Information Systems and Data Modeling – IT1090 **Assignment**

Title: (Library Management System	
Batch Number: (**Y1.S2. WD 01.01**)	Group Number: (**MLB_01.01_12**)

Declaration:

We hold a copy of this assignment that we can produce if the original is lost or damaged.

We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.

Group Members:

	Student Name	**Student Registration Number**	**Date**	**Signature**
1	**IT21826368**	**Nanayakkara Y.D.T.D**	30/10/2022	
2	**IT21820946**	Weragala R.T.L	30/10/2022	
3	**IT21822612**	Mendis H.R.M	30/10/2022	
4	**IT21823220**	Vithanage H.P	30/10/2022	

Submitted on: <30/10/2022>

Contents

Introduction

A Library Management system is the management system that is used to assist administration to manage library in an accurate manner. It can ease the tasks faced daily. When designing a system, we need to identify basic requirements, its components and working behavior. So, the aim of this project was to design and develop a Library Management System. In this report we have included requirement analysis, data requirements, ER-diagrams, performance requirements, Schema of database, relational database and SQL queries.

Hypothetical Scenario

Library Management System is where guests can preview books, search trending books and sign up to the system to become a member of the library. Members should renew their membership once a year. They can sign up as a basic member or a premium member. The difference between them is basic members can only borrow one book at a time and premium members can borrow 2 or 3 books at once. Members can borrow books, also they can see the remaining time to return the book and see what books they borrowed if they don't return the book on time, they must pay the fine and members can pay the fine online. Also, members can see special notices and members can edit their own profile by changing their own username and password.

All the member details, borrow details and fine details are managed by the librarian and the staff. They must log in through the staff account to manage the system.

Staff have their own preview of the system as they can manage members, add/remove or delete the inventory books and members and issue or return books to members and they also can edit their own account by changing username and password.

Requirement Analysis

Customer (Guest, Member), Staff (Assistant Librarian, Librarian), Administrator have permission to access the system. The Front end of the system can access by the Customers.

Customer

Customer Requirements:

- View library application.
- View available books in library.
- Customer can search books via book name, category, author, ISBN.
- Customers can see special notices published by the library.
- Customers can see trending books, new arrival books.
- Customer can rate the books.
- Customers can send enquiries without logging in.
- Customer needs to reserve books online.
- Customer can see borrowed books and remaining time.
- Customer should be able to see total fine and details of the fine for delayed return book.
- Customer can pay the fine online.
- Customer should be able to see the receipts of the payments and the details.
- Customer can edit their own profile.
- Customer should be able to change username and password.

System requirements

- system should check the validation of the user login.
- The system must verify the customer registration.
- System should store the customer details.
- System should allow the user to edit the user's own profile.
- System must allow customers to change username and password.
- System checks the inventory.
- The system should show trending books and new arrival books.
- System should show the search results.
- System should verify the fine payment.

Staff

Staff requirement

- Staff should login to the system.
- Staff can issue/return books.
- Staff can manage book inventory (add/remove/update).
- Staff should be able to see issued books.
- Staff can manage members (add/remove/ update).
- Staff can edit their own profile.
- Staff can change their username and password.
- Staff can see an overview of the library.

System requirements

- System should allow login as a staff to the system.
- System should give permission to staff to manage the library.
- System should allow staff to edit their profile.
- System should allow staff to get an overview of the system.

Administrator

Administrator requirements:

- Needs to login as an admin.
- Admins can manage roles (add/remove/update).
- Admin can manage staff (add/remove/update)
- Admin should be able to add notices to staff members.
- Admin can manage the homepage.
- Admin can check user enquiries.

System requirements

- System should allow login as a staff to the system.
- System should manage permissions.
- The system should allow admin to check user enquiries.
- Allow admin to manage roles of the system.

