CMPSC 431W Project Guideline

This project asks you to design, implement, document, and present a database-backed web application for an **online bookstore**. This project aims not to produce a complete implementation of the system, as it would require too much time and resources for the students. Instead, students will focus on i) the process of the database design that may pose a significant risk due to uncertainty and the lack of understanding in requirements, and ii) the prototyping of certain system functions as a proof of concept. It is anticipated that based on careful examination of the tested prototypes, valuable insights and lessons may be obtained. Indeed, this design-prototype approach of the feasibility study is often adopted when the requirements are not well understood. Moreover, a successful prototype can potentially serve as the foundation for the future production system in operation. Besides, an impressive demonstration of the prototyped system, functions, and unique features may attract your prospective employers when listed on your CV.

As part of the prototype, you will need to implement a number of programs that access data in the designed database in order to support the functions of the system. The project consists of two phases:

1. Requirement Analysis, Conceptual Database Design, Technology Survey, Logical Database Design, and Normalization. **(due on March 31)**
2. Protype Implementation **(due on Apr 30.)**

The first Phase is to, based on the provided project description in this guideline, analyze the requirements to specify its system functionality and identify data needed for the system functions and integrity constraints to be imposed upon the needed data. In addition, it also requires you to come up with a conceptual database design by using the *entity-relationship* model to expressed the data and constraints identified. Furthermore, students shall perform a technology survey by researching on the current web/database application technologies, including web frameworks, programming languages, development tools, and database management system, for comparison with those suggested by the CMPSC431W teaching team, i.e., *Flask Web Framework, Python, PyCharm IDE*, and *SQLite*. Additionally, logical database design and schema normalization is to be performed in the first Phase. Each student will submit a well-documented report for the tasks in Phase 1. After this detailed design process, in Phase 2, students will populate the database and implement the system functions.

<u>You will have to carry out this project entirely by yourself as it is an individual project.</u> You should also strictly adhere to the academic integrity rules, which prohibit you from plagiarizing code, design documents, and presentations from your peers or online. ***Your submission will go through automated anti-plagiarism tools for enforcing that.***

OVERALL REQUIREMENTS

Before we jump into the description of each Phase, let's talk about what you are expected to fulfill while working on the project.

Document formatting

We will provide a document template (as .docx file) that you are more than welcome to follow directly. However, you are encouraged to create a template of your own, as long as it maintains a professional image for your cover page, and includes all the components from our provided template. You can also write your documents in LaTeX, MS Word, or other document typesetting languages or tools. Your report should be converted into one PDF file for submission **electronically** on Canvas**.**

Project Management

The success of this project heavily relies on your own personal effort and commitment. This project is segmented into two phases. Thus, there will be no intermediate submissions. Students are expected to set goals and maintain individual milestones, documented as part of the Phase 1 report. For those who do not have prior background on web programming, it is highly recommended to start reading documentation and tutorials during Phase I.

Since there are numerous web application tools and frameworks, it is impossible for the teaching staff to provide guidance on every combination of tools. While students have a choice to use the recommended platform and tools (i.e., *Flask Web Framework*, *Python*, *PyCharm IDE,* and *SQLite)* which the teaching staff are more than happy to provide guidance, you may choose to adopt platforms, tools, programming languages and database management systems of your preference (but you are on your own in this case).

Finally, start EARLY, ideally from day one. The conceptual and logical database design part (phase I) should not be hard, considering we have covered all of it in class. Thus, you will spend most of your time coding and debugging. Even though it is not particularly challenging to implement each part, you still need enough time to plan and execute.

PHASE I – DATABASE DESIGN AND TECHNOLOGY SURVEY

Checklist

Tasks
Requirement Analysis
Conceptual Database Design
Technology Survey
Logical Database Design and Normalization

Task 1: Requirement Analysis

In this task, you need to specify in detail the system requirements, including system functions, the types of data that must be stored, the integrity constraints to be imposed on the data, and how it will be stored and accessed in support of the expected system functionality.

Note: For Task 1, you are expected to **elaborate** the system functionalities in your own words (and illustrative figures) with more details. In addition, you should identify (also **write down**) the data needed to support the functionality as well as the integrity constraints to be imposed on the data.

