VIET NAM UNIVERSITY HCMC.

INTERNATIONAL UNIVERSITY



ASSIGNMENT REPORT

PROJECT: OBJECT ORIENTED PROGRAMING

Lê Trần Minh Hoàng ITDSIU18042

Cao Ngọc Bảo Long ITDSIU18031

Mai Đặng Nhật Anh ITDSIU19031

# 

# Content

[Content 2](#_Toc61356445)

[Abstract 3](#_Toc61356446)

[I. Introduction 4](#_Toc61356447)

[1.1 Aim & objective of the project. 4](#_Toc61356448)

[1.2 Overview of project process. 4](#_Toc61356449)

[1.3 Overview of the report. 5](#_Toc61356450)

[II. Implement 6](#_Toc61356451)

[1. Design Model 6](#_Toc61356452)

[1.1 Analyze requirement and create entity diagram 6](#_Toc61356453)

[1.2 Sql querry. 8](#_Toc61356454)

[1.3 Database Security. 11](#_Toc61356455)

[2.GUI application 13](#_Toc61356456)

[III Conclusion. 17](#_Toc61356457)

# 

# Abstract

The revolution of information technology over database management system has lead to the widespread application software that apply to the management and organizing of company and organization from every field. At this time, database hosting services for cloud computing have become popular such as Microsoft Azure, Amazon web service, Google cloud platform… so that it is possible for any organization to access to database storing technique.

This report describes the step-by-step design, modeling and implementation of an application using database to manage a small and medium enterprise. This application provides the client-side interface for the customer to interact with the system and the interface for staff manager to interact with the back-end database server to update, modify, manipulate information store in the database.

# Introduction

Github Link: https://github.com/ITDSIU18031/OOP

## 1.1 Aim & objective of the project.

This document is a report for group project “Bookstore management system”. The project attempt to design step-by-step and implement a completely database model for managing a bookstore, besides construct an application that provides user interface for the customer to interact with the system (view books, add book to cart, generate bill..) and user interface for the staff management to retrieve, update, modify, manipulate data.

## 1.2 Overview of project process.

The project was conducted for three months including gain knowledge and related information related to the project, design model and implement. To meet all the stated criteria of the requirement, the project was taken sequentially:

1. Design model – this part was separated into 4 step:
2. Analyze requirement and create entity diagram: investigate the requirements and split into work flows to specify inceptive tables and those fields.
3. Convert into relational model: base on the entity diagram, specify the relation of tables.
4. Create database base on relational model and stub data then using sql query to manipulating data.
5. Display the sql query in form of relational algebra.
6. Enhance database security: apply secure method to defense database system from attacking.

2. Implement the application GUI and connect to database: the software is implemented mostly in Java, especially Java Swing library to support making a flexible GUI.

3. Testing : separate the application into functionalities to figure out errors and fix.

## 1.3 Overview of the report.

The report theoretically describes the process of making the project. The report is split into three main parts:

1.Introduction : a brief introduction, objective and overview of the report.

2.Implementation: a detail presentation on implement project. Including step-by-step design and build the database, how to connect database to the program and construct the GUI.

3.Conclusion: investigate the success and undone matter of the project. Moreover discuss about the vision of development.

# II. Implement

## 1. Design Model

### 1.1 Analyze requirement and create entity diagram

* 1. Requirements: build the database system to store book’s information, manage books, allow user to interact, check for availability of the books and pay the bill.
* Books are organized in into disciplines (language, novel, technology…).
* Customer check the availability of ordered books.
* Customer freely choose books add into cart.
* Check-out counter check the discount then print the bill.
* Customer pay for the bill
  1. Detail analysis:

***Schema.***

* Book has attribute: (ISBN, title, genre, price, amount, year) to store information of the book.:

+ ISBN is the primary key.

* Author has attribute :(fullname, nation, age):

+ fullname is the primary key.

* Discount has attribute: (id, percentage):

+ id is the primary key.

* Publisher has attribute:(name, address, phone):

+ name is the primary key

* Customer has attribute:(id, name, email, address):

+ id is the primary key.

