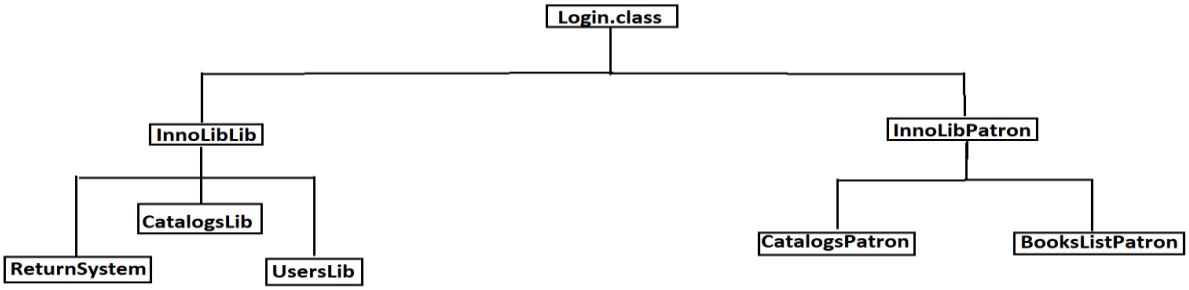
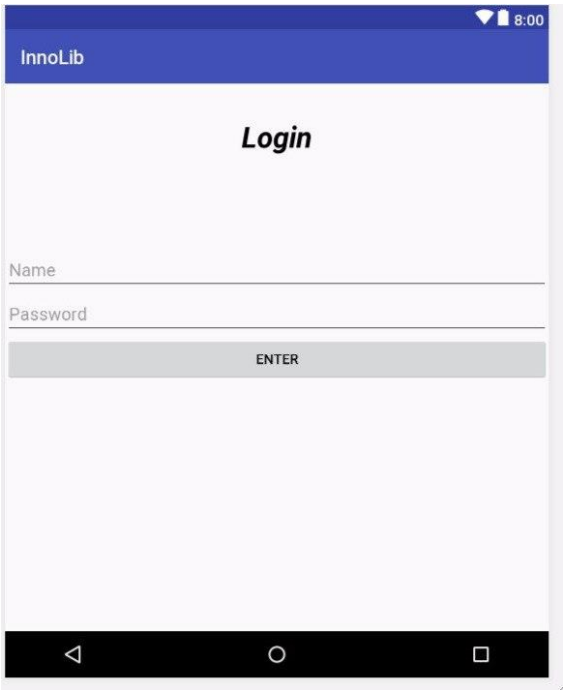
Package Activities (all classes extends from Activity):

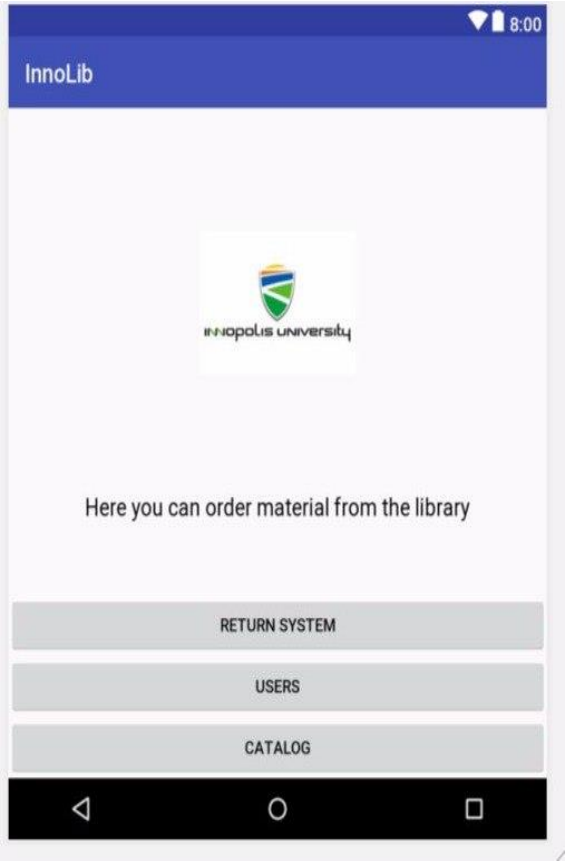


Class Login:



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

Class InnoLibLib:

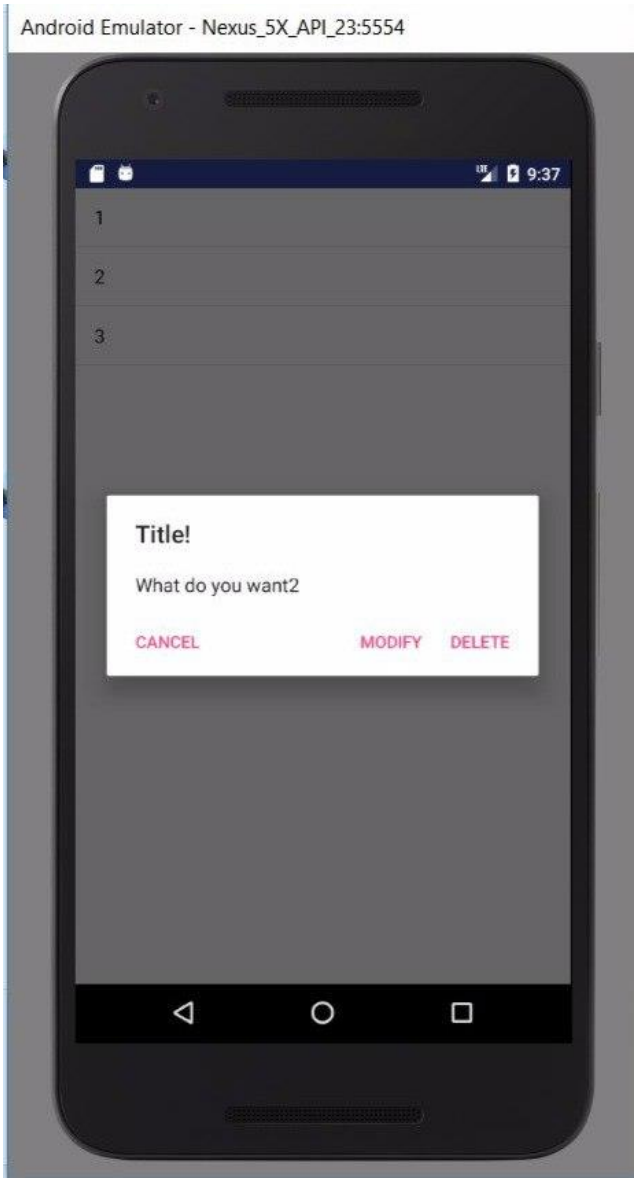


Here the same situation as in previous class : we have 3 buttons and 3 **OnClickListener** for each of them. Button "**RETURN SYSTEM**" will switch current activity on *ReturnSystem* class, "**USERS**" on *UsersLib*, "**CATALOG**" on *CatalogsLib*.

Class ReturnSystem:

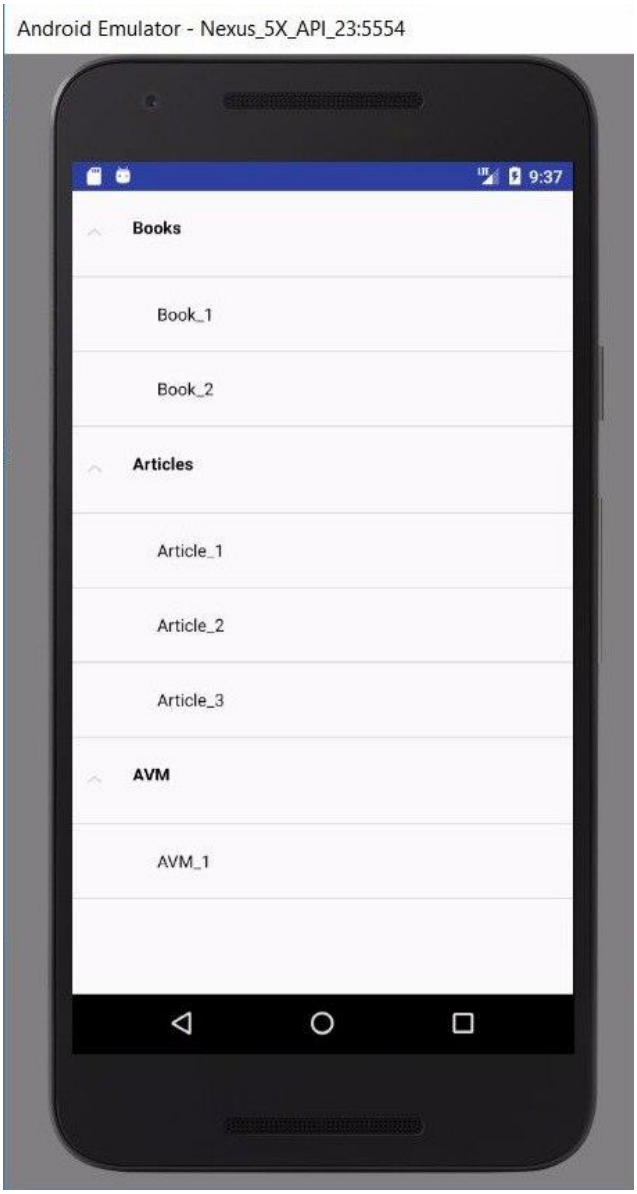
In this class we have **ListView** which contains a list of ordered books from DB. For this list we have method **OnItemClickListener** and when we tap on any element in this list there is append a special **AlertDialog** window where we can confirm recieved book and then delete it from DB.

