Requirements and Analysis document for AWME bokbytarapp

Ali Alladin, Magnus Andersson, William Hugo, Elias Johansson

2020-10-02

0.1

1 Introduction

Give some background and explain the purpose of this application. Describe the functionality of the application. Describe the stakeholders of the project, highlight who will benefit from/use this particular application.

The buying and selling of course literature between students at Chalmers University is today mainly done through different facebook groups. The plan is to build an android application on which you can buy or sell your course books. The application will allow sellers to post their books for free and buyers to search for their desired books and contact sellers if they find something desirable.

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.

2 Requirements

2.1 User Stories

Use the template from the course website and list all user stories here. It is fine to have them in an spreadsheet (or other applications, such as Trello) at first, but they must end up here as well.
These user stories should describe what the user will be able to do. Write the user stories in language of the customer, and give them a unique ID. List the user stories in order of priority.
You need to annotate an user story whether or not it is implemented. We need to know which user stories are implemented, such that we can check this during the oral presentation.

User story - Ongoing

Story Identifier: G001
Story Name: Buyer Search
Priority: High

Description

As a buyer, **I want to** be able to search for books by title **so that** I can find the books I am looking for.

Acceptance criteria

- Functional:
    - Can I search by title?
    - Can I interact with the search result?
    - Am I informed if my search returns no results?
- Non-functional:
    - Availability:
        - Can I search anytime of the day?

User story - Ongoing

Story Identifier: G002
Story Name: Seller Post
Priority: High

Description

As a seller, **I want to** post a listing for my book **so that** other users can buy it

Acceptance criteria

- Functional:
    - Can I post my book?
    - Can I edit my listing after the fact?
    - Can I remove my listing?
    - Can I add a topic to my post?

- Non-functional:
    - Security:
        - Are unauthorised persons prevented from editing my listings?

User story - Done

Story Identifier: G003
Story Name: See Listings
Priority: High

Description

As a user, **I want to** be able to see a infinity list with listings, be able to scroll through it and click a card of interest to see more **so that** the app is easy to get started with

Acceptance criteria

- Functional:
    - Can I see a list?
    - Can I scroll the list?
    - Can I click a card in the list to see more about it?

User story

Story Identifier: Y001
Story Name: Buyer-Seller Contact
Priority: Low

Description

As a buyer, **I want to** be able to contact sellers **so that** I can buy the book

Acceptance criteria

- Functional:
    - Can I message the seller?

- Non-functional:
  - Availability:
    - Will my message reach a seller once they log in if they are not currently logged in?

User story - Ongoing

Story Identifier: Y002
Story Name: Sort
Priority: Low

Description

As a user, **I want to** be able to sort books by subject/course **so that** I can find relevant listings

Acceptance criteria

- Functional:
  - Can I sort by price?
  - Can I sort by topic?
  - Can I sort by course?
  - Can I sort by date?

User story - Done

Story Identifier: Y003
Story Name: Selling List
Priority: Low

Description

As a seller, **I want to** be able to access a list of books that I'm selling **so that** I have an easy way to view all my listings

Acceptance criteria

- Functional:
  - Can I browse all of my listings?

User story

Story Identifier: Y004
Story Name: Selling notebooks
Priority: Low

Description

As a seller, **I want to** be able to post my notebooks from previous courses in the app **so that** I can sell them

Acceptance criteria

- Functional:
    - Can I post a notebook listing?
    - Can I specify which course the notebook was written for?

User story - Done

Story Identifier: O001
Story Name: Saved-list
Priority: Low

Description

As a buyer, **I want to** be able to save books **so that** they're easier to find in the future.

Acceptance criteria

- Functional:
    - Can I add a listing to my saved-list?
    - Can I access my saved-list easily?
    - Can I delete listings from my saved-list?

User story - Done

Story identifier: G004
Story name: Login
Priority: Low

Description

As a user, **I want to** be able to log in to my account **so that** I can use the app securely

Acceptance criteria

- Functional:
    - Can I log in?
    - Can I log out?
- Non-functional:

User story

Story identifier: R002
Story name: Seller Review
Priority: Low

Description

As a user, **I want to** be able to read reviews of a seller **so that** I know which seller to choose.

Acceptance criteria

- Functional:
    - Can I access a seller's reviews?
    - Can I review the seller?

User story

Story Identifier: R003
Story Name: Extended Buyer Search
Priority: Low

Description

As a buyer, **I want to** be able to search for books by things other than titles, such as ISBN **so that** I can find the books I am searching for.

Acceptance criteria

- Functional:
    - Can I search by ISBN?
    - Can I interact with the search result?
    - Am I informed if my search returns no results?
- Non-functional:
    - Availability:
        - Can I search anytime of the day?