Data requirements

Customer

- Customer ID (c_id)
- NIC(c_nic)
- Name (c_name)
- E-mail (c_email)
- contact number (c_contact)
- Date of birth (c_dob)
- Address (c_address)
- Membership type (m_id)

Staff

- Staff ID (s_id)
- Name (s_name)
- Email(s_email)
- Contact number (s_contact)
- NIC(s_nic)
- Date of birth (s_dob)
- Address(s_address)
- Role iD (r_id)

Administrator

- Administrator ID (a_id)
- Name (a_name)
- Email(a_email)
- Contact number (a_contact)
- NIC(a_nic)
- Date of birth (a_dob)
- Address(a_address)
- Role iD (r_id)

Inventory

- Book ID (b_id)
- Name (b_name)
- ISBN (b_isbn)
- Category (i_category)
- Author (I_author)
- Quantity (i_qty)

Role

- Role ID (r_id)
- Role Name (r_name)

- Permissions (r_permission)

Issued Books

- Customer ID (c_id)
- Book ID (b_id)
- Issued Date (i_date)
- Return date (r_date)

Notice

- Notice ID (n_id)
- Role ID (r_id)
- Notice(notice)

Fine

- Fine ID (f_id)
- Customer ID (c_id)
- Fine (f_amount)

Membership

- Membership ID (m_id)
- Membership Name (m_name)

Customer membership details

- customer ID (c_id)
- membership ID (m_id)
- issue date (md_i_date)
- expire date (md_e_date)

Enquiry

- Enquire ID (e_id)
- email (email)
- enquire (enquire)

