Phase 3

Conceptual Database Design

TEAM 3M: My Mongo Maria

<u>TEAM MEMBER:</u>

Eiji Iriyama

Kim Seong Kyu

Weidi Shen

Weitong Wang

Table of Contents

Section 1 Project Description

The project is to implement an online rental system for a library. The core attribute we would like to strive for is user-friendliness. We would like to pattern the features from app stores – using categories to narrow the search down and use lineups in each section to increase the vividness of the application. A general searching feature would of course also be supported. Users can search by key words, author, ISBN, publisher, or any combinations of the search categories. We would also like to implement a ranking/rating system for the popularity of the books available for renting. In general, we want this implementation to be more of a casual manner in presentation, rather than a serious academic setting.

The biggest motivation for this project is because of the common characteristic of online library systems lacking vividness. As the development of electronic readings growing rapidly, the paper-based book reading is showing a trend of decreasing in popularity. One of the major reasons that people in general are reading fewer books is that paper based reading generally gives a feeling of having to be serious, that is why we want to break this stereotype by trying to make the library's system more of a "fun" to use. Through observation and brainstorm amongst our team members, patterning the form of an app store is a practical way to achieve our goal. It is the simplest template that is both vivid and supports all the functionalities the system requires.

In this report, we are going to make some analysis for the requirements discussing what kind of functionality is supported and how to implement them. The requirement analysis is followed by the designs for entities, which will support the implementation. We would also have a deeper discussion on why the attributes are designed this way. Last but not least, a brief summary will be also included in the end of the report.

Section 2 Figures and Tables

2.1 Requirement Analysis:

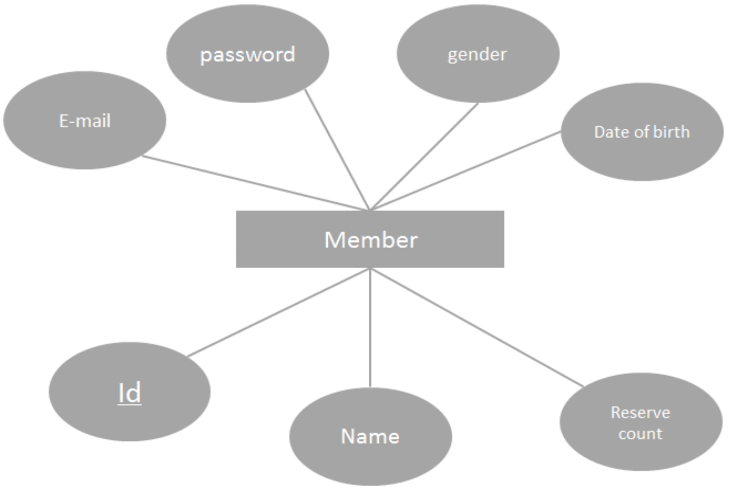
User Perspective:

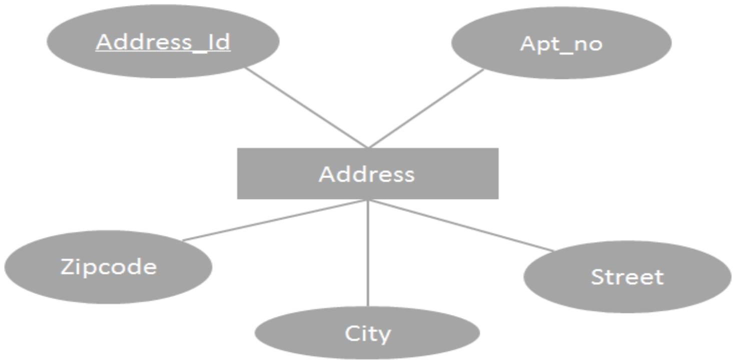
User	**Solution**
Can we easily check if a particular book can be rented?	Convenient web-based search system The quantity of books can be rented or reserved?
The book I want is always on for rent!	Reserve System Construction Number of people on the reservation list and schedule the arrival date of their reserved books.
I want to know the popular books	Top book listings according to most rented and user ratings We will also create the Weekly/Monthly pop list based on the current browsing/searching record.
I want brief summary of the book	Plot, URL to publisher's site. Also, you can jump into user's rating.
How easily can I g et the book?	You can pick it up at pickup station of library which hold the book by your reservation confirmation. Or user could select the delivery option to the user's address. However, you have to pay for the delivery fee.

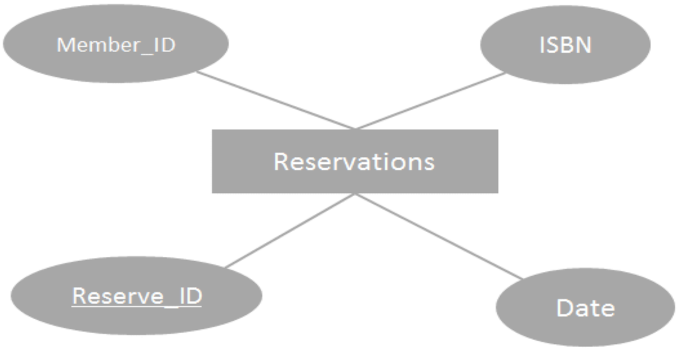
Admin Perspective:

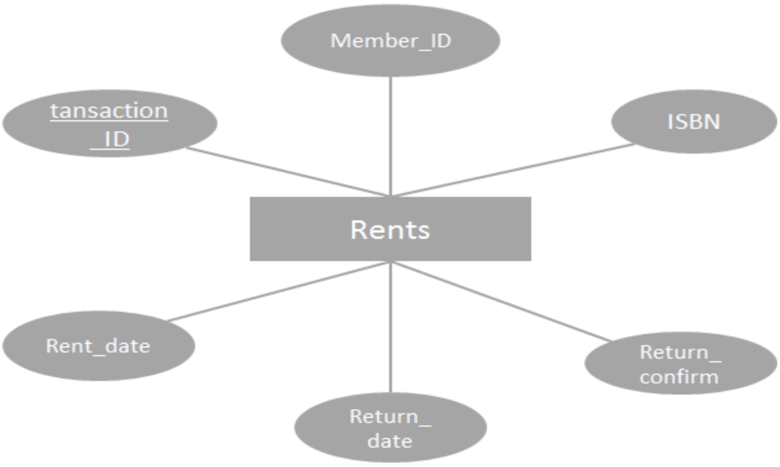
Admin	**Solution**
Can we provide easy rental service to users?	Convenient web-based search system
Can we easily keep on track on rented books?	Configure rental period based on Date
Can we easily manage user information?	Blacklists, Good users, Browsing history

2.2 Conceptual Design

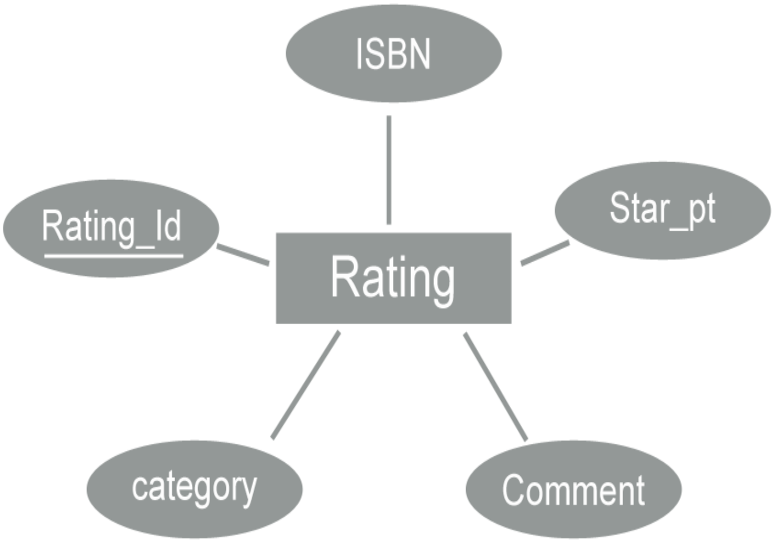
1. MEMBER ENTITY	
Function	Shows users information
ER Model	 password gender E-mail Date of birth Member Id Name Reserve count - Strong Entity Type - Each Attribute is simple attributes - Each Attribute is one valued attributes

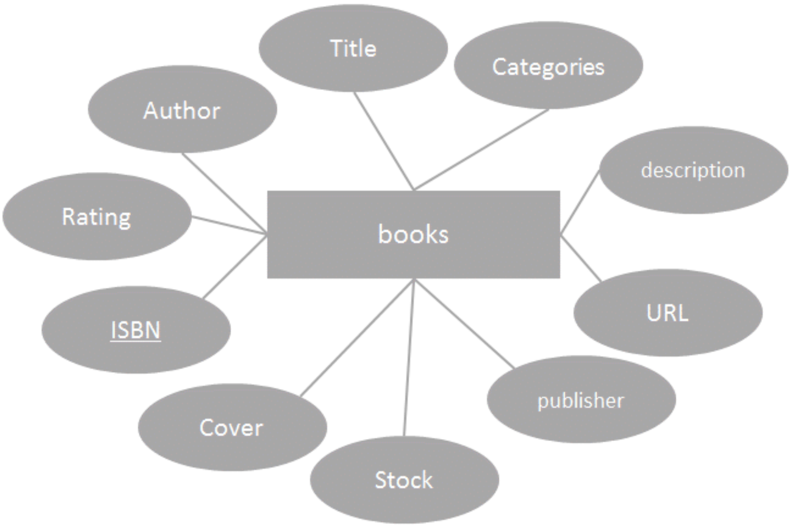
2. ADDRESS ENTITY	
Function	Contains members address
ER Model	 Address_Id Apt_no Address Zipcode Street City - Strong Entity Type - Each Attribute is simple attributes - Each Attribute is one valued attributes

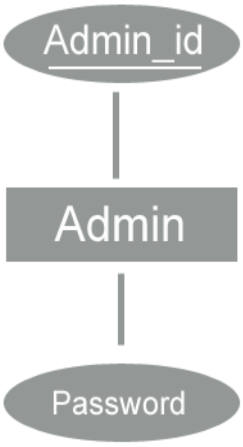
3. RESERVE ENTITY	
Function	Contains reservation information
ER Model	 - Strong Entity Type - Each Attribute is simple attributes - Each Attribute is one valued attributes

4. RENT ENTITY	
Function	Contains rental information
ER Model	 Member_ID tansaction _ID ISBN Rents Rent_date Return_ confirm Return_ date

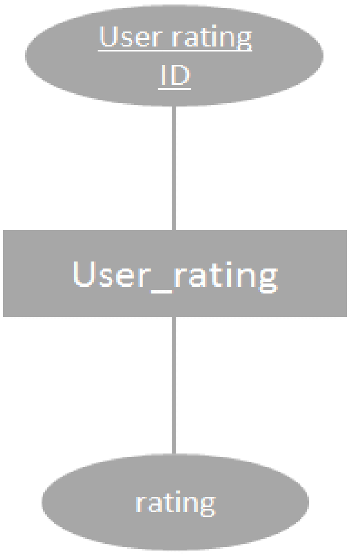
	- Strong Entity Type - Each Attribute is simple attributes - Each Attribute is one valued attributes

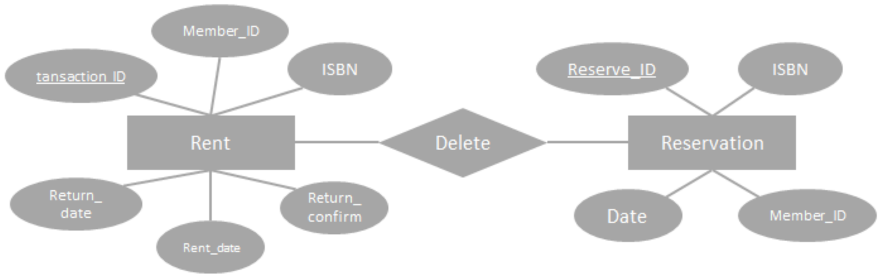
5. BOOK RATING ENTITY	
Function	Contains information about ratings for a book
ER Model	 ISBN Rating_Id Rating Star_pt category Comment - Strong Entity Type - Each Attribute is simple attributes - Each Attribute is one valued attributes

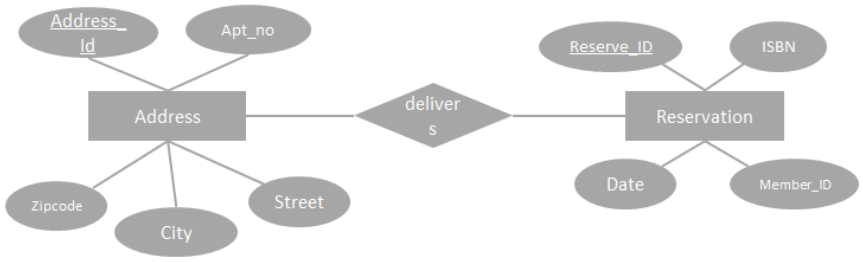
6. BOOK ENTITY	
Function	Contains all formation about the book
ER Model	 Title Categories Author description Rating books URL ISBN publisher Cover Stock - Strong Entity Type - Each Attribute is simple attributes - Each Attribute is one valued attributes

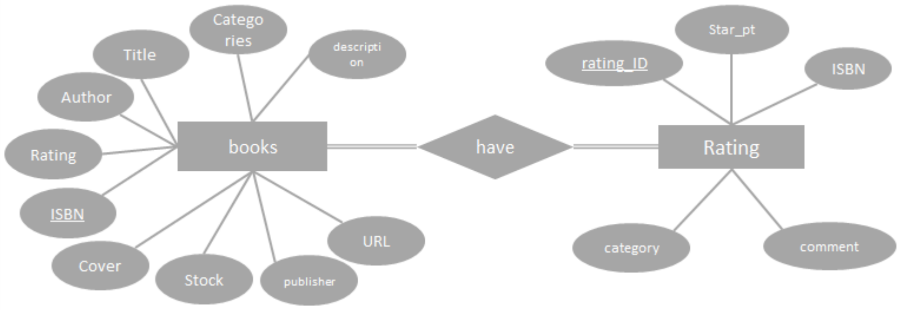
7. ADMIN ENTITY	
Function	The administrator information
ER Model	

	- Strong Entity Type - Each Attribute is simple attributes - Each Attribute is one valued attributes

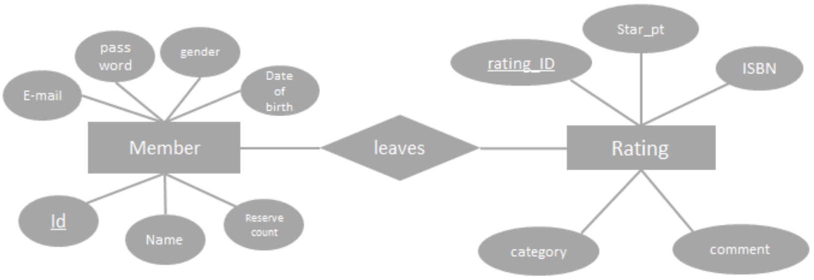
8. USER RATING ENTITY	
Function	Contains information for user’s rating
ER Model	 - Strong Entity Type - Each Attribute is simple attributes - Each Attribute is one valued attributes

9. DELETE RELATIONSHIP	
Function	Deletes user id from reservation list
ER Model	 Member_ID tansaction_ID ISBN Rent Delete Reservation Reserve_ID ISBN Return_ date Rent_date Return_ confirm Date Member_ID - Delete relationship - When reader succeeds on renting book, the user id is deleted from the reservation list.

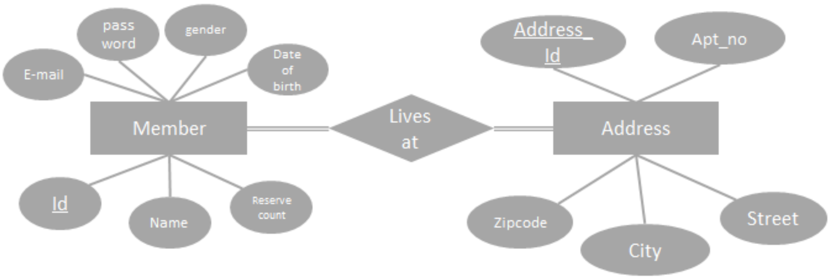
10. DELIVER RELATIONSHIP	
Function	Delivers book to the corresponding address
ER Model	 Address_ Id Apt_no Address deliver s Reservation Reserve_ID ISBN Zipcode City Street Date Member_ID - Deliver relationship - Each reservation has a corresponding address, which will deliver.

