2XB3 - Final Project Description

Instructor: Reza Samavi

Jan. 7, 2019

1 General Project Description

The final project is an important component of this experiential course. The main learning objective of the final project is to prepare you for developing software for real world problems. The final project is a software implementation project with algorithmic content consisting of the components described below. By doing the final project, students will experience team-work in a situation similar to the real world software development environment. Early in the semester, students will form teams of four to five students. All students in a team belong to the same lab section. Although the final project has components that will be completed individually (i.e. project proposal and proposal pitch), students are always advised (and encouraged) to discuss their ideas and development of their individual work with other peers in their teams to receive feedback and comments. Each student in the class will propose a software application project, present the project to peers and then as a member of a team implement the software project. The software project the student is proposing should address a user or a group of users' *need* that can be formulated as a computational problem. You will provide algorithmic solutions to tackle the problem and address the *need*. 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 university without conflicts, an application that process number of Bitcoin transactions based on some criteria etc. The focus of your implementation proposal should not be on the user interface implementation unless the algorithmic challenge in fact lies in the user interface, for example, improving performance of an algorithm to render a geometrical object on a mobile screen for a user to view the object.

2 Pre-study for the project

Since you have to decide about your project topic and write a proposal early on in the semester, it is important to study the diverse application of sorting, searching, graph and regular expression algorithms described in your text book (provided below with the page numbers) before the actual implementation and the technical content of each algorithms being taught in your 2C03 course. This strategy not only helps you to come up with an idea for your project proposal but also will help you understand the contents of each algorithm better when they are taught in 2C03 as you have already studied the potential applications:

- Chapter 1 (pp 215-233), Case study: Union find
- Chapter 2 (pp 336-358), Applications of sorting algorithms
- Chapter 3 (pp 486-513), Applications of searching algorithms
- Chapter 4 (pp 515-517), Graph applications
- Chapter 5 (pp 695-701), Applications of processing strings

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. Further resources listed below will help you find an interesting problem as the topic of your project proposal:

- Apps Gallery of Government of Canada, Open government initiative
- 8 Blockchain Application Ideas That Could Help Your Small Business3
- 30 things you can do with a blockchain
- 5 great apps backed with open data
- Data Science: What are some good toy problems in data science?
- Your City Needs These 7 Open Data Apps
- This The Economist's article, Open Government Data: Out of the box is also very inspiring, particularly the section called: There's money in free stuff!
- The recent McMaster undergraduate students deltaHack event can also help you come up with ideas for your project
- A list of more complex algorithmic problems that you may decide to challenge yourself with

The details of project milestones and deliverables are described below.

3 Forming a Team

The final project implementation is ultimately team work. Nevertheless the project has components that will be completed individually (i.e. project proposal and proposal pitch). Therefore, the first milestone for the final project is to form a team of four to five students. Forming the team earlier will help you receive comments and feedback from your team members for the parts that need to be completed individually. A valid team has at least four students and at most five students. All students in a team must belong to the same lab section. Students will self enrol to a team using Avenue Group option. After self enrollment expiry date, unenrolled students will be randomly assigned to a group by the course instructor.

The team self enrollment expiry date is Jan 18, 23:59.

4 Project Proposal

The second milestone of the project is the project proposal. Every student in the class will propose a project of his or her own choice subject to the rules described below. The project proposal document is individual work and worth 10% of the course mark.

Any project proposal that are substantially similar to the other proposals or to the previously implemented projects by the students of this course in the past years will receive ZERO credit.

Project proposal due on Feb 1, 23:59.

- You should propose an implementation that makes use of at least one of the publicly available Canada Government open datasets or US government open datasets or Stanford Large Network Dataset Collection. An eligible dataset, if it is in CSV (or Excel) or ASCII text format should have at least 100K rows (tuples) and if it is a graph dataset, the number of nodes should exceed 500K.

- You project idea should address a real *need* for a user or a group of users. (e.g., a program or a web service that finds the shortest path between two points, an implementation that schedule final exams in a university such that there is no schedule conflict, or a mobile application that removes repetitive pictures from a photo gallery). You may identify the need from your own experience or from doing some research. The list of related News articles and sample projects links provided in Section 2 can help you to select a project topic.

- The need that you are proposing to address should require algorithmic content in the application level using at least one sorting, one searching, and one graph processing solutions. The focus of your implementation proposal should not be on the user interface implementation unless the algorithmic challenge in fact lies in the user interface.

- You should clearly define the objectives and more importantly the scope of the project you are proposing. You should well describe the scope of the project and demonstrate that the project (or specific aspects of the project) can be implemented by five students during the course time frame.

A template for the project proposal will be provided. The students should follow the template and clearly and concisely answer the questions asked in the project proposal template. The word limit cannot be exceeded for each box in the template however, students can add optional appendices to the template and make a reference to each appendix in the corresponding box. Please note that the text provided in the template should be sufficient to understand the project idea. The template can be downloaded from the course website and contains the description of the items need to be included in the proposal and word limits.

You will submit a PDF file as your project proposal containing the items discussed in the template 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 (e.g., 01).

