2XB3 - Final Project Description

Instructor: Reza Samavi

Feb. 16, 2015

1 General Project Description

Each student in the class will propose a software application project, present the project to the peers and form a team of three students to implement the software project subject to the instructor's approval. The software project should address an engineering problem and provide algorithmic solutions to tackle the problem. Examples of such applications are: a web service that finds the shortest path between two points, a web application that schedule final exams in a department, or a mobile application that remove repetitive pictures from a photo galley. Depending on the scope and complexity of the problem you are interested in, make your proposal focusing only on the back-end and application, the front-end or both. For example for a complex problem that you propose to implement you may focus only on the algorithmic operations and the back-end of the application rather than on the graphical user interface (GUI) and the front-end. In this case the application I/O can be managed using input and output text files. Nevertheless the focus of the project could be on the GUI aspects of an application if the actual algorithmic solution that will be implemented solves a graphical user interface problem (for example, improving performance of an algorithm to render a geometrical object on a mobile screen for a user to view the object).

The following resources will help you find an interesting problem as the topic of your project proposal:

- Chapter 2 (pp 336-358), Chapter 3 (pp 486-513), Chapter 4 (pp 515-517), and Chapter 5 (pp 695-701) of your Algorithms textbook as well as the creative problems of all chapters in this book provide you with a broad list of applications and problems in our real life that require algorithmic solutions.
- The recent McMaster undergraduate students deltaHack event can also help you come up with ideas for your project:

  http://www.cbc.ca/news/canada/hamilton/news/young-programmers-skip-sleep-to-hack-for-social-change-at-mcmaster-event-1.2958464
- For a list of more complex algorithmic problems that you may decide to make yourself familiar with you can check:

http://en.wikipedia.org/wiki/List_of_NP-complete_problems

The details of the project milestones and deliverables are described below.

2 Project Proposal

The first milestone of the project is the project proposal. The project proposal document is individual work and worth 10% of the course mark. **Project proposal will due on Feb 27, 23:59**. Each student in the class will propose a software application project of his or her own choice subject to the following rules:

- The application must require algorithmic contents in the application level (e.g., a web service that finds the shortest path between two points, a web application that schedule final exams in a department, or a mobile application that remove repetitive pictures from a photo galley).
- The application makes use of at least three implemented algorithms in the assignments and in-lab projects or learned in the CS/SE 2C03 class as the application modules.
- At least one of these three algorithms is an algorithm to solve a search or graph problem (e.g., minimum spanning trees).
- The application has a limited scope such that it can be implemented in four weeks by three students.

The project proposal document is limited to about 4 pages (title page excluded with the word limits stated below for each section) and contains the following items:

- **Project title:** a concise and meaningful title for the application, your name, student number, and course code and lab section number.
- **Abstract:** describes the highlights of the proposal (the problem, the solution, the dataset and experiments for verification and validation). (max: 150 words) Hint: you will write the abstract after writing your proposal!
- **Objectives and scope of the project:** concisely describe the problem that you are interested to solve. You should include a background study and the context of the application. You can use symbolic notations or figures to describe the problem. Then you will describe your tasks (what your implementation will accomplish) to address the problem. You will also include in this section your motivation for selecting this topic. You will describe why you think it is important to solve this problem. (max: 500 words)

- **Input/output:** describes the type of input datasets (and the source of the dataset) that you will be using to evaluate your application and the type of output that you will be generating. (max: 200 words)
  - Your Algorithms textbook website include a set of rich datasets for variety of applications that you can use:
    http://introcs.cs.princeton.edu/java/data/
  - Large network datasets are also available on Stanford University website:
    http://snap.stanford.edu/data/
- **Algorithmic challenges of the project:** describes which algorithms (sorting, searching, graphs, strings, etc.) you think you will be implementing to solve this problem and what are the challenges that you should be aware of. (max: 200 words)
- **Project timetable:** describes the timeline of the project with a list of milestones, and deliverables. The list should not exceed 7 milestones. You should use a table or a gantt chart to describe your timetable. (max word: 150 words)
- **References:** use a consistent format (IEEE, APA, etc.) for your references.

You will submit a PDF file as your project proposal containing the items discussed above to the "Project proposal" folder on Dropbox. The filename for your project proposal should be 2XB3_proposal_Lxx_LastName_initials.pdf where LastName and initials should be replaced with your last name and initials and Lxx with your lab section (01 or 02).

3 Project Pitch

After submission of the project proposal, every student has 4 minutes to pitch his/her proposal to the class. The presentation should cover the highlights of your project proposal. After each presentation other students in the class will use the course website to vote on their peers proposal. The number of votes will not impact the students grade for the presentation, however will help students in a team to pick a topic to work on as a team project.

The project pitch is individual work and worth 5% of the course mark (3% for the student presentation and 2% for voting other presentations). Each student will submit a Powerpoint or PDF presentation slides (not more than 6 slides) to the "Project pitch" folder on Dropbox. The filename for your project presentation should be 2XB3_pitch_Lxx_LastName_initials.pdf (or ppt) where LastName and initials should be replaced with your last name and initials and Lxx with your lab section (01 or 02).

