

CS 457/CS 557 – Database Software Design

Project's suggestions

The aim of the final project is to realize an application where the concepts seen in the course Database Software Design are used. This application can be a database management System, mobile application, or Web in Java or PHP. This project can be done by up to four students. Each group of students must provide a final report (up to eight pages) about the project they developed. However, the following instructions must be taken into consideration:

1. The final report must specify the contribution of each member of the group. For example, who was charged for the design, programming, and how you were efficient for the development of this application. Namely, the role of each member of the group must be mentioned.
2. In addition, the report must contain the cost in time for each stage to development this project, for example for design, programming, and maintenance if it is available.
3. The used modeling method must be justified and to explain why this method is well suited for this project.
4. Please mention the kind of problems that you faced this during the realization of this project, and how you solved them. The final report must be presented in a form a research paper. The project and the final report must be submitted no later than the 9th of April, 2021.

Each of the following is a large project, which can be a semester-long project done by a group of up to four students.

Project 1

Pick your favorite interactive Web site, such as Bebo, Blogger, Facebook, Flickr, Last.FM, Twitter, Wikipedia; these are just a few examples, there are many more. Most of these sites manage a large amount of data, and use databases to store and process the data. Implement a subset of the functionality of the Web site you picked. Clearly, implementing even a

significant subset of the features of such a site is well beyond a course project, but it is possible to find a set of features that is interesting to implement, yet small enough for a course project.

Most of today's popular Web sites make extensive use of Javascript to create rich interfaces. You may wish to go easy on this for your project, at least initially, since it takes time to build such interfaces, and then add more features to your interfaces, as time permits. Make use of Web application development frameworks, or Javascript libraries available on the Web, such as the Yahoo User Interface library, to speed up your development.

Project 2

Create a “mashup” which uses Web services such as Google or Yahoo maps APIs to create an interactive Web site. For example, the map APIs provide a way to display a map on the Web page, with other information overlaid on the maps. You could implement a restaurant recommendation system, with users contributing information about restaurants such as location, cuisine, price range, and ratings. Results of user searches could be displayed on the map. You could allow Wikipedia-like features, such as allowing users to add information and edit information added by other users, along with moderators who can weed out malicious updates. You could also implement social features, such as giving more importance to ratings provided by your friends.

Project 3

Your university probably uses a course-management system such as Moodle, Blackboard, or WebCT. Implement a subset of the functionality of such a course-management system. For example, you can provide assignment submission and grading functionality, including mechanisms for students and teachers/teaching-assistants to discuss grading of a particular assignment. You could also provide polls and other mechanisms for getting feedback.

Project 4

Design and implement a shopping cart system that lets shoppers collect items into a shopping cart (you can decide what information is to be supplied for each item) and

purchased together. You should check for availability of the item and deal with non available items as you feel appropriate.

Project 5

Design and implement a Web-based system to record student registration and grade information for courses at a university.

Project 6

Design and implement a system that permits recording of course performance information specifically, the marks given to each student in each assignment or exam of a course, and computation of a (weighted) sum of marks to get the total course marks. The number of assignments/exams should not be predefined; that is, more assignments/exams can be added at any time. The system should also support grading, permitting cut-offs to be specified for various grades.

You may also wish to integrate it with the student registration system of Project 5 (perhaps being implemented by another project team).

Project 7

Design and implement a Web-based system for booking classrooms at your university. Periodic booking (fixed days/times each week for a whole semester) must be supported. Cancellation of specific lectures in a periodic booking should also be supported.

You may also wish to integrate it with the student registration system of Project 6 (perhaps being implemented by another project team) so that classrooms can be booked for courses, and cancellations of a lecture or addition of extra lectures can be noted at a single interface, and will be reflected in the classroom booking and communicated to students via email.

