



SOEN 387, Fall 2015

Web-Based Enterprise Application Design

M/W, 11:45-13:00, H 407

September 9th till December 7th



Stuart Thiel

stuart.thiel@concordia.ca

<http://www.encs.concordia.ca/~sthiel>

EV 11.411

Office Hours: By Appointment (email at least a day ahead)

(514) 848-2424 x7211

Tutorial: F, 11:45-12:35, H967 (FA) H819 (FB)

Tutor: Korosh Koochekian Sabor, Dharani Kumar

Programmer On Duty: F, 13:45-16:45, H843

POD: Steve Morse

Course Website: <http://www.encs.concordia.ca/~sthiel/soen387/>

Midterm: October 28th

Final: TBD

Course Description: This course is about the architecture and technology used to develop Web-Based Enterprise Applications. Whereas a Software Architecture course will give you a broad idea of architectural patterns, some of which you will see again in this course, SOEN 387 will focus on those patterns directly applicable to web application development.

In this course you will learn about the following technologies:

Hypertext Transfer Protocol (HTTP), Web mark-up languages and encodings. Document Object Models (DOM). Client/server and layered architectures for Web-based Enterprise Applications (WEA). Presentation, Domain and Data Source design patterns. Client-side programming. Java servlets and Java Server Pages. Authentication, security and transaction processing.

Prerequisite(s): COMP 353 - Databases ; COMP 354 - Intro to Software Engineering or SOEN 341 - Software Process; SOEN 287 - Web Programming

Organization: The course is a 3-credit course with 2.5 hours of lectures scheduled every week. There is also one 1-hour tutorials every week. You are expected to allocate a considerable amount of time in studying the text and working on the individual programming project. A qualified tutor will provide you with help should you require it, but is mostly there to give you practical guidance for using the environment. A POD will be available for more direct programming support.

Credits: 3

Text(s):

- Patterns of Enterprise Application Architecture
Author(s): Martin Fowler
- Enterprise Application Design Patterns: Improved and Applied
Author(s): Stuart Thiel

http://soenea.htmlweb.com/soenea/browser/Documentation/trunk/Thesis/Thesis_Stuart.pdf?format=raw

Some topics are from external web references to be provided.

Course Objectives:

The objective of this course is to familiarize students with advanced Enterprise-Application Design Patterns and approaches. This course will focus on how web-based applications are designed and built on both the client and the server, making use of Java Servlets, JSPs, MySQL and XML.

Grade Distribution:

Project	20%
Midterm Exam	40%
Final Exam	40%

Letter Grade Distribution: Grades will be curved around a B, unless the class does abysmally as a whole, then it will be curved around a B-. Two standard deviations below the mean is the threshold to **fail**. Two standard deviations above is a guaranteed A+.

Course Policies:

- **General**

- I hate writing code on exams. I hate marking it more. I cannot guarantee that there will be none, but I will sure try to minimize it.
- Quizzes and exams are closed book, closed notes.
- Exams may draw on lecture notes, book material, assignments, tutorials, class exercises or any reasonable expectation of expertise that could be acquired from looking up how to do any of the exercises/tutorials/assignments.

- **Assignments**

- There is an individual project broken into 2 assignment submissions. The purpose of this project is to develop practical experience and competency with the development of a simple webapp throughout the course. A test suite will be provided for each assignment deliverable, and students are responsible for making sure that their code works with this test suite. Grading is done using an extended version of the test suite, and is done automatically. If you did not make your application work with the provided test suite, you will get zero on that submission.
- Submissions will be through EAS <https://fis.encs.concordia.ca/eas/>.
- Late Submissions will be penalized by 50% for each day late.
- Assignments are due by 11:59pm on the day they are listed as due.

- **Attendance and Absences**

I do not care if you show up. It is your responsibility to know what was covered and what you should know. I will try to keep things light, lively and informative, so please show up ready to engage with your classmates and myself.

- **Evaluation**

The majority of the submission evaluation will be based on running test suites against submitted project. Students will be provided with full or partial test suites to gauge the status of their own submission before submitting and to guide them. The exam evaluation will incorporate some general-knowledge testing, but will focus on evaluating application of design and architectural patterns consistent with addressing enterprise application development problems covered in class.