The due date for submission of the presentation slides is Mar 1, 23:59.

All presentations will take place during the lab hours (depending on your lab section) on the week of March 2. It is required by all students to attend both sessions of their lab hours to present their proposals and also listen to the other proposals and cast their votes.

4 Team Project Implementation

4.1 Team Project milestones and deliverables

Milestone	**Deliverable**	**Date**
0. Team Assignment	List of team members	Mar. 6, 23:59 (via discussion board)
1. Selected topic, project objectives, project staffing and roles	Project topic, updated project objectives, roles assigned	Mar 13, 23:59 (via course website - Group dropbox)
2. Requirements specifications	Requirements document according to instructions received in SE 2AA4/CS 2ME3	Mar 27, 23:59 PM (via course website - Group dropbox)
3. Final project presentation slides	Powerpoint or PDF presentation slides	Apr. 5, 23:59 PM (via course website - Group dropbox)
4. Final project presentation	presentation to the class	Week of Apr. 6 (in lecture and lab hours)
5. Final project code	The Eclipse project of the implementation	Apr 10, 23:59 PM (via course website - Group dropbox)
6. Design specifications	Design document according to instructions received in SE 2AA4/CS 2ME3	Apr 10, 23:59 PM (via course website - Group dropbox)
7. Team peer evaluation (prepared individually)	Completed evaluation form for the other team members	Apr 10, 23:59 (via course website - Individual dropbox)

All activities to produce deliverables for Milestones 1 through 7 must be recorded in the Project Log (see format below) and submitted as an updated version of the document to the group dropbox. Note that the Project Log is a living document nothing is deleted or changed in the existing entries; rather new entries are appended to update/revise existing entries.

4.2 Team membership

After all project proposals are presented to the class, individuals will merge into three-member teams. All students in a team must belong to the same lab section. Students will announce their team members on the course Discussion List. The due date for announcing team membership is March 6, 23:59. Passed team membership due date, students who could not find a team, will be randomly assigned to a team by the instructor.

4.3 Teams and roles

After teams are formed, the first job of the team will be topic selection and project planning. Each team will collectively select a project topic. The topic should be one of the members topics presented in the class. The team members may consult the results of the peers voting to pick the topic. Although team members cannot switch to a new topic, they are allowed to change the scope of the project (most likely to make it narrower). Then you should assign team roles (e.g., researcher, designer, programmer, tester, etc.). Among the roles that you decide, two roles are mandatory to be assigned: *Project leader* who is responsible to manage the project to meet all the milestones and produce the prototype, and *Log admin* who is responsible to keep the log as a living document of the project. One person can play multiple roles. But all those roles must be assigned at the beginning of the project. If later during the project implementation for some reasons a role needs to be switched to someone else the decision making process must be captured in the project log.

The selected topic, modified project objectives, project staffing and roles, and the new project plan will be submitted to the assigned team dropbox before Mar 13, 23:59.

4.4 Project log

The Project Log is a record of the project: what happened, what was decided, what was implemented, etc. It could be any events relevant to the project, from an individual activity in programming a class to a group meeting on Google+, and so on. The log is an on-going document, written during the project not after the fact. If there are no entries in the log for a date it simply means that nothing has happened with respect to the project in that date. The project log must be updated and submitted not later than 24 hours after an event. The events cannot be backdated. The project log should always mirror the changes on your project. To do so, there should be one person responsible to keep this crucial document up to the date and submit it to the course website (group dropbox) as the project makes progress. The log document as a tool will help all members of the team stay up-to-date regarding the current status of the project and trace back to find out the root cause of a problem and internal accountability.

The project log file name should be GroupNN_Log_YYMMDD where N is your group number and YYMMDD is the date you are submitting the log file to the group dropbox.

Note that partial marks of your project will be assigned to the log, evaluating how well you have kept the log complete and up-to-date.

The Project Log has a header and a body as follows.

Header:

Team Name: ______________________
Topic: ___________________________

Team Members
Name Role(s)
1. ___________ ___________________________
2. ___________ ___________________________
3. ___________ ___________________________
4. ___________ ___________________________
5. ___________ ___________________________

Body:

The log document body contains an ongoing list of progress toward achieving the project goals. Note that any Task, Issue, Decision, and Changes to the tasks including the addition of new tasks should be recorded here. As described above this section should be updated and submitted on daily basis unless nothing is changed or done for the project in a day.

Timestamp	Originator	Type	Task ID	Status	Comments [sample entries below]	Supporting Document
140217T1223	All	Task	1.0	Completed	The project topic is	xxx.pdf
140218T1223	All	Task	2.1	Completed	team roles assigned	xxx.pdf
140221T1224	Bob	Issue	1		The selection of search algorithm discussed and	yyy.pdf
140228T1224	All	Decision	1		The Datset1 selected for running the experiments	
140228T1224	Alice	Change	1		Task ♯ n.0 changed from ...	xxx_v02.doc
⋮						

The timestamp format should be YYMMDDThhmm. Some log events may have supporting documents (for example the first job of the team which is the topic selection and project planning requires a document describing the roles, project plan etc.). In this case supporting documents should be uploaded to the Group Project Log Dropbox as soon as the event has been logged in this document. You should use meaningful names for supporting documents. For example you can use GroupNN_ProjectPlan.doc, for xxx in the sample log above where NN is your group number.

