LIBRARY MANAGEMENT SYSTEM

TEAM – USA

BATCH - 1

UJJAYANT PRAKASH	190911084
SIDHARTH SAHOO	190911024
APALA BHATT	190911047

SECTION	A
BRANCH	INFORMATION TECHNOLOGY

GUIDED BY:- DR. GIRIJA ATTIGERI & DR . SUMITH N

INTRODUCTION

A library management system is a project that stores and manages book information and its borrow status electronically. It allows both student and admins to search for a desired book and keeps a continuous check weather a book is issued or returned or even late fine in some cases. This system reduces manual work to a great extent and allows a smooth flow of library activities by removing any chances of errors in the details

PROBLEM DEFINITION & SCOPE

We need to make a library management system that allows a member to login and issue or return books and magazines, browse through a catalogue and take an E-pad. This system also stores information about the members and the available books and magazines. It will also keep track of the issue and return status.

There is a big future scope of this application that a lot more elements, for example, online classes, video tutorials, assignments and submissions facility can be added by teachers or services such as online group chats and question forums can be created by students thus making it more interactive and user-friendly.

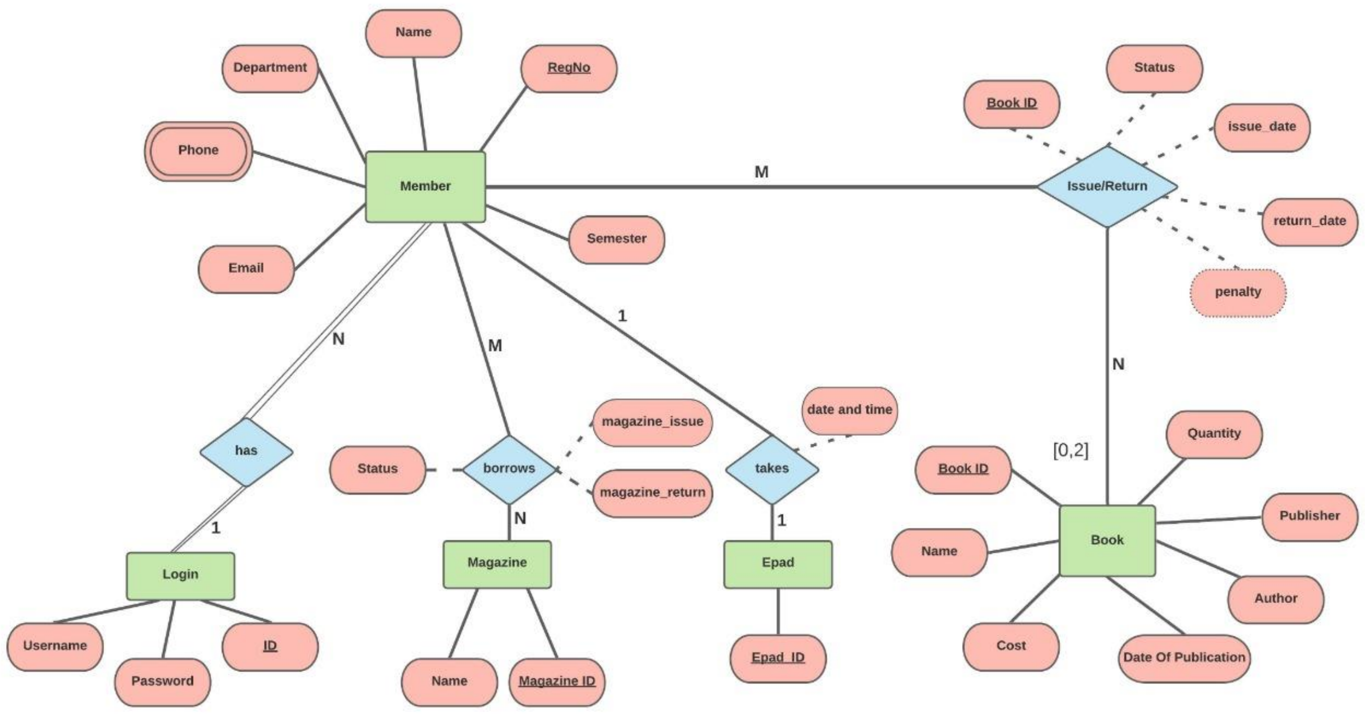
SOFTWARE AND HARDWARE REQUIREMENTS

Softwares that were used in this project are Visual Studio for the user interface, MySQL for creating the database and Lucid charts for making the ER diagram, and a Windows OS. Hardware requirements would be a PC, mouse, keyboard, monitor etc.

DESIGN AND METHODOLOGY

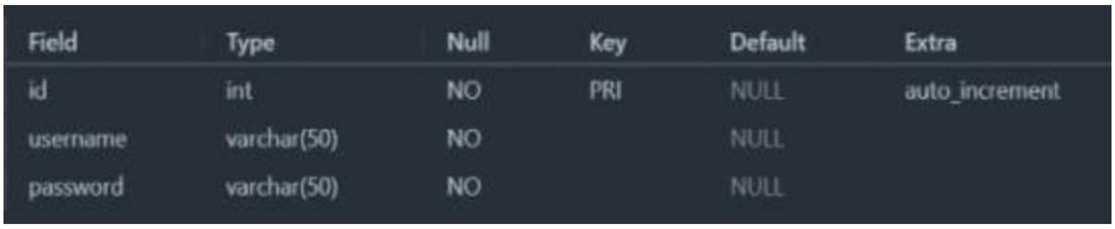
A multi-form windows form application must be developed that has a login page and a landing page that allows a user to browse, issue, return, add books or add students depending on the privilege of the user (either student or librarian). This application must be connected to MySQL server where we would create tables from the schema after normalization.

