the manual team formation process compared to that of the automated version in a short interview.

Results
The study produced positive results and demonstrates how Teammatic improves the team formation process for instructors. It was hypothesized that the scores of the manually created teams would be at most equivalent or less than the compatibility scores of the teams generated from the automatically created teams. In addition, we hypothesized that with Teammatic, the amount of time it would take to create these teams would be significantly improved. Both of these hypotheses were proven true.

Teammatic Created Teams with Higher Compatibility Scores
When given the same constraints as the participants of the study, Teammatic produced teams with an overall average
Figure 7. With Teammatic, participants created teams of higher compatibility scores than teams formed manually.

compatibility score of 98.1%. The average compatibility score for the manually created teams was 84.3%. We sampled different combinations of student characteristics, and given the same constraints as the test above, Teammatic produced an average compatibility score of 91.5%. An analysis of variances was performed with condition (Manual/Teammatic) as a factor and compatibility score as the dependent variable. Teams created with Teammatic had significantly higher compatibility (91.5%, SD=0.5) than manual teams (84.3, SD=1.2) (F(1,26)=11.23, p<0.05). The data gathered also shows that no individual was able to produce an overall average score between the teams with as high of a percentage.

Teammatic Created Teams in Significantly Less Time
This task took participants an average of 30 minutes 50 seconds to complete. With the automated tool, forming teams with the same requirements takes an average of 44.7 seconds. Given the same constraints with different data, Teammatic takes an average of 43.7 seconds to form teams (see Figure 8). When asked what the participant found difficult in creating teams, the most common response was that it was hard to keep track of all the information. Given the many constraints listed in the instructions, many chose to ignore the soft constraints and focused their time on the hard constraints listed. After a certain point in the process of creating teams, it was a common trend that participants would start to randomly place students together without any regard to the constraints given. Teammatic eases the process of team formation and provides a solution to handling several different constraints and data when creating teams.

Teammatic's Mixed Initiative Interface Allowed Participants to Easily Swap Students
Participants were given a scenario with two students in the dataset as explained above. Since mutual preferences place students on the same team, Teammatic automatically accounts for this constraint. Participants were asked what they would do to handle this situation. Results show that all participants would want to separate the two students and prevent them from being on the same team. To do so, many would randomly swap students or they would look for students with similar attributes, but had concerns with maintaining the several other constraints in the requirements. To handle this situation, Teammatic provided suggestions of who to swap students with, keeping in mind the constraints listed.

Participants Were More Confident in the Teams Created with Teammatic
Teammatic focuses on improving the experience of the instructor when forming teams. By measuring the confidence participants had in the teams formed, we were able to evaluate the qualitative difference between the manually formed teams and the teams formed automatically through Teammatic. When asked about the confidence of the manually formed teams on a scale of 1 being least confident and 7 being most confident, participants rated themselves with an average score of 4.3, whereas they rated the teams formed by Teammatic with an average score of 6.2. An analysis of variances was performed with condition (Manual/Teammatic) as a factor and confident scores as the dependent variable. Participants were more confident in teams created with Teammatic than the teams created manually (F(1, 26)=20.12, p<0.05) (see Figure 9). All participants rated the confidence of the teams formed by Teammatic either equal to the teams formed manually or greater. The reasoning many had was that Teammatic was able to successfully handle all constraints given in the requirements when the participants could not.

DISCUSSION
Teammatic provides an alternative improved approach to creating teams within the classroom setting. We have demonstrated the benefit of strategically placing students together, and how current manual processes make it difficult to achieve results quickly. Teammatic creates teams with higher compatibility scores in significantly less time when compared with manual team formation processes. In addition, participants were more confident in the teams created with Teammatic when compared to the manually created teams. With the mixed initiative interface, participants forming the teams have the power to move students around and to work in conjunction with the algorithm Teammatic provides. Experimental results have demonstrated the advantages that Teammatic provides over manual processes and have shown positive results when comparing compatibility scores, time, and confidence.

However, we realize that there are still some limitations with Teammatic. First, we have not yet evaluated the performance of the teams formed, since this would require significantly more time and several instructors willing to form teams with Teammatic. Second, our compatibility measure is defined by the constraints used to create the teams. Another approach to this would be asking experts of the teams themselves to judge compatibility. We currently have not implemented this approach since this would also require a longer time span and several instructors willing to participate. Third, we have a fixed set of constraints, and while it is easy to code in additional constraints, we could further improve Teammatic’s interface to handle any type of constraint.

FUTURE WORK
In future work, we plan to incorporate further interface improvements, as well as add more features to the system. Currently, Teammatic does not have a fully integrated process. It would be ideal to incorporate the entire team formation process into Teammatic, from getting data from students to forming teams to peer feedback at the end of the course. We plan to have a fully integrated system before releasing Teammatic to allow others to use.

Furthermore, we seek to add additional features to improve the experience of creating teams. Currently given that Teammatic utilizes a greedy algorithm, it is capable of generating only a single option for instructors given a data set with certain constraints. Teammatic currently is not able to generate different combinations of teams given the same constraints and student information. By switching from a greedy approach to an optimization approach, we can provide multiple options of team suggestions for those utilizing our tool. In addition, as we have previously evaluated, the speed of the algorithm can be further improved for larger groups of students.

It would also be helpful to allow instructors to lock in certain teams they want to keep and rerun the algorithm to create new teams with the remaining students. In addition being able to add customized constraints will allow for better flexibility for instructors to create the teams they desire. By pursuing these changes, Teammatic will provide an even better approach to the team formation process.

We have shown the benefit of utilizing Teammatic to create teams over manual methods. To further validate the benefit of Teammatic, additional studies would be useful. A longitudinal study is required to understand the performance of teams created with Teammatic over old practices. We hypothesize that by utilizing Teammatic, teams will perform better than the teams generated through old practices.

Another study could be conducted around the performance of team formation created during a given course. We could evaluate the teams formed for ad hoc or short-lived activities in addition to the more long-term projects. As such, Teammatic could be utilized as a platform to test different theories involving team formation.

CONCLUSION
Strategically creating teams of students with a given set of characteristics helps to build their teamwork skills. This paper introduces Teammatic, an innovative mixed initiative interface for team formation. Instructors currently find it difficult to manually assemble dozens of teams keeping in mind several different student attributes, as this process can be complicated and time consuming. However, Teammatic makes the team formation process more efficient and creates more effective teams. Teams created with Teammatic have higher compatibility scores, and individuals feel more confident in the teams formed. With the mixed initiative interface instructors are able to work in conjunction with the algorithm to apply their knowledge of the students and easily swap students from one team to another wherever they see fit. Teammatic has the potential to improve how instructors create teams within the classroom setting.

APPENDIX
Instructions Provided to Participants in Study 2
Pretend that you are the teaching assistant for a project based course. The professor of the course wants you to create teams with the following requirements. These are hard constraints when creating teams:

- 4 students on each team
- Ensure that each student is on one team
- Student preferences are honored

The instructor also wants you to create teams with the following soft constraints in mind to the best of your ability. Each of these constraints are equally as important to this instructor:

- Teams with distributed roles (ex: a team has a programmer, designer, and analyst rather than all programmers)
- At least 1 leader on each team
- Either 0 females or 2 or more females on each team
- At least 6 overlapping time slots in common