

CPSC 431, Database and Application

Database design and application development techniques for a real-world system. System analysis, requirement specifications, conceptual modeling, logic design, physical design and web interface development. Develop projects using contemporary database management system and web-based application development platform.

Final Project

1 Overview

You are a PHP and MySQL database application developer tasked with defining, implementing, deploying, and delivering a 3-tier software solution¹ that keeps track information related to running a sports team. The general concept is for an operator to use a web browser on any network-connected device to connect to a hosting server where authorized information can be viewed and modified based on the operator's role. Requirements provided here are intentionally vague allowing and encouraging you to define your system with some degree of freedom. You will form a team of engineers to complete your task.

Your Final Project has two parts, an Oral Presentation and a Product Delivery. Oral Presentations are conducted the last two weeks of class, and the Product Delivery is due during the scheduled Final Exam. It is understood the Oral Presentation will be conducted before your project has been completed, and your solution may not be fully functional during your presentation. Instead, the Oral Presentation can be thought of as a critical design review where you present your artifacts, reveal your progress, and highlight areas your team has more work to do. Take the opportunity to validate your approach and solicit mid-course corrections from your classmates. Those that present early will not have as much time to prepare but will have more time to react. Those that present late will have more time to prepare and mature their solution but will have less time to react.

From a grading perspective, the Oral Presentation portion of this project is your "Project" and your Product Delivery is your "Final Exam." In other words, your Oral Presentation will be recorded as your "Project" grade, and your Product Delivery will be recorded as your "Final Exam" grade. There is no other Final Exam for this course.

2 Teams

You are to work in teams of three students. Reach out to your fellow classmates, select your teammates, and agree upon an available timeslot to conduct your Oral Presentation. The Oral Presentation schedule is maintained on our class' website in TITANium. Each student must then schedule themselves in the appropriate timeslot by the due date. This serves both to reserve your timeslot and to acknowledge your membership and commitment to the team. It is each student's responsibility to schedule themselves. Students who have not scheduled themselves by the due date will be assigned to a team by the instructor. Teams with less than three students after the due date will have random teammates assigned. The team is expected to share the workload evenly. Each member of the team will submit an evaluation of the other members of their team, which provides the instructor with insight on how the earned grade should be distributed amongst the team members.

¹ Presentation layer (client), Application layer (HTML server and PHP engine), and the Database layer

3 Key Dates

EVENT	DUE DATE	WEEK NO.
PROJECT KICKOFF:	02-APR	11
TEAM FORMED AND ORAL PRESENTATION SCHEDULED:	09-APR	11
FINAL TEAM ADJUSTMENT²:	14-APR	12
ORAL PRESENTATION CONDUCT:	30-APR thru May 9	15 thru 16
PRODUCT DELIVERY:	14-MAY	17

4 Artifacts

4.1 Oral Presentation

Your team will conduct a 15-minute presentation during class, including a few minutes at the end for questions. During your presentation, you will present your Final Project from 3 required and 3 elective viewpoints, for a total of 6 viewpoints. You are to create a virtual poster board³ with 6 regions labeling each region with the view's name and populating with images taken from your project. Display this poster before your presentation. In addition, prepare a 10- to 12-slide⁴ PowerPoint (or similar) presentation to use as your guide as you present. Remember, you only have 15 minutes and will be cut off when time expires, so plan to fill the entire 15 minutes, but no more.

4.1.1 Required Views

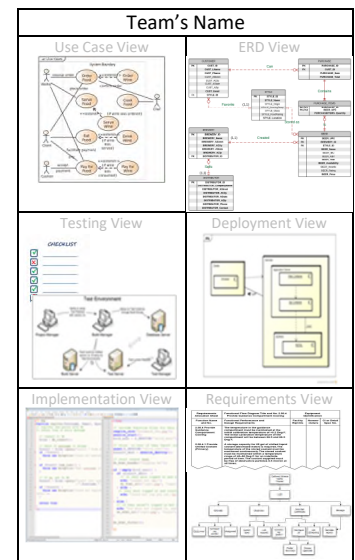
1. Use Case View

Identify the questions your operators will ask of the application. Draw the sequence starting with the forms used and the messages exchanged between the client, application, and database layers. To get you started, consider:

- What teams are in the league?
- What teams did CSUF play last this season?
- What games did Player X play in?
- Who played in Game X?
- Who coached last week's game?
- Who has an account and what is their role?
- Who has logged in in the past 48 hours?

You can also show (describe) activities for each role, for example

- "observers" can
 - Read non-sensitive data
 - you define what this non-sensitive data is
 - change their own password



² This is a rare event, but available upon approval if team dynamics warrant. No team membership changes hereafter

³ A physical 24 x 36" or 27 x 39" poster board is always welcome, but not everyone can do that. Printing each section on standard, legal, or ledger, sized paper and taping together is fine. It only must last for a single class period.

⁴ 6 of the slides will match your poster. The others should be used as supporting or amplifying information. Additional views could also be included if desired.

