



Bilkent University

Department of Computer Engineering

Senior Design Project

Analysis Report

Project Name: BookClub

Group Members: Bikem Çamli
Mert Osman Dönmezyürek
Barış Eymür
Mahin Khankishizade
Deniz Şen

Supervisor: Assoc. Prof. Selim Aksoy

Jury Members: Prof. Dr. Fazlı Can
Assoc. Prof. Cigdem Gunduz Demir

Innovation Expert: Dr. Haluk Altunel

Analysis Report

Nov 12, 2018

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

1.Introduction	4
2. Current System	5
2.1 ThriftBooks	5
2.2 Amazon	5
2.3 GoodReads	5
2.4 Hepsiburada	6
3. Proposed System	6
3.1. Overview	6
3.2 Functional Requirements	8
3.3 Nonfunctional Requirements	8
3.3.1 Usability	8
3.3.2 Scalability	9
3.3.3 Extensibility	9
3.3.5 Reliability	9
3.3.6 Availability	10
3.3.7 Security	10
3.4. Pseudo Requirements	10
3.5. System Models	11
3.5.1. Scenarios	11
3.5.1.1 Cengiz Sets His Information	11
3.5.1.2 Cengiz Creates His Wishlist	11
3.5.1.3 Cengiz Edits His Tradelist	12
3.5.1.4 Cengiz Sells a Book	13
3.5.2. Use Case Model	15
3.5.2.1 Set User Information	15
3.5.2.2 Set a Profile Picture	16
3.5.2.3 Set Privacy/Security Information	17
3.5.2.4 Edit Profile	18
3.5.2.5 Create Tradelist	19
3.5.2.6 Create Wishlist	20
3.5.2.7 Edit Tradelist	20
3.5.2.8 Edit Wishlist	22
3.5.2.9 Add a Book to Wishlist via Heart Button	22
3.5.2.10 Sell a Book	23

3.5.2.11 View the Matches	24
3.5.2.12 Chat with Matched Users	25
3.5.2.13 View Suggestion List	25
3.5.2.14 Submit the Trade Completion	26
3.5.2.15 Comment on Another User	27
3.5.2.16 Filter	27
3.5.2.17 Search	28
3.5.2.18 Rate	29
3.5.2.19 Report	30
3.5.2.20 Block	30
3.5.2.21 View the Block List	31
3.5.3. Object and Class Model	33
3.5.4. Dynamic Models	36
3.5.4.1 Sequence Diagrams	36
3.5.4.1.1 Sign In Sequence Diagram	36
3.5.4.1.2 Log In Sequence Diagram	37
3.5.4.1.3 Add Book To Wishlist Sequence Diagram	37
3.5.4.1.4 Confirm Match Sequence Diagram	38
3.5.4.2 Activity Diagram	40
3.5.4.3. User Interface	43
3.5.4.3.1 Welcome Screen	43
3.5.4.3.2 Login Screen	44
3.5.4.3.3 Main Screen	45
3.5.4.3.4 Exchange Book Screen	46
3.5.4.3.5 Sell a Book Screen	47
3.5.4.3.6 Match List Screen	48
3.5.4.3.7 Suggestion List Screen	49
3.5.4.3.8 Preferences Screen	51
3.5.4.3.9 History Screen	52
3.5.4.3.10 Message Screen	53
3.5.4.3.11 Wishlist Screen	54
4. References	55

1. Introduction

Reading is a fun and useful activity: we can learn new things by reading a scientific book or we can find ourselves in a totally new fictional world by reading a novel. Unfortunately, every beautiful thing comes with a price. Wouldn't it be nice to exchange our books with our friends' books, instead of paying? If we did that, we would have lessened the price of reading significantly. There are groups of people around the world who have noticed the advantages of book sharing. These people have formed book sharing groups, called book clubs.

The main aim of this project is to create a book trading platform which will serve as a universal "BookClub". The users of this mobile app will be able to exchange their used books with books of other users and they will also be able to buy second-hand books from other users by paying a reasonable price. The users will specify the books that they desire to give away or sell and the ones they want to read. Using this information, BookClub will match users with the owners of books they are looking for. While doing this, it will find the best possible match for a user by considering user locations, user ratings, book prices and some other attributes. Additionally, it will make personal suggestions to its users by trying to understand their taste of books.

This report starts with giving information about some current systems which are similar to BookClub. It continues with requirements of our proposed system, which are functional requirements, nonfunctional requirements and pseudo requirements. Then, the system model of BookClub is explained using scenarios, use case model, object and class model, sequence diagrams and an activity diagram. Mockups of the user interface follows the system model. Finally, there are pages for References.

2. Current System

In this section of the report, similar products to our project will be discussed.

2.1 ThriftBooks [1]

- Users can buy both used and new books.
- Users can review other users.
- Users can comment on books.
- Users should pay a shipping price.

2.2 Amazon [2]

- Users can sell their used and new books.
- Amazon has a product suggestion system however it is not specific for books.
- Users can comment and rate other users.
- Users may need to pay a shipping price for some books.

2.3 GoodReads [3]

- GoodReads is a comprehensive website for book lovers.
- Users can search for books by various book attributes.
- Users can recommend a book to their friends.

2.4 Hepsiburada [4]

- Users can sell their new and second-hand books.
- Users can get product recommendation based on their previous purchases.
- Users can comment and rate both users and products.
- Some products may need a shipping price.

FEATURES	BookClub	ThriftBooks	Amazon	GoodReads	Hepsiburada
Selling Books	✓	✓	✓		✓
Exchanging Books	✓				
Book suggestions based on past search	✓		✓		✓
Search by all possible book attribute	✓			✓	
Match system between customers	✓				
Book suggestions based on user's profile and interests	✓				
User Review System	✓	✓	✓		✓
Book Review System	✓	✓			✓
Shipping Price		✓	✓		✓

Figure: Comparison between platforms

3. Proposed System

3.1. Overview

BookClub is an innovative mobile application which can be downloaded for free from Google Play Store. The main difference of BookClub from the similar applications/websites is that in this app, users can exchange the book they desire with a unique match/suggestion algorithm. The users log in/register to the app and create lists of the books they can give away and the books they desire to obtain. Then, the user will get matched with other users. In this situation, by clicking the match screen, they will see the list of matches, the location of the books, the owners and their reviews. They can easily contact the user and arrange a meeting where they will be able to exchange books. This is an advantage for both sides since the users will be worried about neither the safety of the book nor its condition, due to viewing the owner's review and reaching them. No shipment price, no worries

about the reliability of the other user, no pirate versions and no black market. Moreover, the users can sell their books in BookClub, yet, the price has a limit (not higher than the original book's price). There are also cases of not getting the best match, in this case, BookClub's innovative match & suggestion algorithm offers users new books that they may like, by analyzing their search history and their book wishlist. The suggestion algorithm works as follows: assume a case where a user does not need any new book, but they have several books they would like to give away. In this scenario, BookClub analyzes the user's "giveaway" books and examines their preferences. Later, in the main screen, the user will constantly see the books similar to their likes.

Overall, BookClub is an innovative and accessible mobile app that will create an opportunity for book lovers to find new books or trade their books with a simpler method. The aim of the app is to ease book lovers' lives by creating a convenience for them to swap their original books or get them for a cheaper price. With the original and fast match/suggestion algorithm it will decrease both the black market sales and book piracy. At the same time, the app will provide the users with the books they desire.

3.2 Functional Requirements

- The user can add the books he/she is willing to use as a trade material and specify whether it is for sale, for exchange or both.
- The user can make a list of books he/she would like to take or buy.
- The user can edit the list of books that they have listed for sale or exchange and that they wish to obtain.
- The user can change the preferred method of transaction, such as changing the condition from exchanging to selling.
- The user can set the prices of their books that are for sale.

- The user can search for books by filtering their titles, authors, prices, publishers, etc.
- The user can see their matched books.
- The user can see the books that were suggested to them by the system.
- The user can get into contact with another user that was matched for a trade after both sides approve the match.
- The user can rate and/or report another user if they are matched.
- After the exchange is performed, both ends should submit the completion of the trade to BookClub.
- The user can write comments about the other user if both of them approve that the book exchange is successfully performed.

3.3 Nonfunctional Requirements

3.3.1 Usability

- Since BookClub has the customer range from all ages and various groups of people, the application should have a user-friendly interface for all of these potential user types.
- The application should include an explicit user manual which consists of information about the usage of BookClub and the details about matching and suggesting.

3.3.2 Scalability

- As the number of people increments, the application should meet the increased number of users and their demands.
- The system should be powerful enough to be able to store the books that are newly added by the users.

3.3.3 Extensibility

- In our application, we will take the comments of the users into consideration to improve and modify the system according to the users' requests and views.
- Making updates and adding new features is necessary for an application to satisfy the demands of users.

3.3.4 Efficiency

- The system can have user intensity. The application should work efficiently regardless of the number of people using it.
- The system can have a high number of books, therefore, the application should match the users in the best way and as quick as possible.
- When a user is adding a book, the authenticity check time of the book should be minimized.

3.3.5 Reliability

- Match algorithm should prioritize the perfect matched book in the system with the user's preference and give the accurate book match to the user.
- If there is no matched book for a user preference, match algorithm should give suggestions that are suitable with the user's book taste.
- The application should not include any pirate books.
- The second-hand book's price should not exceed the price of the first-hand book.

3.3.6 Availability

- BookClub should be available on Google Play Store.
- The system should be available for user access every time.

3.3.7 Security

- The system should protect the private information of the user such as location, email, password, recent sales and search history.

3.4. Pseudo Requirements

- The project will be collaborated in GitHub.
- The project webpage with all the information and updates is <https://bookclubapp.github.io/>
- The project will be an Android mobile application that is supported via client-server architecture.
- The project's information is going to be stored in MySQL - MariaDB database.
- The client side of the project is going to be implemented in Java.
- The server side of the project is going to be implemented in Python.
- The API between client and server is going to be implemented in RESTful API [5].