<table>
<tr><th colspan="2">11. HAVE RELATIONSHIP</th></tr>
<tr><td>Function</td><td>Possessing relationship</td></tr>
<tr><td>ER Model</td><td>


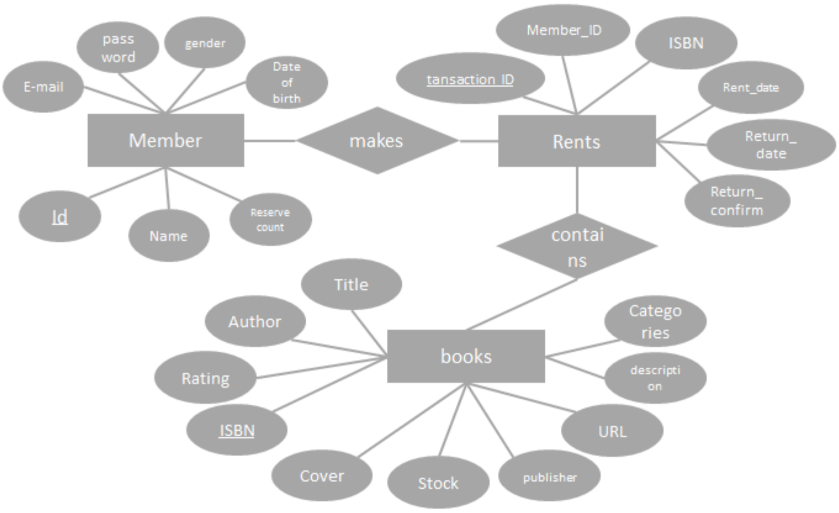
- Have relationship
- Ratings are related to rating.
</td></tr>
</table>

<table>
<tr><th colspan="2">12. LEAVE RELATIONSHIP</th></tr>
<tr><td>Function</td><td>For users to leave comment</td></tr>
<tr><td>ER Model</td><td>


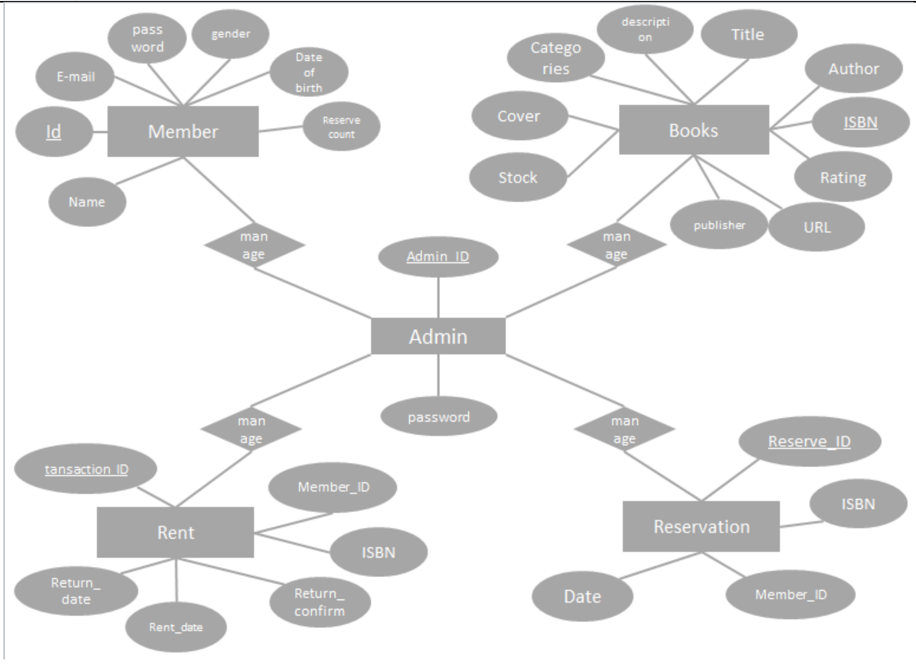
- Leave relationship
- User leaves comments and rates the book.
</td></tr>
</table>

<table>
<tr><th colspan="2">13. LIVES_AT RELATIONSHIP</th></tr>
<tr><td>Function</td><td>Relates addresses to members</td></tr>
<tr><td>ER Model</td><td>


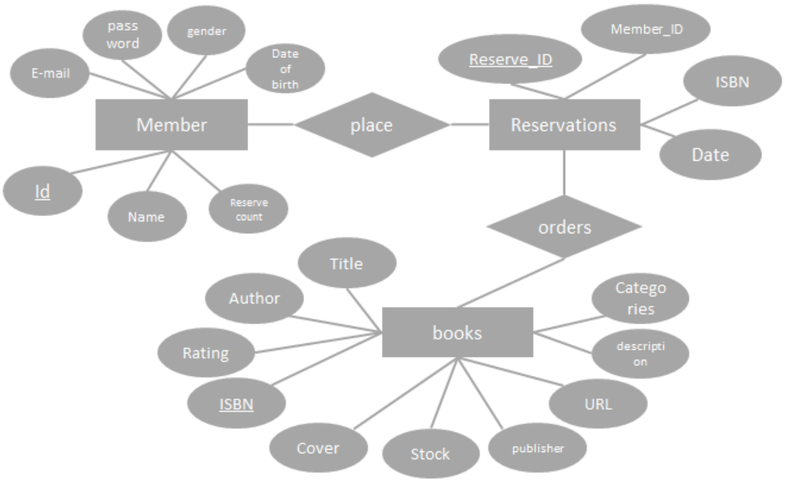
- Lives_at relationship
- A member lives at a corresponding address.
</td></tr>
</table>

<table>
<tr><th colspan="2">14. MAKES_CONTAINS RELATIONSHIP</th></tr>
<tr><td>Function</td><td>Member rents a book</td></tr>
<tr><td>ER Model</td><td>


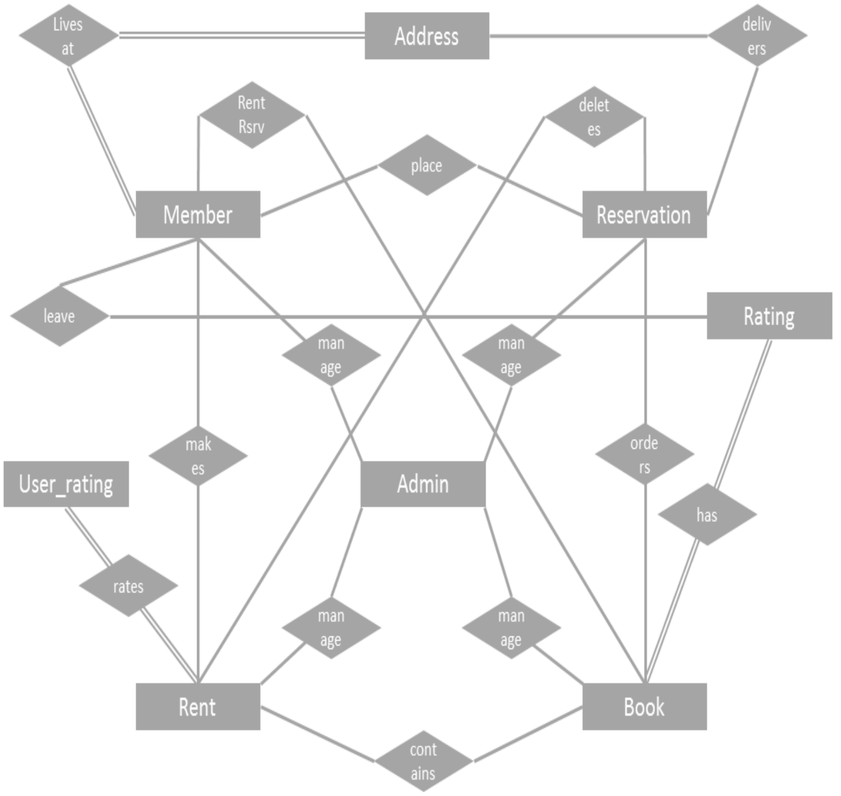
- Make_Contains relationship
- A member makes rents on the certain book
</td></tr>
</table>

<table>
<tr><th colspan="2">15. MANAGES RELATIONSHIP</th></tr>
<tr><td>Function</td><td>Admin manages the major entities</td></tr>
<tr><td>ER Model</td><td>


- Manages relationship
- Admin is the master who can manage and modify the major entities
</td></tr>
</table>

<table>
<tr><th colspan="2">16. PLACE_ORDER RELATIONSHIP</th></tr>
<tr><td>Function</td><td>Member directly orders a book</td></tr>
<tr><td>ER Model</td><td>


- Place_order relationship
- If there is no member on reservation list, direct order is placed.
</td></tr>
</table>

17. OVERALL ER DIAGRAM	
Function	Overall ER diagram for our project
ER Model	 Lives at Address deliv ers Rent Rsrv delet es place Member Reservation leave Rating man age man age mak es orde rs User_rating Admin has rates man age man age Rent Book cont ains - Admin will manage each Entity - Rating Entity has a total participation with Book Entity - User_rating Entity has a total participation with Rent Entity

2.3 Feature design Details:

1) Browsing & searching are combined because we create the category tree and list of items (which is identifier in our case) as the attribute of our Book Entity.

2) We set all the entity as strong type, because we don’t have any entities which really depends on others. For our case, we used to consider the rating entity as a weak type. In common sense, as the certain book is no longer in our rental list, the comments for this book might be gone as well. However, by our search about relative website, we found that we could still read about the comments for a book even if the book is no more available to be rented. At last, we decide to keep rating entity as a strong type. Just in case, if users need those unavailable somewhere else, but they can also learn something about the book from others’ comments.

3) In order to gain fully control of our system, we also create an Admin Entity to manage other major entities, such as Member Entities, Book Entities, Reserve Entities, and Rent Entities.

4) In the ER diagram design, we create a Blacklist Entity for our extend feature. The Admin Entity will manager this entity depending on the rating of users. Once the user returns a damaged book. The Admin Entity will drop this user from Member Entity into the Blacklist Entity. Since we will apply this approach in the actual implementation, we will not include any blacklist-related ER diagram in our report.

Section 3 Associated Attribute

3.1 Baseline features:

3.1.1 Books - The focus here is the books for rental. A book is rented by users via web page. The sources of the book is the library. A unique identifier is assigned to a book when it is in available to be rented. This is used to identify the book. A short description (no more than 500 characters) is associated with a book (provided by the library). In addition, a URL may be provided to link to a more detailed description at publisher's website.

3.1.2 Categories - The books available at the library are categorized using a predefined classification tree. Each node represents a set of items. The root of the tree is labeled 'All' and represents all items. Each, with a descriptive name, represents some subset of the items represented by the parent. An item can be specified by a path through this classification tree. For example, we may categorize a book as:

1. Language > Novels > Science Fictions > **Hunger-Game**

3.1.3 Suppliers - The library supplies all the books and information.

3.1.4 Searching & Browsing - Users are able to search the books by entering some keywords or conditions. As a search result, a list of items that satisfy the search criteria is returned to the user. We will keep the browsing history of the users, and check on the current status of the certain book. When a page is loaded, the status of corresponding books will be updated. Based on the record of those update, the system will evaluate the current weekly or monthly popular book list.

3.1.5 Registered Users - To rent a book, a user must be registered. A registered user is identified by a user name and authenticated with a password. In addition, the maintained information includes, e-mail address, name, age, address (which consists of street, city, state, and zip), phone number, and etc. A user will be given a unique member number. Users are also rated. If the rating is extremely low, the user will be marked in our blacklist system.

3.1.6 Rating - We attempt to suggest most popular books by readers' ratings. A user can rate the book on how much he/she liked it with some little comment (Max 100 words).These ratings and a short explanation (less than 300 characters) are then made available to other users.

3.1.7 User_Rating – Since we determine the user blacklist by our Rating system, we decide to do a different rating system for users. As user are detected to have some problem with the returned books. We will be marked into blacklist.

3.1.8 Transaction History - We will keep track on the transaction history for users. The certain date, and book ID will be store in a table. Users can only access their own transaction history. The library will have access to every user's transaction history, and check on the current status of a certain book (i.e. Book A is currently rented by someone, Book B has been returned by user S.)

3.1.9 Reservation (bidding) - A user could reserve at most 5 different books at the same time. Each book can be put at most 10 reservations. The user in blacklist cannot place any reservations on any books.

3.1.10 Rent (Sale) - The maximum rental date for each users is a week for each book. By default the maximum rental date is week. However, if necessary, the user could extend their rental date up to two weeks. This option is only for user with the good user mark.

3.1.11 Delivery - When users placed a rental order or reservation order, the library will check the users' profile, and deliver the selected books to the address provided by the users. (Our library only delivers books to users in United State)

3.1.12 Reports to Telemarketers - We will monthly report to telemarketers that how many books are rented by a person, the person's information, such as name, address, e-mail, phone, age, and gender.

3.2 Extended features:

3.2.1 Blacklist - We will attached a blacklist to our registered users. It contains a rating ranks for corresponding user, and comments for why this user is added to the blacklist (Max 100 words). These ratings and a short explanation (less than 300 characters) are then made available to other users.

Section 4 Relational Schema in SQL

All the create table SQL statements for entities are listed below. From the final ER diagram, we developed the relational schema in SQL for all the entities and relationships. There are 8 entities and 15 relationships in our model. For the create table statements for entity, they are straight-forward conversion from the ER model we have shown above.

create table member (member_id int, gender varchar(10), date_of_birth date, reserve_count int, name varchar(30), password varchar(20), email varchar(30), primary key(member_id));
create table address(street_no int, apt_no int, zipcode int, state varchar(20), city varchar(30), member_id int, primary key(member_id));
create table reservations(reserve_id int, member_id int, isbn varchar(30), reserve_date date, primary key(reserve_id));
create table rent(member_id int, transaction_no int, rent_date date, return_date date, isbn verchar(30), return_confirm boolean, primary key(member_id));
create table rating(isbn varchar(30), isbn varvhar(30), rating_id int, comment varchar(1500), star int, primary key(isbn), unique(rating_id));
create table books(isbn varchar(30), stock int, avg_rating int, title varchar(200), auther(200), description varchar(1500), cover blob, publisher varchar(100), url varchar(100), category varchar(20), primary key(isbn));
create table admin(admin_id int, password varchar(30), primary key(admin_id));
create table user_rating(user_rating_id int, rating int, primary key(user_rating_id));

All the create table SQL statements for relationships are listed below. For the create table statements for relationships that we converted from our ER model, we have a cascade option in the rates table which deletes the tuple in user_rating table when the member_id in the rent table is deleted. When we tried to create tables in MySQL, we found that there is a constraint in MySQL that we cannot define more than 1 primary key in a table, so in the implementation, we will adjust our design based on MySQL constraints.