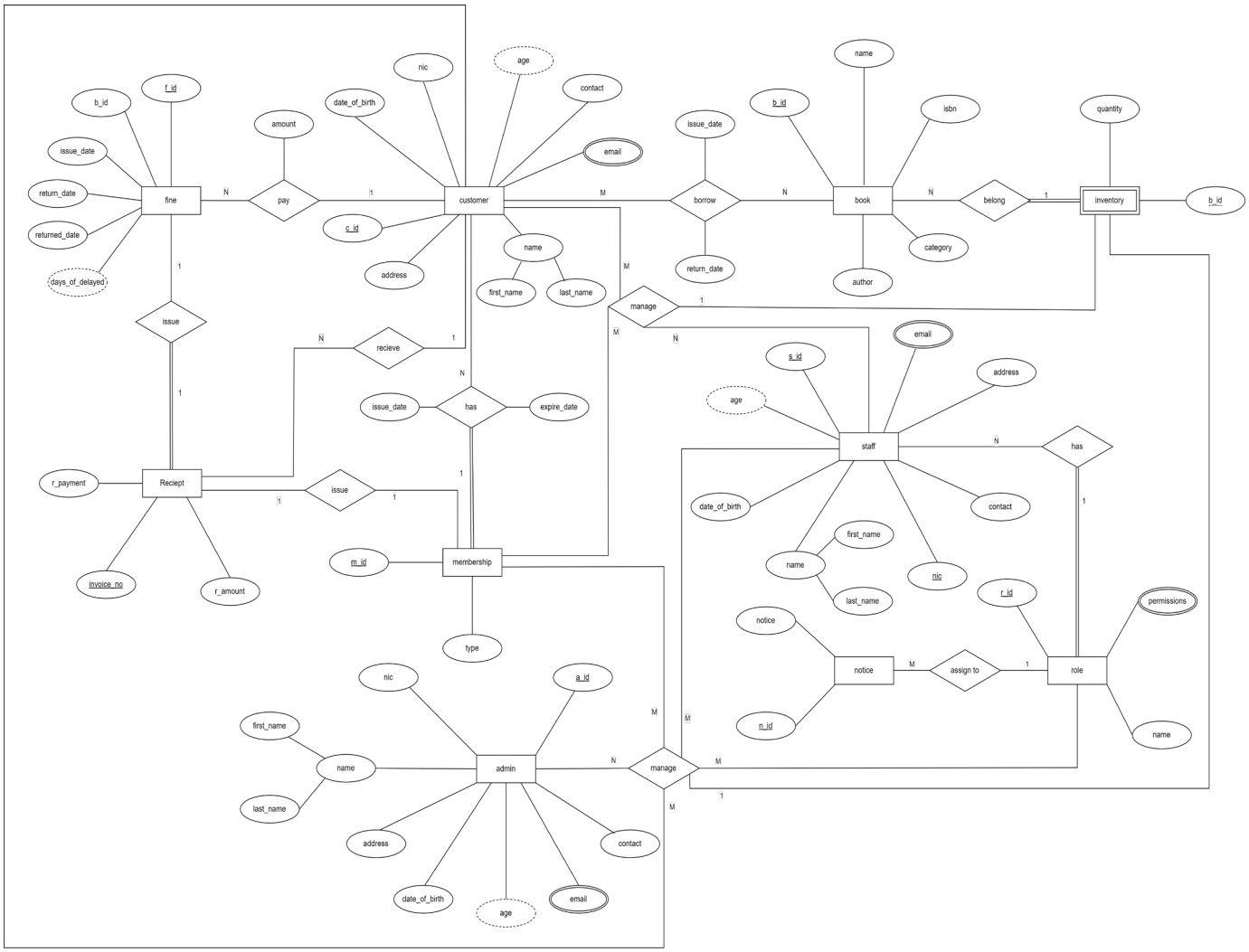
Reciept

- Invoice No (invoice_no)
- Payment for (r_payment)
- Amount (r_amount)

Non-functional Requirements

1. Availability
    - The system should provide 24/7 service.
2. Security
    - Permissions and roles to secure data from unwanted data access.
3. Compatibility
    - System users and customers can use the system from any pc with a stable internet connection.
4. Performance
    - Quick system response time.
    - Any number of users can use the system at the same time.
    - Low system specifications to run the system.
5. Maintainability and Manageability
    - Can easily fix components and Errors.
    - Administrators can easily manage the system.
6. Usability
    - System is simpler even unexperienced users can manage it easily.
    - Users can do functions more easily.

