Spelman College®
A Choice to Change the World

CIS 472, Project Overview

Project Deliverables				
Iteration #1			Iteration #2	
Report #1	Report #2	Demo #1	Report #3	Demo #2

1. Purpose and Method

The purpose of the course project is to provide the students with the knowledge of software engineering methodology and the skills to apply it. The particular project is not the goal in itself; rather, it serves as a vehicle to apply your knowledge and to develop the skills.

Projects also introduce students to teamwork, which is unavoidable for large-scale software development. Teamwork has positive and negative aspects, and a familiarizing yourself with both helps you get ready for your future workplace. There are many challenges to good teamwork: time coordination; building on each others strengths and compensating for weaknesses; reconciling different working styles, preferences, and levels of motivation; etc. A key challenge is to make possible group efforts that discern contributions of each individual. This is a challenge in the course, where each student needs to be assigned a grade, as well as in your future workplace where your raise and promotion will depend on your perceived contributions.

This project consists of two iterations of a software product. The first iteration is *exploratory* and represents the first attempt at developing the proposed software product. The deliverables for the **first iteration** are reports #1 and #2 and demo #1. After these, the instructor will provide feedback and the students should reconsider and possibly revise the project goals before moving on to the second iteration. Keep in mind that it is perfectly acceptable to modify your objective in the middle of the semester, once you learn more about the project and have better understanding of what you can accomplish within the semester timeframe. This is why we have two (2) iterations, so that in the second iteration you can perform the necessary adjustments, based on what you learned in the first iteration. The deliverables for the **second iteration** are report #3 and demo #2.

1.1 Course Project

For this semester, you are to create a jobs portal for employers and jobseekers as a Java app. This job board must contain search filters, job functionalities and suitable interfacing with both sides of the employment spectrum: employers and jobseekers. Examples of existing jobs portals include: Indeed, Glassdoor, Monster, Dice, Blendoor. Each may be considered as a model for the team. Existing desktop-based, web-based or mobile-friendly based templates can not be used in the construction of this software.

2. Teamwork and Project Management

Each team consists of 4 to 5 students working on the same project. Teamwork is **required** since team work is an integral part of large-scale software development. Larger teams should in principle be able to develop more sophisticated products. However, larger teams are also more difficult to coordinate and manage. It is more difficult to split the work in larger teams and ensure that every team member plays a meaningful and important role. Also see the grading policy below.

You should form a team by September 12, 2019 @ 12p and **notify by email** the instructor about the following:

1. Team member names and email addresses
2. Your project web site URL (one web site per team)

All team members must take part in all project activities and **none** of the activities should be done exclusively by one student. Each team member or a pair must be responsible for all aspects of development required for the features they own, although it is acceptable to seek help from colleagues who are more knowledgeable about particular technical work (e.g., database integration, user interface design, etc.).

Project management deals with organizational matters that are needed for effective teamwork. Team members may elect one "team leader" who will take a lead in project coordination activities, or may decide to share coordination responsibilities. Project management includes the following:

- Organizing group meetings and keeping track of deadlines
- Managing shared resources (such as website, database, software repository, etc.)
- Integrating individual contributions in a coherent manner, and resolving ambiguities and conflicting information, including
    - collating different sections into a project report representing the entire team
    - proofreading the collated report to ensure consistent layout, font styles, section numbering, and language style
    - integrating different parts of the program into one software system
    - running integration tests and ensuring that the whole system works as intended
    - fixing software bugs that result from incompatibilities between software modules contributed by different team members
- Anything else that affects everyone in the team

You will soon realize that teamwork is difficult. Here are some suggestions (none of this is required!) about teamwork:

- Choose the team leader (see role description below) to serve as a focal point. Discuss individual skills and strengths (algorithm design, coding, report writing, etc.)
- Set the meeting agenda and time limits. Once the task responsibilities are known, in the beginning of the meeting have each student report about the progress on their task since the

last meeting. If you are working on a project report, everyone should hand out drafts of their section or show slides of their diagrams or screen shots. Students are encouraged to confer frequently and work together on solving the problem