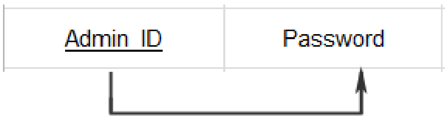
create table lives_at(member_id int, primary key(member_id) foreign key(member_id) references member, foreign key(member_id) references address);
create table delivers(member_id int, reserve_id int, primary key(member_id, reserve_id), foreign key(member_id) references member, foreign key(reserve_id) references reservations);
create table rent_reserve(member_id int, isbn varchar(30), primary key(member_id, isbn), foreign key(member_id) references member, foreign key(isbn) references books);
create table deletes(member_id int, isbn varchar(30), primary key(member_id, isbn), foreign key(member_id) references rent, foreign key(isbn) references books);
create table place(member_id int, reserve_id int, primary key(member_id, reserve_id), foreign key(member_id) references member, foreign key(reserve_id) references reservation);
create table leave(member_id int, isbn varchar(30), primary key(member_id, isbn), foreign key(member_id) references member, foreign key(isbn) references rating);
create table makes(member_id int, primary key(member_id), foreign key(member_id) references member, foreign key(member_id) references rent);

create table orders(reserve_id int, isbn varchar(30), primary key(reserve_id), foreign key(reserve_id) references reservations, foreign key(isbn) references books);
create table has(isbn varchar(30), primary key(isbn), foreign key(isbn) references rating, foreign key(reserve_id) references books);
create table rates(user_rating_id int, member_id int, primary key(user_rating_id, member_id), foreign key(member_id) references rent, foreign key(user_rating_id) references user_rating, on delete cascade);
create table contains(member_id int, isbn varchar(30), primary key(member_id, isbn), foreign key(member_id) references rent, foreign key(isbn) references books);
create table manage1(member_id int, admin_id int, primary key(member_id, admin_id), foreign key(member_id) references member, foreign key(admin_id) references admin);
create table manage2(reserve_id int, admin_id int, primary key(reserve_id, admin_id), foreign key(reserve_id) references reservations, foreign key(admin_id) references admin);
create table manage3(isbn varchar(30), admin_id int, primary key(isbn, admin_id), foreign key(isbn) references books, foreign key(admin_id) references admin);
create table manage4(member_id int, admin_id int, primary key(member_id, admin_id), foreign key(member_id) references rent, foreign key(admin_id) references admin_id);

Section 5 Functional Dependency

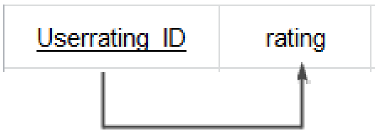
5.1 Functional Dependency

Admin



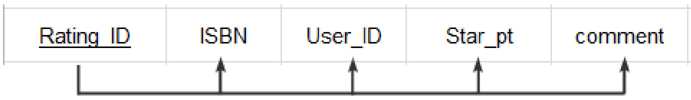
Admin_ID	Password

User Rating



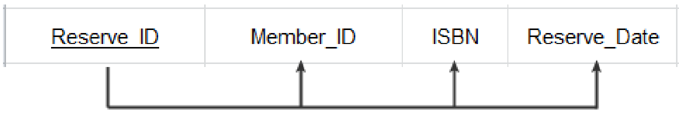
Userrating_ID	rating

Rating



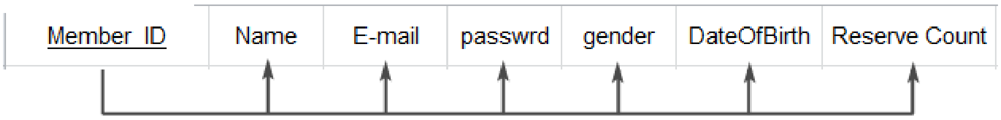
Rating_ID	ISBN	User_ID	Star_pt	comment

Reservation



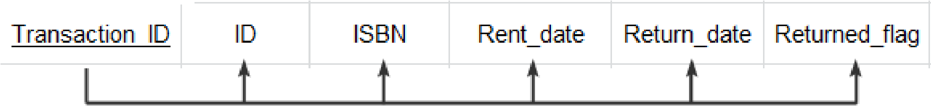
Reserve_ID	Member_ID	ISBN	Reserve_Date

Member



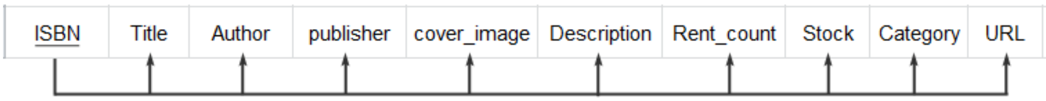
Member_ID	Name	E-mail	passwrd	gender	DateOfBirth	Reserve Count

Rent



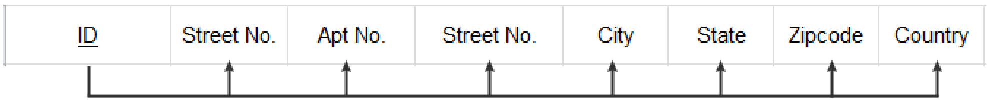
Transaction_ID	ID	ISBN	Rent_date	Return_date	Returned_flag

Book



ISBN	Title	Author	publisher	cover_image	Description	Rent_count	Stock	Category	URL

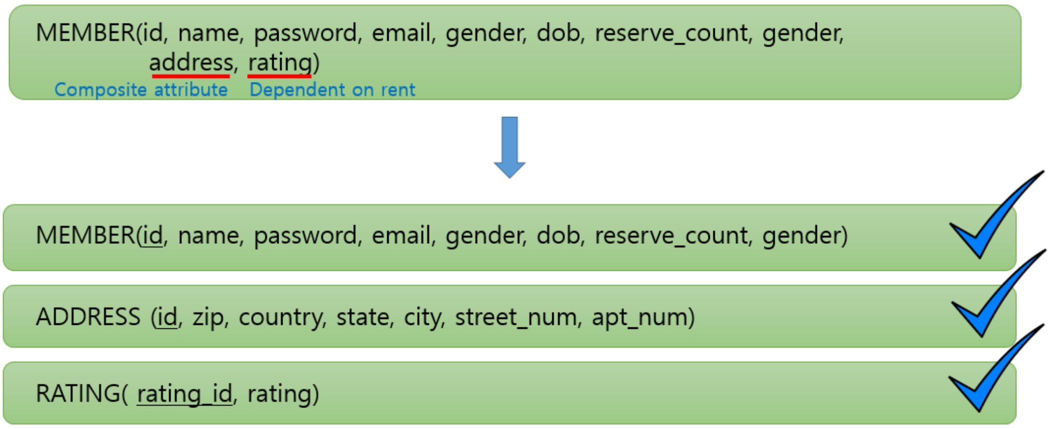
Address



ID	Street No.	Apt No.	Street No.	City	State	Zipcode	Country

5.2 Normalization

5.2.1 Normalization 1



MEMBER(id, name, password, email, gender, dob, reserve_count, gender, address, rating)

Composite attribute

Dependent on rent

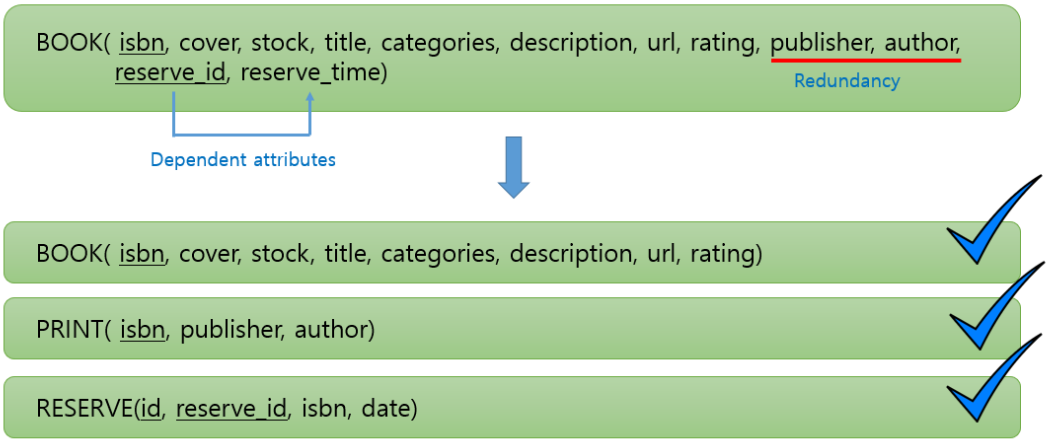
MEMBER(id, name, password, email, gender, dob, reserve_count, gender) ✓

ADDRESS (id, zip, country, state, city, street_num, apt_num) ✓

RATING(rating_id, rating) ✓

Address is a composite attribute that contains various attributes, so we decided to make it entity.
Rating is mainly dependent on the Rent entity in our design, so we decided to make it an entity and relate to rent.

5.2.2 Normalization 2


BOOK(isbn, cover, stock, title, categories, description, url, rating, publisher, author, reserve_id, reserve_time)

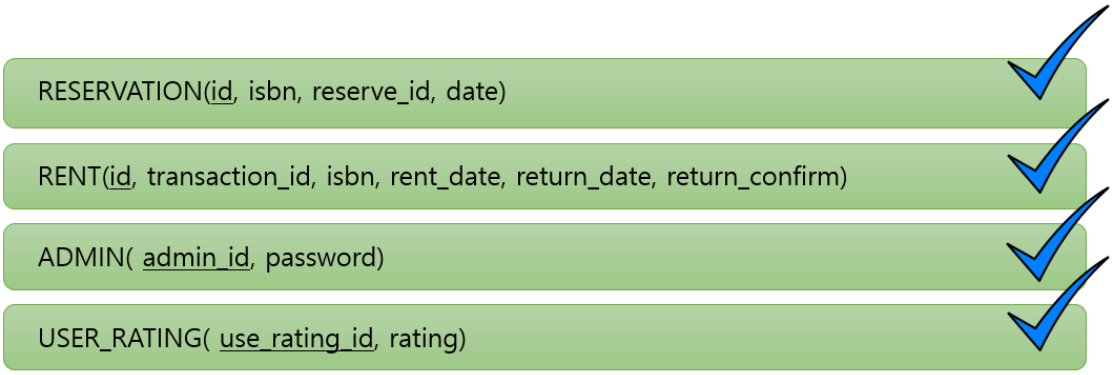
BOOK(isbn, cover, stock, title, categories, description, url, rating) ✓

PRINT(isbn, publisher, author) ✓

RESERVE(id, reserve_id, isbn, date) ✓

Publishers, and author will be redundant in many cases, such as serious of books so we decided to move it out to reduce redundancy.
Reserve_time is only dependent on **reserve_id**, therefore we made an separate entity for it.

5.2.3 Normalization 3


RESERVATION(id, isbn, reserve_id, date) ✓

RENT(id, transaction_id, isbn, rent_date, return_date, return_confirm) ✓

ADMIN(admin_id, password) ✓

USER_RATING(use_rating_id, rating) ✓

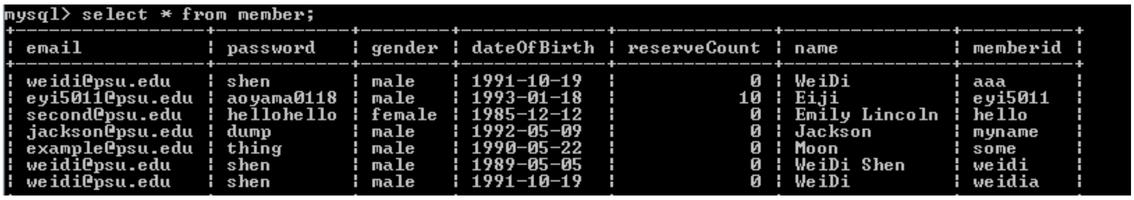
Other entities are legitimately normalized.

By normalization we reduced redundancy, and made dependency more clear which helped to support our relational design,

Section 6 Implementation

6.1 Database

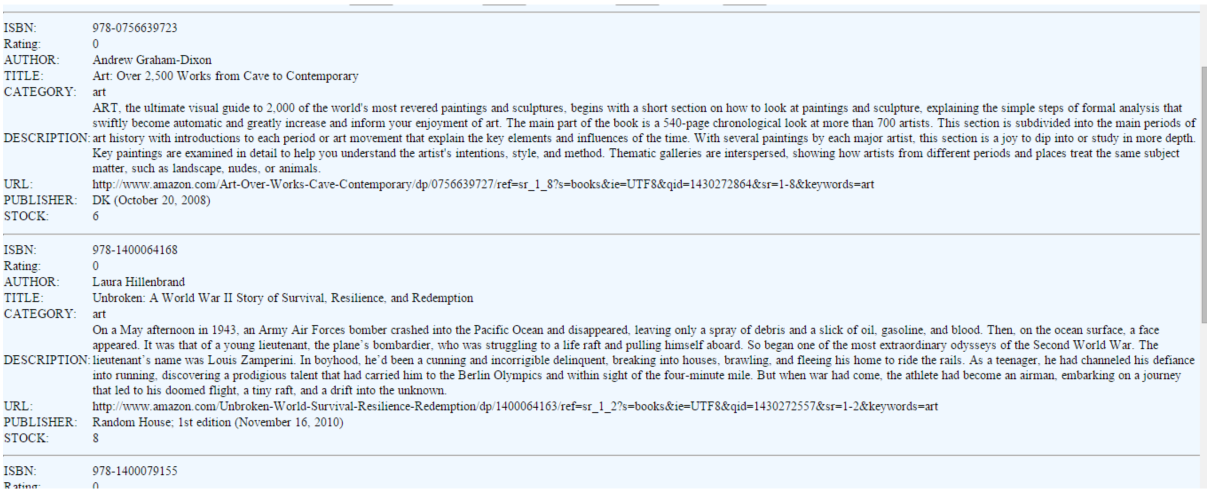
6.1.1 Users Aspect - The primary key for member is the id, we will use id to make connection to other tables which are associated to member entity. For example, we use our user registration feature to create 15 different accounts for 15 different user. Those information will be stored in the member database. During the registration process, users are required to fill out their personal information. For implementation need, we asked for their salary for the telemarking report. However, this is optional. Once the registration is done, each user are associated with the reserve database, rating database, rent history database and a blacklist database. We will use comments of users, which is the rating system in our case to determine if the user should be in the blacklist database. Once the user is marked in the blacklist database, he/she cannot do any reservations any more. When they login, we will also show a warning message on the webpage. One user can give comments and rating on a scale of 5 to another user. There is no limit on leaving the comments. Those comments are stored in the rating database.



```
mysql> select * from member;
+-------------------+------------+--------+-------------+--------------+----------------+----------+
| email             | password   | gender | dateOfBirth | reserveCount | name           | memberid |
+-------------------+------------+--------+-------------+--------------+----------------+----------+
| weidi@psu.edu     | shen       | male   | 1991-10-19  |            0 | WeiDi          | aaa      |
| eyi5011@psu.edu   | aoyama0118 | male   | 1993-01-18  |           10 | Eiji           | eyi5011  |
| second@psu.edu    | hellohello | female | 1985-12-12  |            0 | Emily Lincoln  | hello    |
| jackson@psu.edu   | dump       | male   | 1992-05-09  |            0 | Jackson        | myname   |
| example@psu.edu   | thing      | male   | 1990-05-22  |            0 | Moon           | some     |
| weidi@psu.edu     | shen       | male   | 1989-05-05  |            0 | WeiDi Shen     | weidi    |
| weidi@psu.edu     | shen       | male   | 1991-10-19  |            0 | WeiDi          | weidia   |
```

Figures 6.1.1: Partial snapshot for member table from MySQL

6.1.2 Items Aspect - The primary key for the book entity is ISBN, we will use it to make connections to those tables which are associated to book entity. In the project, user can search their desired book in different ways. They can search books by the book name, ISBN, book type, and Description. Once they find the book, they can place a reservation on this book. For our auction part, we did a little bit different. The priority is based on "first come first served." The admin will handle the reservation in the reserve database. Based on the address information in the member database, admin will deliver the books to the corresponding user. Each book is also associated with a book rating database. Users can leave comments on those book which is similar to the user rating. There is no blacklist for books.