User story

Story identifier: R001
Story name: Google-Login
Priority: Low

Description

As a user, **I want to** be able to log in to my account with Google **so that** I don't have to use several password and use the app quickly

Acceptance criteria

- Functional:
    - Is the login handled by google?
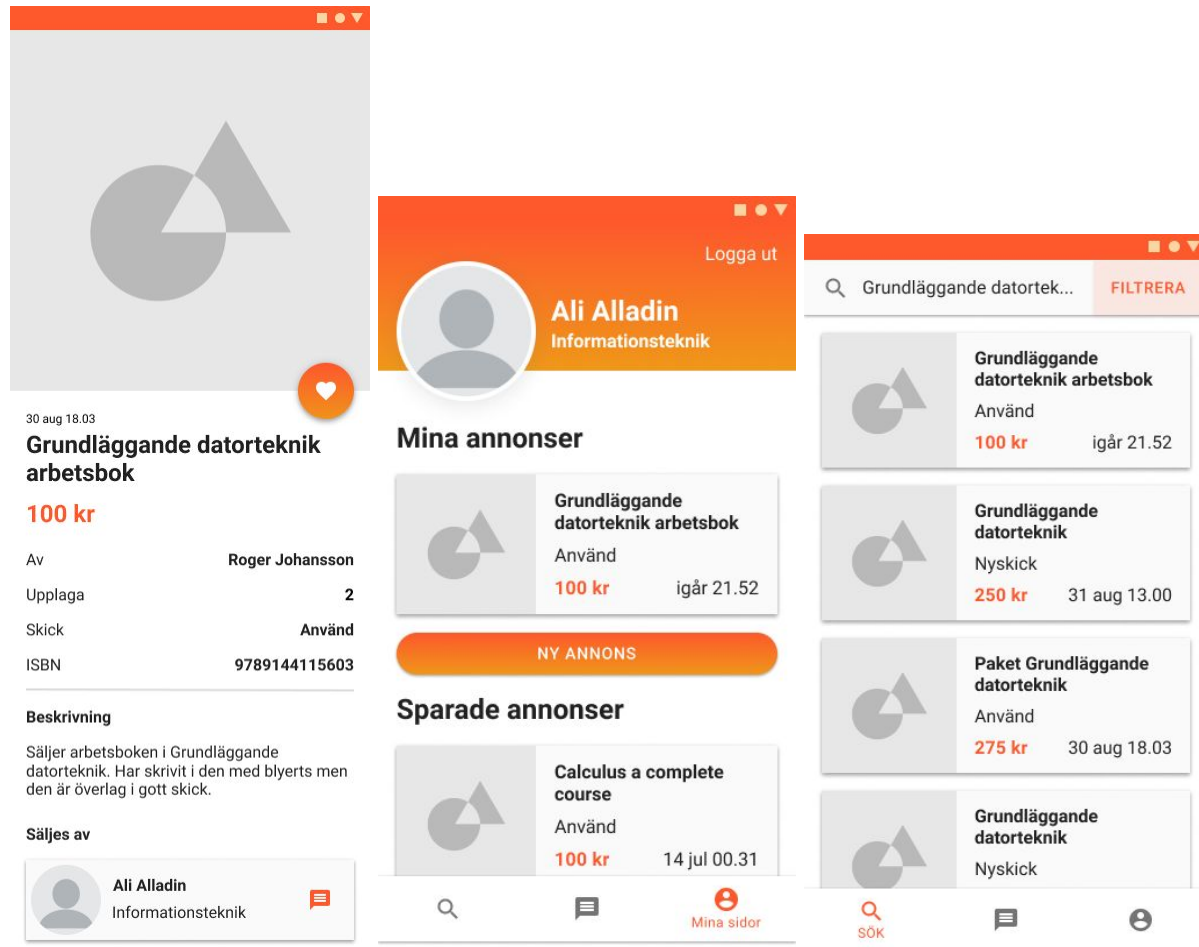   - Can I avoid using Google to sign in?

2.2 Definition of Done

In this section you list the acceptance criteria that are common for all user stories. For example, the code should reviewed and tests, it should be under version control, etc.

All user stories need to be tested, reviewed and be under version control to be able to be completed.

