CS4400 Database Project
Spring Semester 2015
(Version 1.3)

Look at the last page for edits made for each version

Please read the entire description of the project before starting to work on it.

Purpose of the Project

Analyze, specify, design, implement, document and demonstrate an information system application to support a library. You are required to use the Classical Methodology for Database Development. The system should be implemented using a relational DBMS that supports standard SQL queries. Class administrators will provide you with information about how to access a college-managed MySQL server in order to implement your database and the application. The professors must approve any other alternative implementations. *In no circumstances can you use a tool that automatically generates SQL or automatically maps programming objects into the database.*

Project Phases

The three phases of the project cover the following work-processes from the Classical Methodology for Database Development (see notes on T-square under resources). Slides on database design methodology will be useful for phases I and II: These slides have been posted on t-square.

Phase		Due Date
I	Analysis & Specification	Feb 5
II	Logical Database Design	Mar 12
III	Implementation & Testing	April 21
	Demonstrations	April 22 to April 24

Groups

Project groups may have 3 or 4 members. Groups of more than 4 or less than 3 will NOT be allowed. You are allowed to form groups across the two sections (A & B) of the class.

A group may remove a member from further participation in the group when Phase I is turned in or when Phase II is turned in. A written notification with a proper justification must be provided to the professor and the head TA at that time.

Deliverables

Phase I (Soft copy and hard copy)

The deliverables include:

1. A cover page listing all members in the team with their respective sections and email addresses and T-square username.
2. Enhanced Entity Relationship (EER) Diagram
3. Information Flow Diagram
4. A list of logical constraints that will be enforced. Do not include any constraints that can be shown in the ER diagram, but rather semantic, business logic related constraints. You are required to include at least three constraints, although a fully-specified system will probably have more than that. Constraints that can be specified directly using ER notation will not count toward the three required. Constraints related to data type are not accepted as constraints.
5. Any assumptions made including explanations.

Notes:

1. The EER must capture the constraints of the system as much as possible whenever applicable, i.e. total participation, super/sub class, weak entities.
2. The design of your system must satisfy all the constraints. You are allowed to make up additional assumptions and constraints as long as they do not conflict with the specified constraints and requirements. If possible, those additional assumptions and constraints should be included in the ER diagram.

You <u>must turn in a hard copy</u> of your report in class. One hard copy should be turned in for the entire group, although <u>each group member</u> should upload a soft copy on t-square individually. Group numbers will be assigned to the groups after they are declared in Phase 1. For all subsequent submissions, please note down your Group Number clearly on anything you record or submit.

Phase II (Soft copy and hard copy)

1. Cover Page
2. Copy of the ER Diagram (either from phase I (with any revisions) or from the solution provided)
3. Copy of the Information Flow Diagram from phase I (either from phase I (with any revisions) or from the solution provided)
4. Relational Schema Diagram (with primary and foreign keys identified, referential integrity is shown by arrows)
5. Create Table statements, including domain constraints, integrity constraints, primary keys, and foreign keys
6. SQL statements for each task (*follow the template in the phase II design methodology*)

Notes: A set of SQL statements may be required in order to complete one task. However, in such cases, the last SQL statement should show the output according to the specification. Views and nested queries may be used to support the tasks. A nested query can be broken down into views to make the query more readable.

Phase III

Prior to the demo, the TA will give guidelines for populating the database with data. The database has to be populated with this data set prior to the demo. **5% will be deducted from the grade otherwise.**

Implement a working application with all functionality described in this document. Your source code should be mailed to the respective TA who grades your project by the deadline.

Deliverables for Phase 3:

When the deadline for Phase 3 occurs (midnight of April 21st), you will be uploading the SQL query text file and all your code on T-square as instructed.

1. Bring your laptop for the demo.

Heavyweight option:

The heavy weight option requires you to develop the entire application as a stand-alone application including the front end, menu options and the control flow.

2. Make sure you have a text file (soft copy) with all your SQL queries only (This is just in case your implementation doesn't work.)
3. Working functional application with embedded SQL statements that accesses your database (This is your actual application.)

Lightweight option:

The lightweight option requires you to do the SQL queries and statements to accomplish each task independently.

2. Make sure you have a text file (soft copy) with all your SQL queries only.

Grading

The project will consist of three phases (deliverables) as well as a final demonstration to the TA.

Phase I and Phase II of the project are each worth 10% credit.

Phase III (20% or 5% credit, depending on option):

Heavy Weight Option (20 %): The students would be required to use the embedded SQL feature of MySQL which allows you to embed SQL statements in a Java program or web application. (You can use whatever programming language you are comfortable with)

Light Weight option (5 %): The students would be required to demo the SQL queries on the MySQL console. They would be required to take the Final exam.

Final Exam (15 %): This would be only taken by students who have opted for the lightweight phase III. Under no circumstances would a heavy weight option student be allowed to take the Final.

Library Management System

LMS is a Library Management system for managing books in a multi-floored High School library in Atlanta which is a 24 hour library. The system would enable the users to borrow books from the library. The books are either reserved for reading in the library itself or available for requesting for checkout. The staff of the library handles check-in check-out procedures using the LMS and is responsible for maintaining the correct book status online.

The following sections contain a functional description of the system along with some screen mockups. Each section would explain a particular functionality and then present an example screen about it. You don't have to follow the UI designs shown, it is rather encouraged that you come up with your own design. A complete reorganization of the user interface is allowed as long as your application supports all the functionality listed below.

You may implement the project as a traditional standalone application (e.g., using Java GUIs) or as a web application (e.g., using a web scripting language like PHP). **There is no restriction on the choice of language.**

1. Logging In

Fig 1. Shows the OLMS login screen.

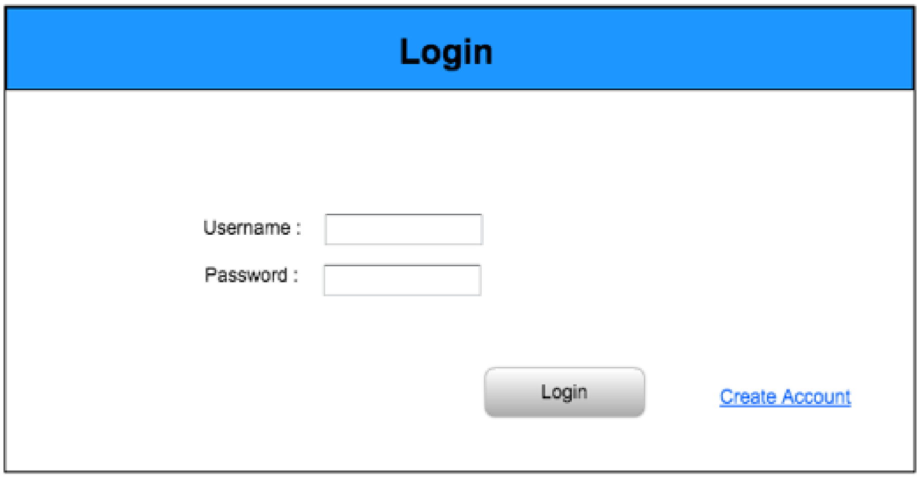


Figure 1. Login

There are three kinds of users of the system namely ‘students’, ‘faculty’ and ‘staff’ each identified by their **Username**. A valid **Username** and **Password** combination is required to log in to the system and proceed. If the user provides invalid log in credentials, an error message should be displayed and the user should be redirected to the log in screen.

