

# BLG361E – DATABASE MANAGEMENT SYSTEMS

## 2013-2014 FALL

	<b>CRN: 10795</b>	<b>CRN: 10797</b>
<b>Instructors:</b>	Assoc. Prof. Dr. Şule Öğüdücü	Lec. H. Turgut Uyar
<b>Office Hours:</b>	<i>send e-mail for appointment</i>	Tuesday 13:30-14:30
<b>Assistants:</b>	Nagehan İlhan – Mahiye Uluyağmur	

**Textbook:** Chris J. Date, “An Introduction to Database Systems”, Addison-Wesley, ISBN 0-321-19784-4, 2004.

### Weekly Plan

Week	Date	Lecture	Exercise
1	18 Sep. 2013	Introduction	Java
2	25 Sep. 2013	Relational Model	Java
3	2 Oct. 2013	Relational Model - SQL	Wicket - Introduction
4	9 Oct. 2013	Relational Algebra	Wicket - Structure
	16 Oct. 2013	HOLIDAY	
5	23 Oct. 2013	Relational Algebra - SQL	Wicket - Forms
6	30 Oct. 2013	Application Development	Wicket - JDBC
7	6 Nov. 2013	Database Design	E/R Diagrams
8	13 Nov. 2013	Recitation	
9	20 Nov. 2013	Midterm Exam	
10	27 Nov. 2013	Concurrency	Concurrency
11	4 Dec. 2013	NoSQL Databases	
12	11 Dec. 2013	Optimization	
13	18 Dec. 2013	Project Presentations and Demos	
14	25 Dec. 2013	Project Presentations and Demos	

### Grading

- Midterm exam: 25%, Project: 30%, Final exam: 45%.
- Conditions for participation at the final exam:
  - 70% attendance (8/12 weeks, no check on weeks 8 and 9)
  - project grade at least 30/100
  - weighted average of midterm exam and project grades at least 30/100
- Students with overall average under 40/100 will fail.

### Important Notes

- You can follow the course announcements, your exam results and attendance status on the Ninova system (<http://ninova.itu.edu.tr/>). Check the Ninova site regularly for updates.
- Course related e-mail notifications will be sent to your ITU account, check it regularly.
- When sending e-mail to the instructors or assistants, use your ITU account and always include your full name at the end of the message.

## PROJECT

In the project, groups of 4-5 students will develop a web application using the Java language, the Apache Wicket platform, and an SQL database. Any SQL product is allowed as long as the project uses JDBC. Other technologies like JPA and object-relational mappers are not allowed.

Every group member will be responsible for implementing the database and interface operations of at least three objects. At the end of the term, all students will submit the following items:

- project code (the same files for all group members)
- individual report (written in DocBook)
- group report (combined individual reports, the same files for all group members)
- presentation document (as PowerPoint or PDF, the same files for all group members)

The Mercurial version control system will be used for monitoring the progress of group members. The projects will be hosted on the pikacode.com site.

## Schedule

*Note:* The deadline for each work item is the end of the working hours before the class day, i.e. Tuesday 17:30. So, for example "by the 2nd week" means "by 24 Sep. 2013, 17:30". Holiday weeks do not count.

Each group proposes a project which will be checked and approved by the instructors. The proposal forms are submitted to the Ninova system by the 2nd week. The proposals can include 1 to 5 group members; students from different sections can be in the same group. Every student must submit the proposal document, even if the forms for all members will be the same. Every group will also hand in a printed copy of the proposal, signed by all group members. Due to resource limitations, the instructors might decide to reorganize the groups.

Project groups and topics are confirmed and announced on the 3rd week by the instructors.

Each group member completes the add-delete-update-search operations of at least one object by the 8th week.

All groups submit their project codes and presentations at the beginning (Monday 10:00) of the 13th week to the Ninova system.

All groups give a hands-on demo to the responsible TA according to the announced schedule on the 13th and 14th weeks. Some (if possible, all) of the groups present their projects to the class. The groups that are not selected also have to attend these presentations.

All students upload the source files (DocBook) and the printable version (PDF) of their individual reports on the 14th week to the Ninova system. The report consists of an installation guide, a user manual, and a technical manual. Every student gives detailed information about the objects that she/he is responsible for, in addition to the common information about the project.

All groups submit the source files and the printable version of the group report to the Ninova system by the day of the final exam. A printed group report will also be handed in at the designated report box at the department secretarial office.

## Grading

Each student will get grades for coding, report, group presentation and progress. Grades will be weighted as 45% for coding, 30% for report, 15% for presentation, and 20% for progress.

Each student's coding grade will be determined by her/his project work and by her/his performance at the demo session. The demo session has a strict "no show" policy and absent group members will get 0 for the coding grade.

The groups that are selected for the presentation will be graded according to their presentation performance in addition to the presentation document. The groups that do not present their project, or the students who do not attend the presentation session will get 0 for their individual presentation grade.

The progress grade will be given according to the weekly commits to the version control repository. The progress grades will be announced on the Ninova system as part of the attendance sheet, one for each Tuesday starting on week 5, and they will indicate the amount of work done during the previous Wednesday and that Tuesday. Weekly progress marks are announced as 0, 1, or 2 where 0 indicates no work or a negligible amount of work, 1 indicates an expected amount of work, and 2 indicates extra work. There might also be "-" or "+" marks to indicate "borrow" or "carry" into later weeks. For example, "1+" means "more than 1, not as much as 2, but less work is needed for a 1 next week". Note that, it's not just the amount of committed code that gets graded, other factors play into the evaluation:

- Commits that do not provide enough information in the commit message are ignored. For example, commits with no commit message, or commits with unknown username. Make sure that your username is set in Mercurial and check if it shows up correctly in the changelog on Pikacode.
- Large changesets that bundle irrelevant changes will get at most a 1, even if they contain a lot of extra work.
- Commits that obviously break the code get a 0. For example, commits that contain conflicts. You can spot conflicts by the lines that start with ">>>".
- Trivial code such as basic getters and setters do not contribute to the grade.
- Poor coding style such as inconsistent indentation, bad variable/method/class names, etc. will result in lesser grades.