A description of the core functionality and data items follows. Some functions will be provided to the customers and others to the manager of the bookstore.

The system should manage information about books in the bookstore, inventory, (registered) users, and books they have ordered. It should also store information about user comments and book ratings. While you are working, keep in mind that these items are a minimal set of requirements.

- **Book Data:** An ISBN for each book, title, author(s), publisher, language, publication date, number of pages, stock level in the inventory, price, keywords, subject (Sci-Fi, western, history, etc.). You should add additional information if necessary.
- **Customer Data:** For each registered customer, you need to maintain: his/her full name, login name, password, address, phone number, books he/she has ordered and information regarding the order.
- **Comments:** Users can provide feedback for a book as a score (1-10) along with optional short text. Users can also rate other users' feedback as useless, useful, very useful; finally, they are allowed to declare other users as trusted or not-trusted

Your system should support at least the following events. **Moreover, you should add at least 3-5 additional functionalities that you believe beneficial to the system.**

1. **Customer Registration:** a new user has to provide the appropriate information; he/she can pick a login name and a password. Login names of the users must be unique to each other.
2. **Multiple Managers:** there can be multiple users who can access the full functionality of the system. For example, only managers can create a new book while the user cannot. Note

that only a manager can register a new manager to the system. There should be a superuser being the first manager when the system starts for the first time.

3. **Ordering:** After registration, a new user can order one or more books. The total amount of its order is reported to him/her. A user may order multiple copies of a book one or more times.
4. **New book:** The store managers may record the details of a new book and the number of books that have arrived in the warehouse.
5. **Arrival of more copies:** The store managers may change the stock level of a given book.
6. **Comments:** Users can record their comments for a book. You should record the date, a numerical score (0= terrible, 10=masterpiece), and an optional short text. A comment can only be changed by the customer who made it; only one comment per user per book is allowed.
7. **Usefulness ratings:** Users can access a comment and give it a usefulness score in three levels, namely, useless, useful, and very useful. A user should not be allowed to provide a usefulness rating for his/her own comments.
8. **Trust recordings:** A user may declare zero or more other users as trusted or not-trusted.
9. **Book browsing:** User may search for books by asking conjunctive queries on authors, and/or publisher, and/or title words, and/or language. Your system should allow the user to specify that the results are to be sorted (a) by publish date or (b) by the average numerical score of the comments, or (c) by the average numerical score of the trusted user feedbacks.
10. **Useful Comments:** For a given book, a customer could ask for the top n most useful comments. The value n is user-specified (say 5 or 10). The usefulness of a comment is its average usefulness score.
11. **Buying suggestions:** Like most e-commerce websites, when a user orders a copy of book A, your system should give a list of other suggested books. Book B is suggested if there is another user that bought both A and B. The suggested books should be sorted on the decreasing sales amount (i.e., most popular first) among the users who also bought A;
12. **Degrees of separation:** Given an author X, a user may want to know the books written by authors who are 1-degree and 2-degree separated with X. The degree of separation is defined as follows: two authors A and B are 1-degree away if they co-authored at least one book together; they are 2-degrees away if there exists an author C who is 1-degree away from each of A and B **AND** A and B are not 1-degree way at the same time.
13. **Book Statistics:** Every semester, the store manager wants:
    - The list of m (say m = 10) most popular books (in terms of copies sold this quarter)
    - The list of m most popular authors (in terms of copies sold this quarter)
    - The list of m most popular publishers (in terms of copies sold this quarter)
14. **User awards:** The store manager also wants to give awards to the “best” users; Thus, the manager nees to know:
    - The top m most trusted users (the trust score of a user is the count of users trusting him/her minus the count of users not-trusting him/her)
    - The top m most useful users (the usefulness score of a user is the average usefulness of all his/her feedbacks combined).