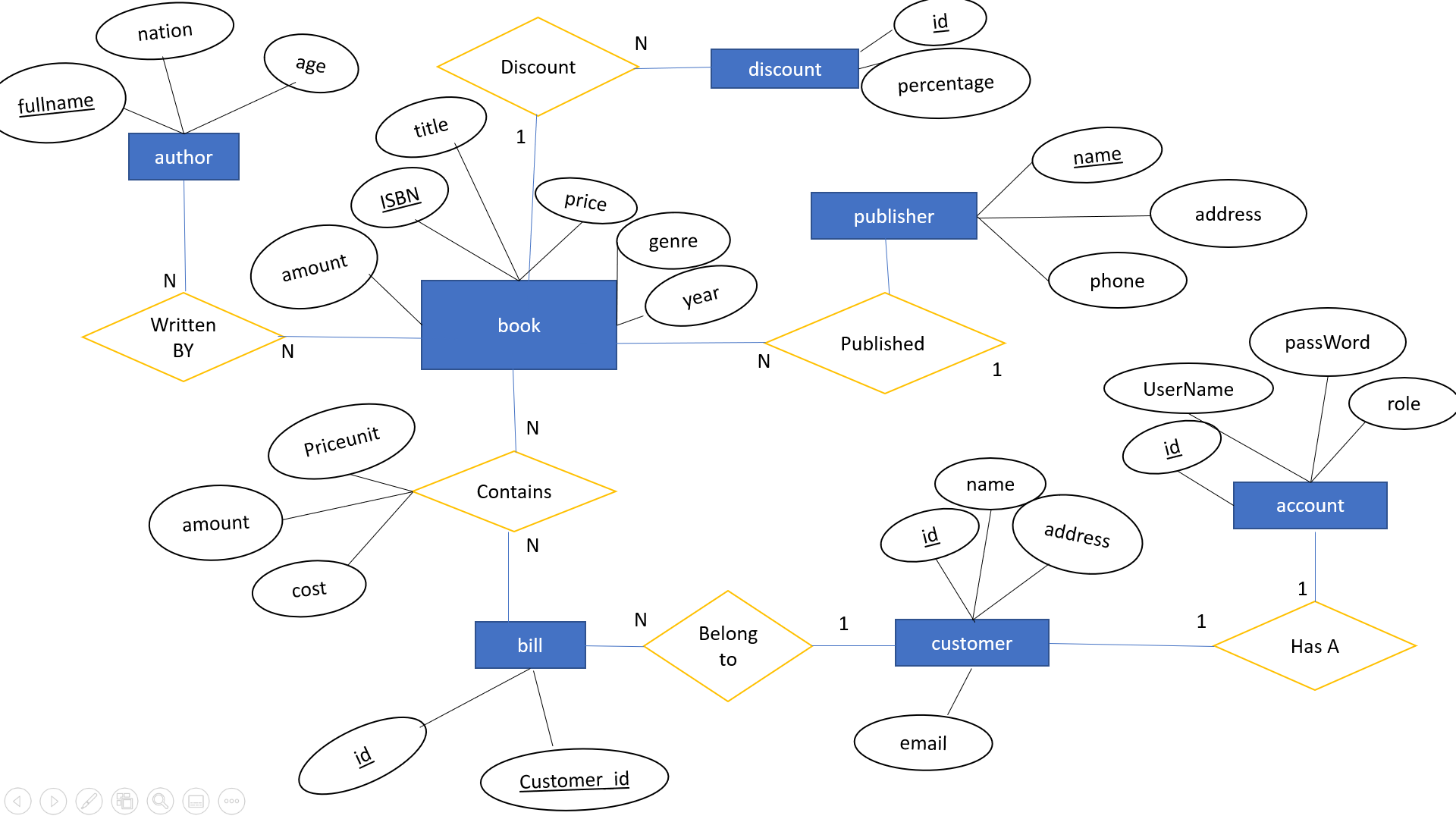
* Account has attribute: (id, userName, password, role):