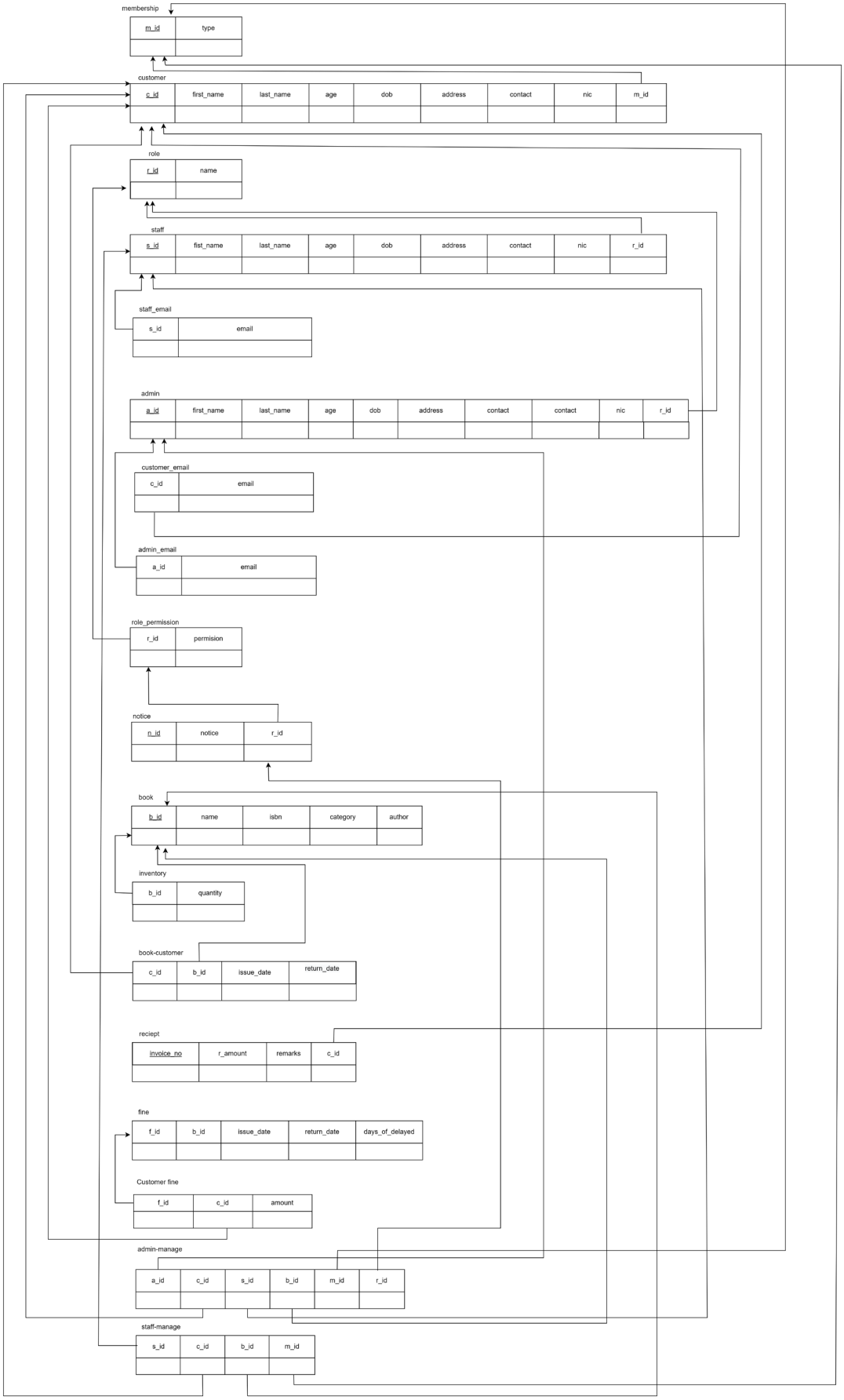
Entity Relationships and Diagrams



Schema of Database

1. membership(m_id, m_type)
2. customer(c_id, f_name, l_name, age, dob, address, contact, nic, m_id)
3. role(r_id, r_name)
4. staff(s_id, f_name, l_name, age, dob, address, contact, nic, r_id)
5. staff_email(s_id, email)
6. admin(a_id, f_name, l_name, age, dob, address, contact, nic, r_id)
7. customer_email(c_id, email)
8. admin_email(a_id, email)
9. role_permission(r_id, permission)
10. notice(n_id,notice, r_id)
11. book(b_id, name, isbn, category, author)
12. inventory(b_id, quantity)
13. book_customer(c_id, b_id, issue_date, return_date)
14. reciept(invoice_no, r_amount, remarks, c_id)
15. fine(f_id, b_id, issue_date, return_date, days_of_delayed)
16. customer_fine(f_id, c_id, amount)
17. admin_manage(a_id, c_id, s_id, b_id, m_id, r_id)
18. staff_manage(s_id, c_id, b_id, m_id)

