DBMS MINI-PROJECT

TEAM:

ANAY VOLDIE JAIN	- 1PI13CS026
DEEPAK MAHENDRAKAR	- 1PI13CS053
HARSHITH G	- 1PI13CS065

TOPIC: BOOK EXCHANGE SYSTEM DATABASE

DESCRIPTION:

- A book exchange system facilitates the exchange of books between students of a university.
- Each user will have an account, it contains list of books to be lended. Borrower can find a lender/book and request for borrow.
- The lender will submit the book to the system, the borrower should pay 25% of the price of book as deposit while borrowing. This initiates a transaction.
- The duration of borrowal is decided between lender-borrower.
- If the borrower returns the book in stipulated time, 20% of deposit will be returned, book will be returned to Borrower.
- Borrower will be fined(deducted from initial deposit) on delay while returning the book.
- If the book is lost/damaged the system will reimburse the cost for a new book.
- Regular users will be awarded points for each successful transaction, which can later be exchanged with coupons or reduction in interest.

QUERIES:

1. Account creation
2. Searching for book based on-
    a. User name(fname,lname)
    b. Department
    c. Semester
    d. Subject
3. Transaction Processing
4. Notification generation
5. Points system
6. Book reservation

FRONT END: HTML,javascript,css
S/W TOOLS : mysql, php

SCHEMA:

USER

usn	fname	lname	section	department	semester	phone_no	email_id	points

BOOK

id	title	publisher	edition	**subject_id**	price	author

SUBJECT

id	title	semester	department

TRANSACTION

id	**lender_id**	**lendee_id**	**book_id**	start_date	end_date	status	deposit

BOOK_OWNER

book_id	**owner_id**	availability

RESERVATION

requester_id	**requestee_id**	**book_id**	seq_no

SQL COMMANDS:

```
CREATE TABLE USER(
id bigint not null auto_increment,
fname varchar(50) not null,
lname varchar(50) not null,
section char,
department varchar(50),
semester int,
phone_no varchar(10) not null,
email_id varchar(50) not null,
points int default 0,

constraint user_pk
  primary key(id)
);
```

```
CREATE TABLE SUBJECT(
id bigint not null auto_increment,
title varchar(100) not null,
semester smallint,
department varchar(50),


constraint subject_pk
  primary key(id)
);
```

```
CREATE TABLE BOOK(
id bigint not null auto_increment,
title varchar(100) not null,
publisher varchar(50),
subject_id bigint not null,
price decimal(6,2) not null,
author varchar(100),
edition int,
constraint book_pk
  primary key(id),
constraint book_subject_fk
  foreign key(id) references SUBJECT(id)
);


CREATE TABLE TRANSACTION(
id bigint not null auto_increment,
lender_id bigint not null,
lendee_id bigint not null,
book_id bigint not null,
start_date varchar(10) not null,
end_date varchar(10) not null,
status int default 0,
deposit real not null,
fine real default 0,
return_date varchar(10),
duration int,


constraint transaction_pk
  primary key(id),
constraint transaction_lender_fk
  foreign key(lender_id) references USER(id),
constraint transaction_lendee_fk
  foreign key(lendee_id) references USER(id),
constraint transaction_book_fk
  foreign key(book_id) references BOOK(id)
);


CREATE TABLE BOOK_OWNER(
id bigint not null auto_increment,
book_id bigint not null,
owner_id bigint not null,
availability tinyint default 1,
constraint book_owner_pk
  primary key(id),
constraint book_owner_book_fk
  foreign key(book_id) references BOOK(id),
constraint book_owner_owner_fk
  foreign key(owner_id) references USER(id)
);



CREATE TABLE RESERVATION(
id bigint not null auto_increment,
requester_id bigint not null,
requestee_id bigint not null,
book_id bigint not null,
sequence_no int default 1,


constraint reservation_pk
  primary key(id),
constraint reservation_requester_fk
  foreign key(requester_id) references USER(id),
constraint reservation_requestee_fk
  foreign key(requestee_id) references USER(id),
constraint reservation_book_fk
  foreign key(book_id) references BOOK(id)
);

CREATE TABLE USER_PASSWORD(
user_id bigint not null,
password varchar(32),
email varchar(100),

constraint user_password_pk
	primary key(user_id),
constraint user_password_user_id_fk
	foreign key(user_id) references USER(id)
);
```

```
CREATE TABLE NOTIFICATION(
    id bigint not null auto_increment,
    seen tinyint default 0,
    message varchar(1000),
    uid bigint,
    tid bigint,
    type smallint,
    constraint pk primary key(id),
    constraint fk1 foreign key(tid) references TRANSACTION(id),
    constraint fk2 foreign key(uid) references USER(id)
    );
```

QUERIES:

Queries for this application is not intensive. But there are lot of interrelated queries for a particular action. The queries can be found in the php files.