ISBN:	978-0756639723
Rating:	0
AUTHOR:	Andrew Graham-Dixon
TITLE:	Art: Over 2,500 Works from Cave to Contemporary
CATEGORY:	art
DESCRIPTION:	ART, the ultimate visual guide to 2,000 of the world's most revered paintings and sculptures, begins with a short section on how to look at paintings and sculpture, explaining the simple steps of formal analysis that swiftly become automatic and greatly increase and inform your enjoyment of art. The main part of the book is a 540-page chronological look at more than 700 artists. This section is subdivided into the main periods of art history with introductions to each period or art movement that explain the key elements and influences of the time. With several paintings by each major artist, this section is a joy to dip into or study in more depth. Key paintings are examined in detail to help you understand the artist's intentions, style, and method. Thematic galleries are interspersed, showing how artists from different periods and places treat the same subject matter, such as landscape, nudes, or animals.
URL:	http://www.amazon.com/Art-Over-Works-Cave-Contemporary/dp/0756639727/ref=sr_1_8?s=books&ie=UTF8&qid=1430272864&sr=1-8&keywords=art
PUBLISHER:	DK (October 20, 2008)
STOCK:	6
ISBN:	978-1400064168
Rating:	0
AUTHOR:	Laura Hillenbrand
TITLE:	Unbroken: A World War II Story of Survival, Resilience, and Redemption
CATEGORY:	art
DESCRIPTION:	On a May afternoon in 1943, an Army Air Forces bomber crashed into the Pacific Ocean and disappeared, leaving only a spray of debris and a slick of oil, gasoline, and blood. Then, on the ocean surface, a face appeared. It was that of a young lieutenant, the plane's bombardier, who was struggling to a life raft and pulling himself aboard. So began one of the most extraordinary odysseys of the Second World War. The lieutenant's name was Louis Zamperini. In boyhood, he'd been a cunning and incorrigible delinquent, breaking into houses, brawling, and fleeing his home to ride the rails. As a teenager, he had channeled his defiance into running, discovering a prodigious talent that had carried him to the Berlin Olympics and within sight of the four-minute mile. But when war had come, the athlete had become an airman, embarking on a journey that led to his doomed flight, a tiny raft, and a drift into the unknown.
URL:	http://www.amazon.com/Unbroken-World-Survival-Resilience-Redemption/dp/1400064163/ref=sr_1_2?s=books&ie=UTF8&qid=1430272557&sr=1-2&keywords=art
PUBLISHER:	Random House; 1st edition (November 16, 2010)
STOCK:	8
ISBN:	978-1400079155
Rating:	0

Figure 6.1.2 Partial snapshot for book table from the website.

6.1.3 Browsing Tree – For the browsing tree, not including the root, we have 3 levels, and 37 categories in total.

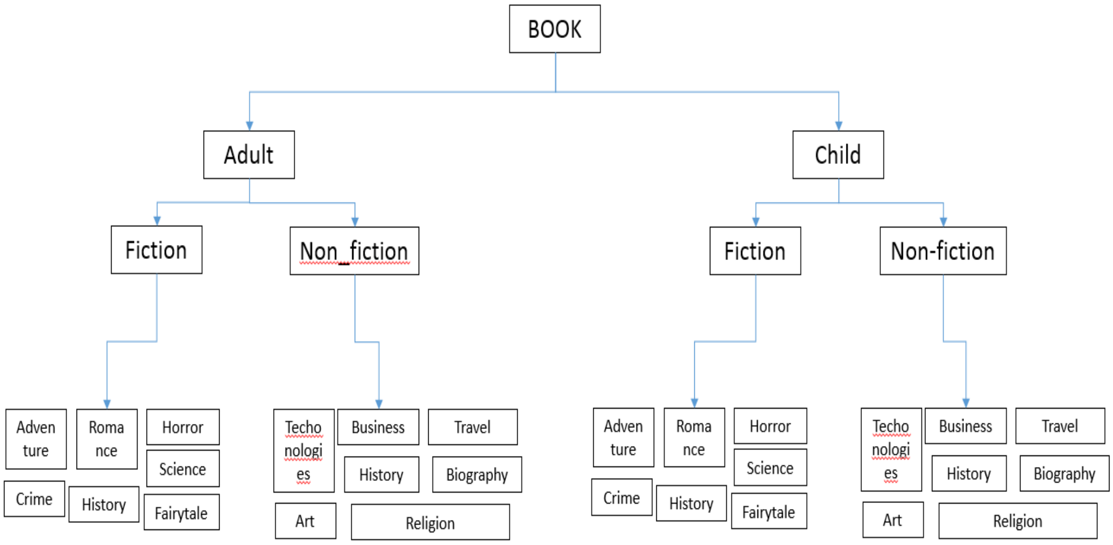


Figure 6.1.3 Diagrams for browsing tree

6.2 Transaction

6.2.1 AddUser – When a new user try to register an account, she/he will be asked to fill out some necessary information, such as gender, date of birth, userid, password, and so on. Once the registration step is done, we will store all these information into the member table.

6.2.2 SearchingItems - In our system, we offer four different ways for searching. 1) You can enter the book's name to find the book. 2) You can enter the ISBN to find the book. 3) You can search certain type of the books. 4) You can enter some key words to help you identify your desired book. No matter which you use, the information you provide will help retrieve the certain relevant book from the books tables.

6.2.3 BrowsingItems – The browsing step can be done in our library search engine. If you don't input any words. You can click submit to show all the books stored in the database. You can also select certain type to show all the relevant books from the databse.

6.2.4 BuyItem (Place Reservation) – Our project is only an online book rental system. User can place reservation on books instead of buying them. Each user can reserve up to 10 books as maximum.

6.2.5 AuctionItem (Reservation) – Since we don't have sell the book, we don't really have auction system in our project. The priority of a book reservation will based on "first come first served."

6.2.6 BidItem – Since we don't have the auction system, we don't have bid item either. We will have certain amount of books in our library. Once a certain books have been rented out, the next user who want to rent this book will have to wait until someone return the book.

6.2.7 TerminateAuction (Delivered Books) – In our project, we will deal with terminate auction in a different way. Once admin have finished the delivery step, the trade record will be in processing status. When user returned the book, admin will closed the trade case and keep the record information in the member's reservation history. This certain information of a book rental case will only be visible to the corresponding user.

6.2.8 TeleMarketingReport – For every 2 weeks, a report including the name, address, e-mail, phone number, age, gender, and annual income (optional) of users, and the total number of rental activities will be sent to a telemarketing firm.

6.3 Issues Discussion

During our implementation, we definitely have encountered a few noteworthy issues, and we think we successfully resolved them.

First of all, we noticed that some schema should be changed for some tables. For example, we needed a way to interchange between reservation, delivery, and renting. We added new user types to give different interfaces to resolve this problem. It was a rather simple example of changing our schema however it implies that we'll need to change the queries and source codes accordingly and that was a tedious job to do. This taught us a good lesson on the importance of accurately analysis the requirements.

We also encountered a problem when we need to monitor logon activities, we'll need to know which user is doing the actions and we didn't have a particular table/place to store the ids that are logged in, so we created a text file as a temporary storage.

Another problem that we encountered is on the css styling file. We are using different headers for ordinary users and admin pages; however the syntax and functions didn't seem to agree with this design. We experimented a lot with different formats and actually resolved this problem by changing the name that was actually in conflict with html keyword – navbar. It was simple, to be honest, but it took us a long time to figure out.

Sample inputs were also a big barrier in our implementation. It is extremely hard to find sample data for books that perfectly matches our schema, we had to enter everything by hand, and it was a tedious job to do.

Also in general, we were all new to php and css and everything, we all had busy schedules, but we managed to come up with this implementation. We would say that completing it alone is a success, even though we didn’t make it perfect and look as good as it possibly can be.

This project taught us collaboration, the theme of lifelong learning

Section 7 Summary

With a similar idea of Himalaya.com, we tried to design a book rental system for a library. Before we start it, we consider this project for two perspective: the user perspective and the admin perspective. We try to provide enough features to fulfill the requirement of our target potential user. At the same time, we also made the system free so almost everyone could be able to find what they in our system. In our design concept, we consider the Member Entity, Address Entity, Reserve Entity, Rent Entity, Rating Entity, Book Entity and Admin Entity as major component, and construct relationship for each entities in order to build our DBMS. During the actual brainstorming, we tried to relate our case to the real world problem. Then, we decided to have our major entity as strong type. With further discussion, we determine many necessary associated attributes for our entities, which most of them mapping to the baseline features, and the blacklist feature mapping to the extended feature.

Our unique features which distinguish our project to against other competitions is the rating system. Based on the rating, we will determine a blacklist feature. The purpose is that we want to increase the communications among our users. We will be glad to see that people using our system to discuss what books they like and what books they don’t like. However, our administrators will also gain control over these comments.

In our process of implementation, we implemented a new user type to perform the transition f or a reserve relationship to a rent relationship. As we noticed that a delivery must be done by a delivery person, and after the book is being delivered, the reservation is thus changed to an actual renting relationship. So we decide to give the delivery person the authority and respons ibility to check that the book is delivered. Similarly we created a type of user for Tele market viewer, so that they have restricted viewing to only the data they require.

This basically concludes all our ideas of design and implementation, a few changes in details were made along the way of our implementation as discussed in our implementation section. There are still definitely space for improvement such as better look and maybe a more direct way of viewing. However, due to the time frame of the project and the group’s limited techni cal ability, we were only able to finish the basic functionalities for the system. Nevertheless, a s a prototype, we believe it is a sufficiently completed system.

Section 8 Appendices

Appendix A Design Approach

In our design, we have an Admin as the system manager that gains the control over all entities. We expand our list of the entities based on the real world cases and we also did researches online about similar web applications such as Himalaya.com to identify our approach. Since we are trying to implement the book online rental system, we want to have similar features as Himalaya.com. We want to do deliver, online book browsing or searching, create account for users, and ratings for both users and the books. In our research, we found that there is a perfect rating system for those existing application. So we decided to enhance the rating system by creating the blacklist.

In our project, we will focus on the admin entity first. Because we want this entity have fully control over our application. For example, the admin entity could manage the rating for users. Once the user returned a damaged book, the administrator marks a low rating for that particular renting transaction (which points to the user), and will eventually cause that user being put into blacklist.

That is mainly the reason we split rent and reservation into separate entities, which seems intuitively should be relationships. We want the admin to be able to have relationships with these entities.

The other action we did to refine our schema is to separate the user rating entity. We find that every renting transaction should result in a rating, as mentioned above. We also made it in full participation constraint to fulfil the requirement

Appendix B Project Plan

Schedule:

Deadlines	Tasks
Week1	
Week2	Team assigned
Week3	Brainstorming the topic
Week4	Brainstorming the details and design of the project
Week5	Define project details and tools
Week6	Complete conceptual design
Week7	Implement database(1)
Week8	Implement database(2)
Week9	Spring break
Week10	Implement website(1)
Week11	Implement website(2)
Week12	Integrate the website with the database(1)
Week13	Integrate the website with the database(2)
Week14	Additional implementation
Week15	Testing
Week16	Final review

Appendix C Progress Report

Progress Report (5) for CMPSC431W

Summary

We are coming to the final stage of the project and the course. We have learnt the basics about php and successfully applying it and built up a system from scratch. During the process we modified a few schemas and clarified a few specifications regarding to the project amongst our group. We also worked on the details of implementation, even though we had a few issues, we successfully overcame.

Individual Activities

Eiji Iriyama

- Assignments – Coding the system, Help with Front end: 4/6/2015 – 4/29/2015 (30 hr)
- Background – more PHP tutorials and details for implementation (30 hr)
- Meetings - Group meeting: 4/15-4/29 (30hr)
- Design – Application logic
- Implementation – Coding the system and debugging
- Testing – Testing with random samples and debugging

Jackson(Weitong) Wang

- Assignments – Sample input, Coding the system, Project Report, Presentation slides: 4/6/2015 – 4/29/2015 (28 hr)
- Background – PHP tutorials (20 hr)
- Meetings - Group meeting: 4/15-4/29 (30hr)
- Design – Application logic
- Implementation – Coding the system
- Testing - none

WeiDi Shen

- Assignments – Sample inputs, Progress report, Project report, Database design and application design: 4/6/2015 – 4/29/2015
- Background – PHP tutorials (20 hr)
- Meetings - Group meeting: 4/15-4/29 (30hr)
- Design – Application logic
- Implementation - Coding the system
- Testing – finding bugs

Seong Kyu Kim

- Assignments – Front end, Power Point, Database design and application design: 4/6/2015 – 4/29/2015

- Background – PHP tutorials, css tutorials (20 hr)
- Meetings - Group meeting: 4/15-4/29 (30hr)
- Design – Front end design
- Implementation – css file
- Testing – Front end debugging

Group Activities

- Group meeting: 4/15-4/29 (30hr)
- Brainstorming and discussion about application logic, bugs, etc
- Debugging, testing.
- Suggestion of improvement of current implementation

Future Task

- Presentation and if time allows, further implement to make it look better.

Progress Report (4) for CMPSC431W

Summary

Based on the phase 2 report, we tried to create all the tables for our entities and relations into SQL on MySQL. After we had finished all the code design and implementation, we randomly take some inputs to check if our code is working as we expected.

Individual Activities

Eiji Iriyama

- Assignments - Writing partial SQL command and by using MySQL and writing progress report: 4/4/2015 – 4/5/2015 (8hr)
- Background - Studied basic commands for MySQL(3hr)
- Meetings - Group meeting: 4/4/2015 – 4/5/2015 (8hr)
- Design – Create tables for relations
- Implementation – Implement SQL through MySQL
- Testing – Take input to check the SQL code accuracy

Jackson(Weitong) Wang

- Assignments - Writing partial SQL command by using MySQL and writing progress report: 4/4/2015 – 4/5/2015 (8hr)
- Background - Student basic MySQL knowledge(3hr)
- Meetings - Group meeting: 4/4/2015 – 4/5/2015 (8hr)
- Design –Create tables for relations
- Implementation - Implement SQL through MySQL
- Testing - Take input to check the SQL code accuracy

WeiDi Shen

- Assignments –Writing partial SQL command and by using MySQL and writing progress report: 4/4/2015 – 4/5/2015 (8hr)
- Background - Studied web-based programming (3hr)
- Meetings - Group meeting: 4/4/2015 – 4/5/2015 (8hr)
- Design - Create tables for entities
- Implementation - Implement SQL through MySQL
- Testing - Take input to check the SQL code accuracy

Seong Kyu Kim

- Assignments - Writing partial SQL command and by using MySQL and writing progress report: 4/4/2015 – 4/5/2015 (8hr)
- Background - Web development hands on: Node.js, Ajax, JQuery

- Meetings - Group meeting: 4/4/2015 – 4/5/2015 (8hr)
- Design - Create tables for entities
- Implementation - Implement SQL through MySQL
- Testing - Take input to check the SQL code accuracy

Group Activities

- Group meeting 4/4/2015 – 4/5/2015 (8hr)
- Brainstorming and discussion about the project
- Figure out SQL command for each entities and relations through MySQL.
- Testing code for accuracy with inputs.
- Writing progress report

Future Task

As we finish all the basic code and testing requirement for the actual implementation. We will try our best to develop the actual system through web.

Progress Report (3) for CMPSC431W

Summary

For our phase 2 report, we first fix some issues that we had from phase 1 report. As we kept keeping on our missing items. We re-design our whole overall ER diagram. Once we finished those basic works, we wrote the relation schema based on our ER diagram. Then, we figure out the all functional dependencies for all the entities and relationship. At last, we convert these functional dependencies into third normal form.

Individual Activities

Eiji Iriyama

- Assignments - Writing relation schema by using mysql: 2/27/2015 – 3/5/2015 (8hr)
- Background - studied basic commands for MySQL(3hr)
- Meetings - Group meeting: 2/27/2015 – 3/5/2015 (26hr)
- Design –Relation schema
- Implementation – Implement relation schema through mysql
- Testing – testing relation schema through mysql

Jackson(Weitong) Wang

- Assignments - Writing Project Report, Design ER diagram: 2/27/2015 – 3/5/2015 (8hr)
- Background - Student basic MySQL knowledge(3hr)
- Meetings - Group meeting: 2/27/2015 – 3/5/2015 (26hr)
- Design – Normalization, ER diagram
- Implementation - none
- Testing - none

WeiDi Shen

- Assignments – Design ER diagram, Functional Dependency, and figure out normalization: 2/27/2015 – 3/5/2015 (8hr)
- Background - Studied web-based programming (3hr)
- Meetings - Group meeting: 2/27/2015 – 3/5/2015 (26hr)
-
- Design – Normalization, Functional Dependency, ER diagram
- Implementation - none
- Testing -none

Seong Kyu Kim

- Assignments - Writing Project Report, Power Point: 2/27/2015 – 3/5/2015 (8hr)
- Background - Web development hands on: Node.js, Ajax, JQuery
- Meetings - 2/27/2015 – 3/5/2015 (26hr)
- Design – Normalization
- Implementation - none
- Testing – none

Group Activities

- Group meeting(26hr) 2/27/2015 – 3/5/2015
- Brainstorming and discussion about the project
- Figure out relation schema, functional dependency, and normalization.
- Writing project report, progress report, and power point for phase 2 presentation

Future Task

- As we finish all the basic requirement for the actual implementation. We will try our best to develop the actual system through web.

