CPSC Distributed Library

Database Design

Problem:

There are, within the offices of the Computer Science Department faculty, many textbooks that could be circulated to other faculty and students if

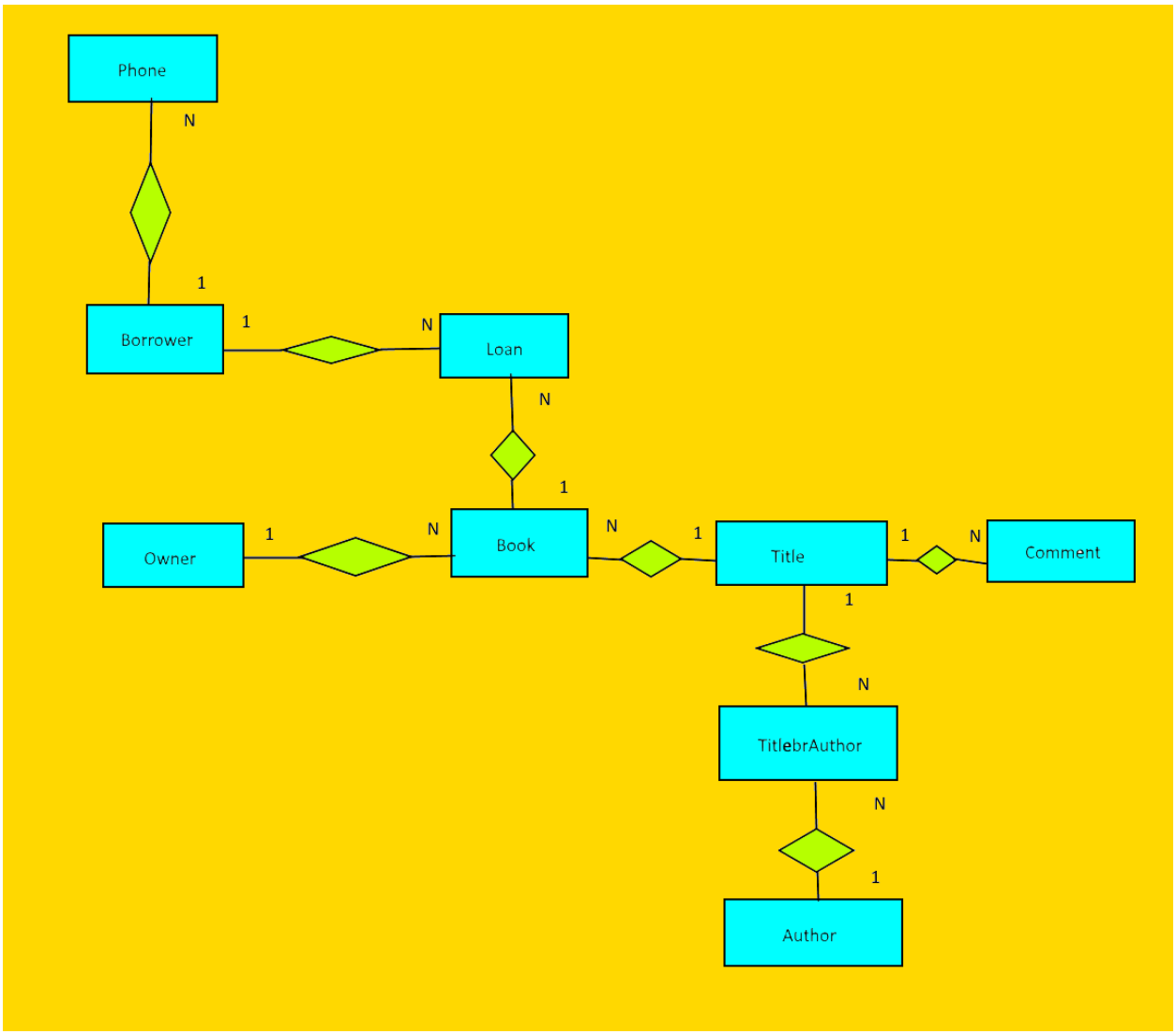
1. it was possible to view a complete listing of the books in the collection
2. there was a way to check books in and out (keep track of other faculty and students who borrow them)
3. the availability of a book could be determined by anyone interested in using it
4. there was information available regarding the usefulness of a book

Solution:

Create a distributed library with an electronic inventory system that can be accessed to

1. List all books contained in the collection. Being able to list books by author, title, and content area (DB, PL, etc.) would be very useful.
2. List all books from the collection that are available to be checked out.
3. List information needed to order a new book (title, publisher, author, isbn, etc.)
4. Submit notes about a book's content.
5. Read notes submitted by previous users.
6. Allow a user to make a request to borrow a book.
7. Record check-out information if someone wants to borrow the book. Useful check-out information might include borrower's name, phone, and email as well as date borrowed and estimated return date.
8. Record check-in information.