3.5. System Models

3.5.1. Scenarios

3.5.1.1 Cengiz Sets His Information

Use Case: Set User Information

Primary Actor: Cengiz

Entry-conditions:

- Cengiz registers to the application
- Cengiz validates his/her e-mail address

Exit condition:

- Cengiz completes the application and starts using the application freely

Success Scenario Event Flow:

1. Cengiz installs the application
2. Cengiz registers to the application
3. The system requires a profile validation
4. After validation, the system requires a profile information
5. Cengiz fills the profile information required by the system
6. Cengiz can now use the application freely

3.5.1.2 Cengiz Creates His Wishlist

Use Case: Create Wishlist

Primary Actor: Cengiz

Entry-conditions:

- Cengiz clicks the Wishlist Screen

Exit conditions:

- Cengiz adds a new book to the Wishlist

Success Scenario Event Flow:

1. Cengiz clicks the Profile Screen
2. Cengiz clicks the Wishlist Screen
3. Cengiz clicks “add” button to add a new book that he would like to have
4. Cengiz enters the new book’s required information and if the book exists in the system’s database the book is being added as a wanted book

5. The system uses a book instance to find the best match for Cengiz

3.5.1.3 Cengiz Edits His Tradelist

Use Case: Edit Tradelist

Primary Actor: Cengiz

Entry-conditions:

- Cengiz clicks the Tradelist Screen from his profile

Exit conditions:

- Cengiz edits the Tradelist

Main Event Flow:

1. Cengiz clicks the Profile Screen
2. Cengiz clicks the Tradelist Screen
3. Cengiz presses on the book instance and chooses 'edit' button
4. Cengiz changes the book's information
5. Cengiz completes the edit and saves the changes
6. The system uses new information for the best match algorithm

3.5.1.4 Cengiz Sells a Book

Use Case: Sell a Book

Primary Actor: Cengiz

Entry-conditions:

- Cengiz is on the Sell Screen

Exit conditions:

- Cengiz clicks the publish button

Main Event Flow:

1. Cengiz clicks the plus button on the Main Screen

2. Cengiz selects the Sell button from the addition screen
3. Cengiz fills the information of the book
4. Cengiz sets the price
5. Cengiz clicks the publish button to save and publish the book

3.5.1.5 Cengiz Chats with the Matched User, Bilal

Use Case: Chat with Matched Users

Actors: Cengiz and Bilal

Entry-conditions:

- Cengiz swipes the screen one time to be on the Match Screen
- Cengiz should swipe the Match Screen to the left to open the chat with the user Bilal

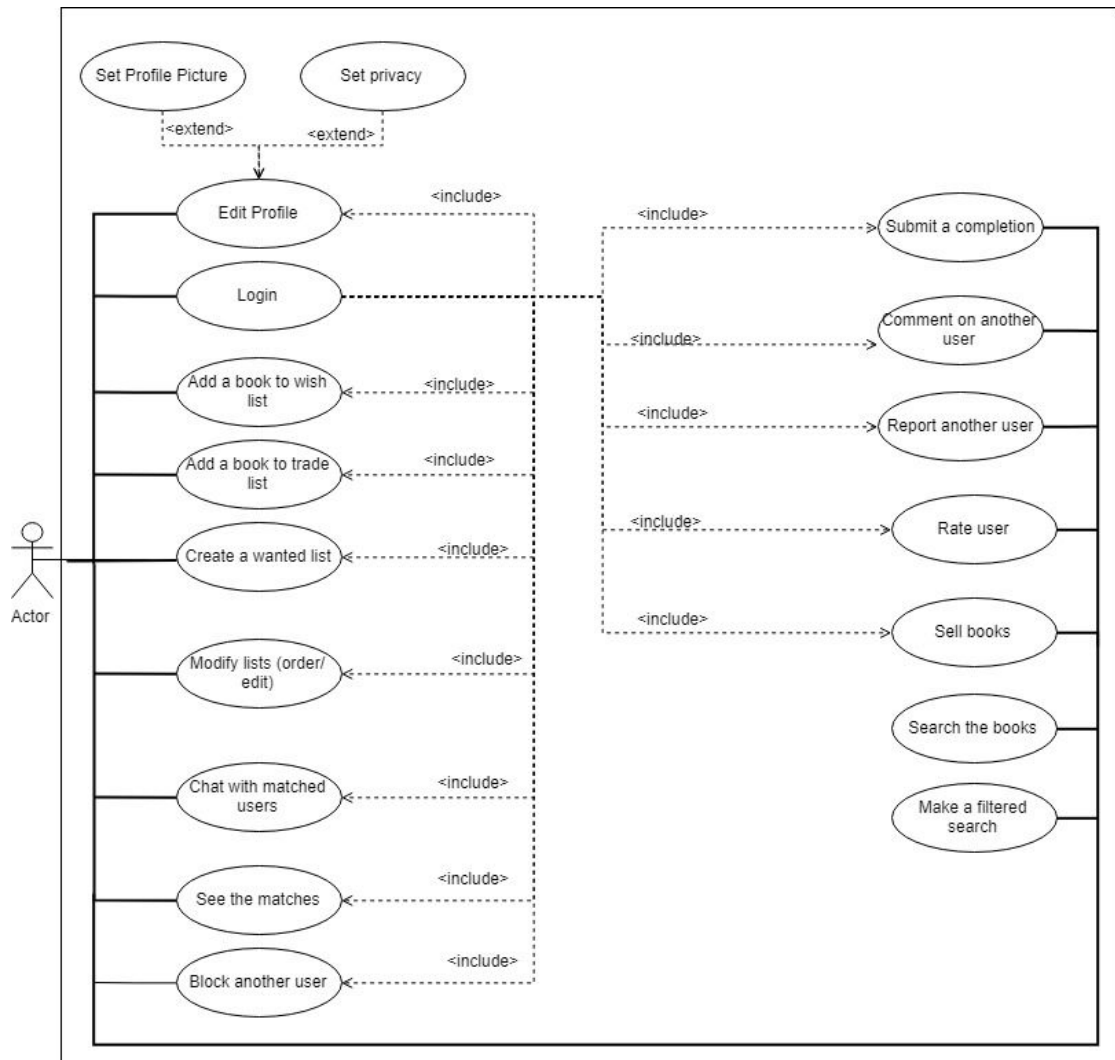
Exit conditions:

- Cengiz chats with Bilal

Main Event Flow:

1. Cengiz views the Match list
2. Cengiz swipes the match that was convenient for himself/herself
3. The chat section for Cengiz and Bilal opens automatically
4. Cengiz chats with Bilal

3.5.2. Use Case Model



3.5.2.1 Set User Information

Use Case: Set User Information

Primary Actor: User

Entry-conditions:

- The user should register to the application
- The user should validate his/her e-mail address

Exit conditions:

- User logs out from the application without filling the required information
- The user presses the home button without filling the required information
- User deletes the application without filling the required information
- The user completes the application and starts using the application freely

Main Event Flow:

1. The user installs the application
2. User registers to the application
3. The system requires a profile validation
4. After validation, the system requires a profile information
5. The user fills the profile information required by the system
6. The user now can use the application freely

Alternative Event Flow:

1. The user installs the application
2. The user registers to the application
3. The system requires a profile validation
4. After validation, the system requires a profile information
5. The user refuses to fill the required information
6. The user fails to use the application

3.5.2.2 Set a Profile Picture

Use Case: Set a Profile Picture

Primary Actor: User

Entry-conditions:

- The user should already be registered to the application
- The user should fill the system's required information

- The user can click the settings button on the top of any screen in the application, then click the profile picture section

Exit conditions:

- The user has a profile picture
- Other users may see the user's profile picture
- The user will be able to change the profile picture in the future

Main Event Flow:

1. The user fills the required information in the profile creation stage
2. The system suggests the user to add a profile picture
3. The user chooses a picture from the phone's gallery
4. The system adjusts the chosen picture to its profile picture standards and sets the profile picture for the user

Alternative Event Flow:

1. The user fills the required information in the profile creation stage
2. The user never sets a profile picture

3.5.2.3 Set Privacy/Security Information

Use Case: Set Privacy/Security Information

Primary Actor: User

Entry-conditions:

- The user should already be registered to the application
- The user should fill the system's required information
- The user can click the settings button on the top of any screen in the application, then click the profile privacy/security section

Exit conditions:

- The user sets the privacy/security settings depending on their preferences

- Depending on the new privacy/security preferences, other users will have restricted actions on this user
- The user will be able to change the settings in the future

Main Event Flow:

1. The user fills the required information after registration
2. User clicks the settings on the top of any screen in the application
3. User chooses the privacy/security section
4. The user may choose various settings such as hiding their information, profile picture or to be invisible in the application
5. The user may use the application after saving the settings

Alternative Event Flow:

1. The user fills the required information after registration
2. User never sets his/her privacy/security information

3.5.2.4 Edit Profile

Use Case: Edit Profile

Primary Actor: User

Entry-conditions:

- The user should be on the Settings screen
- The user should already have a filled profile information or other information such as profile picture, privacy/security information so that they could be changed

Exit conditions:

- User changes desired parts of the profile information and save the new settings
- User returns to the Main Screen

Main Event Flow:

1. User clicks the settings on the top corner of any screen inside the application
2. The user chooses any profile section he/she wants to edit
3. Depending on the section, the system requires new information from the user
4. User edits the information
5. The user presses the “save changes” button
6. User returns to the main menu

Alternative Event Flow:

1. User never edits his/her profile

3.5.2.5 Create Tradelist

Use Case: Create Tradelist

Primary Actor: User

Entry-conditions:

- The user should be on the Profile Screen
- The user should click the Tradelist Screen

Exit conditions:

- The user adds a new book to the Tradelist
- The system uses this book for creating new matches

Main Event Flow:

1. The user clicks the Profile Screen
2. The user clicks the Tradelist Screen
3. User clicks “add” button to add a new book for a trade
4. The user enters the new book’s required information and if the book exists in the system’s database the book is being added as a trade
5. The system uses a book instance to find the best match for the user

Alternative Event Flow:

There is no alternative event flow for this particular use case. Since tradelist is one of the main parts of the application, user has to create it.

3.5.2.6 Create Wishlist

Use Case: Create Wishlist

Primary Actor: User

Entry-conditions:

- The user should be on the Profile Screen
- The user should click the Wishlist Screen

Exit conditions:

- The user adds a new book to the Wishlist
- The system uses this book for creating new matches

Success Scenario Event Flow:

1. The user clicks the Profile Screen
2. The user clicks the Wishlist Screen
3. User clicks “add” button to add a new book that they would like to have
4. The user enters the new book’s required information and if the book exists in the system’s database the book is being added as a wanted book
5. The system uses a book instance to find the best match for the user

Alternative Event Flow:

There is no alternative event flow for this particular use case. Since wishlist is one of the main parts of the application, user has to create it.