Progress Report(2) for CMPSC431W

Summary

We decided to change our topic of the project from Pokemon website to an online book rental system for a library. The core attribute we would like to strive for is user-friendliness. We would like to pattern the features from app stores – using categories to narrow the search down and use lineups in each section to increase the vividness of the application. A general searching feature would of course also be supported. Users can search by key words, author, ISBN, publisher, or any combinations of the search categories. Also, we have developed several diagrams for ER-model to our project.

Individual Activities

Eiji Iriyama

- Assignments - Writing Progress Report: 2/8/2015(1hr)
- Background - studied basic commands for MySQL(3hr)
- Meetings - Group meeting: 2/8/2015(3hr)
- Design - none
- Implementation - none
- Testing - none

Jackson(Weitong) Wang

- Assignments - Writing Project Report, Power Point: 2/8/2015(2hr)
- Background - Student basic MySQL knowledge(3hr)
- Meetings - Group meeting: 2/8/2015(3hr)
- Design - none
- Implementation - none
- Testing - none

WeiDi Shen

- Assignments - Writing Progress Report: 2/8/2015(1hr)
- Background - Studied web-based programming (3hr)
- Meetings - Group meeting: 2/8/2015(3hr)
- Design - none
- Implementation - none

- Testing -none

Seong Kyu Kim

- Assignments - Writing Project Report, Power Point: 2/8/2015(2hr)
- Background - Web development hands on: Node.js, Ajax, JQuery
- Meetings - Group meeting: 2/8/2015(2hr)
- Design - none
- Implementation - none
- Testing – none

Group Activities

- Group meeting(3hr) 2/8/2015
- Brainstorming and discussion about the project
- We have developed the first phase ER-model
- Writing project report, progress report, and power point for phase 1 presentation

Future Task

- Since we only have one teammate who has experience with web-programming, we are going to keep studying about web-programming, such as jsp.

Progress Report(1) for CMPSC431W

Summary

1/25/2015 – We decided the topic for the project:

Pokemon Open Dictionary where people can prompt the attributes and characteristics of a certain pokemon on web which is going to be saved in a DBMS. The saved data is to be interpreted to allow look up(search), communications between the users, and provide with team combination recommendations according to the user requirements.

Individual Activities

Eiji Iriyama

- Assignments - Writing progress report: 1/25/2015(1hr)
- Background - studied basic commands for MySQL(4hr)
- Meetings - Group meeting: 1/25/2015(2hr)
- Design - none
- Implementation - none
- Testing - none

Jackson(Weitong) Wang

- Assignments - Assist writing progress report: 1/25/2015(1hr)
- Background - Have experience in software application development with use of database
- Meetings - Group meeting: 1/25/2015(2hr)
- Design - none
- Implementation - none
- Testing - none

WeiDi Shen

- Assignments - Idea brainstorming, web implementation possibilities: 1/25/2015(1hr)
- Background - Studied basic concepts of jsp(2hr)
- Meetings - Group meeting: 1/25/2015(2hr)
- Design - none
- Implementation - none
- Testing - none

Seong Kyu Kim

- Assignments - Idea brainstorming, web implementation possibilities: 1/25/2014(1hr)
- Background - Web development experience with HTML, CSS, and JavaScript
- Meetings - Group meeting: 1/25/2015(2hr)
- Design - none

- Implementation - none
- Testing - none

Group Activities

- Group meeting(2hr) 1/25/2015

Appendix D Source Code

1. **About :**

```
<?php
include "header.php";

//**********************************print the explanation
print <<< eot
        <HTML>
                <HEAD>
                </HEAD>
                <BODY>
                <div class = "about">
                        <CENTER>
                                <FONT SIZE = "8">
                                        Library Hours<BR>
                                </FONT>
                                <FONT SIZE = "5">
                                        Monday - Friday         : 7:00am -
1:00am<BR>
                                        Saturday                : 12:00pm -
9:00pm<BR>
                                        Sunday                  : Closed<BR>
                        </CENTER>
                </div>
                </BODY>
        </HTML>


eot;

?>
```

2. **Account_info:**

```
<?php
include_once "global.php";
include_once "db_connect.php";
include "header.php";

$edit = isset($_GET["edit"])?$_GET["edit"]:null;
$submit = isset($_GET["submit"])?$_GET["submit"]:null;

if($submit<>""){
    $name = isset($_GET["name"]) ? $_GET["name"]:null;
    $gender = isset($_GET["gender"]) ? $_GET["gender"]:null;
    $dateOfBirth = isset($_GET["dateOfBirth"]) ? $_GET["dateOfBirth"]:null;
    $email = isset($_GET["email"]) ? $_GET["email"]:null;
    $userid = isset($_GET["userid"]) ? $_GET["userid"]:null;
    $password = isset($_GET["password"]) ? $_GET["password"]:null;
}else{
    $query = "select name, gender, dateOfBirth, email, memberid, password from
member where memberid = '$userid'";
    $return = mysql_query($query) or die(mysql_error());
    $result = mysql_fetch_array($return);

    $name = $result[0];
    $gender = $result[1];
    $dateOfBirth = $result[2];
    $email = $result[3];
    $userid = $result[4];
    $password = $result[5];

    $query = "select * from address where addressid = '$userid'";
    $return = mysql_query($query) or die(mysql_error());
    $result = mysql_fetch_array($return);

    $aptNumber = $result[1];
    $street = $result[2];
    $city = $result[3];
    $zipcode = $result[4];
}
if($edit<>""){
//************************************Edit pressed

print <<< eot1
    <HTML>
    <HEAD>
    </HEAD>
    <BODY>
            <CENTER>
                    <TABLE>
                            <FORM METHOD = "get" ACTION =
"account_info.php">
                            <TR><TH><BR>User Information</TH></TR>
                            <TR><TD>Name:</TD><TD><INPUT TYPE = "text"
VALUE = $name NAME = "name"></TD></TR>
eot1;
                    if($gender == "male"){
                            print '<TR><TD>Male:</TD><TD><INPUT TYPE =
"radio" NAME = "gender" VALUE = "male" checked></TD></TR>';
                            print '<TR><TD>Female:</TD><TD><INPUT TYPE =
"radio" NAME = "gender" VALUE = "female"></TD></TR>';
                    }else{
```

```
                                print '<TR><TD>Male:</TD><TD><INPUT TYPE = 
"radio" NAME = "gender" VALUE = "male"></TD></TR>';
                                print '<TR><TD>Female:</TD><TD><INPUT TYPE = 
"radio" NAME = "gender" VALUE = "female" checked></TD></TR>';
                        }
print <<< eot2
                                <TR><TD>Date Of Birth:</TD><TD><INPUT TYPE = 
"date" NAME = "dateOfBirth" VALUE = $dateOfBirth></TD></TR>
                                <TR><TD>Email:</TD><TD><INPUT TYPE = "text" 
VALUE = $email NAME = "email"></TD></TR>
                                <TR><TD>User ID:</TD><TD><INPUT TYPE = "text" 
VALUE = $userid NAME = "userid"></TD></TR>
                                <TR><TD>Password:</TD><TD><INPUT TYPE = "text" 
VALUE = $password NAME = "password"></TD></TR>
                                <TR><TH><BR>Address</TH></TR>
                                <TR><TD>Apartment #:</TD><TD><INPUT TYPE = 
"text" VALUE = $aptNumber NAME = "password"></TD></TR>
                                <TR><TD>Street:</TD><TD><INPUT TYPE = "text" 
VALUE = $street NAME = "password"></TD></TR>
                                <TR><TD>City:</TD><TD><INPUT TYPE = "text" 
VALUE = $city NAME = "password"></TD></TR>
                                <TR><TD>Zipcode:</TD><TD><INPUT TYPE = "text" 
VALUE = $zipcode NAME = "password"></TD></TR>
                                </FORM>
                        </TABLE>
                        <INPUT TYPE = "submit" VALUE = "Submit" NAME = 
"submit">
                </CENTER>
    </BODY>
    </HTML>
eot2;

//**********************************//initialize $edit to null
$edit = null;

}else if($edit == null && $submit == null){
    //**********************************print account infomation
print <<< eot3
                <HTML>
    <HEAD>
    <STYLE>
                table, TR{
                        border: 3px solid #a00000;
                        padding: 5px;
    }
    </STYLE>
    </HEAD>
    <BODY>
                <CENTER>
                        <TABLE>
                                <TR><TH>Your Account Information</TH></TR>
                                <TR><TD>NAME:</TD><TD>$name</TD></TR>
                                <TR><TD>GENDER:</TD><TD>$gender</TD></TR>
                                <TR><TD>Date Of
Birth:</TD><TD>$dateOfBirth</TD></TR>
                                <TR><TD>Email:</TD><TD>$email</TD></TR>
                                <TR><TD>User ID:</TD><TD>$userid</TD></TR>
                                <TR><TD>Password:</TD><TD>$password</TD></TR>
                                <TR><TH><BR>Address</TH></TR>
                                <TR><TD>Apartment
#:</TD><TD>$aptNumber</TD></TR>
                                <TR><TD>Street:</TD><TD>$street</TD></TR>
                                <TR><TD>City:</TD><TD>$city</TD></TR>
                                <TR><TD>Zipcode:</TD><TD>$zipcode</TD></TR>
                        </TABLE>
                        <FORM METHOD = "get" ACTION = "account_info.php">
                                        <INPUT TYPE = "submit" VALUE = "Edit"
NAME = "edit">
                                </FORM>
                </CENTER>
    </BODY>
    </HTML>
eot3;
}else if($edit == null && $submit<>""){
    //***********************************************user changed the
account info

    if(!($name<>"") || !($gender<>"") || !($dateOfBirth<>"") || !($email<>"")
|| !($userid<>"") || !($password<>""))
    {
                print "Invalid Input in sign up!!!<BR>";
    }else{
    $query = "select reserveCount from member where memberid = '$userid'";
    $return = mysql_query($query) or die(mysql_error());
    $result = mysql_fetch_array($return);
    $reserveCount = $result[0];

    $query = "delete from member where memberid = '$userid'";
    $return = mysql_query($query) or die(mysql_error());

                $query = "insert into member values('$email', '$password', '$gender',
'$dateOfBirth', '$reserveCount' ,'$name', '$userid')";
                $result = mysql_query($query) or die(mysql_error());
    }
```

```
		$submit = null;
		header("location:account_info.php");
	}
	?>
```

3. Add_admin:

```
<?php
include_once "global.php";
include_once "db_connect.php";
include "admin_header.php";

$adminid = isset($_GET["adminid"])?$_GET["adminid"]:null;
$deliveryid = isset($_GET["deliveryid"])?$_GET["deliveryid"]:null;
$telemarketid = isset($_GET["telemarketid"])?$_GET["telemarketid"]:null;
$passwordAdmin = isset($_GET["passwordAdmin"])?$_GET["passwordAdmin"]:null;
$passwordDelivery = isset($_GET["passwordDelivery"])?$_GET["passwordDelivery"]:null;
$passwordTelemarket =
isset($_GET["passwordTelemarket"])?$_GET["passwordTelemarket"]:null;
$admin = isset($_GET["admin"])?$_GET["admin"]:null;
$delivery = isset($_GET["delivery"])?$_GET["delivery"]:null;
$telemarket = isset($_GET["telemarket"])?$_GET["telemarket"]:null;

if($admin<>""){
		$query = "insert into admin values('$adminid','$passwordAdmin')";
		$return = mysql_query($query);
print <<< eot
		<CENTER>
		<FONT SIZE = 8>
				A NEW ADMINISTRATOR ACCOUNT HAS BEEN CREATED<BR>
		</FONT>
		</CENTER>
eot;
}else if($delivery<>""){
		$query = "insert into delivery values('$deliveryid','$passwordDelivery')";
		$return = mysql_query($query);
print <<< eot
		<CENTER>
		<FONT SIZE = 8>
				A NEW DELIVERY STAFF ACCOUNT HAS BEEN CREATED<BR>
		</FONT>
		</CENTER>
eot;
}else if($telemarket<>""){
		$query = "insert into telemarket values('$telemarketid','$passwordTelemarket')";
		$return = mysql_query($query);
print <<< eot
		<CENTER>
```

```
    <FONT SIZE = 8>
        A NEW TELEMARKET STAFF ACCOUNT HAS BEEN CREATED<BR>
    </FONT>
    </CENTER>
eot;
}

print <<< eot3
        <HTML>
    <HEAD>
    <STYLE>
        table, TR{
            border: 3px solid #a00000;
            padding: 5px;
    }
    </STYLE>
    </HEAD>
    <BODY>
        <CENTER>
                <FORM METHOD = "get" ACTION = "add_admin.php">
            <TABLE>
                <TR><TH>Create a new administrator
account</TH></TR>
                <TR><TD>Administrator ID:</TD><TD><INPUT TYPE
= "text" NAME = adminid></TD></TR>
                <TR><TD>Password:</TD><TD><INPUT TYPE = "text"
NAME = passwordAdmin></TD></TR>
                <TR><TD><INPUT TYPE = "submit" VALUE = "Enter"
NAME = "admin"></TD></TR>
                <TR><TD><BR></TR></TD>
                <TR><TH>Create a new delivery staff
account</TH></TR>
                <TR><TD>Delivery staff ID:</TD><TD><INPUT TYPE =
"text" NAME = deliveryid></TD></TR>
                <TR><TD>Password:</TD><TD><INPUT TYPE = "text"
NAME = passwordDelivery></TD></TR>
                <TR><TD><INPUT TYPE = "submit" VALUE = "Enter"
NAME = "delivery"></TD></TR>
                <TR><TD><BR></TR></TD>
                <TR><TH>Create a new temelarket staff
account</TH></TR>
                <TR><TD>Telemarket staff ID:</TD><TD><INPUT
TYPE = "text" NAME = telemarketid></TD></TR>
                <TR><TD>Password:</TD><TD><INPUT TYPE = "text"
NAME = passwordTelemarket></TD></TR>
                <TR><TD><INPUT TYPE = "submit" VALUE = "Enter"
NAME = "telemarket"></TD></TR>
```

```
                              </TABLE>
                                        </FORM>
                    </CENTER>
          </BODY>
          </HTML>
eot3;


?>
```

