BOOK BARTER

CSE 5324: SOFTWARE ENGINEERING: ANALYSIS, DESIGN, AND TESTING

PROJECT GROUP 12: THE FIVERS

Iteration 3 - 12/04/2023

Team Members

Varad Nair
Kushwanth Sai Bollepalli
Srujan Chinta
Niharika Dandu
Sreelakshmi Mopuri

Table Of Contents

APP TITLE: BOOK BARTER

"Book Barter" is the solution to the age-old quest for the perfect book exchange. It connects book enthusiasts, addresses the need to find, exchange, and explore books effortlessly in a sustainable development way. Users register to establish a personalized reading identity, searching and browsing allows them to locate their next literary adventure. Messaging facilitates seamless communication and location services make in-person exchanges simple. Books Rating empowers readers to make informed choices, and safety & privacy features ensure secure interactions. Wishlist Management guarantees one will never miss a desired read. In short, "Book Barter" transforms book sharing into a hassle-free, enjoyable experience.

Function 1: User Registration & Profile Creation:

- Allows new users to sign up and existing users to log in. Profiles may include personal reading preferences (comedy, thrillers, fiction, etc.), book collection lists, and ratings or reviews which can be edited.

Function 2: Search & Browse Capability:

- This enables the users to search for specific titles, authors, or genres and browse through available books based on categories or user recommendations.

Function 3: Matching System:

- Automatically matches users who have books desired by others, facilitating potential exchanges based on book availability and genre preferences.

Function 4: Direct Messaging:

- Aids users to communicate directly, discussing book conditions, barter terms, or arranging meetup/pickup location details.

Function 5: Location Services:

- Empowers the users to find potential barter partners nearby themselves, offering convenience and reducing the need for long-distance exchanges.

Function 6: Book Reviews & Ratings:

- Allows the lender to rate, review and post books they have read, offer insights, and help others make informed barter choices.

Function 7: Notification System:

- Notifies users of potential matches, new book listings in their preferred genre, or when someone expresses interest in their listed books. Users can unsubscribe from the notification service at any time.

Function 8: Barter Confirmation:

- Once a barter is agreed upon, users can confirm the exchange and update the status of the book(available or not), ensuring transparency and accountability.

Function 9: Safety & Privacy Features:

- This function ensures users' personal details remain confidential, offering features like safe meeting point suggestions and anonymous browsing.

Function 10: History & Wishlist Management:

- Users can keep track of past barters and manage or update their Wishlist of books they're interested in.

Function 11: Postal services:

- This function introduces an option for users to request a book by using postal services like FedEx. To maintain transparency and integrity, user ratings and reviews are employed to gain trust.

Resources used for the application:

Database:

- Specifically, Firebase's Cloud Fire store is used to organize and provide data to Users in real time.

Camera:

- Camera service provided by Android is utilized for capturing the picture of the book for reference.

Location Services:

- Location service offered by android is utilized for knowing the pickup location of the book lender.

Notification Services:

- Notification service is utilized to notify about new books that match their genre, the status of their book request.

Team's Introduction:

Kushwanth Sai Bollepalli: During my under graduation, with my enthusiasm for learning Android development, I created a few mini Android applications using Android Studio and Java. Subsequently, as a member of the Entrepreneurship Cell, a student club, I designed an Android app to streamline event registration and provide event details for an event called E-Summit. By working on that Android application, I got familiarized with the Firebase's Cloud Firestore database, Firebase Authentication, fragments, Intents, and various layout configurations.

Srujan Chinta: As a recent undergraduate with a strong passion for Android development, I've gained valuable experience through coursework, culminating in a comprehensive E-commerce app project. This project improved my knowledge of Java, Kotlin, and Android Studio while also exposing me to crucial concepts like UML design, user authentication, and user profiles. I've successfully included user profiles, UML design, and authentication into a simple e-commerce

application. I created an organized and scalable application architecture using UML design principles. I also included user profiles, which enhanced the app's personalization and provided users with a seamless experience.

Niharika Dandu: During my undergraduate studies, I worked on an Android Studio project and created an app called Bus Pass Management System. Having a passion for creating unique and user-friendly mobile applications as an Android app developer. Using the robust platform, Android Studio, I've had the luxury of navigating the fascinating and always-changing field of Android app development. I am knowledgeable in both Java and Kotlin, which are the two main programming languages utilized in the creation of Android apps. I can write readable, streamlined, and maintainable code in both languages. My goal in developing apps was to transform original concepts.

Sreelakshmi Mopuri: During my undergrad, I started learning how to create apps. I became intrigued by the potential of mobile technology to have a significant positive impact on people's lives from the beginning, which drove me to create the "Toll Gate App For Android-Based Payment" app. To develop this app, I used the tools Android Studio, databases- Firebase Real-Time Database & SQLite/Room, and Java programming language. In addition, I excel in problem-solving, Testing and Debugging, Firebase, documentation, user experience design, API integration, App Deployment, version control, and database administration.

Varad Nair: With my passion for learning, during my under graduation, I created a few mini and a web app as a major project. It was a ‘Cloud Based Notes and Assignment Sharing system’ with the help of Web Development technologies, Postgres, and Django(Python) built on the ORM Model. I also have almost 2 years of work experience not only limited to development but analytics domain as well. I have knowledge of Java, Object Oriented Concepts, UML, hands-on Agile methodology, and DBMS. My critical assets are self-confidence, exposure to effective leadership, and problem-solving skills.

Requirements:

Req ID	Requirement Description	Line Reference
R1	The system shall provide a registration / sign-up option for the new users	10
R2	The system shall provide a log-in option for existing users	10
R3	The system shall allow users to reset their password in case they forget	Derived
R4	The system shall allow users to include or edit their personal reading preferences like Sci-Fi, Comedy, Thriller, Detective, Romantic, Horror, History, Comic in profile section to receive notifications.	10 to 12
R5	The system shall allow users to search for specific books based on book title, author name, or genre.	14
R6	The system shall provide an option for users to request or initiate exchange	Derived
R7	The system shall provide a chatting facility for users to discuss barter terms, book condition and meeting/pickup locations	20, 21
R8	The system shall access the location service to find potential barters nearby, willing to exchange	22, 23
R9	The system shall facilitate Book Lender's to post new books along with their review and rating.	25 to 27
R10	The system shall notify users about new book listings in their preferred genre, status of their request.	29, 30
R11	The system shall allow users to unsubscribe from notification service	30, 31
R12	The system shall update the availability of book once barter is agreed upon	33, 34
R13	The system shall store and provide information about past barters	39
R14	The system shall provide an option for users to request book, by using postal services like FedEx instead of fixing a pickup location	42, 43

R15	The system shall let users provide ratings based on their experience with other users	43

Use Case List:

Use Case #	Use Case Name
UC1	Register
UC2	Login
UC3	Update Profile
UC4	Book Search
UC5	Post Book
UC6	Initiate Book Exchange
UC7	Send Message
UC8	Barter Confirmation
UC9	Utilize Location Services
UC10	User Rating
UC11	Manage Notifications
UC12	Manage Barter History
UC13	Request Book via Postal Service
UC14	Reset Password

High-Level Use Case:

TUCBW: The use case begins with
TUCEW: The use case ends with

- **UC1: Register**

  - TUCBW user providing registration details (Name, Email, Phone Number, Password) and clicks on Register Button.
  - TUCEW user being registered.

- **UC2: Login**

  - TUCBW user enters credentials (email, password) and clicks Login button
  - TUCEW the User gets signed in and the respective screen is displayed (Wishlist selection or Home Screen).

- **UC3: Update Profile**

  - TUCBW user navigates to profile section to update profile information.
  - TUCEW user updating the details successfully.

- **UC4: Book Search**

  - TUCBW The user navigates to the search bar on the top of home screen and enters Book name or Author Name or Genre to filter posts.
  - TUCEW user getting list of filtered posts that match with keyword.

- **UC5: Post Book**

  - TUCBW user navigates to “Post” navigation tab.
  - TUCEW User posted the book successfully.

- **UC 6: Initiate Book Exchange**

  - TUCBW users express their interest by clicking on the Interested button on the respective book post.
  - TUCEW either user requesting the book or cancel the request.

- **UC7: Send Message**

  - TUCBW User navigates to the search bar and enters the person’s name to discuss book conditions and barter terms.
  - TUCEW The other user receiving the message

- **UC8: Barter Confirmation**

    - TUCBW user click on “View Requests” button.
    - TUCEW Request being accepted, and notification sent.

- **UC9: Utilize Location Services**

    - TUCBW The user navigates to the search bar on the top of home screen and enters a city name to find potential barters in that location.
    - TUCEW user getting list of posts originated from specified location.

- **UC 10: User Rating**
    - TUCBW user clicking on username of other person/ barter at the bottom of the post.
    - TUCEW user rating the other user/ barter.

- **UC11: Manage Notifications**

    - TUCBW user clicks on Manage Notifications on profile screen.
    - TUCEW user subscribed/Unsubscribed to genres that he/she is interested to receive notifications.

- **UC12: Manage Barter History**

    - TUCBW user clicks on “Manage Barters”.
    - TUCEW User managing barter posts.

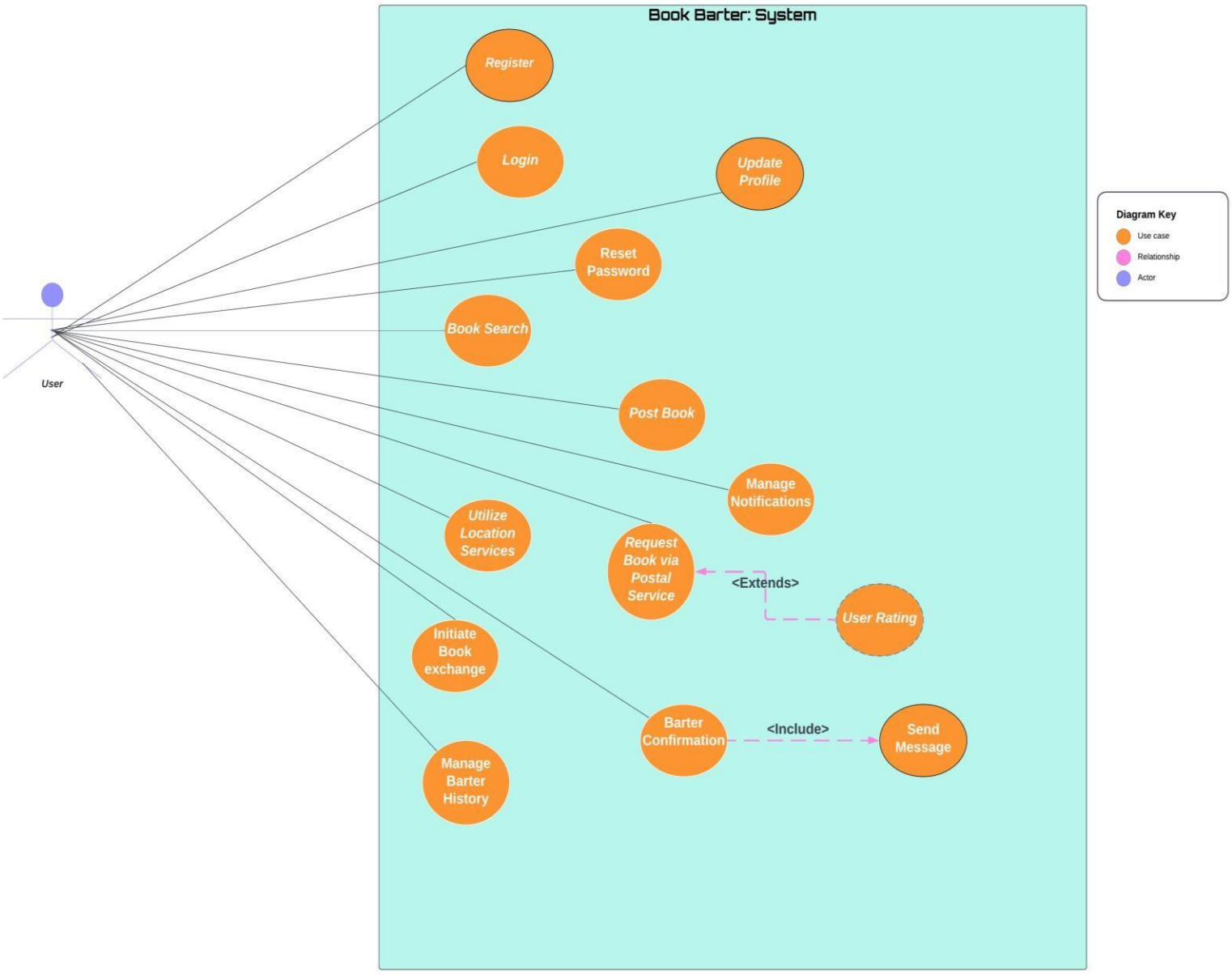
- **UC13: Request Book via Postal Service**

    - TUCBW User clicks on Postal option.
    - TUCEW user opting for postal preference and request sent successfully.

- **UC14: Reset Password**

    - TUCBW the user clicking the forget password link.
    - TUCEW password being updated.

Use Case Diagram:



Requirements to Use Case Traceability Matrix:

Req ID	Priority	UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7	UC 8	UC 9	UC 10	UC 11	UC 12	UC 13	UC 14
R1	2	X													
R2	3		X												
R3	4														X
R4	4			X											
R5	1				X										
R6	1						X								
R7	1							X							
R8	2									X					
R9	1					X									
R10	3					X						X			
R11	4											X			
R12	2								X						
R13	4												X		
R14	1													X	
R15	3										X				
	SCORE	2	3	4	1	4	1	1	2	2	3	7	4	1	4

REQUIREMENTS USE CASE TRACEABILITY MATRIX: Priority weights are assigned such that 1= highest priority weight and 4= lowest priority weight. R= Requirement ID UC = Use Case

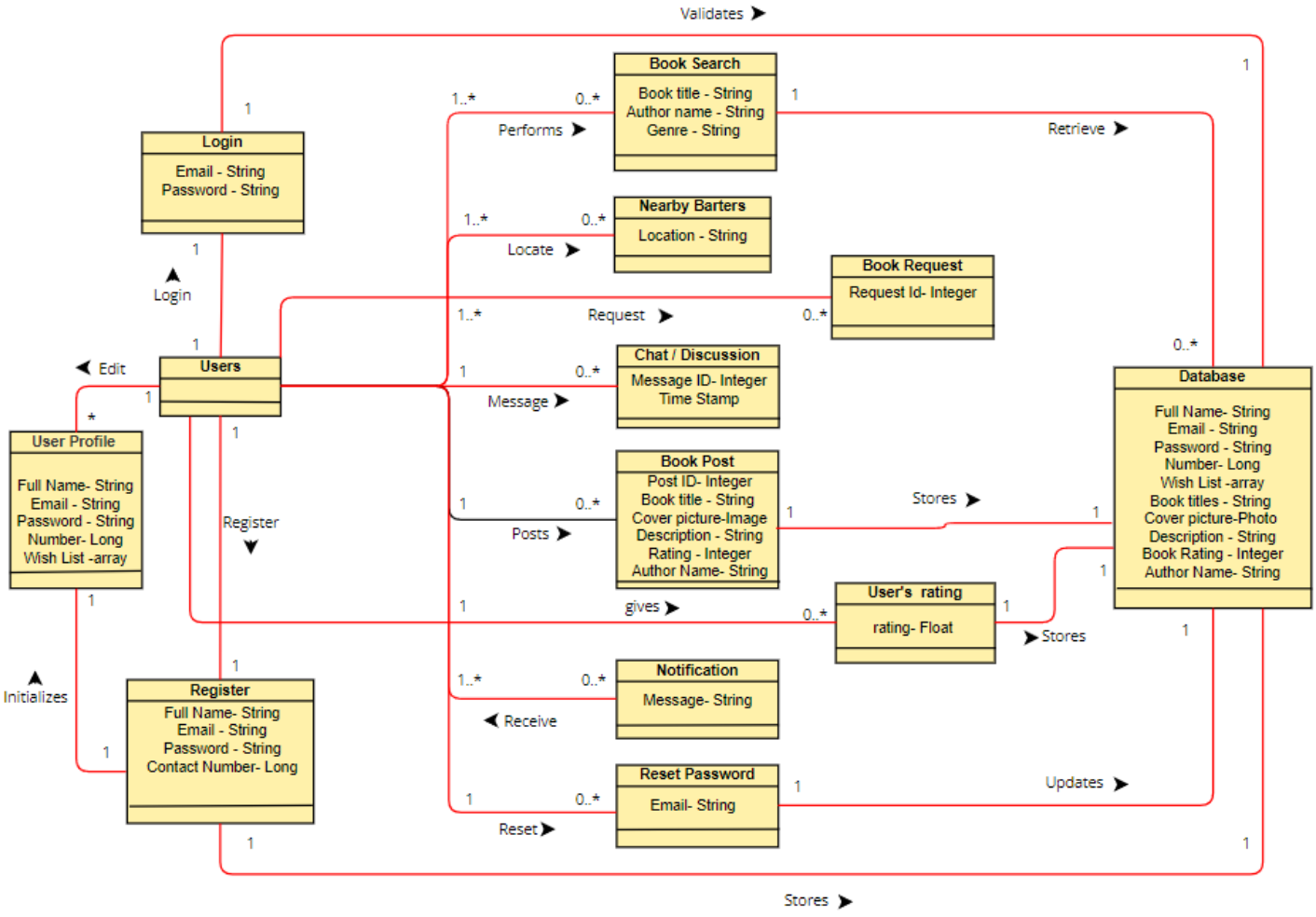
Increment Matrix:

Use Case	Priority	Effort (person-week)	Depends On	Assigned to	Iteration 1 (10-06-23)	Iteration 2 (11-03-23)	Iteration 3 (12-04-23)
UC 1	2	2	None	VN,SM	1	1	
UC 2	3	2	UC1	KB,ND	1	1`	
UC 3	4	2	UC2	SC,VN	1	1	
UC 4	1	4	UC2, UC5	SM,KB		2	2
UC 5	4	5	UC2	SC,ND		3	2
UC 6	1	3	UC2,UC5	VN,KB		2	1
UC 7	1	6	UC2,UC8	SM,ND		2	4
UC 8	2	3	UC2	SC,KB		2	1
UC 9	2	5	UC2,UC4	ND,VN		1	4
UC 10	3	4	UC2, UC13	SM,SC		2	2
UC 11	7	3	UC1,UC5	KB,ND		1	2
UC 12	4	3	UC2,UC8	SC,VN		2	1
UC 13	1	2	UC2,UC8	SM,KB		1	1
UC 14	4	2	UC1	VN,SM	1	1	
Total Effort		46			4	22	20

1 Person-Week = 5 hrs.

Team Members:
VN- **V**arad **N**air
KB- **K**ushwanth Sai **B**ollepalli
SC- **S**rujan **C**hinta
ND- **N**iharika **D**andu
SM- **S**reelakshmi **M**opuri

Domain Model:



Expanded Use Cases:

EUC1: Register

Precondition: This Use Case assumes user is on the Launcher Screen and wants to Register.	
Actor – User	**System** – Book Barter
1.TUCBW user providing registration details (Name, Email, Phone Number, Password) and clicks on Register Button.	**0.** System displays Registration screen (See Figure 1a).
	***2.** The System validates user input. If the validation is successful, the system displays "Successfully Registered" and redirects to Login screen (See Figure 1b) and sends the verification link to user's email (See Figure 1c).
3.User Clicks on Verification Link	
	4.The System verifies the User and displays " Your Email has been Verified" (See Figure 1d).
5.TUCEW user being registered.	
Postcondition: User is registered for application and can login	