Note:
1. You should add the following sentence in the title page of your project proposal:

By virtue of submitting this document I electronically sign and date that the work being submitted is my own individual work.

2. For the writing style you can consult the following sources:
http://homepages.inf.ed.ac.uk/jbednar/writingtips.html
http://twp.duke.edu/uploads/media_items/academic-style-guide.original.pdf
3. Using Freelancer or any similar services for any course deliverable is serious academic dishonesty and will be investigated with zero tolerance.

5 Project Pitch

After submission of the project proposal, every student has 5 minutes to pitch his/her proposal to the class. The presentation should cover the highlights of your project proposal and encourage others to invest in your project. After each presentation other students in the class will use the course website or i>clicker to vote on their peer's proposals. The number of votes will not impact the student's grade for the presentation, however, it will give team members an idea of which topic is favored by peers. The team may or may not choose to pick the topic with the highest vote.

The project pitch is individual work and worth 5% of the course mark (3% for the student presentation and 2% for participation in voting for 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 Feb 3, 23:59.

All presentations will take place during the lab hours (depending on your lab section) on the week of Jan 29. 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.

6 Team Project Implementation

Project implementation is teamwork. After all project proposals are presented to the class, the team collectively select a topic among the topics presented by the team members and plan the project. After topic selection and project planning the team will identify the requirements, create a design concept and implement it as a prototype. The project implementation must meet the requirements stated in the project proposal section of this document and will provide the following deliverables:

- Requirements specifications
- Design specifications
- Implemented project
- Project log

Finally, the work carried out in the project will be presented by all members of the teams to the class in a 10-minute presentation. The team work portion of the final project is worth 30% of the course mark. The details of the team's milestones and deliverables are described in the following subsections:

6.1 Team Project milestones and deliverables

Milestone	**Deliverable**	**Date**
0. Team Assignment	List of team members	Jan 18, 11:59pm
1. Selected topic, project objectives, project staffing and roles	Project topic, updated project objectives, roles assigned	Feb 10, 23:59 (via course website - Group dropbox)
2. Requirements specifications	Requirements document according to the instructions received in SE 2AA4/CS 2ME3	Feb 24, 23:59 (via course website - Group dropbox)
3. Project progress checkpoint	the first prototype demonstration	Mar 13-15 during the practice lab hours
4. Final project presentation slides	Powerpoint or PDF presentation slides	Apr. 7, 23:59 (via course website - Group dropbox)
5. Final project presentation	presentation to the class	Week of Apr. 8 (during the lecture and lab hours)
6. Final project code	The Eclipse project of the implementation	Apr 14, 23:59 (via course website - Group dropbox)
7. Design specifications	Design document	Apr 14, 23:59 (via course website - Group dropbox)
8. Team peer evaluation (prepared individually)	Completed evaluation form for the other team members	Apr 14, 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 Section 6.3) 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.

6.2 Teams and Roles

The teams are formed a few weeks before the actual work on implementation starts. Students in a team meet regularly depending on their plans and schedules however there is one mandatory weekly meeting that happens during the practice labs. In the practice labs students from one group will sit close to each other to facilitate peer cooperation and consultation in a quiet manner to avoid disrupting other teams. The first 45 minutes of each practice lab is assigned for group meeting. Members of each team will meet, discuss the project progress, complete the project progress weekly form (using the meeting minutes template) and communicate the issues such as team dynamics or any specific issues related to the final project (e.g., project topic selection, proposal, implementation, documentation). The course instructor and/or the assigned teaching assistant will be present to facilitate the meetings and address the questions or issues that may arise. For a number of practice labs, members of one team might be asked to meet with students from another team to exchange team work experience.

The official project implementation work starts right after completion of project proposal pitch by all members of a group. The first job of the team will be the 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., client, 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.

Each team member is responsible for contributing equally to the team work. The team leader will inform the instructor and/or the assigned TA on how the work within the team is distributed (as well as reporting in the designated section of the written report). Each student will receive a grade that is a combination of his/her individual grade and the grade for the entire team. The peer evaluation forms that the students will complete at the end of the group work will be factored into students combined grade.

The selected topic, modified project objectives, project staffing and roles, and the new project plan must be submitted to the assigned group dropbox by Feb 3, 23:59.

6.3 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 Dataset ??? selected for running the experiments	
140228T1224	Alice	Change	1		Task ♯ n.0 changed from ...	xxx_v02.doc
140230T1400	All	Team Conflict		Resolved	members discussed lack of cooperation by member xx and documented the issues in the meeting minutes	MM_yy.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.

6.4 Requirements Specifications Document

Your requirements and specifications documents should be based on the format and standards described in your SE 2AA4/CS 2ME3 course. The Requirements document is described in page 164 and page 394-396 of your Software Engineering text book. In a nutshel,l a requirements specification document defines the external behaviour of the system and can comprise of the following items:

- **The domain**. Brief description of the application domain and of the goals you should fulfill by developing an implementation. This includes a precise documentation of the domain knowledge that is relevant to derive specifications: who are the stakeholders and what are their goals and expectations? What are the main entities that characterize the domain? What are their main relationships? How are they affected by the system we will develop?
- **Functional requirements**. These describe what the product does by using informal, semi-formal, or formal notations or a suitable mixture of them (you may include use case document and a UML use case diagram and/or a class diagram).
- **Non-Functional requirements**. These may be classified into the following categories: reliability (availability, integrity, security, safety, etc.), accuracy of results, performance, human-computer interface issues, operating constraints, physical constraints, portability issues, and others.
- **Requirements on the development and maintenance process**. These include quality control procedures (in particular, system test procedures), priorities of the required functions, likely changes to the system maintenance procedures, and other requirements.