4. Add_book:

```
<?php
include_once "global.php";
include "admin_header.php";
include_once "db_connect.php";

//*******************************receive book info from add_book.php
$isbn = isset($_GET["isbn"])?$_GET["isbn"]:null;
$author = isset($_GET["author"])?$_GET["author"]:null;
$title = isset($_GET["title"])?$_GET["title"]:null;
$category = isset($_GET["category"])?$_GET["category"]:null;
$description = isset($_GET["description"])?$_GET["description"]:null;
$url = isset($_GET["url"])?$_GET["url"]:null;
$publisher = isset($_GET["publisher"])?$_GET["publisher"]:null;
$stock = isset($_GET["stock"])?$_GET["stock"]:null;
$enter = isset($_GET["enter"])?$_GET["enter"]:null;

$rating = 0;

//admin entered add book
if($enter<>""){
          //*************************************insert escape character before all
the ' char
          $description = addslashes($description);
          $query = "insert into books
values('$isbn','$rating','$author','$title','$category','$description','$url','$publisher','$stock')";
          $return = mysql_query($query) or die(mysql_error());
print <<< eot
          <CENTER>
          <FONT SIZE = 8>
                    THE BOOK IS SUCCESSFULLY ADDED<BR>
          </FONT>
          </CENTER>
eot;
}
print <<< eot3
```

```
        <HTML>
    <HEAD>
    <STYLE>
        table, TR{
            border: 3px solid #a00000;
            padding: 5px;
    }
    </STYLE>
    </HEAD>
    <BODY>
        <CENTER>
            <TABLE>
                <FORM METHOD = "get" ACTION = "add_books.php">
                <TR><TH>Fill out the book information</TH></TR>
                <TR><TD>ISBN:</TD><TD><INPUT TYPE = "text"
NAME = isbn></TD></TR>
                <TR><TD>AUTHOR:</TD><TD><textarea NAME =
author ROWS = 1 COLS = 100></textarea></TD></TR>
                <TR><TD>TITLE:</TD><TD><textarea NAME = title
ROWS = 5 COLS = 100></textarea></TD></TR>
                <TR><TD>CATEGORY:</TD><TD><SELECT NAME =
"category">
                        <OPTION VALUE =
"art">Art</OPTION>
                        <OPTION VALUE =
"business">Business</OPTION>
                        <OPTION VALUE =
"children">Children</OPTION>
                        <OPTION VALUE =
"fiction">Fiction</OPTION>
                        <OPTION VALUE =
"history">History</OPTION>
                        <OPTION VALUE =
"medicines">Medicines</OPTION>
                        <OPTION VALUE =
"religion">Religion</OPTION>
                        <OPTION VALUE =
"science">Science</OPTION>
                    </SELECT></TD></TR>
                <TR><TD>DESCRIPTION:</TD><TD><textarea NAME
= description ROWS = 15 COLS = 100></textarea></TD></TR>
                <TR><TD>URL:</TD><TD><textarea NAME = url
ROWS = 5 COLS = 100></textarea></TD></TR>
                <TR><TD>PUBLISHER:</TD><TD><textarea NAME =
publisher ROWS = 3 COLS = 100></textarea></TD></TR>
                <TR><TD>STOCK:</TD><TD><INPUT TYPE = "text"
NAME = stock></TD></TR>
```

```
                              </TABLE>
                                                <INPUT TYPE = "submit" VALUE = "Enter"
NAME = "enter">
                                        </FORM>
                    </CENTER>
          </BODY>
          </HTML>
eot3;

?>
```

5. admin_header:

```
<?php

//*******************************read adminid from file created in the top page
$file = "adminid.txt";
$userid = file_get_contents($file);

//*************************************************
print <<< eot2
          <HTML>
          <HEAD>
                    <link rel="stylesheet" type="text/css" href="style.css"/>
                    <TITLE>MY MONGO MARIA</TITLE>
          </HEAD>
          <BODY>
                    <BODY BGCOLOR = "AliceBlue">
                    <div class="adminNavbar">
                              <ul>
                                        <li><a href="admin_page.php">Home</a></li>
                                        <li><a href="add_books.php">Add Books</a></li>
                                        <li><a href="delete_books.php">Delete Books</a></li>
                                        <li><a href="confirm_rent.php">Confirm Rent</a></li>
                                        <li><a href="confirm_return.php">Confirm
Return</a></li>
                                        <li><a href="blacklist.php">BlackList</a></li>
                                        <li><a href="add_admin.php">Add Administrator</a></li>
                              </ul>
                    </div>
                    <CENTER>
                              <IMG SRC = "image/logo.png">
                              <BR><BR>
                    </CENTER>
          </BODY>
          </HTML>
eot2;
```

```
?>
```

6. admin_page:

```
<?php
include_once "global.php";
include "admin_header.php";
include_once "db_connect.php";


?>
```

7. background:

```
<?php
print <<< eot1

        <HTML>

                <HEAD>
                <TITLE>MY MONGO MARIA</TITLE>
                </HEAD>
                <BODY BGCOLOR = "AliceBlue">
                <BODY>
                <CENTER>
                        <IMG SRC = "image/logo.png">
                        <BR><BR>
                </CENTER>
                </BODY>
                </HTML>
eot1;
?>
```

8. blacklist:

```
<?php
include_once "global.php";
include_once "db_connect.php";
include "admin_header.php";

$query = "select m.name, avg(u.rating), r.memberid from member as m, rents as r, userrating
as u where m.memberid = r.memberid and r.transactionid = u.transactionid group by
r.memberid";
$return = mysql_query($query)or die(mysql_error());
while($result = mysql_fetch_array($return)){
print <<< eot
```

```
        <CENTER>
                        <TABLE>
                                <TR><TH>User Rating Order by Ascending</TH></TR>
                                <TR><TD>NAME:</TD><TD>$result[0]</TD></TR>
                                <TR><TD>MEMBER
ID:</TD><TD>$result[2]</TD></TR>
                                <TR><TD>RATING:</TD><TD>$result[1]</TD></TR>
                                <TR><TD><BR></TR></TD>
                        </TABLE>
        </CENTER>
eot;
}


?>
```

9. confirm_rent

```
<?php
include_once "global.php";
include_once "db_connect.php";
include "admin_header.php";

//receive variables from confirm_return.php
$select = isset($_GET["select"])?$_GET["select"]:null;
$confirm = isset($_GET["confirm"])?$_GET["confirm"]:null;
$idRent = isset($_GET["idRent"])?$_GET["idRent"]:null;
$nameRent = isset($_GET["nameRent"])?$_GET["nameRent"]:null;
$isbnRent = isset($_GET["isbnRent"])?$_GET["isbnRent"]:null;

if($confirm<>""){
        //get memberid
        //********************************************insert a new rent tuple into
rents
        $date = date("Y-m-d");
        $query = "insert into rents (memberid, isbn, returnConfirm, returnDate, rentDate)
values ('$idRent','$isbnRent',0,null,'$date')";
        $return = mysql_query($query) or die(mysql_error());
        //**********************************************delete the confirmed
reservation
        $query = "delete from reservations where memberid = '$idRent' and isbn =
'$isbnRent'";
        $return = mysql_query($query) or die(mysql_error());

        $select = null;
}
if($select<>""){
	//**********************************get user's name
	$query = "select name from member where memberid = '$idRent'";
	$return = mysql_query($query) or die(mysql_error());
	$nameResult = mysql_fetch_array($return);
	print "USER NAME: ".$nameResult[0];

	$query = "select * from reservations where memberid = '$idRent'";
	$return = mysql_query($query) or die(mysql_error());
	while($result = mysql_fetch_array($return)){
		$queryTitle = "select title from books where isbn = '$result[1]'";
		$returnTitle = mysql_query($queryTitle) or die(mysql_error());
		$resultTitle = mysql_fetch_array($returnTitle);

print <<< eot3
		<TABLE>
			<TR><TD>REVERVE ID:</TD><TD>$result[3]</TD></TR>
			<TR><TD>ISBN:</TD><TD>$result[1]</TD></TR>
			<TR><TD>TITLE:</TD><TD>$resultTitle[0]</TD></TR>
			<TR><TD>RESERVED
DATE:</TD><TD>$result[2]</TD></TR>
		</TABLE>
			<FORM METHOD = "get" ACTION = "confirm_rent.php">
			<INPUT TYPE = "hidden" VALUE = $idRent NAME = "idRent">
			<INPUT TYPE = "hidden" VALUE = $nameResult[0] NAME =
"nameRent">
			<INPUT TYPE = "hidden" VALUE = $result[1] NAME =
"isbnRent">
			<INPUT TYPE = "submit" VALUE = "Confirm" NAME =
"confirm">
			</FORM>
eot3;
	}
}else{

print <<< eot1
		<CENTER>
		<FONT SIZE = "6">
			Select a User from Name
		</FONT>
		<FORM METHOD = "get" ACTION = "confirm_rent.php">
			<SELECT NAME = "idRent">
eot1;

	//*******************************************

	$query = "select distinct memberid from reservations";
	$return = mysql_query($query) or die(mysql_error());
```

```
    while($result = mysql_fetch_array($return)){
        $queryName = "select name from member where memberid = '$result[0]'";
        $returnName = mysql_query($queryName) or die(mysql_error());
        $resultName = mysql_fetch_array($returnName);
        print "<OPTION VALUE = $result[0]>$resultName[0]</OPTION>";
    }

print <<< eot2
            </SELECT></BR>
            <INPUT TYPE = "submit" VALUE = "Select" NAME = "select">
            </FORM>
            <BR><BR>
        </CENTER>
eot2;
}
?>
```

10. confirm return:

```
<?php
include_once "global.php";
include_once "db_connect.php";
include "admin_header.php";

//receive variables from confirm_return.php
$select = isset($_GET["select"])?$_GET["select"]:null;
$confirm = isset($_GET["confirm"])?$_GET["confirm"]:null;
$idReturn = isset($_GET["idReturn"])?$_GET["idReturn"]:null;
$nameReturn = isset($_GET["nameReturn"])?$_GET["nameReturn"]:null;
$isbnReturn = isset($_GET["isbnReturn"])?$_GET["isbnReturn"]:null;
$rating = isset($_GET["rating"])?$_GET["rating"]:null;
$transactionid = isset($_GET["transactionid"])?$_GET["transactionid"]:null;

if($confirm<>""){
    //************************************************update the return information
    $date = date("Y-m-d");
    $query = "update rents set returnDate = '$date' where memberid = '$idReturn' and isbn = '$isbnReturn'";
    $return = mysql_query($query) or die(mysql_error());

    $query = "update rents set returnConfirm = 1 where memberid = '$idReturn' and isbn = '$isbnReturn'";
    $return = mysql_query($query) or die(mysql_error());

    $query = "insert into userrating (rating, transactionid) values ('$rating','$transactionid')";
```

```
		$return = mysql_query($query) or die(mysql_error());
		//increment stock by 1 in books
		$query = "update books set stock = stock + 1 where isbn = '$isbnR' and stock > 0";
		$return = mysql_query($query) or die(mysql_error());
print <<< eot
		<HTML>
		<HEAD>
		</HEAD>
		<BODY>
			<CENTER>
				<FONT SIZE = 8>
					The return is confirmed<BR>
				</FONT>
			</CENTER>
		</BODY>
		</HTML>

eot;
		$select = null;
}
if($select<>""){
			//**********************************get user's name
		$query = "select name from member where memberid = '$idReturn'";
		$return = mysql_query($query) or die(mysql_error());
		$nameResult = mysql_fetch_array($return);
		print "USER NAME: ".$nameResult[0]."<BR><BR>";

		$query = "select * from rents where memberid = '$idReturn' and returnConfirm = 0";
		$return = mysql_query($query) or die(mysql_error());
		while($result = mysql_fetch_array($return)){
			$queryTitle = "select title from books where isbn = '$result[2]'";
			$returnTitle = mysql_query($queryTitle) or die(mysql_error());
			$resultTitle = mysql_fetch_array($returnTitle);

print <<< eot3
			<TABLE>
				<TR><TD>TRANSACTION
ID:</TD><TD>$result[0]</TD></TR>
				<TR><TD>ISBN:</TD><TD>$result[2]</TD></TR>
				<TR><TD>TITLE:</TD><TD>$resultTitle[0]</TD></TR>
				<TR><TD>RENT DATE:</TD><TD>$result[5]</TD></TR>
			</TABLE>
				<FORM METHOD = "get" ACTION = "confirm_return.php">
				<INPUT TYPE = "hidden" VALUE = $idReturn NAME =
"idReturn">
				<INPUT TYPE = "hidden" VALUE = $nameResult[0] NAME =
"nameReturn">
```

```
            <INPUT TYPE = "hidden" VALUE = $result[2] NAME = 
"isbnReturn">
            <INPUT TYPE = "hidden" VALUE = $result[0] NAME = 
"transactionid">

            <input id="rating10" type="radio" name="rating" value="10">
            <label for="rating10">10</label>
            <input id="rating9" type="radio" name="rating" value="9">
            <label for="rating9">9</label>
            <input id="rating8" type="radio" name="rating" value="8">
            <label for="rating8">8</label>
            <input id="rating7" type="radio" name="rating" value="7">
            <label for="rating7">7</label>
            <input id="rating6" type="radio" name="rating" value="6">
            <label for="rating6">6</label>
            <input id="rating5" type="radio" name="rating" value="5">
            <label for="rating5">5</label>
            <input id="rating4" type="radio" name="rating" value="4">
            <label for="rating4">4</label>
            <input id="rating3" type="radio" name="rating" value="3">
            <label for="rating3">3</label>
            <input id="rating2" type="radio" name="rating" value="2">
            <label for="rating2">2</label>
            <input id="rating1" type="radio" name="rating" value="1">
            <label for="rating1">1</label>
            <BR>

            <INPUT TYPE = "submit" VALUE = "Confirm" NAME = 
"confirm">
            </FORM>
eot3;
    }
}else{
    //****************************************first page when admin click 
confirm return
print <<< eot1
        <CENTER>
        <FONT SIZE = "6">
            Select a User to confirm book returned
        </FONT>

        <FORM METHOD = "get" ACTION = "confirm_return.php">
            <SELECT NAME = "idReturn">
eot1;
    //***********************************************

    $query = "select memberid from rents where returnConfirm = 0";
```

```
        $return = mysql_query($query) or die(mysql_error());
        while($result = mysql_fetch_array($return)){
                $queryName = "select name from member where memberid = '$result[0]'";
                $returnName = mysql_query($queryName) or die(mysql_error());
                $resultName = mysql_fetch_array($returnName);
                print   "<OPTION VALUE = $result[0]>$resultName[0]</OPTION>";
        }

print <<< eot2
                        </SELECT></BR>
                        <INPUT TYPE = "submit" VALUE = "Select" NAME = "select">
                        </FORM>
                        </CENTER>
                        <BR><BR>
eot2;
}
?>
```

11. db_connect:

```
<?php
//*************************connect to the database
include_once("global.php");
$session = mysql_connect($server, $user, $password);
mysql_select_db($dbName);
?>
```

12. delete_book:

```
<?php
include_once "global.php";
include_once "db_connect.php";
include "admin_header.php";

$isbn = isset($_GET["isbn"])?$_GET["isbn"]:null;
$delete = isset($_GET["delete"])?$_GET["delete"]:null;
if($delete<>""){
        $query = "delete from books where isbn like '$isbn'";
        $return = mysql_query($query) or die(mysql_error());

print <<< eot
        <CENTER>
        <FONT SIZE = 8>
                THE BOOK IS SUCCESSFULLY DELETED<BR>
        </FONT>
        </CENTER>
eot;
}


print <<< eot3
        <HTML>
    <HEAD>
    <STYLE>
        table, TR{
            border: 3px solid #a00000;
            padding: 5px;
    }
    </STYLE>
    </HEAD>
    <BODY>
        <CENTER>
            <TABLE>
                <FORM METHOD = "get" ACTION =
"delete_books.php">
                <TR><TH>Enter the ISBN of the book to
delete</TH></TR>
                <TR><TD>ISBN:</TD><TD><INPUT TYPE = "text"
NAME = isbn></TD></TR>
            </TABLE>
                    <INPUT TYPE = "submit" VALUE = "Delete"
NAME = "delete">
                </FORM>
        </CENTER>
    </BODY>
    </HTML>
eot3;


?>
```

13. delivery_header:

```
<?php

//*********************************read adminid from file created in the top page
$file = "deliveryid.txt";
$userid = file_get_contents($file);

//****************************************************
print <<< eot2
    <HTML>
    <HEAD>
        <link rel="stylesheet" type="text/css" href="style.css"/>
```

```
		<TITLE>MY MONGO MARIA</TITLE>
	</HEAD>
	<BODY>
		<BODY BGCOLOR = "AliceBlue">
		<CENTER>
			<IMG SRC = "image/logo.png">
			<BR><BR>
		</CENTER>
	</BODY>
	</HTML>
eot2;

?>
```