ER DIAGRAM



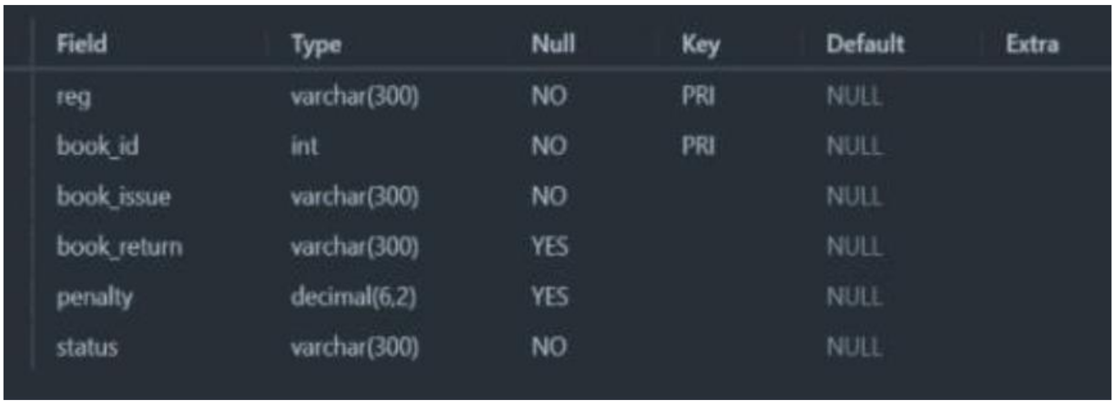
SCHEMA DIAGRAM

Login



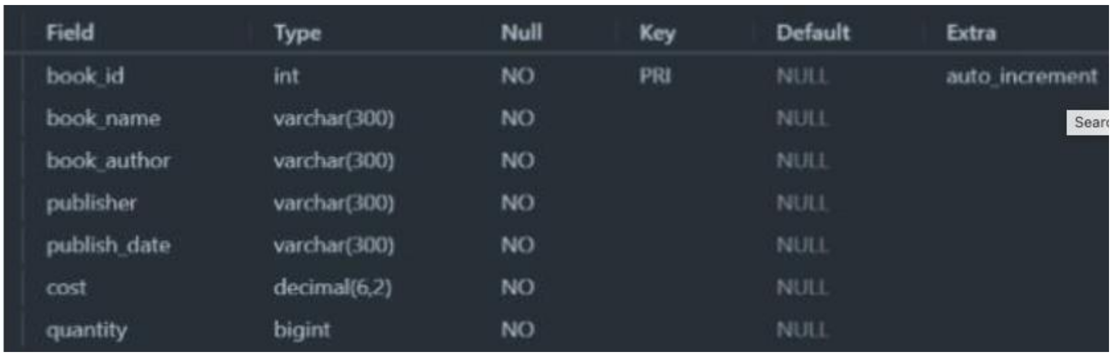
Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
username	varchar(50)	NO		NULL	
password	varchar(50)	NO		NULL	

Issue/Return



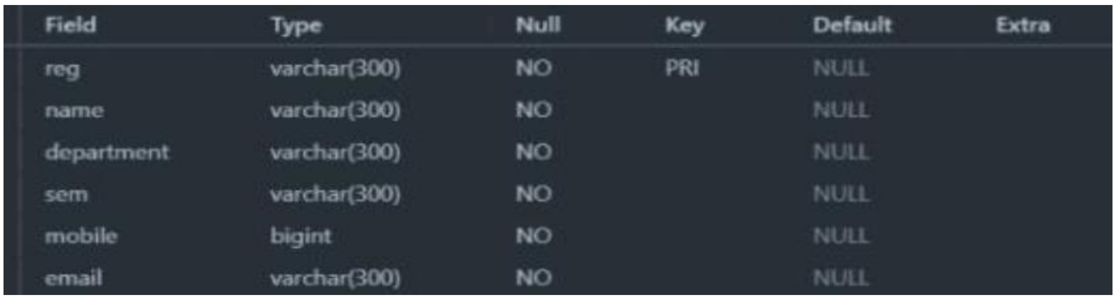
Field	Type	Null	Key	Default	Extra
reg	varchar(300)	NO	PRI	NULL	
book_id	int	NO	PRI	NULL	
book_issue	varchar(300)	NO		NULL	
book_return	varchar(300)	YES		NULL	
penalty	decimal(6,2)	YES		NULL	
status	varchar(300)	NO		NULL	

Book



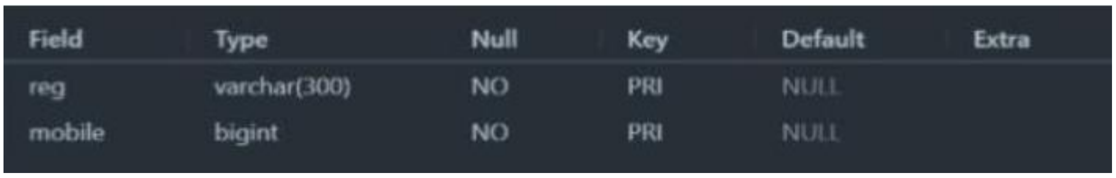
Field	Type	Null	Key	Default	Extra
book_id	int	NO	PRI	NULL	auto_increment
book_name	varchar(300)	NO		NULL	
book_author	varchar(300)	NO		NULL	
publisher	varchar(300)	NO		NULL	
publish_date	varchar(300)	NO		NULL	
cost	decimal(6,2)	NO		NULL	
quantity	bigint	NO		NULL	