Fig. 1a

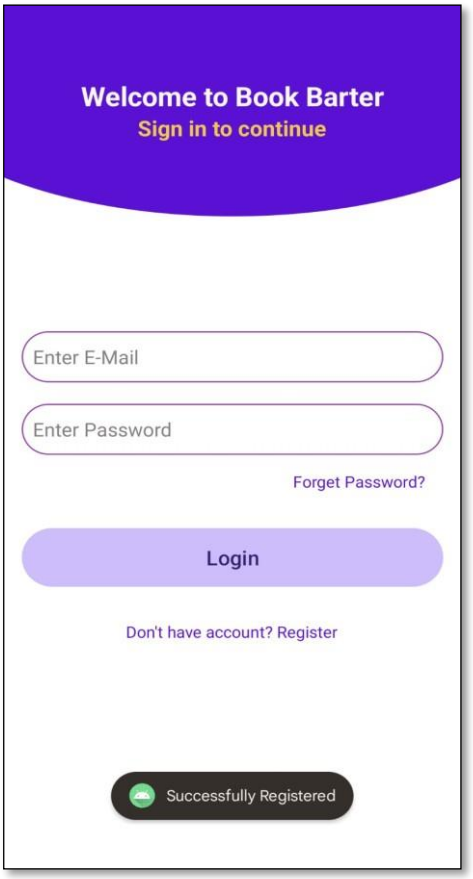


Fig. 1b

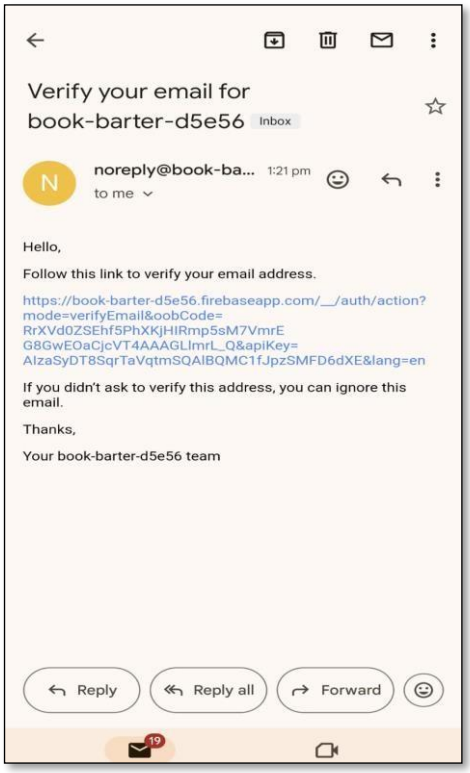


Fig. 1c

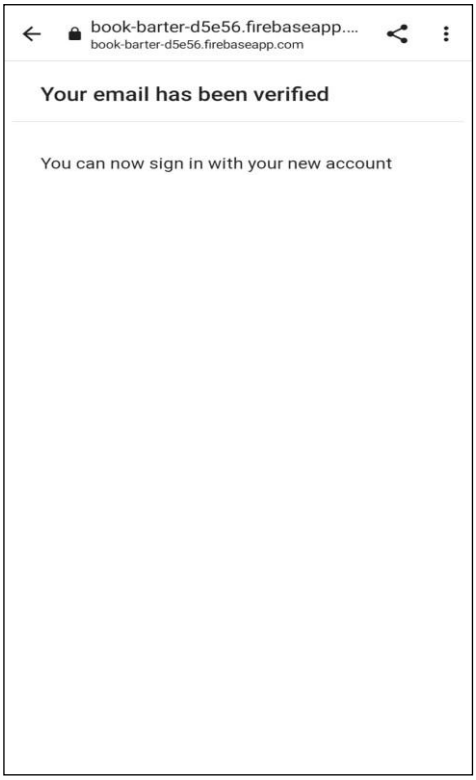


Fig. 1d

EUC2: Login

Precondition: The user should be registered and verified email	
Actor – User	**System** – Book Barter
1.TUCBW user enters credentials (email, password)and clicks Login button	**0.** System displays login screen (See Figure 2a).
	*2. The System validates & verifies credentials and if the User is verified, system displays "Logged In Successfully"and redirects to Wishlist Selection screen (See Figure 2b) ifthe user is logging in for the first time, else to the home screen. If not verified, system displays "Please Verify email to continue" (See Figure 2c).
3. TUCEW The User gets signed in and the respectivescreen is displayed (Wishlist selection or Home Screen).	
Postcondition: User is logged in to the application	

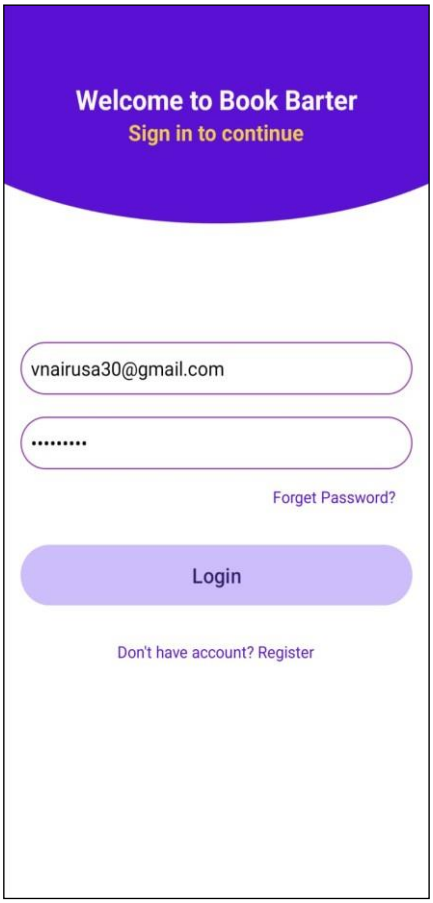


Fig. 2a

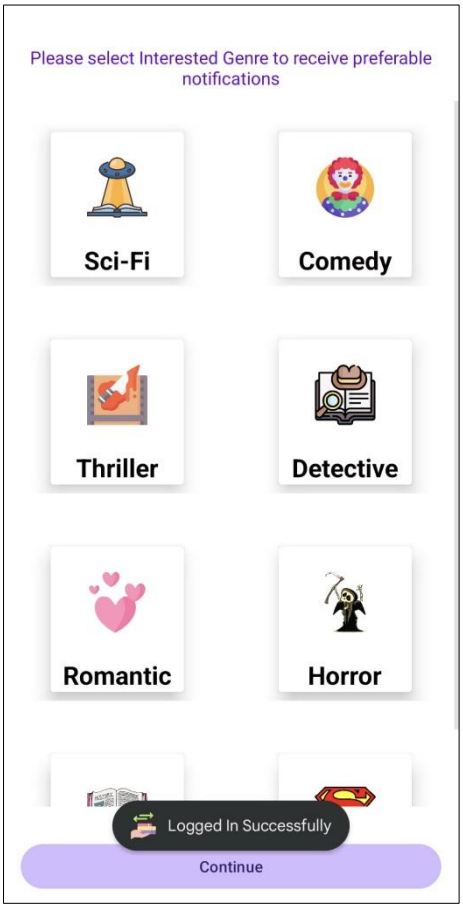


Fig. 2b

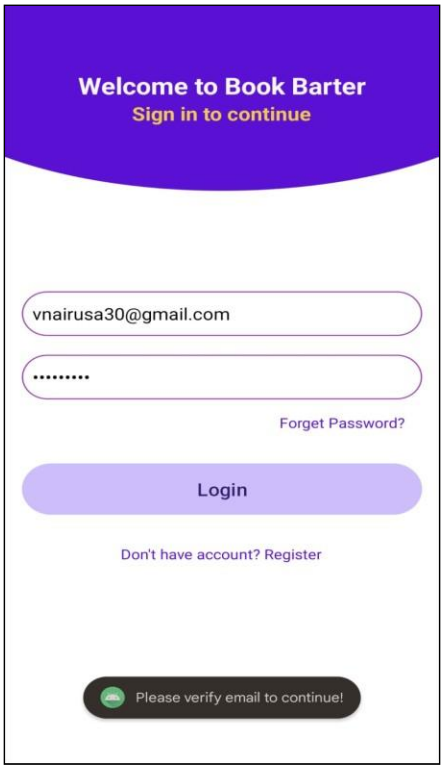


Fig. 2c

EUC3: Update Profile

Precondition: This use case assumes user is on the home screen and wants to update details.	
Actor – User	**System** – Book Barter
	0. System displays home screen (See Figure 5a).
1.TUCBW user navigates to profile section to update profile information.	
	***2.** The System displays user profile screen with user details along with an option to upload image and edit mobile number (See Figure 3a).
3.The user clicks a) Upload Image and selects Image from gallery b) Edit Number and enters new Phone Number	
	4.The system a) updates profile pic and displays "Uploaded Successfully" (See Figure 3b). b) updates phone number and displays "Updated Successfully" (See Figure 3c).
5. TUCEW user updating the details successfully.	
Postcondition: Updated user profile is visible to the user	

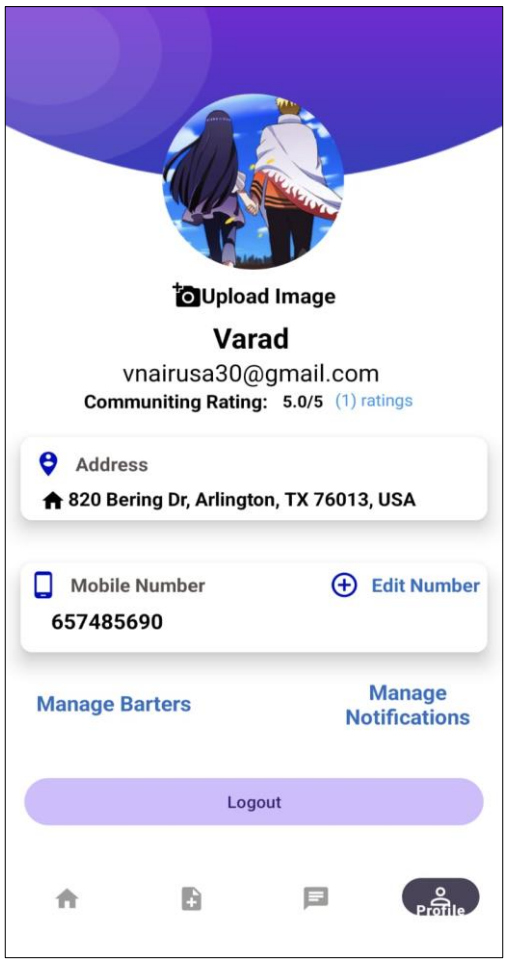


Fig. 3a

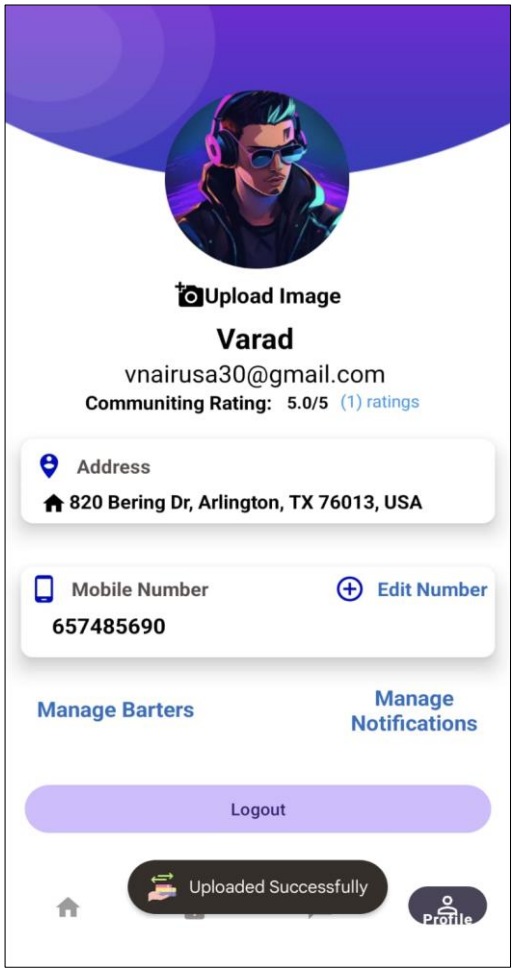


Fig. 3b

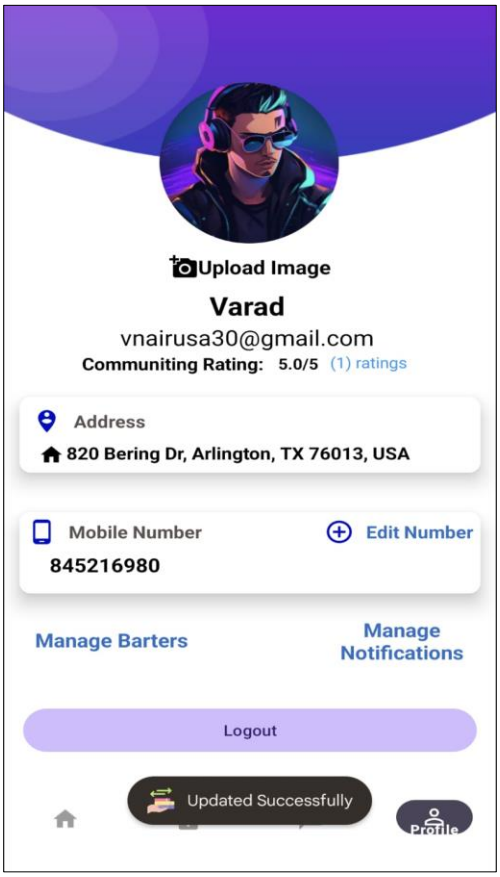


Fig. 3c

EUC4: Book Search

Precondition: This use case assumes user is on the home screen and wants to search book	
Actor –User	**System -** Book Barter
	0. System displays home screen (See Figure 5a).
1.TUCBW The user navigates to the search bar on the top of home screen and enters Book name or Author Name or Genre to filter posts.	
	***2.**The system displays a list of book posts that match with the keyword (See Figure 4a) , else "Search result not found" toast is displayed (See Figure 4b)
3.TUCEW user getting list of filtered posts that match with keyword.	
Postcondition: The user finds anticipated book	

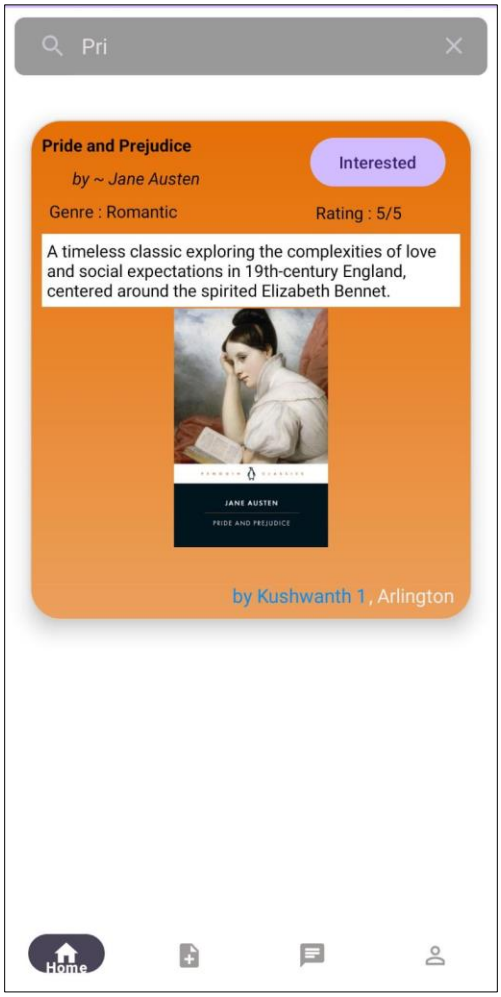

Pri
Pride and Prejudice
Interested
by ~ Jane Austen
Genre : Romantic
Rating : 5/5
A timeless classic exploring the complexities of love and social expectations in 19th-century England, centered around the spirited Elizabeth Bennet.
JANE AUSTEN
PRIDE AND PREJUDICE
by Kushwanth 1, Arlington
Home

Fig. 4a

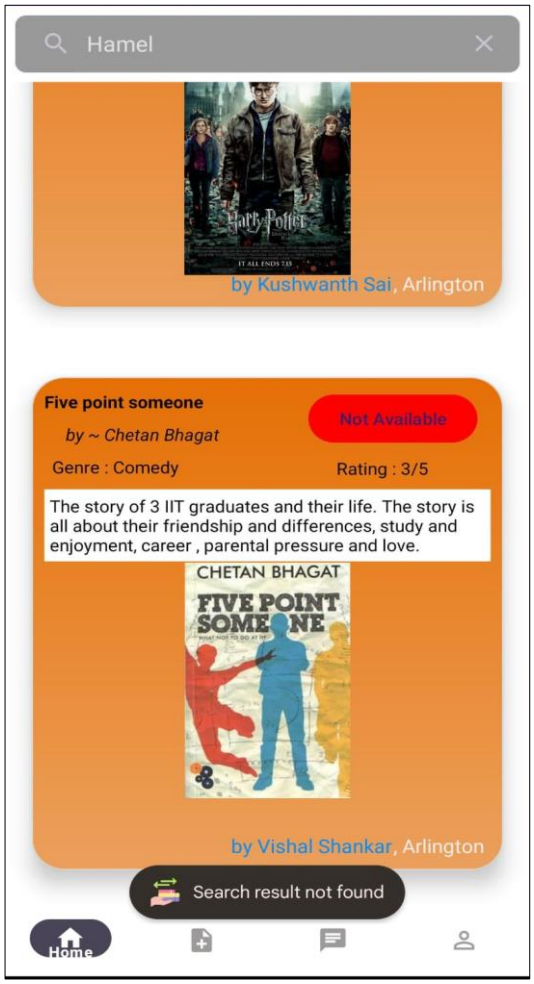

Hamel
by Kushwanth Sai, Arlington
Five point someone
Not Available
by ~ Chetan Bhagat
Genre : Comedy
Rating : 3/5
The story of 3 IIT graduates and their life. The story is all about their friendship and differences, study and enjoyment, career , parental pressure and love.
CHETAN BHAGAT
FIVE POINT SOMEONE
by Vishal Shankar, Arlington
Search result not found
Home

Fig. 4b

EUC5: Post Book

Precondition: This Use Case assumes user is on the Home Screen and wants to Post Book.	
Actor – User	**System** – Book Barter
1.TUCBW user navigates to "Post" tab	**0.** System displays Home screen of the logged in User (See Figure 5a).
	2.The system displays a screen to input details: Book title, Author name, Book description, Genre, image of the Book, Book Owner's rating for the Book (See Figure 5b).
3. The user provides details (Book title, Author name, Book description, Genre, image of the Book, Book Owner's rating for the Book) and clicks on Submit button.	
	***4.** The System validates user input. If the validation is successful, the system displays "Uploaded Successfully" (Figure 5c) message and send notification to respective subscribed users of the genre (Figure 5d) else system displays error message in the respective field (See Figure 5e).
5. TUCEW User posted the book successfully.	
Postcondition: post is available in users feed	

