Requirements and Analysis Document for Boket

Oscar Bennet, Alexander Jyborn, Pajam Khoshnam, Albin Landgren and Tarik Porobic
Grupp 1337
9/30
Version 2

1 Introduction

The project aims to create an application that makes it easier for students to find and sell their used course literature. Students will be able to subscribe to books and get a notification when someone is selling the book. It will be easy to create an ad since the student can easily scan the barcode of the book to auto generate an ad.

1.1 Definitions, acronyms, and abbreviations

Create a word list to avoid confusion and give a definition of every abbreviation you use in the document.

- app = the application
- user = an account in the application
- book = a book object used by the user
- ad = an advertisement published by a user to sell a book
- database = data storage of the user, book and ad models
- firebase = handles the backend and data storage

2 Requirements

2.1 User Stories

DONE!

1. As a user I would like to be able to make an account which I can log out so that I can save my subscriptions and ads to my profile safely.

   Acceptance:

   Sign up screen

   - App can receive input from user (DONE)
   - App can make a user in database once all fields are filled (DONE)
   - App can verify if the input is legit for an account (DONE)

Login screen

- App can receive input from user (DONE)
- App can find a user in database with the information (DONE)
- Only possible to login with an existing user (DONE)

Profile screen

- App can get user's subscriptions and ads (DONE)
- Users subscriptions are displayed (DONE)
- Users ads are displayed (DONE)

2. As a buyer I would like to be able to find and subscribe to a book by searching the book name, ISBN number or scanning the barcode of the book I am looking for.

   Acceptance:

   - Search bar can get inputs by typing book name, ISBN number or by scanning the barcode of a book. (DONE)
   - Search fragment can use the input to search for books in database (DONE)
   - App can show results from the search in a list of books (DONE)
   - A book in the list should be able to recognize when it is chosen by a click (DONE)
   - Show a new fragment with the chosen book where a subscribe button will be (DONE)
   - Subscribe the user to the book when subscribe button is pressed (DONE)

DONE!

3. As a seller I would like it if the book information could fill automatically when I want to create an ad by scanning the barcode of the book. So that the process could go faster and more efficiently.

   Acceptance:

   - Have a camera button which opens the camera (DONE)
   - Camera can read a barcode and communicate with the database (DONE)
   - Read the relevant information from database (DONE)
   - Write the information in right place in the ad (DONE)

DONE!

4. As a buyer I want to be able to contact the seller so that we could negotiate a deal.

   Acceptance:

- A button should appear when a seller is chosen (DONE)
- The button should open the local mail app on the phone (DONE)
- In the mail app the mail of the seller, subject and a pre text about the book should be auto generated. (DONE)

PARTIALLY DONE!

5. As a seller I want to be able to manage my book ads

   Acceptance:

   - View all book ads (DONE)
   - Edit information (Not implemented)
   - Archive ad (DONE)

DONE!

6. As a buyer I want to be able to manage my subscriptions

   Acceptance:

   - View all subscriptions (DONE
   - Be able to unsubscribe (DONE)

DONE!

7. As a buyer I want to see all the sellers for the book so that I can find the best option for me.

   Acceptance:

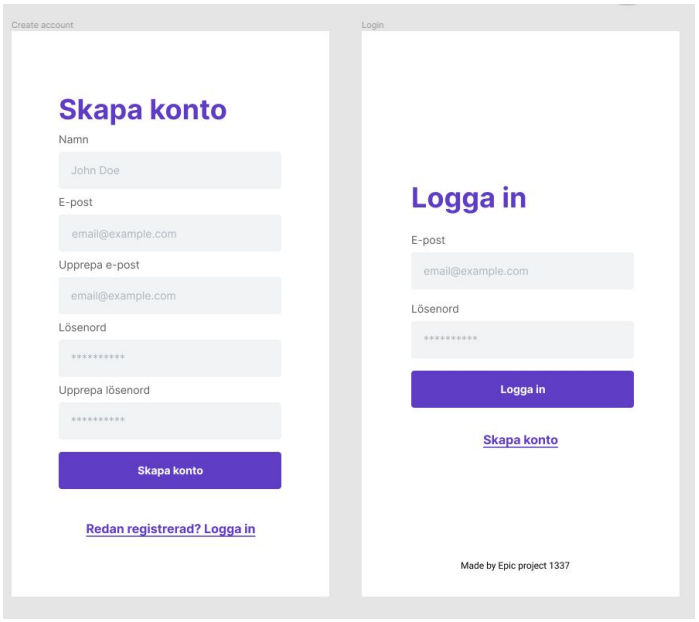
   - When a book is chosen a new fragment should be shown (DONE)
   - The new fragment will showcase the book and a little information about it (DONE)
   - The new fragment will have a list of all sellers who sell the chosen book (DONE)
   - App will be able to get all the sellers from the database and represent them in the list (DONE)

