

Requirements and Analysis Document for Chalmers Book Market

An application encouraging reuse of
student literature

**Pegah Amanzadeh, Simon Holst, Kevin
Pham, Carl-Magnus Wall**

Group X

TDA367 Object-oriented programming project

Software Engineering
Chalmers University of Technology
October 2021

Contents

1	Introduction	2
1.1	Definitions, acronyms, and abbreviations	2
2	Requirements	3
2.1	User Stories	3
2.2	Definition of Done	14
2.3	User interface	14
3	Domain model	20
3.1	Class responsibilities	20
3.1.1	ApplicationModel	20
3.1.2	User	20
3.1.3	Listing	21
3.1.4	Book	21
3.1.5	Notification	21
3.1.6	Empty classes: EmptyUser, EmptyBook and EmptyList- ing	21
4	References	22
4.1	IDE	22
4.2	Build Tools	22
4.2.1	Imported libraries	22
4.2.2	Structure analysis	22
4.2.3	Quality analysis	22
4.3	Design Tools	22

1 Introduction

The application's purpose is to benefit all Chalmers students, creating an universal platform where previously owned student literature can be reacquired for second hand use.

Potential stakeholders include, but are not limited to, Chalmers University of Technology, which might be interested in creating a more sustainable market of student literature with an increased life span.

The target group of the product will be the students at Chalmers University of Technology. Although this application may not be relevant for an extended list of schools yet, this document aims to serve as a foundation, discussing and analysing requirements that could prove useful for extended models.

The main functionality of the application is to be able to purchase and sell used course literature. The application is targeted to a mobile audience.

1.1 Definitions, acronyms, and abbreviations

- CBM - Chalmers Book Market
- CID - Chalmers ID

2 Requirements

The application seeks to work as a prototype of a mobile book market, tailored specifically for students at Chalmers. This means the application should at the very least showcase the idea of logging in with a CID, being able to search for different types of course literature, being able to put a book up for sale, being able to subscribe to certain literature and being able to reserve (i.e. buy) books. Ultimately, the software is a proof-of-concept more than anything else.

To realize the application, the following user stories and criteria are considered.

2.1 User Stories

1. Story Identifier: CBM001 - IMPLEMENTED
Story Name: Main Feature - Ability to sell

Description:

As a Chalmers student I would like to be able to easily sell my used literature, in order to make some extra money.

Confirmation:

- Are all elements pertaining to selling implemented?

Functional:

- Can I freely pick a price point?
- Can I add an indication of book condition?
- Can I edit or remove a listing?

Availability:

- Can I list a book at any time of the day?
- Can I access my account page at any time of the day?
- Will notifications regarding my listings be sent immediately?

Security:

- Am I the only who can see whom I have sold a book to?

1.2. Story Identifier: CBM012 - IMPLEMENTED

Story Name: Listing Details

Description:

As seller I want to be able to upload a description and price to increase the odds of getting consumers.

Confirmation:

- Is it possible to add a listing in the application?

Functional:

- Can I adjust the price freely, in order to attract different customers?
- Can I freely add a proper description of the book I am trying to sell?
- Can I adjust these after publishing the listing?

Availability:

- Can I edit my listing at any time of the day?

Security:

- Am I the only who can edit my listings?

1.3. Story Identifier: CBM013 - NOT IMPLEMENTED

Story Name: Listing picture

Description:

As seller I want to be able to upload a picture of the book, in order to prove the condition of the book.

Confirmation:

- Are picture uploads implemented?

Functional:

- Can I upload a picture while adding a listing?
- Can I upload in JPEG or PNG?
- Can remove sale notifications?

Availability:

- Is it possible to upload a picture at any time of the day.

Security:

- Are my uploads safe?

1.4. Story Identifier: CBM014 - NOT FULLY IMPLEMENTED

Story Name: Sale notification

Description:

As user I want to receive a notification whenever there has been an update regarding book reservations, book subscriptions, or if I have made a sale.

Confirmation:

- Are all different notifications implemented, and are they pushed accordingly?

Functional:

- Can I see when somebody has bought or reserved one of my books?
- Can I reach my sale notifications from the home screen?
- Can remove sale notifications?

Availability:

- Will notifications be sent immediately when sales or reservations are made?

Security:

- Am I the only who can see my notifications?

2. Story Identifier: CBM002 - IMPLEMENTED

Story Name: Main Feature - Ability to buy

Description:

As a Chalmers student I would like to be able to easily purchase used course literature from other Chalmers students, in order to save some money.

Confirmation:

- Are all elements pertaining to purchasing implemented?

Functional:

- Can I reserve literature and make a purchase later?
- Can I remove a reservation?
- Can I abort a purchase?