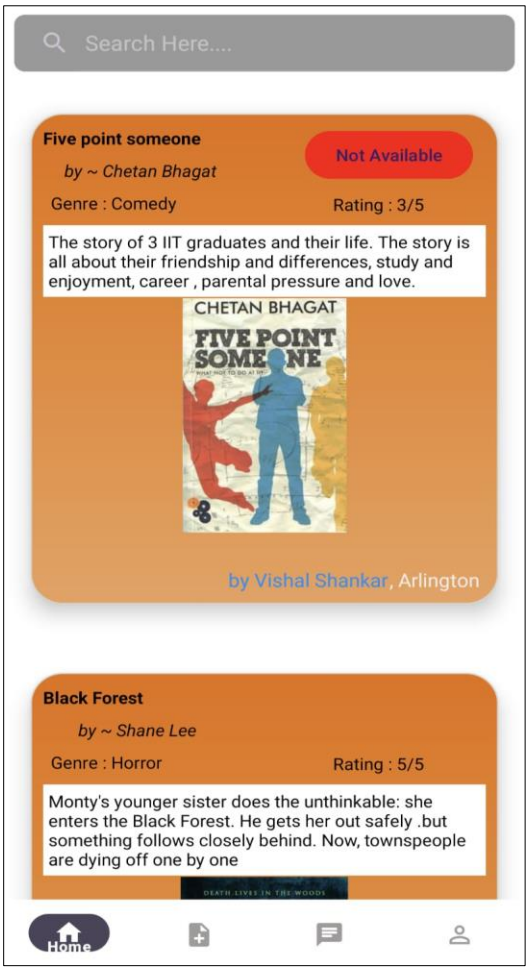


Fig. 5a

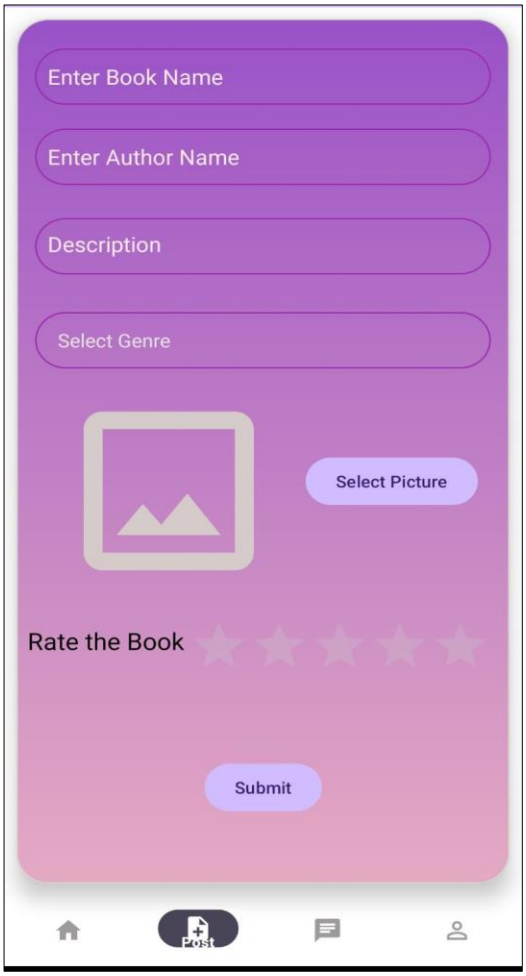


Fig. 5b

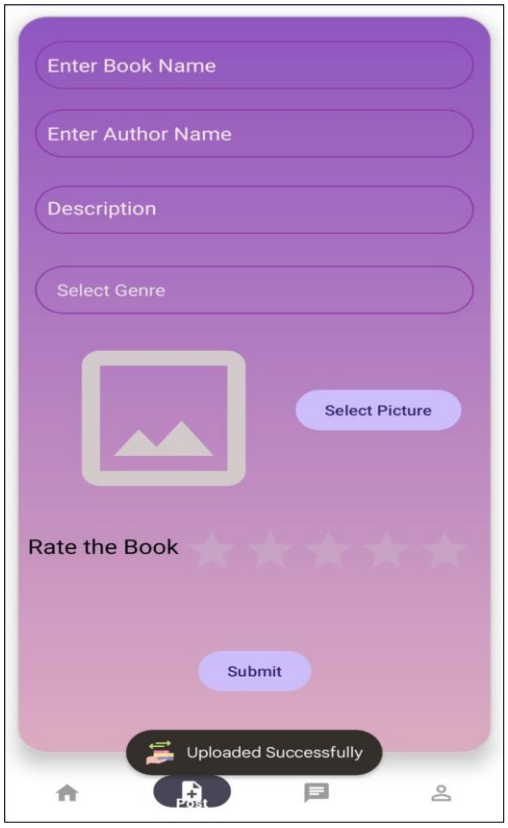


Fig. 5c

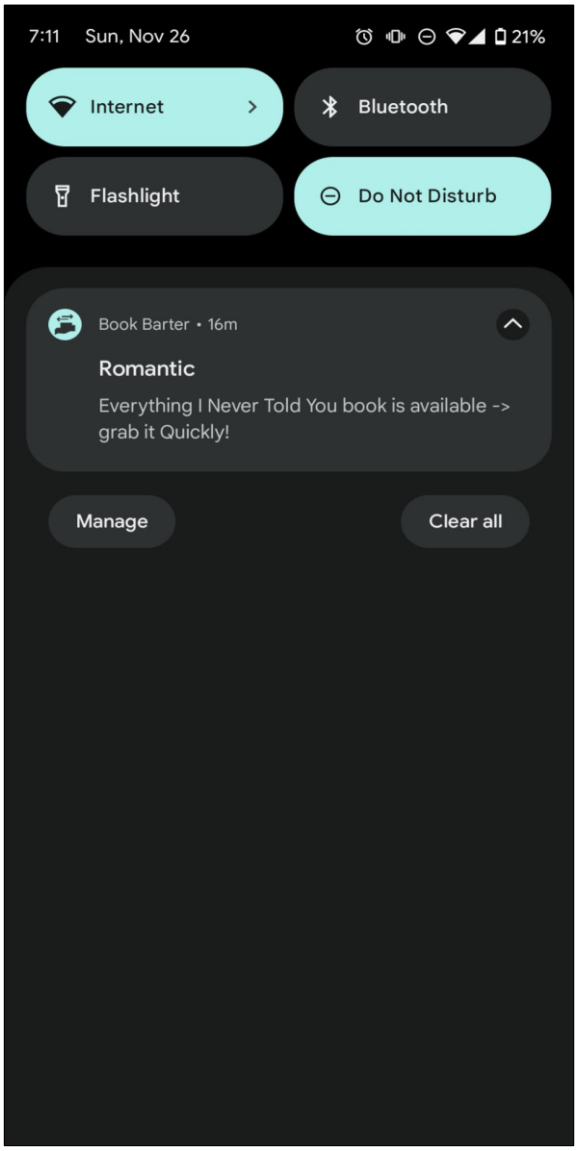

7:11 Sun, Nov 26
21%
Internet
Bluetooth
Flashlight
Do Not Disturb
Book Barter • 16m
Romantic
Everything I Never Told You book is available -> grab it Quickly!
Manage
Clear all

Fig. 5d

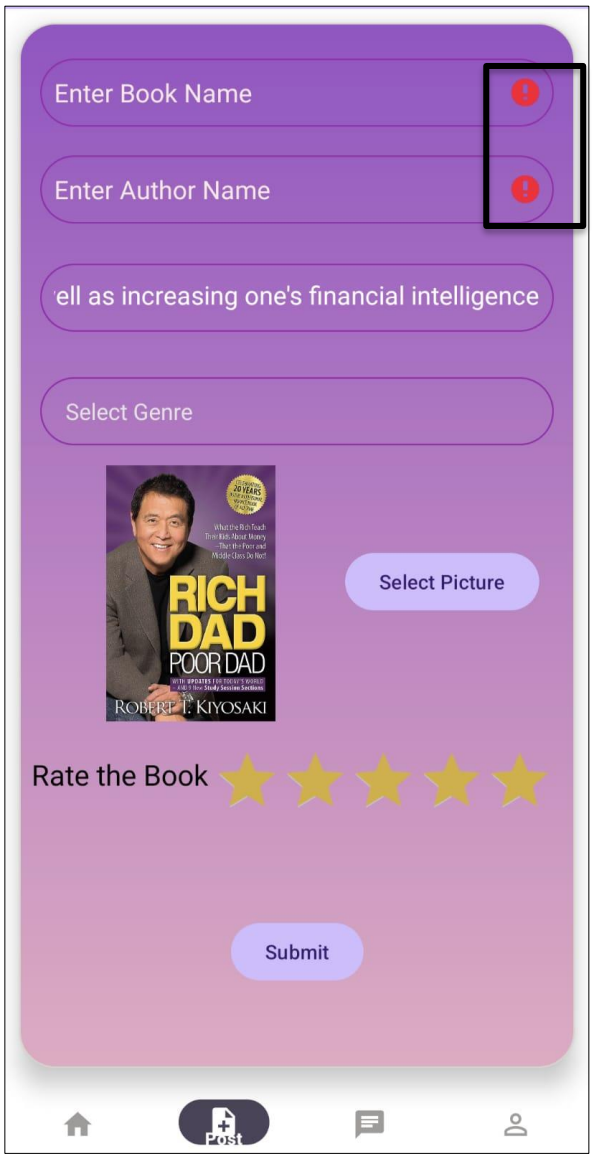

Enter Book Name
Enter Author Name
ell as increasing one's financial intelligence
Select Genre
RICH DAD POOR DAD
ROBERT T. KIYOSAKI
Select Picture
Rate the Book
Submit
Post

Fig. 5e

EUC 6: Initiate Book Exchange

Precondition: This use case assumes User is on Home screen and intends to request for a book.	
Actor – User	**System** – Book Barter
	0. System displays home screen (See Figure 5a).
1.TUCBW users express their interest by clicking on the Interested button on the respective book post.	
	2. The system validates user requests and displays a dialog box to select their preferences between postal & in-person, or even cancel if not sure (See Figure 6a).
3.The user clicks on a) either In-Person or Postal b) cancel	
	4.The System *a) send the user request along with user details to the book/post owner or barter and displays "Request sent successfully" (See Figure 6b). b) cancels the request
5.TUCEW either user requesting the book or cancel the request.	
Postcondition: User request will be available for that book.	

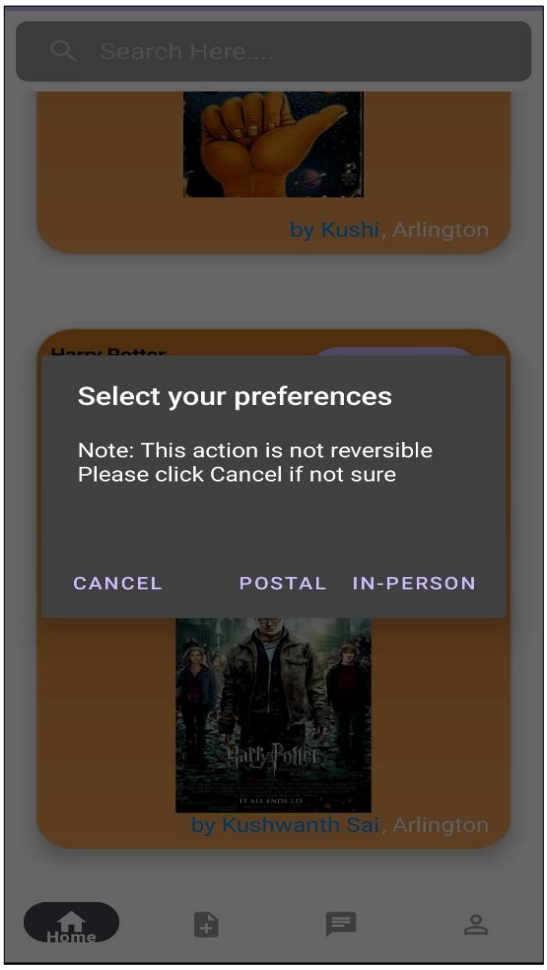

Search Here....
by Kushi, Arlington
Select your preferences
Note: This action is not reversible
Please click Cancel if not sure
CANCEL
POSTAL
IN-PERSON
by Kushwanth Sai, Arlington
Home

Fig. 6a

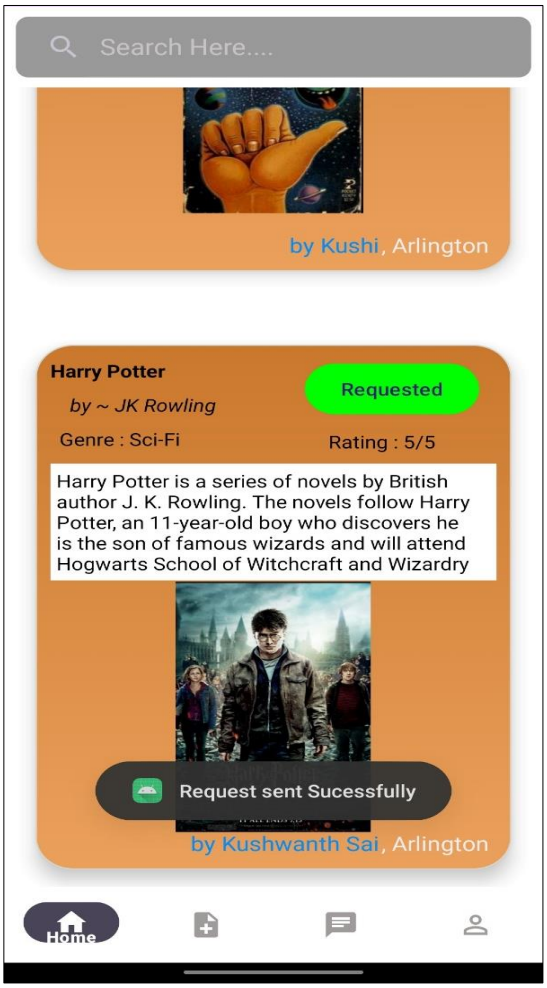

Search Here....
by Kushi, Arlington
Harry Potter
by ~ JK Rowling
Requested
Genre : Sci-Fi
Rating : 5/5
Harry Potter is a series of novels by British author J. K. Rowling. The novels follow Harry Potter, an 11-year-old boy who discovers he is the son of famous wizards and will attend Hogwarts School of Witchcraft and Wizardry
Request sent Sucessfully
by Kushwanth Sai, Arlington
Home

Fig. 6b

EUC 7: Send Message

Precondition: This use case assumes User is on Recent chat screen	
Actor – User	**System** – Book Barter
	0. System displays Recent chat screen (See Figure 7a).
1.TUCBW User navigates to the search bar and enters the person's name to discuss book conditions and barter terms.	
	***2.** The system displays list of users that matches with the search keyword (See Figure 7b).
3.The user clicks on the person's profile to start the chat session.	
	***4.** The system displays chat session screen which includes an option to type and send message (See Figure 7c).
5. The user enters the message and click the send button	
	6.The system displays the messages sent and received in chat format.
7.TUCEW The other user receiving the message .	
Postcondition: Users communicating seamlessly	

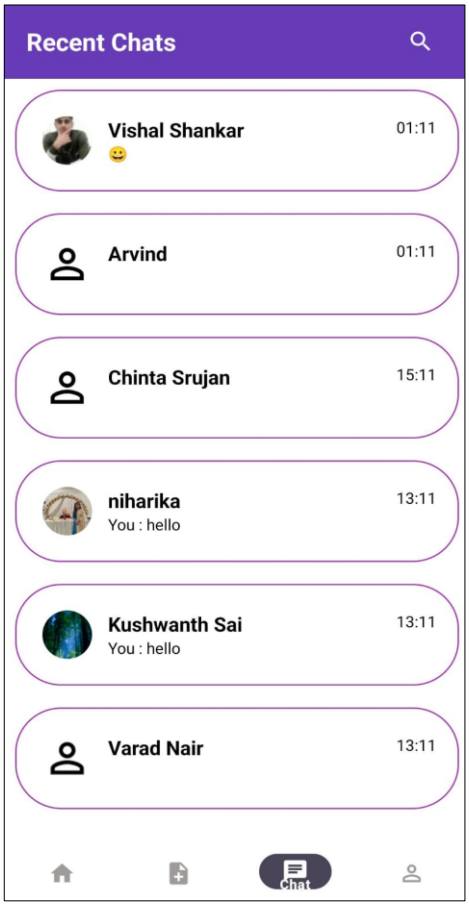


Fig. 7a

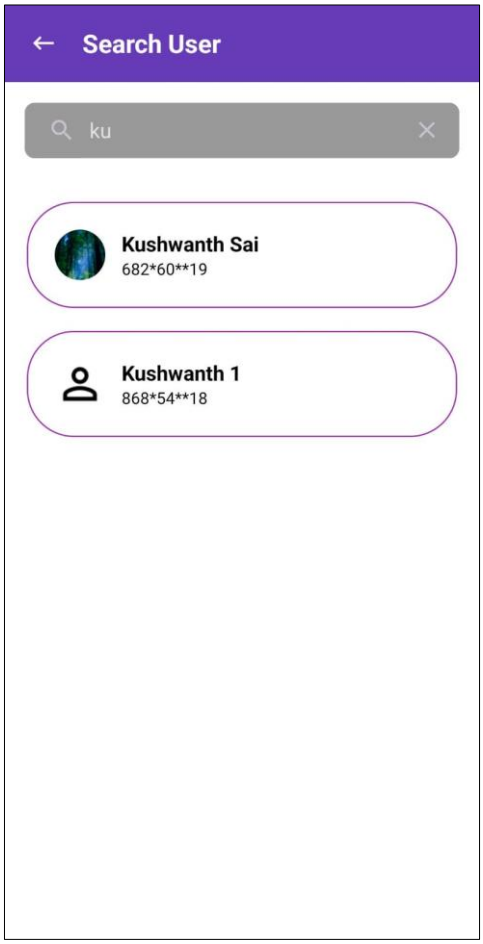


Fig. 7b

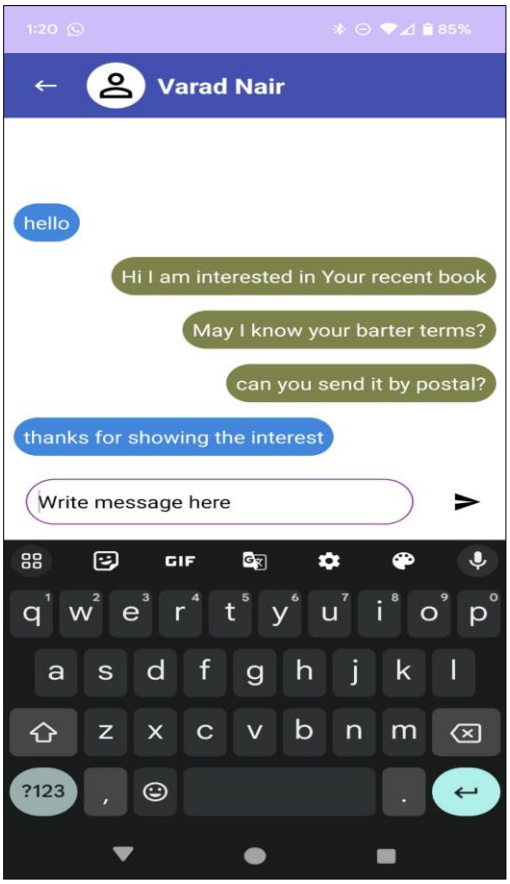


Fig. 7c

EUC8: Barter Confirmation

Precondition: This Use Case assumes user is on Manage Barter Screen after discussing the barter terms with the other user	
Actor – User	**System** – Book Barter
	0. System displays Manage Barter Screen of the logged in User (See Figure 8a).
1.TUCBW user clicks on "View Requests" button.	
	***2.** The System displays a list of users who have requested for the book, with their name, type of request (In-Person/Postal Service) and address of the postal service requestors (See Figure 8b), else "No Requests are made Yet" is displayed (See Figure 8c).
3. The user clicks on the Accept button of the chosen request.	
	4.The System displays "Accepted and Notified Successfully" (Figure 8d), set availability to "Not Available" (See Figure 8e) and send notification to the chosen user (See Figure 8f).
5. TUCEW Request being accepted, and notification sent	
Postcondition: Barter confirmation done.	

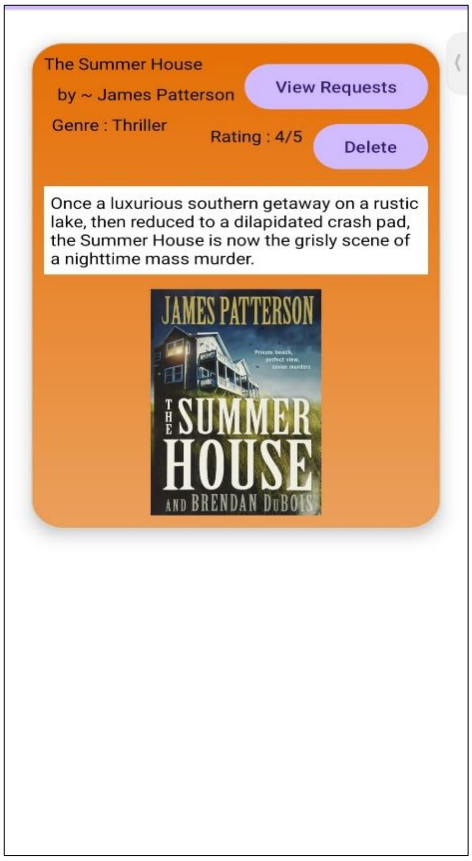


Fig. 8a

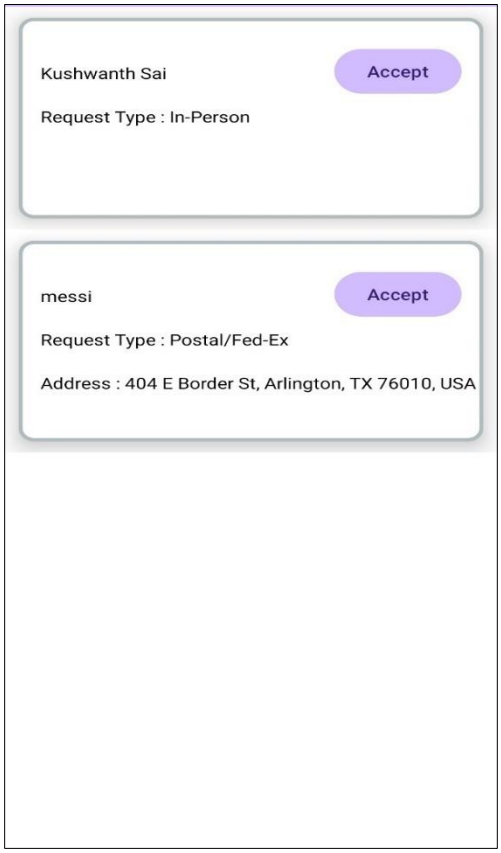


Fig. 8b

Fig. 8c

Fig. 8d

Fig. 8e

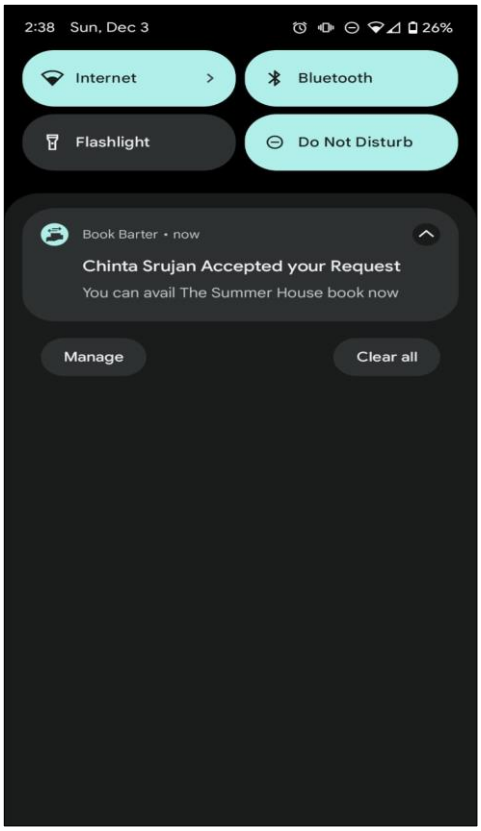


Fig. 8f

EUC 9: Utilize Location Services

Precondition: This use case assumes User is on the home screen and intends to filter posts based on the city from which they were posted	
Actor – User	**System** – Book Barter
1.TUCBW The user navigates to the search bar on the top of home screen and enters a city name to find potential barters in that location. . **3.TUCEW** user getting list of posts originated from specified location	**0.** System displays home screen (See Figure 9a). ***2.** The system access the location of the other users and displays a list of book posts, posted by barters located in the specified area (See Figure 9b).
Postcondition: The user finds book posts from anticipated location.	