2.2 Definition of Done

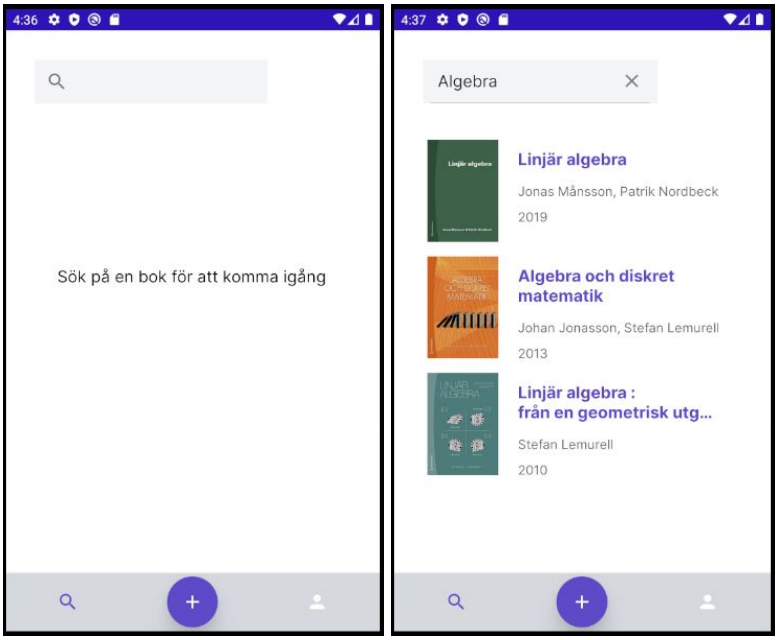
New functions must be tested before labeled “Done” and it should all be under version control. All public functions and interfaces must be javadoc commented and all classes must have a javadoc header.

2.3 User interface

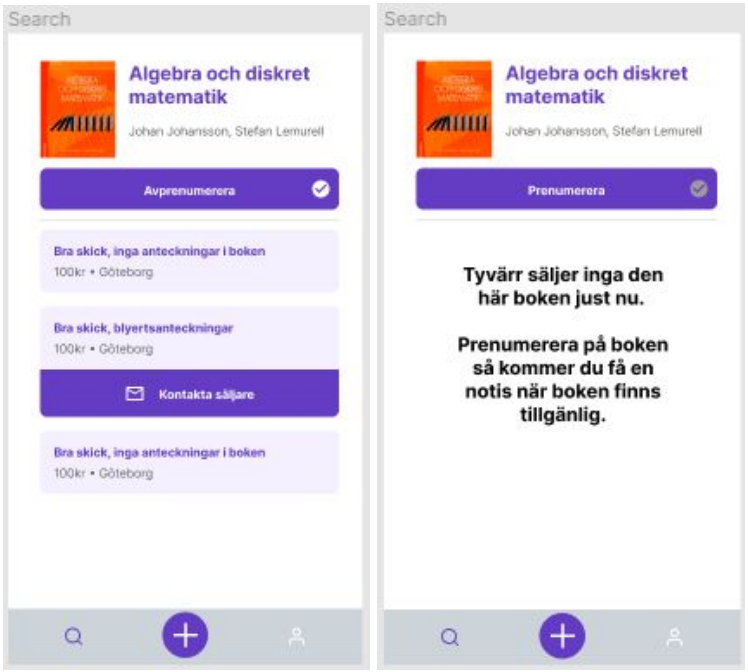
Include sketches, drawings and explanations of the application's user interface. Describe the navigation between the different views.



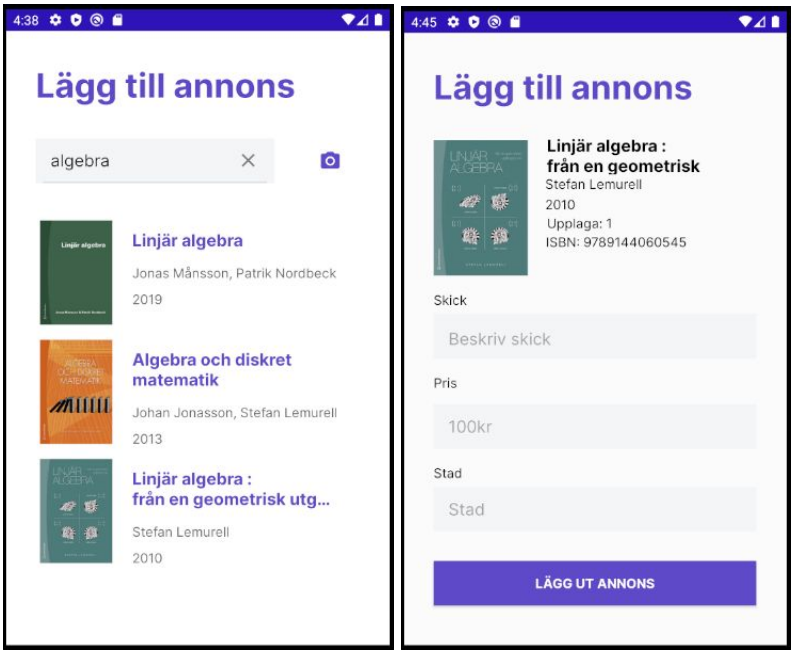
Signup and login page. Both the "Skapa konto"-button and the "Logga in"-button navigate the user to the search page.



