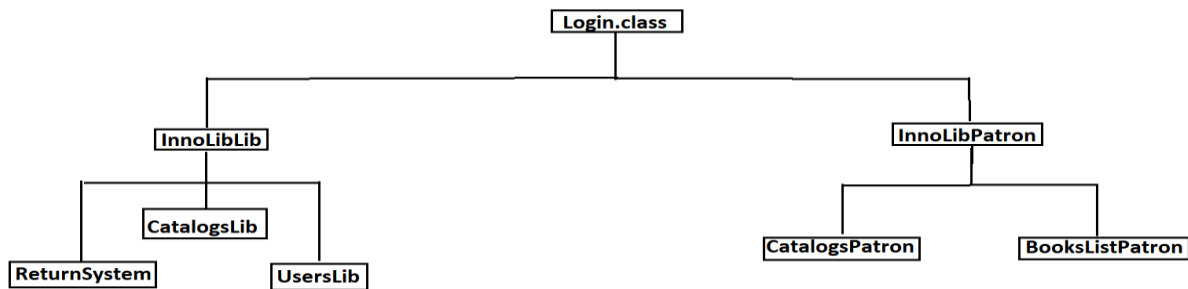


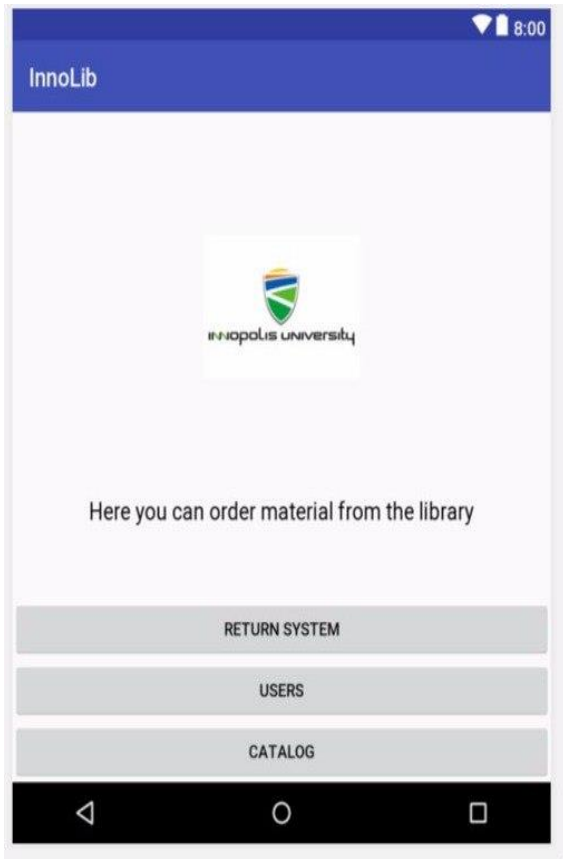
## Package Activities ( all classes extends from Activity ):