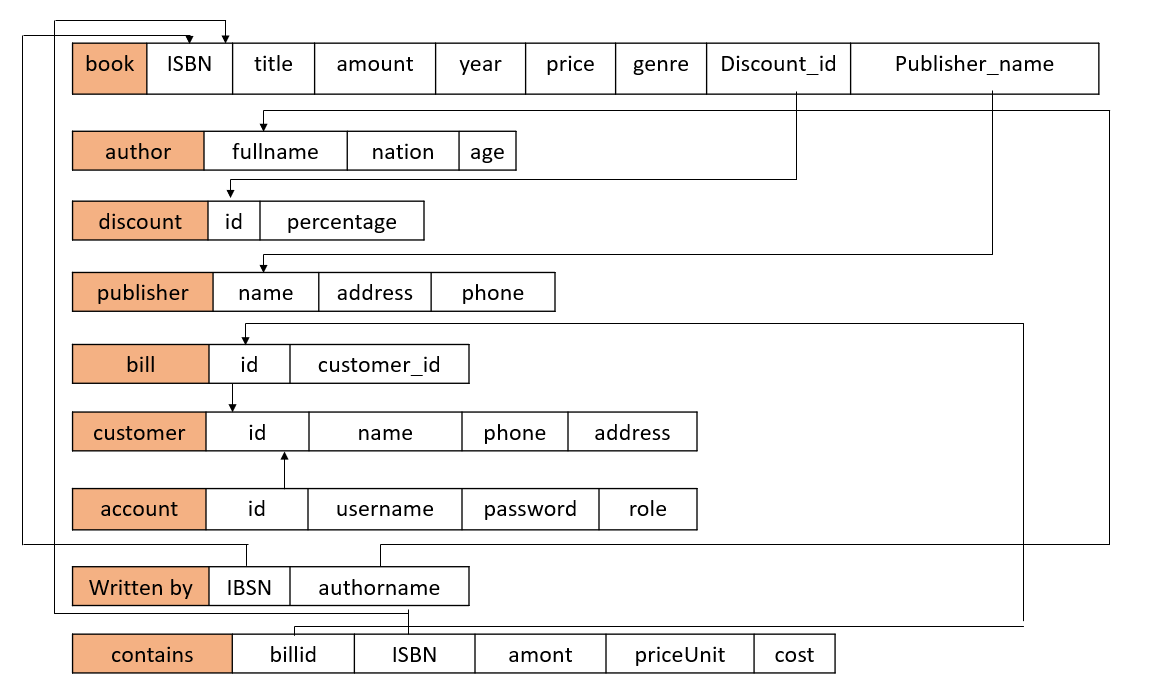
+ id is the primary key.

***Relation.***

* Each customer has only 1 account and 1 account belongs to exactly 1 customer.
* Books are written by many author, many author write many books.
* Books can have 1 discount price, 1 discount belongs to many books.
* Books can have only 1 publisher, 1 publisher can public many books.
* Books belongs to many bills, bills have many books.
* Bill belongs to only 1 customer, customer can have many bills.
* 1 customer has only 1 account, 1 account belongs to 1 customer.

***Entity diagram:***





### 1.2 Sql querry.

**Create table**

use bookstore;

SET SQL\_SAFE\_UPDATES = 0;

--Create author table

CREATE TABLE author

( fullname varchar(40) ,

age int,

nation varchar(10),

CONSTRAINT author\_pk PRIMARY KEY (fullname));

--Create discount table

CREATE TABLE discount

( id int NOT NULL AUTO\_INCREMENT,

percentage float ,

CONSTRAINT discount\_pk PRIMARY KEY (id));

--Create publisher table

create table publisher

(name varchar(40),

address varchar(50),

phone int,

CONSTRAINT publisher\_pk PRIMARY KEY (name));

--Create customer table

CREATE TABLE customer

( id int NOT NULL AUTO\_INCREMENT,

name varchar(40) ,

address varchar(40),

email varchar(40),

CONSTRAINT customer\_pk PRIMARY KEY (id));