2. New User Registration

A new user needs to register before using the system. The staff personnel already have their credentials created behind the scene. Clicking the ‘Create Account’ link on the login screen displays the new user registration form as shown in Fig. 2.

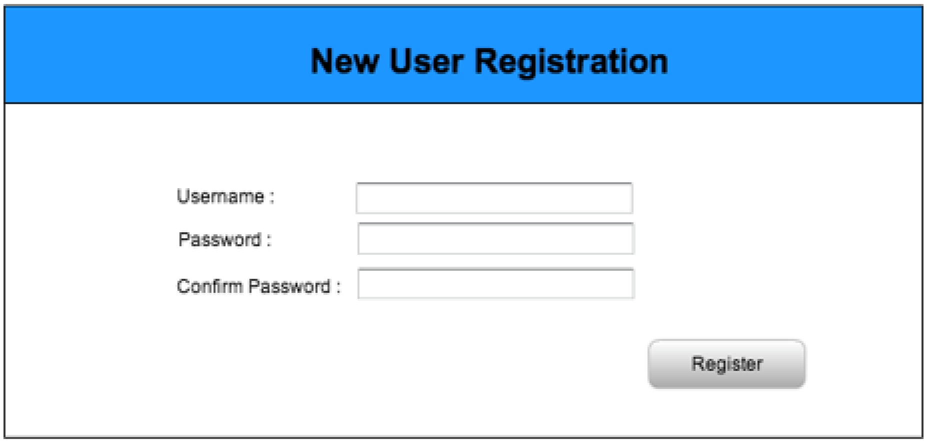


Figure 2. Create Account

After the user clicks **Register**, the system should verify that all fields are filled in, that the **Username** has not already been registered, and that the **Password** and **Confirm Password** fields are equal. If any of these validations fail, the user should be returned to this screen to make corrections. The user should be provided with meaningful error messages so he (from here onwards we would use ‘he’ to represent the user without any intended bias) knows what to correct.

3. Make Profile

After creating an account the user has to make a profile filling their basic information. There should be a check if the user is a faculty or not and if checked yes then the faculty should be asked for his/her associated department in the university. The department field is not rendered/visible for student users.

Fig 3. Shows a mock-up screen.

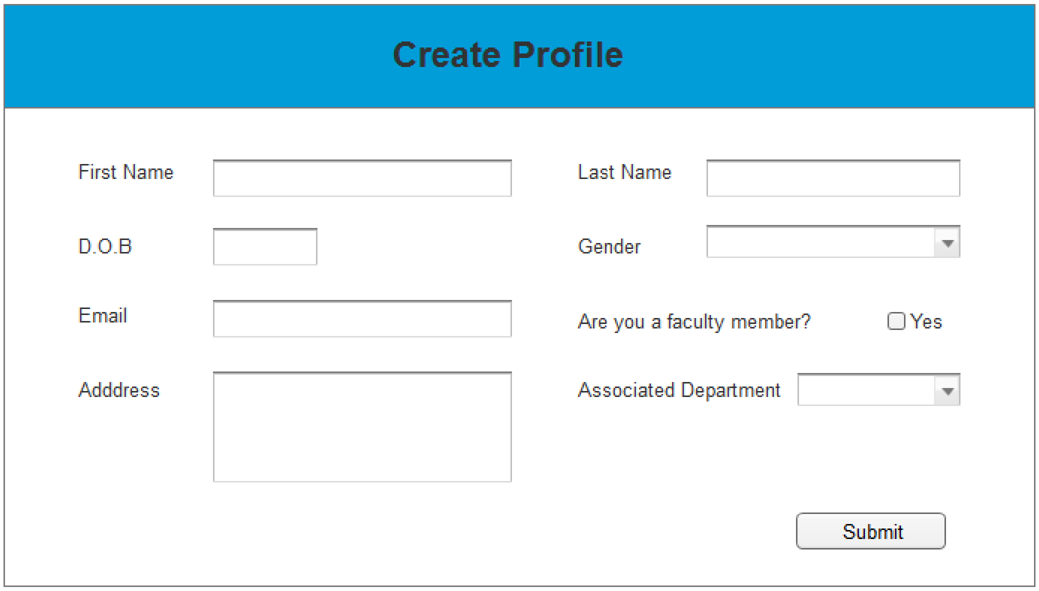


Figure 3: Create User Profile

4. Search Books

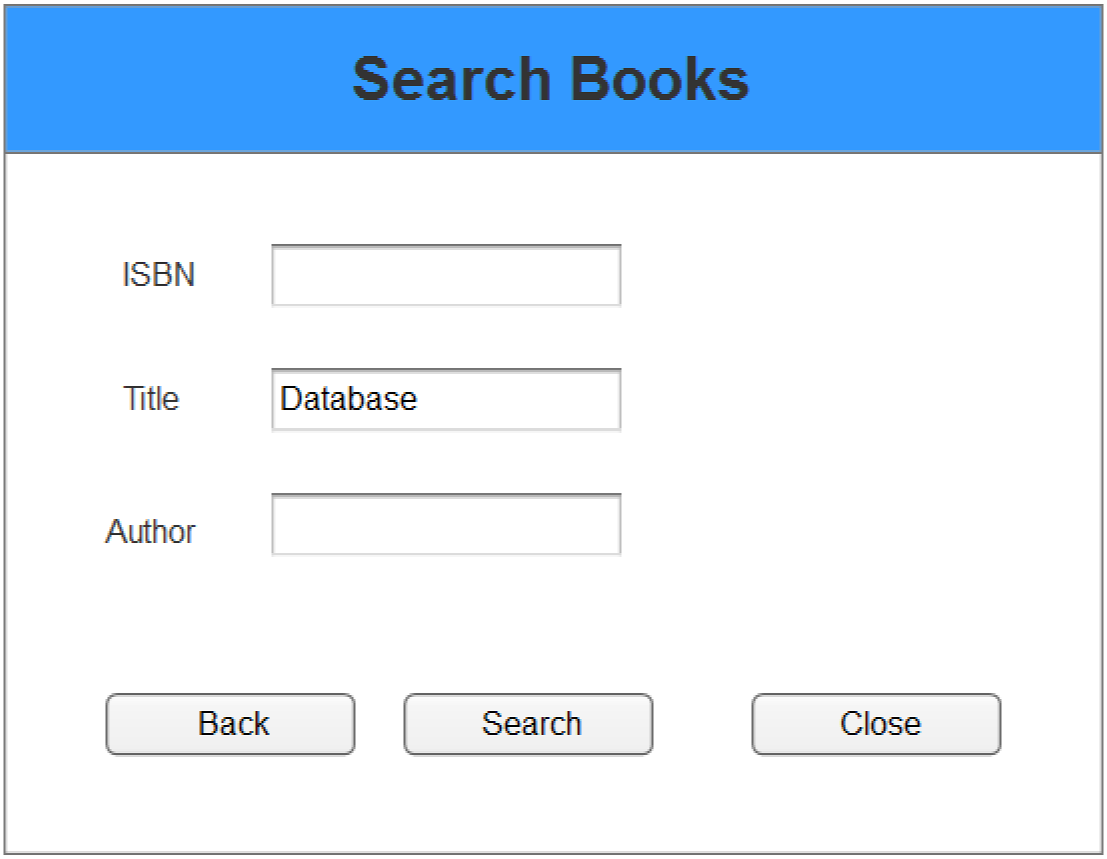
The new user after registration or the existing user after logging in should be taken to the Search Books screen. **Here the user could search a book with its ISBN, Title or Author. Only one of these fields must be used to search for a book.**