14. delivery_page:

```
<?php
include_once "global.php";
include_once "db_connect.php";
include_once "delivery_header.php";

//receive variables from confirm_return.php
$select = isset($_GET["select"])?$_GET["select"]:null;
$confirm = isset($_GET["confirm"])?$_GET["confirm"]:null;
$idRent = isset($_GET["idRent"])?$_GET["idRent"]:null;
$nameRent = isset($_GET["nameRent"])?$_GET["nameRent"]:null;
$isbnRent = isset($_GET["isbnRent"])?$_GET["isbnRent"]:null;

if($confirm<>""){
	//get memberid
	//*******************************************insert a new rent tuple into
rents
	$date = date("Y-m-d");
	$query = "insert into rents (memberid, isbn, returnConfirm, returnDate, rentDate)
values ('$idRent','$isbnRent',0,null,'$date')";
	$return = mysql_query($query) or die(mysql_error());
	//*********************************************delete the confirmed
reservation
	$query = "delete from reservations where memberid = '$idRent' and isbn =
'$isbnRent'";
	$return = mysql_query($query) or die(mysql_error());

	$select = null;
}
if($select<>""){
	//**********************************get user's name
	$query = "select name from member where memberid = '$idRent'";
```

```
        $return = mysql_query($query) or die(mysql_error());
        $nameResult = mysql_fetch_array($return);
        print "USER NAME: ".$nameResult[0];

        $query = "select * from reservations where memberid = '$idRent'";
        $return = mysql_query($query) or die(mysql_error());
        while($result = mysql_fetch_array($return)){
                $queryTitle = "select title from books where isbn = '$result[1]'";
                $returnTitle = mysql_query($queryTitle) or die(mysql_error());
                $resultTitle = mysql_fetch_array($returnTitle);

print <<< eot3
                <TABLE>
                        <TR><TD>REVERVE ID:</TD><TD>$result[3]</TD></TR>
                        <TR><TD>ISBN:</TD><TD>$result[1]</TD></TR>
                        <TR><TD>TITLE:</TD><TD>$resultTitle[0]</TD></TR>
                        <TR><TD>RESERVED
DATE:</TD><TD>$result[2]</TD></TR>
                </TABLE>
                        <FORM METHOD = "get" ACTION = "confirm_rent.php">
                        <INPUT TYPE = "hidden" VALUE = $idRent NAME = "idRent">
                        <INPUT TYPE = "hidden" VALUE = $nameResult[0] NAME =
"nameRent">
                        <INPUT TYPE = "hidden" VALUE = $result[1] NAME =
"isbnRent">
                        <INPUT TYPE = "submit" VALUE = "Confirm" NAME =
"confirm">
                        </FORM>
eot3;
        }
}else{

print <<< eot1
                <CENTER>
                <FONT SIZE = "6">
                        Select a User from Name
                </FONT>

                <FORM METHOD = "get" ACTION = "confirm_rent.php">
                        <SELECT NAME = "idRent">
eot1;

        //***********************************************

        $query = "select distinct memberid from reservations";
        $return = mysql_query($query) or die(mysql_error());
        while($result = mysql_fetch_array($return)){
```

```
			$queryName = "select name from member where memberid = '$result[0]'";
			$returnName = mysql_query($queryName) or die(mysql_error());
			$resultName = mysql_fetch_array($returnName);
			print "<OPTION VALUE = $result[0]>$resultName[0]</OPTION>";
		}

print <<< eot2
				</SELECT></BR>
				<INPUT TYPE = "submit" VALUE = "Select" NAME = "select">
				</FORM>
				<BR><BR>
			</CENTER>
eot2;
}
?>
```

15. finish_reserve:

```
<?php
include_once "global.php";
include_once "db_connect.php";
include "header.php";

  //**********************************receive isbn from reserve.php
$date = date("Y-m-d");

$isbnR = isset($_GET["isbnR"])?$_GET["isbnR"]:null;
$query = "insert into reservations(memberid, isbn, date) values ('$userid', '$isbnR', '$date')";
$return = mysql_query($query) or die(mysql_error());

//***********************************increment by 1 for reserveCount in member
$query = "update member set reserveCount = reserveCount + 1 where memberid = '$userid'";
$return = mysql_query($query) or die(mysql_error());

//***********************************decrement 1 from stock in books
$query = "update books set stock = stock - 1 where isbn = '$isbnR' and stock > 0";
$return = mysql_query($query) or die(mysql_error());
print <<< eot
		<HTML>
		<HEAD>
		</HEAD>
		<BODY>
			<CENTER>
				<FONT SIZE = 8>
					You successfully reserved the book!<BR>
					Please pick it up in a week from "$date"
				</FONT>
```

```
        </CENTER>
    </BODY>
    </HTML>

eot;
?>
```

16. Global:

```
<?php
//*************************define all the variables for login
$server = "localhost";
$user = "root";
$password = "aoyama0118";
$dbName = "lib_system";
$userid;

?>
```

17. Header:

```
<?php
//******************************************read userid from file created in top page
$file = "userid.txt";
$userid = file_get_contents($file);
//************************************************************
print <<< eot2
    <HTML>
    <HEAD>
        <link rel="stylesheet" type="text/css" href="style.css">
        <TITLE>MY MONGO MARIA</TITLE>
    </HEAD>
    <BODY>
        <BODY BGCOLOR = "AliceBlue">
        <div class="adminNavbar">
            <ul>
                <li><a href="user_page.php">Home</a></li>
                <li><a href="about.php">About</a></li>
                <li><a href="reserve.php">Reserve</a></li>
                <li><a href="history.php">History</a></li>
                <li><a href="account_info.php">Account Info</a></li>
            </ul>
        </div>
        <CENTER>
            <IMG SRC = "image/logo.png">
            <BR><BR>
```

```
			</CENTER>
		</BODY>
		</HTML>
eot2;

?>
```

18. History:

```
<?php
include_once "global.php";
include_once "db_connect.php";
include "header.php";

//****************************************get variable from history.php
$rate = isset($_GET["rate"])?$_GET["rate"]:null;
$rating = isset($_GET["rating"])?$_GET["rating"]:null;
$isbn = isset($_GET["isbn"])?$_GET["isbn"]:null;
$comment = isset($_GET["comment"])?$_GET["comment"]:null;

if($rate<>""){
		$query = "select count(*) from rating where memberid = '$userid' and isbn = '$isbn'";
		$return = mysql_query($query) or die(mysql_error());
		$num_row = mysql_fetch_array($return);
		//create a new rating if it doesn's exist
		if($num_row[0] == 0){
			print "NEW";
			$query = "insert into rating(isbn, starPoint, comment, memberid) values
('$isbn','$rating','$comment','$userid')";
			$return = mysql_query($query) or die(mysql_error());
			$result = mysql_fetch_array($return);
		}else{
			print "OLD";
			//*******************************************if the rating
exists, update it
			$query = "update rating set starPoint = '$rating' where memberid = '$userid'
and isbn = '$isbn'";
			$return = mysql_query($query) or die(mysql_error());
			$query = "update rating set comment = '$comment' where memberid =
'$userid' and isbn = '$isbn'";
			$return = mysql_query($query) or die(mysql_error());

		}
}

print <<< eot4
		<FONT SIZE = "6">Books Rented</FONT>
```

```
eot4;

//**********************************************receive data from history.php
$query = "select distinct isbn from rents where memberid = '$userid'";
$return = mysql_query($query) or die(mysql_error());

        while($result = mysql_fetch_array($return)){
        //**************************************get book title
        $queryTitle = "select title from books where isbn = '$result[0]'";
        $returnTitle = mysql_query($queryTitle) or die(mysql_error());
        $resultTitle = mysql_fetch_array($returnTitle);

        $query = "select transactionid, rentDate, returnDate from rents where memberid =
'$userid' and isbn = '$result[0]'";
        $returnExtra = mysql_query($query) or die(mysql_error());
        $resultExtra = mysql_fetch_array($returnExtra);
        //**************************************get book rating
        $queryRate = "select starPoint, comment from rating where memberid = '$userid' and
isbn = '$result[0]'";
        $returnRate = mysql_query($queryRate) or die(mysql_error());
        $resultRate = mysql_fetch_array($returnRate);
        $comment = $resultRate[1];
        if($resultRate[0] == null){
                $rating = 0;
        }else{
                $rating = $resultRate[0];
        }

print <<< eot1
                        <TABLE>
                                <FONT SIZE = "5">
                                        <TR><TD>Transaction
ID:</TD><TD>$resultExtra[0]</TD></TR>

        <TR><TD>TITLE:</TD><TD>$resultTitle[0]</TD></TR>

        <TR><TD>ISBN:</TD><TD>$result[0]</TD></TR>
                                        <TR><TD>Rent
Date:</TD><TD>$resultExtra[1]</TD></TR>
                                        <TR><TD>Return
Date:</TD><TD>$resultExtra[2]</TD></TR>
                                        <TR><TD>Rate:</TD><TD>$rating</TD></TR>
                                </FONT>
                        </TABLE>
                        <FORM METHOD = "get" ACTION = "history.php">
                                <input id="rating10" type="radio" name="rating"
value="10">
```

```
                                        <label for="rating10">10</label>
                                        <input id="rating9" type="radio" name="rating" value="9">
                                        <label for="rating9">9</label>
                                        <input id="rating8" type="radio" name="rating" value="8">
                                        <label for="rating8">8</label>
                                        <input id="rating7" type="radio" name="rating" value="7">
                                        <label for="rating7">7</label>
                                        <input id="rating6" type="radio" name="rating" value="6">
                                        <label for="rating6">6</label>
                                        <input id="rating5" type="radio" name="rating" value="5">
                                        <label for="rating5">5</label>
                                        <input id="rating4" type="radio" name="rating" value="4">
                                        <label for="rating4">4</label>
                                        <input id="rating3" type="radio" name="rating" value="3">
                                        <label for="rating3">3</label>
                                        <input id="rating2" type="radio" name="rating" value="2">
                                        <label for="rating2">2</label>
                                        <input id="rating1" type="radio" name="rating" value="1">
                                        <label for="rating1">1</label>
                                        <BR>
                                        COMMENT     :<BR><textarea NAME = comment ROWS
= 10 COLS = 100>$comment</textarea>
                                        <BR>
                                        <INPUT TYPE = "hidden" VALUE = $result[0] NAME =
"isbn">
                                        <INPUT TYPE = "submit" VALUE = "Rate" NAME =
"rate">
                              </FORM>
                              <BR><BR>
eot1;
}
print "<BR><HR><BR>";

print <<< eot4
          <FONT SIZE = "6">Books Reserved</FONT>
eot4;

//*******************************************print all the reserved books
$query = "select distinct isbn from reservations where memberid = '$userid'";
$return = mysql_query($query) or die(mysql_error());

while($result = mysql_fetch_array($return)){
          //**************************************get book title
          $queryTitle = "select title from books where isbn = '$result[1]'";
          $returnTitle = mysql_query($queryTitle) or die(mysql_error());
          $resultTitle = mysql_fetch_array($returnTitle);
```

```
    $query = "select reserveid, date from reservations where userid = '$userid' and isbn = 
'$result[0]'";
    $return = mysql_query($query) or die(mysql_error());
    $idDate = mysql_fetch_array($return);

print <<< eot2
            <TABLE>
                <FONT SIZE = "5">
                    <TR><TD>TRANSACTION 
ID:</TD><TD>$idDate[0]</TD></TR>

    <TR><TD>TITLE:</TD><TD>$resultTitle[0]</TD></TR>

    <TR><TD>ISBN:</TD><TD>$result[1]</TD></TR>
                    <TR><TD>Reservation 
Date:</TD><TD>$idDate[1]</TD></TR>
                    <TR><TD>Rate:</TD><TD>Not 
Available</TD></TR>
                </FONT>
                <BR><BR>
            </TABLE>
eot2;
}
?>
```

19. Reserve

```
<?php
include_once "global.php";
include_once "db_connect.php";
include "header.php";

//************************************receive variables from reserve.php
$title = isset($_GET["title"])?$_GET["title"]:null;
$isbn = isset($_GET["isbn"])?$_GET["isbn"]:null;
$category = isset($_GET["category"])?$_GET["category"]:null;
$description = isset($_GET["description"])?$_GET["description"]:null;

$titleS = isset($_GET["titleS"])?$_GET["titleS"]:null;
$isbnS = isset($_GET["isbnS"])?$_GET["isbnS"]:null;
$categoryS = isset($_GET["categoryS"])?$_GET["categoryS"]:null;
$descriptionS = isset($_GET["descriptionS"])?$_GET["descriptionS"]:null;

//************************************get SQL statement if user searches
if($title<>"" && $titleS<>""){
    $query = "select * from books where title like '%$title%'";
}else if($isbn<>"" && $isbnS<>""){
    $query = "select * from books where isbn like '$isbn'";
}else if($category<>"" && $categoryS<>""){
    $query = "select * from books where category like '%$category%'";
}else if($description<>"" && $descriptionS<>""){
    $query = "select * from books where description like '%$description%'";
} else if($titleS<>"" || $isbnS<>"" || $category<>"" || $descriptionS<>""){
    print "Invalid Input!!!<BR>";
}

//********************************************count the number of books user renting +
reserving
$check = "select count(*) from reservations where memberid = '$userid'";
$return = mysql_query($check) or die(mysql_error());
$result = mysql_fetch_array($return);
$total = $result[0];

$check = "select count(*) from rents where memberid = '$userid' and returnConfirm = 0";
$return = mysql_query($check) or die(mysql_error());
$result = mysql_fetch_array($return);
$total = $total + $result[0];
//********************************************user searches
if($titleS<>"" || $isbnS<>"" || $categoryS<>"" || $descriptionS<>""){
    //************************************search the books
    $return = mysql_query($query) or die(mysql_error());

    //************************************prints on the results
    while($result = mysql_fetch_array($return)){
        $result[5] = nl2br($result[5]);
print <<< eot2
        <TABLE>
        <HR>
        <TR><TD>ISBN:</TD><TD>$result[0]</TD><TR>
        <TR><TD>Rating:</TD><TD>$result[1]</TD><TR>
        <TR><TD>AUTHOR:</TD><TD>$result[2]</TD><TR>
        <TR><TD>TITLE:</TD><TD>$result[3]</TD><TR>
        <TR><TD>CATEGORY:</TD><TD>$result[4]</TD><TR>
        <TR><TD>DESCRIPTION:</TD><TD>$result[5]</TD><TR>
        <TR><TD>URL:</TD><TD>$result[6]</TD><TR>
        <TR><TD>PUBLISHER:</TD><TD>$result[7]</TD><TR>
        <TR><TD>STOCK:</TD><TD>$result[8]</TD><TR>
        </TABLE>
eot2;
        if($result[8] > 0 && $total >= 0 && $total < 10){
print <<< eot3
                <FORM METHOD = "get" ACTION =
"finish_reserve.php">
                <INPUT TYPE = "hidden" VALUE = $result[0] NAME =
"isbnR">
                                        <INPUT TYPE = "submit" VALUE = "Reserve" NAME = 
"reserve"></FORM>
eot3;
                    print "<HR>";
                    }
          }
}else{
print <<< eot1
          <HTML>
          <HEAD>
          </HEAD>
          <BODY>
                    <CENTER>
                              <TABLE>
                                        <FORM METHOD = "get" ACTION = "reserve.php">
                                        <TR>
                                                  <TH>Search by Title</TH>
                                                  <TH>Search by ISBN</TH>
                                                  <TH>Search by Category</TH>
                                                  <TH>Search by Description</TH>
                                        </TR>
                                        <TR>
                                                  <TD><INPUT TYPE = "text" NAME = 
"title"></TD>
                                                  <TD><INPUT TYPE = "text" NAME = 
"isbn"></TD>
                                                  <TD><SELECT NAME = "category">
                                                            <OPTION VALUE = 
"fiction">Fiction</OPTION>
                                                            <OPTION VALUE = 
"science">Science</OPTION>
                                                            <OPTION VALUE = 
"mathematics">Mathematics</OPTION>
                                                            <OPTION VALUE = 
"business">Business</OPTION>
                                                            <OPTION VALUE = 
"art">Art</OPTION>
                                                            <OPTION VALUE = 
"religion">Religion</OPTION>
                                                            <OPTION VALUE = 
"medical">Medical</OPTION>
                                                  </SELECT></TD>
                                                  <TD><INPUT TYPE = "text" NAME = 
"description"></TD>
                                        </TR>
                                        <TR>
```

```
                                                        <TD><INPUT TYPE = "submit" VALUE =
"Submit" NAME = "titleS"></TD>
                                                        <TD><INPUT TYPE = "submit" VALUE =
"Submit" NAME = "isbnS"></TD>
                                                        <TD><INPUT TYPE = "submit" VALUE =
"Submit" NAME = "categoryS"></TD>
                                                        <TD><INPUT TYPE = "submit" VALUE =
"Submit" NAME = "descriptionS"></TD>
                                            </TR>
                                            </FORM>
                                  </TABLE>
eot1;

$query = "select * from books";
$return = mysql_query($query) or die(mysql_error());

while($result = mysql_fetch_array($return)){
print <<< eot4
                                  <TABLE>
                                  <TR><TD>ISBN:</TD><TD>$result[0]</TD></TR>
                                  <TR><TD>Rating:</TD><TD>$result[1]</TD></TR>
                                  <TR><TD>AUTHOR:</TD><TD>$result[2]</TD></TR>
                                  <TR><TD>TITLE:</TD><TD>$result[3]</TD></TR>
                                  <TR><TD>CATEGORY:</TD><TD>$result[4]</TD></TR>
                                  <TR><TD>DESCRIPTION:</TD><TD>$result[5]</TD></TR>
                                  <TR><TD>URL:</TD><TD>$result[6]</TD></TR>
                                  <TR><TD>PUBLISHER:</TD><TD>$result[7]</TD></TR>
                                  <TR><TD>STOCK:</TD><TD>$result[8]</TD></TR>
                                  <TD><TR><HR></TD></TR>
                                  </TABLE>
                       </CENTER>
           </BODY>
           </HTML>
eot4;
}
}

?>
```