3.5.2.7 Edit Tradelist

Use Case: Edit Tradelist

Primary Actor: User

Entry-conditions:

- The user should be on the Profile Screen
- The user should click the Tradelist Screen

Exit conditions:

- User edits the book's information.
- User deletes the book instance
- User changes the book instance's transaction settings (exchange/sale)
- User edits the Tradelist

Main Event Flow:

1. The user clicks the Profile Screen
2. The user clicks the Tradelist Screen
3. The user presses on the book instance and chooses 'edit' button
4. The user changes the book's information
5. The user completes the edit and saves the changes
6. The system uses new information for the best match algorithm

Alternative Event Flow:

1. The user clicks the Profile Screen
2. The user clicks the Tradelist Screen
3. The user presses on the book instance and chooses edit button
4.
 1. The user deletes the book
 2. The user changes the trade setting from exchange to sale
 3. The user changes the trade setting from sale to exchange
 4. The user edits the book's price
5. The user completes the edit and saves the changes
6. The system uses the new information for the best match algorithm

3.5.2.8 Edit Wishlist

Use Case: Edit Wishlist

Primary Actor: User

Entry-conditions:

- The user should be on the Profile Screen
- The user should click the Wishlist Screen

Exit conditions:

- The user finished editing the Wishlist

Main Event Flow:

1. The user clicks the Profile Screen
2. The user clicks the Wishlist Screen
3. The user presses on the book instance
4. The user presses the edit button
5. The user deletes the book from the list.
6. The user completes the edit and saves the changes
7. The system uses new information for the best match algorithm

Alternative Event Flow:

1. The user clicks the Profile Screen
2. The user clicks the Wishlist Screen
3. The user presses on the book instance
4. The user drags the book to the top of the list, orders the wishlist
5. The user completes the edit and saves the changes
6. The system uses new information for the best match algorithm

3.5.2.9 Add a Book to Wishlist via Heart Button

Use Case: Add a Book to Wishlist via Heart Button

Primary Actor: User

Entry-conditions:

- The user should be on the Main Screen and see the list of books
- The user can search books and see the list of books
- The user can filter books and see the list of books

Exit conditions:

- User clicks the heart button on the book instance and adds it to the Wishlist

Main Event Flow:

1. User search a book they want to have
2. The user sees the list of the books on the screen
3. User clicks on the heart button located on the book instance
4. The system adds the hearted books to the user's Wishlist
5. The system uses the new book instance in finding best matches

Alternative Event Flow:

There is no alternative event flow for this particular case. The user either adds the book to the wishlist or not.

3.5.2.10 Sell a Book

Use Case: Sell a Book

Primary Actor: User

Entry-conditions:

- The user should be on the Sell Screen

Exit conditions:

- User clicks the publish button

Main Event Flow:

1. The user clicks the plus button on the Main Screen
2. The user selects the Sell button from the addition screen

3. The user fills the information of the book
4. The user sets the price
5. The user clicks the publish button to save and publish the book

Alternative Event Flow:

1. The user enters the Tradelist screen
2. The user presses on the book instance
3. The user clicks the edit button
4. The user changes the book's trade setting from exchange to sale
5. The user sets a price for the book
6. The user saves and completes the sale

3.5.2.11 View the Matches

Use Case: View the Matches

Primary Actor: User

Entry-conditions:

- The user should be on the Main Screen
- The user swipes the screen one time to be on the Match Screen

Exit conditions:

- The user views the Match list in decreasing order: from best to worst matches.

Main Event Flow:

1. The user swipes the Main Screen one time
2. The user views the Match list from the best to the worst matches
3. The user clicks the Match to see its details

Alternative Event Flow:

1. The user enters the Profile Screen
2. The user clicks the Match/Suggestion screen

3. The user views the match list

3.5.2.12 Chat with Matched Users

Use Case: Chat with Matched Users

Primary Actor: User

Entry-conditions:

- The user swipes the screen one time to be on the Match Screen
- The user should swipe the Match Screen to the left to open the chat with the user

Exit conditions:

- User chats with a user of the other book(s)

Main Event Flow:

1. The user views the Match list
2. The user swipes the match that was convenient for himself/herself
3. The chat section of two users opens automatically
4. The user chats with another user

Alternative Event Flow:

There is no specific alternative for this use case. The chatting feature is extremely confidential and only happens if the both sides of the trade transaction approve the match.

3.5.2.13 View Suggestion List

Use Case: View Suggestion List

Primary Actor: User

Entry-conditions:

- The user should be on the Main Screen
- The user swipes the screen two times to be on the Suggestion Screen

Exit conditions:

- The user views the Suggestion list

Main Event Flow:

1. The user swipes the Main Screen two times
2. The user views the Suggestion list
3. User clicks the Suggestion to see its details

Alternative Event Flow:

1. The user enters the Profile Screen
2. The user clicks the Match/Suggestion screen
3. The user views the match list

3.5.2.14 Submit the Trade Completion

Use Case: Submit the Trade Completion

Primary Actor: User

Entry-conditions:

- The user should trade/exchange the books in order to submit the completion of the trade

Exit conditions:

- User submits the Trade Completion

Main Event Flow:

1. After exchanging/selling the book to another user, the system requires the user to submit the completion of the trade
2. The user clicks the submit button and submits the completion.
3. The system deletes the match from both users

Alternative Event Flow:

1. After exchanging/selling the book to another user, the system requires the user to submit the completion of the trade
2. The user submits the completion, however the other user does not
3. The system pushes notification to the other user to submit completion
4. The other user does not submit
5. The system warns the other user
6. The system closes the trade after a while

3.5.2.15 Comment on Another User

Use Case: Comment on Another User

Primary Actor: User

Entry-conditions:

- The users should submit the trade completion in order to be able to comment on another user

Exit conditions:

- The user saves and publishes the comment

Main Event Flow:

1. After a user submits the trade completion, the system suggests the user to comment on the user that he/she had a trade with
2. The user comments on the user
3. The system uses this comment for enhancing the match and suggest algorithm

Alternative Event Flow:

1. The user never comments on another user

3.5.2.16 Filter

Use Case: Filter

Primary Actor: User

Entry-conditions:

- The user should be on the Main Screen
- The user should click the filtering option located on the top of the screen

Exit conditions:

- The user filters the books
- The user may clear filter in order to reset the book list

Main Event Flow:

1. User clicks the filtering option located on the Main Screen
2. User filters the books depending on its name, author name, published year, genre, owner, owner's rating or the location of the owner
3. The system filters the books and creates a new book list for the user
4. The user sees the list of the filtered books

Alternative Event Flow:

There is no alternative for this use case.

3.5.2.17 Search

Use Case: Search

Primary Actor: User

Entry-conditions:

- The user should be on the Main Screen
- The user should click the searching option located on the top of the screen

Exit conditions:

- The user searches a book
- The user may clear search in order to return the initial booklist

Main Event Flow:

1. User clicks the searching option located on the Main Screen
2. The user can search a book via its name, author name, published year or location
3. The system searches the books using the search query and creates a new book list for the user
4. The user sees the list of the books identical to the search query

Alternative Event Flow:

There is no alternative for this use case

3.5.2.18 Rate

Use Case: Rate

Primary Actor: User

Entry-conditions:

- The users should submit the trade completion in order to be able to rate another user

Exit conditions:

- The user rates another user

Main Event Flow:

1. After a user submits the trade completion, the system suggests the user to rate on the user that he/she had a trade with
2. The user rates another user
3. The system uses this rating score for enhancing the match and suggest algorithm

Alternative Event Flow:

1. The user does not rate the other user

3.5.2.19 Report

Use Case: Report

Primary Actor: User

Entry-conditions:

- The user saw an inappropriate behavior from another user towards himself/herself or others
- The user was scammed by another user

Exit conditions:

- The user reports another user and provides the explanation of the report

Main Event Flow:

1. After receiving the best match, the user starts talking to another user in the chat section
2. The other user talks in a bad manner
3. The user reports the other user
4. The user provides the report explanation
5. The system analyzes the report and its explanation
6. The system warns or bans the other user

Alternative Event Flow:

1. The user does not report the other user

3.5.2.20 Block

Use Case: Block

Primary Actor: User

Entry-conditions:

- The user saw an inappropriate behavior from another user towards himself/herself

Exit conditions:

- The user blocks another user

Main Event Flow:

1. After receiving the best match, the user starts talking to another user in the chat section
2. The other user talks in a bad manner
3. The user blocks the other user
4. The other user can no longer see or text the user

Alternative Event Flow:

1. The user does not block the other user

3.5.2.21 View the Block List

Use Case: View the Block List

Primary Actor: User

Entry-conditions:

- The user should be on the Profile Screen
- The user should click the Block List
- The user should block at least one user in order to view the block list

Exit conditions:

- The user views the block list
- The user unblocks the blocked user

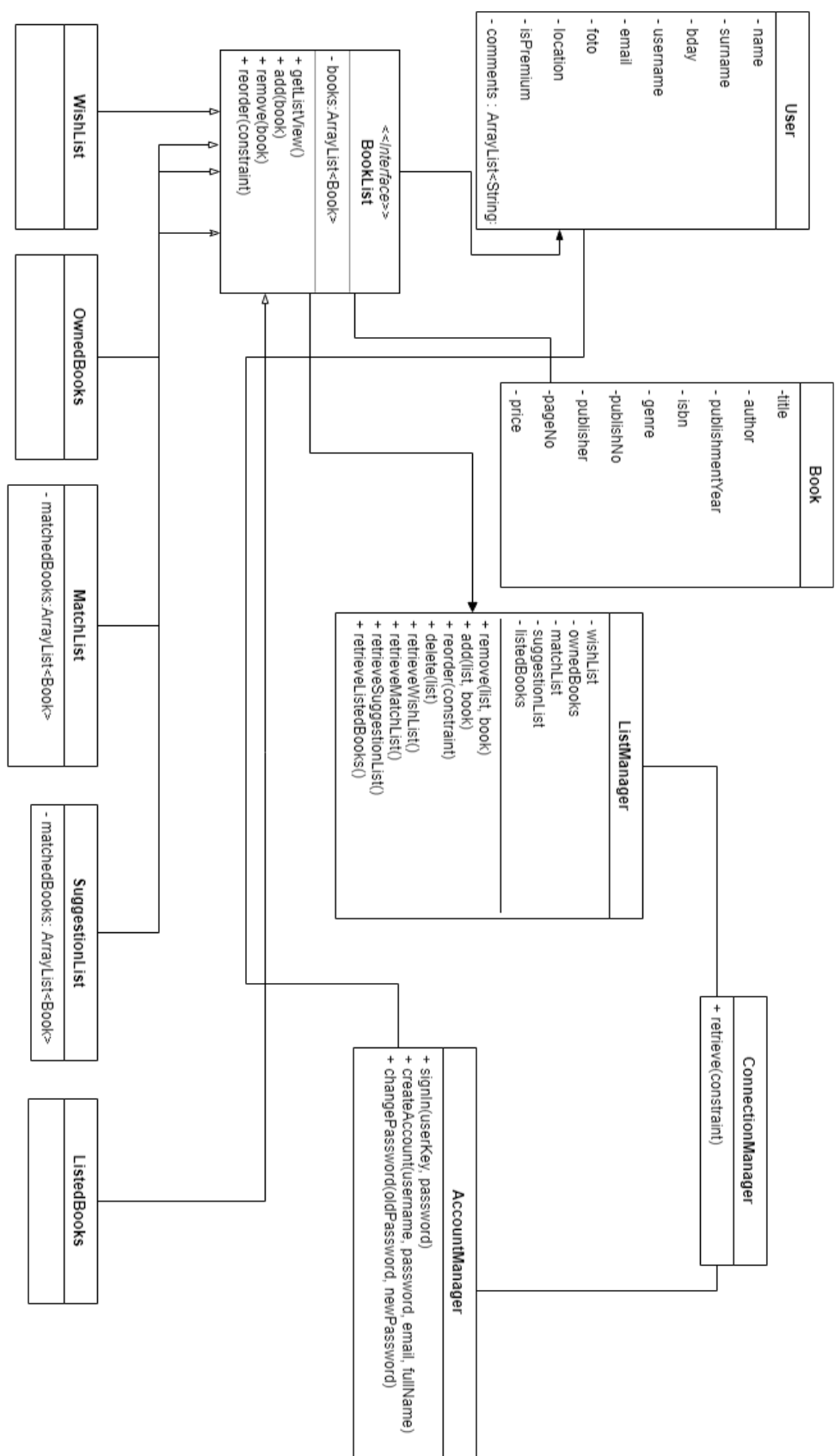
Main Event Flow:

1. The user clicks the Profile button on the top of any screen in the application
2. The user clicks the Block list
3. The user views the list of the users that have been blocked by him/her
4. The user returns to the Main Screen

Alternative Event Flow:

1. The user clicks the Profile button on the top of any screen in the application
2. The user clicks the Block list
3. The user unblocks a user
4. The user returns to the Main Screen

3.5.3. Object and Class Model



3.5.3.1 ConnectionManager Class

This class will retrieve information from the server and send information to the server from the client side.

3.5.3.2 AccountManager Class

This class will be used to give authentication to an existing user and creating a new account for a user.

3.5.3.3 User Class

This class is representing a user of BookClub. It has user-specific information like name, surname, email, password, etc.

3.5.3.4 Book Class

This class represents a book object in the system of BookClub. It contains book-specific information, such as book title, author, publisher, price etc.

3.5.3.5 BookList Abstract Class

This abstract class will represent the common functionalities of different kinds of book lists. Each kind of book list extends this abstract class.

3.5.3.6 ListedBooks Class

This class is created to ease the procedure of adapting the list of all the books currently on sale.

3.5.3.7 WishList Class

Each user has a wish list which will hold the books that the user wants to retrieve in a trade. This class will hold the objects of these books.

3.5.3.8 OwnedBooks Class

Each user has a list of books that they own and use as a trade material. This will hold the objects of these books.

3.5.3.9 MatchList Class

This class will hold a list of books of other users' books which are matched by the system. This list is different from the list taken from the abstract parent. The content of the list gotten by the abstract parent will have the current user's matched books.

3.5.3.10 SuggestionList Class

This class will also have a list different from the parent's data structure. Again, it will contain the suggested book instances of the other people's books. The content can differ according to the type of suggestion, i.e. exchange or sell.

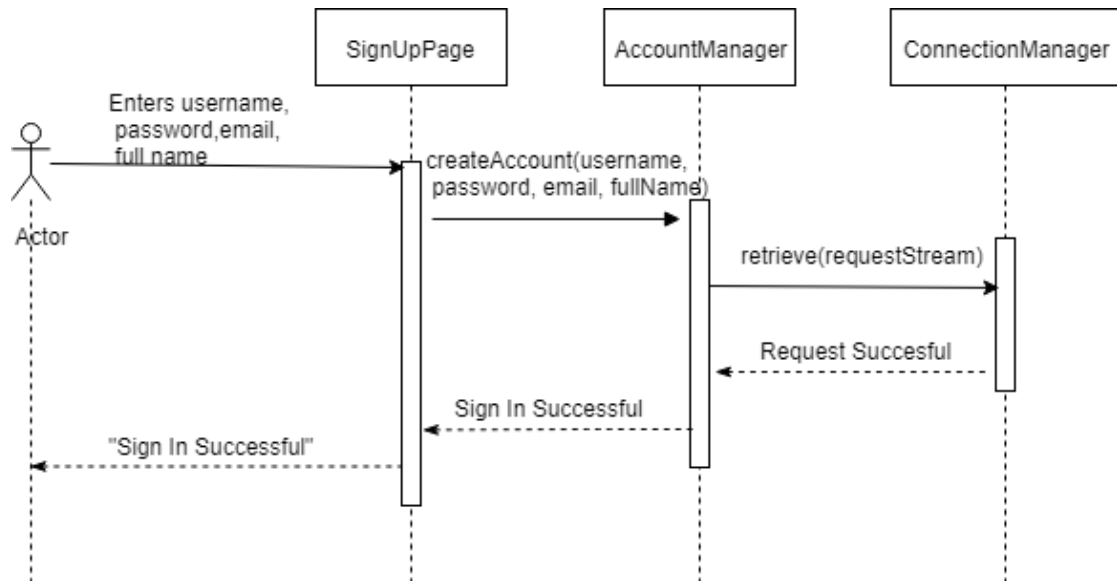
3.5.3.11 ListManager Class

This static class will be the interface between the ConnectionManager class and the classes that extend BookList. The functionality will include the generation of the request stream regarding BookList operations for the ConnectionManager.

3.5.4. Dynamic Models

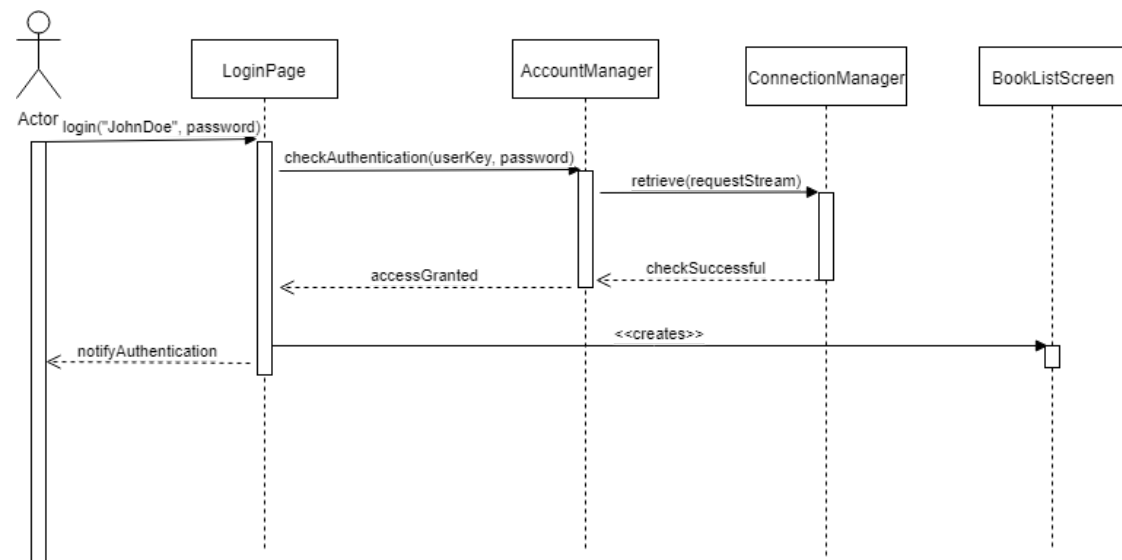
3.5.4.1 Sequence Diagrams

3.5.4.1.1 Sign In Sequence Diagram



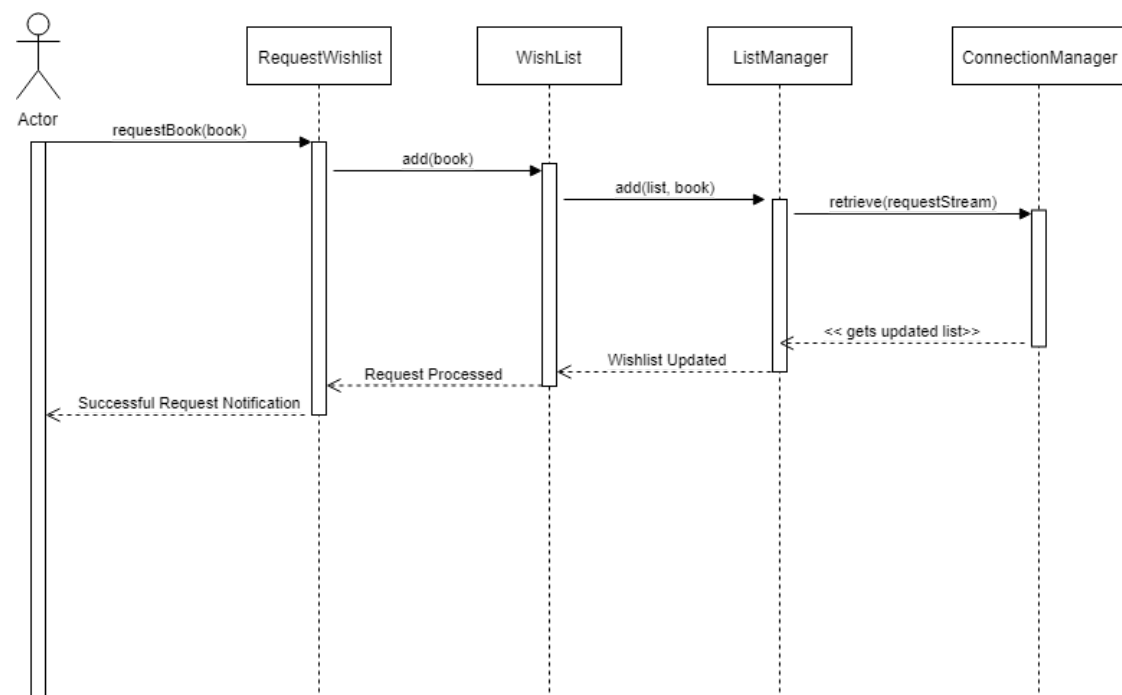
This sequence diagram shows the procedure of creating an account for BookClub of an actor. He first types his mandatory information and proceeds. Then, AccountManager communicates with ConnectionManager with the required request stream and ConnectionManager returns the state of the operation.