Member



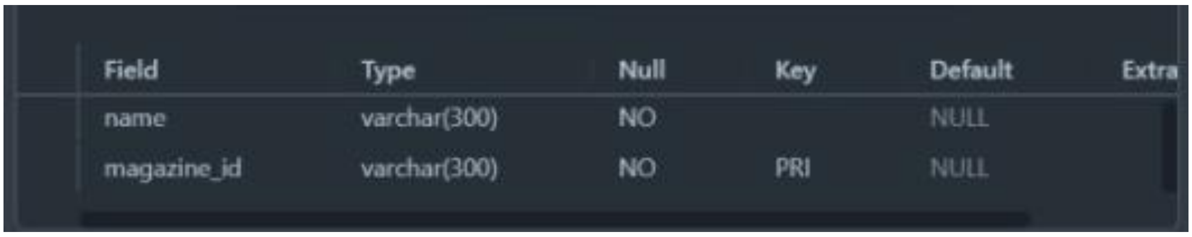
Field	Type	Null	Key	Default	Extra
reg	varchar(300)	NO	PRI	NULL	
name	varchar(300)	NO		NULL	
department	varchar(300)	NO		NULL	
sem	varchar(300)	NO		NULL	
mobile	bigint	NO		NULL	
email	varchar(300)	NO		NULL	

Phone/Mobile attribute



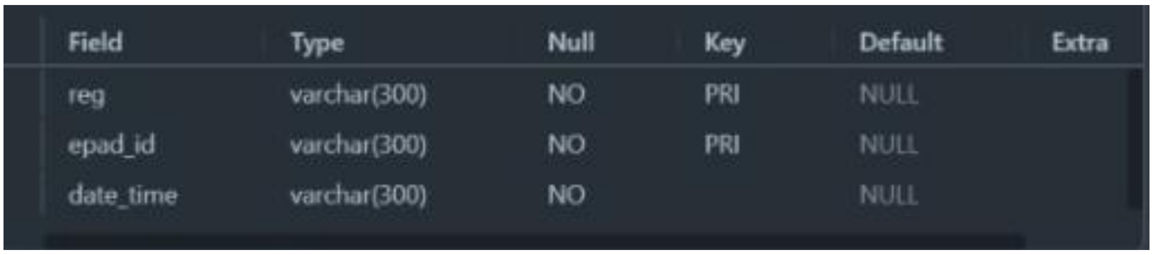
Field	Type	Null	Key	Default	Extra
reg	varchar(300)	NO	PRI	NULL	
mobile	bigint	NO	PRI	NULL	

Magazine



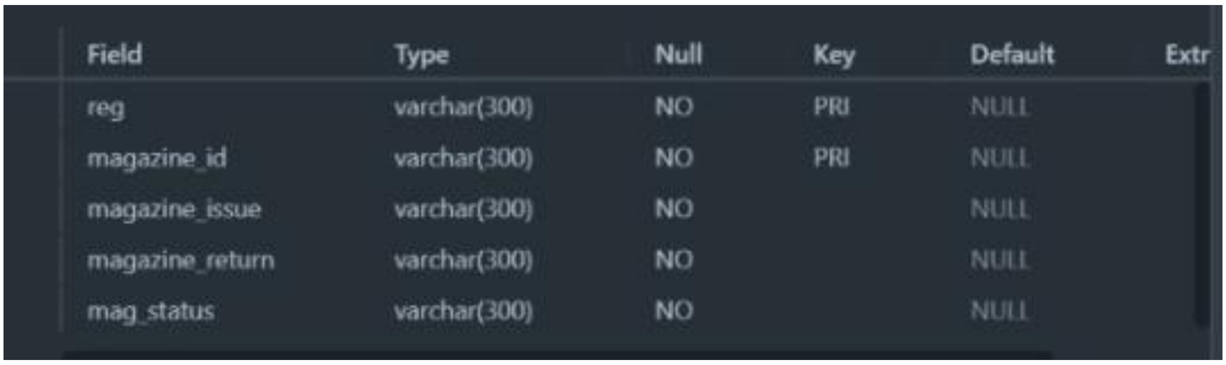
Field	Type	Null	Key	Default	Extra
name	varchar(300)	NO		NULL	
magazine_id	varchar(300)	NO	PRI	NULL	

EPAD



Field	Type	Null	Key	Default	Extra
reg	varchar(300)	NO	PRI	NULL	
epad_id	varchar(300)	NO	PRI	NULL	
date_time	varchar(300)	NO		NULL	

Borrows (Magazine)



Field	Type	Null	Key	Default	Extr
reg	varchar(300)	NO	PRI	NULL	
magazine_id	varchar(300)	NO	PRI	NULL	
magazine_issue	varchar(300)	NO		NULL	
magazine_return	varchar(300)	NO		NULL	
mag_status	varchar(300)	NO		NULL	

ER TO SCHEMA CONVERSION

CONVERT ALL STRONG ENTITY SETS INTO RELATIONS

- only constituent simple attributes will be mapped in case of composite attribute.
- exclude multivalued attribute from mapping into tables /relations.