The case study in page 395-397 provides a good example for requirement specifications.

6.5 Design Specifications Document

The Design Specification document of your final project should have a common section at the beginning of the document that includes the following items:

- **Cover page**: Project Title, Version Number, Date, Group # and members, Course # and Course name, Department & University
- **Revision page**: Report revision history, Team members, student numbers, and their roles and responsibilities, and the following attestation and consent:

  By virtue of submitting this document we electronically sign and date that the work being submitted by all the individuals in the group is their exclusive work as a group and we consent to make available the application developed through [CS] or [SE]-2XB3 project, the reports, presentations, and assignments (not including my name and student number) for future teaching purposes.

- **Contribution page**: This page includes a table with four columns: Name, Role(s), Contributions, and Comments. Each row is assigned to one member of the group. This table will list contributions made to the project by each individual member of the group. You cannot repeat a contribution item for another member of a group. You should breakdown the contributions such that each contribution falls into exactly one row. You can add a comment column to clarify the contributions if it is necessary. The contributions listed in this table should be consistent with the project log.

  Note that this table SHOULD be completed based on consensus among the members of the group. Everyone in the group will agree with the assertions made in this table by the virtue of submitting the report to the group's dropbox folder.

- **Executive Summary**: An abstract about the project not more than 200 words.

- **Table of content**

The other sections of 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 Feb 24, 23:59 and for Design document is Apr 14, 23:59. Note that the Requirements document may change after the original submission due to the fact that your software development methodology is iterative and in each iteration you may identify a number of new requirements. The document that will be evaluated for the final project marking will be the latest version of the requirements document submitted to the group dropbox.

6.6 Final Project Implementation Files

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 14, 23:59.

6.7 Final Project Presentation

All members should be present and contribute in presenting the completed project to the class. 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_ProjectPresentation.pdf (or ppt).

The final project presentation will occur during the lab and lecture hours of week Apr. 2. Lab and class attendance during the final project presentation is mandatory.

The presentation should cover the following items:

- project objective, scope and motivation
- the open data set you used
- main requirements (functional and/or non functional) specifications of your implementation
- main design specifications of your implementation including the algorithmic challenges and major input/output
- the verification and validation methods of your implementation
- a brief demonstration of your implementation

The presentation should be shorter than 10 minutes. Members of the group will take turn to present different aspects of the project in the first 6-7 minutes and then the presenters will answer questions that might be asked by the course instructor and/or students. The team should plan the presentation such that enough time is given to each member of the group to present his/her part but the timing does not need to be exactly equal for all members of a group.

You will be evaluated based on the overall content of the presentation and your communication skills. You will use your own device to present your work to the class. Please make sure you have the required adapter to connect your device to the VGA cable.

The due date for submitting the slides is Apr. 7, 23:59 and the presentations will take place on the week of Apr. 8.

7 Team Peer Evaluation

At the end of the project, each team member must individually submit a review of each team member. The review will follow a strict format provided by the instructor. 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. Scores without comments will not be used. You can also include the lessons you learned during your team work and your recommendations to improve the team dynamics. 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.

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

The instructor will use the reviews as part of the criteria to assign 5% of the final project mark individually to each team member.

The due date for submitting the peer evaluation document is Apr. 14, 23:59.

8 Team conflict resolution protocol

Preventing conflicts is the best approach for conflict resolution! Dealing with conflict immediately, being open, practicing clear communication, not letting conflict get personal, focusing on actionable solutions, not looking for blame, demonstrating respect and keeping team issues within the team are only some of the tecniques that can be used to prevent conflicts at the first place. In case of a conflict in the team that the team could not resolve the following protocol needs to be followed:

1. `Document the issues`. The project log and meeting minutes are two documents that must be used to document any issues in the team. The project leader is the responsible member to ensure all issues are being documented. If a member is not punctuate or misses meetings without informing the team leader, the issues need to be reflected in the meeting minutes.

2. `Prepare for resolution`. Assign a specific meeting to the conflict and invite all member(s). Discuss issues one by one and document the meeting minutes. If a member is absent or late, this needs to be documented too. Clarify positions, list facts, assumptions and beliefs, analyze the situation in details and discuss facts underlying each position.

3. `Request a supervised meeting with your assigned TA`. If the conflict was not resolved in the previous step, request a supervised meeting with the TA and follow step 2 and document in the meeting minutes and the project log. The TA will help the team resolve the conflict.

4. `Request a meeting with the course instructor`. If none of the above steps helped resolve the conflict and if everything is being documented and submitted to the group dropbox, a meeting with the instructor can be requested to resolve the conflict.

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.