3.5.4.1.2 Log In Sequence Diagram



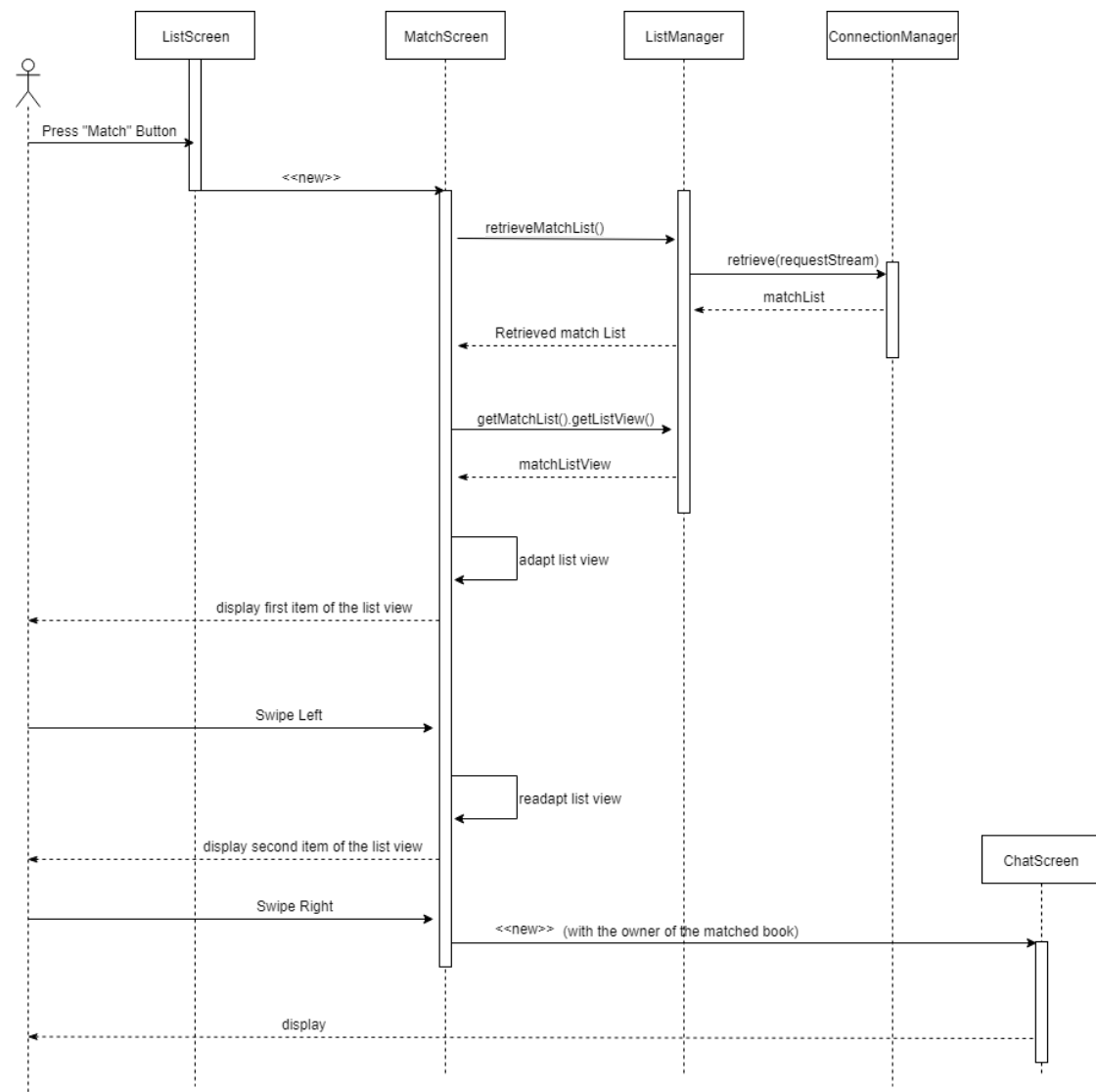
The diagram above shows the sequence of logging in BookClub. The actor enters his credentials and proceeds. AccountManager communicates with ConnectionManager and returns with a response. As the credentials are satisfactory and correct, the actor is directed to the BookListScreen.

3.5.4.1.3 Add Book To Wishlist Sequence Diagram



The diagram above shows the operation of an actor adding a book to his wishlist. He chooses his book of preference on RequestWishlist screen. This book is added to the WishList object; thanks to ListManager and ConnectionManager, the addition is done on the server side. ConnectionManager returns a response as an updated WishList object. The actor is then notified about the change.

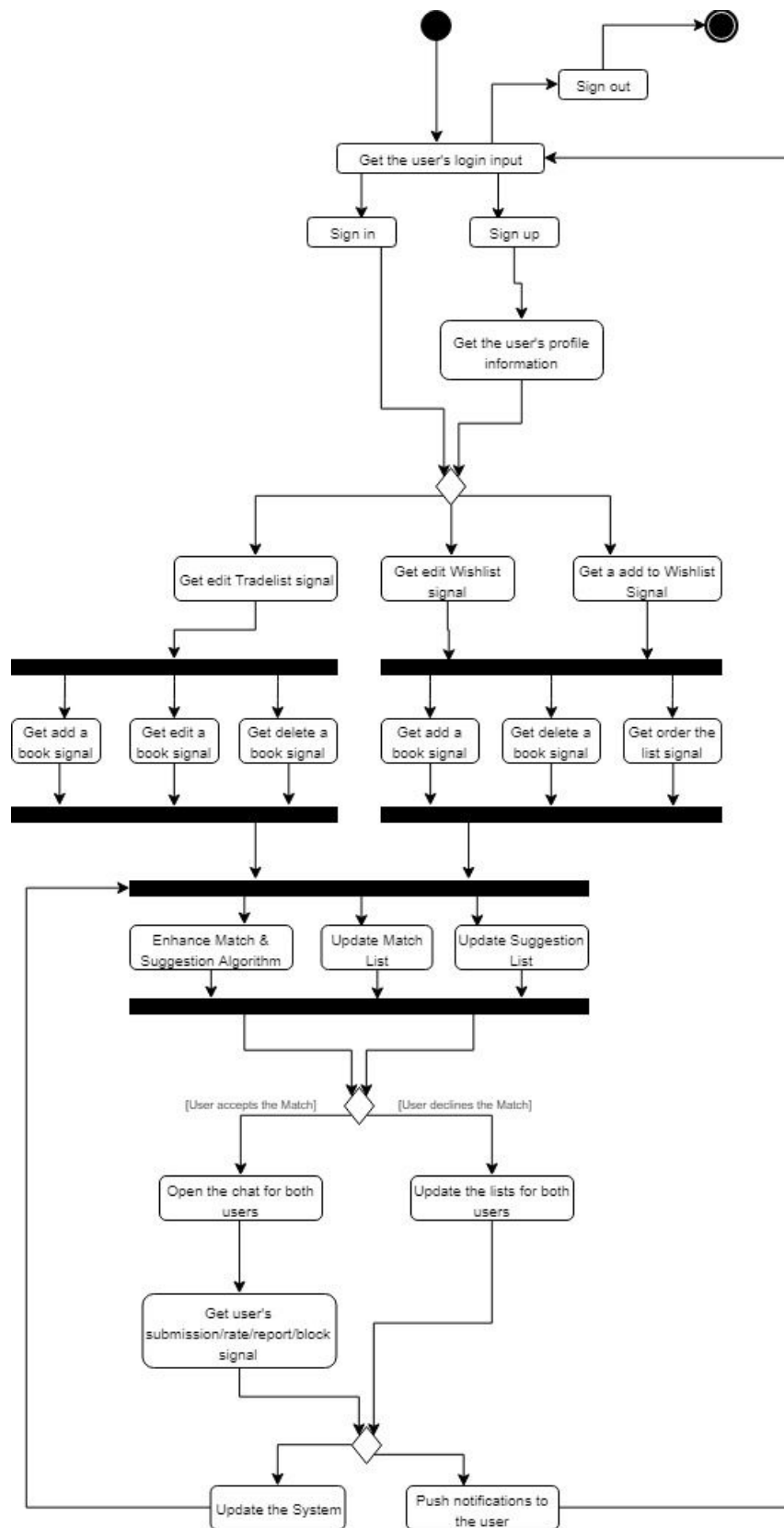
3.5.4.1.4 Confirm Match Sequence Diagram



The sequence diagram above shows the operation of an actor deciding his favorite match. From the ListScreen, the actor presses the “Match” button in

the middle top section of the screen. Then the screen creates a MatchScreen where the actor sees their matches. Newly created MatchScreen calls ListManager to retrieve the MatchList of the user. This MatchList object is passed to the MatchScreen, furthermore, this section gets the list view adaptation of MatchScreen from the object and adapts it to the screen. The first item of the list is shown to the user as a card. The actor does not like the match so he swipes left, which makes the screen readapt the list and show the actor the next match card. The actor likes the match, therefore, swiped the card to the right which opens up a ChatScreen where he can chat with the owner of the matched book.

3.5.4.2 Activity Diagram



In the beginning, the system waits for a user input. At this point, user may start the sign in or sign up. Additionally, the user may sign out if he/she is already logged in the application. When the user signs out, system goes to the final node. If the user signs in or signs up, the application starts its process. However, there is a difference between signing in and up. When the user signs in, the system recognizes the user and does not require any additional information. On the contrary, when the user signs up, the system detects the new user and requires him/her to provide the base information about themselves. After finishing the sign in step, the system allows the user to use the platform freely. Since the platform has more than 20 use case scenarios, for simplicity, the activity diagram above is focused on the user's trading activity.

The user may either edit their tradelist, edit their wishlist or add a book to wishlist without entering the wishlist screen.

First case: the user edits the tradelist. System gets the user's signal about editing the tradelist. Since system does not know what the user is going to do yet, it prepares to the user's actions at the same time. In other words, whatever the user does, the system is going to support the action in the same speed. In this case, the user may either edit the existing book, delete it or add a new book to the tradelist.

Second case: the user edits the wishlist. System gets the user's signal about editing the tradelist. Since system does not know what the user is going to do yet, it prepares to the user's actions simultaneously. In other words, whatever the user does, the system is going to support the action in the same speed. In this case, the user may either edit the order of the existing books, delete it or add a new book to the wishlist.