Class UsersLib :



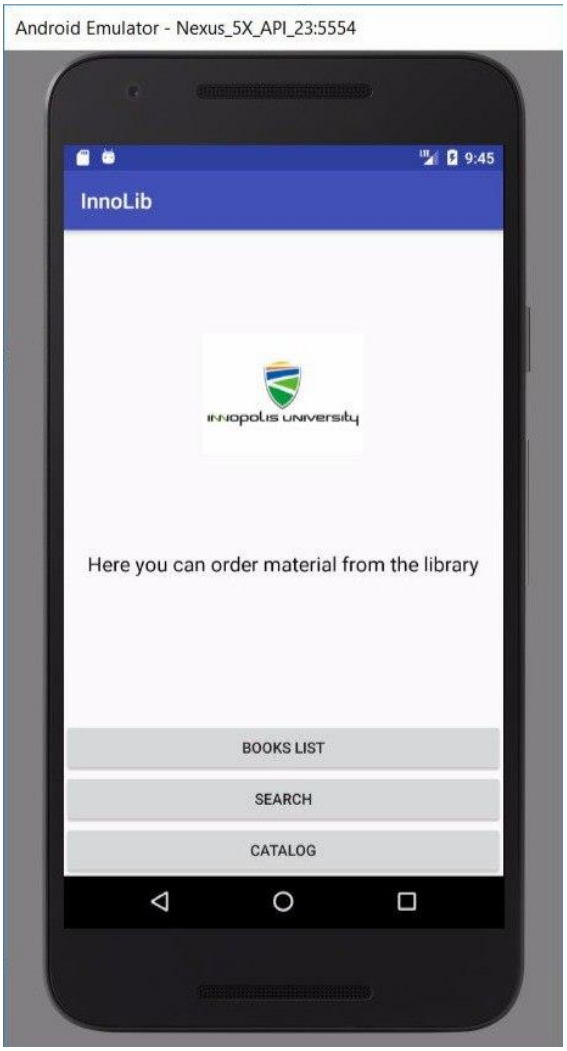
Here we have a **ListView** of all users For this list we have method **OnItemClickListener** and when we tap on any element in this list there is append a special **AlertDialog** window where we can **modify** or **delete** current user.

Class CatalogsLib:



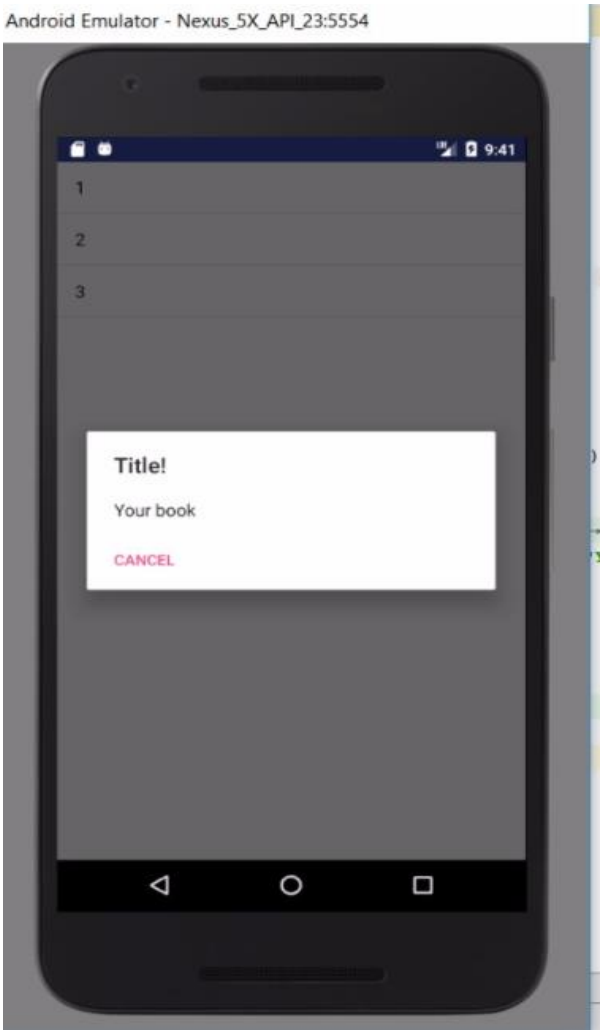
In this class we have **ExpandableListView** where we see all available Documents.
In the future we wil be able to **modify,delete and add Documents** like in *UsersLib*