MAPPING 1:1 RELATIONSHIP SETS

- *Foreign key approach*
  Let R&S be 2 entity sets
  1. identify the entity sets with total participation(s).
  2. add primary key of R into S as foreign key.
- *Merged relation approach*
  1. if both entity sets are having total participation, then they can be merged into single relation.
- *Crossed reference approach*
  1. create a third relation comprising primary key of both entity sets.

MAPPING M : N RELATIONSHIP SETS

- create a third relation containing the primary keys of both entity sets and descriptive attribute (if any).
- Descriptive attribute: the attribute used for describing the relationship.

MAPPING MULTIVALUED ATTRIBUTES

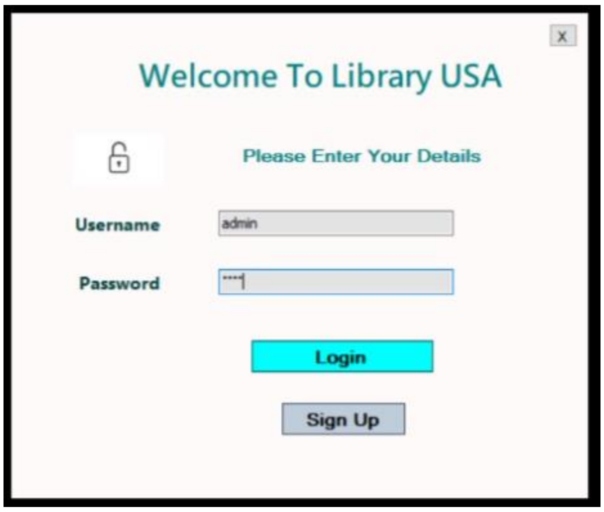
- For each multivalued attributes create a separate relation.
- Add primary key of the entity set in new relation as foreign key.
- The foreign key attribute and multivalued attribute will become composite key.

NORMALIZATION

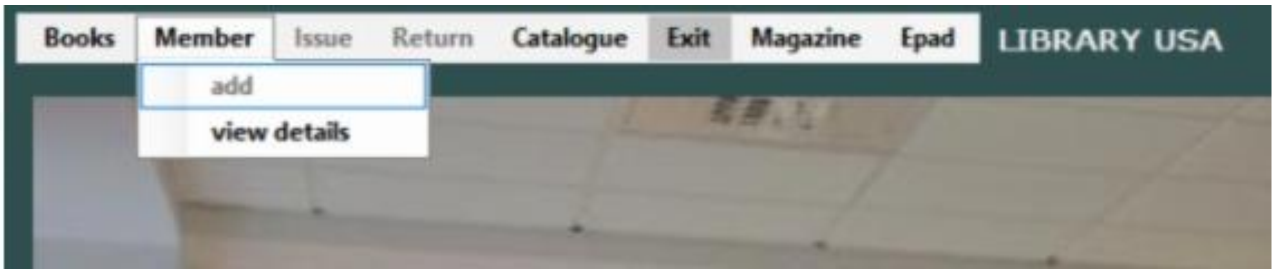
All tables are in the 2nd Normal Form, that means there is no partial dependency and only single valued attributes.