Third case: the user adds a book to the wishlist without entering the wishlist. In other words, the user may press the heart button located under book instance in the main screen, or after filtering/searching specific books. In this case, the system does the same actions it would do for the second case, since the end point of these cases are the same.

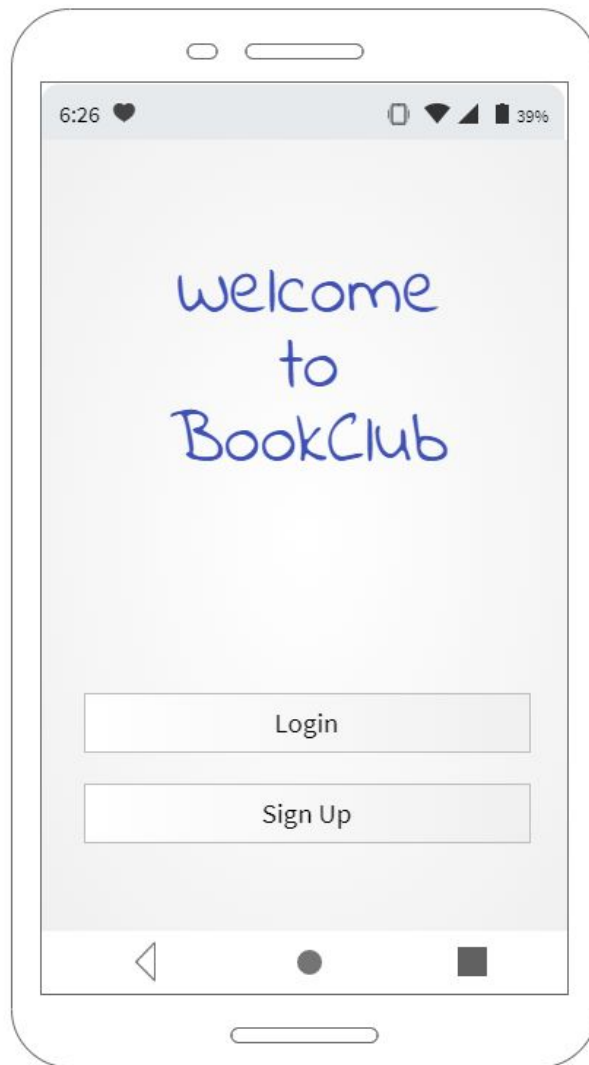
After the system gets the updates on the lists, it proceeds to update itself. In this case, the system simultaneously enhances match and suggestion algorithm, updates the match list for all users (or the users that get affected by the previous actions of the user) and updates the suggestion list for all users. The updates on Match/Suggestion list brings new matches to the user's match list. Thus in this case, the user may either decline or accept the match.

First Case: The user accepts the match. The system gets the signal from the user that the match was accepted. The system opens a chat section for both users that have accepted the same match. While users are chatting, the system waits for one of the following actions: rate, report, block or submit the trade completion. After getting the signal of any actions stated above, the system updates itself and starts from enhancing matching and suggestion algorithm again.

Second Case: The user declines the match. In this case, the system gets the signal that one/both side(s) of the match declined the match case. The system updates both users' match lists, pushes notifications about the refusal and waits for the user's initial actions.

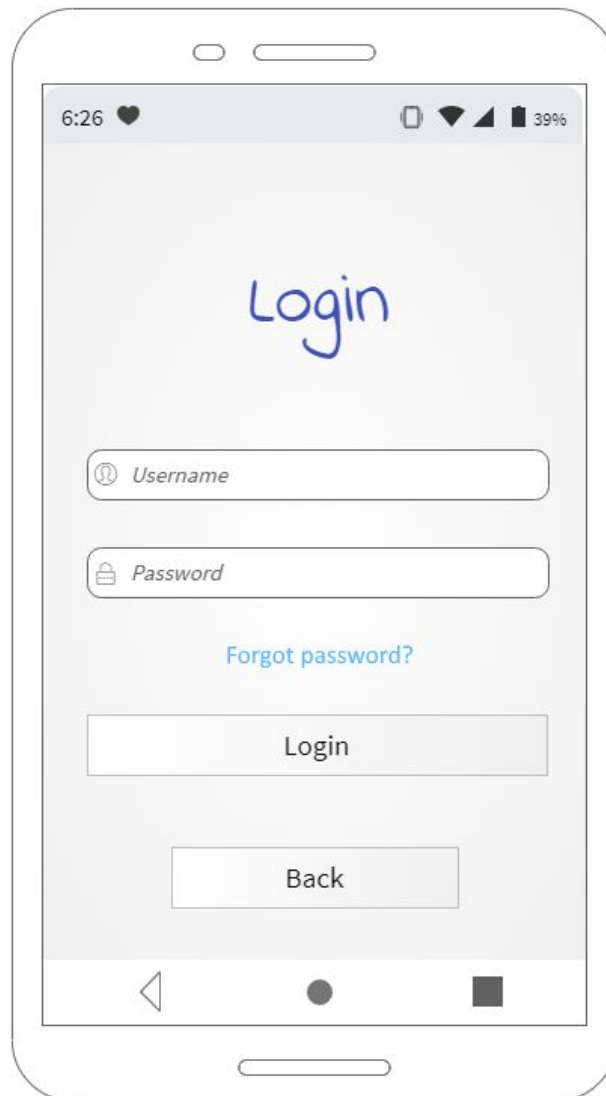
3.5.4.3. User Interface

3.5.4.3.1 Welcome Screen



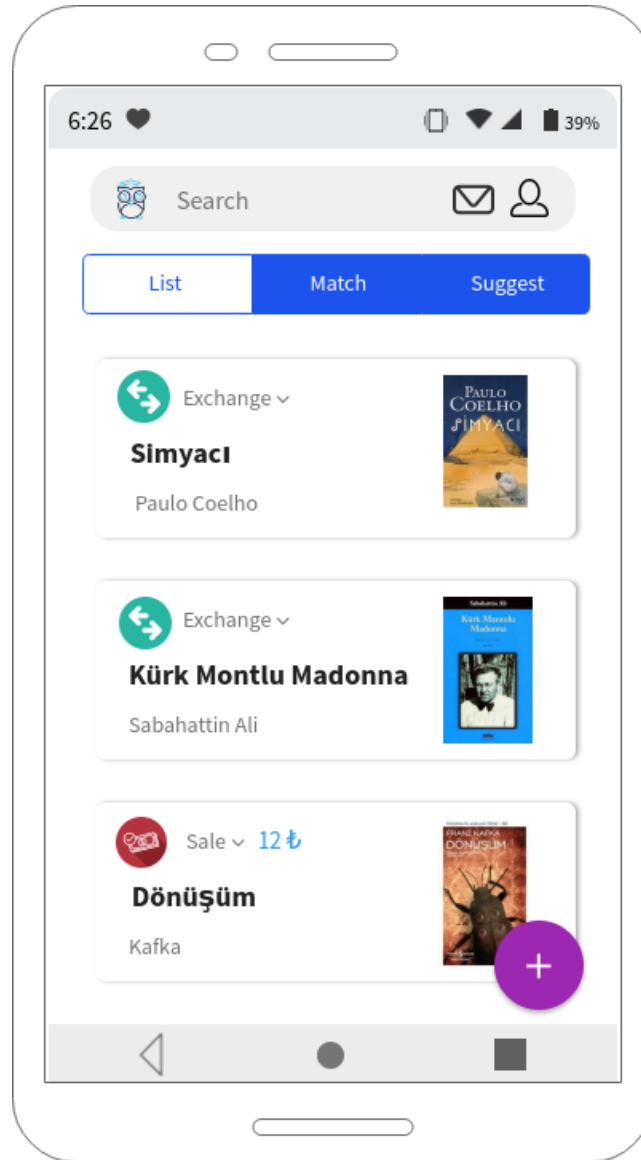
Since our platform requires an account, the very first page will welcome user and allow them to login with their account or sign up the system.

3.5.4.3.2 Login Screen



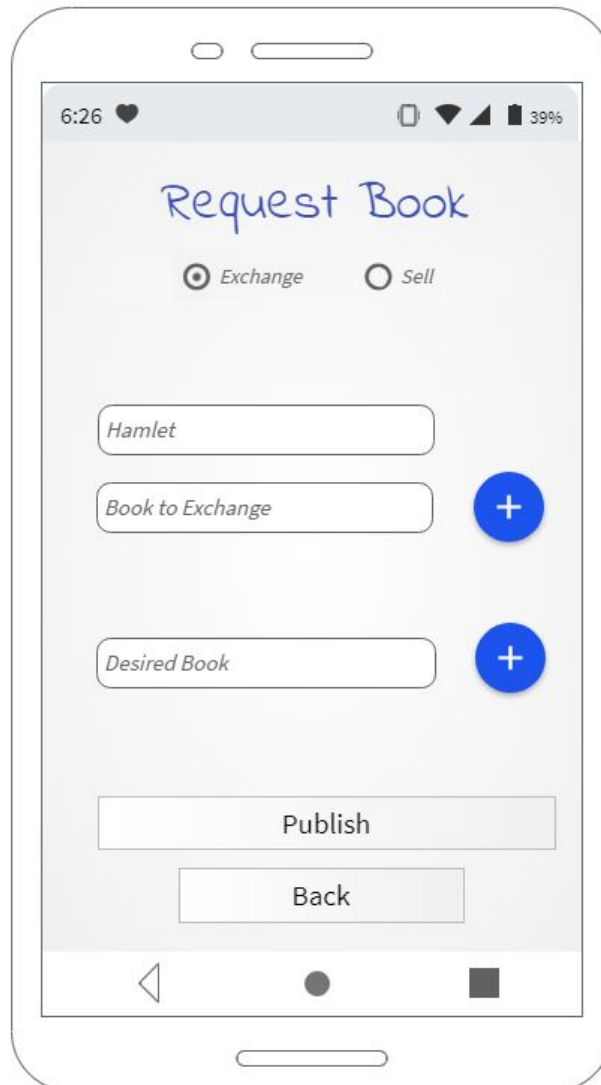
User can login with their username and password. If they have forgotten their password, they can also recover it in this screen.