Relational Schema



SQL Queries

```
create database Library;
use Library;

/*Member*/
create table membership(
m_id varchar(4) not null,
m_type varchar(20)not null,
constraint m_pk primary key(m_id),
constraint mid_ck check(m_id LIKE '[m/M][0-9][0-9][0-9]')
);
/*customer*/
create table customer(
c_id varchar(6) not null,
f_name varchar(50),
l_name varchar(50),
dob date,
address varchar(300),
contact int not null,
nic int not null,
m_id varchar(4) not null,
constraint c_pk primary key(c_id),
constraint c_fk foreign key(m_id) references membership(m_id),
constraint cid_ck check(c_id LIKE '[c/C][0-9][0-9][0-9][0-9][0-9]'),
constraint cmid_ck check(m_id LIKE '[m/M][0-9][0-9][0-9]')
);
/*role*/
create table role(
r_id varchar(4) not null,
r_name varchar(30),
constraint r_pk primary key(r_id),
constraint rid_ck check(r_id LIKE '[r/R][0-9][0-9][0-9]')
);
```

```
/*staff*/
create table staff(
s_id varchar(6) not null,
f_name varchar(50),
l_name varchar(50),
dob date,
address varchar(300),
contact int not null,
nic int not null,
r_id varchar(4) not null,
constraint s_pk primary key(s_id),
constraint s_fk foreign key(r_id) references role(r_id) ,
constraint sid_ck check(s_id LIKE '[s/S][0-9][0-9][0-9][0-9][0-9]'),
constraint srid_ck check(r_id LIKE '[r/R][0-9][0-9][0-9]')
);
/*staff email*/
create table staff_email(
s_id varchar(6) not null,
email varchar(100) not null,
constraint se_fk foreign key(s_id) references staff(s_id),
constraint seid_ck check(s_id LIKE '[s/S][0-9][0-9][0-9][0-9][0-9]')
);
/*admin*/
create table admin(
a_id varchar(6) not null,
f_name varchar(50),
l_name varchar(50),
dob date,
address varchar(300),
contact int not null,
nic varchar(12) not null,
r_id varchar(4) not null,
constraint a_pk primary key(a_id),
constraint a_fk foreign key(r_id) references role(r_id),
constraint aid_ck check(a_id LIKE '[a/A][0-9][0-9][0-9][0-9][0-9]'),
constraint arid_ck check(r_id LIKE '[r/R][0-9][0-9][0-9]')
);
```

```
/*customer email*/
create table customer_email(
c_id varchar(6) not null,
email varchar(100) not null,
constraint ce_fk foreign key(c_id) references customer(c_id),
constraint ce_ck check(c_id LIKE'[c/C][0-9][0-9][0-9][0-9][]0-9]')
);
/*admin email*/
create table admin_email(
a_id varchar(6) not null,
email varchar(100) not null,
constraint ae_fk foreign key(a_id) references admin(a_id),
constraint aeid_ck check(a_id LIKE'[a/A][0-9][0-9][0-9][0-9][0-9]')
);
/*role permission*/
create table role_permission(
r_id varchar(4) not null,
permission varchar(100) not null,
constraint rp_fk foreign key(r_id) references role(r_id),
constraint rp_ck check(r_id LIKE'[r/R][0-9][0-9][0-9]')
);
/*notice*/
create table notice(
n_id varchar(6) not null,
notice varchar(200),
r_id varchar(4) not null,
constraint n_pk primary key(n_id),
constraint n_fk foreign key(r_id) references role(r_id),
constraint n_ck check(n_id LIKE'[n/N][0-9][0-9][0-9][0-9][0-9]'),
constraint r_ck check(r_id LIKE'[r/R][0-9][0-9][0-9]')
);

/*book*/
create table book(
b_id varchar(6) not null,
name varchar(50)not null,
isbn int not null,
category varchar(100)not null,
author varchar(100),
constraint b_pk primary key(b_id),
constraint b_ck check(b_id LIKE'[b/B][0-9][0-9][0-9][0-9][0-9]'),
);
/*inventory*/
create table inventory(
b_id varchar(6) not null,
quantity int,
constraint i_fk foreign key(b_id) references book(b_id),
constraint ib_ck check(b_id LIKE'[b/B][0-9][0-9][0-9][0-9][0-9]')
);
/*book_customer*/
create table book_customer(
c_id varchar(6) not null,
b_id varchar(6) not null,
issue_date date not null,
return_date date not null,
constraint bc_fk1 foreign key(c_id) references customer(c_id),
constraint bc_fk2 foreign key(b_id) references book(b_id),
constraint bccid_ck check(c_id LIKE'[c/C][0-9][0-9][0-9][0-9][0-9]'),
constraint bcbid_ck check(b_id LIKE'[b/B][0-9][0-9][0-9][0-9][0-9]')
);
```

```
/*recipt*/
create table reciept(
invoice_no varchar(6) not null,
r_amount int,
remarks varchar(50),
c_id varchar(6)not null,
constraint re_pk primary key(invoice_no),
constraint re_fk foreign key(c_id) references customer(c_id),
constraint rei_ck check(invoice_no LIKE'[i/I][0-9][0-9][0-9][0-9][0-9]'),
constraint recid_ck check(c_id LIKE'[c/C][0-9][0-9][0-9][0-9][0-9]')
);

/*fine*/
create table fine(
f_id varchar(6) not null,
b_id varchar(6) not null,
issue_date date not null,