Class InnoLibPatron:



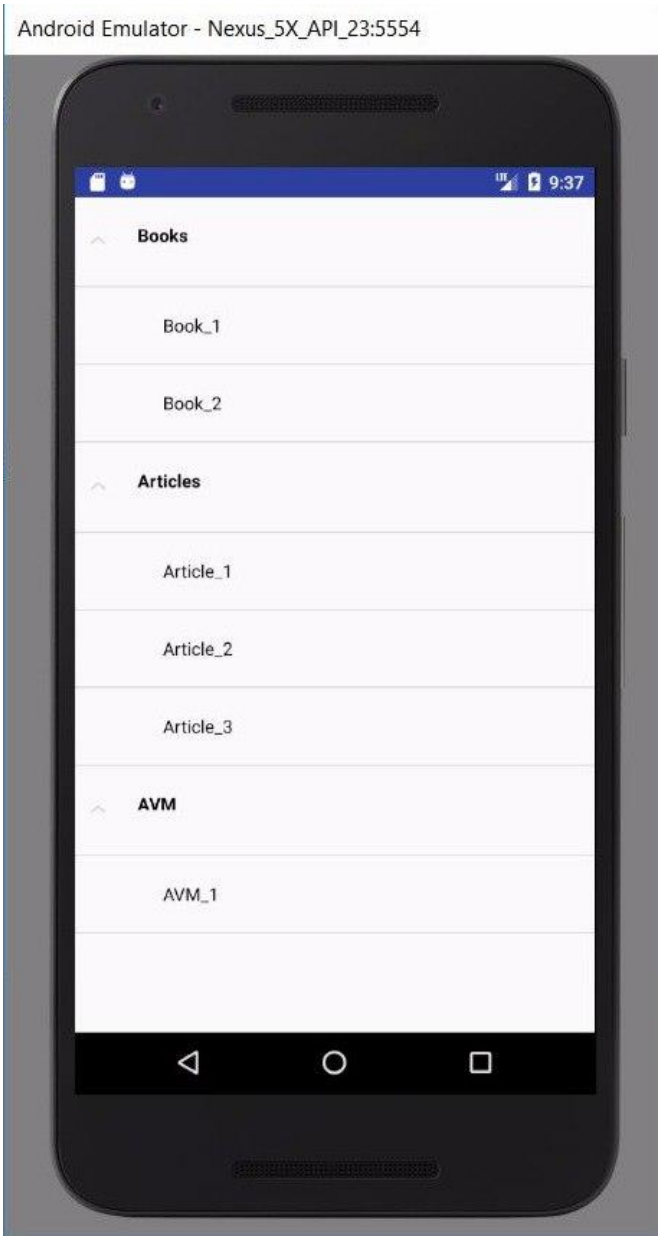
Here we have 3 buttons which have the same principe as **InnoLIbLib.**

Class BooksListPatron:



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 were each element is field o f the book or arguments that are fields of the book. It contains all the informati on about one book that is in data base.

Class Librarian(extends UserCard):

Class librarian contains constructor through array or inputs. It contains informa tion about one user and also contains method that returns all patrons from dat a 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 lin e that signals that this user took book one exactly book. Also it contains metho d 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 thi s method is book id. Return true or false

Class UserCard:

Class usercard contains constructor through array or inputs. It contains informatio n 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 m ethods 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 ti me 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 metho d
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 meth od add new book to the database.

9) Method *addArt*:

Arguments of this method are all information about Article. This met hod 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 argum ent is id of the book that we want update, other arguments new infor mation of the book that will change with old information.

16) Method *updateArticle*:

This method update information about article in database. First argu ment is id of the article that we want update, other arguments new in formation 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 informatio n of the AV that will change with old information.

18) Method *updateUser*:

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

19) Method *getArrayUser*:

This method return array that contain information about User, each el ement of this array contain each field of User. Argument of this metho d 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 e lement of this array contain each field of Book. Argument of this meth od 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 o f 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 check s whether the user has this book. Return true or false.
26) Method *updateTimeChecker*:
This method update table in database “TimeChecker” that helps to ch eck how much time is left for the user to return the book. 4 argument s: user id, book id, time that will be new time, type – type of documen t(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 tha t is debtors.