<u>For students doing the Light –weight option:</u> Write three separate queries for the three search cases (first for searching the book with ISBN, second with Title, third with Author)

<u>For students doing the Heavy-weight option:</u> You must embed the SQL query for searching the book in your code, but you can take care of the if-

else case scenario from the implementation point of view. (E.g. If($title != null → run the sql query to search with *title*, else → …..)

Fig 4. Shows a mockup screen.



Title Database

Figure 4: Search Books

Each book in the system has information about its ISBN, Title, Author/s, Publisher, Place of Publication, Edition, whether it is a book on reserve or not **(A reserved book is only available to be read in the library not for checkout)**, Copyright Year, and cost of the book. Remember, the ISBN is unique for each edition of a particular book. Multiple copies of the same book edition would have the same ISBN. The staff would track the copies of the same book with the copy# for that particular book. This copy# would mainly be for the staff use so that they can track each copy specifically for checking if the copy is damaged or not, and if the copy has been physically checked out of the library or not. (The copy number would be written on each physical book)

E.g., say the library stores 3 copies of each edition 4 and edition 6 of the book 'Fundamentals of Database Systems', and 2 copies of Edition 3 of 'Data Mining: Concepts and Techniques' book. The information would look like this– Fundamentals of Database Systems*, Edition 4 and ISBN 0-321-12226-7*, and there are three copies for this book numbered copy1, copy2, copy3. Each copy will have the same ISBN 0-321-12226-7.
Fundamentals of Database Systems*, Edition 6 and ISBN 0-136-08620-9*, and there are three copies for this book numbered copy1, copy2, copy3. Each copy will have the same ISBN 0-136-08620-9.
Data Mining: Concepts and Techniques, Edition 3 and ISBN 0-123-81479-0, and there are two copies for this book numbered copy1, copy2. Each copy will have the same ISBN 0-123-81479-0.

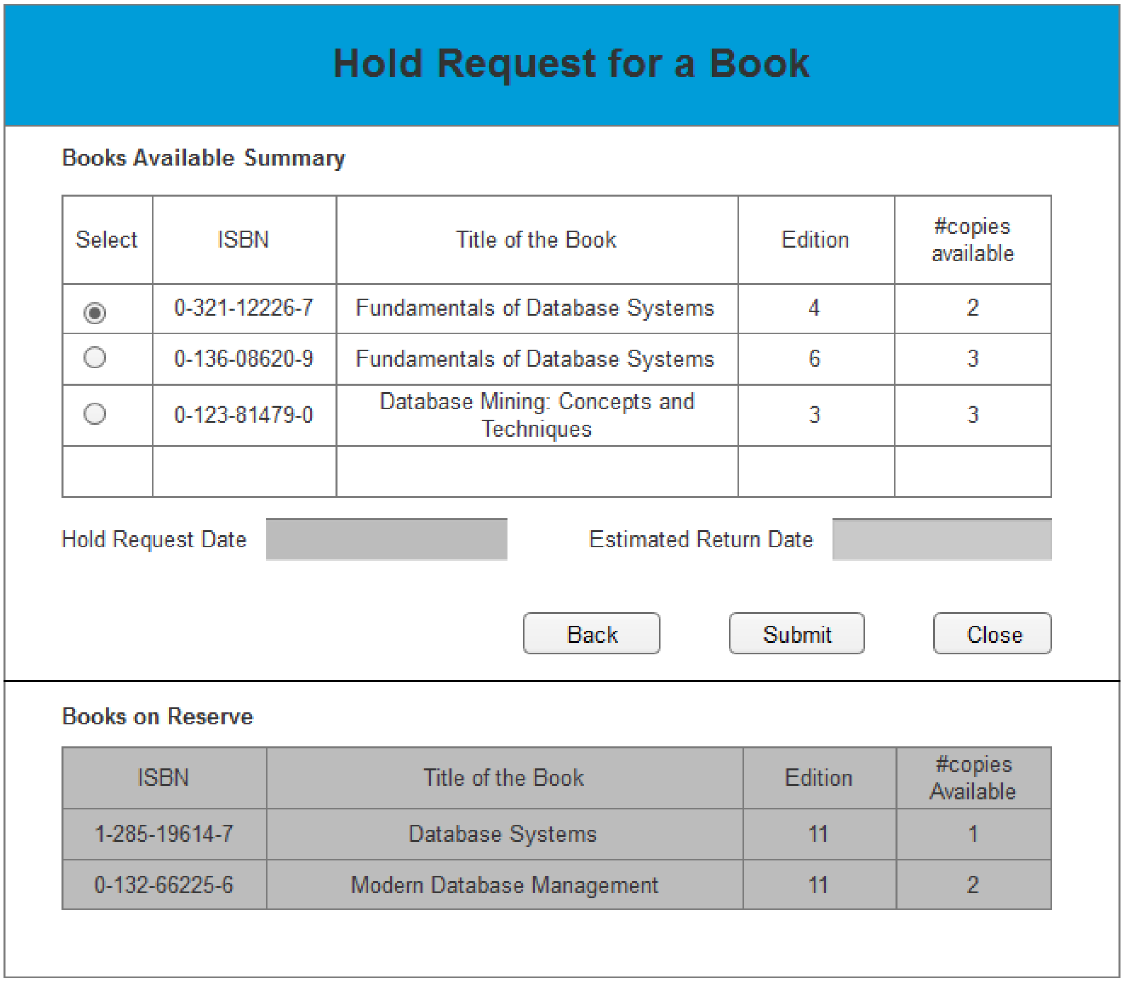
After looking for the desired book, the user would be shown the availability of the book in the next screen. If the book is not available, then the system maintains the earliest date when it becomes available. That date is based on the last check-out of that copy or the hold-status. Non-availability of the book implies that none of the copies of the book are available.
The 'book on reserve' field would be pre-populated from the database along with the number of the book copies available for a given ISBN. Reserved books are kept in a separate area in the library and are available for a day-checkout inside the library only. The user cannot request a hold on any book on reserve. For those books he should go to the library and request them in person. All copies of the same book (that means same ISBN) are either reserved or available for checkout.