return_date date not null,
returned_date date not null,
constraint f_pk primary key(f_id),
constraint f_fk foreign key(b_id) references book(b_id),
constraint fid_ck check(f_id LIKE'[f/F][0-9][0-9][0-9][0-9][0-9]'),
constraint fbid_ck check(b_id LIKE'[b/B][0-9][0-9][0-9][0-9][0-9]')
);

/*customer fine*/
create table customer_fine(
f_id varchar(6) not null,
c_id varchar(6) not null,
amount int not null,
constraint cf_fk1 foreign key(f_id) references fine(f_id),
constraint cf_fk2 foreign key(c_id) references customer(c_id),
constraint cffid_ck check(f_id LIKE'[f/F][0-9][0-9][0-9][0-9][0-9]'),
constraint cfcid_ck check(c_id LIKE'[c/C][0-9][0-9][0-9][0-9][0-9]')
);
```

```
/*admin manage*/
create table admin_manage(
a_id varchar(6) not null,
c_id varchar(6) not null,
s_id varchar(6) not null,
b_id varchar(6) not null,
m_id varchar(4) not null,
r_id varchar(4) not null,
constraint am_fk1 foreign key(a_id) references admin(a_id),
constraint am_fk2 foreign key(c_id) references customer(c_id),
constraint am_fk3 foreign key(s_id) references staff(s_id),
constraint am_fk4 foreign key(b_id) references book(b_id),
constraint am_fk5 foreign key(m_id) references membership(m_id),
constraint am_fk6 foreign key(r_id) references role(r_id),
constraint amid_ck check(a_id LIKE '[a/A][0-9][0-9][0-9][0-9][0-9]'),
constraint amcid_ck check(c_id LIKE '[c/C][0-9][0-9][0-9][0-9][0-9]'),
constraint amsid_ck check(s_id LIKE '[s/S][0-9][0-9][0-9][0-9][0-9]'),
constraint amb_ck check(b_id LIKE'[b/B][0-9][0-9][0-9][0-9][]0-9]'),
constraint ammid_ck check(m_id LIKE '[m/M][0-9][0-9][0-9]'),
constraint amrid_ck check(r_id LIKE'[r/R][0-9][0-9][0-9]')
);
/*staff manage*/
create table staff_manage(
s_id varchar(6) not null,
c_id varchar(6) not null,
b_id varchar(6) not null,
m_id varchar(4) not null,
constraint sm_fk1 foreign key(s_id) references staff(s_id),
constraint sm_fk2 foreign key(c_id) references customer(c_id),
constraint sm_fk3 foreign key(b_id) references book(b_id),
constraint sm_fk4 foreign key(m_id) references membership(m_id),
constraint smid_ck check(s_id LIKE '[a/A][0-9][0-9][0-9][0-9][0-9]'),
constraint smcid_ck check(c_id LIKE '[c/C][0-9][0-9][0-9][0-9][0-9]'),
constraint smbid_ck check(b_id LIKE'[b/B][0-9][0-9][0-9][0-9][]0-9]'),
constraint smmid_ck check(m_id LIKE '[m/M][0-9][0-9][0-9]'),
);
```

```
/*Insert value to membership*/
insert into membership values ('M001','basic');
insert into membership values ('M002','premium');



/*Insert value to customer*/

insert into customer values('C00001','Ashan','Vithanage','2002-04-01','No 03 Flower road,Malabe',
'0788410318','2002454568','M001');
insert into customer values('C00002','Dilshan','Mendis','2001-09-20','No 03 Temple road,Ragama',
'0752555338','2001478563','M002');

insert into customer values('C00003','Ravindu','Nanayakkara','2000-02-14','No 24 Hokandara south,Hokandara',
'0719960110','2000545331','M001');

insert into customer values('C00004','Kavindu','Avishanka','2001-04-14','No 06/10 Hospital road,Angoda',
'0767532191','2001863313','M001');

insert into customer values('C00005','Chandimal','Weerasinghe','1973-12-25','No 03 Flower road,Malabe',
'0788410318','2002454684','M002');



/*Insert value to role*/
insert into role values('R001','Admin');
insert into role values('R002','Librarian');
insert into role values('R003','Assistant librarian');


 /*Insert value to staff*/



insert into staff values('S00001','Kamal','Hathurusinghe','1987-04-10','No 07 Church road,Jaela','075110485','874564142','R001');
insert into staff values('S00002','Samal','Gamage','1987-01-10','No 07 Mosque road,Kadavatha','075110489','874564131','R002');
insert into staff values('S00003','Ramal','Rajapaksha','1987-07-10','No 07 Temple road,Kiribathgoda','075110484','874564129','R003');




/*Insert value to staff email*/


insert into staff_email values('S00001','KamalHathuru2@gmail.com');
insert into staff_email values('S00001','HathuruKamal@gmail.com');
insert into staff_email values('S00002','samalgamage87@gmail.com');

/*Insert value to admin*/

insert into admin values('A00001','Mahinda','Rajapaksha','1936-04-01','No 12, Mirihana road, Thalavathugoda','0719204567','36456718','R001');

/*Insert value to email*/
insert into admin_email values('A00001','mahindarox@gmail.com');
insert into admin_email values('A00001','tsunami2004@gmail.com');