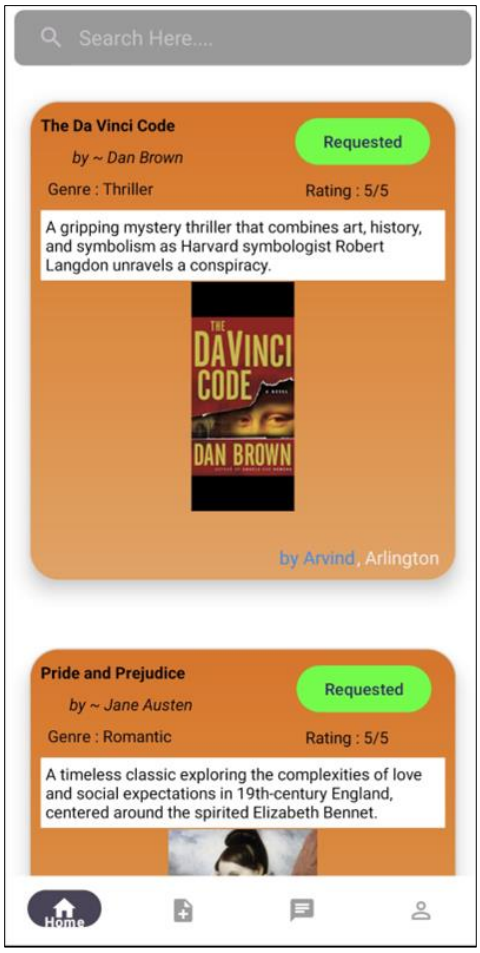

Search Here....
The Da Vinci Code
Requested
by ~ Dan Brown
Genre : Thriller
Rating : 5/5
A gripping mystery thriller that combines art, history, and symbolism as Harvard symbologist Robert Langdon unravels a conspiracy.
THE DA VINCI CODE
DAN BROWN
by Arvind, Arlington
Pride and Prejudice
Requested
by ~ Jane Austen
Genre : Romantic
Rating : 5/5
A timeless classic exploring the complexities of love and social expectations in 19th-century England, centered around the spirited Elizabeth Bennet.
Home

Fig. 9a

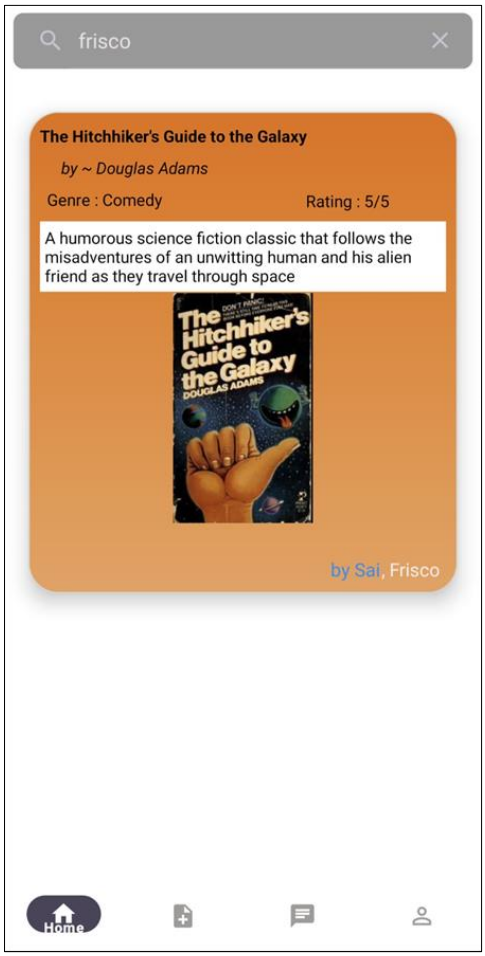

frisco
The Hitchhiker's Guide to the Galaxy
by ~ Douglas Adams
Genre : Comedy
Rating : 5/5
A humorous science fiction classic that follows the misadventures of an unwitting human and his alien friend as they travel through space
The Hitchhiker's Guide to the Galaxy
DOUGLAS ADAMS
by Sai, Frisco
Home

Fig. 9b

EUC10: User Rating

Precondition: This use case assumes user is on the home screen and wants to rate other user.	
Actor - User	**System -** Book Barter
	0. System displays Home Screen (See Figure 5a).
1. TUCBW user clicking on username of other person/ barter at the bottom of the post.	
	***2.** The System displays the other user profile along with an option to rate the user (See Figure 10a).
3. The user rates and clicks submit.	
	4. The System updates the other user Overall rating and displays "Rating Submitted" (See Figure 10b).
5. TUCEW user rating the other user/ barter.	
Postcondition: The other user/barter overall rating gets updated and visible to users	

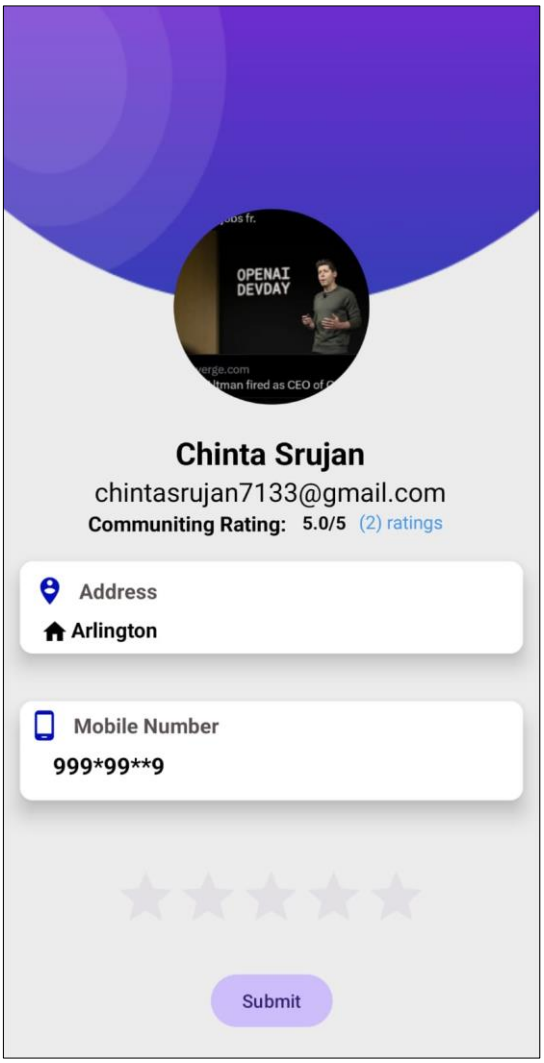


Fig. 10a

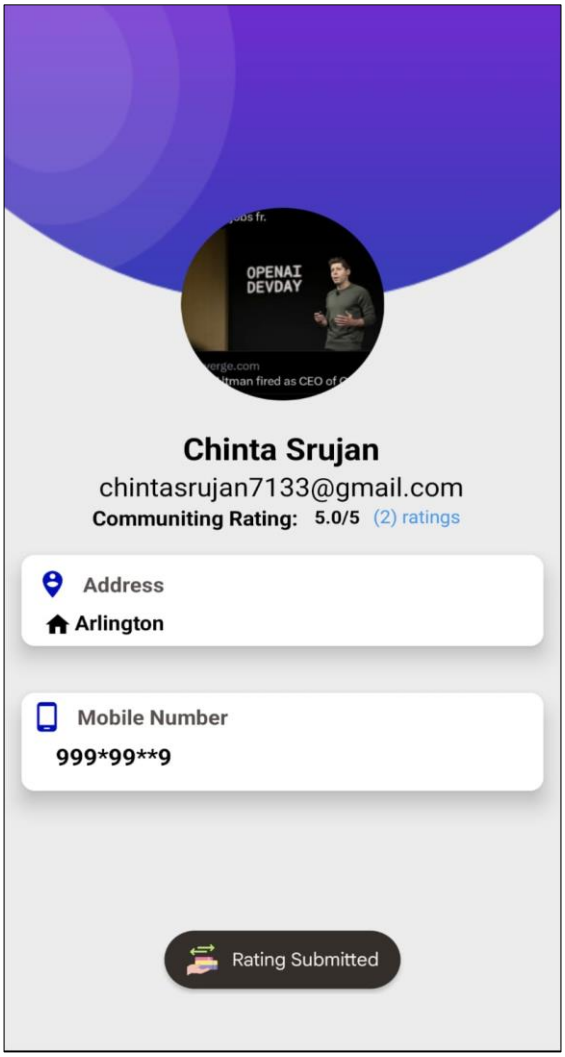


Fig. 10b

EUC11: Manage Notifications

Precondition: This use case assumes User is on the profile screen.	
Actor – User	**System** – Book Barter
1.TUCBW user clicks on Manage Notifications on profile screen. .	**0.** System displays profile screen (See Figure 11a).
	2. The system displays list of genres (Sci-Fi, Comedy, Thriller, Detective, Romantic, Horror, History, Comic) for the user to subscribe (See Figure 11b).
3.The user clicks on the genre **a)** To subscribe **b)** To unsubscribe	
	***4.** The system displays the genre **a)** In Green color if subscribed along with a message "Subscribed {Genre name}" (See Figure 11c) **b)** In white color if unsubscribed along with a message "Unsubscribed {Genre name}" (See Figure 11d)"
5.TUCEW user subscribed/Unsubscribed to genres that he/she is interested to receive notifications.	
Postcondition: The user manages notifications.	

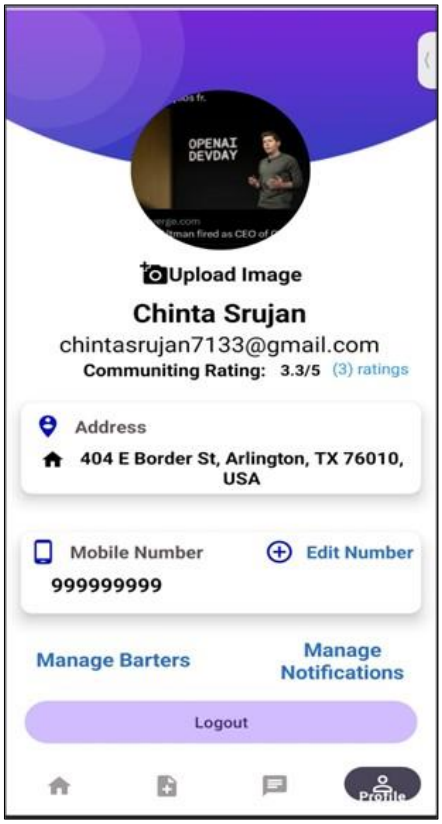


Fig. 11a

Fig. 11b

Fig. 11c

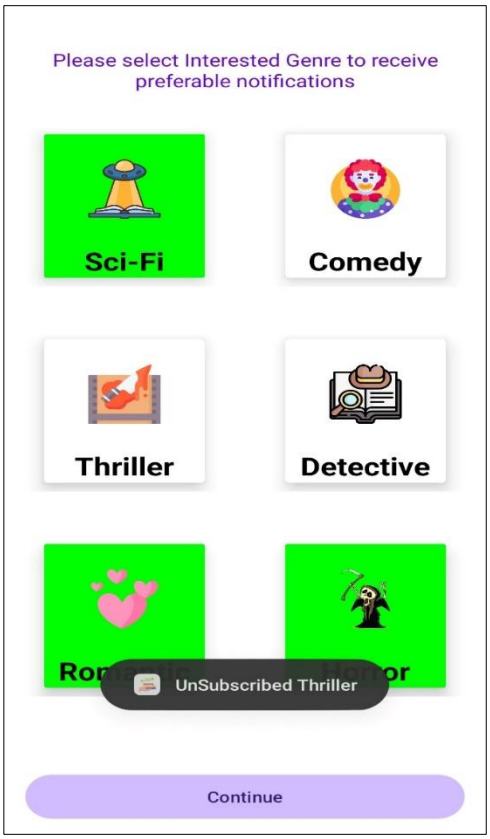


Fig. 11d

EUC12: Manage Barter History

Precondition: This Use Case assumes user is on the User profile Screen.	
Actor – User	**System** – Book Barter
1.TUCBW user clicks on "Manage Barters".	**0.** System displays User profile screen of the logged in User (See Figure 12a).
	***2.** The System displays the book posted by the user, along with 2 options for the user, "View Requests" or "Delete" (See Figure 12b).
3. The user (a) Clicks "View Requests" button (b) Clicks on "Delete" button	
	4. The System accordingly (a) displays a list of users who have requested (See Figure 12c). (b) deletes the post and displays "Successfully Deleted" (See Figure 12d).
5.TUCEW User managing barter posts	
Postcondition: User manages barters	

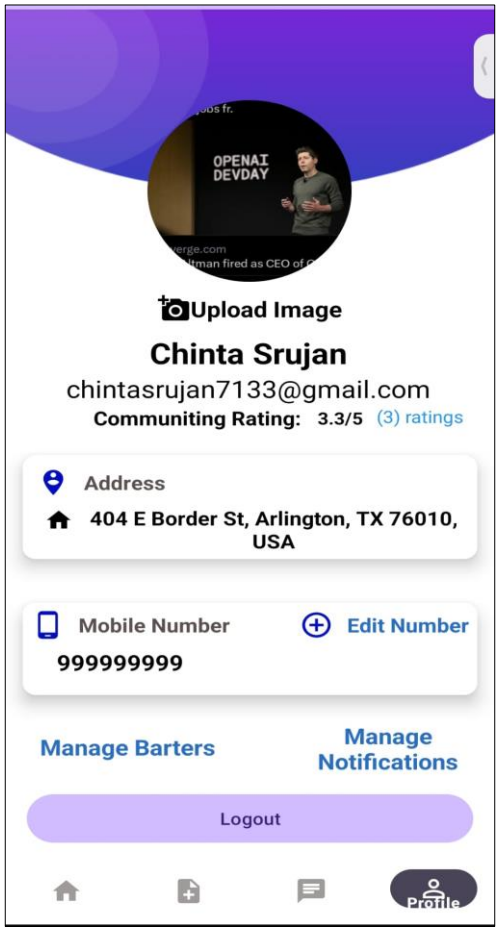


Fig. 12a

Fig. 12b

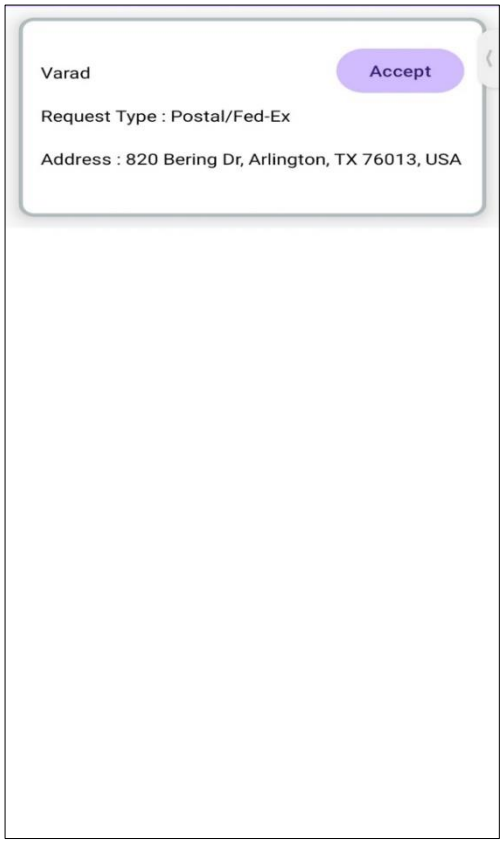


Fig. 12c

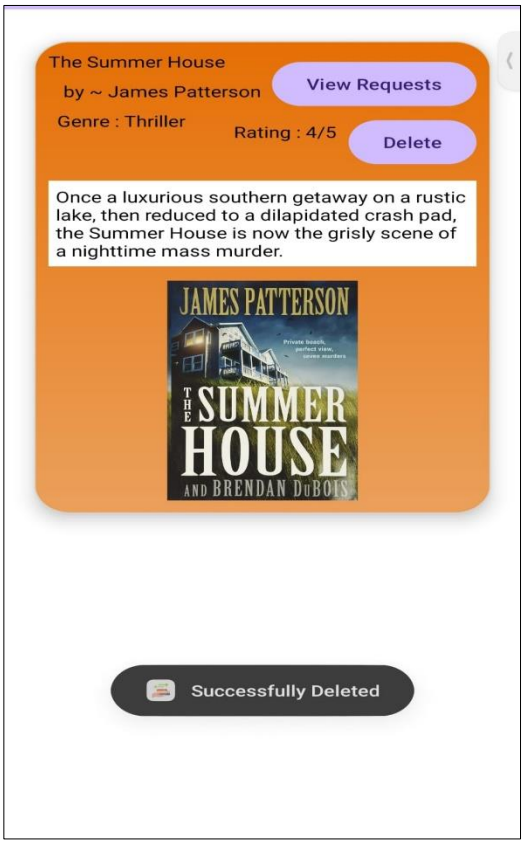


Fig. 12d

EUC 13: Request Book via Postal Service

Precondition: This use case assumes User is on the home screen and clicked on Interested button on the respective book post.	
Actor – User	**System** – Book Barter
1.TUCBW User clicks on Postal option. . **3.TUCEW** user opted for postal preference and request sent successfully.	**0.** System displays dialog box (See Figure 13a). ***2.** The system validates the request and sends request to the barter along with the requestor address and displays "Request sent successfully" (See Figure 13b).
Postcondition: The request is visible with request type as postal to the barter	

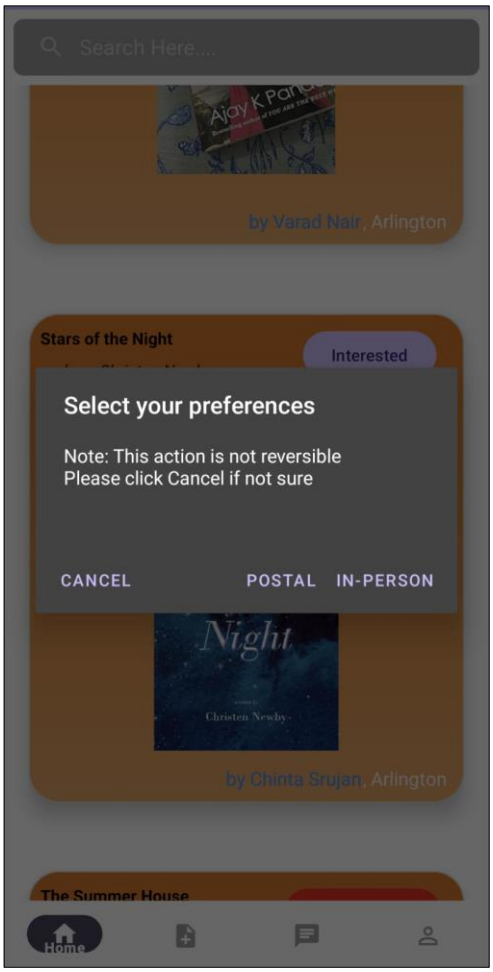


Fig. 13a

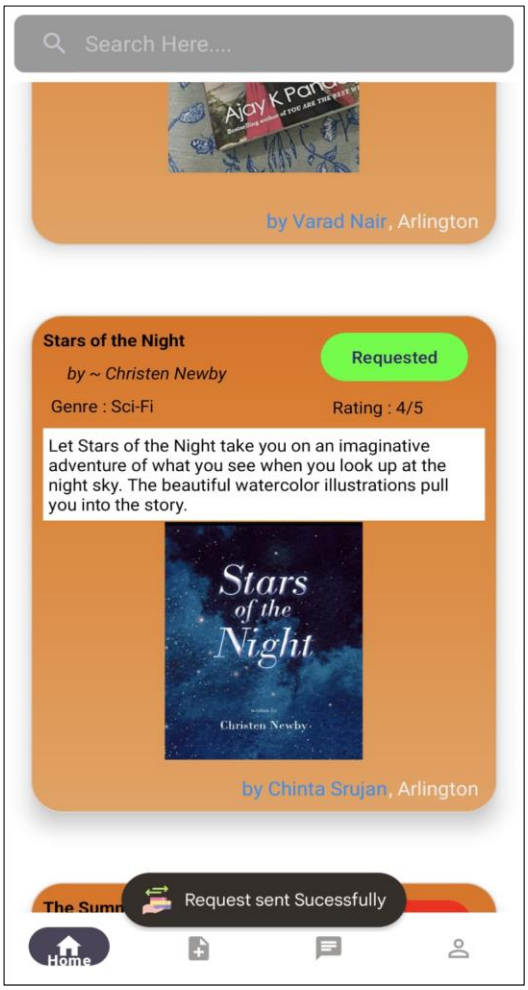


Fig. 13b

EUC14: Reset Password

Precondition: The User Should be Registered with a valid email Id	
Actor – User	**System** – Book Barter
1.TUCBW the user clicking the forgetpassword link.	**0.** System displays login screen (See Figure 2a).
	2.The System displays Forget Password screen (See Figure 14a).
3.User enters email address associated with the account and clicks Reset Password button.	
	***4.** The system sends an email along with reset link (SeeFigure 14b, 14c) check if the user is registered and display "Reset Link sent successfully" (See Figure 14d).
5.User opens the reset link received on emailand enters new password.	
	6.The System displays Password Changed "You can now sign in with the new Password (See figure 14e).
7.TUCEW password being updated.	
Postcondition: New Password replaced with old password	