5. Request hold on a book

Once the availability of the book is shown, the user, if the book is available for checkout, can request a hold on the book. When a user puts a book on hold, it is valid for three days during which the user must go and check the book out. (The checkout functionality is shown under library staff functionalities). Books on hold are brought to the desk area by the check-out staff for a speedy check-out. After three days the book is put back on the shelf and becomes available to other users if not picked up. In any case, when the book is requested for hold, the today’s system date would be taken as the hold request date and the system would automatically generate an estimated return date of 17 days later (which includes 3 days on hold and 14 more days for check-out). This is to give the user a buffer of maximum three days to pick his book from the library and a 14 day period to keep the book. When the user physically checks out the book from the library, this estimated return date is updated to exactly 14 days from the checkout date. For a particular book there can be only one hold request by one user. Hold requests on a book with multiple copies are applied to the copies in ascending order of copy number. Every time a book is placed on hold, the count of the number of copies available decreases by 1 in the database. Any attempts to make additional hold request will be rejected when all copies are on hold.

The system would also generate an ‘issue_id’ associated with a request which would be recorded and displayed to the user for editing his request at a later date if needed.

Fig 5. Shows a mockup screen.



Hold Request for a Book

Books Available Summary

Select	ISBN	Title of the Book	Edition	#copies available
●	0-321-12226-7	Fundamentals of Database Systems	4	2
○	0-136-08620-9	Fundamentals of Database Systems	6	3
○	0-123-81479-0	Database Mining: Concepts and Techniques	3	3

Hold Request Date

Estimated Return Date

Books on Reserve

ISBN	Title of the Book	Edition	#copies Available
1-285-19614-7	Database Systems	11	1
0-132-66225-6	Modern Database Management	11	2

Figure 5: Hold Request for a Book

6. Request extension for an issued Book

If a user wants to keep the book for a longer time period than assigned, he can request an extension (reissue) from the system as well. **He can only use this functionality on a book that he has physically checked out from the library.** For a student maximum 2 extensions are allowed per issue of a book. For faculty this number is 5. The estimated return date of the book updates to 14 days from the date of the extension request if that request is accepted. Note that a student can only keep a book for a maximum of 28 days from the original check-out date whereas a faculty can keep it for up to 56 days. . When the book is first checked out the checkout date and extension-request-date are same and the expected- return-date is 14 days from check-out. Thereafter, with every extension, the extension-request-date is updated, and the expected-return-date is updated. The original checkout date is always recorded to check that a user can keep a book for a maximum of 28 days. **Also, the extension of a book is only allowed if that book has not already been requested for future hold by another user.**

For e.g. If I have a book 'Fundamentals of Database Systems*,* Edition 4' issued to me from 2015-01-02 to 2015-01-16 and if I want to extend this duration, I need to go to the Extension Request form and reissue the book. If I do so on 2015-01-10 and if the book hasn't been requested by some other user then my return date would become 2015-01-24. And this would be counted as one extension, leaving me with 1 more extensions possible for this particular issue of the book.

Each user is allowed 2 extensions for each issue of a book. (For a new check-out of the same book by a user, 2 extensions will be granted again for each new issue.) This information about how many times a user rechecks out a book is recorded. And if the user is a faculty then he is allowed 5 such requests.

Below we show an example of how the dates would be updated with every extension requests made. (Just an example, this does not show all attributes involved in the transaction, such as ISBN, copy# etc.)-

Book Checked-out on January 1st.

Username	Issue_id	Checkout date	Request Extension date	Estimated Return Date
Malvika	101	2015-01-01	2015-01-01	2015-01-15

First extension request made on Jan 6th.

Username	Issue_id	Checkout date	Request Extension date	Estimated Return Date
Malvika	101	2015-01-01	2015-01-06	2015-01-20

Second extension request made on Jan 18th.

Username	Issue_id	Checkout date	Request Extension date	Estimated Return Date
Malvika	101	2015-01-01	2015-01-18	2015-01-29

Fig 6. Shows a mockup screen.

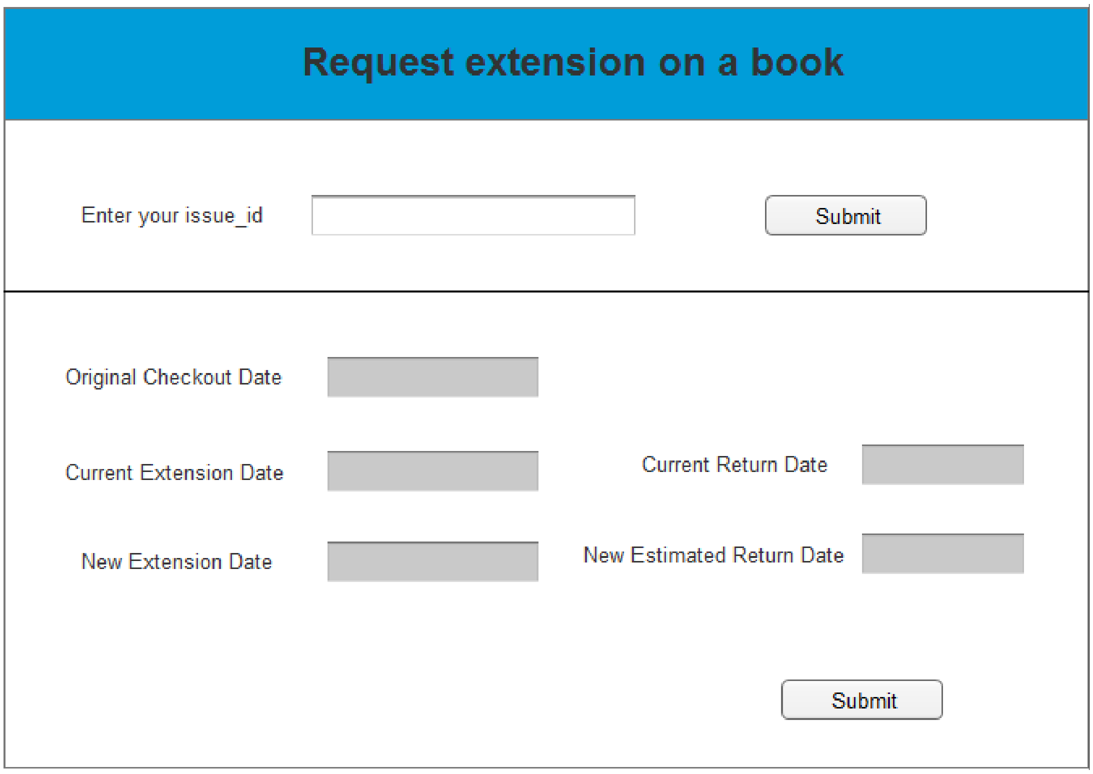


Figure 6: Request Extension on a Book

The user enters the issue_id assigned to him and then submits. The system looks for his record and auto-populates the current checkout date, current extension request date (which will be same as the checkout date if this is the first extension being requested) and current return date. (These would be non-editable fields). The new extension request would be the current system date (uneditable) when he is making this request and the new return date (uneditable) would be system generated date of 14 days later than new extension request date. Once the user submits, the system would check if the book has been requested on hold by someone or not; if not, the extension is accepted. If any book has a future hold then extension is denied for that book.