ERD:



Example Use Case Diagram:

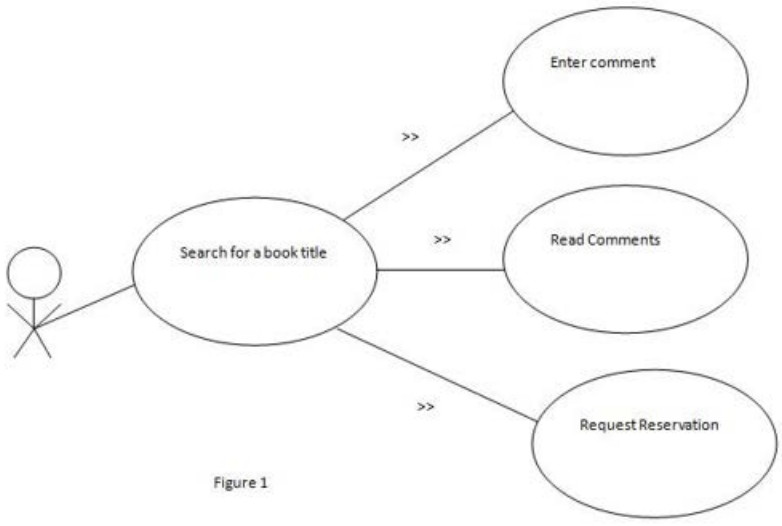


Figure 1

Example Use Case Narrative:

Example Flight Reservation Itinerary Use Case

Craig Borysowich (Chief Technology Tactician) posted 12/1/2007 | Comments (1)

Based on the original problem statement, here is a sample narrative style use case:

Name: Change Flight Reservation Itinerary

Last revised: yyyy/mm/dd

Summary:

A traveler who has an active flight reservation requests a flight itinerary change. The system [AutomatedTravelAgent] requests information from the traveler to identify her active flight reservation and flight itinerary change. If the proposed flight itinerary change (or a variation of it) can be made, the system effects the change. If the change cannot be made, the system explains why the proposed flight itinerary change request cannot be satisfied.

Actors: Traveler

Pre-conditions:

The traveler has a valid system user and a valid travel account and an active flight reservation.

Basic course:

The traveler requests a flight itinerary change. The system retrieves current information about the status of the traveler's account, and asks the traveler to identify her active flight reservation, which can be identified by specifying either the record locator or the flight number, departure airport and departure date. [Exception: No active flight reservation located.]

The system retrieves and displays the current flight itinerary and requests information about the proposed flight itinerary. The system checks to see if there are any available flights that satisfy the constraints of the proposed flight itinerary change. If there are available flights the system displays their flight schedules and airfares and requests that the traveler: 1) select one of the available flights ; 2) reformulate the proposed flight itinerary change; or 3) cancel the proposed flight itinerary change. If the traveler reformulates her proposed flight itinerary change the system processes the request as before.

If the traveler selects an available flight the system reserves the new flight, cancels the old flight, and updates the traveler's current flight itinerary and travel account.

If there are no available flights, the system displays an explanation and recommends an alternative flight itinerary. [Exception: No more alternative flight itineraries to recommend.]

The use case terminates when the traveler selects one of the available flights or cancels the proposed flight itinerary change request.

Alternative course: N/A

Exceptions:

No active flight reservation located: This exception is raised if the agent cannot find an active flight reservation. The traveler information is checked again for accuracy and the search repeated. In the event that the search is again unsuccessful, partial match searches are initiated. If all search efforts fail, the system informs the traveler that no flight reservation is locatable. The use case terminates.

No more alternative flight itinerary changes to recommend: This exception is raised if the agent has no more alternative flight itinerary changes to propose based on the traveler's constraints. The system informs the traveler that there are no more alternatives and asks if she wants to cancel her active flight reservation. [See Cancel Flight Reservation use case.] The use case terminates.