USER INTERFACE

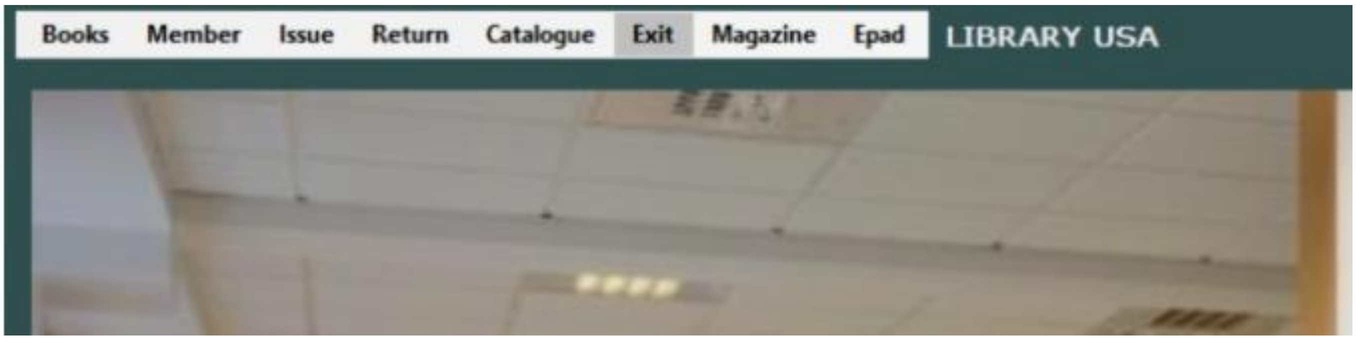
The UI starts with a simple login page



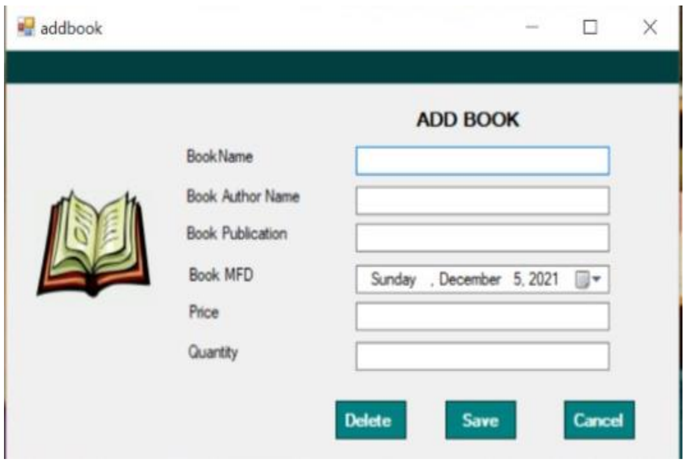
On logging with either 'student' or 'admin', different privileges are granted. A student cannot add books, issue books or return books meanwhile an admin can. This is how the landing page is like for a student:



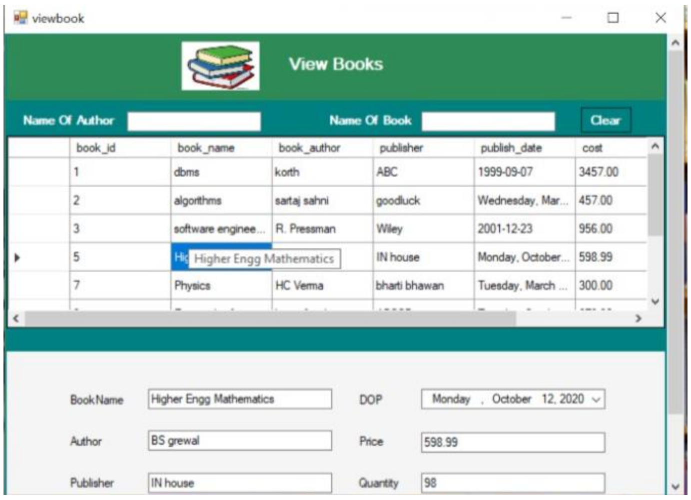
Meanwhile for an admin,



There are a total of 11 forms: login page, landing page, catalogue, E-pad, view book, view student, borrow magazine and view magazine which are accessible to all users plus issue book, add book and add student which are only accessible to admin.

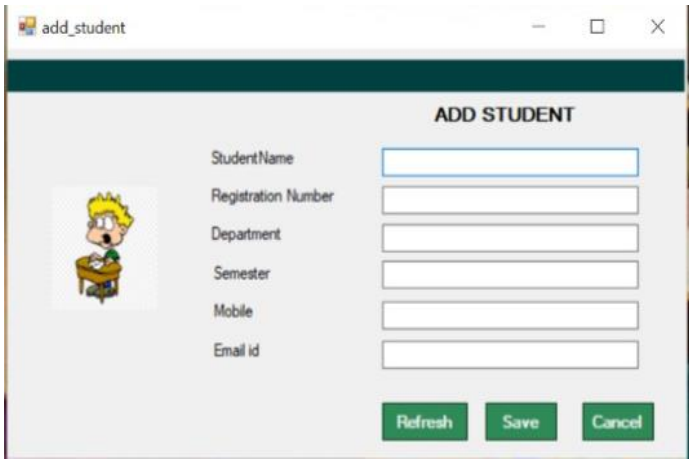


Book Name	
Book Author Name	
Book Publication	
Book MFD	Sunday , December 5, 2021
Price	
Quantity	

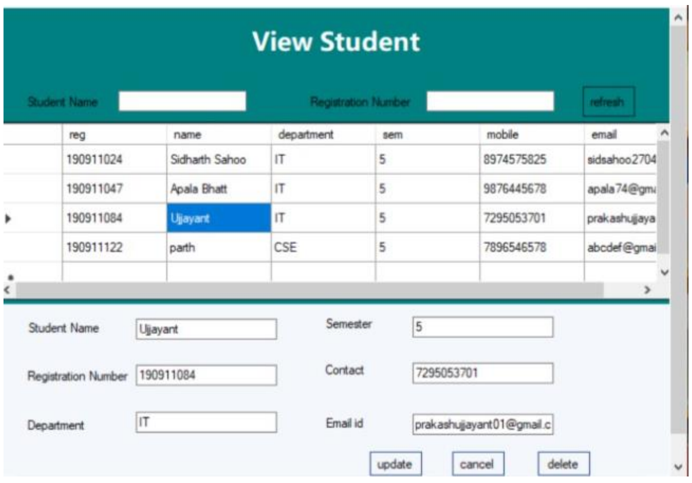


	book_id	book_name	book_author	publisher	publish_date	cost
	1	dbms	korth	ABC	1999-09-07	3457.00
	2	algorithms	sartaj sahni	goodluck	Wednesday, Mar...	457.00
	3	software enginee...	R. Pressman	Wiley	2001-12-23	956.00
▶	5	Higher Engg Mathematics		IN house	Monday, October...	598.99
	7	Physics	HC Verma	bharti bhawan	Tuesday, March ...	300.00

Book Name	Higher Engg Mathematics	DOP	Monday , October 12, 2020
Author	BS grewal	Price	598.99
Publisher	IN house	Quantity	98