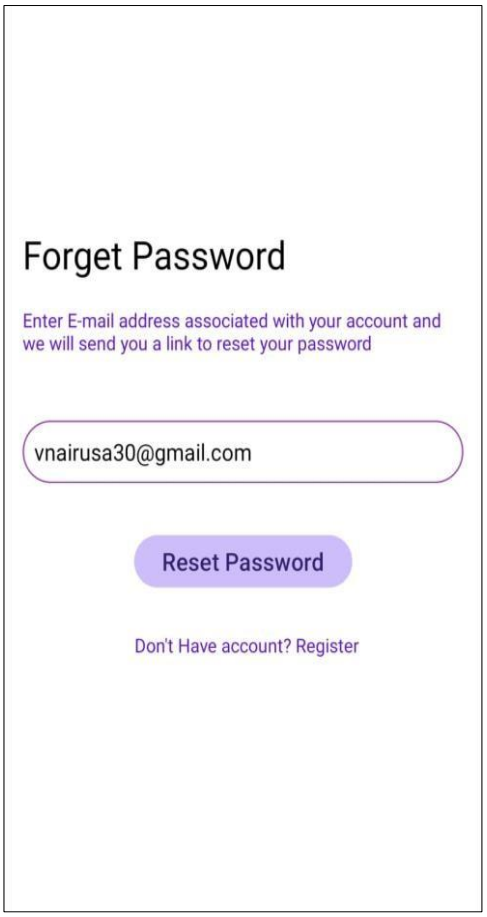


Fig. 14a

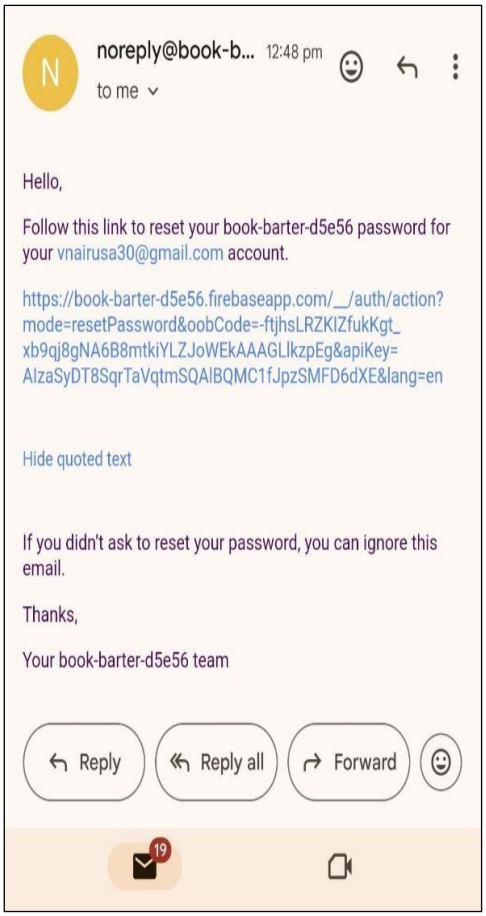


Fig. 14b

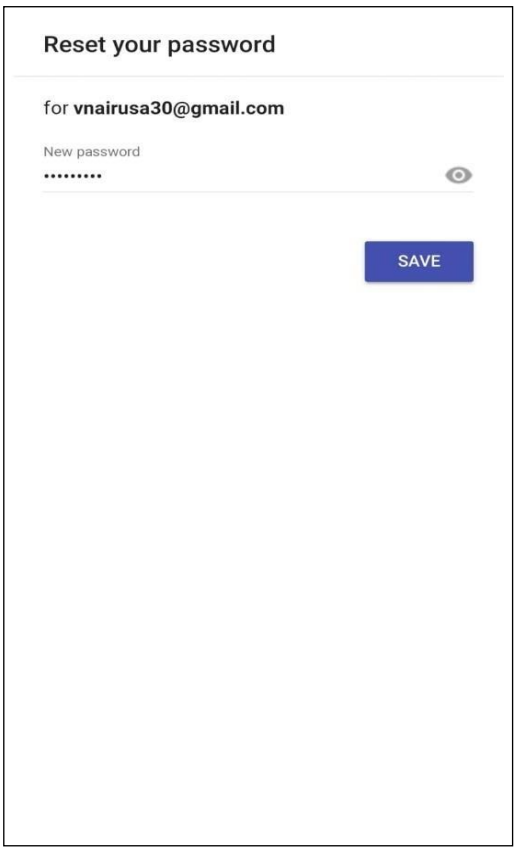


Fig. 14c

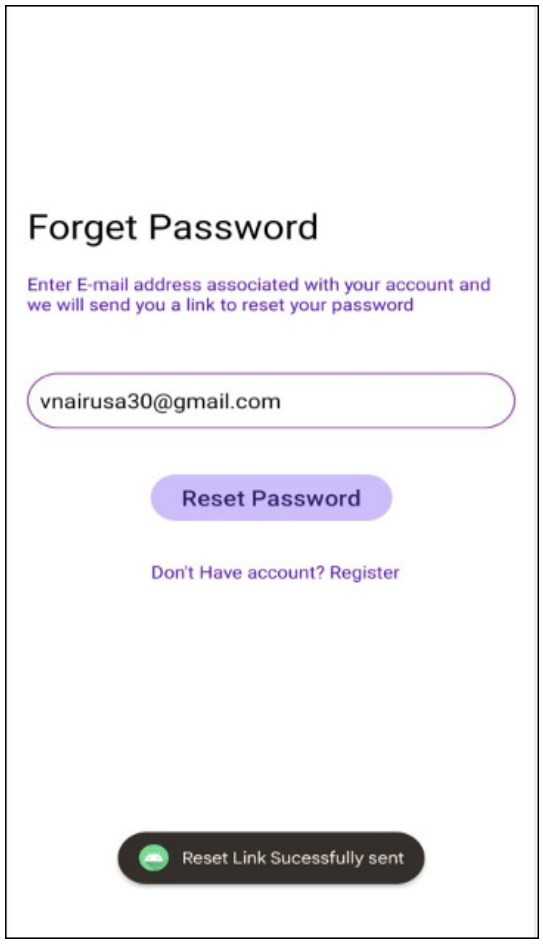


Fig. 14d

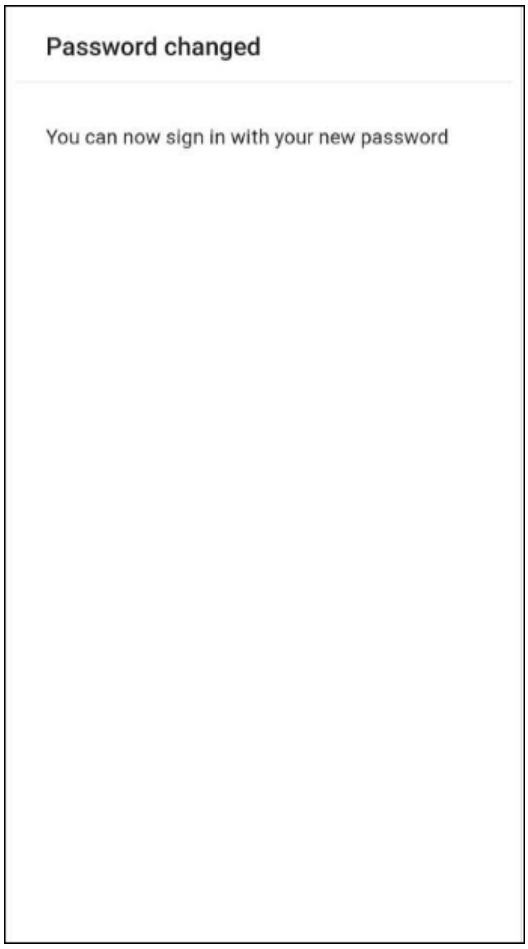
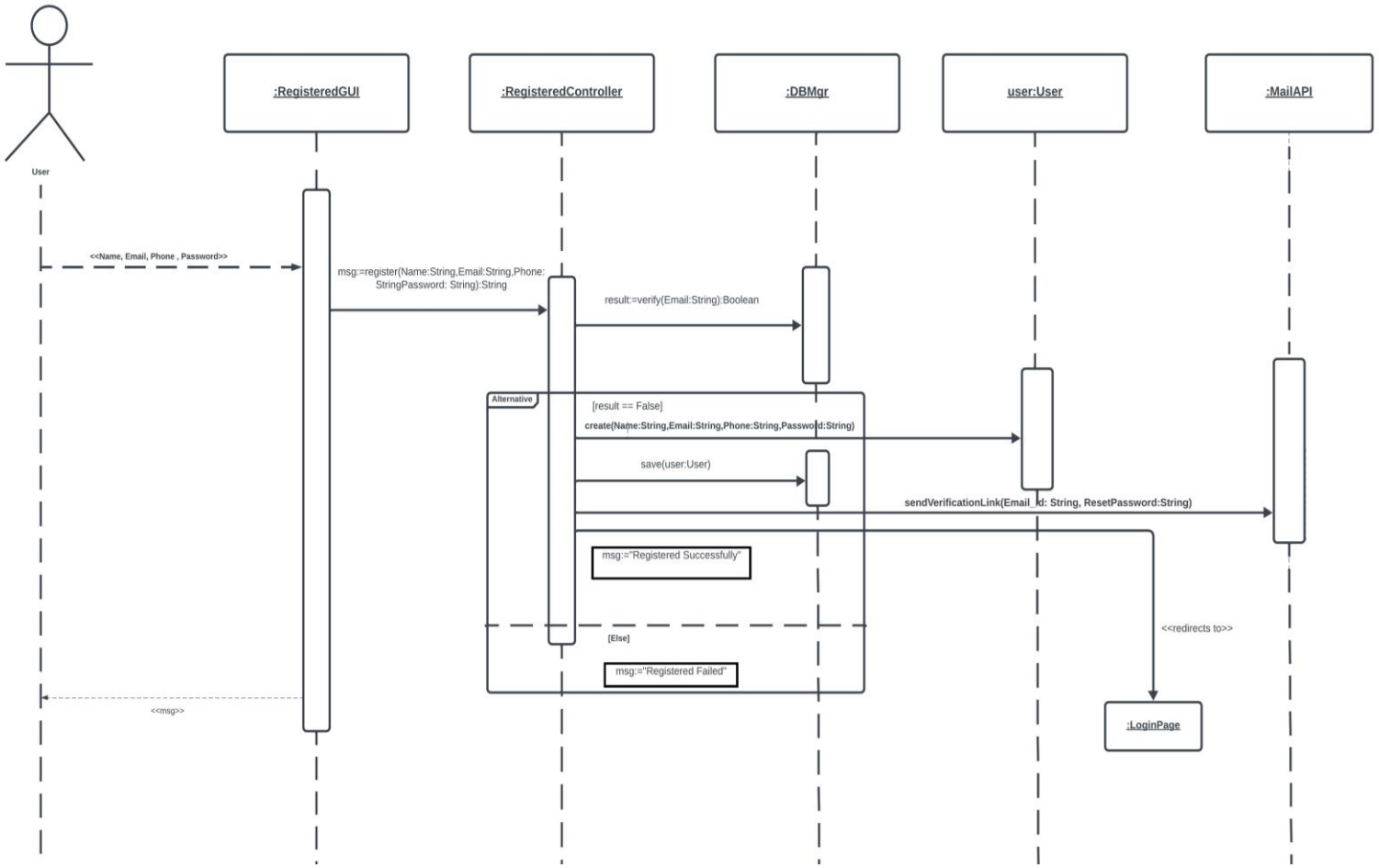


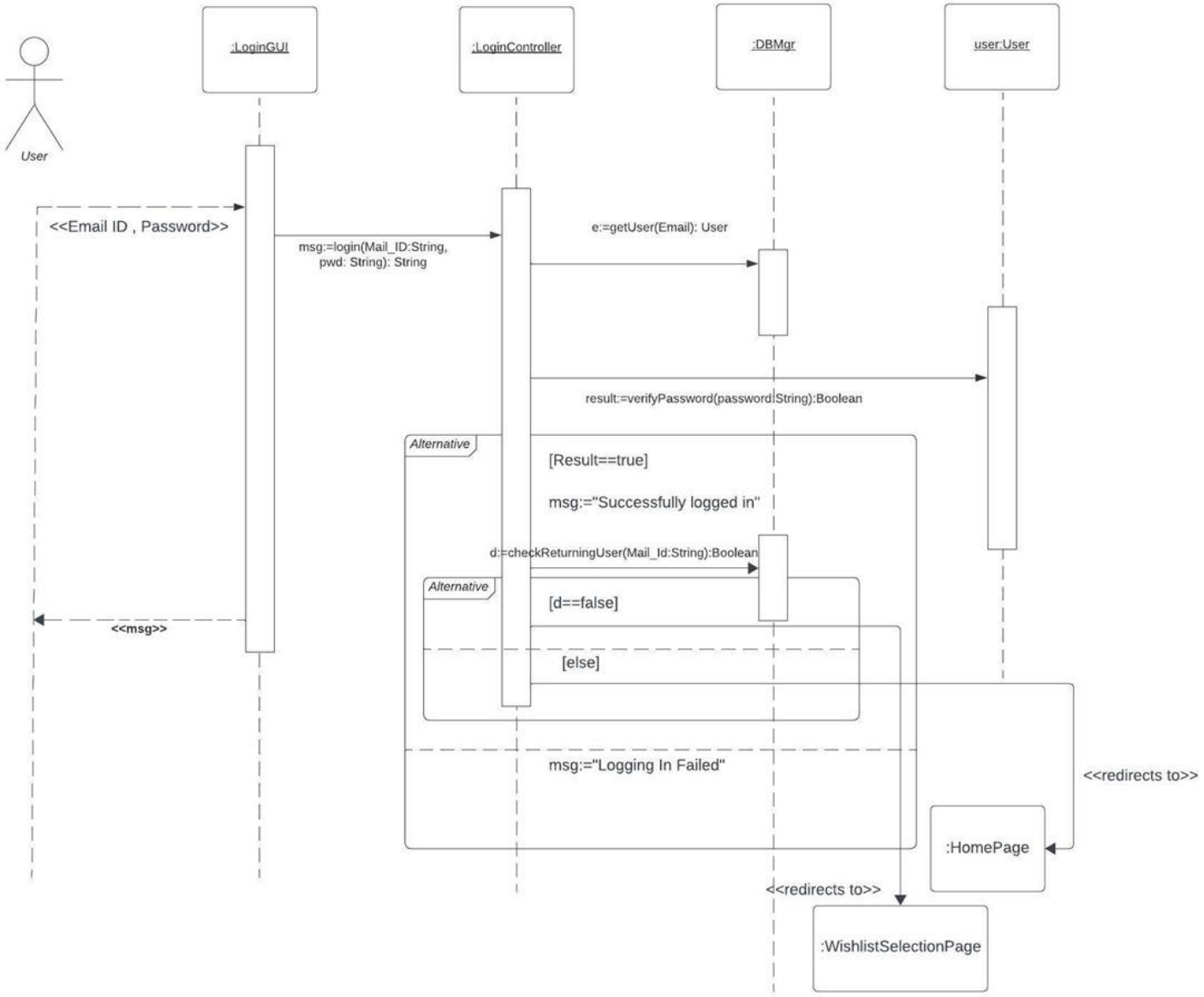
Fig. 14e

Sequence Diagrams:

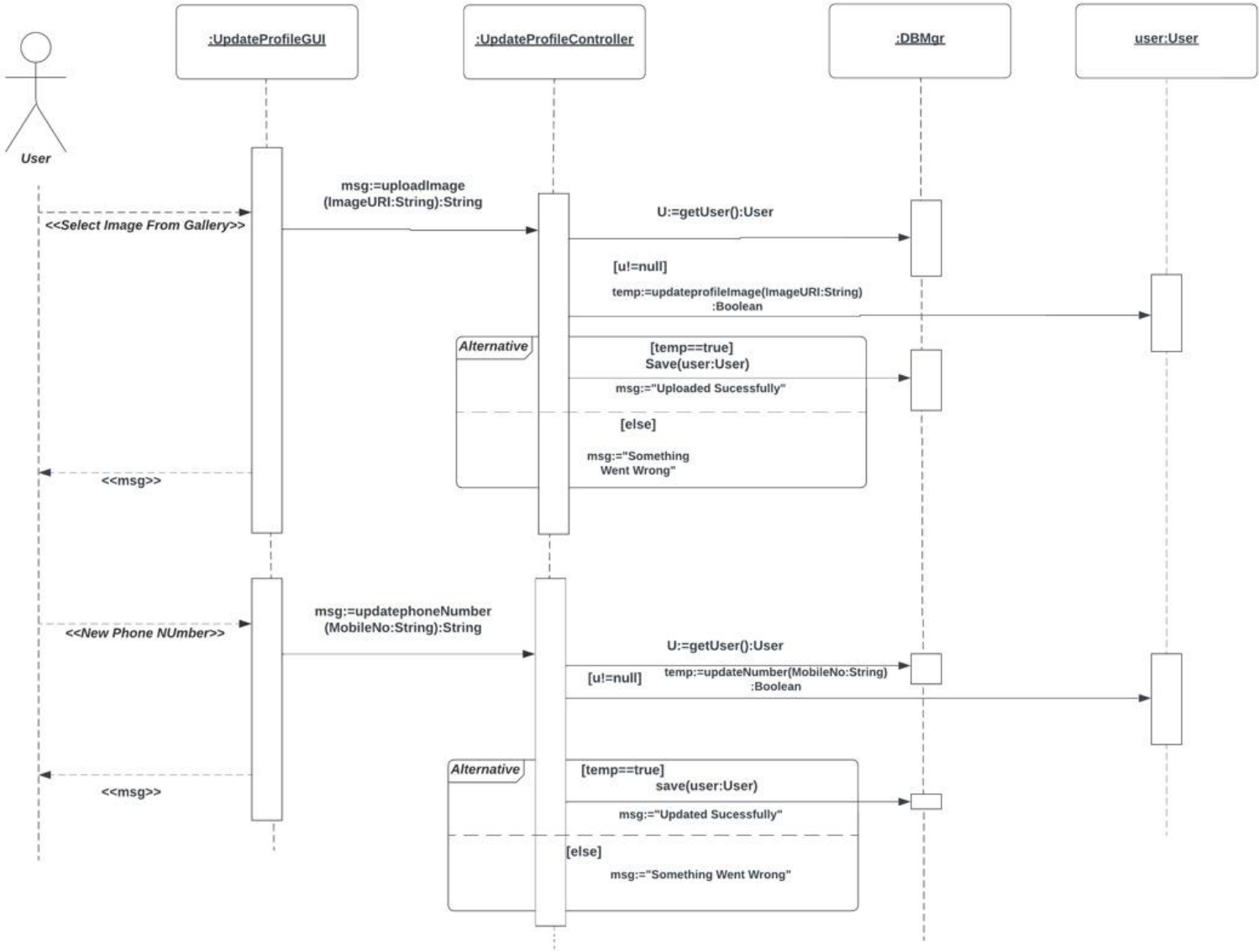
Sequence Diagram For Register:



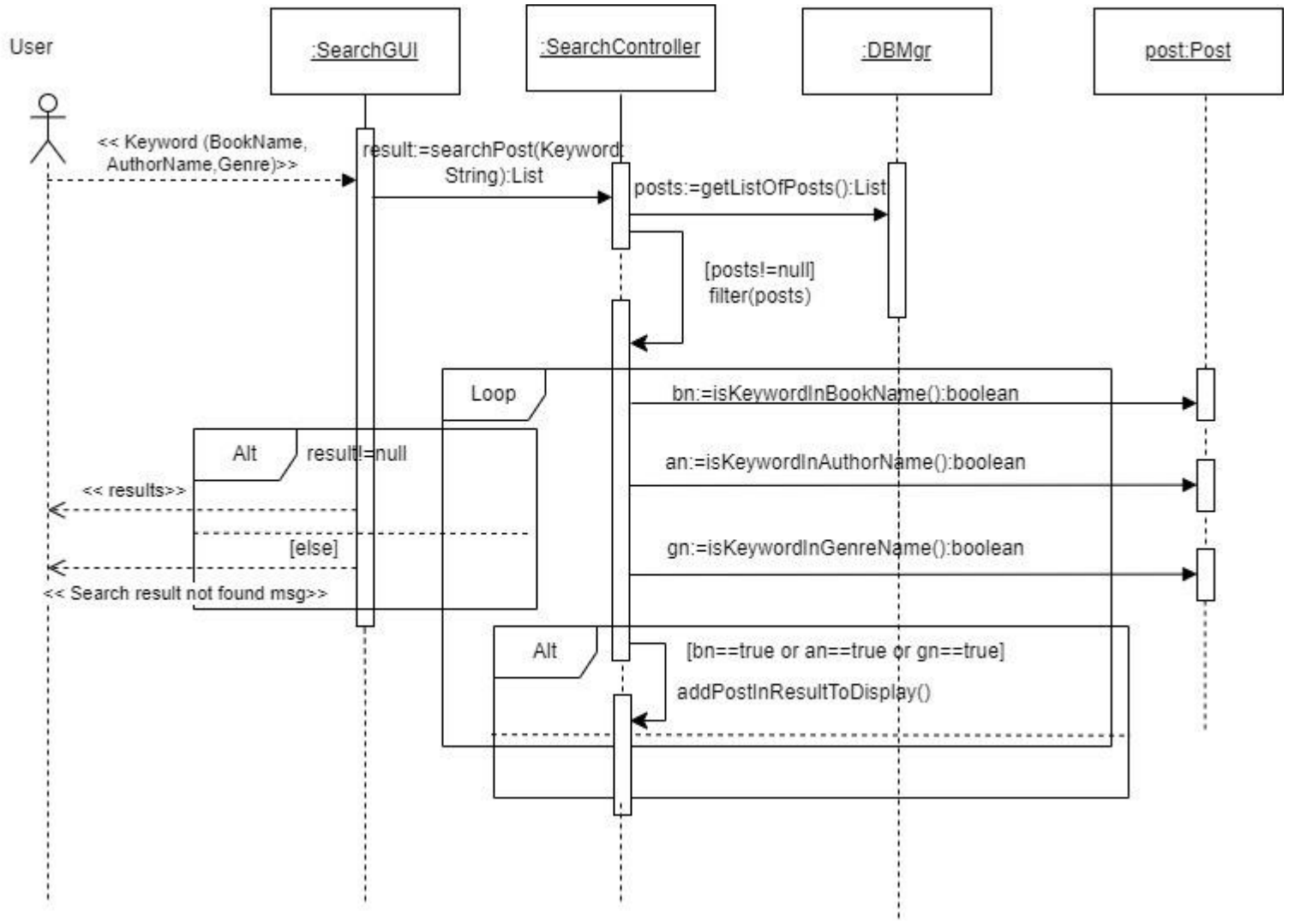
Sequence Diagram For Login:



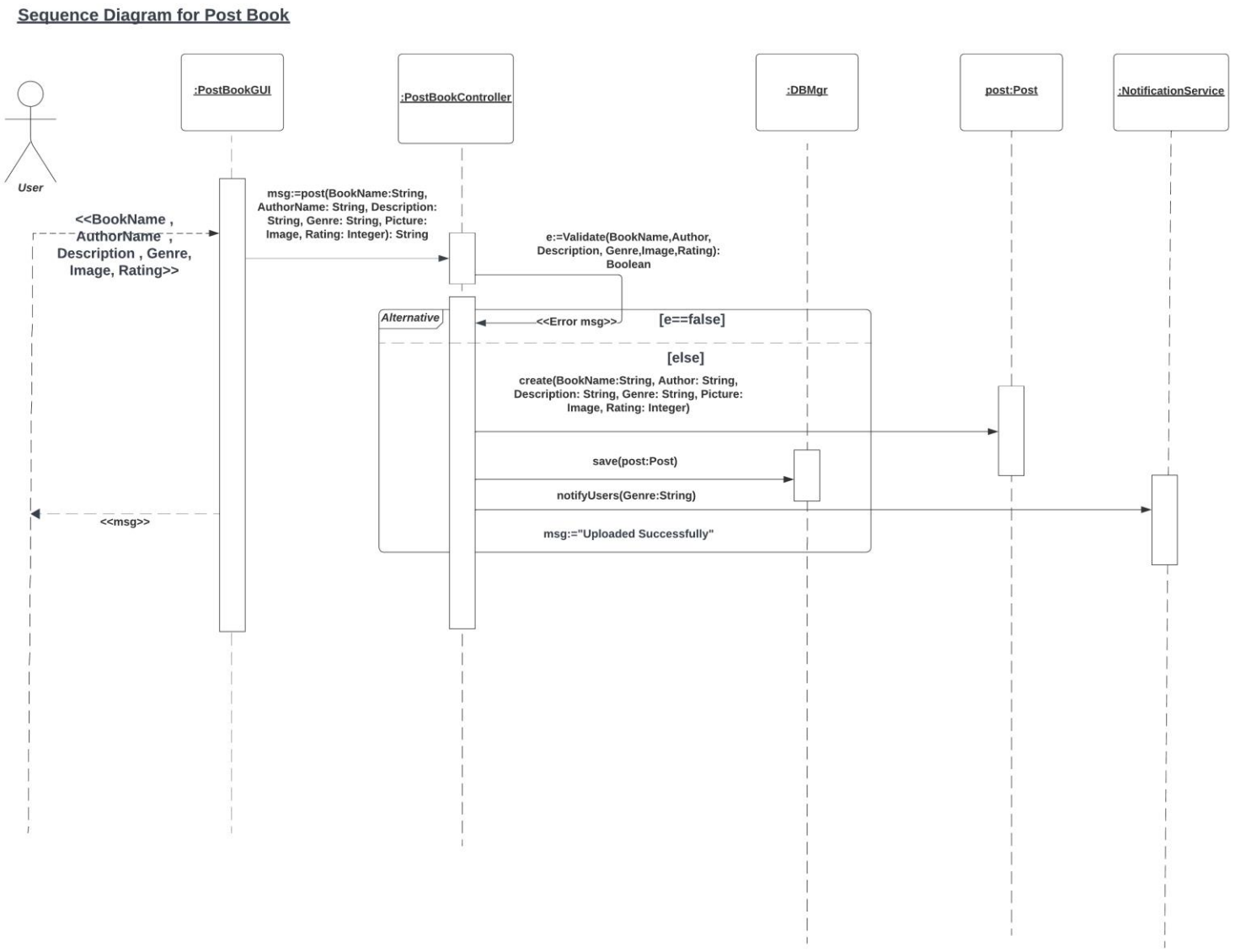
Sequence Diagram for Update Profile:



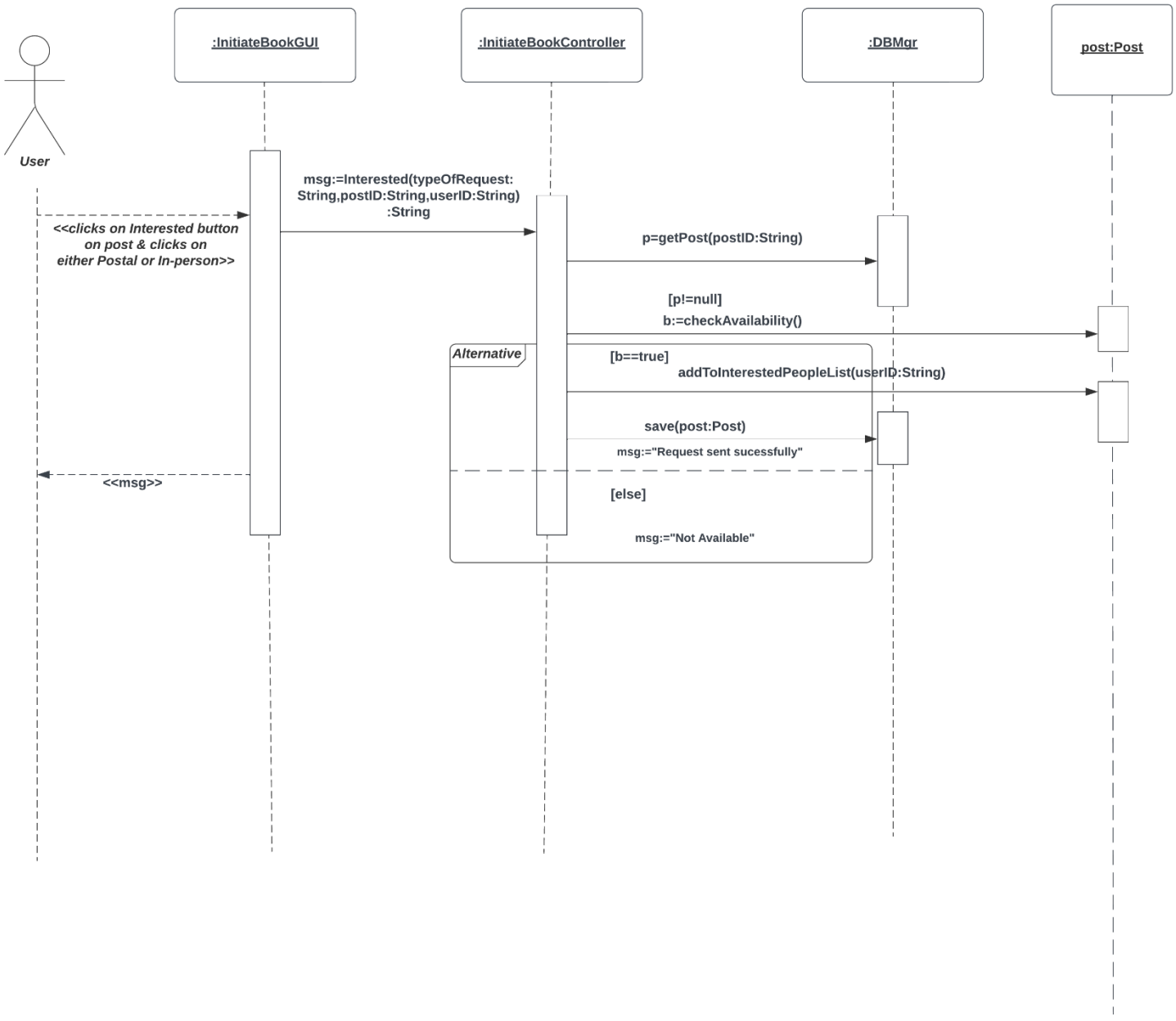
Sequence Diagram for Book Search:



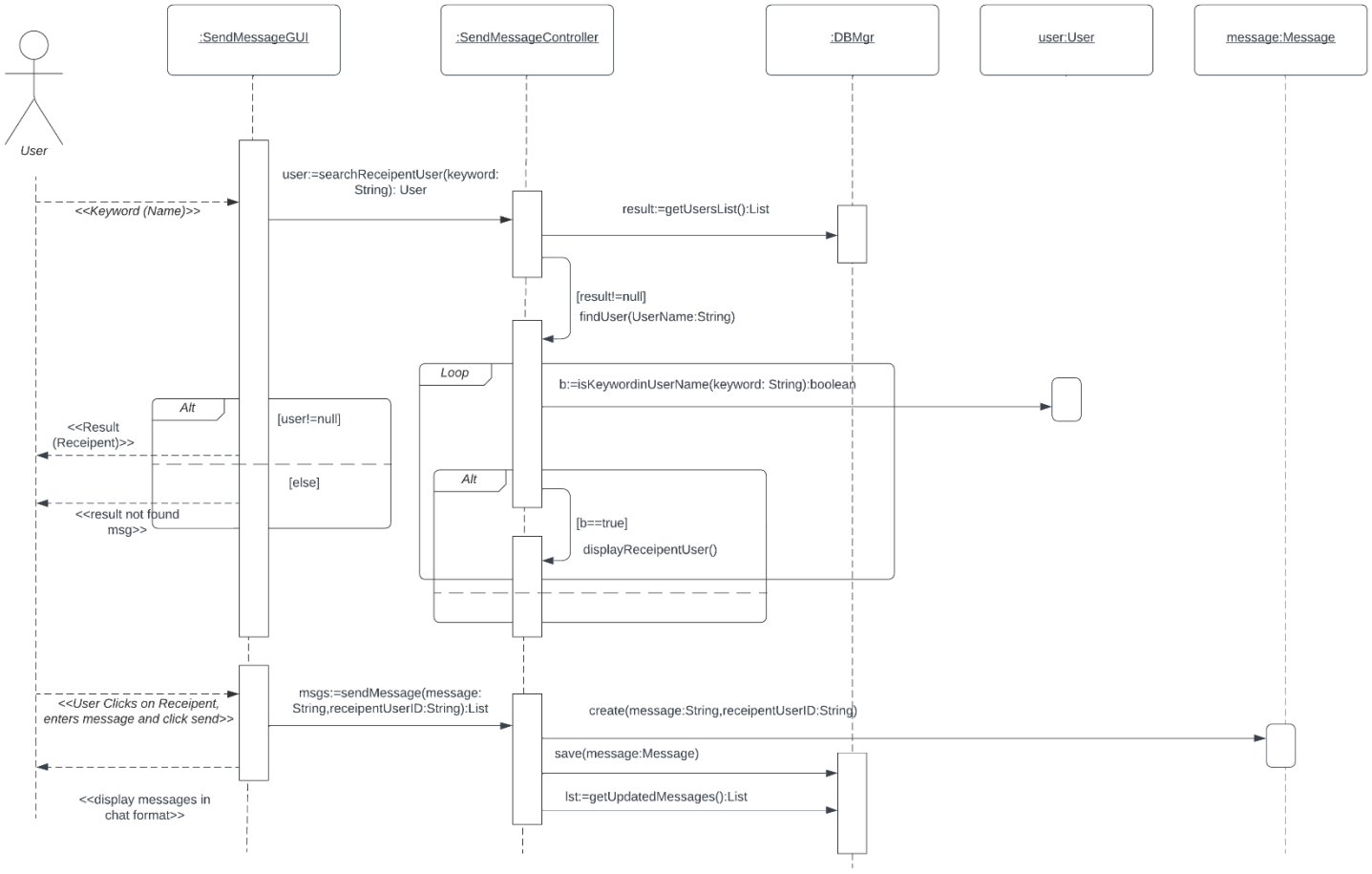
Sequence Diagram for Post Book:



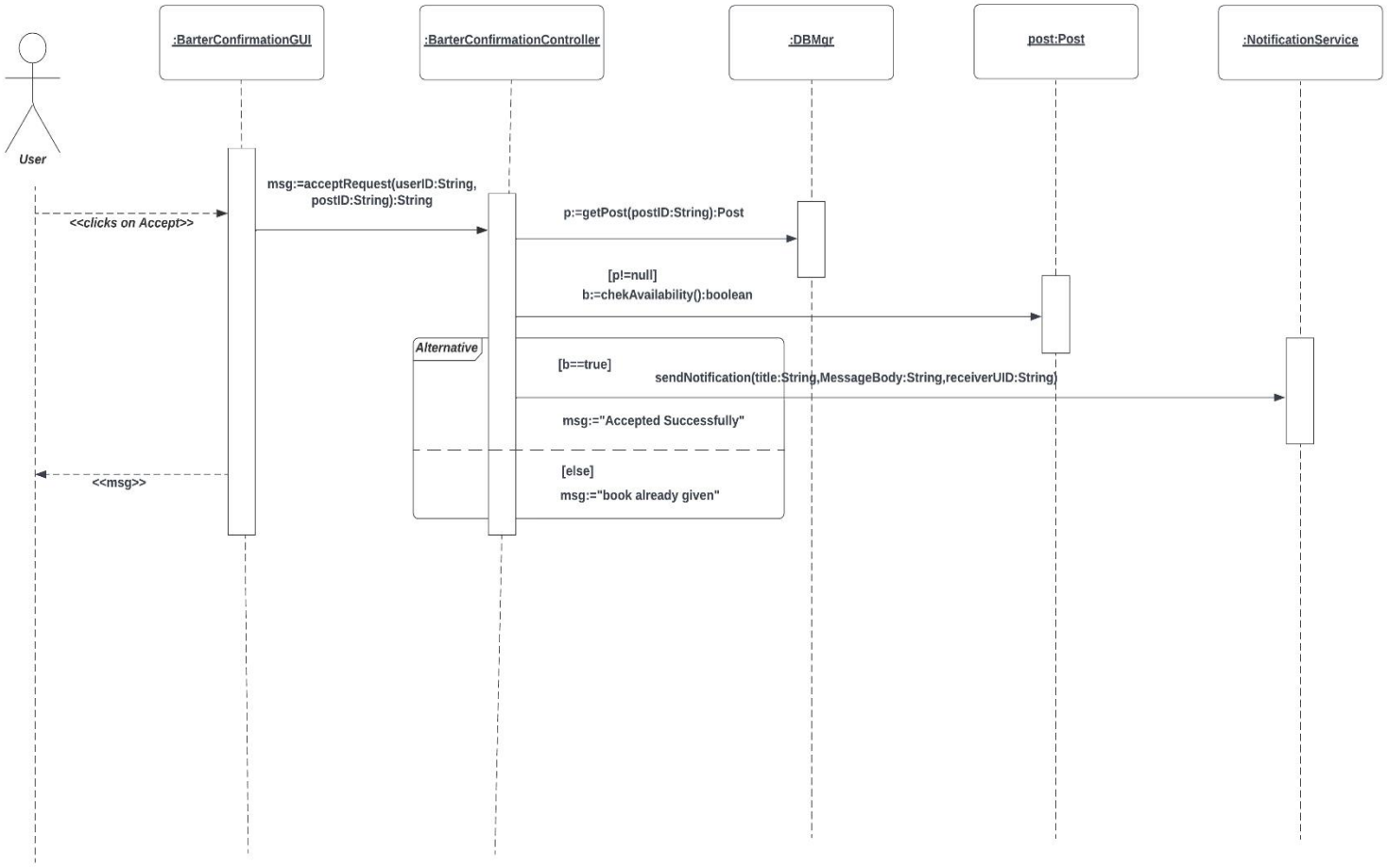
Sequence Diagram for Initiate Book Exchange:



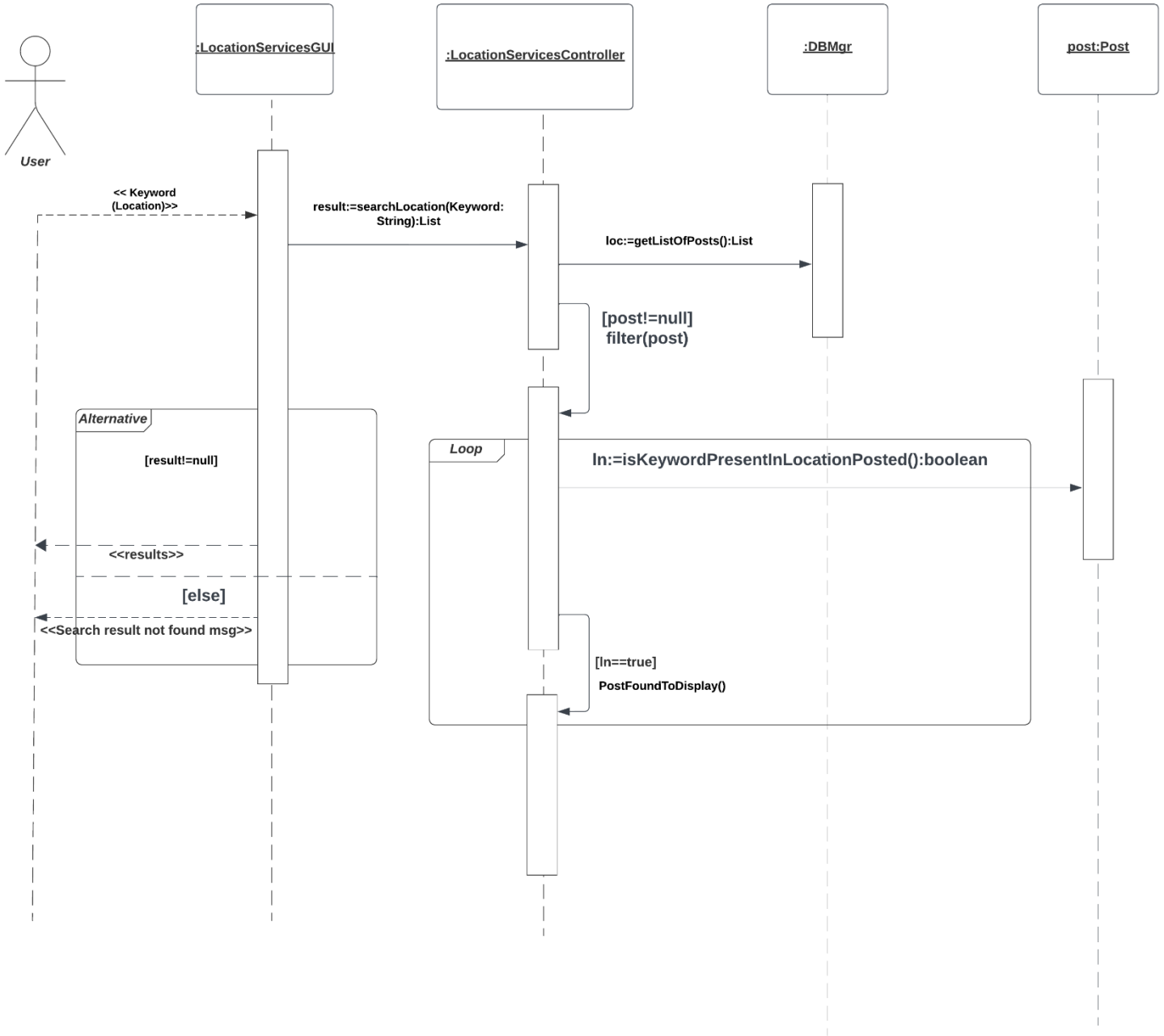
Sequence Diagram for Send Message:



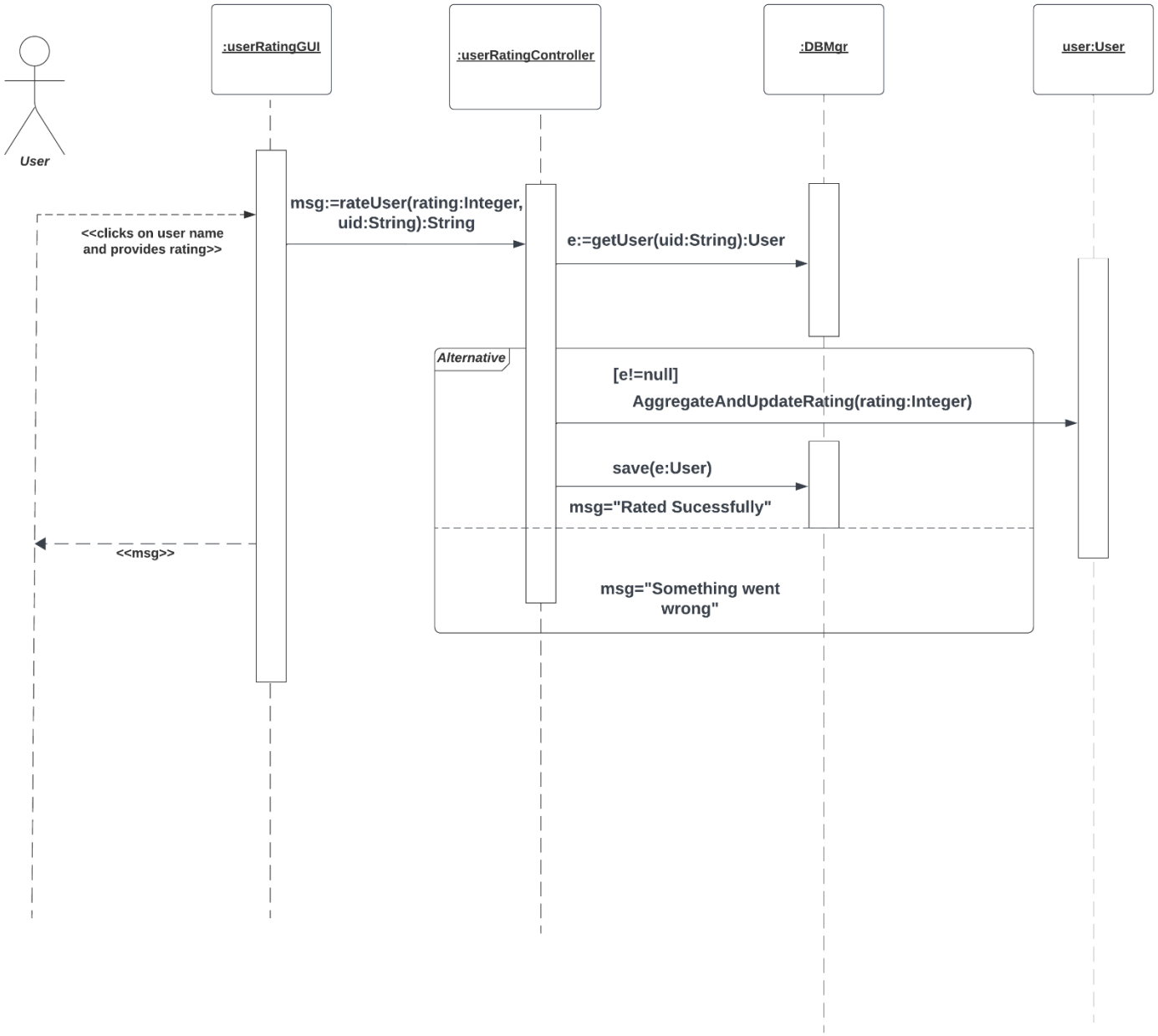
Sequence Diagram for Barter Confirmation:



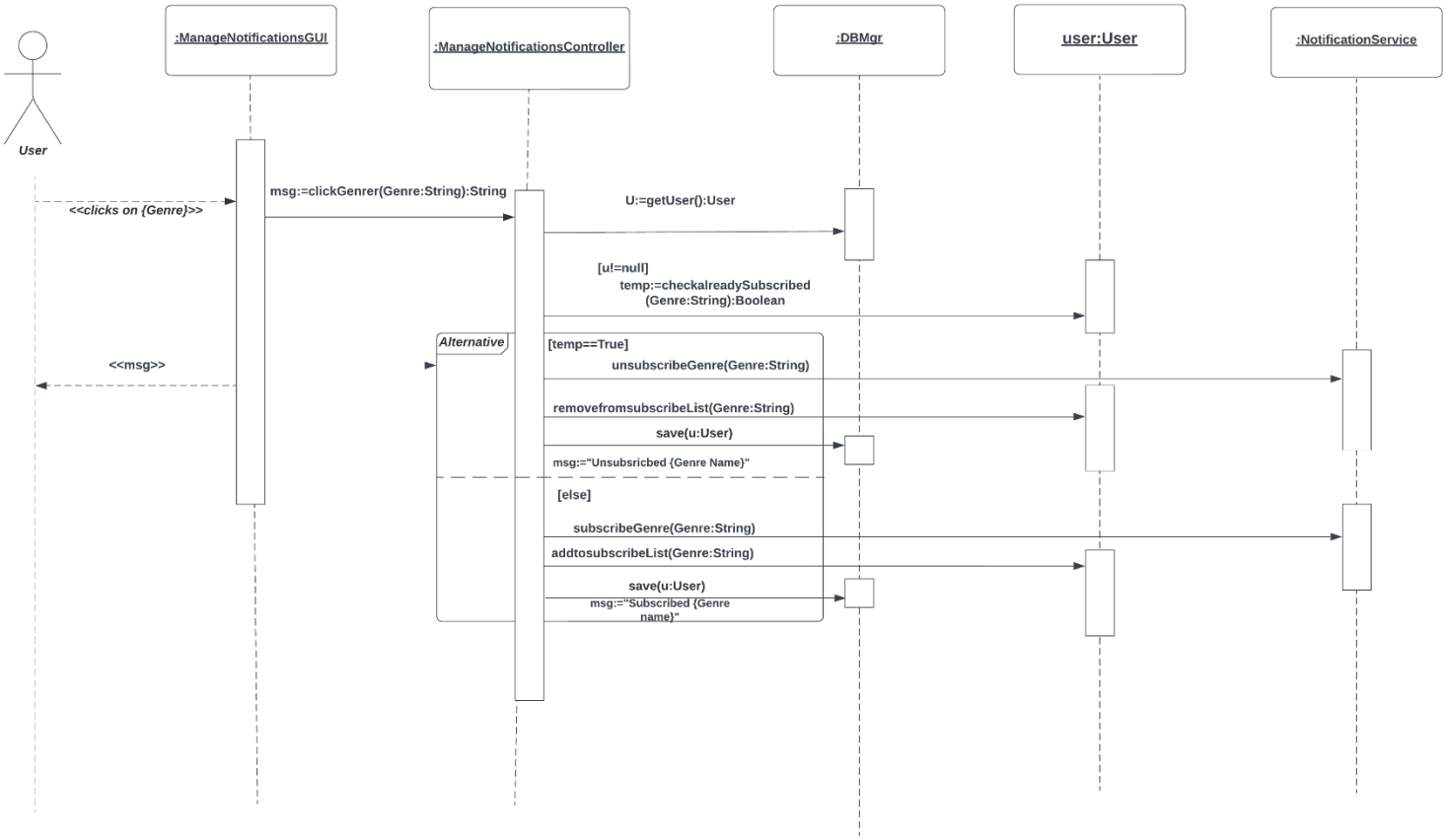
Sequence Diagram for Utilize Location Services:



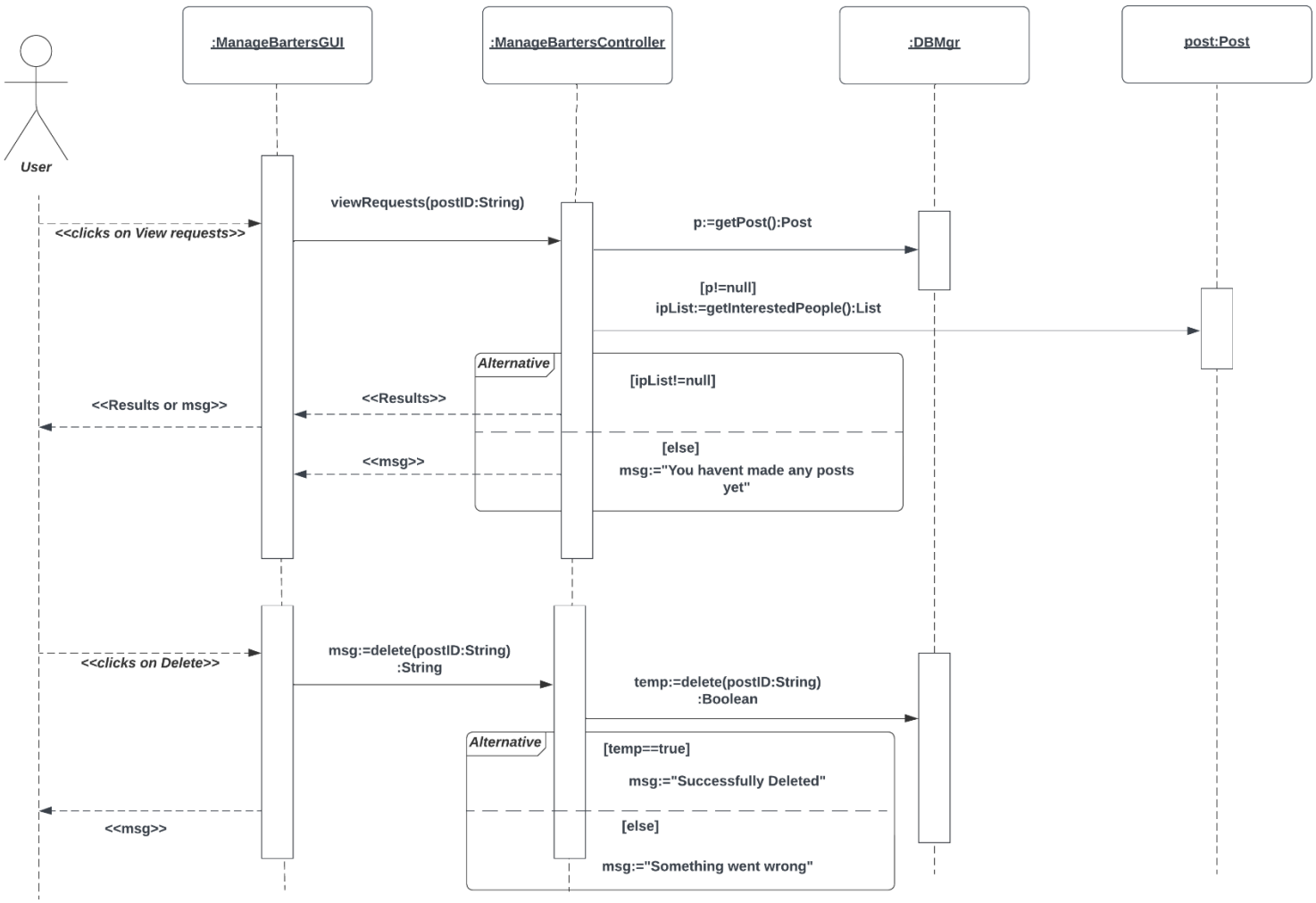
Sequence Diagram for User Rating:



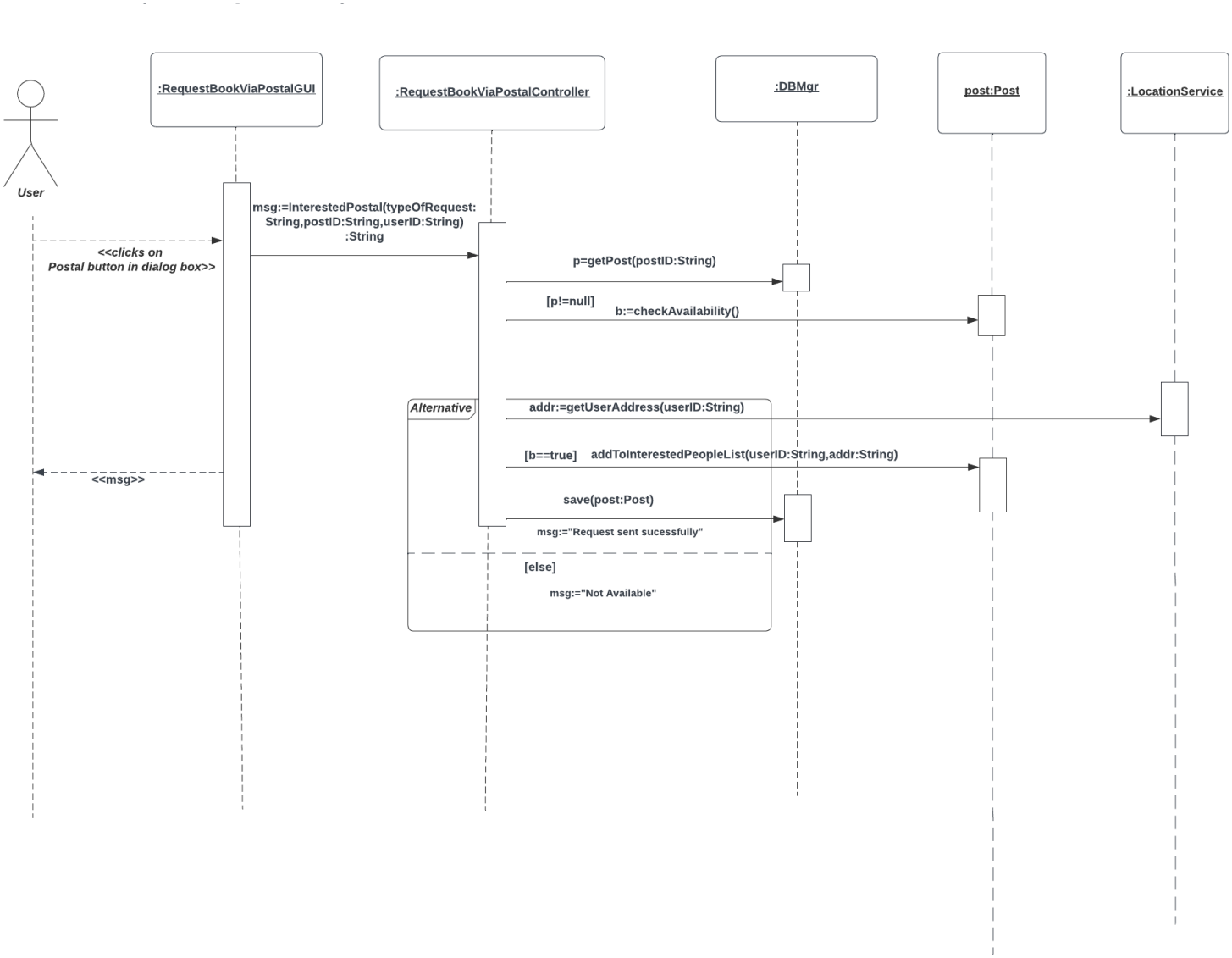
Sequence Diagram for Manage Notifications:



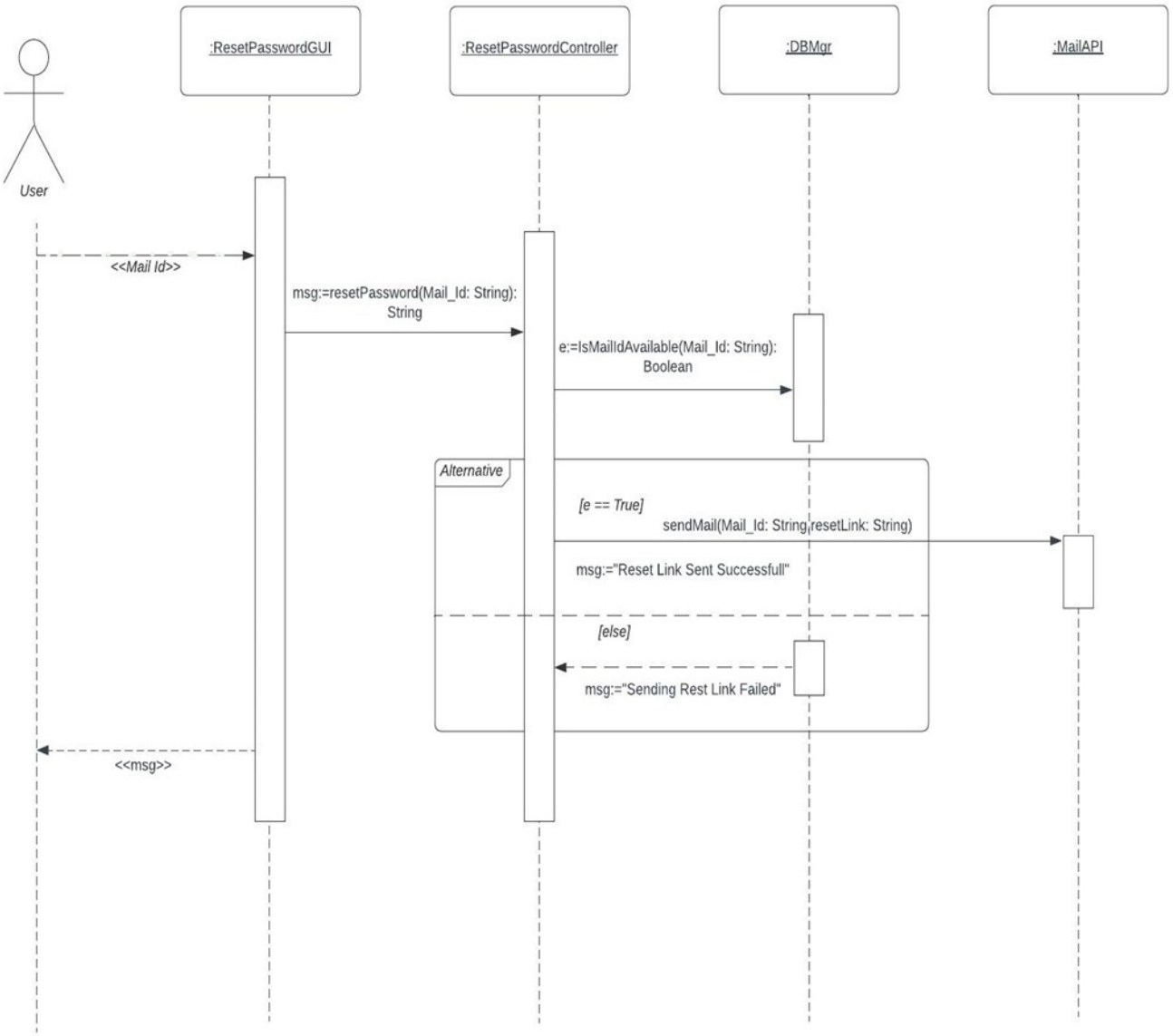
Sequence Diagram for Manage Barter History:



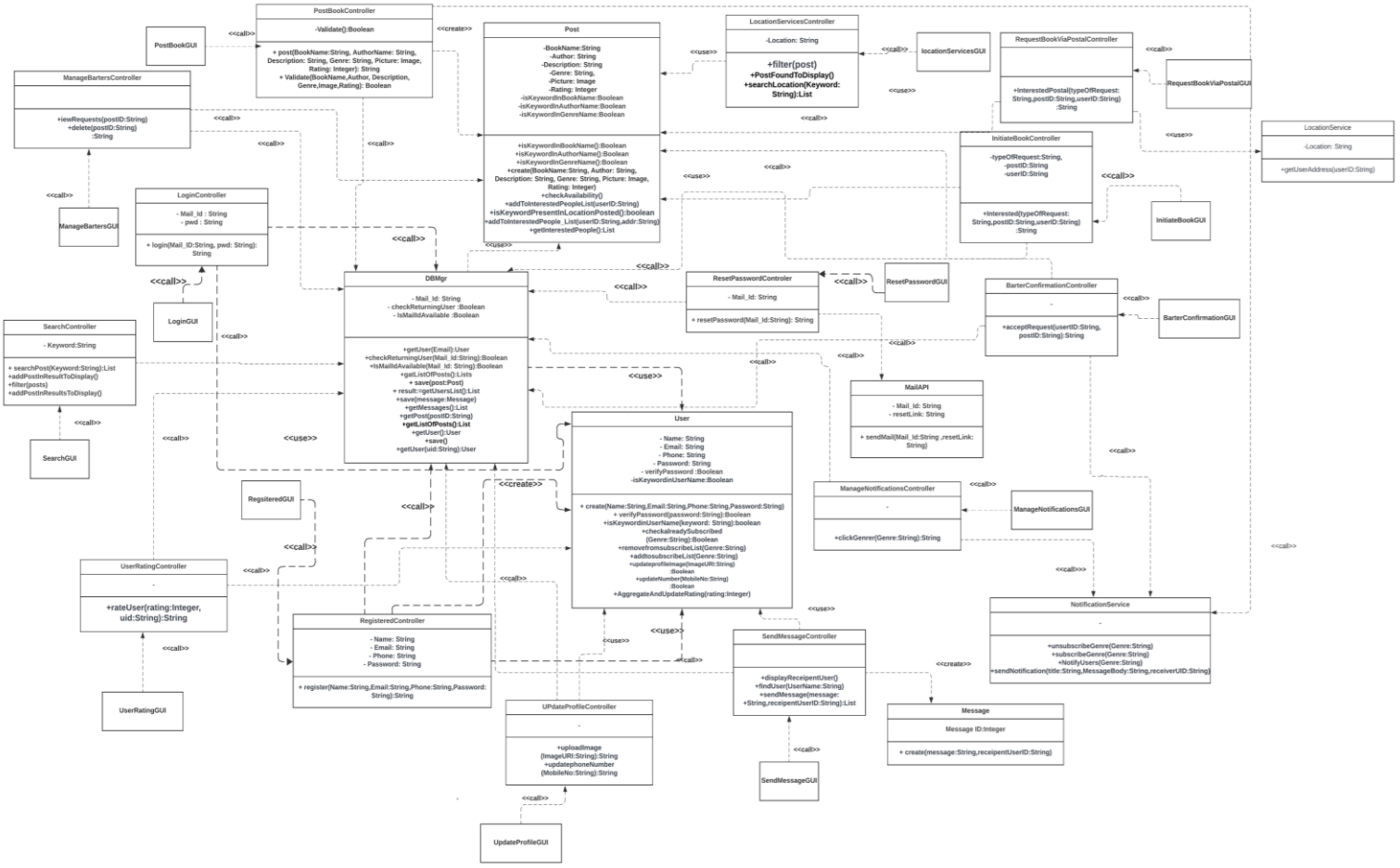
Sequence Diagram for Request Book via Postal Service:



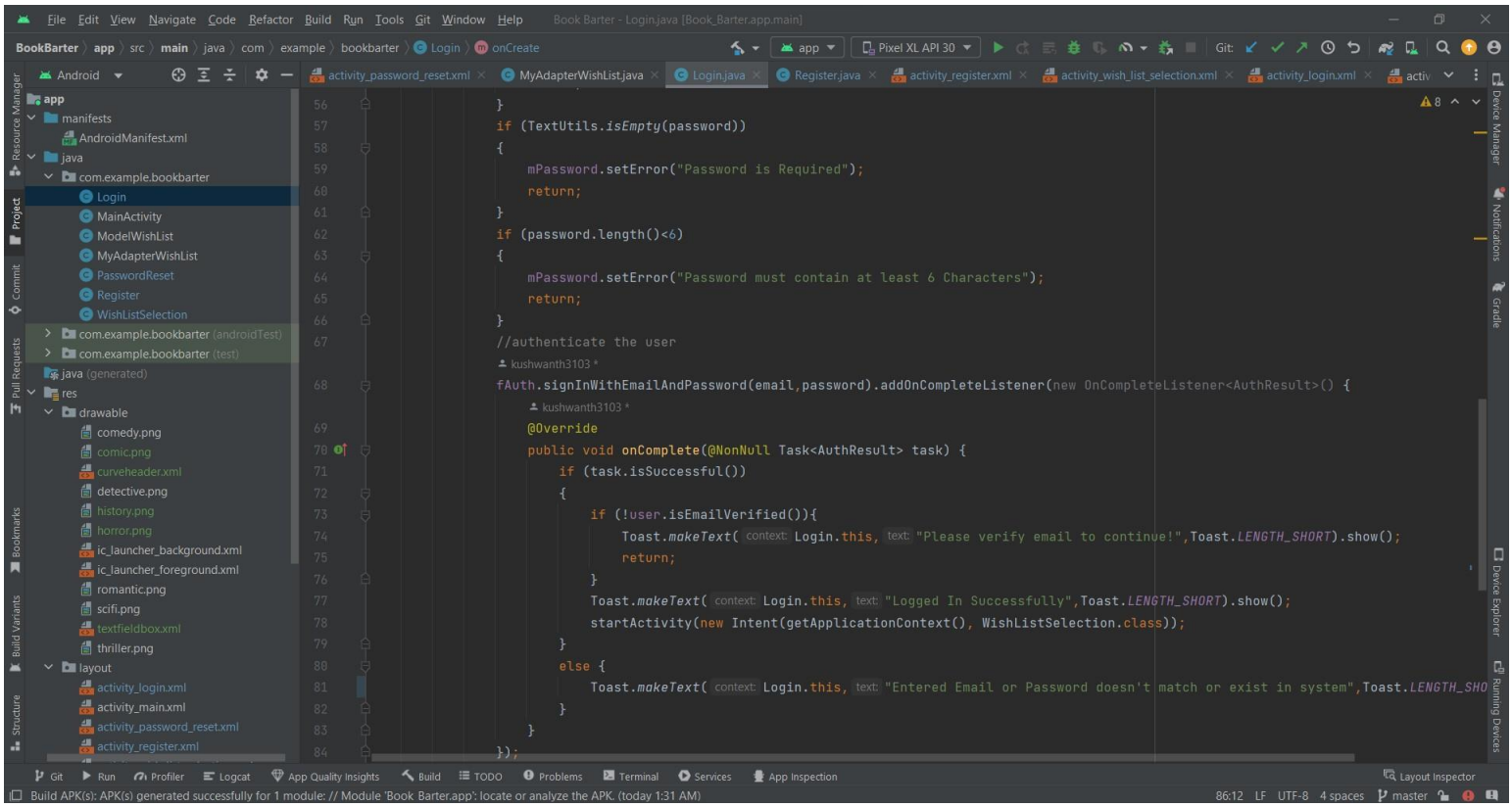
Sequence Diagram For Reset Password:



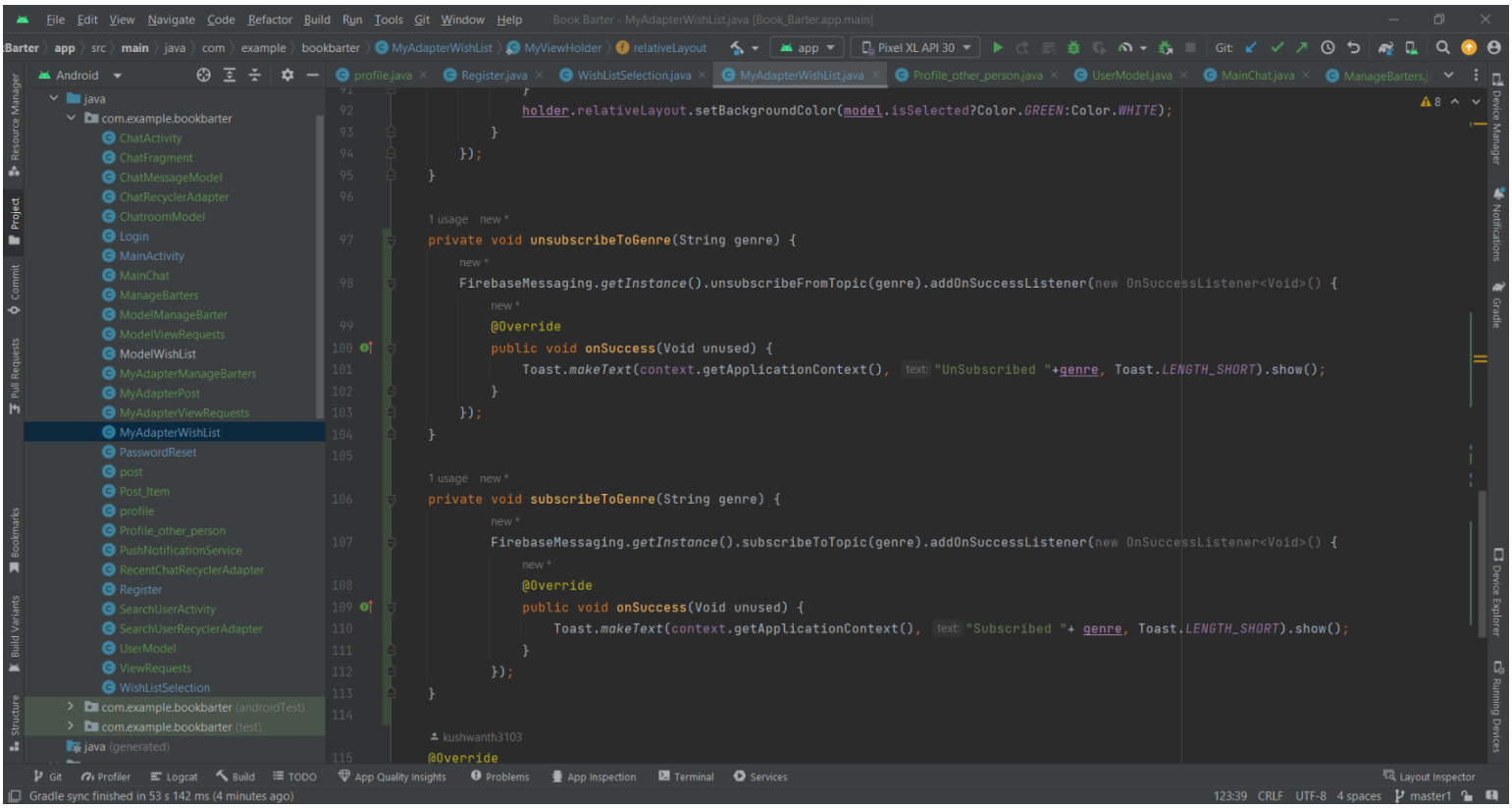
Design Class Diagram:



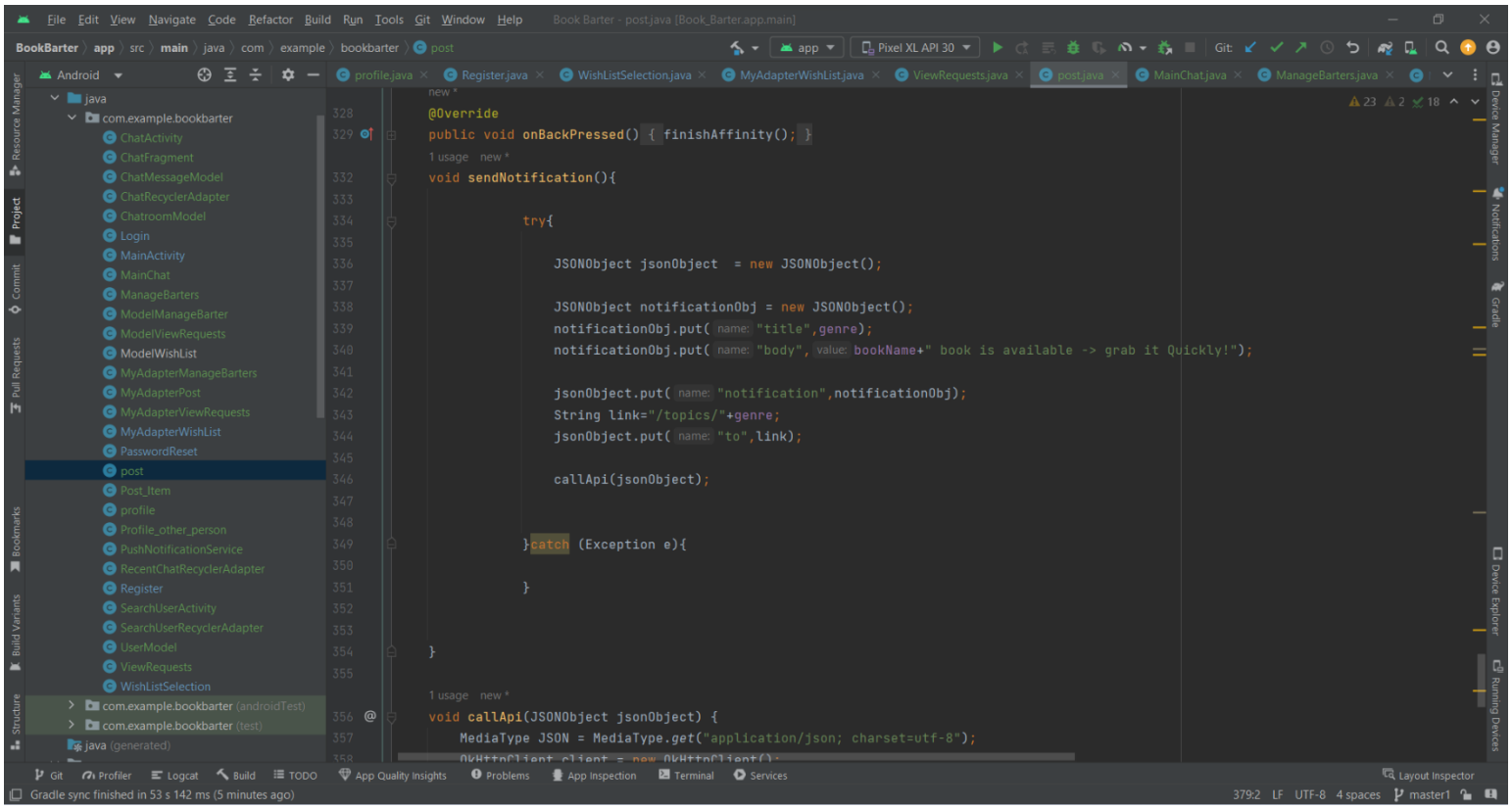
Code Snippets:



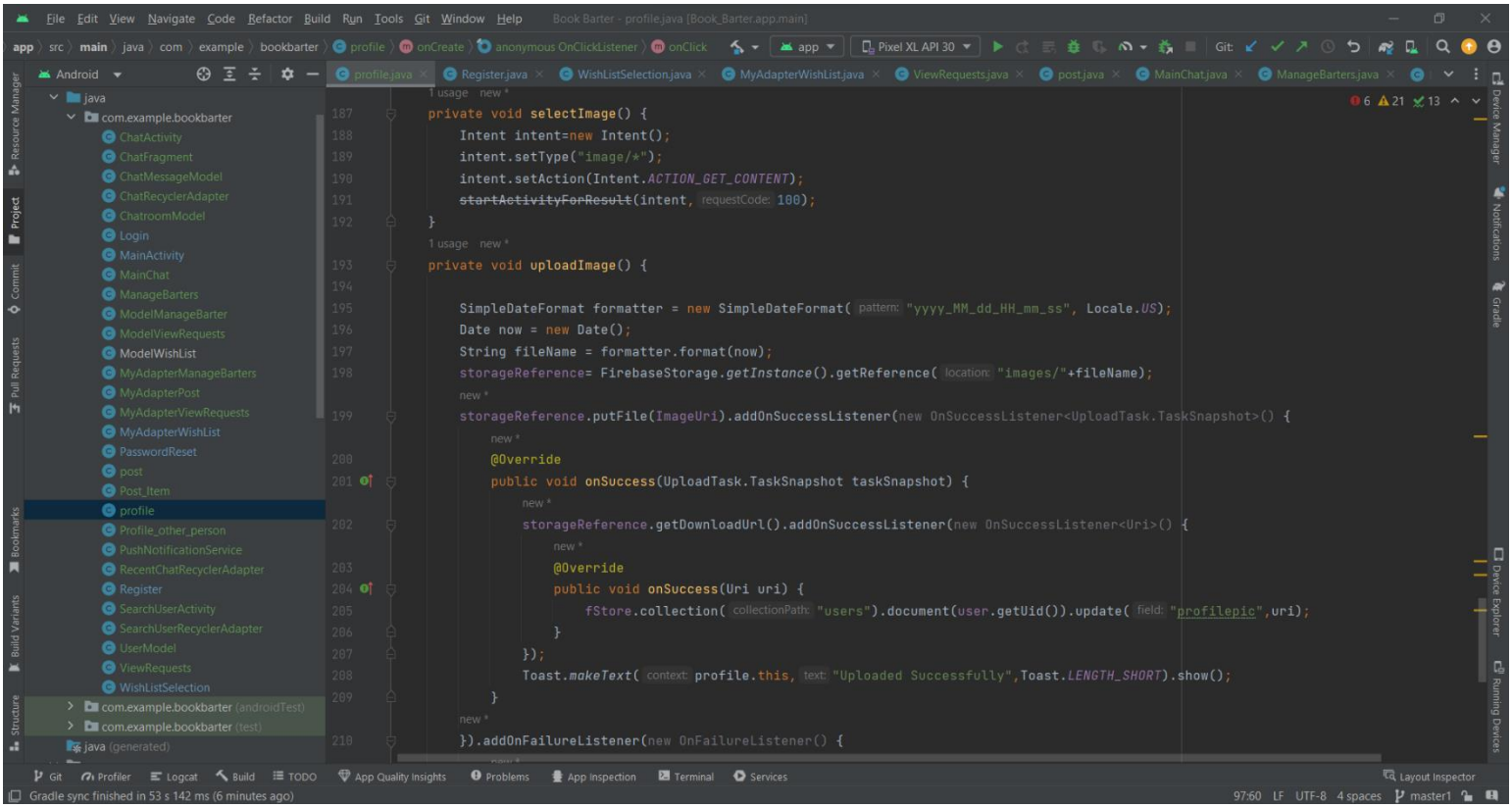
```
}
if (TextUtils.isEmpty(password))
{
    mPassword.setError("Password is Required");
    return;
}
if (password.length()<6)
{
    mPassword.setError("Password must contain at least 6 Characters");
    return;
}
//authenticate the user
kushwanth3103 *
fAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
    kushwanth3103 *
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if (task.isSuccessful())
        {
            if (!user.isEmailVerified()){
                Toast.makeText( context: Login.this, text: "Please verify email to continue!",Toast.LENGTH_SHORT).show();
                return;
            }
            Toast.makeText( context: Login.this, text: "Logged In Successfully",Toast.LENGTH_SHORT).show();
            startActivity(new Intent(getApplicationContext(), WishListSelection.class));
        }
        else {
            Toast.makeText( context: Login.this, text: "Entered Email or Password doesn't match or exist in system",Toast.LENGTH_SHO
        }
    }
});
```



```
        holder.relativeLayout.setBackgroundColor(model.isSelected?Color.GREEN:Color.WHITE);
    }
});
}

1 usage  new *
private void unsubscribeToGenre(String genre) {
    new *
    FirebaseMessaging.getInstance().unsubscribeFromTopic(genre).addOnSuccessListener(new OnSuccessListener<Void>() {
        new *
        @Override
        public void onSuccess(Void unused) {
            Toast.makeText(context.getApplicationContext(),  text: "UnSubscribed "+genre, Toast.LENGTH_SHORT).show();
        }
    });
}

1 usage  new *
private void subscribeToGenre(String genre) {
    new *
    FirebaseMessaging.getInstance().subscribeToTopic(genre).addOnSuccessListener(new OnSuccessListener<Void>() {
        new *
        @Override
        public void onSuccess(Void unused) {
            Toast.makeText(context.getApplicationContext(),  text: "Subscribed "+ genre, Toast.LENGTH_SHORT).show();
        }
    });
}

kushwanth3103
@Override
```



```
@Override
public void onBackPressed() { finishAffinity(); }

void sendNotification(){

            try{

                JSONObject jsonObject  = new JSONObject();

                JSONObject notificationObj = new JSONObject();
                notificationObj.put( name: "title",genre);
                notificationObj.put( name: "body", value: bookName+" book is available -> grab it Quickly!");

                jsonObject.put( name: "notification",notificationObj);
                String link="/topics/"+genre;
                jsonObject.put( name: "to",link);

                callApi(jsonObject);


            }catch (Exception e){

            }


}

void callApi(JSONObject jsonObject) {
    MediaType JSON = MediaType.get("application/json; charset=utf-8");
```



```
private void selectImage() {
    Intent intent=new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(intent, requestCode: 100);
}
private void uploadImage() {

    SimpleDateFormat formatter = new SimpleDateFormat( pattern: "yyyy_MM_dd_HH_mm_ss", Locale.US);
    Date now = new Date();
    String fileName = formatter.format(now);
    storageReference= FirebaseStorage.getInstance().getReference( location: "images/"+fileName);
    storageReference.putFile(ImageUri).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
            storageReference.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                @Override
                public void onSuccess(Uri uri) {
                    fStore.collection( collectionPath: "users").document(user.getUid()).update( field: "profilepic",uri);
                }
            });
            Toast.makeText( context: profile.this, text: "Uploaded Successfully",Toast.LENGTH_SHORT).show();
        }
    }).addOnFailureListener(new OnFailureListener() {
```

Demo Link

https://youtu.be/bVFFYQTWGHM