4.5 Requirements and design documents

Your requirements and specifications documents should be based on the format and standards described in your SE 2AA4/CS 2ME3 course. More specifically your requirements specifications should describe required behaviours in terms of functionality and performance with necessary rigour and formality. Your design document should include:

- a description of the classes/modules you have decided to use in your application, and your explanation of why you have decomposed the application into those classes; You should include a UML class diagram showing a static representation of your application classes and relationship between classes;
- for each class, a description of the interface (public entities), and make

sure that there is a description of the semantics (behaviour) of each public method in the class, as well as a description of the syntax;

- a view of the uses relationship;
- include a trace back to requirements in each class interface;
- for each class, a description of the implementation (private entities), including class variables - include enough detail to show how the class variables are maintained by the methods in the class; you should include two UML state machine diagrams for two most interesting classes in your implementation;
- an internal review/evaluation of your design.

The due date for Requirements document is March 27, 23:59 and for Design document is Apr 10, 23:59.

4.6 Final project code

Your implementation language will be Java and all classes and dependencies (including the input datasets) must be contained in an Eclipse project. You should document your code so that it is clear how the code follows its design, and also explain design decisions in the code that were not included in the design document.

You will submit an Eclipse project as your final project code. Your Eclipse project name should be GroupNN_FinalProject. You should include a txt file named 2xb3_GroupNN.txt containing the following information (each item in a separate line):

- The course code (COMP SCI 2XB3 or SFWR ENG 2XB3)
- Team members: student numbers, names and role(s)
- a dated statement that attests to ”the fact that the work being submitted by you is your group work.”

The due date for Final project code is Apr 10, 23:59.

4.7 Final project presentation

All members should be present and contribute in presenting the completed project to the class. The presentation will take 10 minutes and the requirements and design specifications as well as the verification and validation process will be communicated. Members of the team will take turn to present different aspects of the project.

You will submit a Powerpoint or PDF presentation slides (not more than 10 slides) to the group dropbox. The filename for your project presentation should be GroupNN_ProjectPresnentation.pdf (or ppt).

The due date for submitting the slides is Apr. 5, 23:59 and the presentations will take place on the week of Apr. 6.

5 Team Peer Evaluation

At the end of the project, each team member must individually submit a review of each team member. For each person reviewed, the review should specify the amount of interaction that the reviewer had with the person together with an evaluation of each of the following criteria on a 5-point scale (1=low, 3=average, 5=high):

- The effort that the person put into the project.
- The quality of the work performed.
- The person's professionalism in terms of meeting deadlines, doing their share of the project, being easy (and even pleasant) to work with, etc.

These ratings should be accompanied by written comments, justifying the scores given. You can also include the lessons you learned during your team work and your recommendations to improve the team dynamics.

Team peer evaluations are to be done independently and are confidential. No one but the instructor will read the reviews.

The reviews will be used by the instructor as part of the criteria to assign 20% of the final project mark individually to each team member. Provide your opinion based on your interactions with the person. The reviews are not a popularity contest: each person is expected to honestly evaluate and comment on the contribution of each team member. Abuse of the process will not be tolerated.

Final Note

Any milestone that is missed (i.e., turned in late or not done satisfactorily even if it is on time) costs you marks! Milestones are cumulative: you cannot achieve milestone ♯ 3 without having achieved ♯ 1 and ♯ 2. So if you do not satisfactorily complete a milestone, you need to resubmit it before the next milestone is due in order to be able to achieve the next milestone.

General rules that are applicable to your assignments and in-lab projects will be applicable to the implementation submission of the project too:

- A submitted solution that does not compile or run gets 0 credit.
- A solution that runs but is partially correct gets partial credit (depending on the progress towards a full solution).
- Providing adequate, concise, and meaningful comments throughout your code is part of the solution grade (i.e., a piece of code that correctly solves a problem without (or with inadequate) comments will score less than a well-commented piece of code that does the same).
- Not following the project instructions properly for the requested formatting will cost you marks.

- Every hour after a deliverable deadline 2% will be deducted from the mark assigned to that deliverable.

- The work you submit must be your own group work. Both copying projects and allowing others to copy your assignment are strictly forbidden and will be treated as an academic offence. All projects deemed to be substantially similar to each other will get 0 credit.

- If you include libraries from any sources other than your own or from the course material (course lecture notes and lab notes/instructions) you must acknowledge them and explicitly give proper credit with meaningful comments inside your code (when using methods from the external libraries). Properly cited external codes can only be included as Java libraries, i.e. you are not allowed to copy full or partial codes from other resources and include them inside your code. The included libraries should not be a substantial part of your implementation. Your work will be checked for plagiarism to account for this.