Post-conditions:

A traveler either retains her original active flight reservation or receives an updated flight itinerary with a new flight reservation. Otherwise: none

Assignment:

Working in teams of two or three, create use case diagrams and narratives for the following use cases:

1. Search for a book by title (Borrower)

In the offices within the computer science department faculty there are many textbooks that are not currently in use. These textbooks could be circulated to other faculty and to students if there were a way have a listing of all of the books by author, title, and content area. A database solution would contain a listing of the books with information about the title, publisher, author, and ISBN.

In making this database there needed to be a way for a student or faculty to search for a book by title, in order to think about becoming a borrower. The borrower should be able to search the database, and get a return listing of the faculty members who have the book, as well as an indication of the book's availability.

2. Reserve a book (Borrower)

Once students decide on a book that is best for them from the faculty collection of textbook, there needs to be a way for them to become the borrower and put their name on the book, before they are able to come to get it. Once a student has decided on a book that they want to reserve and the database allows them to mark it, it needs to be visible to other students and faculty that this book is no longer available. Students and faculty who are looking for a book will then be able to know to look for something else when they are also looking to become a borrower.

There then needs to be a system in the database that will show once they have gotten the book and how long they are allowed to keep this book before it needs to be returned. The database should be able to show students this so that if the book that is gone is the specific book that they are looking for, they will know when they can expect to come back and get it.

3. Enter a comment (Borrower or Owner)

Another problem that would need to be solved is the need for a way to enter comments about the book's content area and usefulness. A student who has become a borrower may realize that the book was not as useful as that student initially though, and should be able to communicate that information on in the form of a comment. Likewise the faculty who is the owner of the book needs a way to enter comments regarding the book. The owner has likely read the book before, and so can more accurately enter comments regarding the content area of the book and the overall usefulness from a teaching standpoint.

4. Read a comment (Borrower or Owner)

An aspect of the database that would be helpful to other students and faculty, would be for them to be able to look at certain selections and see comments that have been written. Both the owner and the borrower need to be able to write comments about the book after they have read it to help other borrowers know if this is the right book for them. Future borrowers need to have both borrowers and the owner's opinion because they will be able to decipher if the owner's opinion in bias in any way.

Additionally, this is helpful for the owner because the owner is able to see what the borrowers are thinking of this book. This will help them in being able to guide future borrowers. They should be able to comment back to comments by borrowers, or edit their own comment to reflect more as they have more knowledge of the book and it's applications.

5. Check-out a book (Owner)

In order for this database to be useful there needs to be a way to keep track of which books are currently checked-out by a borrower. In this way the owner would always know what borrower had what book, and whether or not a book was available. The owner needs to be able to check-out a book to a borrower and update the book's availability in the process. When the owner checks-out the book they need to record the borrower's name, phone, and email, as well as the date borrowed and the estimated return date.

6. Check in a book (Owner)

It is not enough for the database to be able to keep track of what books are being checked-out, but the borrower needs to be able to return that book and have this reflected in the database as well. The owner will be able to see what books have been brought back and they will again be available for more borrowers to be able to check this book back out. This will reflect if the book was checked-in by the time it was supposed to be and the database should reflect if there needs to be any fee for anyone who was late in checking their book back in.

7. List all books that are checked out (Owner)

From the database standpoint there should be a way to list all of the books that are currently checked-out. It should list the book's owner as well as who the borrower is, and the borrower's contact information (phone and email). The listing should also return the date borrowed for all of the checked-out books, as well as the estimated return date.