## Class Login:

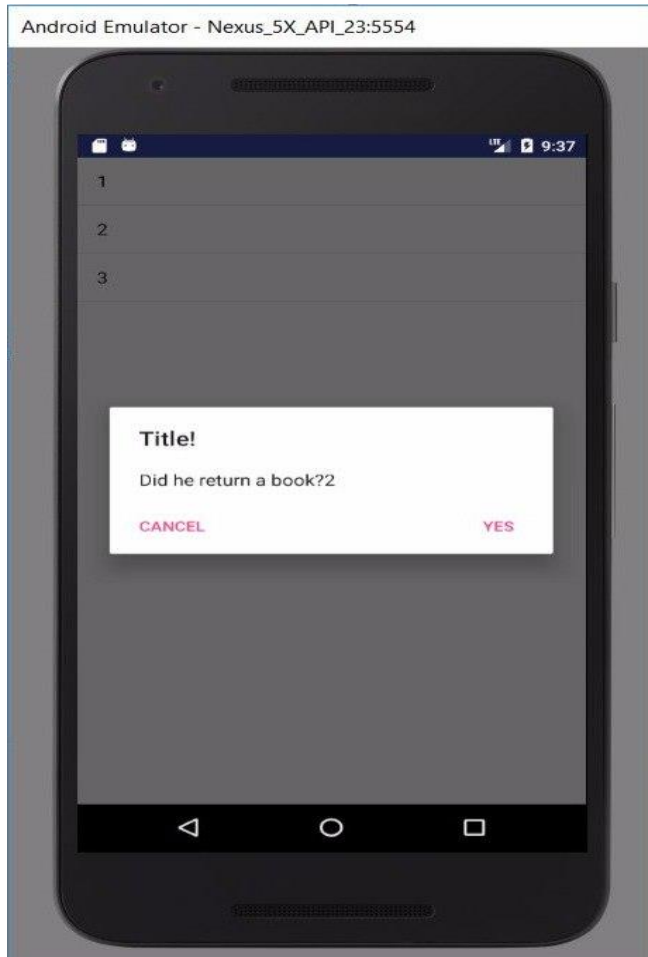
In class *Login* we have two fields for input and one button. In method **OnClickListener** of this button we choose which view will have user - *Librarian's* or *Patron's* (it depends on login which we will compare with database info)

## Class InnoLibLib:



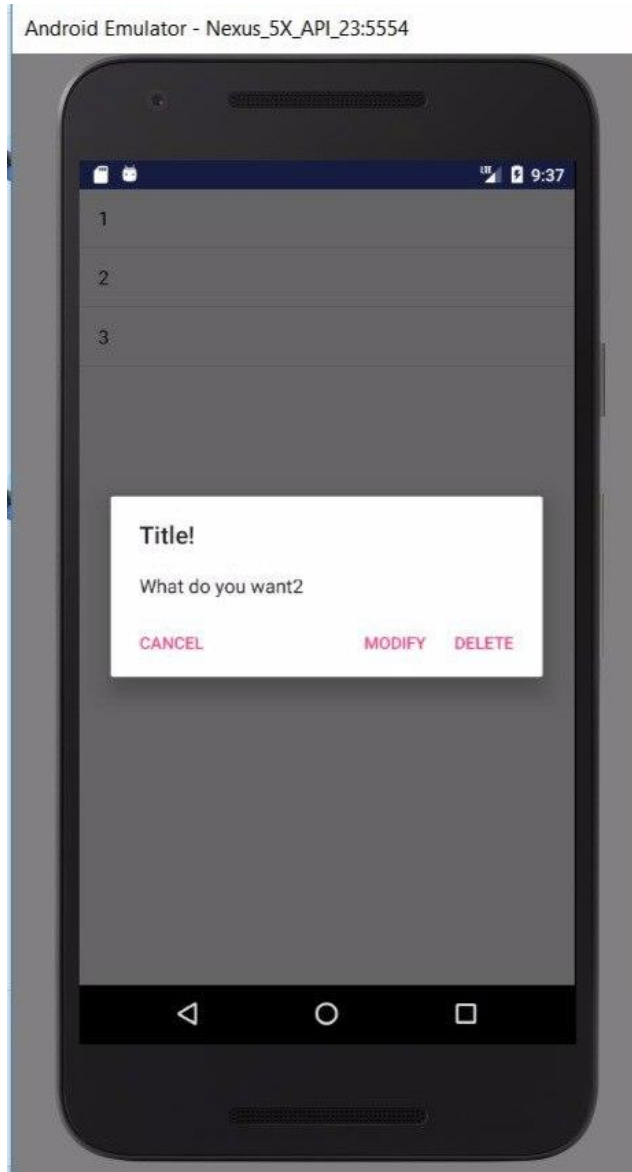
Here the same situation as in previous class : we have 3 buttons and 3 **OnClickListener** for each of them. Button "**RETURN SYSTEM**" will switch current activity on *ReturnSystem* class, "**USERS**" on *UsersLib*, "**CATALOG**" on *CatalogsLib*.