3.5.4.3.3 Main Screen



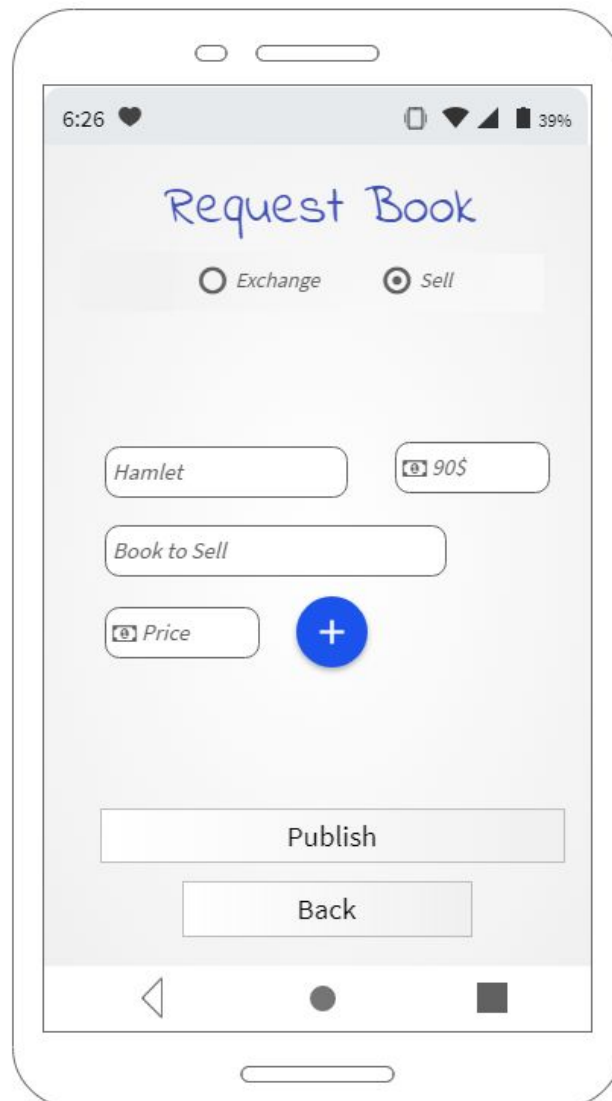
Once the users log in the system successfully, they will see the list of all the exchange or sale inputs that are added to the platform by other users. Users can add their requests (exchange or sale) by pressing the add button. By clicking the search title, they can start to type a keyword to search a specific book. Letter icon on the top right leads to the messages and the profile icon near it takes the user to the preference screen. If the user swipes the screen left, she can go to the match or suggest screens.

3.5.4.3.4 Exchange Book Screen



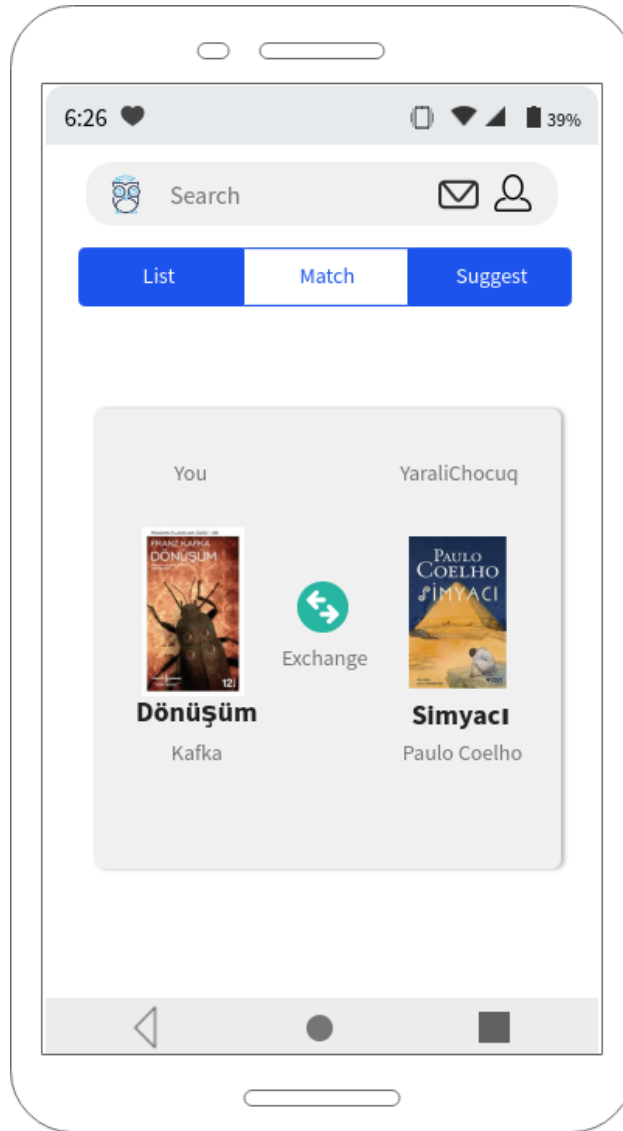
When the add button on the list screen is clicked, a popup window appears. The user can add a book he has by clicking the upper add button. Everytime he adds a book, a new slot to add new one will open. To request a book in exchange, the user can click on the lower add button. Everytime he desires a book, a new slot will open. These steps are taken for the exchange situation. If user wants to sell a book, he needs to select the sell radio button on top.

3.5.4.3.5 Sell a Book Screen



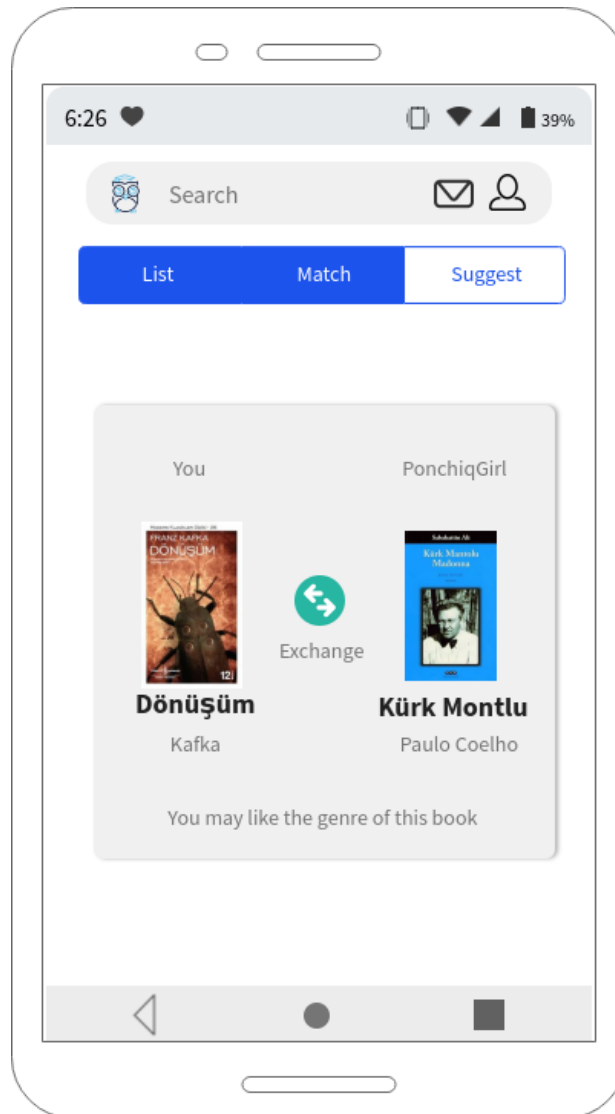
If user selects the sell radio button on the request screen, he can add only the books he owns. However, the price slot appears in this case. User can type a price and publish the request.

3.5.4.3.6 Match List Screen

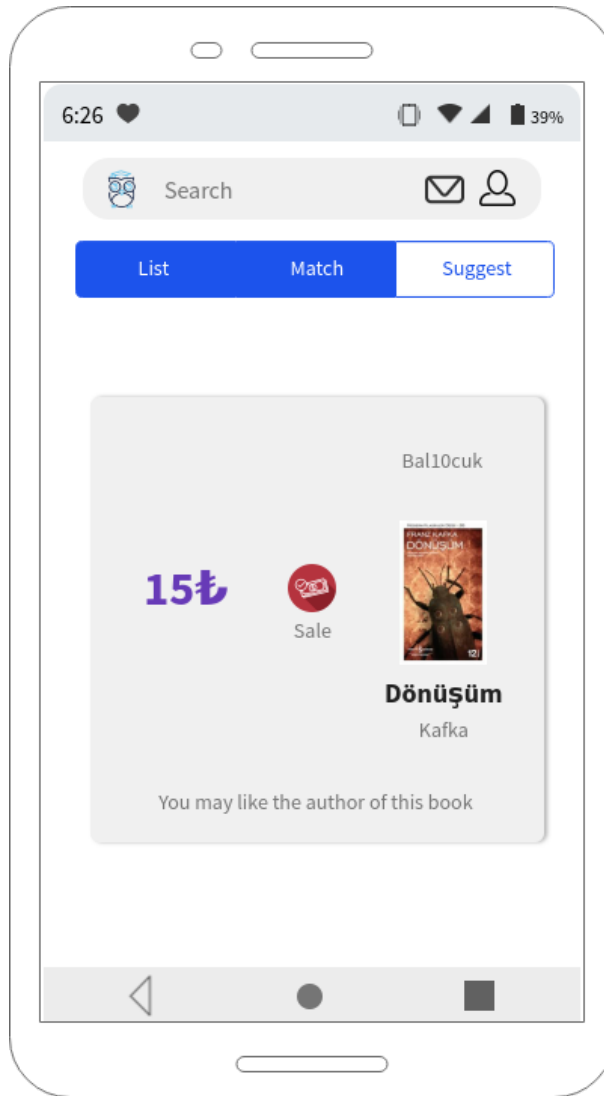


In the match screen, the matches found by the algorithm will appear one by one. If the user swipes the match to the right, it means he accepts the match. Then, the message screen opens. If the user swipes the match left, it means he does not accept the match. Then, a new match card appears.