- Decide how will the team communicate, e.g., email, text, GroupMe, Slack, ... As you make progress on your task, send drafts to others and ask for comments/suggestions. Do not wait until you have everything polished and finalized, but make them aware that this is a draft and tell them when to expect the next version
- Make every effort to ensure that *all* team members feel comfortable about each others' contributions; be open about grievances. It is a good idea to be collegial and professional in your communications and avoid conflicts. However, you want to be open about any issues, because your final grade depends on it.
- If in the days before a project deadline you are prevented (travel, illness, ...) from attending the team meetings, immediately notify all your team members and try to contribute to the best of your abilities.
- If for whatever reason your contribution falls short of that of other team members, you should do extra work in the subsequent deliverable(s). This does **not** mean that you will ask your teammates to do less work so that you can compensate your previous shortfall. No. Rather, you should let them do their work as they usually would, but you should do extra work that will make your deliverable brilliant.
- Plan for disaster scenarios: often team members depend on each other's individual contributions. Set firm internal deadlines (well ahead of the actual deadline) for contributing individual parts, so that if a team member fails to deliver, the others will have time to take over and complete the missing part.
- General policy: **If in doubt, communicate!** Redundancy is OK. Every time you send an email related to the project, *copy the email to all team members*. Do not assume that they are not interested in it or they know what you are talking about. *Acknowledge* any electronic communication specifically targeted to you.

Team meetings should be structured to involve:

- Report on the progress since the last meeting
- Discussion of tasks that need to be done—come up with specific action items
- A period of time to assign responsibilities, draft the parts each person is responsible for, and ask questions to be sure you all are on the correct path

Tools to help organize the teamwork:

- Doodle, GroupMe, OpenProject, etc.

Saying that "nobody asked me to do this or that," or, "I did everything that I was asked to do" is an **unacceptable excuse**. Each team member should be **proactive** and not wait passively to be assigned responsibilities. Do **not** ask others what should be done; rather, take initiative and suggest what should be done to make your project successful. Take every opportunity to redistribute and/or

rotate the responsibilities, make your personal suggestions be heard! Many times defining the problem and determining what needs to be done is more difficult than actually doing it. Hence, problem defining and task assignment must be contributed to by all team members, rather than by the team leader alone.

The following free online e-book contains useful advice on how to work on student software engineering projects:
Tips to Succeed in Software Engineering Student Projects
Author: Damith C. Rajapakse, National University of Singapore
Publication Date: 27 June 2008.

What is the role of a team leader?

First, keep in mind that having a team leader is **optional**. You will soon realize that there are many things that need to be coordinated and having a single central point-of-contact may be helpful. You should elect the team leader only if you believe this would facilitate your group work. Alternatively, you can appoint individuals as needed to coordinate meetings, deadlines, and track project progress, and rotate these roles among the team members.

The role of a team leader should not be misunderstood. The team leader may take lead in "project management" (i.e., providing organizational and logistical support, such as organizing team meetings and keeping track of work progress). The team leader is **not** expected to set the objectives, partition the tasks, and devise solution strategies. These responsibilities must be shared equitably among all team members.

If you choose so, it is a good idea to select the team leader based on his or her leadership, organizational, and social skills, rather than their technical skills and knowledge. Past experience has shown that people with poor organizational/social skills turn out to be poor leaders and their teams end up being dysfunctional, regardless of their technical knowledge.

The leader is **not** accountable for "failing to lead" and cannot be blamed for the lack of communication skills or general lack of success of the team's project. For example, the leader has no such responsibilities as "knowing what needs to be done," assigning work loads (fairly or otherwise), or distributing the responsibilities. This must be agreed upon by consensus. Most importantly, ***e*ach team member must be proactive**, rather than waiting to be assigned the duties by the team leader or anybody else.

Remember, all team members are **equally** responsible for **all** aspects of the project.

There are neither benefits nor responsibilities for the team leader. The team leader will **not** receive any rewards (e.g., higher grade) for serving in this capacity. You will notice that a certain number of grade points for each deliverable is allocated for "project management." However, these points will **not** be automatically allocated to the team leader. All teams are required to submit a breakdown of contributions.