## Class ReturnSystem:



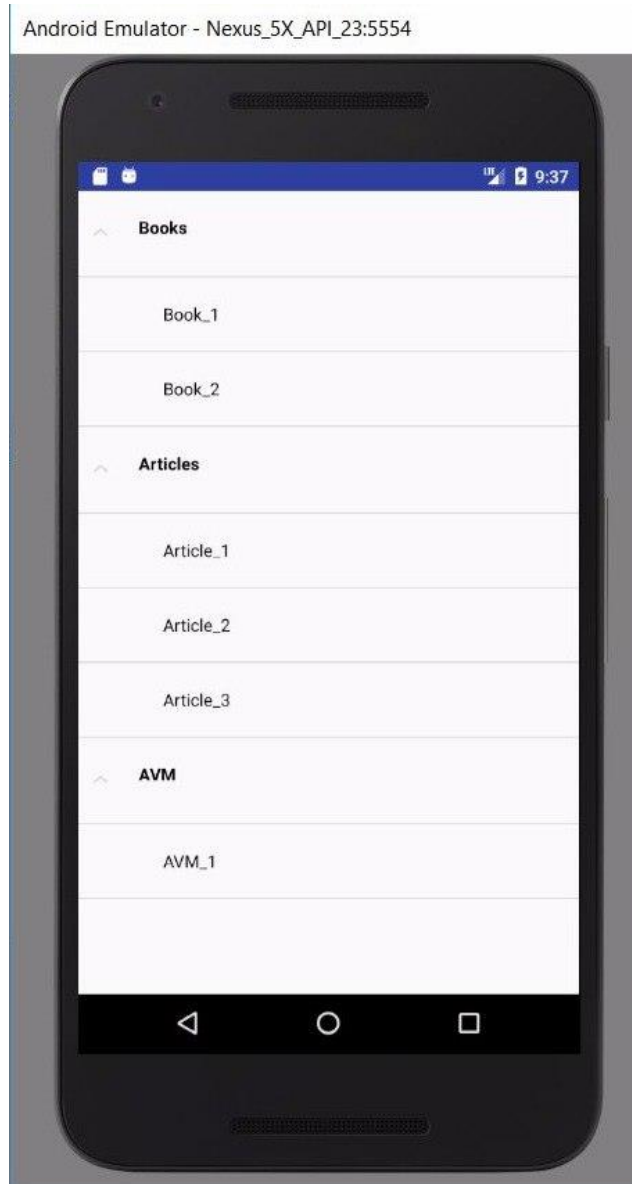
In this class we have **ListView** which contains a list of ordered books from DB. For this list we have method **OnItemClickListener** and when we tap on any element in this list there is append a special **AlertDialog** window where we can confirm recieved book and then delete it from DB.

## Class UsersLib :



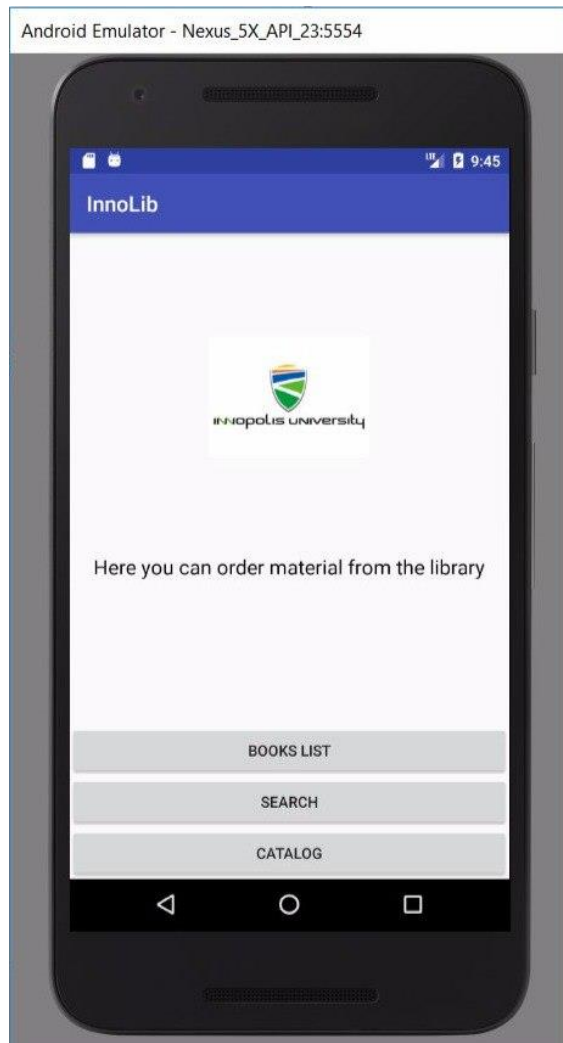
Here we have a **ListView** of all users For this list we have method **OnItemClickListener** and when we tap on any element in this list there is append a special **AlertDialog** window where we can **modify** or **delete** current user.