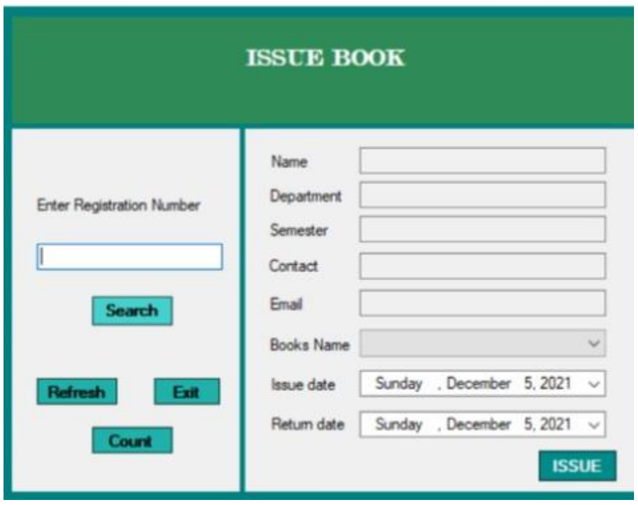


Student Name	
Registration Number	
Department	
Semester	
Mobile	
Email id	

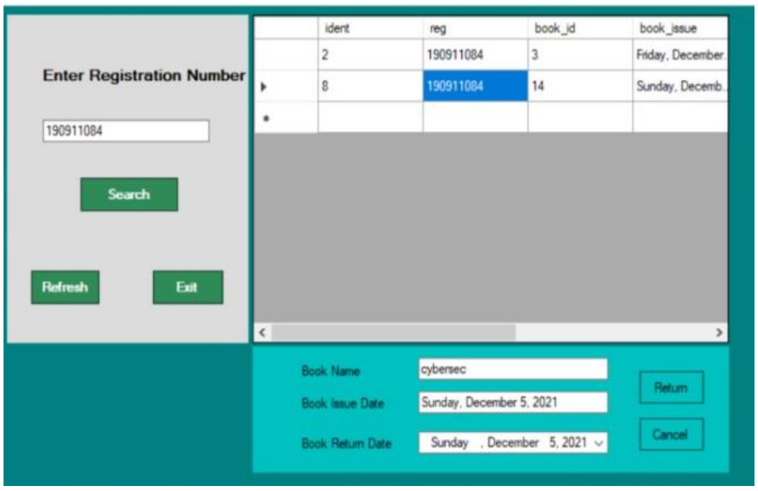


	reg	name	department	sem	mobile	email
	190911024	Sidharth Sahoo	IT	5	8974575825	sidsahoo2704
	190911047	Apala Bhatt	IT	5	9876445678	apala74@gm
▶	190911084	Ujayant	IT	5	7295053701	prakashujaya
	190911122	parth	CSE	5	7896546578	abcdef@gmai

Student Name	Ujayant	Semester	5
Registration Number	190911084	Contact	7295053701
Department	IT	Email id	prakashujayant01@gmail.c



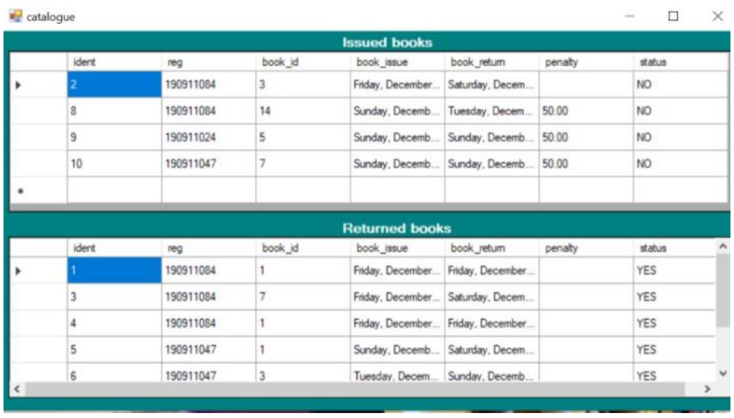
Name	
Department	
Semester	
Contact	
Email	
Books Name	
Issue date	Sunday , December 5, 2021
Return date	Sunday , December 5, 2021



Enter Registration Number: 190911084

	ident	reg	book_id	book_issue
	2	190911084	3	Friday, December
▶	8	190911084	14	Sunday, Decemb

Book Name	cybersec
Book Issue Date	Sunday, December 5, 2021
Book Return Date	Sunday , December 5, 2021



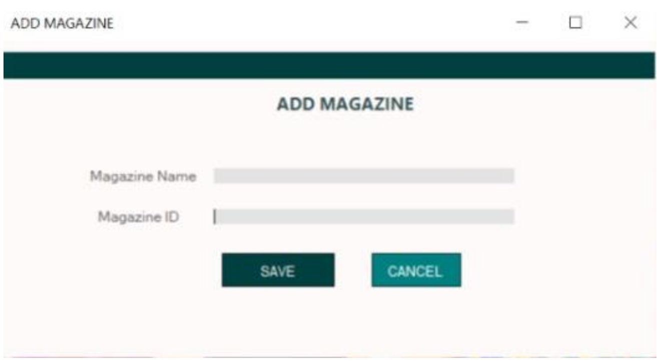
catalogue

Issued books

ident	reg	book_id	book_issue	book_return	penalty	status
2	190911084	3	Friday, December...	Saturday, Decem...		NO
8	190911084	14	Sunday, Decemb...	Tuesday, Decem...	50.00	NO
9	190911024	5	Sunday, Decemb...	Sunday, Decemb...	50.00	NO
10	190911047	7	Sunday, Decemb...	Sunday, Decemb...	50.00	NO