7. **Future Hold Request for a Book**

This feature allows a user to have his username recorded against a book as a future hold requester. The system automatically sends a message to the requester one day before the available date for the book he requested that the book is now available. (You are not responsible for this functionality of sending a message.) You just have to record in the database if a book has been requested for hold by any user.

Fig 7 shows a mockup screen.

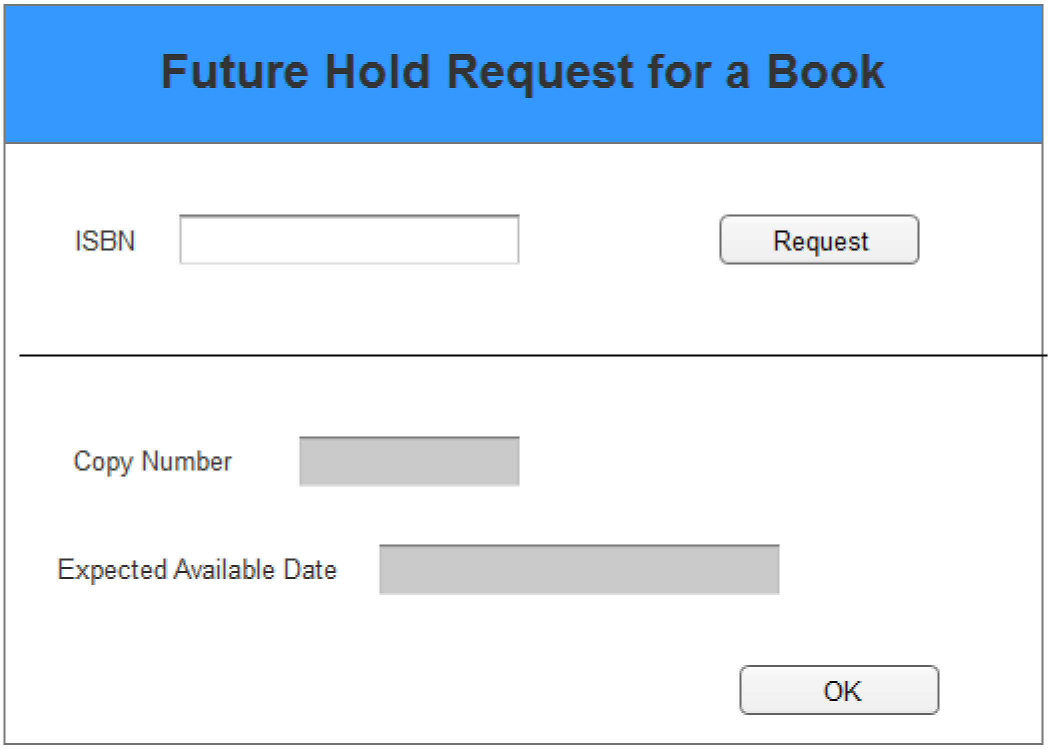


Figure 7: Future Request for a Book

When the user submits an ISBN, the system looks up the copy with the earliest expected available date, informs the user, and records his username as the future requestor against that copy in the database.

If a book is checked out or on hold, it has an expected return date which is taken as the expected available date; such a book can be requested for future and the system will accept the request and populate the "future requester" field for the book with the username of the requester.

8. **Track location**

This functionality helps one locate a book in the library. Each book is associated with a subject category and each book can belong to only one subject. Each subject has a name, number of journals, and keywords associated with it. The library has three floors. Each floor has a floor number, number of copiers, and a number of student assistants. Each floor can hold books on multiple subjects. All books of the same subject are on the same floor. And there can be multiple shelves on each floor. Each shelf has a shelf number, an aisle number and can hold multiple books.

Fig 8 shows a mockup screen. The grey boxes are the output boxes.

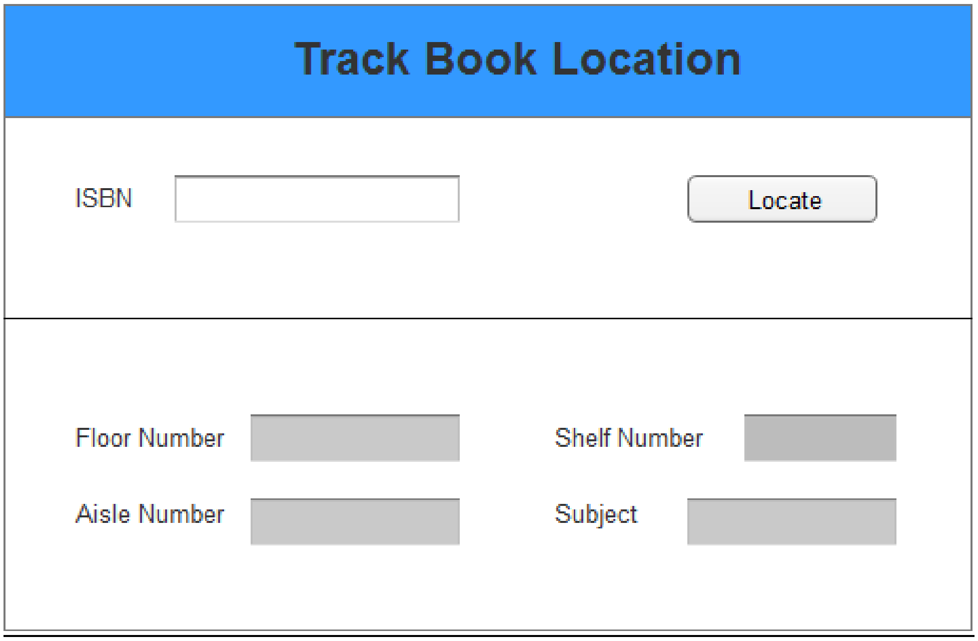


Figure 8: Track Book Location

We are assuming that all copies of the same book are placed together on the same shelf.

The Functions of Library Staff

There is a station where the user physically checks out a book that he requested to be put on hold or he just took out from the shelves. There is another return station where users return a book. Both stations are manned by library staff to make sure no user leaves the library without using the checkout screen, and to make sure when the user is returning a book, it is getting checked for any damage done. So there is always a record of when a user returns the book, and if he returns it in a damaged condition, or if the book is lost and not returned on time. (The loss of book may be informed by the user by some means. Lost books are so recorded in the database.)

9. Checkout Screen

After requesting a hold on the book, the user has to check out the book from the library. He can come to the library, go pick up the book and then go to the terminal where he'll checkout from this screen. The process will be facilitated by the staff. The user can enter the issue_id and the book will have the information about the ISBN and copy number which will be scanned and shown on the screen along with the user name. (Imagine that the user scans his card which automatically populates his username and name.) The date fields are also auto-populated from the database. If there was a hold request placed by that user on that copy, then it will be dropped from the system when the book is checked out against it.