Task 2: Conceptual Database Design

Based on the result of Task 1, you should present an ***entity-relationship (ER) diagram*** describing your conceptual database design. Also, your report should include a narrative description on all aspects of the diagram in detail. In addition to an overall ER-diagram of your conceptual design, parts of the ER diagram such as entities, relationships and integrity constraints are expected to be explained in detail. Note that the conceptual database design is application-oriented. ***Please do not assume the underlying database system (just yet) and thus do not map your design into relations.***

Task 3: Technology Survey

An aspect of working in the IT industry is to always keep your knowledge updated with the current market trends and technology innovations. Thus, for the project, it's important for you to have a good knowledge of the current web programming and database technology. A technology survey is meant for you to research various web programming frameworks, programming languages, tools, and database management systems as well as the trends in those technological areas. ***Besides your own choice of developing stack, you have to come up with 2- 3 trending alternatives to make a comparison.*** It is expected for you to write persuasively about which tools you think as the best fit for the project. Please consider as many aspects as possible regarding the roles of those technologies in your project, and to *justify* your own recommendation by discussing the reasons, pros and cons of your choices. Also discuss the impact and relevance of those technology trends to the computer science fields and a broader segment of society or business/industry.

Task 4: Logical database design and Normalization

Based on the Requirement Analysis and Conceptual Database (ER) design you developed in Task 1-2; you will finalize the relational schema for your database. You should produce a refined schema that reduces data redundancy to an acceptable level (i.e., the final schemas need to be in the Boyce Codd Normal Form) while not unduly affecting performance. Your schemas should also support the enforcement of most, if not all, of the integrity constraints that you identify in this Phase as well as those newly identified or added after relational schemas are generated from the ER diagram. You should present the specific details of how the process of schema generation and normalization is done, i.e., you are expected to apply both of translating the ER diagram to relations and then perform schema normalization. ***Note that even if your schema generated from ER diagram directly is in BCNF, you should justify why it is in BCNF in the report as well.***

Grading

Different from exams, the project provides a way for students to show their commitment and effort in learning and practicing the knowledge learned from CMPSC431W. While the course project is work intensive, students who show their strong motivation and effort through the high quality of their reports will be rewarded. The project report, fulfilling the Writing requirement of the Penn State curriculum, is graded in terms of correctness, completeness, presentation, and clarity. Again, your effort will be reflected in your report and rewarded. Please be reminded that the project is work intensive -- start early and prioritize your time.

Extra credit of up to 10% will be rewarded for reports who include more than 5 new functionalities (i.e., proposed by you, not stated in the above project description), expressed explicitly and clearly within the report. This is graded on novelty of the new functionality and clarity the student is able to present in various tasks of Phase 1.

Submission

As mentioned above, a template will be provided as a .docx which serves as a mandatory style guideline. **The deliverable is a report in PDF containing the requirement analysis, conceptual database design, technology survey, and logical database design and normalization.** The document must have page numbers, section numbers and a table of contents. Figures should be used for illustration of your design. In addition, the document should also include a project plan (including schedule, deliverables, and milestones) as an appendix in the project report. It is expected that the report be at least 10 pages (**not including the cover page and table of contents**). *Turnitin* will be where you submit a .PDF version of your report. As stated in classes as well as the syllabus, academic integrity is the most important and there will be no tolerance and exceptions towards violation. An incomplete but attempted submission is always far better than committing an academic integrity violation.

PHASE II – SYSTEM PROTOTYPE

As we have a logical design for the database supporting our application, we now move on to validate the idea by prototyping a web application based on it and see how it works. We will provide some book information from a legacy system in a CSV file that you can use to populate your database to start with. Besides, you should fill other tables in your database with some data so that every proposed functionality in Phase I can be tested.

In this phase, we will test and evaluate **the student implementations based on the ability to execute designated tasks and the overall user experience from using the website.**

Checklist

Tasks
Database Population
Functionality Implementation
Reflections and Final Deliverables

Project description

In Phase II, we will provide a CSV file containing some raw data for books, and you are supposed to reorganize the raw data in accordance with the schema and populate the database. Besides, you will implement the web application for the online bookstore and demonstrate its functionality as detailed in Task 2 below.

Task 1: Database population

The first task is to create and populate the database. You will be provided with a data file where the columns contain values for various fields. However, please note that those are RAW data. In other words, do not simply treat those spreadsheets as 'tables' in the database since they may not comply to the integrity constraints. You are supposed to organize the raw values in the files based on the schema provided to create all of the necessary tables, define primary keys, foreign keys, and specify any integrity constraints as appropriate. In addition, define any views of the schema as you see appropriate. Again, please inspect all the columns and the corresponding data types in the CSV files before you proceed to populate the database. ***To support additional functionality you wish to add, you are encouraged to augment the schema and add your own data.*** You can use Python libraries like Pandas or other database facilities/tools to parse the CSV files and insert values to the table. You should provide (and maintain) one or more SQL scripts used in your data population, so that you can easily regenerate the database or restore the database states.