- “users” can
 - Create, Read, Update, and Delete non-sensitive data
 - Same non-sensitive data as you defined above
 - change their own password
- “executive managers” can
 - Create Read Update Delete sensitive and non-sensitive data
 - reset anyone’s password,
 - assign an existing user to a different role
 - but cannot create or alter the database schema.
- “DB administrators” can
 - do everything/anything

2. Entity Relationship View (ERD)

3. Requirements View

Use Cases are a form of functional requirements and can complement this view. But some requirements are not easily expressed in a use case. To get you started, consider:

- “shall” statements
- constraints
- roles and access
- webpage content and navigation
- legal values, ranges, etc.
- negative paths and feedback

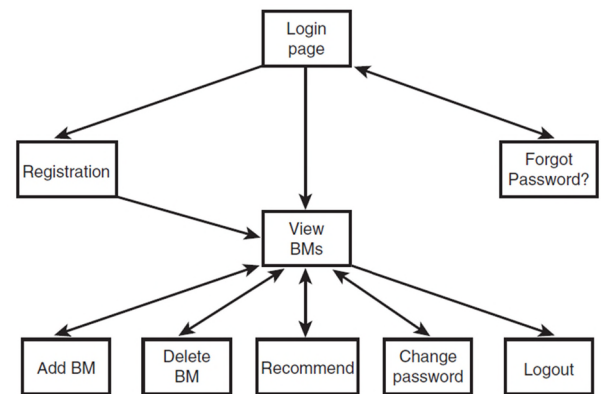


Figure 27.1 The possible paths through the PHPbookmark system

Webpage navigation Requirements

4.1.2 Elective Views

1. Deployment View

- software and versions, hardware, network
- Client
- Web server
- PHP engine
- Database server

2. Security View

- CIA

3. Implementation View

- source code
- programming languages

4. Testing View

- Test cases (should match use case),
- Results and action taken
- Testing environment(s)

5. Configuration Management View

- How did you control the code, including the DDL, amongst your team?
- How well did it work?

4.2 Product Delivery

Login to your application and execute your key use cases for each supported role taking many screenshots along the way. Then archive (zip, tar, whatever) everything into a single file and submit:

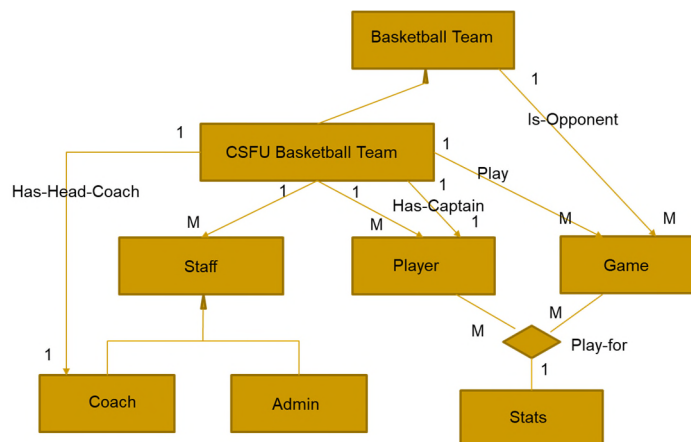
- Team Member Responsibility Table
- Oral Presentation material plus a description of actions taken since
- DDL scripts
- PHP scripts
- Test data
- Other relevant information



Team Member
Responsibility Table.

5 Database

You are to define and design a database schema to support your application. The following diagram was used in class to discuss types of relationships and entities you might anticipate and is offered here to get you started. Your entities and relationships should support the questions your users will ask, which should be captured in your use case and requirement views.



Remember that the entities here, such as Player, represent the contents of your data and not the users of your system. In addition, you will need to provide the entities to store user account information.

6 User Authentication

The *Implementing Authentication Methods* concepts described in chapter 16 and the *Logging in and authenticating users* and *Managing password* concepts described in chapter 27 should be incorporated in this project. It reads, in part (page 562)

- Users should be able to register their usernames and passwords. You need some restrictions on the length and format of each username and password. You should store passwords in an encrypted format for security reasons.
- Users should be able to log in with the details they supplied in the registration process.
- Users should be able to log out after they have finished using a site. This capability is not particularly important if people use the site from their home PC but is very important for security if they use the site from a shared PC such as at a library.
- The site needs to be able to check whether a particular user is logged in, and then access data for the logged-in user.
- Users should be able to change their passwords as an aid to security.

- Users should be able to reset their passwords without needing personal assistance from you. A common way of doing this is to send a user's password to him or her at an email address he or she has provided at registration. This means you need to store the user's email address at registration. Because you store the passwords in an encrypted form and cannot decrypt the user's original password, you actually need to generate a new password, set it, and mail the new password to the user.

7 Key points I'll be looking for

- ✓ Does your DDL script create the database and schema required to support your application. I will run the script from a privileged DB account (e.g. root) several times, so be sure it can both construct your database from scratch and remove and replace itself thereafter. Be sure to populate the entities with any necessary data.
- ✓ Does your application run? I'm looking for how your application works, not that it works.
- ✓ Is your database schema engineered reasonably?
- ✓ Is data passed between the architectural layers reasonably?
- ✓ Are connections made to the SQL server reasonably, especially with the right accounts?
- ✓ Are your queries formed reasonably?
- ✓ Is data received from the client sanitized before use?
- ✓ Have you eliminated SQL or HTML injection vulnerabilities?
- ✓ Have you implemented role-based access? That is, do you allow *observers* to view non-sensitive data; *users* to view and modify non-sensitive data; and *executive managers* to view and modify both sensitive and non-sensitive data?
- ✓ Have you identified what data is sensitive and what is non-sensitive?
- ✓ Have you implemented the *Logging in and authenticating users* and *Managing password* concepts, including emailing a user a new password after he or she has reset it?
- ✓ Have you identified, implemented, and tested your key Use Cases? Do you have documenting screenshots?
- ✓ Are your views fairly complete and accurate?
- ✓ Has each team member contributed?
- ✓ Database design, session variables, user authentication and password management, and role-based access are the new things since the midterm and will be looked at more closing.
- ✓ If time runs short, in high to low priority order, it more important to have:
 1. Submitted something on time
 2. Completed Team Member Responsibility Table
 3. Completed Teammate Assessment
 4. Database schema designed, implemented, and the DDL scripted
 5. Login authentication and password management using session variables
 6. Role-based access
 7. One working use case for each role
 8. Reset password with eMail message