Block Diagram
Novello

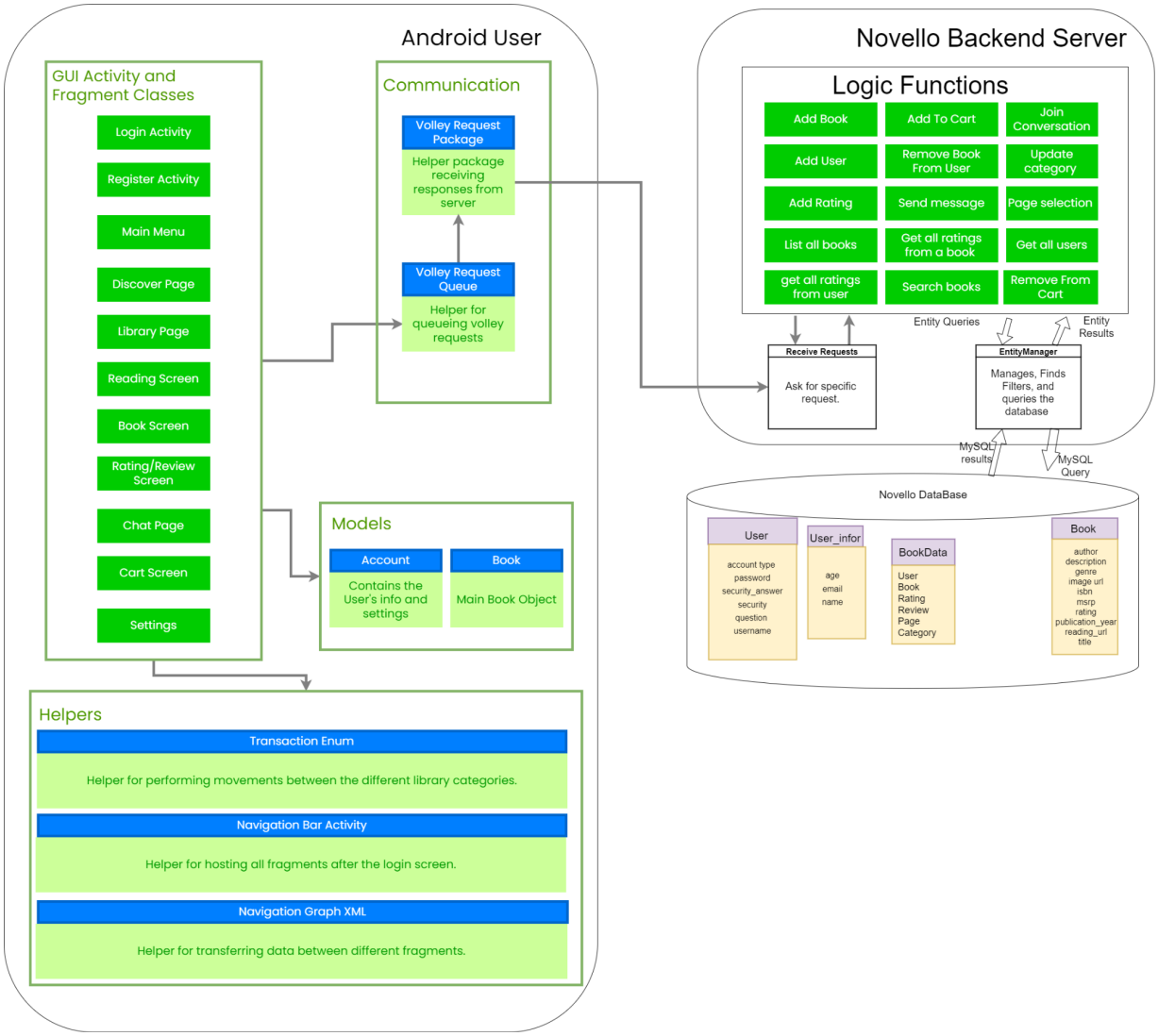
1_yn_1

Kyle Geerts

Kyle Goben

Roba Abbaiabal

Maxim Popov


Android User
GUI Activity and Fragment Classes
Login Activity
Register Activity
Main Menu
Discover Page
Library Page
Reading Screen
Book Screen
Rating/Review Screen
Chat Page
Cart Screen
Settings
Communication
Volley Request Package
Helper package receiving responses from server
Volley Request Queue
Helper for queueing volley requests
Models
Account
Contains the User's info and settings
Book
Main Book Object
Helpers
Transaction Enum
Helper for performing movements between the different library categories.
Navigation Bar Activity
Helper for hosting all fragments after the login screen.
Navigation Graph XML
Helper for transferring data between different fragments.
Novello Backend Server
Logic Functions
Add Book
Add To Cart
Join Conversation
Add User
Remove Book From User
Update category
Add Rating
Send message
Page selection
List all books
Get all ratings from a book
Get all users
get all ratings from user
Search books
Remove From Cart
Entity Queries
Entity Results
Receive Requests
Ask for specific request.
EntityManager
Manages, Finds Filters, and queries the database
MySQL results
MySQL Query
Novello DataBase
User
account type
password
security_answer
security question
username
User_infor
age
email
name
BookData
User
Book
Rating
Review
Page
Category
Book
author
description
genre
image url
isbn
msrp
rating
publication_year
reading_url
title

Design Descriptions

Android User GUI (Maxim)

The user begins in the Login activity and can then navigate to either the Create Account or Main Menu Activity, the latter by providing login credentials. All other screens are fragments built upon the Main Menu activity. Those are the Discover, Library, Reading, Book, Rating/Review, Chat, Cart, and Settings fragments. Each activity and fragment has a corresponding XML file defining its user-interactable contents. Each activity and fragment also has helper classes, primarily presenters and models, to do computations and obtain information from the database.