Note that ***you may only need a few columns in the raw data we provide for books***, and ***you are responsible for populating other tables with a reasonable amount of data*** as well. You have to make sure all functionalities (including the ones you proposed) can be tested upon your populated database.

Task 2: Functionality implementation

You should prototype (i.e., implement) the system functionality as specified below and be able to demonstrate the following operations on your final system prototype. The following is a description of the core expected functionality.

1. **User Login**: This is for a user (a customer or a store manager) to log in. The system recognizes the user by his username and password. Please be sure that the user's **password is hashed when you store it and it is not displayed while being entered.** Managers should have access to a **dashboard** (which customers do not have access to) that provides additional management functionalities.
2. **Registration**: A new customer should be able to register a user by themselves. A new store manager can only be registered through another manager's dashboard.
3. **Browse Book**: Users should be able to search for books based on author/publisher/title/ language and ordered them by year and average scores. On the page for each book, customers should see all book information except for the stock-level (but they should know if it is out of stock) and all comments by other users. Besides, customers should be able to comment on the book and mark usefulness scores to other comments. There should also be an option only to show top n useful comments for the current customer.
4. **Browse Customer Profiles**: A customer should be able to review other customers' basic information, some of their comments, and the number of trusters/non-trusters through a profile page. Besides, he/she should also be able to mark "trusted" or "not-trusted" on that page as well.
5. **Book Recommendation:** For each customer, you should provide a page specialized for him/her recommending other books he/she didn't buy.
6. **Degree of separation search**: There should be a search functionality embedded in the system to look up books related to a given author in the sense of direct authorship or 1/2-degree of sepeartion authorship.
7. **Placing Order**: On the page for each book, a user should be able to place an order of a particular number of copies. *You are not required to support a shopping cart and multiple different books in the same order, but you will be awarded a bonus if you do so.*
8. **Stock Management**: Managers should be able to add a new book or change the stock level through their dashboard.
9. **Reviewing Statistics**: Managers should be able to review statistics about book sales information and customers through their dashboard.

Note that you are welcome to add more functionalities to the system in addition to what listed above. If you do so, you will be awarded bonus points based on how much extra work you are doing.

Please note that while the correctness of functionality above is critical, the design of a user-friendly web interface is also very important to the success of the prototype. Please pay attention and effort to the web design of the prototype.

Task 3: Reflection

At the end of the project, it's beneficial for you to flashback what you expected at the beginning of the semester regarding the project and reflect on whether you meet the goals you set back then. Make a comparison between what was expected and what has been achieved. Discuss the lessons learned – for example, why some parts of the required functionality were not completed (if any), how you added some new features (and why, if any), how you would proceed with the project the same or differently in the way you do in the course, and your thoughts about the project, etc. Please plan and include the reflections as part of your video presentation (see more details of the video presentation in the Deliverables section below).

Final Deliverables

The final deliverables in Phase 2 include **a video presentation and a demonstration of your system**. In your video presentation (no more than 5 minutes), please discuss the important design and implementation details you considered and make a reflection. Your video presentation should be submitted via Canvas. For help with making the video presentation, please have a look at: https://mediacommons.psu.edu/support/tutorials/

In the project demonstration, you are expected to show the prototype implementation of your system to the teaching team. We will use the website and check some/all of the compulsory tasks with data from the CSV files provided to you. You may also show the optional functionality for extra credit. Please be well-versed with your code as we may ask you to explain the web server code and the queries you wrote. Please note that all files of your code have to be submitted on **Canvas** in a zip file. You have to include a ReadMe file to describe your code functionality and control flow. Your code has to be properly documented for full credits.

As stated in classes and the syllabus, academic integrity is the most important -- there will be no tolerance and exceptions towards violation! An incomplete but attempted submission is always far better than committing an academic integrity violation.

Your extra efforts will be rewarded with extra credits!