Returned books

ident	reg	book_id	book_issue	book_return	penalty	status
1	190911084	1	Friday, December...	Friday, December...		YES
3	190911084	7	Friday, December...	Saturday, Decem...		YES
4	190911084	1	Friday, December...	Friday, December...		YES
5	190911047	1	Sunday, Decemb...	Saturday, Decem...		YES
6	190911047	3	Tuesday, Decem...	Sunday, Decemb...		YES



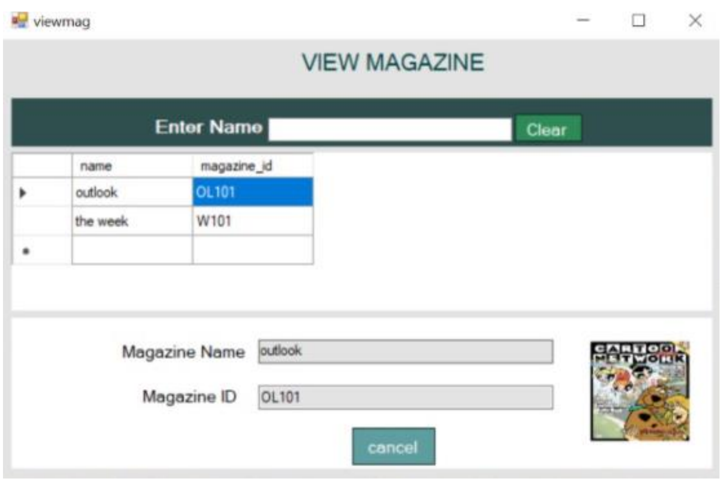
ADD MAGAZINE

ADD MAGAZINE

Magazine Name

Magazine ID

SAVE CANCEL



viewmag

VIEW MAGAZINE

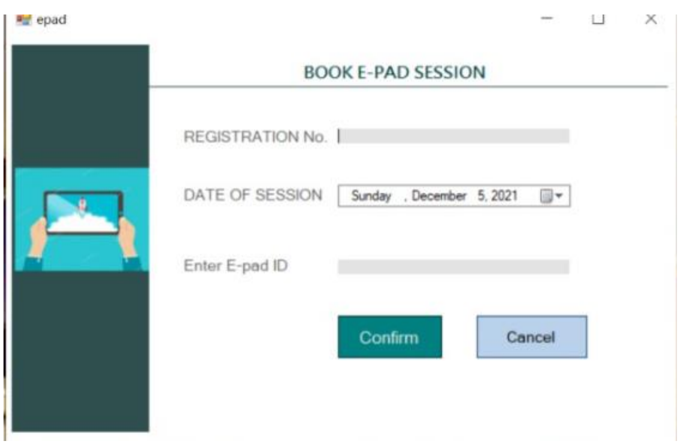
Enter Name Clear

name	magazine_id
outlook	OL101
the week	W101

Magazine Name: outlook

Magazine ID: OL101

cancel



epad

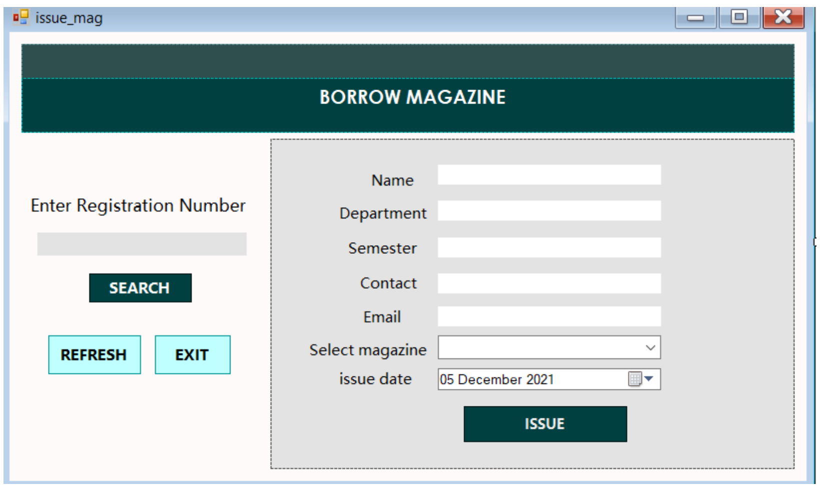
BOOK E-PAD SESSION

REGISTRATION No.

