Concordia University
Department of Computer Science and Software Engineering
SOEN 490: Capstone Software Engineering Design Project
COURSE OUTLINE
Tse-Hsun (Peter) Chen

Office hours: Please request to schedule an in-person (or Zoom if you prefer) meeting for office hours. Feel free to use our SOEN 490 Slack workspace for any question you have.

Course Description:

SOEN 490 Capstone Software Engineering Design Project (4 credits)
Prerequisite: SOEN 390. Students work in teams of 8 to 10 members to construct a significant software application. The class meets at regular intervals. Team members will give a presentation of their contribution to the project. Lectures: one hour per week. Laboratory: two hours per week. Two terms.

Health and Safety

Students will complete the Minerva module on "Safety Management Systems and Leadership." Students will also comply with the rules and regulations governing the 490 lab space and servers.

Iterations	Evaluation criteria	Weight
Release 1 **Sep 6 - Nov 6** Sprint #1 Sep 6 - Sep 25 Sprint #2 Sep 26 - Oct 16 Sprint #3 Oct 17 - Nov 6	-**User Stories Backlog** (You should create user stories for all requirements. Before the beginning of each Sprint you will plan the user stories to be completed and estimate their user story points using planning poker): **3** points -**Agile Planning** (Each release consists of 3 Sprints and each Sprint is 3-week long. For each Sprint you will include the Sprint summary, burndown chart and retrospective. All actions in the retrospective should be justified based on your experiences from the previous sprints. You must use an Agile planning software, such as ZenHub or JIRA, which provides a Board and Agile metrics, and you must provide access to the instructor): **2** points -**Software Architecture** (Domain Model, Component Diagram, Class Diagram, ER Model, analysis of external libraries, justification of design patterns): **4** points -**UI Prototypes** (Mockups linked to each user story, all mockups will be annotated to explain the functionality of buttons/inputs/outputs, Persona descriptions, UI variations based on the personas, show updates to the UI mockups after receiving feedback from the product owner): **5** points -**Unit/System Testing Report** (Unit test coverage report for the current release, Record automated System/UI tests as shown in https://github.com/RGPosadas/WayFinder/wiki/Acceptance-Tests-GIFs): **3** points -**Issue Tracker** (Create bug reports with replication steps, screenshots, links to the commits fixing the bug for traceability, links to commits updating tests to cover the reported bug. All issues related to bugs should be tagged with "bug" label, issues related to performance should be tagged with "performance" label): **2** points -**Refactoring activity** (List the commits and pull requests in which refactoring took place. The commits and pull requests should explicitly mention/discuss/motivate the type(s) of refactoring. The complete list of refactoring types is available at https://refactoring.com/catalog/): **2** points -**Code reviewing** (All features will be contributed in the form of pull requests. Each pull request will have assigned code reviewers and discussions between the reviewers and authors about potential improvements related to code/design quality): **3** points -**Code Quality and Security Report** (Software metrics [size, duplication, documentation, complexity, coupling, cohesion] evolution, Report the technical debt, security vulnerabilities, and code convention violations for the current release. List the commits and pull requests in which technical debt, security vulnerabilities, and code convention violations have been addressed. The commits and pull requests should explicitly mention/discuss/motivate the type(s) of technical debt addressed): **3** points	30%

	-**Presentation to the Instructor** (the content of the presentation will be announced before the release): **3** points	
Release 2 **Nov 8 - Jan 29** Sprint #4 Nov 7 - Nov 27 Sprint #5 Nov 28 - Dec 18 Christmas break Sprint #6 Jan 9 - Jan 29	-**Agile Planning**: **2** points -**Software Architecture** (Discuss about the source code implementation of your design patterns and how you extended these patterns over time as new requirements were implemented): **2** points -**Unit/System Testing Report**: **3** points -**Usability/Performance testing** (Develop code in your app or use tools for collecting user session recordings. If your software does not have UI, but it is a service, you can record service logs using logging libraries, and perform Load and Stress testing. Analyze the collected data to find ways to improve the UI experience or service performance): **5** points -**Issue Tracker**: **3** points -**Refactoring activity**: **4** points -**Code reviewing**: **4** points -**Code Quality and Security Report**: **3** points -**Presentation to the Instructor**: **4** points	30%
Release 3 **Jan 31 - Apr 2** Sprint #7 Jan 30 - Feb 19 Sprint #8 Feb 20 - Mar 12 Sprint #9 Mar 13 - Apr 2	-**Agile Planning**: **2** points -**Unit/System Testing Report** (At this point all user stories should be automated): **4** points -**Usability/Performance testing** (Implement the improvement opportunities you found in the previous release and rerun your usability/performance tests to measure the improvements quantitatively): **5** points -**Issue Tracker** (At this point all open bugs, performance issues should be addressed): **3** points -**Refactoring activity**: **4** points -**Code reviewing**: **4** points -**Code Quality and Security Report** (At this point all technical debt, security vulnerabilities, code convention violations, and code smells should be addressed. If not addressed, you should explain the reasons): **4** points -**Presentation to the Instructor**: **4** points	30%
Acceptance tests	Percentage of the number of user stories signed-off by the product owner through acceptance tests over the total number of user stories in the backlog. **For each user story there should be an acceptance test in the issue tracker of the project. The test is considered as accepted if it is signed-off with the account of the product owner by making an accept comment on the corresponding issue.**	10%