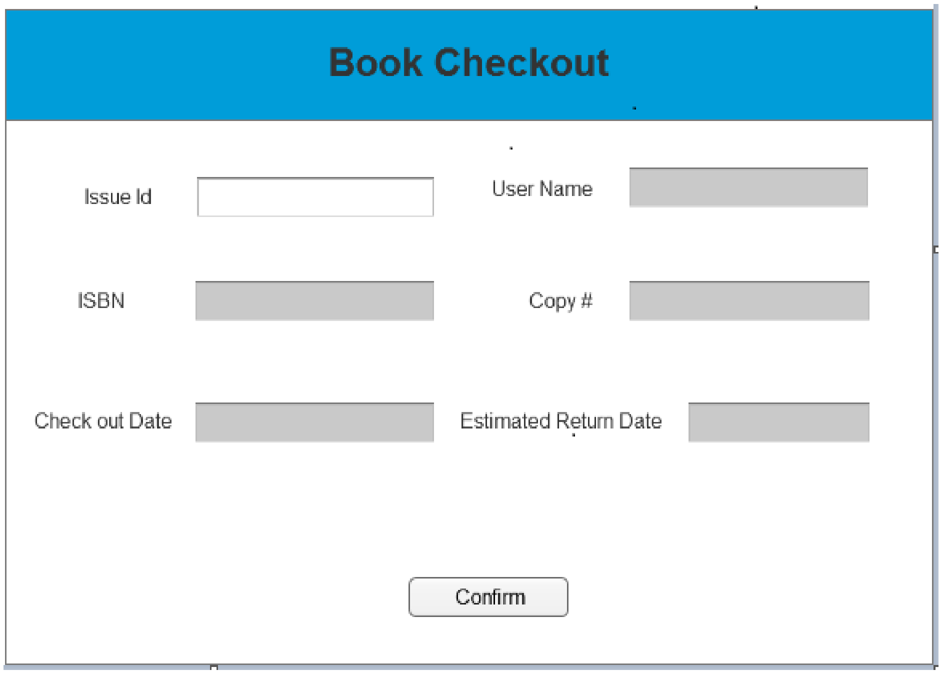
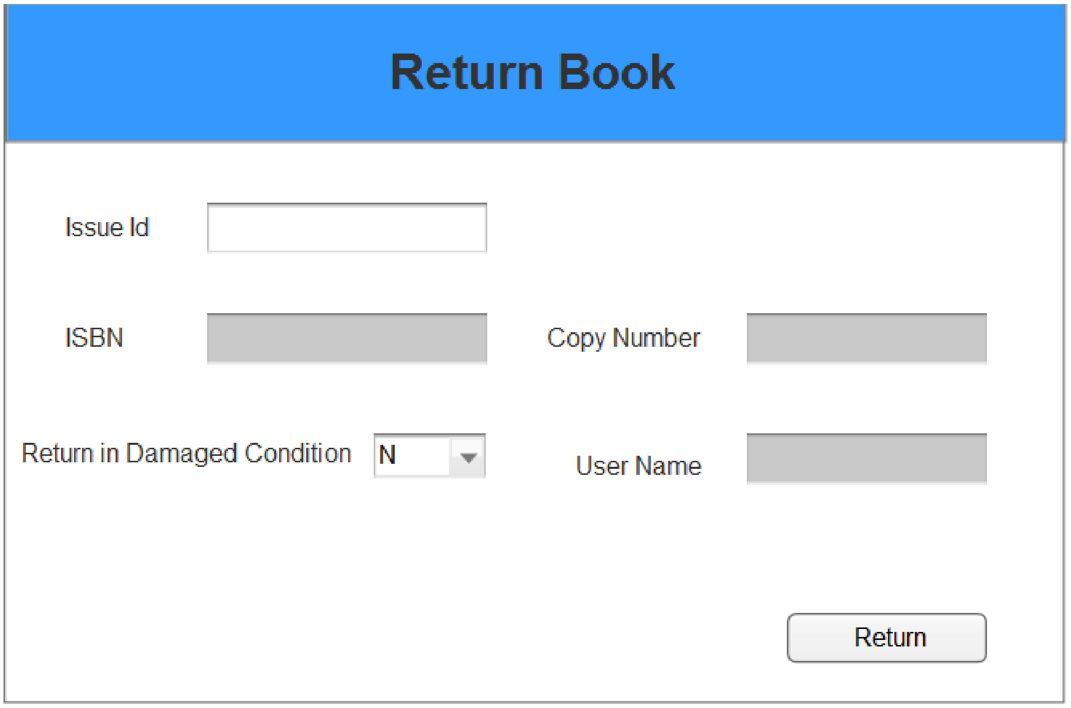


Figure 9: Checkout Screen

Remember that when this check out is done, the estimated return date for the issue is to be updated to 14 days from this checkout date or the last allowed date(based on the maximum number of days allowed to him and the maximum number of extensions allowed to him), and the isCheckedout flag has to be updated as well. Also, in case the user is coming to pick up his book after the admissible grace period of 3 days, the system should throw him an error saying that his hold has been dropped. If a copy of that book is still physically in the library, he can locate it, and still check it out. If no copy is physically available, he may place a "request for future" on a copy that has the lowest estimated availability date.

10.Return Book Screen

Similar to the checkout function, the terminal would have a return book function for the user. When the user comes in to return a book, the staff would check if the book is damaged or not. If it is damaged, the staff will go to the Penalty charges screen shown next to charge the user account. Otherwise, the staff accepts the book and system marks the isCheckedout flag as zero.