Start search page and search page with results. By pressing on a book the app navigates to the page which shows the ads of the book.



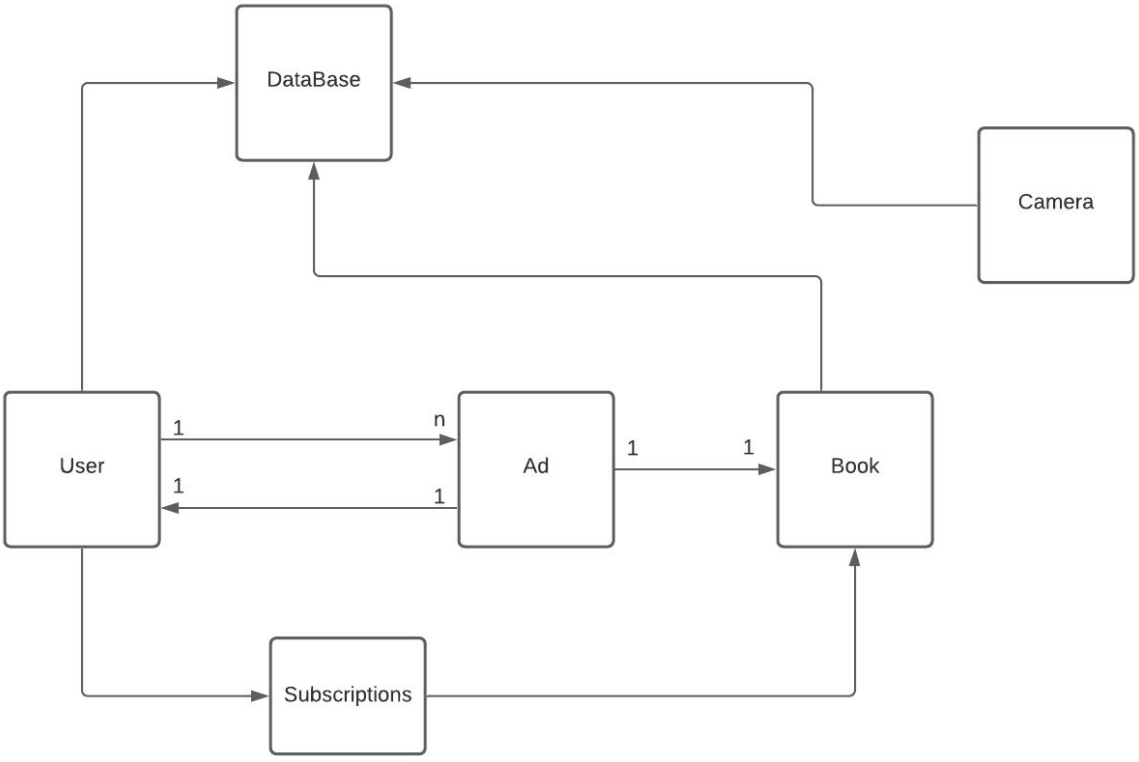
Selected book page[i] with sellers and with no sellers.



Create an ad by first searching for the book and then entering the details. By pressing on a book the app navigates to the page where the ad is created.

Profile page. If you press on a book which you are subscribed to you to go to page [i] and can see all the sellers for the book.

3 Domain model



3.1 Class responsibilities

- User: User holds the information of an account which it gets from the database such as email, name, location.
- Subscriptions: Connects the user with the books the user is subscribed to. So the user can be notified when an ad for that book is published.
- Ad: Ad holds a reference to the book which is being sold and to the user who published the ad.
- Book: Book holds the information from the database such as isbn, title, author, edition, release date and an image of the book.
- Camera: The camera uses an external library provided by firebase-mlkit, to scan the barcode.
- Database: Firebase database that holds all the data needed for the application.

4 References

- Firebase (https://firebase.google.com)
- Glide (https://github.com/bumptech/glide)
- Algolia (https://www.algolia.com/)