3.5.4.3.7 Suggestion List Screen

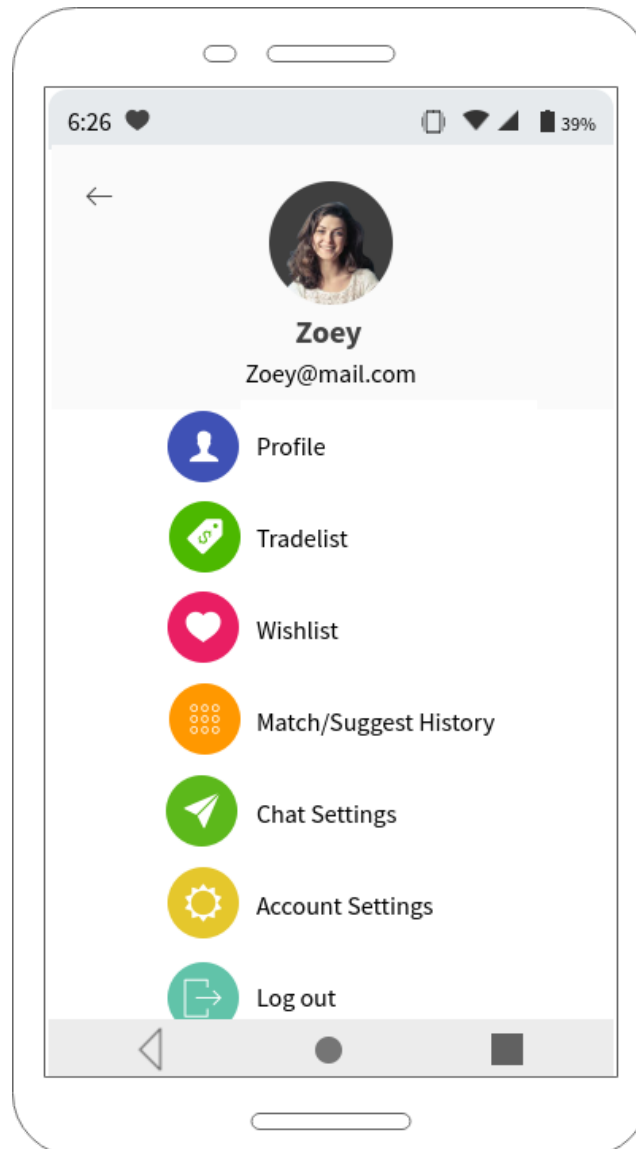


In the suggest screen, the same manner like in the match screen applies. In addition, an information text about why the card is suggested shows up.



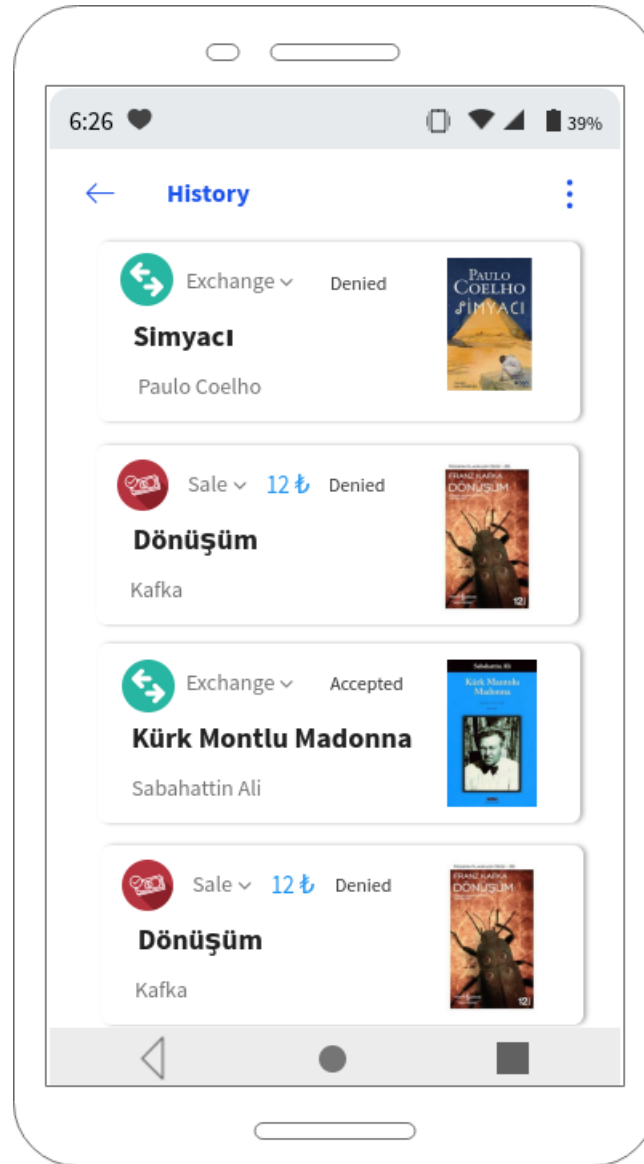
In the suggest screen, both exchange cards and sale cards can appear according to the user's preferences.

3.5.4.3.8 Preferences Screen



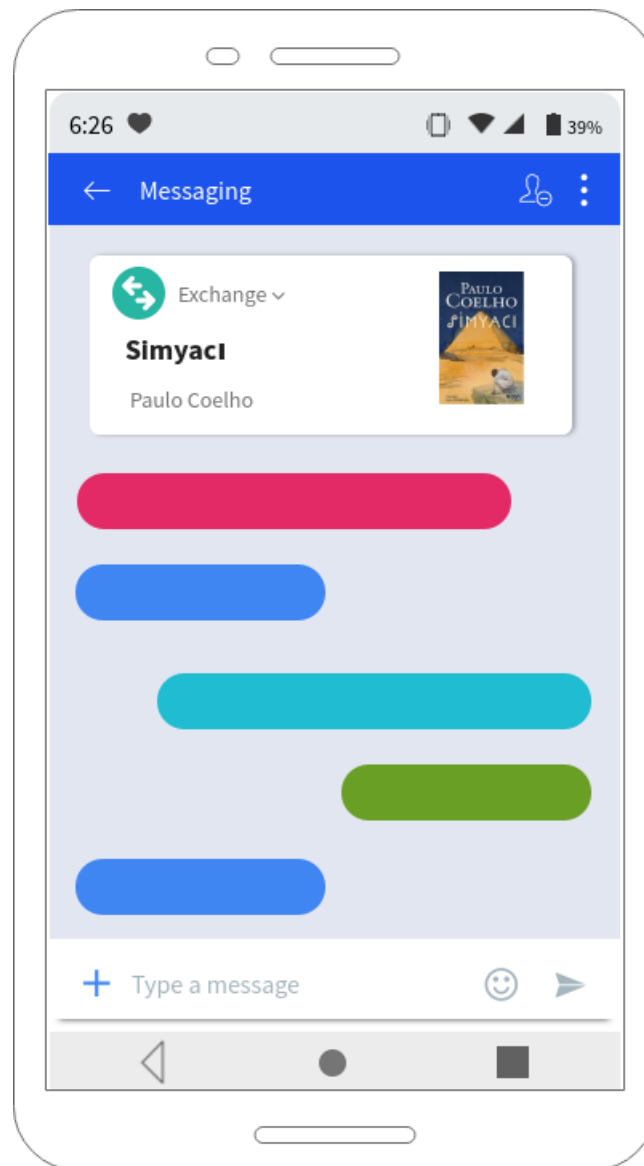
In the preferences screen, the user can see the overview of his account, manage his wishlist and tradelist (requested books, owned books, books in sale), check his match/suggest history that contains the match cards the user accepted or denied, block other users or change his preferences in chat settings, and finally manage his privacy and other options related to his account in the account settings. Log out option unlinks the account from the application, then user needs to sign in again.

3.5.4.3.9 History Screen



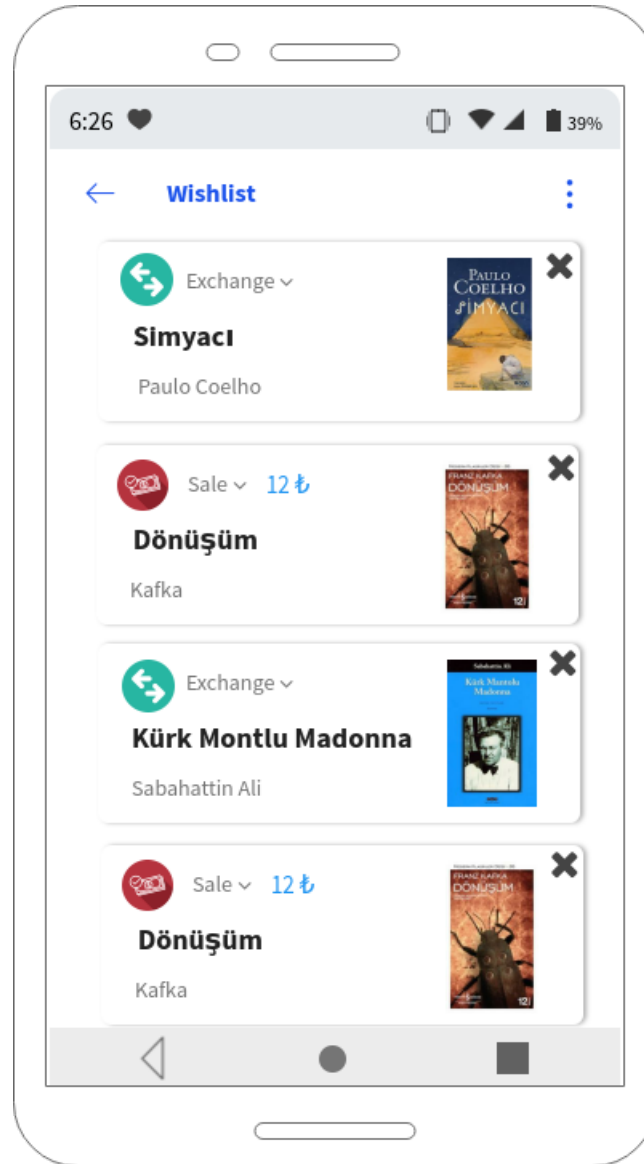
In the history screen, user can see which matches he denied or accepted.

3.5.4.3.10 Message Screen



After user accepts a match or suggestion, the message screen will open. Both users can type to each other. They also can see what book they are talking about.

3.5.4.3.11 Wishlist Screen



In the wishlist screen, the exchanges and the sales which are added from the main list on the main screen will be listed here. User can remove these additions by clicking the x mark.

4. References

- [1] "New & Used Books from ThriftBooks | Buy Cheap Books Online". [Online]. Available: <https://www.thriftbooks.com/>. [Accessed: 13 Oct, 2018].
- [2] "Books at Amazon". [Online]. Available: <https://www.amazon.com/books-used-books-textbooks/b?ie=UTF8&node=283155>. [Accessed: 10 Nov, 2018].
- [3] "GoodReads - Share your book recommendations with your friends". [Online]. Available: <https://www.goodreads.com/>. [Accessed: 8 Nov, 2018].
- [4] "HepsiBurada - En Çok Satan Kitaplar & Kitap Önerileri". [Online]. Available: <https://www.hepsiburada.com/kitaplar-c-2147483645>. [Accessed: 10 Nov, 2018].
- [5] "REST API Tutorial". [Online]. Available: <https://restfulapi.net/>. [Accessed: 10 Nov, 2018].