20. Session:

```
<?php
include_once "global.php";

           session_start();             //start php session
           session_regenerate_id(true);                 //regenerate session and delete the old
session
```

```
?>
```

21. Signup:

```
<?php
include_once "global.php";
include_once "db_connect.php";
include "background.php";
//include "CSS/style.css";
//*********************************************receive sign up data
$name = isset($_GET["name"]) ? $_GET["name"]:null;
$gender = isset($_GET["gender"]) ? $_GET["gender"]:null;
$dateOfBirth = isset($_GET["dateOfBirth"]) ? $_GET["dateOfBirth"]:null;
$email = isset($_GET["email"]) ? $_GET["email"]:null;

$signupID = isset($_GET["signupID"]) ? $_GET["signupID"]:null;
$signupPassword = isset($_GET["signupPassword"]) ? $_GET["signupPassword"]:null;
$signup = isset($_GET["signup"]) ? $_GET["signup"]:null;

//***********************************************variable for address
$aptNumber = isset($_GET["aptNumber"]) ? $_GET["aptNumber"]:null;
$street = isset($_GET["street"]) ? $_GET["street"]:null;
$city = isset($_GET["city"]) ? $_GET["city"]:null;
$zipcode = isset($_GET["zipcode"]) ? $_GET["zipcode"]:null;

if($signup<>"")
{
        if(!($name<>"") || !($gender<>"") || !($dateOfBirth<>"") || !($email<>"")
|| !($signupID<>"") || !($aptNumber<>"") || !($street<>"") || !($city<>"") || !($zipcode<>""))
        {
                print "Invalid Input in sign up!!!<BR>";
        }else{
                //******************************************create a new user
account
                $query = "insert into member values('$email', '$signupPassword', '$gender',
'$dateOfBirth', 0 ,'$name', '$signupID')";
                $result = mysql_query($query) or die(mysql_error());
                $userFile = "userid.txt";
                file_put_contents($userFile, $signupID);
                //******************************************create as new address
                $query = "insert into address values
('$signupID','$aptNumber','$street','$city','$zipcode')";
                $result = mysql_query($query) or die(mysql_error());
                header("location:user_page.php");
        }
}
```

```
print <<< eot3
	<CENTER>
		<FORM METHOD = "GET" ACTION = "signup.php">
			<TABLE>
				<TH>PLEASE SIGN UP HERE </TH>
				<TR><TD>NAME:</TD><TD><INPUT TYPE = "text" 
NAME = "name" SIZE = "30"></TD></TR>
				<TR><TD>GENDER:</TD><TD>
				<INPUT TYPE = "radio" NAME = "gender" VALUE = 
"male">Male<BR>
				<INPUT TYPE = "radio" NAME = "gender" VALUE = 
"female">Female</TD></TR>
				<TR><TD>DATEOFFBIRTH:</TD><TD><INPUT TYPE 
= "date" NAME = "dateOfBirth"></TD></TR>
				<TR><TD>EMAIL:</TD><TD><INPUT TYPE = "text" 
NAME = "email" SIZE = "50"></TD></TR>
				<TR><TD>USERID:</TD><TD><INPUT TYPE = "text" 
NAME = "signupID" SIZE = "30"></TD></TR>
				<TR><TD>PASSWORD:</TD><TD><INPUT TYPE = 
"text" NAME = "signupPassword" SIZE = "30"></TD></TR>
				<TR><TD><BR>Enter your address</TD></TR>
				<TR><TD>APARTMENT #:</TD><TD><INPUT TYPE 
= "text" NAME = "aptNumber" SIZE = "30"></TD></TR>
				<TR><TD>STREET:</TD><TD><INPUT TYPE = "text" 
NAME = "street" SIZE = "30"></TD></TR>
				<TR><TD>CITY:</TD><TD><INPUT TYPE = "text" 
NAME = "city" SIZE = "30"></TD></TR>
				<TR><TD>ZIPCODE:</TD><TD><INPUT TYPE = "text" 
NAME = "zipcode" SIZE = "30"></TD></TR>

			</TABLE>
		<INPUT TYPE = "submit" NAME = "signup" VALUE = "Submit">
		</FORM>
	</CENTER>
eot3;
?>
```

22. telemarket_header:

```
<?php

//**********************************read adminid from file created in the top page
$file = "telemarketid.txt";
$userid = file_get_contents($file);

//******************************************************
print <<< eot2
```

```
	<HTML>
	<HEAD>
		<link rel="stylesheet" type="text/css" href="style.css"/>
		<TITLE>MY MONGO MARIA</TITLE>
	</HEAD>
	<BODY>
		<BODY BGCOLOR = "AliceBlue">
		<CENTER>
			<IMG SRC = "image/logo.png">
			<BR><BR>
		</CENTER>
	</BODY>
	</HTML>
eot2;

?>
```

23. Telemarket_page:

```
<?php
include_once "global.php";
include_once "db_connect.php";
include_once "telemarket_header.php";

$query = "select * from member order by reserveCount desc";
$return = mysql_query($query) or die(mysql_error());
while($result = mysql_fetch_array($return)){
print <<< eot

	<CENTER>
		<TABLE>
			<TR><TD>NAME:</TD><TD>$result[6]</TD></TR>
			<TR><TD>GENDER:</TD><TD>$result[2]</TD></TR>
			<TR><TD>DATE OF BIRTH:</TD><TD>$result[3]</TD></TR>
			<TR><TD>EMAIL:</TD><TD>$result[0]</TD></TR>
			<TR><TD>ACTIVITY
COUNT:</TD><TD>$result[4]</TD></TR>
			<TD><TR><HR></TD></TR>
		</TABLE>
	</CENTER>

eot;
}

?>
```

24. Test:

```
<?php


include_once "global.php";
include_once "db_connect.php";
include "header.php";
print <<< eot
            <span class="rating">
               <input id="rating10" type="radio" name="rating" value="10">
               <label for="rating10">10</label>
               <input id="rating9" type="radio" name="rating" value="9">
               <label for="rating9">9</label>
               <input id="rating8" type="radio" name="rating" value="8">
               <label for="rating8">8</label>
               <input id="rating7" type="radio" name="rating" value="7">
               <label for="rating7">7</label>
               <input id="rating6" type="radio" name="rating" value="6">
               <label for="rating6">6</label>
               <input id="rating5" type="radio" name="rating" value="5">
               <label for="rating5">5</label>
               <input id="rating4" type="radio" name="rating" value="4">
               <label for="rating4">4</label>
               <input id="rating3" type="radio" name="rating" value="3">
               <label for="rating3">3</label>
               <input id="rating2" type="radio" name="rating" value="2">
               <label for="rating2">2</label>
               <input id="rating1" type="radio" name="rating" value="1">
               <label for="rating1">1</label>
            </span>
            <BR>
            <form name="testform" method="post" action="test.php">

       <h2>Choose a category</h2>
       <p>
             <input type="radio" value="1" name="category" id="category_1" class="hidden"
/>
             <a id="category_fruit" class="radio-picture">&nbsp;</a>
            </p>
            </form>
eot;
?>
```

25. Top:

```
<?php
```

```
//*************************read info for db****************************//
//require_once("library/login_info.php");
//$session = mysql_connect($server, $user, $password) or die(mysql_error());
//print "CONNECTED";
//mysql_select_db("lib_system");
include_once "global.php";
include_once "db_connect.php";
include "background.php";


//*****************************************delete old userid.txt or adminid.txt
$deleteUser = "userid.txt";
$deleteAdmin = "adminid.txt";
$deliveryAdmin = "deliveryid.txt";
$telemarketAdmin = "telemarketid.txt";
@unlink($deleteUser);
@unlink($deleteAdmin);
@unlink($deliveryAdmin);
@unlink($telemarketAdmin);
//*************************get client's IP****************************//
//$ip = getenv("REMOTE_ADDR");
//********************************************************************//
//********************************************receive log in data
$loginType = isset($_GET["loginType"]) ? $_GET["loginType"]:null;
$id = isset($_GET["id"]) ? $_GET["id"]:null;
$password = isset($_GET["password"]) ? $_GET["password"]:null;
$login = isset($_GET["login"]) ? $_GET["login"]:null;
$signup = isset($_GET["signup"]) ? $_GET["signup"]:null;

//$tName = isset($_GET["sName"]) ? htmlspecialchars($_GET["sName"]):null;
if(($login<>""))
{
        if((!($loginType<>"") || !($id<>"") || !($password<>"")))
        {
                print "Invalid Input in Log in<BR>";
                //header("location:top.php");
        }else if($loginType == "user"){
                $query = "select * from member where memberid = '$id'";
                $result = mysql_query($query) or die(mysql_error());
                $num_row = mysql_num_rows($result);
                if($num_row > 0){
                        $query = "select * from member where password = '$password'";
                        $result = mysql_query($query) or die(mysql_error());
                        $num_row = mysql_num_rows($result);
                        if($num_row > 0){
                                //**********************************user
successfully loged in
```

```
                print "User Loged in<BR>";
                $userFile = "userid.txt";
                file_put_contents($userFile, $id);
                header("location:user_page.php ? userid = $id");
                }else{
                //*********************************user fail to log
in
                print "User failed to log in<BR>";
            }
        }else{
            //*********************************user fail to log in
            print "User failed to log in<BR>";
        }
    }else if($loginType == "admin"){
        $query = "select * from admin where adminid = '$id'";
        $result = mysql_query($query) or die(mysql_error());
        $num_row = mysql_num_rows($result);
        if($num_row > 0)
        {
            $query = "select * from admin where password = '$password'";
            $result = mysql_query($query) or die(mysql_error());
            $num_row = mysql_num_rows($result);
            if($num_row > 0){
                //*********************************admin
successfully loged in
                print "Admin Loged in<BR>";
                $adminFile = "adminid.txt";
                file_put_contents($adminFile, $id);
                header("location:admin_page.php ? adminid = $id");
                }else{
                //*********************************admin fail to log
in
                print "Admin failed to log in<BR>";
            }
            //*********************************admin successfully
loged in

        }else{
            //*********************************admin fail to log in
            print "Admin failed to log in<BR>";
        }
    }else if($loginType == "delivery"){
        $query = "select * from delivery where deliveryid = '$id'";
        $result = mysql_query($query) or die(mysql_error());
        $num_row = mysql_num_rows($result);
        if($num_row > 0)
        {
            $query = "select * from delivery where password = '$password'";
            $result = mysql_query($query) or die(mysql_error());
            $num_row = mysql_num_rows($result);
            if($num_row > 0){
                //*********************************delivery
successfully loged in
                print "Delivery staff Loged in<BR>";
                $deliveryFile = "deliveryid.txt";
                file_put_contents($deliveryFile, $id);
                header("location:delivery_page.php ? deliveryid = $id");
            }else{
                //*********************************delivery fail to
log in
                print "Delivery staff failed to log in<BR>";
            }
        }else{
            //*********************************delivery fail to log in
            print "Delivery staff failed to log in<BR>";
        }
    }else if($loginType == "telemarket"){
        $query = "select * from telemarket where telemarketid = '$id'";
        $result = mysql_query($query) or die(mysql_error());
        $num_row = mysql_num_rows($result);
        if($num_row > 0)
        {
            $query = "select * from delivery where password = '$password'";
            $result = mysql_query($query) or die(mysql_error());
            $num_row = mysql_num_rows($result);
            if($num_row > 0){
                //*********************************telemarket
successfully loged in
                print "Telemarket staff Loged in<BR>";
                $telemarketFile = "telemarketid.txt";
                file_put_contents($telemarketFile, $id);
                header("location:telemarket_page.php ? telemarketid =
$id");
            }else{
                //*********************************telemarket fail
to log in
                print "Telemarket staff failed to log in<BR>";
            }
        }else{
            //*********************************telemarket fail to log in
            print "Telemarket staff failed to log in<BR>";
        }
    }
}else if($signup<>"")
```

```
{
    header("location:signup.php");
}
mysql_close($session);

//*******************Input Thread Name, Link to top******************//

print <<< eot3
            <CENTER>
                <!-- user log in-->
                <STYLE>
                    TABLE, TH, TD{border: solid black;}
                </STYLE>
                <TABLE>
                    <TR>
                    <TD>
                    <FONT SIZE = "5">
                        PLEASE LOG IN HERE
                    </FONT>
                    </TD>
                    <TD>
                    <FONT SIZE = "5">
                        USER SIGN UP<BR>
                    </FONT>
                    </TD>
                    </TR>
                    <TR>
                    <TD>
                        <FORM METHOD = "GET" ACTION =
"top.php"><BR>
                        <INPUT TYPE = "radio" NAME =
"loginType" VALUE = "user">User<BR>
                        <INPUT TYPE = "radio" NAME =
"loginType" VALUE = "admin">Admin<BR>
                        <INPUT TYPE = "radio" NAME =
"loginType" VALUE = "delivery">Delivery<BR>
                        <INPUT TYPE = "radio" NAME =
"loginType" VALUE = "telemarket">Telemarket<BR>
                        ID:<BR><INPUT TYPE = "text" NAME =
"id" SIZE = "30"><BR>
                        PASSWORD:<BR><INPUT TYPE =
"text" NAME = "password" SIZE = "30"><BR>
                        <INPUT TYPE = "submit" NAME =
"login" VALUE = "log in">
                    </TD>
                    <TD>
                        <INPUT TYPE = "submit" NAME =
"signup" VALUE = "sign up"><BR>
                                                        </FORM>
                                                </TD>
                                                </TR>
                                        </TABLE>
                                </CENTER>
                        </BODY>
                </HTML>
eot3;

?>
```

26. User_page:

```
<?php
include_once "global.php";
include_once "db_connect.php";
include "header.php";
//**************************************************

//**************************************************show library information
print <<< eot1
        <HTML>
        <HEAD>
        <STYLE>
                table{
                border: 5px solid #a00000;
                padding: 5px;
                position: absolute;
                left: 20px;
        }
        </STYLE>
        </HEAD>
        <BODY>
                <TABLE>
                        <TR>
                                <TH>ADDRESS</TH>
                        </TR>
                                <TR><TD>Pattee Library Paterno Library,</TD></TR>
                                <TR><TD>510 Patterno Library,</TD></TR>
                                <TR><TD>University Park,</TD></TR>
                                <TR><TD>PA 16802</TD></TR>
                                <BR>
                                <TR><TD>TEL: 814-865-3616</TD></TR>
                                <TR><TD>FAX: 814-863-7502</TD></TR>
                </TABLE>
                <CENTER>
```

```
                    <IMG SRC = "image/pattee.png">
            </CENTER>
      </BODY>
      </HTML>

eot1;


?>
```