2.3 User interface

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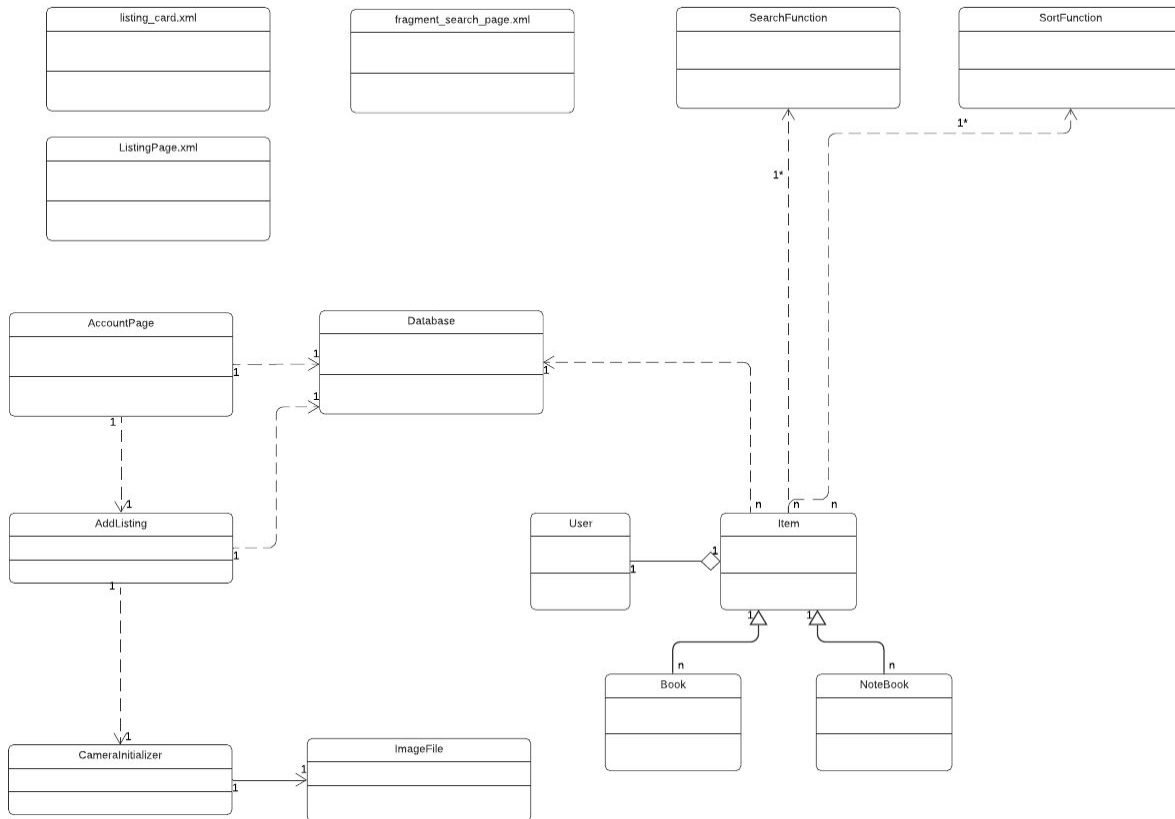

The user interface is inspired by existing marketplace applications which will make it easy for new users to adapt to it. It mainly consists of three pages that can be navigated in-between by a bottom-navigation bar. The first page is the search-page where all listings are shown and the user can choose to search for specific items and filter and sort the result. Each listing is represented by a card that contains the most essential information. The card opens up the detail page for each listing (if pressed) where all the information about the listing is available. The second page is the messages-page where the user has all the conversations regarding different listings with other users. The third and final page is the Account-page where the users can easily add new listings, access all of their own listings and see their "wish-list"/saved listings. The user-interface of the applications is overall quite simple and easy to use. Only a couple of colors are used throughout the application to keep it clean but still aesthetically appealing.

3 Domain model

3.1 Class responsibilities

Explanation of responsibilities of classes in diagram

AccountPage

AddListing

AddListing handles creation and submission of a new listing to the database

Book

Book is a subclass and realization of Item with unique behaviour.

CameraInitilizer

CameraInitilizer uses the ImageFile class to set up a temporary file that is then returned to the AddListing class. It also creates and returns a Uri for it.

Database

Database handles all communication with the Firebase database and has various methods for adding new content.

ImageFile

ImageFile represents the file that the phone-camera will store the captured image in. In the class this file is created with a unique name to avoid conflicts.

Item

Item is an abstract class that all listings in the database inherits from.

NoteBook

Notebook is a subclass and realization of Item with unique behaviour.

SearchFunction

SearchFunction handles a query in the form of a CharSequence from the user to match that sequence with the title of the Item in the database and then returns a List of items that match the query.

SearchPage

SortFunction

SortFunction sorts all items in the database by different parameters such as price or date added.

User

User represents the user of the application.

3 References

List all references to external tools, platforms, libraries, papers, etc.

- [Android Studio](#)
- [Firebase](#)