Android Code Helpers (Roba)

There are three helper classes to help us program our application: the transaction enumerated type, the navigation bar activity class, and the navigation graph XML file. The navigation activity XML file contains all fragments to be included in our application. It defines each screen transaction to be made between fragments, as well as defining the arguments to be transferred between the two specified fragments. The transaction enumerated class defines all categorical relationships that a user can hold with a book: none, wishlist, cart, library, backlog, currently reading, and read. It also contains methods that can perform Volley transactions to switch books between user categories. Lastly, the navigation bar activity is a class that hosts fragments. It initializes our bottom navigation bar, holds the current user that is logged in (obtained from login activity), and contains a controller that can be used to navigate between different fragments in our UI.

Android Models (Maxim)

We have two primary models that are frequently used in our application: Account and Book. The former defines a user account and stores all necessary information relating to it that cannot be easily obtained from the database. The foremost among that information is the user ID which is only received upon successful login. The Book class defines a book and contains all data related to an individual book. This is used to display information about the book and to do computations regarding book fields.

Android Communication (Roba)

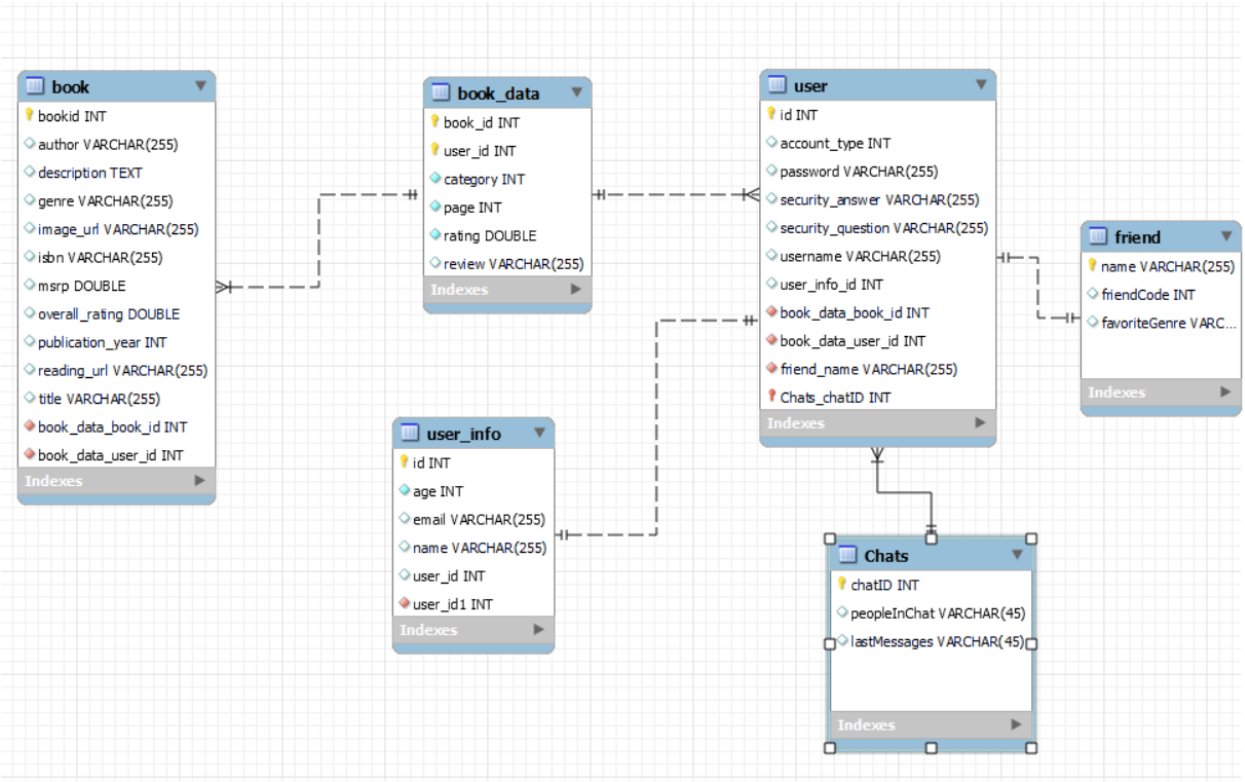
For a fragment to communicate with the server, a fragment's model class must first instantiate an implementation of Requester in our Volley request package. There are four types of requests based on the data being sent or received: String Request, JSON Object Request, JSON Array Request, and Image Request. After instantiating one of these four requesters, the user can intuitively call a method named after the HTML request type (e.g. GET, PUT, DELETE). This request gets sent to our volley request queue, and once a response is received from our server, it calls the implemented VolleyCommand to be run.

Backend Databases (Kyle)

The user table is used to store new users while the userInfo table is used to store extra specific information for a selected user. The Book Table is used to add and store new books and all of their information as an administrator. While BookData is used to store all of the book information a user has interacted with such as what books are in their cart, the library, currently reading what page they are on and the reviews and ratings they have left on books they have read.

Backend Logic Functions (Goben)

Our logic functions are a large number of the requests that are used to modify/update the database. Using requests can update the book, user, userInfo, or bookInfo accordingly to the type of update that the user needs. They can make new, update, or delete entries in any of the 4 tables.



One to one = User to User_Info: This relationship is used to create a connection between the user and their about page that stores their name age and email address.

Many to many = Users to Books: Every user can have multiple books and every book can be in multiple user libraries.

One to many = User_Review to Books: Each user can only place one review on every book that they have in their user library.

One to many user to friend: Each user can have amny friends.

One to many: Each user can be in many chats but each chat can only have one instance of each user.