

THE SOCIAL BOOKMARKING SYSTEM – PROJECT BRIEF

Learning Aims

In this integrative project you will apply principles of database design and information retrieval to create a small but relatively complex information system. You will learn to use professional development tools and technologies (e.g., NetBeans 5.0, JSP, PostgreSQL, JDBC, Lucene search API). Most importantly, you will learn how to design and implement an information system that satisfies a relatively vague, open-end problem statement. Through this process you will gain confidence for tackling complex technical problems, develop your skills for design and implementation, and come to appreciate the beauty that underlies information systems.

Objective

The objective of this project is to develop a working prototype calling SBS for Social Bookmarking System. To guide this 8 week process, you will prepare a set of deliverables.

Root concept

SBS is an information system that enables people to collect items. People can describe items and tag them with keywords. Then, using the keywords people can browse and search their collections. In addition, your search should allow people to issue 'best-match' searches to retrieve a ranked list of the best matching items in your system.

Various kinds of social bookmarking systems are in wide use. One example is: <http://del.icio.us>

Short scenarios

Consider these short scenarios that reify the root concept:

- A high school teacher wants to collect a set of websites on environmental education, especially in the Pacific Northwest.
- A professor wants her class to develop a collection of academic papers on the topic of "Why Information System Development Project often fail?"

Before reading further, please identify a similar scenario that might benefit from this approach. What features of your scenario make it like these examples?

User Tasks in SBS

Consider the following user tasks that people might perform with SBS. In these tasks, SMALL-CAPS-TEXT represents an entity/noun and underlined-text represents an operation/verb.

1. A USER can create a PROJECT.
2. A USER can submit ITEMS to his or her PROJECT.
3. A USER can assign zero, one, or more KEYWORDS to an ITEM.
4. A USER can organize the KEYWORDS into a HIERARCHY.
5. A USER can browse and search ITEMS by KEYWORD through a HIEARARCHY.
6. A USER can search ITEMS by a free-text, best-match query (i.e., by words within items).
7. A USER can assign editing privileges to a group of USERS.
8. A USER'S BEHAVIOR is tracked and recorded for future analysis.

Before reading further, select a scenario from above or one of your own choosing and consider which tasks you think are most relevant. Does your scenario raise new tasks? Do some tasks for your scenario need to be modified? Do simpler tasks underlie the above tasks?

Please note: You will not need to implement solutions for all these tasks!

General Requirements

Your solution should satisfy the following requirements:

1. Develop an operational definition for the concept of keyword so that a solution can be implemented. You should be able to do useful things with keywords and it should be possible to represent a hierarchy of keywords of arbitrary depth. For example:
 keyword-A (sports)
 keyword-1 (skiing)
 keyword-a (downhill)
 keyword-b (x-skiing)
 keyword-c (telemark)
 keyword-2 (cycling)
 ...
 ...
 ...
 keyword-B (eating)
 ...
2. Develop an abstraction that allows you to represent various kinds of items (e.g., webpages, music, movies, journal articles, and books). Assume that all item types have a title and a http link to the source.
3. Choose ONE item for your implementation but develop a general design so that MANY different kinds of items can potentially be represented in your system.
4. Develop an abstraction that allows you to track what users do to your system (e.g., track the keywords added, items added, items visited, etc.).
5. Develop an abstraction for users that includes a profile, including such information as age, gender, interests, favorite places, and so on.
6. Assume that image, audio and other such files are stored in an external storage infrastructure and that these files have unique identifiers that can be accessed by hypertext links. Thus, a file upload function and general storage infrastructure is not required.
7. Do not implement a secure user authentication scheme. You may encode unencrypted user identifiers on HTTP requests.
8. A high quality visual design is not required. The emphasis of this project should be on interesting and powerful functions. The user-interface needs only to reveal these functions and be neat and clear.

As you proceed through the project, you should repeatedly return to this list of requirements and explore how they can help you reduce the complexity of your project.

Team Organization

Teams will consist of three people and will be selected by the instructor and teaching assistant based on a skills profile. If necessary, you will have an opportunity to change teams.

As outlined below some deliverables are the joint responsibility of the team. Other deliverables are the responsibility of individual team members. For the individual deliverables, you should collaborate closely

and help each other, but the design and coding must be done individually by a designated lead developer. This person must be clearly identified in the submitted deliverables.