Availability:

- Can I list a book at any time of the day?
- Can I access my account page at any time of the day?
- Will notifications regarding my listings be sent immediately?

Security:

- Am I the only who can see who I have sold a book to?

2.2. Story Identifier: CBM022 - IMPLEMENTED

Story Name: Book reservation

Description:

As a buyer I want to be able to reserve a book, in order for me to make a purchase at a later point in time.

Confirmation:

- Is it possible to reserve a book?

Functional:

- Can I reserve literature and make a purchase later?
- Can I remove a reservation?
- Can I see all my current reservations?

Availability:

- Can I reserve a book at any time of the day?
- Can I see my reservations at any time of the day?

Security:

- Am I the only who can see which books I have reserved?

2.3. Story Identifier: CBM023 - NOT FULLY IMPLEMENTED

Story Name: Seller rating

Description:

As a buyer I want to be able to see the seller's rating, in order for me to know whether the seller is credible or not.

Confirmation:

- Is it possible to see a seller's rating when looking at a listing?
- Can I rate a seller after making a purchase?

Functional:

- Can I rate the sellers I buy from?
- Can I clearly see how highly or lowly rated a particular seller is?

Availability:

- Are the ratings updated immediately after a seller has received a rating?
- Are the seller ratings available at any time of the day?

Security:

- Can others see what ratings I have given?
- Can I see what ratings buyers have given me?

2.4. Story Identifier: CBM024 - NOT FULLY IMPLEMENTED

Story Name: Purchase confirmation

Description:

As buyer I want to be able to get a confirmation of purchase to ensure the purchase was successful.

Confirmation:

- Is the notification system implemented?

Functional:

- Do I get a confirmation when a purchase has been completed?

Availability:

- Do I get a notification as soon as a purchase has been completed?

Security:

- Can others see that I've completed a purchase?

2.5. Story Identifier: CBM025 - IMPLEMENTED

Story Name: Book search

Description:

As buyer I want to be able to search for books in order to find exactly what I seek.

Confirmation:

- Is it possible to search for books? Do I get a list of results that I can interact with?
- Can I search for author, book code or category?

Functional:

- Can I search for books?
- Can I search by book name?

Availability:

- Can I search for books at any time of the day?

Security:

- Can others see my search history?

2.6. Story Identifier: CBM026 - IMPLEMENTED

Story Name: Listing details 2

Description:

As buyer I want to see a book description in a listing in order to confirm that I have found the correct book.

Confirmation:

- Does all listings have descriptions?

Functional:

- Can I see a description of the book I am looking to buy?

Availability:

- Is a book description required?
- Can I always read the descriptions?

Security:

- N/A

3. Story Identifier: CBM003 - IMPLEMENTED

Story name: Personal account

Description:

As student I want to be able to have a personal account so I can have personal functions such as being able to subscribe to books, see my purchase history etc.

Confirmation:

- Are all elements pertaining to the personal account implemented?

Functionality:

- Can I log in with my CID?
- Can I see a list of my reserved books?
- Can I see a list my published books?
- Can I see a list my previous purchases?
- Can I see a list my subscribed books?
- Can I see my E-mail address?
- Can I see my rating?

Availability:

- Can I access my account at any time of the day?
- Can I edit my different lists at any time of the day?

Security:

- Can others see my account details?
- Can others see my reservations, purchases and subscriptions?

2.2 Definition of Done

For each user story, certain criteria regarding functionality will be set up. Each task adapted for a certain user story is directly connected to logic and methods in the code. To assert that a task is done, proper testing of the methods and the logic behind it will be required. Testing will be performed using unit testing with JUnit. Each piece of logic, and each method will be put through a series of tests to assert that they are performing as expected. Once all tasks related to a user story pass testing, then the tasks will be reviewed by all members of the group. If nobody has any objections, the tasks and the corresponding user story will be considered as done.

2.3 User interface

In the sketch, which commenced the development, a global navigation towards the bottom of the screen was established, as shown in Figure 1.