--Create bill table

create table bill(

id int NOT NULL AUTO\_INCREMENT,

customer\_id int,

CONSTRAINT bill\_pk PRIMARY KEY (id),

FOREIGN KEY (customer\_id) REFERENCES customer(id));

--Create book table

create table book(

ISBN int NOT NULL AUTO\_INCREMENT,

title varchar(40),

genre varchar(15),

price float,

year int,

amount int,

discount\_id int,

publisher\_name varchar(40),

CONSTRAINT book\_pk PRIMARY KEY (isbn),

FOREIGN KEY (discount\_id) REFERENCES discount(id),

FOREIGN KEY (publisher\_name) REFERENCES publisher(name),

CHECK (amount>=0));

--Create writtenby table

create table writtenby(

isbn int,

authorname varchar(40),

FOREIGN KEY (isbn) REFERENCES book(isbn),

FOREIGN KEY (authorname) REFERENCES author(fullname));

--Create contain table

create table contain(

billid int,

isbn int,

amount int,

priceunit float,

cost float,

FOREIGN KEY (isbn) REFERENCES book(isbn),

FOREIGN KEY (billid) REFERENCES bill(id) );

--Create account table

CREATE TABLE ACCOUNT(

ID int,

Username varchar(30),

Password varchar(15),

Role varchar(10),

FOREIGN KEY (ID) REFERENCES customer(id),

Check role=’staff’ or ‘customer’

);

**Update:**

1.Set Constraint, key and update cascade for table.

Alter table writtenby drop foreign key writtenby\_ibfk\_1;

alter table writtenby add foreign key (isbn) references book(isbn) ON DELETE set null ON UPDATE CASCADE;

alter table writtenby add foreign key (authorname) references author(fullname) ON DELETE set null ON UPDATE CASCADE;

Alter table contain drop foreign key contain\_ibfk\_1;

Alter table contain drop foreign key contain\_ibfk\_2;

alter table contain add foreign key (isbn) references book(isbn) ON DELETE set null ON UPDATE CASCADE;

alter table contain add foreign key (billid) references bill(id) ON DELETE set null ON UPDATE CASCADE;

alter table bill drop foreign key bill\_ibfk\_1;

alter table bill add foreign key (customer\_id) references customer(id) on delete SET NULL ON UPDATE CASCADE;

alter table book drop foreign key book\_ibfk\_1;

alter table book drop foreign key book\_ibfk\_2;

alter table book add foreign key (discount\_id) references discount(id) on delete SET NULL ON UPDATE CASCADE;

alter table book add foreign key (publisher\_name) references publisher(name) on delete SET NULL ON UPDATE CASCADE;

2.Update data value

#querry 1. Update new attribute “series” into table book.

ALTER TABLE book

ADD series String

#querry 2. update set book naruto epsion 3 increase amount of 20 books

SET SQL\_SAFE\_UPDATES = 0;

UPDATE book

SET amount=amount+20

WHERE title='naruto epsion 3 ';

#querry : add new attribute series to book

ALTER TABLE book

ADD series varchar(20);

#querry : delete bill which customer\_id is null

DELETE from bill

WHERE customer\_id IS NULL;

### 1.3 Database Security.

Database is the one of the most essential part of any organization or company, database deal with large amount of importance or even secret information of organization. Many people try to access to those database illegally for devastating purpose. One of the most common but still dangerous is *sql injection*.

Sql injtection is the database penetrating technique through input form with unusual symbol to mislead the database engine.

Consequence:

+ Extract senditive information: user account, credit card…

+ Detroy data.

+ Inject code to be executed when user login.

Ex: input the information to mislead the database engine:

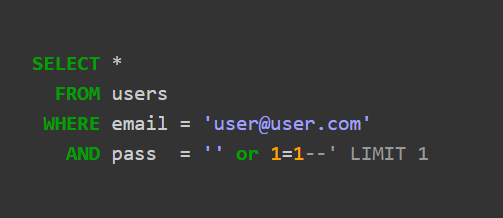
User Name

user@user.com

Password

‘ or 1=1 --

In the database query :



The result of relational algebra OR return true when 1 of 2 statement is true, the statement “1=1” is true , so allow to connect to the database successfully without the real password. Forturnately, a solution called *Parameterized Statement* was born to deal with this injection.