## Class CatalogsLib:



In this class we have **ExpandableListView** where we see all available Documents. In the future we will be able to **modify, delete and add Documents** like in *UsersLib*

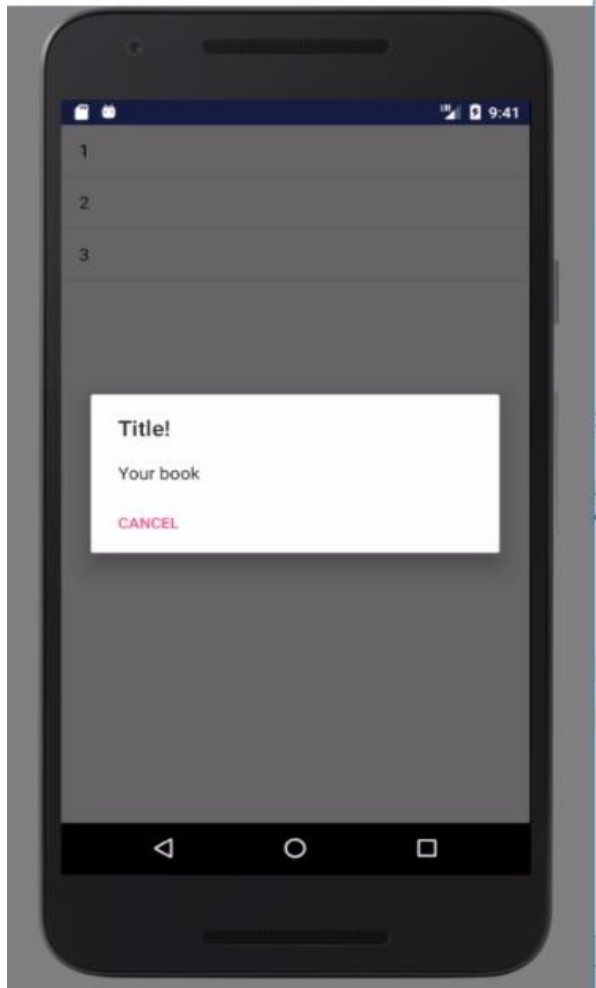
## Class InnoLibPatron:



Here we have 3 buttons which have the same principle as **InnoLibLib**.