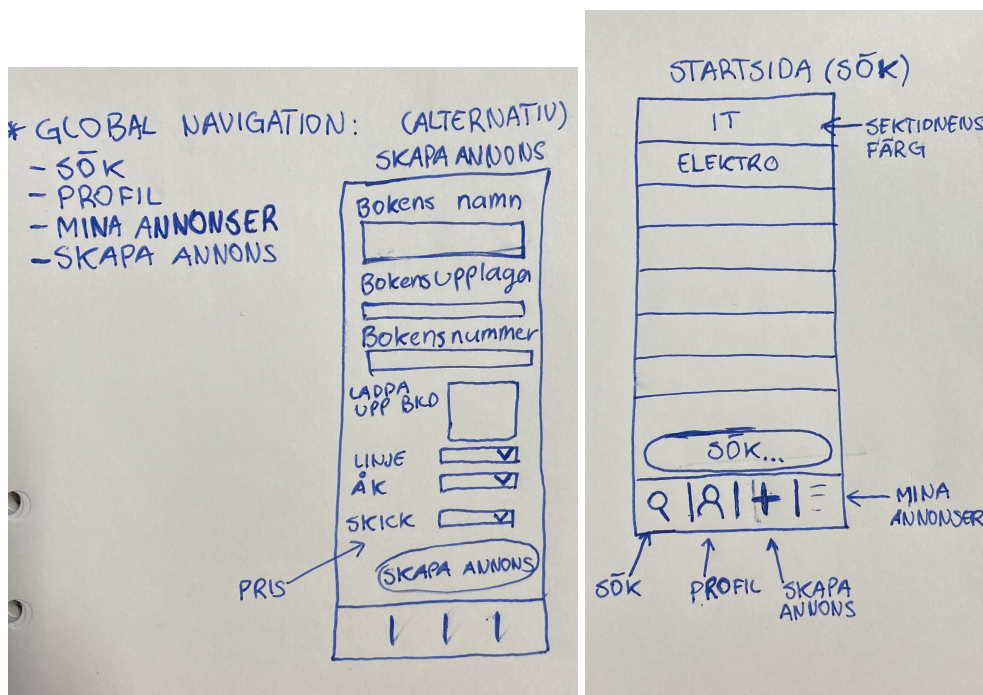


Figure 1: Initial sketch of the application's user interface. Left: Page destined to create new listings. Right: Categorized books under different school institutions.

Asserting an always-on-display navigational option offers an escape hatch

to the user as well as the convenience of reducing navigational excise.

When the paper-sketch felt thought through and roughly finished the design process was continued in the digital design tool Figma. Several concepts were created to diversify the design in order to explore all conceivable possibilities, as shown in Figure 2.

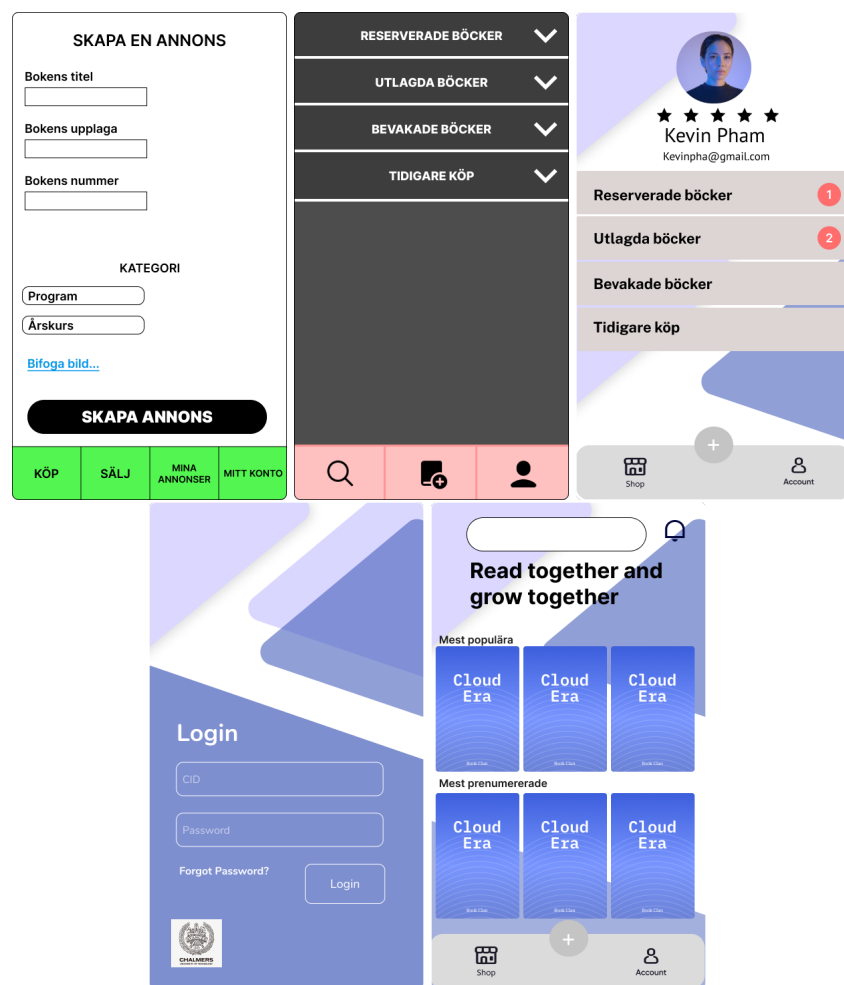


Figure 2: Multiple interpretations of the application's user interface.