Return Book

Issue Id

ISBN

Copy Number

Return in Damaged Condition: N

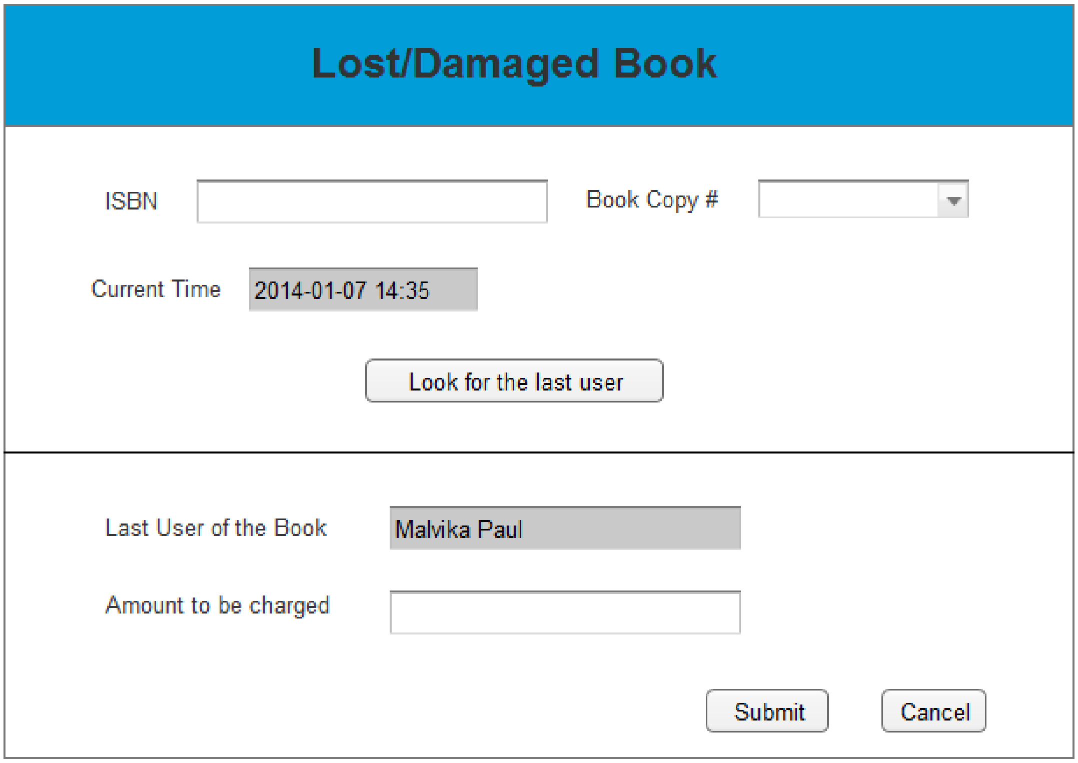
User Name

Figure 10: Return Book Screen

If the book is being returned after the due date, then the user account would be automatically charged with the late penalty. Penalty is assessed at $0.50 per day for each late day (regardless of days of the week). The penalty is recorded in a user billing account – users are expected to settle these charges in another financial system which is not covered here. (When 30% and 40% limits are reached and the book is still not returned, an automated message and reminder will be sent to the user). Users who reach a cumulative penalty of more than $100 (this covers the penalty for late returns, lost books, damaged books) are debarred (their Debarred Status is activated) and they are not allowed to check a book for the current semester. (The messaging and debarring functionality is outside our scope. Do not worry about it. You just have to record the correct cumulative penalty amount against a user.) It is assumed that once the book is back in the library, the staff members replace it at the designated location on the shelf.

11. **Penalty charges for lost/damaged book**

The staff would charge penalty to the user student account if the book is lost or damaged by the borrower. E.g., for a damaged book, the user would be charged a penalty equal to 50% of the price of the book. And if the book is reported as lost, then the user would be charged the cost of the book.
Fig 11 shows an example UI.



Current Time 2014-01-07 14:35

Last User of the Book Malvika Paul

Figure 11: Report Lost/Damaged Book

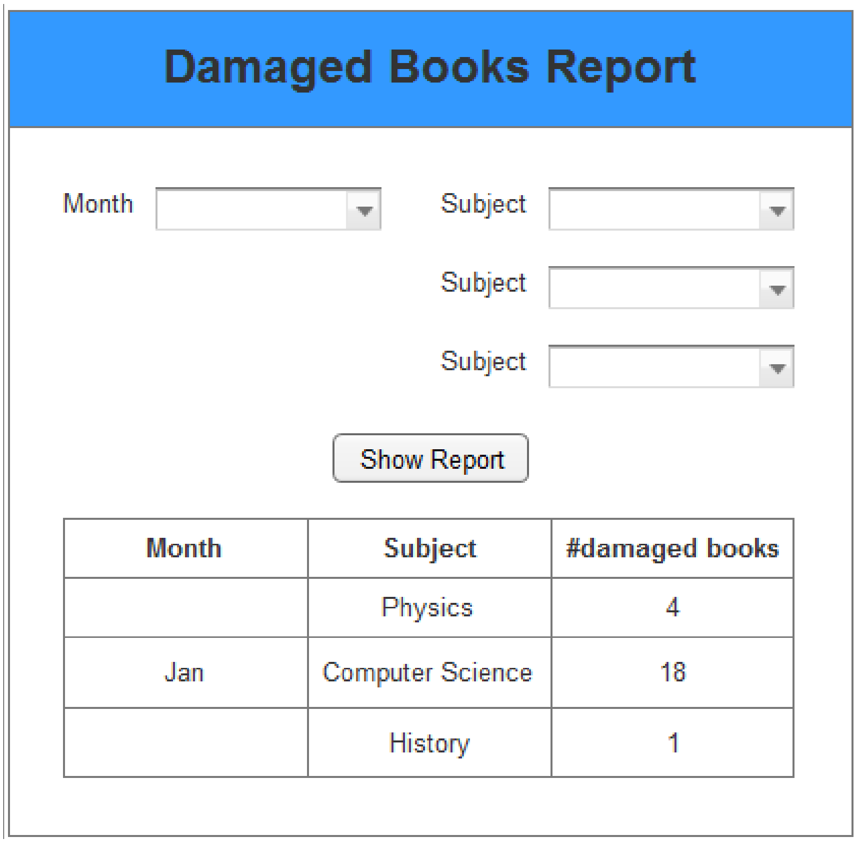
The system time would be taken as current time. Once the library staff clicks on ‘Look for the last user’, the application should take the book ISBN, copy# and the current time, based on which the result should return the last user of the book as shown in the bottom half of the screen. The staff would then manually enter the penalty charges on the user account. Once a book is marked damaged, it cannot be used for any future issues (is not available for borrowing anymore).

12.Reports

We will assume that the data you will populate will be for the months of Jan, Feb and March. Further instructions on loading proper data into tables will be given after Phase II is returned.

12.1. Damaged Books Report

This report shows the number of damaged books for three selected subjects for a given month.

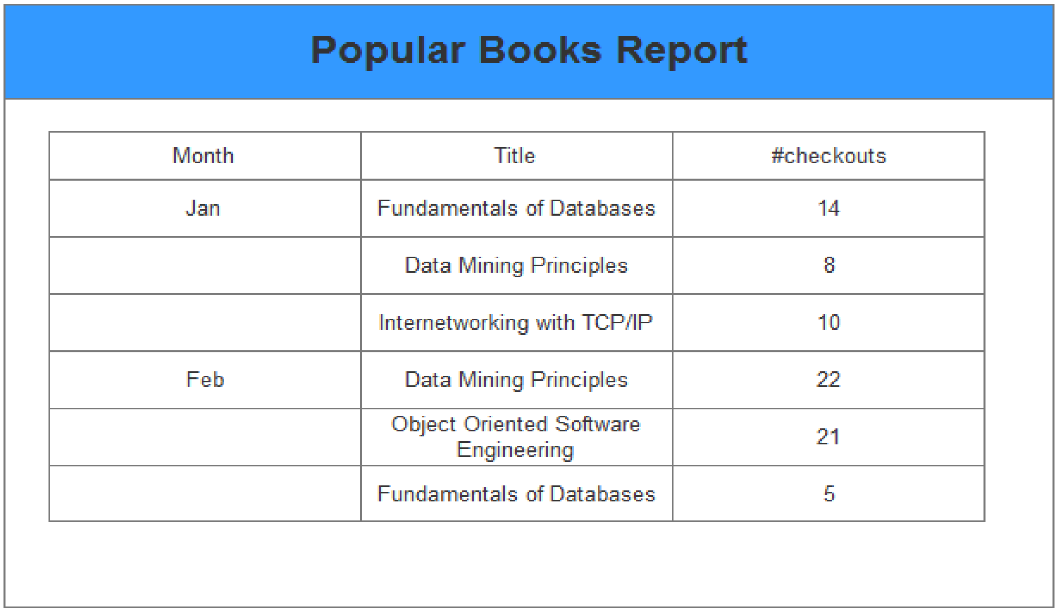


Month	Subject	#damaged books
	Physics	4
Jan	Computer Science	18
	History	1

Figure 12: Damaged Book Report

12.2. Popular books Report

This report shows the 3 most popular books issued by the users by the number of checkouts grouped by month. You need to show the report for the months of January and February.