Note that there will be a **peer-evaluation**. Everyone is expected to contribute to the project and we may ask individuals to submit additional materials to verify your contributions. The peer evaluation forms, along with repository contribution statistics will be used to reward team members having an outstanding contribution with bonus marks and penalize team members with very limited contribution with penalty marks.

Course Organization

- You will be using an Agile method, calculating story points, and demonstrating a working tested system early on.
- You will evaluate different libraries for the project. You will also need to demonstrate that you have done significant engineering work beyond these libraries and frameworks.
- You will work in three week iterations. Christmas and other breaks will see longer iterations. At the end of each iteration you will meet with the TA to discuss your progress.
- There will be three releases. Before the release, you must have customer signoff on each story. A story will not be considered complete until you have acceptance tests signed for it.
- As part of each release you will have to address potential competition, legal, social, and other aspects of software development, if applicable.
- After each release, there will be a meeting with the instructor and the TAs to present and discuss your software engineering practices. We will grade each release, and each release is worth the following:
    - Release 1: 30%
    - Release 2: 30%
    - Release 3: 30%
- After your second release, you will try and get real users using your system. You will collect logs and user session recordings, analyze them to find performance bottlenecks and usability issues.
- You will need continuous integration so you can test and push new features live.

Graduate attributes and learning outcomes

As part of both the Computer Science and Software Engineering program curriculum, the content of this course includes material and exercises related to the teaching and evaluation of graduate attributes. Graduate attributes are skills that have been identified by the Canadian Engineering Accreditation Board (CEAB) and the Canadian Information Processing Society (CIPS) as being central to the formation of Engineers, computer scientists and information technology professionals. As such, the accreditation criteria for the Software Engineering and Computer Science programs dictate that graduate attributes are taught and evaluated as part of the courses. This particular course aims at teaching and evaluating the following graduate attributes which are linked to the learning outcomes in the table below.

1. A Knowledge-base for Engineering: Demonstrated competence in university level mathematics, natural sciences, engineering fundamentals, and specialized engineering knowledge appropriate to the program.

2. Problem analysis: An ability to use appropriate knowledge and skills to identify, analyze, and solve complex engineering problems in order to reach substantiated conclusions.

3. Investigation: An ability to conduct investigations of complex problems by methods that include appropriate experiments, analysis and interpretation of data, and synthesis of information in order to reach valid conclusions.

4. Design: An ability to design solutions for complex, open-ended engineering problems and to design systems, components or processes that meet specified needs with appropriate attention to health and safety risks, applicable standards, and economic, environmental, cultural and societal considerations.

5. Use of engineering tools: An ability to create, select, apply, adapt, and extend appropriate techniques, resources, and modern engineering tools to a range of engineering activities, from simple to complex, with an understanding of the associated limitations.

6. Individual and team work: An ability to work effectively as a member and leader in teams, preferably in a multi-disciplinary setting.

7. Communication skills: An ability to communicate complex engineering concepts within the profession and with society at large. Such abilities include reading, writing, speaking and listening, and the ability to comprehend and write effective reports and design documentation, and to give and effectively respond to clear instructions.

8. Professionalism: An understanding of the roles and responsibilities of the professional engineer in society, especially the primary role of protection of the public and the public interest.

9. Impact of engineering on society and the environment: An ability to analyze social and environmental aspects of engineering activities. Such abilities include an understanding of the interactions that engineering has with the economic, social, health, safety, legal and cultural aspects of society.