What if your team is not functioning well?

If you notice that your team does not function well or the team and this could negatively impact your project performance (and your final grade), you should make every effort to discuss this with other team members. If the "problematic" team members (including the team leader) refuse to cooperate, you should discuss your concerns with the instructor, the sooner the better.

3. Individual Contributions Breakdown

Many students find team work difficult due to different personal interests and working habits. Therefore, each student should keep track of his or her contributions to the project. The exact **breakdown of individual contributions** must be provided for every project deliverable, so that individual grades can be fairly assigned.

Here are a few likely comments about group work:

- "It turns out that some groups have forced one student to do most of the work because they are not very knowledgeable with coding, or for other reasons. This is unfair, and students that want to work alone should be allowed to."
- "I didn't like the group project experience cause kids that do not know anything and do not the work to be carried on the other kids that are actually doing the work. Having to show a Responsibility Matrix in the reports is unfortunately unfair. Half the time, I had to skew and scale the results of other teammates in order to make it look like they contributed equally; meanwhile I had to spend hours of my own time writing over 9/10ths of each report and over 80% of the code."

First, course survey is not the place to provide such information. This information should be given in your contributions breakdown. Second, unfortunately this is how real world works. In your future workplace, you cannot expect to be nice to all your co-workers and say that everyone has contributed equally, but then expect to receive a raise or promotion that you "really deserve." You are not being genuine by outwardly promoting equality and secretly hoping for inequality (so that you get a better grade). The best approach is to be honest and report the contributions accurately. If you wish to help your colleagues, help them learn the course material better instead of doing their work for them.

The breakdown of individual contributions should be submitted:

- In case of written reports, as the second page of the report
- In case of project demos, each student submits individually to Moodle (a submission box will be provided) an itemized list of their own contributions only.

- **Each student** should provide an **itemized list** of his or her own contributions to the components of the particular deliverable, such as:
    - requirements specification (use cases and non-functional requirements
    - software design (whole system or list the specific modules),

- coding (whole system or list the specific modules),
- debugging (whole system or list the specific modules),
- report preparation (whole report or list the specific sections/diagrams),
- Other: any other relevant contribution.
  If several students contributed to a particular component, **quantify, as a percentage,** each student's contribution to this component. Also provide a short description of your own contribution.

Team members who are not listed in the joint breakdown or do not submit their contribution list will be considered as **not contributed** to the deliverable in question and will be assigned **zero credit** for this deliverable.

A potential problem has been observed in prior software engineering projects. These projects are sufficiently complex that it is impossible for a single student to do everything. Good teamwork requires division of labor and building on each other's strengths. Although on real projects the developers may split the work along their specialties, because this is an academic course, it is critical that all team members are engaged in all aspects of the project. Remember: the key purpose of these projects is for students to learn the class material through hands-on experience. Ideally, all team members should learn design, coding and testing the software and communicate their work through report writing.

Please keep in mind that you must inform the instructor accurately of who did what if you expect us to assign the grades fairly.

4. Project Resources and Communications

Online Tools for Software Development

You may find useful to consider Virtual Studio Code, an IDE for programming or IntelliJ, an IDE for professional Java programming, **CVS**, open source version control, **ANT**, a Java build tool and **JUnit**, a unit testing environment.

Bitbucket allows you to host, manage, and share Git in the cloud. Free, unlimited private repositories for up to 5 developers give teams the flexibility to grow and code without restrictions. Your team may elect to work directly on GitHub (preferred). DropBox and MediaFile may be suitable options for free file hosting.

5. Report Format

Three reports are required for the semester. A report is submitted by the whole team and will be graded as a whole, compared to other reports, and points will be split among the team members according to the declared contributions (details here).

A report must have a **cover page** containing:

- the course title,

- group name,
- project title,
- URL of the project website
- submission date, and
- all team-member names.

Negative points will be assigned to reports missing- or having an incomplete cover page.