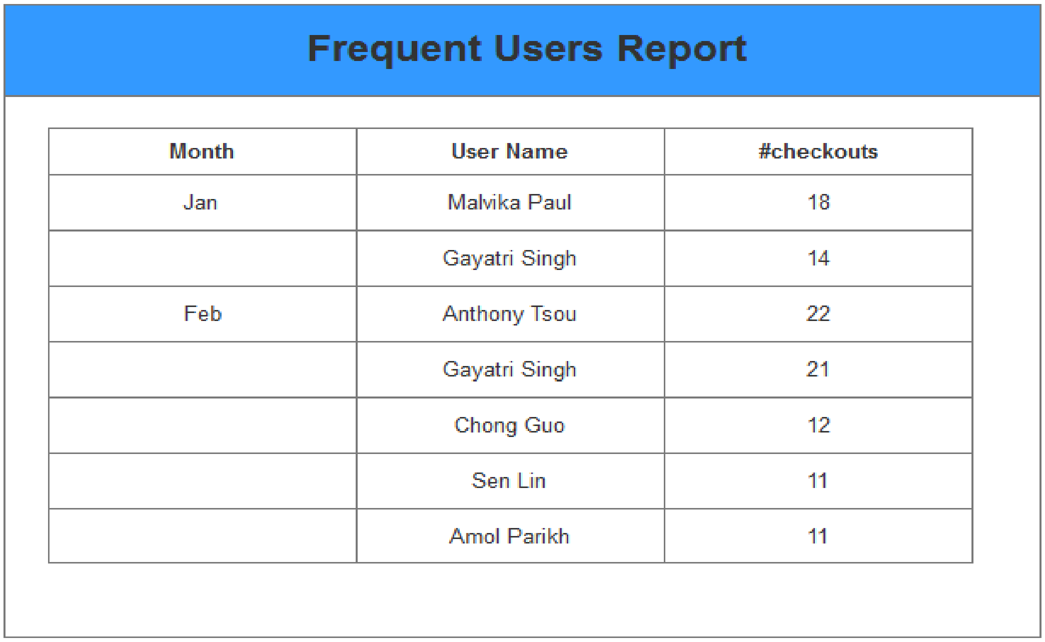
Popular Books Report

Month	Title	#checkouts
Jan	Fundamentals of Databases	14
	Data Mining Principles	8
	Internetworking with TCP/IP	10
Feb	Data Mining Principles	22
	Object Oriented Software Engineering	21
	Fundamentals of Databases	5

Figure 13: Popular Books Report

12.3. Frequent User Report

This report shows the up to 5 users (with at least more than 10 checkouts) by the number of checkouts grouped by month. You need to show the report for top 5 users (if there are more than 5) for the months of January and February.



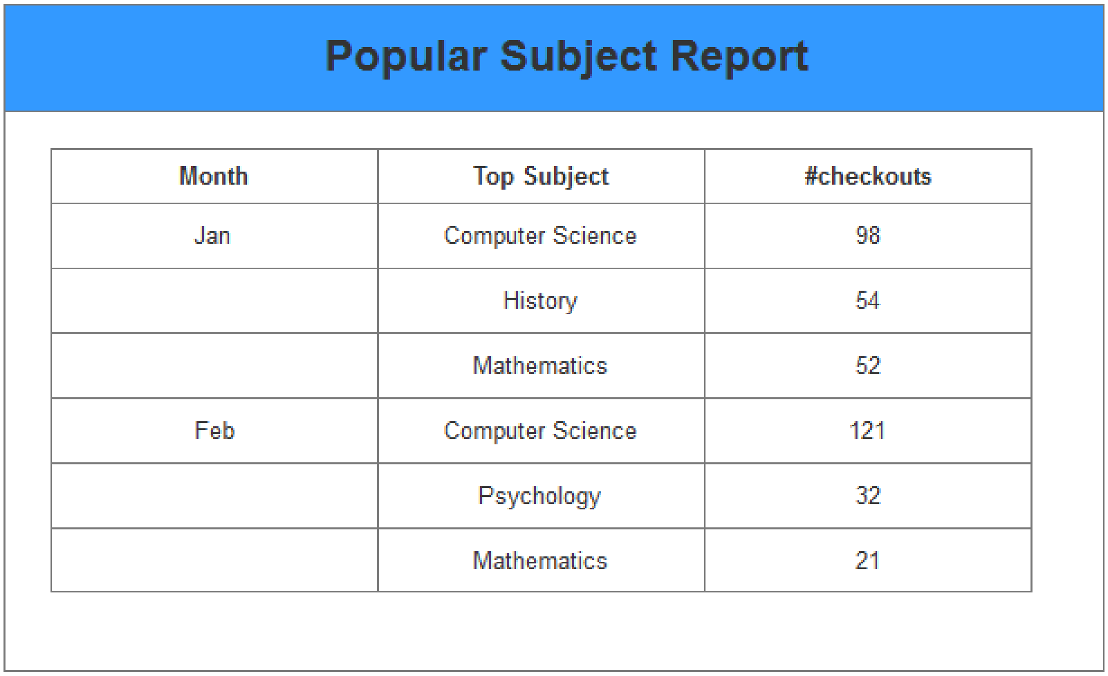
Frequent Users Report

Month	User Name	#checkouts
Jan	Malvika Paul	18
	Gayatri Singh	14
Feb	Anthony Tsou	22
	Gayatri Singh	21
	Chong Guo	12
	Sen Lin	11
	Amol Parikh	11

Figure 14: Frequent Users Report

12.4. Popular Subject Report

This report shows the most popular subject of the books issued by the users for January and February. For that subject, list the number of checkouts made that month.



Popular Subject Report

Month	Top Subject	#checkouts
Jan	Computer Science	98
	History	54
	Mathematics	52
Feb	Computer Science	121
	Psychology	32
	Mathematics	21

Figure 15: Popular subject Report

END OR LMS DESCRIPTION (V 1.0 January 17, 2015)

Version information:

Version Number	**Comments**
1.0	Original Draft
1.1	All books of the same subject are on the same floor.
1.2	Once a book is marked damaged, it cannot be used for any future issues (is not available for borrowing anymore). **But it remains in the database since you have to generate a report later for the same.**
1.3	4. Search Books - Here the user could search a book with its ISBN, Title or Author. Only one of these fields must be used to search for a book. **For students doing the Light –weight option: Write three separate queries for the three search cases (one for searching the book with ISBN, second with Title, third with Author)** **For students doing the Heavy-weight option: You must use the SQL query for searching, but you can take care of the if-else case scenario from the implementation point of view. (E.g. If ($title != null → run the sql query to search with *title*, else → …..)** 7. Future Hold Request - Language clarified. 9. Checkout Screen – Screen modified to work with respect to Issue_id.

	10. Return screen modified to work wrt issue_id. Maximum penalty clause removed. 12.4. Report to be generated for the months of Jan and Feb.