- **The Midterm**

There is one midterm. The purpose of the exam is to test if you (a) have understood the material of the lecture/book; (b) worked on the assignment project and learned the right lessons from the experience; (c) can provide a complete solution to at least one thought-requiring design/architecture question. Treat the midterm as a wakeup call. There will be no second midterm or a make-up class. The midterm will consist of short-answer questions and multiple-choice/multiple-answer questions.

- **The Final** The final exam covers the whole curriculum. It is likely to be a mix of multiple-choice, short-answer and long-answer questions. The final has the same three evaluation aims as the midterms (see above).

- **Graduate Attributes**

As per the template provided by Dr. Joey Paquet:

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 3 graduate attributes. The following is a description of these attributes, along with a description of how these attributes will be incorporated in the course.

The following attributes are considered and evaluated within the course:

1. Design
2. Use of Engineering Tools
3. Knowledge Base for Engineering, and
4. Problem Analysis

As per the description of Graduate Attributes:

Attribute 1: Knowledge-base: Knowledge of Hypertext Transfer Protocol (HTTP), web mark-up languages and encodings. eXtensible Markup Language (XML). Client/server and layered architectures for Web-based Enterprise Applications (WEA). Presentation, Domain and Data Source design patterns. Client-side programming. Java servlets and Java Server Pages. Authentication, security and transaction processing.

Attribute 2: Problem analysis: Analyze different problems and derive the requirements, design and implementation constraints for the deployment of web-based enterprise applications.

Attribute 4: Design: Use of eXtensible Markup Language (XML), client/server and layered architectures for Web-based Enterprise Applications (WEA), presentation, domain and data source design patterns. Use of class diagrams and sequence diagrams to express system design

Attribute 5: Use of Engineering tools:: Choice and use of appropriate tools for the development of web-based applications, including client- and server-side application development languages and tools.

Attribute 1 will be evaluated directly through the multiple-choice/multiple-answer questions on the exams.

Attribute 2 will be evaluated indirectly through the automated evaluation of the assignments and directly through multiple-choice/multiple-answer questions on the exams.

Attribute 4 will be evaluated directly in terms of implementation and validation quality through the automated evaluation of the assignments. Architecture and design elements of attribute 4 will be evaluated directly on long-answer questions on the exam.

Attribute 5 will be evaluated indirectly through the assignments.

- **Rights and Responsibilities**

What is plagiarism?

The most common offense under the Academic Code of Conduct is plagiarism which the Code defines as "the presentation of the work of another person as one's own or without proper acknowledgement" (Article 16a).

This could be material copied word for word from books, journals, internet sites, professor's course notes, etc. It could be material that is paraphrased but closely resembles the original source. It could be the work of a fellow student, for example, an answer on a quiz, data for a lab report, a paper or assignment completed by another student. It might be a paper purchased through one of the many available sources. Plagiarism does not refer to words alone - it can also refer to copying images, graphs, tables, and ideas. "Presentation" is not limited to written work. It also includes oral presentations, computer assignments and artistic works. If you translate the work of another person into French or English and do not cite the source, this is also plagiarism. If you cite your own work without the correct citation, this too is plagiarism.

In simple words:

DO NOT COPY, PARAPHRASE OR TRANSLATE ANYTHING FROM ANYWHERE WITHOUT SAYING FROM WHERE YOU GOT IT! DON'T FORGET TO USE QUOTATION MARKS!

Source: <http://provost.concordia.ca/academicintegrity/plagiarism>, Sept. 2015

Tentative Course Outline:

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments.

Chapters are:

F: Fowlers Patterns of Enterprise Application Architecture

S: Stuart Thiels Masters Thesis

Week	Content
Week 1	<ul style="list-style-type: none">• Mon. 7th Sep: HOLIDAY• Wed. 9th Sep: Intro to WEA, course scope, web basics refresher
Week 2	<ul style="list-style-type: none">• Mon 14th Sep: Servlets, contexts, servlet interface, http method interface, servlet config• Wed 16th Sep: Servlets, JSP, EL, Taglibs• Fri 18th Sep: Tutorial: Hello World, eclipse and servlets
Week 3	<ul style="list-style-type: none">• Mon 21st Sep: Chapters F:1-2, 14.1 S: 2, Architectural Styles, Layered Architecture, 3-Layer approaches, MVC, good design• Wed 23rd Sep: Chapters F: 4, 9.1, 10.2 S: 3.3, 3.4 Database connections/pooling, Transaction Scripts, Data-Layer Patterns, Interacting with the DB, RDGs• Fri 25th Sep: Tutorial: Transaction Scripts to RDGs
Week 4	<ul style="list-style-type: none">• Mon 28th Sep: Chapters F: 4, 14.4, 14.5 S: 3.5 Presentation Views, Template/Transform Views, JSP, XSLT• Wed 30th Sep: TDD, Outputting XML/JSON, HTML• Fri 1st Oct: Tutorial: Setting up for Test Driven Development
Week 5	<ul style="list-style-type: none">• Mon 5th Oct: Chapters F: 5 Concurrency, Lost Updates, Inconsistent Read• Wed 7th Oct: Chapters F: 12.1, 14.2, 16.1 S: 3.6 Optimistic Offline Lock, Identity, HTTP/Idempotency• Wed 7th Oct: Assignment 1 Posted• Fri 9th Oct: Tutorial: Follow-Up with TDD, now that you have the assignment
Week 6	<ul style="list-style-type: none">• Mon 12th Oct: HOLIDAY• Wed 14th Oct: Chapters F: 9.2 10 S: 3.7 Domain Model Approaches, Active Record, TDGs, Data Mappers• Fri 15th Oct: Tutorial: Hello World, Getting TS + Domain Model + JSP to support Updates
Week 7	<ul style="list-style-type: none">• Mon 19th Oct: Chapters F: 14.2-5, 14.7 S: 3.8 Application Layer, Page Controller, Front Controller, Front Command, Use Cases• Wed 21th Oct: Chapters F: 11.2, 11.3 S: 3.9 Lazy Loading, Identity Map, Midterm Review• Fri 23rd Oct: Tutorial: Hello World, With Buddies, Buddy Ages and of course Lazy Loading and Identity Map

Week	Content
Week 8	<ul style="list-style-type: none"> • Wed 26th Oct: Enterprise Java Beans (or general topic, maybe cloud) • Mon 28th Midterm • Fri 6th Nov: Tutorial: Review last two tutorials
Week 9	<ul style="list-style-type: none"> • Mon 2nd Nov: Chapters F: 11.1, 12.4 S: 3.10 Unit of Work, Dependant Mapping • Wed 4th Nov: Chapters S: 4.1, 4.2, 4.3 Refactoring I, Domain Objects, Lazy Loading and Lists • Fri 6th Nov: Tutorial: Last-minute Assignment 1 support • Fri 6th Nov: Assignment 1 Due
Week 10	<ul style="list-style-type: none"> • Mon 9th Nov: Chapters S: 4.4, 4.5, 4.6 Refactoring II, General Use Identity Maps, UoW Caution, Organizing Data Source Layer • Mon 9th Nov: Assignment 2 Posted • Wed 11th Nov: Chapters S: 4.4, 4.5, 4.6 Referential Integrity, Tying up this refactoring • Fri 13th Nov: Tutorial: What went wrong with assignment 1, looking towards assignment 2
Week 11	<ul style="list-style-type: none"> • Mon 16th Nov: Chapters F: 12.7, 12.8, 12.9, 12.10 Inheritance with Domain Objects • Wed 18th Nov: Chapters S: 5 Leveraging a Framework, SoenEA as an example framework WebSequenceDiagrams: Login Example • Fri 20th Nov: Tutorial: Assignment 2 support
Week 12	<ul style="list-style-type: none"> • Mon 23rd Nov: REST, HTTP, Idempotency, AJAX WebSequenceDiagrams: Challenge Example • Wed 25th Nov: Client-side tools/architecture, Filters/Wrappers, REST/AJAX • Fri 27th Nov: Tutorial: Assignment 2 support, Setting up Permalinks
Week 13	<ul style="list-style-type: none"> • Mon 30rd Nov: Simplifying Sequence Diagrams, Managing Notifications by example • Wed 2nd Dec: Simplifying Class Diagrams • Fri 4th Dec: Tutorial: Assignment 2 support
Week 14	<ul style="list-style-type: none"> • Mon 7th Dec: Review • Tue 8th Dec: Assignment 2 Due