/*Insert value to permission*/

insert into role_permission values('R001','manage members');
insert into role_permission values('R001','manage inventory');
insert into role_permission values('R001','manage library');
insert into role_permission values('R001','manage users');
insert into role_permission values('R001','manage roles');
insert into role_permission values('R001','manage homepage');
insert into role_permission values('R001','manage inquiries');
insert into role_permission values('R001','manage notices');
insert into role_permission values('R002','manage members');
insert into role_permission values('R002','manage inventory');
insert into role_permission values('R002','manage library');
insert into role_permission values('R003','manage library');
```

```sql
/*Insert value to notice*/

insert into notice values('N00001','Hello world','R001');
insert into notice values('N00002','Visit Sri Lanka','R002');

/*Insert value to book*/

select * from book

insert into book values('B00001','rocket boys','0440257','history','Homer Hickam');
insert into book values('B00002','John Adams','5547894','biography','David Mccullough');
insert into book values('B00003','Harry Potter','4477896','fantacy','J.K.Rowling');
insert into book values('B00004','Hoobit','5557896','fantacy','Tolkien');
insert into book values('B00005','Madol duwa','7744256','biography','Martin Wickramasinghe');

/*Insert value to Linventory*/

insert into inventory values('B00001','10');
insert into inventory values('B00002','15');
insert into inventory values('B00003','25');
insert into inventory values('B00004','5');
insert into inventory values('B00005','12');

/*Insert value to book_customer*/


insert into book_customer values('C00001','B00001','2022-10-30','2022-11-30');
insert into book_customer values('C00002','B00002','2022-09-30','2022-10-30');
insert into book_customer values('C00003','B00003','2022-06-20','2022-07-20');
insert into book_customer values('C00004','B00004','2021-02-25','2021-03-25');
insert into book_customer values('C00005','B00005','2021-06-11','2021-07-11');
```

```
/*Insert value to recipt*/

insert into reciept values('I00001','300','Pay fine','C00001');
insert into reciept values('I00002','100','Pay fine','C00002');
insert into reciept values('I00003','500','Pay fine','C00003');
insert into reciept values('I00004','600','Pay fine','C00004');
insert into reciept values('I00005','700','Pay fine','C00005');



/*Insert value to customer fine*/

insert into customer_fine values('F00001', 'C00001', '500');
insert into customer_fine values('F00002', 'C00002', '200');
insert into customer_fine values('F00003', 'C00002', '100');
insert into customer_fine values('F00004', 'C00003', '500');
insert into customer_fine values('F00005', 'C00004', '500');


/*Insert value to fine*/
insert into fine values('F00001', 'B00001', '2022-09-28', '2022-10-28', '2022-10-30');
insert into fine values('F00002', 'B00002', '2022-09-28', '2022-10-28', '2022-10-30');
insert into fine values('F00003', 'B00002', '2022-08-28', '2022-09-28', '2022-09-30');
insert into fine values('F00004', 'B00003', '2022-07-28', '2022-08-28', '2022-08-30');
insert into fine values('F00005', 'B00004', '2022-09-28', '2022-10-28', '2022-10-30');
```

Performance Requirements

Performance is a main priority in a system. Following are the identified performance requirements in the Library Management System,

1) Once user Login system should be able to verify user.
1) System users should be able to update the system.
2) System must be response quickly .
3) System must be available (24/7).
4) System can be accessed from anywhere with stable internet connection
5) System must notify the users of any system errors.
6) The system should display the customer's preferred search results.
7) The system should allow the customer to enquire at their convenience.

Security Requirements

Security is a main criterion in a system. If not, threats can happen.

In our system, we have some security goals as follows,

1) User details must be secured.
2) Payment information like card details must be secured.
3) System must be able to validate real username and passwords.
4) Passwords and username must be secured.
5) Only authorized personnel can access and change the system (like manage the users).