1. Parameterized Statement.

* The java library java.sql.Statement and java.sql.PreparedStatement provide type of constain that take input from the user form and exclude the symbol( “,’,`…). If the user intendedly input the string including(; , ’ ”), the program will stop and return an alert. So that the sql driver cannot be confused to pass the authentication without the real password.

1. Hashing

* In case the hacker have already injected to the database, the user account is one of the most target to take over.
* Md5 algorithm : is a one way hash secure algorithm.

+ There is no way to turn the hashed password back.

+ MD5 Algorithms are helpful because it is easier to compare and store these smaller hashes than to store a large text of variable length.

Ex: 123 → Hash()→ 000fich2d351g9881i092ag48600fea8

1. Least Privilege.

* Software should apply the “levels of clearance” to ensure that each process or component can access and affect only resources it needs.
* User account is “read-only” account.
* Admin account is the only one account that could update, modify, drop… data.

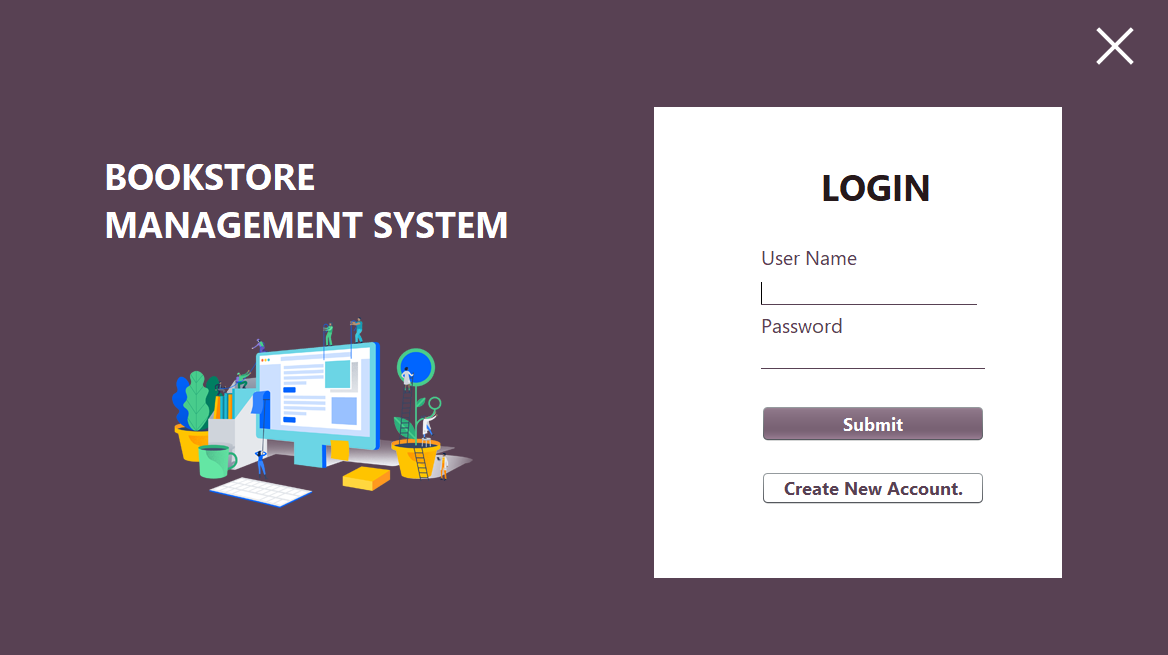
## 2.GUI application

Application configuration:

* Java sdk 11.
* IDE: Eclipse IDE for Enterprise Java Developers - 2020-09
* Jdbc connector: mysql-connector-java-5.1.49
* GUI library: java swing.
* Database: Mysql Workbench.