DATE OF SESSION: Sunday , December 5, 2021

Enter E-pad ID

Confirm Cancel



issue_mag

BORROW MAGAZINE

Enter Registration Number

SEARCH

REFRESH EXIT

Name

Department

Semester

Contact

Email

Select magazine

issue date: 05 December 2021

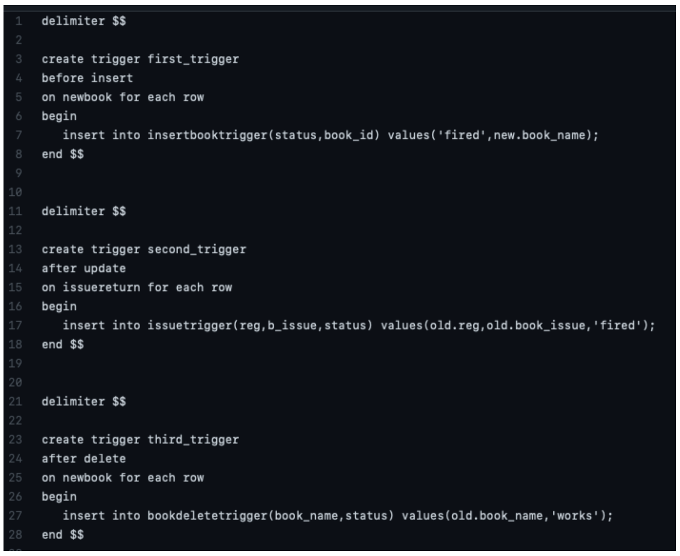
ISSUE

The view book, view student, view magazine, return book and catalogue have a <u>data grid</u> which displays the data by making a connection between C# and MySQL. Other important tools used besides <u>labels</u>, <u>buttons</u>, <u>panels</u> and <u>textbox</u> is <u>date time picker</u> and <u>combo box</u> on add book/issue book plus a <u>menu strip</u> and <u>picture box</u> in the landing page.

PL/SQL

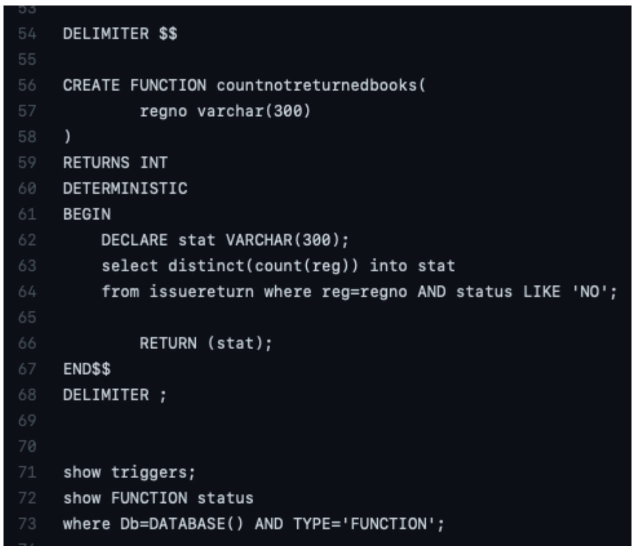
Three triggers, a procedure and a function were implemented in this system.

First trigger is fired when a new book is inserted which then inserts a tuple into the book table, second trigger is fired on updating the issue of any student, and the final trigger deletes information of a book.

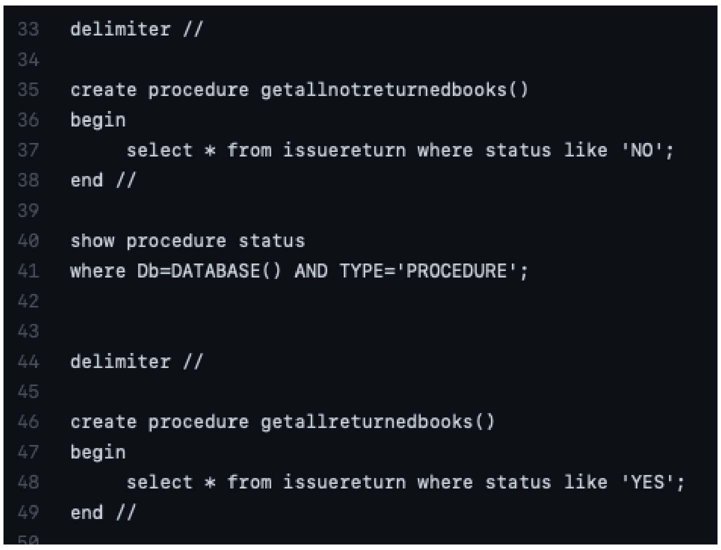


```
delimiter $$

create trigger first_trigger
before insert
on newbook for each row
begin
   insert into insertbooktrigger(status,book_id) values('fired',new.book_name);
end $$


delimiter $$

create trigger second_trigger
after update
on issuereturn for each row
begin
   insert into issuetrigger(reg,b_issue,status) values(old.reg,old.book_issue,'fired');
end $$


delimiter $$

create trigger third_trigger
after delete
on newbook for each row
begin
   insert into bookdeletetrigger(book_name,status) values(old.book_name,'works');
end $$
```

The procedure is responsible for producing all the books that have not been returned for a person and the function produces a count of all the not returned books.



```
DELIMITER $$

CREATE FUNCTION countnotreturnedbooks(
        regno varchar(300)
)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE stat VARCHAR(300);
    select distinct(count(reg)) into stat
    from issuereturn where reg=regno AND status LIKE 'NO';

        RETURN (stat);
END$$
DELIMITER ;


show triggers;
show FUNCTION status
where Db=DATABASE() AND TYPE='FUNCTION';
```



```
delimiter //

create procedure getallnotreturnedbooks()
begin
     select * from issuereturn where status like 'NO';
end //

show procedure status
where Db=DATABASE() AND TYPE='PROCEDURE';


delimiter //

create procedure getallreturnedbooks()
begin
     select * from issuereturn where status like 'YES';
end //
```

RESULTS

A highly efficient working application compatible with Windows was created that can perform all the user requirements. This component can be plugged onto many other systems.

CONCLUSION

This library management system allows the librarian to manage the data in a much more efficient way that saves their time and manual effort. The process of allotment and maintaining dues has also become much easier. Besides the librarian, this system is also useful for the student as it can quickly search for books and check availability.