The objective of this project is to develop a detailed specification and a working prototype. To do this, you will be asked to prepare a set of deliverables (see below).

DELIVERABLES

As it stands, the Social Bookmarking Systems is a large ill-defined project. It is up to you to create a design and to implement a solution.

To help you manage its complexity a set of deliverables and a timeline is proposed below.

Preliminaries

You will submit three reports:

- Report #1: Working vision report (week #4)
- Report #2: Draft report (week #8)
- Report #3: Final report (week #10)

The audience for the final report is a technical project manager; that is, someone with a background in information system design who plans and manages projects. This person will appreciate concise, clear, interesting writing and seeks to understand **what you did** and **why you did it**. In addition, this person will appreciate innovative ideas, critical discussion about their merits, and realistic plans for evaluating them.

You should create a simple website to post your project reports and any other necessary materials. Your website, of course, should link to the Social Bookingmarking System prototype.

The front door of your website should include the team name, team members, the course name, course date, and so on. You should invent an appropriate name for your prototype.

For a degree of privacy, you may select an obscure URL for your website. You will give this URL to the teaching assistant and instructor so they can grade your work.

Report #1: Working vision report

The aim of report #1 is to start thinking about the project! The first report should consist of the following sections:

1. **Title Page**
Name of project, list of team members, etc.
2. **Scenario** (< 1 page)
A brief description of who the users of SBS are and what tasks they will complete (< 1 page)
3. **User Interface Sketch**
A sketch the major screens and interaction flow for your system. You should sketch the user interface with pencil on **blank** paper sheets OR sketch a design on a whiteboard and take images of your design OR sketch a design on Table PC OR sketch a design with a simple drawing package. The goal is quickly (and easily) specify your design clearly but not in great detail. High fidelity drawings are not desirable at this stage – clear thinking is.

4. **Conceptual model** (< 3 pages)

A list of entities, attributes and relationships that your system will need to represent in order to allow you to implement the user interface. You are welcome to use an Entity-Relationship Diagram for this but it is not necessary. Please note: This is a draft and you are free to change at anytime during the quarter.

5. **Questions**

A list of questions that you have about this project – we will discuss these questions in class.

Structure of Report (Report #2 is a revision of Report #1)

Title Page (~ 1 page)

Project name • Team members • System URL • etc.

Executive Summary (~ 1 page)

Concise summary of the project, indicating the vision, key technical components, and current status

Table of Contents (~ 1 page)

Present a table of contents for the final report

1. **Introduction** (~2 pages)

Please present a brief introduction to the Social Bookmarking System. The introduction should orient the reader to the report and describe what the SBS is, why you are working on it, and so on.

1.1 Vision

The vision should briefly describe who the users are and what they will do with the system. You can select one of the above scenarios and elaborate it with your own ideas or you can propose a new scenario that is consistent with the root concept of social bookmarking.

1.2 Functional requirements

Include a list of functional requirements for your system. Functional requirements are terse statements of what the system will do but not how.

They take the following form: *The system will have the ability to ...*

2. **Conceptual Model** (1 page for a diagram PLUS 1 page for description)

Please present the conceptual model. The conceptual model describes the entities, attributes, and relationships of the system independent of the target implementation platform.

You must use **Enhanced Entity-Relationship** modeling and UML notation. In addition to the diagram, this section must include a brief description of the key features of the model.

3. **Logical Model** (1 page for a diagram PLUS 1 page for description)

Please present the logical model, where the conceptual model (section 2) is transformed into a set of entities, attributes, and relationships.

In addition, your model (including description) must specify the following:

- a) The primary and foreign keys;
- b) The integrity constraints for the model;
- c) The parts of the overall model that you will implement.

You must use UML for your logical model. Your logical model should consist of entities, and 1-1 and 1-M relationships only. It must not contain M-M relationships, complex relationships, and so on.

In appendix A, you must include the SQL script used to define your database. The script should be neat and well commented.

4. Information Retrieval Model (~ 2 pages; final report only)

Please present the IR model. This model consists of a description of the best-match searching that your system will offer. Referring back to the conceptual model, you should include a description of how you represent documents, what metadata they contain, the indexing policies used, and how data from the Information Retrieval module is kept consistent with the database module.

