

# CSEP 545

Transaction Processing for E-Commerce  
Winter (January – March) 2012

Course Information

# Instructors

- Philip A. Bernstein
  - philbe@microsoft.com
  - (425)706-2838
  - <http://www.research.microsoft.com/~philbe/>
- Sameh Elnikety
  - samehe@microsoft.com
  - (425)538-2234
  - <http://www.research.microsoft.com/~samehe/>

# Meeting Time

- Wednesdays 6:30 – 9:20 PM
  - From January 4, 2012 to March 7, 2012
- A one-hour project demo
  - During weeks of March 5 and 12
  - (exact dates TBD)
- Exams
  - Mid-term exam, probably Feb 22, 2012
  - Possibly a second exam

# Textbook

“Principles of Transaction Processing”

2<sup>nd</sup> Edition

Philip A. Bernstein and Eric Newcomer

Morgan Kaufmann Publishers, 2009

# Teaching Assistants

- YongChul Kwon
  - [yongchul@cs.washington.edu](mailto:yongchul@cs.washington.edu)
  - <http://www.cs.washington.edu/homes/yongchul/>
- Emad Soroush
  - [soroush@cs.washington.edu](mailto:soroush@cs.washington.edu)
  - <http://www.cs.washington.edu/homes/soroush/>
- They're advanced PhD students in DB systems
- Their main job is to support the course project

# Web Site and Mailing Lists

Web site:

<http://www.cs.washington.edu/education/courses/csep545/12wi>

Mailing list:

<http://mailman.cs.washington.edu/mailman/listinfo/csep545>

**Please subscribe immediately to this mailing list**

# Lecture Outline

- Introduction (Chapter 1)
- Shadow-based recovery, in support of project (Chap 7, Sec 6).
- Database Concurrency Control, Part 1 (Chap 6, Sections 1-4)
- Database Recovery (Chapter 7)
- Basic Application Servers, in support of the project.
- Two-Phase Commit (Chapter 8)
- Database Concurrency Control, Part 2 (Chap 6, Sections 4-11)
- Queuing (Chapter 4)
- Replication (Chapter 10)
- Business Process Management (Chapter 5)
- Application Servers (Chapters 2 and 3)

# Lecture Outline - Remark

- Exact order of topics may vary slightly
- Mapping of topics to course meetings
  - Hard to predict
  - Topics are of different lengths
  - Some class time will be consumed by
    - Discussions of assignments
    - Discussions of project



# Prerequisites

- There are no specific prerequisites
  - Besides a good general understanding of software systems
- Helpful to have some knowledge of SQL
  - To be able to write simple queries
- Project
  - Good working knowledge of C# or Java

# Grading

- Your grade has two components
  - Exams
    - There will be a mid-term exam
    - Possibly a second exam
  - Project
    - You will do a substantial two-person project
- Grade is a weighted sum
  - One exam
    - Exam (20%) and project (80%)
  - Two exams
    - Exams (30%) and project (70%)

# Assignments

- Weekly assignments for much of the course
- Cover material that's easiest to learn by solving structured problems
- Solution discussed in lecture the following week
- The assignments will NOT be graded
- Make a serious attempt at solving them
  - Effective way to learn the material
  - Some exam questions will be minor variations

# Project

## Objective:

Transaction processing is a systems engineering problem, with many interacting parts. The goal of the project is to deepen your understanding of how the parts fit together. It involves implementing a distributed transaction system.

## Distributed system:

Distributed server will allow customers to make/cancel/query travel reservations, will allow airlines to add/cancel/modify flight information, and will allow hotels and car rental agencies to post room and car availability.

# Project

## We provide:

Clients, the interface the clients expect, a lock manager, the basic application, and a general design framework for the (ultimately distributed) server.

# Project

## You provide:

You will build mechanisms for persistence, recovery, and two-phase commit. The transaction system will concurrently support multiple clients and multiple servers.

*Make sure you have a solid working solution of basic features, and save a snapshot of that code, before adding more advanced features.*

Only a short write-up is required to summarize what you have done and to provide a blueprint of the code, as long as the code is readable. A final demonstration is required.

# Project

## Groups:

Except in rare cases, people will work in groups of two.

- It helps you, to have design discussions with a partner.
- And it helps us, to make our reviewing load manageable.

## Implementation:

Either C# or Java.

# Project Dates

- Jan 12 – Groups
  - Email message telling us who you will be working with and which language (Java or C#).
- Jan 25 – Milestone One
  - You should have at least the first couple of steps implemented.
- Feb 15 – Milestone Two
  - You should have at least half of the final project features running. Hand that in, with enough of a write-up that Emad and YongChul can understand your overall design and how it maps to your code.
- March 7 – Final Project is due.
  - We review as many of the demos as possible this week.



# Project Milestones

We have two Milestones.

Milestone reports are largely for your benefit, to ensure you're pacing the work and to check that you're on the right track.

Milestones will be reviewed but not graded. If you do not hand in a milestone, and we later find that you're making some serious design errors, it will be *your* problem for not having given us the chance to flag the error early enough for you to fix it.