--Conection to MySql





Login UI

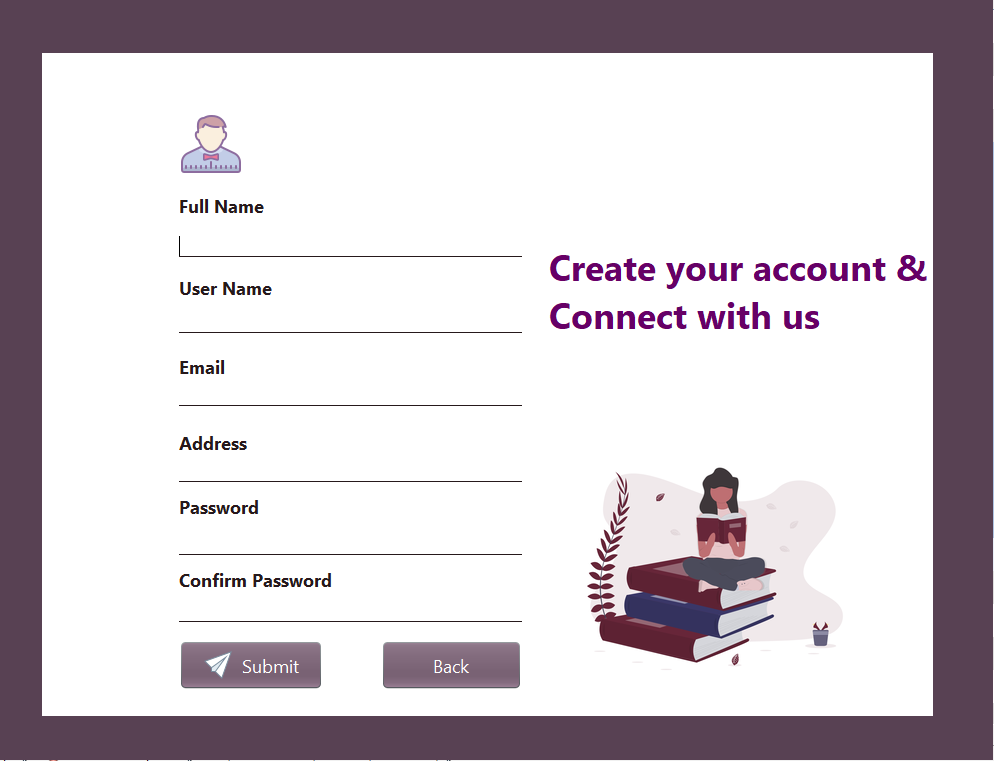
Staff

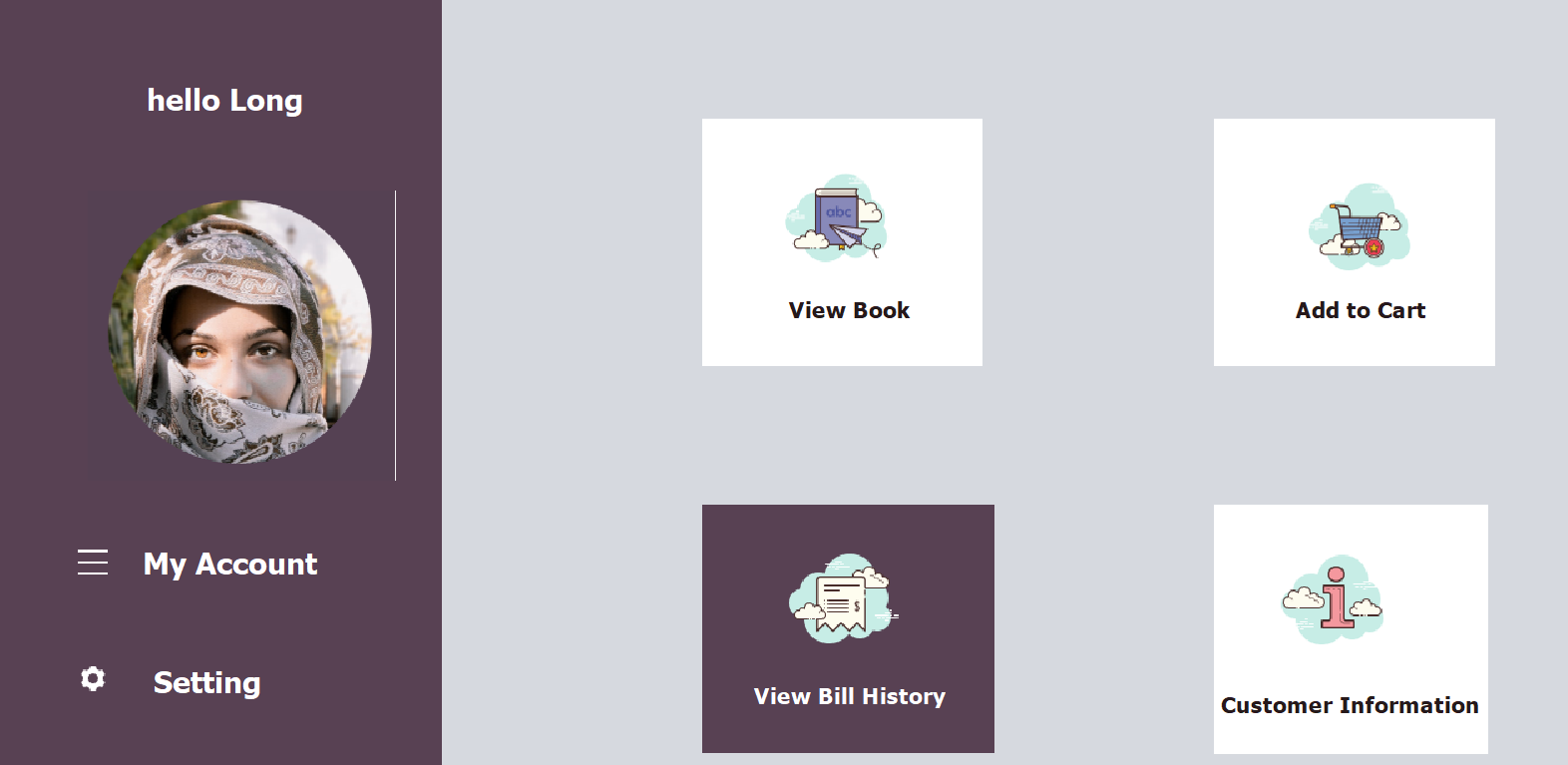
Username: ‘admin’

Password:’admin123’

In login stage, the authentication algorithm check the userName and password with the userName and hashed password in the database and role of the account to decentralize the access right.

For the new user, button Create New Account pop up the CreateAccount frame.