The **second page** of each report must detail the **breakdown of individual contributions** to the project (use more pages if necessary). **Each student** should provide an **itemized list** of his or her contributions to components of the report, such as:

- requirements specification (use cases and non-functional requirements),
- domain modeling (whole system or list the specific modules),
- software design (whole system or list the specific modules),
- report preparation (whole report or list the specific sections/diagrams),
- Other: any other relevant contribution.

If several students contributed to a particular component, **quantify, as a percentage,** each student's contribution this component.

If you find it unnecessary and tedious to quantify details of your work, and if all team members agree that everyone genuinely contributed to the success of their project, it is acceptable that you just write "All team members contributed equally" instead of a detailed breakdown.

The reports missing- or having an incomplete contributions breakdown page will be graded accordingly.

The rest of the report must follow the detailed descriptions given in:

1. Report #1 document
2. Report #2 document
3. Report #3 document

Reports should be prepared using a word processor and printed on a (preferably) laser printer. Handwritten reports or reports which contain handwritten material (e.g., figures, tables, etc.) will **not** be accepted. It is easier for me if you turn in your documents electronically in PDF format.

6. Project Demos

The purpose of project demos is to complement the written project reports. See more on the demo format document.

Each team will give two demo presentations, each lasting about **15 min**. The demos will take place in the course designated classroom unless stated otherwise. If you are using your own laptop/ notebook to project the slides and run your software, please ensure that the video connector is compatible with the cables in the lab **before the actual demo**.

After the demo, each student should provide an itemized list of his or her contributions to components of the demo, such as:

- software coding (based on, but not including, the designs specified in project reports) list the specific modules,
- system integration,
- testing and debugging (whole system or list the specific modules),
- slide preparation and product brochure preparation,
- demo presentation,
- Other: any other relevant contribution.

If several students contributed to a particular component, **quantify, as a percentage,** each student's contribution this component.

If all members of the team feel that everyone contributed equally, you can just write **"All team members contributed equally"** instead of a detailed breakdown.

It is essential that you document the specific contributions for each team member. Some people are more comfortable with public presentation and quicker to answer questions, so the instructor may get impression that these students know the best and contributed the most. Hence, they will receive more grade points for the demo. If you did less-visible parts (coding, debugging, slide preparation, etc.), but you are not skilled with public presentations, then this is your chance to state your contributions. We cannot know what you did (and assign the grades fairly) unless you tell us!

7. Project Grading Policy

Given that this is a design project, it is impossible to define precise grading criteria. The grades will inevitably depend on what student teams have to offer—so, grading is "market-based." Those that offer the most-impressive product, receive the highest grade. Which means that, if you receive a grade you are not happy with, it is not necessarily that you did bad work; rather, it is that others did better. Of course, there may be several teams with an impressive product, so several teams may receive the maximum grade.

The grading policy for individual deliverables will be posted in the description of that deliverable. The specific requirements to be met will be listed. The reports will be graded based upon the technical content and clarity of exposition.

However, it is **not** enough to meet all the listed requirements to receive the maximum grade. For example, having a perfect report for a mediocre project will result in a low overall grade. Thus, the

overall perception of the project (relative to others) is the key scaling factor for all other aspects of the grade.

Each student must be aware that a major part of his or her final grade depends on the team project. The failure to cooperate with other team members and invest equitable amount of effort can lead to undesirable outcomes. It is essential that you accurately disclose the individual contributions if you expect us to assign the grades fairly.

Grading the project-related deliverables works as follows. Say there are three teams, the first team has four students, the second team has three students, and the third team has five students. Then

1. The project deliverable for each team is graded independently of each other. See, for example, how report #1 is graded. The maximum possible score that a team member can earn depends on his or her reported contribution breakdown. The actual earned points depend on the judged quality of these contributions.

2. Next, all team deliverables are compared as a whole relative to one another, and a holistic project grade is assigned.
   Let us say that Team 1 and Team 3 appear to have done about the same size project, and Team 2 worked on a smaller project, so it receives 70 %. The teams project grades are estimated as Team 1 = Team 3= 100 %, and Team 2 = 70 % (relative to the other two teams).