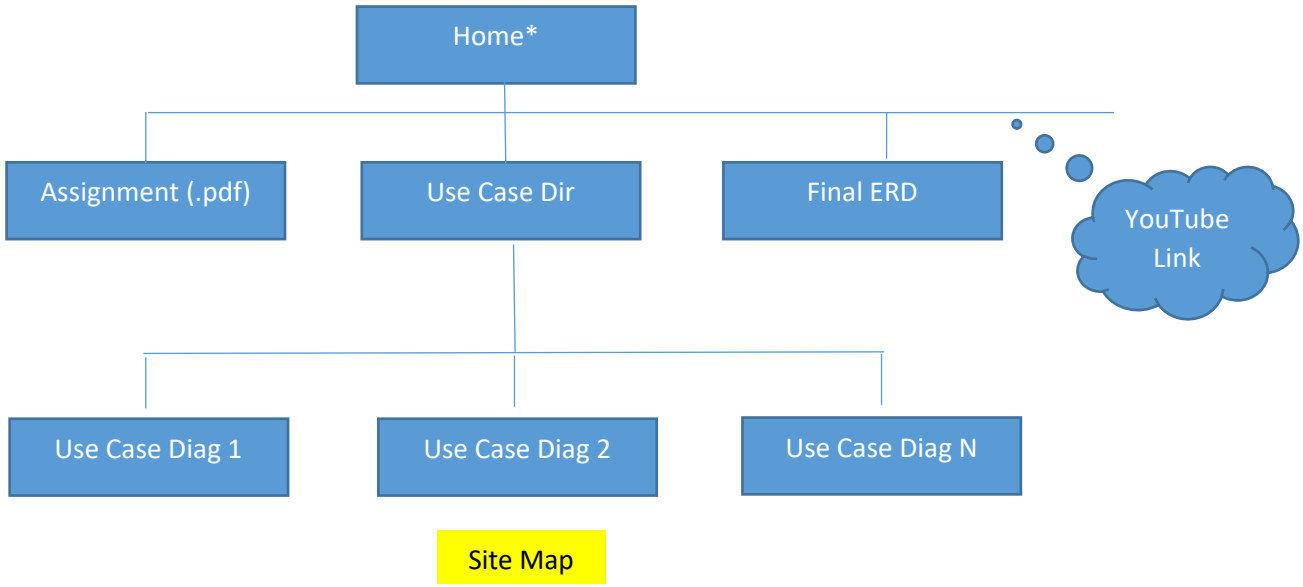
8. List all books by author and content area (Borrower or Owner)

The database needs to be able to show all of the books available so that people will know what their options are. This needs to be sorted by author and content area so that if borrowers do not know the book that they are looking for, they are able to look at certain types of books and find ones that will fit their need. It is important that both the borrower and owner can use this. For the borrower, it is a simple way for them to find what they are looking for, no matter how they go about looking. For the owner, it is a way for them to know what all books they have available and where they could grow their collection in area's that might be lacking.

Design the use case diagrams so that they provide links to corresponding use case narratives – use case diagram acts as image map link to use case narrative. Use case narratives should show potential entities and attributes (significant nouns) highlighted.

When you are done, harvest your use case narratives for nouns, create a non-attributed ERD then, continue to use the use cases to develop a fully attributed and normalized ERD. Eventually, you are to transform your ERD into an image map with each entity linking to the normalized collection of attributes*. YOU ARE NOW READY TO IMPLEMENT YOUR DESIGN IN Sql Server.

Submit your project on Tuesday 10/31/2016. Turn in a folder with properly linked content (see diagram below) along with a YouTube link to a (team) video walkthrough*.



Site Map

Attribute Name	Key (PK/FK)	Data Type (string, Numeric, Date)	Check Constraint	Allow NULL (t/f)
BOOK_ID	PK	Numeric - AutoNumber		f

* Normalized Collection of Attributes – do this for each entity

Video Walkthrough:

- Length: 10 – 15 minutes with each member contributing equally
- Content:
  - Introduction of team members
  - Overview of problem

    - Overview of design process: Problem Description >> Identify actors and Use Cases >> Model Use Cases with Use Case Diagrams >> Model each Use Case Diagram with Use Case Scenario >> Harvest Nouns …
- Design Features:
    - Note any contentious aspects of the design (differences between individual and final group design)
    - Note any interesting aspects of the design related to cardinalities, check constraints, etc.

The video will conclude with a pie chart whereby each team member will summarize what they contributed to the project.

The "home page" will provide a link the YouTube video and a listing of team members.

Resources:

You will need the following:

1. HTML 5 code generator (Expression Software Suite is a free download or I have it on DVD).
2. Drawing tools for ERD and Use Case Diagrams
3. Graphics editor for cropping and saving diagrams in .png or .jpg file format.
4. Video recording software (Trial version of Camtasia, QuickTime, and Expression Encoder are all possibilities).

Grading Criteria:

- Accuracy, quality, and completeness of documents (ERD, Use Case Diagrams, Use Case Scenarios, Web Pages, Image Maps [50%]
- Quality and completeness of YouTube Video [20%]
- Accuracy, quality, and completeness of Sql Server DB [30%]