Project 8

Design and implement a system for managing online multiple-choice tests. You should support distributed contribution of questions (by teaching assistants, for example), editing of questions by whoever is in charge of the course, and creation of tests from the available set of questions. You should also be able to administer tests online, either at a fixed time

for all students, or at any time but with a time limit from start to finish (support one or both), and give students feedback on their scores at the end of the allotted time.

Project 9

Design and implement a system for managing email customer service. Incoming mail goes to a common pool. There is a set of customer service agents who reply to email. If the email is part of an ongoing series of replies (tracked using the in-reply-to field of email) the mail should preferably be replied to by the same agent who replied earlier. The system should track all incoming mail and replies, so an agent can see the history of questions from a customer before replying to an email.

Project 10

Design and implement a simple electronic marketplace where items can be listed for sale or for purchase under various categories (which should form a hierarchy). You may also wish to support alerting services, whereby a user can register interest in items in a particular category, perhaps with other constraints as well, without publicly advertising her interest, and is notified when such an item is listed for sale.

Project 11

Design and implement a Web-based newsgroup system. Users should be able to subscribe to newsgroups, and browse articles in newsgroups. The system tracks which articles were read by a user, so they are not displayed again. Also provide search for old articles. You may also wish to provide a rating service for articles, so that articles with high rating are highlighted, permitting the busy reader to skip low-rated articles.

Project 12

Design and implement a Web-based system for managing a sports “ladder.” Many people register, and may be given some initial rankings (perhaps based on past performance). Anyone can challenge anyone else to a match, and the rankings are adjusted according to the result. One simple system for adjusting rankings just moves the winner ahead of the

loser in the rank order, in case the winner was behind earlier. You can try to invent more complicated rank-adjustment systems.

Project 13

Design and implement a publication-listing service. The service should permit entering of information about publications, such as title, authors, year, where the publication appeared, and pages. Authors should be a separate entity with attributes such as name, institution, department, email, address, and home page. Your application should support multiple views on the same data. For instance, you should provide all publications by a given author (sorted by year, for example), or all publications by authors from a given institution or department. You should also support search by keywords, on the overall database as well as within each of the views.

Project 14

A common task in any organization is to collect structured information from a group of people. For example, a manager may need to ask employees to enter their vacation plans, a professor may wish to collect feedback on a particular topic from students, or a student organizing an event may wish to allow other students to register for the event, or someone may wish to conduct an online vote on some topic. Create a system that will allow users to easily create information collection events. When creating an event, the event creator must define who is eligible to participate; to do so, your system must maintain user information, and allow predicates defining a subset of users. The event creator should be able to specify a set of inputs (with types, default values, and validation checks) that the users will have to provide. The event should have an associated deadline, and the system should have the ability to send reminders to users who have not yet submitted their information. The event creator may be given the option of automatic enforcement of the deadline based on a specified date/time, or choosing to login and declare the deadline is over. Statistics about the submissions should be generated, to do so, the event creator may be allowed to create simple summaries on the entered information. The event creator may choose to make some of the summaries public, viewable by all users, either continually (e.g., how many people have responded) or after the deadline (e.g., what was the average feedback score).

Project 15

Create a library of functions to simplify creation of Web interfaces. You must implement at least the following functions: a function to display a JDBC result set (with tabular formatting), functions to create different types of text and numeric inputs (with validation criteria such as input type and optional range, enforced at the client by appropriate JavaScript code), functions to input date and time values (with default values), and functions to create menu items based on a result set. For extra credit, allow the user to set style parameters such as colors and fonts, and provide pagination support in the tables (hidden form parameters can be used to specify which page is to be displayed). Build a sample database application to illustrate the use of these functions.

Project 16

Design and implement a Web-based multiuser calendar system. The system must track appointments for each person, including multi-occurrence events, such as weekly meetings, and shared events (where an update made by the event creator gets reflected to all those who share the event). Provide interfaces to schedule multiuser events, where an event creator can add a number of users who are invited to the event. Provide email notification of events. For extra credits implement a Web service that can be used by a reminder program running on the client machine.