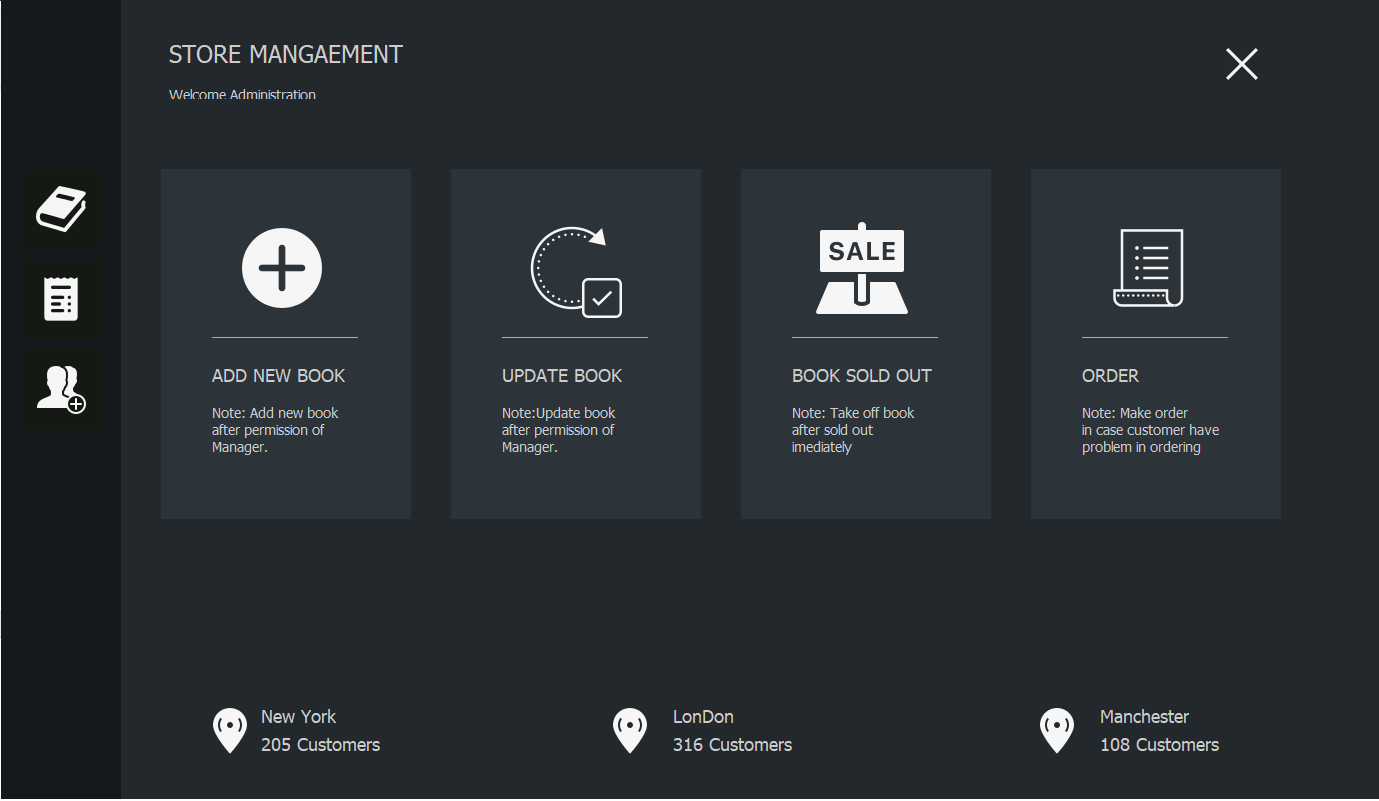


Customer UI

After login as customer role, the program lead to the Customer UI.

Customer have these functionalities :

* View Book: Gui display all book on the store. Customer an also search for any information relate to the book( title, genre, year, author..).
* Add to cart: customer input the ISBN of the book and amount for order.
* View Bill History: Customer can retrieve all of bills that has been ordered unitl now, each bill has the information of the ISBN of books, amount and the total cost that customer have to pay.
* Customer Information: view customer personal information, customer can update there information.



Staff UI

After Login as Staff role, the program lead to Staff UI.

The Staff have these functionalities:

* View Book: Gui display all book on the store. Customer an also search for any information relate to the book( title, genre, year, author..).
* Update Book: Staff have permission to update the book information and store in database.
* Book Sold out: display the books have been sold out.
* Order: Staff can make an order for offline customer.
* View all Bill: Staff can look for bills of all customers of the store.
* View all Customer: Staff can look for all customer information.

# III Conclusion.

This part summarizes the success features that have done in the project and discuss about how to extend the project and vision of development. The key conclusions:

* Design and build successfully a complete database model:

+ Analyze requirement and create entity diagram: investigate the requirements and split into work flows to specify inceptive tables and those fields.

+ Convert into relational model: base on the entity diagram, specify the relation of tables.

+ Create database base on relational model and stub data then using sql query to manipulating data.

+ Display the sql query in form of relational algebra.

+ Enhance database security: apply secure method to defense database system from attacking.

* Implement successfully an application that connects to database and provides UI for user and staff can interact with the system.

Development vision:

* Store image of the books in database.
* Add function “Recharge money to account” so that user can pay the bill online.
* Develop web version of the system which can run parallel with this application.

THE END.