

CS 4523 A - Design Project

(Version 1)
Fall 2018

Instructor's name & title

Professor Strauss, CCP
Department of Computer Science and Engineering
NYU – Tandon School of Engineering

Office

2MT 10.048

Phone

(646) 997-3308

Office hours

Tuesday 12:30 pm – 1:30 pm
Thursday 12:30 pm – 1:30 pm

e-mail

fs817@NYU.edu

Laboratory and Laboratory Monitor

Software Engineering Laboratory (SEL) - RH 223

Course Information

Course level

Senior

credits

3.0

Class hours

Tuesday	8:00 – 9:20 am (Project Team Meetings TBA)	JABS 474
Thursday	8:00 – 9:20 am (Project Team Meetings TBA)	JABS 474

Prerequisites

CS-4513 - Software Engineering

Course Description

This is the second course in a two-course design project sequence (DP I and DP II). This a project coursed in which a student or several students work as a group with a faculty member and/or graduate students on a current topic in computer science. Each term a project course with a particular theme is offered by the Department of Computer and Information Science. A faculty member will assign individual or group projects to students in the class. The project will be highly structured and will be

under close supervision of the faculty. It is expected that students will make use of the design and project management skills learned in CS 4513, software engineering. Alternatively, students can work with a faculty member to develop an individualized project of mutual interest. A Written design report, project demonstration and 30 minute formal (per team member) oral presentation is required. Informal project team presentations will be conducted throughout the semester with the instructor.

Objectives

Projects are evaluated for quality and completeness: project planning and management (documentation and execution), requirements specification, design and architecture specification, conformance with standards and implementation completeness.

Project demonstrations are conducted in the CIS Software Engineering or Parallel and Distributed Processing Laboratories. These laboratories are comprised of state-of-the-art 3 tier technologies (hardware, networking, and application development tools) with required software (middleware, languages, database management systems, capacity and performance tools, and a full complement of automated life cycle development tools). Currently the laboratory is based on fixed base resources (application and database servers, clients, and networking).

Students are encouraged to gain knowledge about how they solve software problems through "best practices", in which they not only perform the work, but are analyzed how it was accomplished. To achieve the goal of the senior project to operate within time, cost and resource constraints the following criteria is used in selecting a project:

- The problem is within the context of a larger development effort. This requires establishment and tracking of project scope and interfacing with the external environment.
- Clients outside the development team are the intended users of the software systems.
- Students have the opportunity to demonstrate individual achievement. Each team member has a unique set of documented roles throughout the project life cycle. These roles are selected and the project organization are developed by the project team and specified in the project plan.
- Project presents a significant state-of-the-art technical challenge.
- Students have the opportunity to apply knowledge and skills gained in previous courses.
- The content of the system is implemented in the laboratory environment.

Core Competencies

- Opportunity to demonstrate that state-of-art software engineering techniques. Student design and implement a computer based software system covering life cycle phases of requirements engineering, architecture, analysis, design, and implementation.
- Written and oral presentation skills are demonstrated using a series of formal documents and presentations. Students will document their systems to software engineering, the software development life (project proposal, project plans, requirements specification and analysis, design description and implementation), project demonstration and formal project presentation (see above). A formal oral project presentation will be conducted at the end of the project.
- Software design techniques include Object-Oriented, function-based and real-time systems.
- Demonstrate software quality and reliability, software life cycle support processes including testing (verification and validation), software reviews, configuration management, and defect detection and correction strategies

General content

This course introduces the software engineering life cycle processes and techniques for the management, development, and documentation of medium and large software systems. Design techniques include Information Engineering, Object-Oriented, and quality/complexity measures. Testing methods such as path testing, exhaustive test models, and construction of test data. Software engineering tools and project management techniques are presented. Student projects involve team software development and tracking.

Grading

Project Proposal	Must be completed
Requirements/Analysis Specification (RAS)	10
Software project Management Plan (SPMP)	10
Software Design Description (SDD) (initial SDD with initial code)	40
Presentation/Demonstration (Capstone competition or VIP May replace the presentation)	35
Attendance/participation is required	5%
Total	100%

Methods of instruction

The primary method of instruction is classroom and online lectures and project team meetings supplemented with related technology and software engineering materials such as IEEE processes and standards, SEI/CMM practices, PSP, best practices, visual modeling tools, configuration management, team dynamics, and quality.