10. Ethics and equity: An ability to apply professional ethics, accountability and equity.

11. Economics and project management: An ability to appropriately incorporate economics and business practices including project, risk and change management into the practice of engineering and to understand their limitations

12. Life-long learning: An ability to identify and to address their own educational needs in a changing world, sufficiently to maintain their competence and contribute to the advancement of knowledge.

attributes coverage description	level	indicator	evaluation method
Attribute 1: Knowledge-base: This course is about the demonstration of application of most of the skills learned in all courses in the program in a real-life project, ranging from project management issues, to analysis and design modelling, implementation and testing.	●●●	**Indicator 1.3:** Knowledge-base in a specific domain	Demonstrated use of knowledge in project documents and constructed project
Attribute 2: Problem analysis: The project is open-ended and requires an analysis phase that is to be done using techniques/documentation taught in preceding courses dedicated to analysis.	●●●	**Indicator 2.1:** Problem identification and formulation	requirements documentation project artifact
	●●●	**Indicator 2.2:** Modeling	requirements documentation project artifact
	●●●	**Indicator 2.3:** Problem Solving	requirements documentation project artifact
Attribute 3: Investigation: Design elaborated testing using elaborated tools and techniques to prove important system qualities.	●●●	**Indicator 3.1:** Background and hypothesis formulation	software testing project artifact
	●●●	**Indicator 3.2:** Designing experiments	software testing project artifact
	●●●	**Indicator 3.3:** Conducting experiments and collection of data	software testing project artifact

	●●●	**Indicator 3.4:** Analysis and interpretation of data	software testing project artifact
Attribute 4: Design: Investigate on different proposed solutions for the project. The project is open-ended and requires elaborated architectural and detailed design solutions to be documented and implemented.			
	●●●	**Indicator 4.2:** Idea generation and selection	software architecture documentation project artifact
	●●●	**Indicator 4.3:** Architectural and detailed design	software architecture documentation project artifact
	●●●	**Indicator 4.4:** Implementation and validation	implementation
Attribute 5: Use of Engineering tools: The project requires the use of many tools whose use was taught in preceding courses, during all phases of development, and to manage the project. Resource analysis needs to be undertaken as part of the project, and proper standards, tools and libraries be found and used for the implementation.	●●●	**Indicator 5.1:** Ability to use appropriate tools, techniques and resources	utilization of software engineering tools in the project(e.g., code management, code analysis, project tracking)
	●●●	**Indicator 5.2:** Ability to select appropriate tools, techniques, and resources	discussions on selection/evaluation of tools in various aspects of the project

Attribute 6: Individual and team work: The project is undertaken in teams of six to nine members. Each member is assigned specific tasks, even though all team members share the same grades for the project. Every student must prepare an individual report.	●●●	**Indicator 6.1:** Cooperation and work ethics	peer evaluation
	●●●	**Indicator 6.2:** Contribution: practical/conceptual	peer evaluation
	●●●	**Indicator 6.3:** Initiative and leadership	peer evaluation
	●●●	**Indicator 6.4:** Delivering results	sprint planning reports
Attribute 7: Communication skills: Team members will give a presentation of their contribution to the project. The project requires extensive documentations on all aspects of system development.			
	●●●	**Indicator 7.3:** Documentation	Project documentation artifacts
	●●●	**Indicator 7.4:** Oral presentation	Project presentation for each iteration
Attribute 8: Professionalism: The project requires professional interactions with the stakeholders, and within the team.	●●●	**Indicator 8.1:** Role and responsibilities of professional engineers	peer evaluation evaluation from stakeholder
	●●●	**Indicator 8.2:** Professional practice	peer evaluation evaluation from stakeholder
Attribute 9: Impact of engineering on society and the environment: Analyze the	●●●	**Indicator 9.1:** Awareness of society	initial project analysis

possible impacts of the problem or its solution on society, as well as possible sustainability aspects.		and environmental impact	final project report
	●●●	**Indicator 9.2:** Sustainability in design	initial project analysis final project report
Attribute 10: Ethics and equity: Analyze the possible ethical aspects of the problem or its solution.	●●●	**Indicator 10.1:** Professional ethics and accountability	initial project analysis final project report
	●●●	**Indicator 10.2:** Equity	initial project analysis final project report
Attribute 11: Economics and project management: Project cost/effort estimation and tracking, team management.			
	●●●	**Indicator 11.2:** Economic evaluation of projects	project cost/effort estimation
	●●●	**Indicator 11.3:** Project planning and implementation	risk management plan project documentation artifact quality reports customer satisfaction unit and acceptance testing
Attribute 12: Life-long learning: The project involves ample research regarding the problem domain, existing solutions, standards, and libraries	●●●	**Indicator 12.1:** Identifying missing knowledge and learning opportunities	Report on research results at every stage of the project

appropriate for the implementation.	●●●	**Indicator 12.2:** Continuous improvement and self-learning	Report on research results at every stage of the project