Mô tả thiết kế lớp

1. User Case: Browsing Catalog

**public class UserCard {**

private User Borrower;

private Date ExpiredDate;

/\*

\* Lay ten nguoi so huu the

\*@return User.name: ten nguoi so huu the

\*/

public char getUserName() {

return User.name;

}

/\*

\* Lay ngay het han the

\*@return ExpiredDate ngay het han cua the

\*/

public Date getExpireDate() {

return ExpiredDate;

}

}

**public class SearchController {**

private ListCatalog List\_Catalog;

private Catalog catalog;

/\*

\* Lay toan bo catalog

\* @return List catalog

\*/

public Array(Catalog) getListCatalog() {

return List\_Catalog.getAllCatalog;

}

/\*

\* Lay toan bo book trong catalog da chon

\*@param Catalog\_id ma catalog da chon

\* @return List book

\*/

public Array(Book) getListBook(int Catalog\_id) {

catalog=List\_Catalog.getCatalog(Catalog\_id);

return catalog.getListBook;

}

/\*

\* Lay quyen sach co id tuong ung trong catalog da chon

\* @param book\_id ma book da chon

\* @return Book

\*/

private Book SearchBook(int Book\_number) {

return catalog.getListBook[Book\_number];

}

}

1. User Case: Borrow Register

**public class Borrower\_bookManagement {**

/\*

\* Kiem tra xem nguoi su dung có User\_number = Borrower\_number co cuon nao qua han tra k

\*@param Borrow\_number ma so nguoi muon

\*@return true neu nguoi muon co sach chua tra dung han, false neu nguoi muon da tra het sach hoac so sach dang muon chua qua han

\*/

public boolean getOverdueUnreturnBook(int Borrower\_number) {

Records= lay tat cac cac Borrow\_Book co borrower\_number = Borrower\_number

foreach(Record in Records){

if Record.Suppose\_to\_return> ngay hien tai

return true

}

return false;

}

}

**public class Book{**

private int Book\_number;

private int Catalog\_id;

private String Title;

private String Publisher;

private String Author;

private int ISBN;

private Array(Copy) Copy\_List;

public char getBookTitle() {

return 0;

}

/\*

\* Kiem tra xem có còn tồn tại các bản copy khả dụng không

\*@return true if moi ban copy deu borrowable

\*/

public boolean CheckCopyStatus() {

foreach(Record in Copy\_List){

if Record.Type==borrowable

return true;

}

return false

}

/\*

\* Dem so ban copy co the muon duoc

\*@return so ban copy co the muon dc

\*/

public int getNumberofValidBook(int Book\_number) {

int dem=0;

foreach(Record in Copy\_List){

if Record.Type==borrowable

dem=dem+1;

}

return dem

}

}

**public class BorrowRequestManagement {**

/\*\*

\*save valid borrow Request to Database

\* @param Borrow\_numer ma so the muon sach

\* @param Copy\_number ma so ban copy

\*\*/

public void save(int Borrow\_number, int Copy\_number) {

Generate ra ngay hien tai

Save bản ghi với đầy đủ thông tin: Borrow\_number, Copy\_number, Borrow\_date vào database

}

}

**public class BorrowRegisterController {**

private Borrower\_bookManagerment manage;

privet BorrowRequestManagement requestManage;

/\*

\* kiem tra xem borrow request co hop le

\* @Book\_number ma so book muon chon

\* @number so ban copy muon muon

\* @Borrower\_number ma so nguoi muon

\* @return true neu request hop le, false neu khong

\*/

public boolean Borrow(int Book\_number,int number, int Borrower\_number) {

Return checkUserCardStatus(Borrower\_number)&&checkWantedCopy(Book\_number,number)&&manager.getOverdueUnreturnBook(Borrower\_number);

return;

}

/\*

\* Kiem tra có save Request thanh cong khong va save request

\* @param Copy\_number ma so ban copy

\* @param Borrower\_number ma so nguoi muon

\* @return true neu thanh cong

\*/

public boolean saveRequest(int Copy\_number, int Borrower\_number) {

return requestManage.save(Copy\_numberm, Borrower\_number);

}

/\*

Kiem tra so luong ban muon muon co nho hon hoac bang so luong ban copy phu hop khong

\*@param Book\_number ma sach muon muon

\*@param number so luong sach muon muon

\*@return true neu so luong copy muon muon < so luong copy co the muon

\*/

private boolean checkWantedCopy(int Book\_number, int number) {

Book book = tim book trong database where book.Book\_number=Book\_number;

if(book.checkCopyStatus== false)

return false;

if(book.getNumberofValidBook<number)

return false;

return true;

}

/\*

\* kiem tra tinh trang the ng dung

\* @param Borrower\_number ma so nguoi dung

\* @return true neu the con co the su dung

\*/

private boolean checkUserCardStatus(int Borrower\_number){

UserCard card= tim card trong database where card.borrower\_number=Borrower\_number

if(card.getExpireDate>now)

return true;

else return false;

}

}