Attendance and Participation

See policy statement

Collaboration on Programming Assignments

See CIS department policy statement

Systems Project

As describe above, an essential requirement of this course is the systems project. Virtually all analysis and design activities are carried out in project teams, or groups in which communication and cooperation are vital to success. The group project is intended to give you experience in performing systems development activities as part of a team.

I will be available for consulting with groups at all stages of the project. **Do NOT fall behind!** The project will be divided into milestones and are specified on the project schedule below.

Project Presentation

Each project team is required to deliver a formal in class presentation describing the technical details of their project focusing on design through implementation life cycle phases and the development process. The presentation delivery, format, and content should be based on material covered in CS 4513 and presentation preparation course (EG at Poly or other course such as public speaking). Presentation worksheets will be distributed to assist teams in the development of their presentations. Each team members is expected to participate in the presentation. Presentation details are:

- Type: Formal presentation
- Participation: All team members must participate (30 minutes/member)
- Audience: Instructor (acting as project manager), class, faculty and administration

- Dress: Business (formal or casual)
- Format/Media: PowerPoint or other delivery mechanism (i.e. Web based)
- Content:
 - Overview, team member introductions
 - Project process/management
 - Requirements review (Use Cases, written requirements)
 - Design (detailed design using UML, final component architecture)
 - Implementation (implementation and deployment architecture, coding, design patterns)
 - Future work - Open Issues/Extensions
 - PIR (what did we do right? What did we wrong? What would we do differently?)
 - Conclusions
- Open question/Answers – Instructor and class
- Submission: Presentations are to be posted to the team assignment menu on NYU Classes and all project documentation on CD media.

Project Schedule/Milestones

<u>Milestone</u>	<u>Due date</u>
Project Proposal	September 11
Requirements and Analysis Documentation (RAS)	September 20
Project Management Plan (SPMP)	September 27
Design Description (SDD) - Initial	October 18
Design Document Final (w/Code)	November 29
Implementation/Demonstration	Starts December 6
Formal Oral Presentations	Last two weeks Of the semester

Course Calendar and Schedule

Week/Lesson	Date	Topic	Assessment (due date)
1/1	9/4	Project Work	
1/2	9/6	Project Work	
2/1	9/11	Project Work	
2/2	9/13	Project Work	Present Business Overview (5 minutes/team)
3/1	9/18	Business Model Canvas	
3/2	9/20	Business Model Canvas	
4/1	9/25	Customer Discovery	
4/2	9/27	Talking to Customers	Present Business Model Canvas #1 (10 minutes/team)
5/1	10/2	Value Propositions	
5/2	10/4	Customer Segments	Present Business Model Canvas #2 (10 minutes/team)
6/1	10/9	Legislative Day	
6/2	10/11	Channels - Revenue	
7/1	10/16	Team work	Present Business Model Canvas #3 (10 minutes/team)
7/2	10/18	Partners, Key Resources	
8/1	10/23	Activities	Present Business Model Canvas #4 (10 minutes/team)
8/2	10/25	Wrap Up	
9/1	10/30	Agile	
9/2	11/1	Agile	
10/1	11/6	Agile	
10/2	11/8	Agile	
11/1	11/13	Agile	
11/2	11/15	SCRUM	
12/1	11/20	SCRUM	
12/2	11/22	Thanksgiving Recess	
13/1	11/27	SaaS	
13/2	11/29	SaaS	
14/1	12/4	SaaS	
14/2	12/6	Project Presentations	
15/1	12/11	Project Presentations	
15/2	12/13	Project Presentations	
	12/17 – 5/21	Final Examination	

Team Status Reports (*)

- Team Number and Project Name
- Project Description
- Team Status
- What did we accomplish since the last status report?
- Any obstacles?
- What are we going to achieve before the next report?
- Issues or Concerns

(*) Using the concepts of: continuous uncertainty , change, experimentation, learning, and implementation/delivery

Initial Overview Presentations

5 slides

1. Cover
2. Current business model
3. Market size estimate
4. Type of business
5. Proposed experiments to test customer segment
(two-way communication)

Lean Retrospective Presentations

3 slides

1. Cover slide
2. Current business model w/changes
3. What did you learn from your customer:
 - a. Hypothesis – what we thought
 - b. Experiments – what we did
 - c. Results – what we found
 - d. Iterate (next Step)