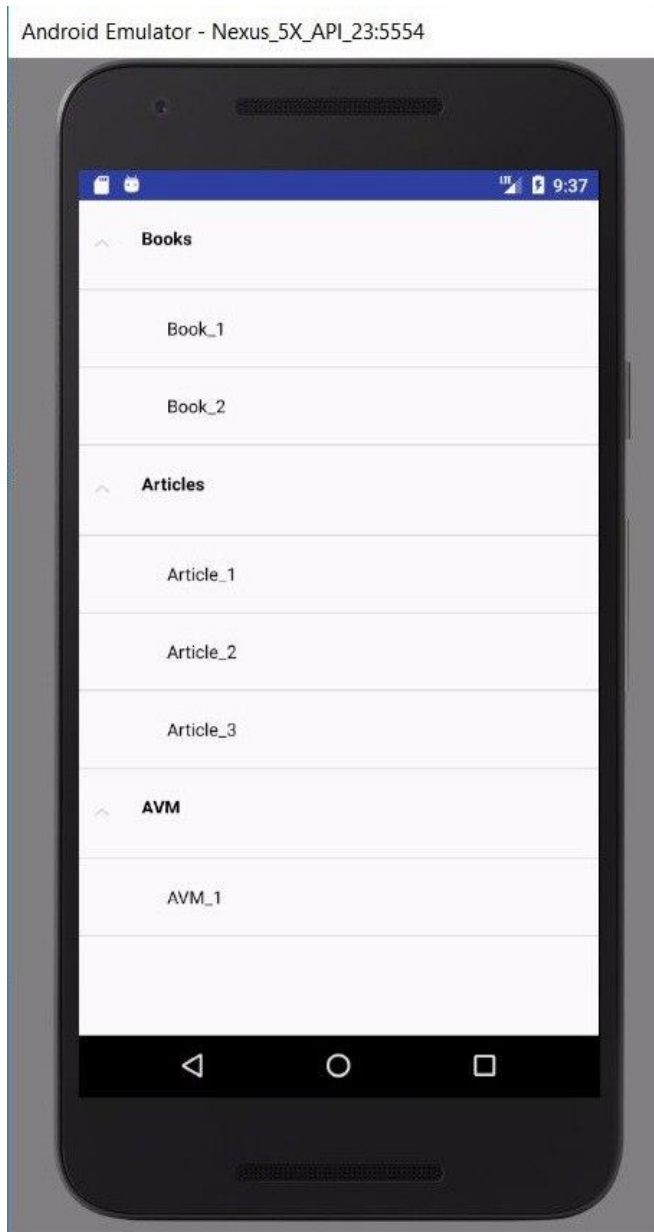
## Class BooksListPatron:

Android Emulator - Nexus\_5X\_API\_23:5554



Here User can see the ShortInformation of his ordered Documents and in **AlertDialog** we see FullInformation.

## Class CatalogPatron:



Here user see all availables Documents and can order them.(only interface)



## Package Classes

### Class Books:

Class books contains 2 constructors through array where each element is field of the book or arguments that are fields of the book. It contains all the information about one book that is in data base.

### Class Librarian(extends UserCard):

Class librarian contains constructor through array or inputs. It contains information about one user and also contains method that returns all patrons from data base.

Method *getUsers*:

Return ArrayList of objects of type Patron

### Class Patron(extends UserCard)

Class patron contains constructor through array or inputs. It contains methods check out that take one book from data base, add to another data base the line that signals that this user took book one exactly book. Also it contains method that returns books from data base

1) Method *checkOut*:

This method check out book for patron if available.

2) Method *getAvailableBooks*:

This method return list of books

3) Method *hasCopy*:

This method check if patron has copy of book or not. Argument of this method is book id. Return true or false

### **Class UserCard:**

Class usercard contains constructor through array or inputs. It contains information about one user and have all needed getters and setters

### **Package DataBase:**

#### **Class DataBaseHelper :**

Through this class we connect and update database. Also here located all methods for working with database.

1) method *CreateDatabase*:

Creates an empty database and overwrites it with our own database(we don't need create new database because we already created it)

2) method *checkDataBase*:

Checks if this database already exists so that it does not copy each time the application starts. Return true if exist, false if doesn't.

3) method *openDataBase*:

open data base for reading

4) method *OnCreate*:

this method is empty because data base already exist

5) method *createUser*:

this method add new user in database, class librarian use this method

6) method *getStringUser*:

return String with all information about user with id that is argument of this method

7) method *getStringBook*:

return String with all information about book with id that is argument of this method

8) method *addB*:

Arguments of this method are all information about book. This method add new book to the database.