5. System Architecture (1 page for a diagram PLUS 1 page for description)

Please clearly describe the system architecture used in your implementation. It should describe what these technologies are and how they fit together: JSP pages, HTTP requests and responses, Java application classes, PostgreSQL, Lucene, Tomcat web server, JDBC, web browsers, WAR files, and so on. You must use a three-tier model to present your system architecture.

6. Implementation notes (be concise and, please, no busy work)

This section should include a description of your implementation. You should provide an overview of how you've implemented the various functions of your system. You should include screen shots of your system, describe strengths and weaknesses of your implementation, etc. After reading this section the reader should have a good understanding of the merits of your implementation. It is suggested that you organize your implementation into the following sections (although you may include different functional components):

6.1 Functions for phase #1

- 6.1.A Create user (<developer name>)
- 6.1.B Create keyword (<developer name>)
- 6.1.C Create item (<developer name>)

6.2 Functions for phase #2 (final report only)

- 6.2.A Browse by keyword (<developer name>)
- 6.2.B Search by keyword (<developer name>)
- 6.2.C Create project (<developer name>)

6.3 Functions #3 (final report only)

- 6.3.A Best-match search (<all>)

7. Status and Next Steps (~1 page; final report only)

Summarize the state of the SBS by answering questions such questions as: 1) How effective is the design; 2) How much of the implementation is complete; 3) What needs to be revised; and 4) What would you do next if you were to continue with the project?

8. SQL Build Script

In Appendix A, please include the SQL script that is used to build the initial version of the database.

9. Team Reflection (~1 page; final report only)

In Appendix B, please include a brief statement of what you learned, how you helped each other, what worked, what you would do differently next time, and so on. You may write this a single group statement or three individual statements.

10. Design tools and Process (~1 page)

In Appendix C, please describe how your team is working together. You should indicate: 1) How often you meet and where; 2) What communication tools you used; 3) What tools you used to coordinate and plan your work; 4) What documents you use to make progress; and 5) The challenges you experience when working together and how you overcome them.

GRADING GUIDELINES

General mechanics

- Please try to keep with the suggested page limits – conciseness and clarity are valued. If you need to go beyond the page limits, do so but have a good reason;
- If necessary, include other appendices (e.g., references, sketches of the user interface, etc.);
- All figures and tables must have captions;
- On the title page, please include an URL to the website;
- The code for the system should be readily available at the website;
- All sketches can be done with pencil and blank paper or whiteboard and then scanned into the document but they must be done neatly;
- If you find yourself doing busy work STOP; then, please talk to the teaching assistant or instructor. Everything in the three reports should help you make progress on your project. And, of course, if you have a better idea for the project or report format, please speak up.
- If you want to go beyond the general requirements do so but please weigh the benefits and manage the risks (and talk with the teaching assistant or instructor).

Teamwork

- All members of the team will receive the same grades for the first three components of the report;
- The fourth component of the report may be graded individually;
- Teams are assigned by the instructor and teaching assistant – you will have an opportunity to suggest revisions to the team assignment (if necessary);
- If you experience difficulties working as a team, talk with the instructor or teaching assistant as soon as possible – we can help.

Values

Note: Please use Report #1 and Report #2 as an opportunity to get feedback. For the sections where you are uncertain, **do your best and ask questions** in writing. We will answer them! The key objective of the first two reports is to receive useful feedback for the final report.

General grading criteria:

- The design is elegant and simple;
- Decisions are made explicit and justified;
- UML notation is fully and rigorously used for the EER and ER diagrams;
- The writing is clear, concise, and interesting;
- All diagrams and sketches are neatly presented, with captions.

1. Overall 10%

- All report sections present;
- Absence of spelling and grammatical errors;
- Captions and legends explain diagrams and charts.

2. Front and Back 10%

- The project is introduced and concluded in an effective manner.

3. Information System Specification 20%

- Data models show entities, relationships, attributes, types, and so on;
- The overall architecture is presented in an effective manner.

4. Implementation 60%

1. The modules that make up your implementation are clearly outlined;
2. The modules are described and their merits – strengths and weaknesses – are discussed;
3. The user interface is included in the report and annotated.