9) Method *addArt*:

Arguments of this method are all information about Article. This method add new Article to the database.

10) Method *addAV*:

Arguments of this method are all information about AV. This method add new AV to the database.

11) Method *deleteBook*:

This method delete book from database. Argument is id of book.

12) Method *deleteArticle*:

This method delete Article from database. Argument is id of Article.

13) Method *deleteAV*:

This method delete AV from database. Argument is id of AV.

14) Method *deletePatron*:

This method delete patron from database. Argument is id of patron.

15) Method *updateBook*:

This method update information about book in database. First argument is id of the book that we want update, other arguments new information of the book that will change with old information.

16) Method *updateArticle*:

This method update information about article in database. First argument is id of the article that we want update, other arguments new information of the article that will change with old information.

17) Method *updateAV*:

This method update information about AV in database. First argument is id of the AV that we want update, other arguments new information of the AV that will change with old information.

18) Method *updateUser*:

This method update information about user in database. First argument is id of the user that we want update, other arguments new information of the user that will change with old information.

19) Method *getArrayUser*:

This method return array that contain information about User, each element of this array contain each field of User. Argument of this method is id of user.

20) Method *getListOfUsers*:

This method return ArrayList that contain objects of type Patron

21) Method *getArrayBook*:

This method return array that contain information about Book, each element of this array contain each field of Book. Argument of this method is id of book.

22) Method *getListOfBooks*:

This method return ArrayList that contain objects of type Books

23) Method *updateBookData*:

This method update information about book in database. Argument of this method is object of type Books.

24) Method *updateUserData*:

This method update information about user in database. Argument of this method is object of type UserCard.

25) Method *hasBook*:

Arguments of this method are user id and book id. This method checks whether the user has this book. Return true or false.

26) Method *updateTimeChecker*:

This method update table in database "TimeChecker" that helps to check how much time is left for the user to return the book. 4 arguments: user id, book id, time that will be new time, type – type of document(book,article,AV)

27) Method *returnListOfUsersBook*:

This method return ArrayList of Books that user have. Argument is Id of user.

28) Method *getShortInformation*:

This method return String that contains information about book(Title, author,edition).

Argument is object of type Books.

29) Method *getFullInformation*:

This method return String that contain information about book (Title, author,edition,Date of creation of Book, published\_by, count of books , is bestseller,price).Argument is object of type Book.

30) Method *debtorUsers*:

This method return ArrayList that contain objects of type patron that is debtors.

## Documentation delivery 3

### Database:

added a new table to the database - "queue" for the priority queue

fixed some tables in Database

### Package DataBase:

class DataBaseHelper:

#### 1.method noOneInQueue:

This method has 2 arguments: int id, int type. Method checks whether the document is queued. Return true if document not in Queue and false if not.

#### 2.Method standInQueue:

This method has 3 arguments: object of type Patron, id of document, type of document. Method add Document in queue

#### 3. Methods for outstanding request and for priority queue

### Package "Classes":

1. Were added 2 new classes – Articles and AV. They include constructor for document information and also getters and setters for them.
2. Class Patron was modified:
  - a) Was changed method checkout(). It was divided on 3 different methods for different type of document. Now, if there are no copies of document you will stand in priority queue.
  - b) Were added methods renew(). They are 3 different methods for different type of document.

All this works according to different type of users (including visitor professors)

- c) Were added methods getListOf(). They are 3 different methods that returns list of document that user checked out.

### Package "Activities":

1. Were added extra class ExpListViewPatron for separating view of checkouted books on 3 categories: Checkouted, In queue and Fine Documents
2. In class Login there were added notification with information about numbers of fine documents and total price for them
3. In class Return System we can see all users who are checkout something, can see the list of their documents and can confirm that they are returned
4. From Librarian View we can have interface to add/modify users/documents\

Package "Dialogs":

There are classes to represent view of special android widgets called Dialogs