After the majority of our design model was realized in actual code the application's graphical user interface was finalized and polished.

Upon entering the application a login screen greets the user, encouraging them to enter their CID as well as their password, see Figure 3.

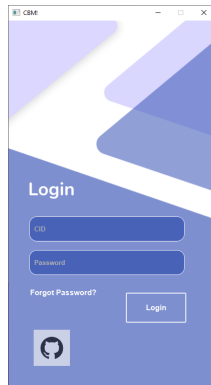


Figure 3: Application's login page.

Subsequently, the user accesses the shopping page; the view offers a wide variety of categorized books to the customer, see Figure 4.

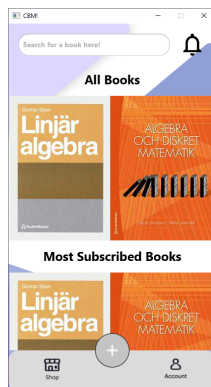


Figure 4: Application's shop page.

From this page the user can browse books by categories or searching for a specific book. When a compelling book is found the customer can click on its cover to enter the detail view, see Figure 5.

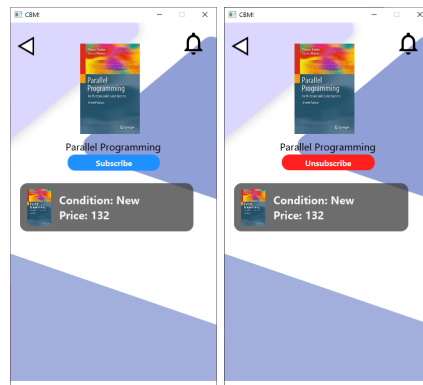


Figure 5: Application's detailed view of a book. Left: Subscribed to a book. Right: Unsubscribed to a book.

Multiple possibilities become available to the user; it can either subscribe to the book to receive notifications when new listings are published with the desired book, or view currently published listings in the form of a list at the bottom. If any published listing seems lucrative the user is able to click the listing in order to bring up its individual page, see Figure 6.



Figure 6: Application's detailed view of a listing.

Detailed information is amassed here, including a description where the seller can specify deviations regarding the book interesting to the buyer. If the user decides to reserve the book, by clicking the *Reserve*-button, the listing disappears from the shop page and is only displayed in the account page, see Figure 7.

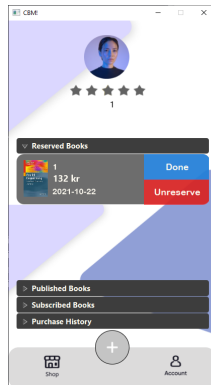


Figure 7: Application's account page, currently displaying the *Reserved Books* section.

From here the user is able to unreserve the listing if the reservation was accidental. Otherwise the customer can contact the seller through their Chalmers mail to setup a meeting on campus where the trade can be completed. When both parts feel satisfied with the exchange they press the *Done*-button to confirm that the trade is successfully complete. Consequently the trade is moved to the section *Purchase History*, see Figure 8.

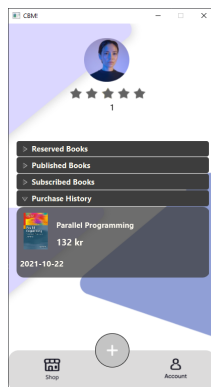


Figure 8: *Purchase History*, displaying the recently made purchase.

Additionally, published and subscribed books can be view here, see Figure 9.

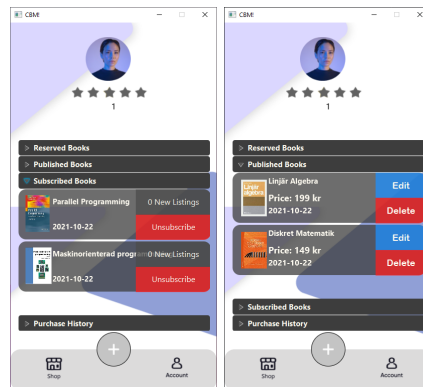


Figure 9: Left: *Subscribed Books*. Right: *Published Books*.

In order to publish a book one must navigate through the global navigation to the sell page, see Figure 10.

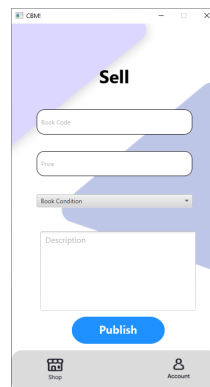


Figure 10: Application's sell page.

All corporated, we have a complete graphical interface capable to assist the user in acquiring their desired course literature.

3 Domain model

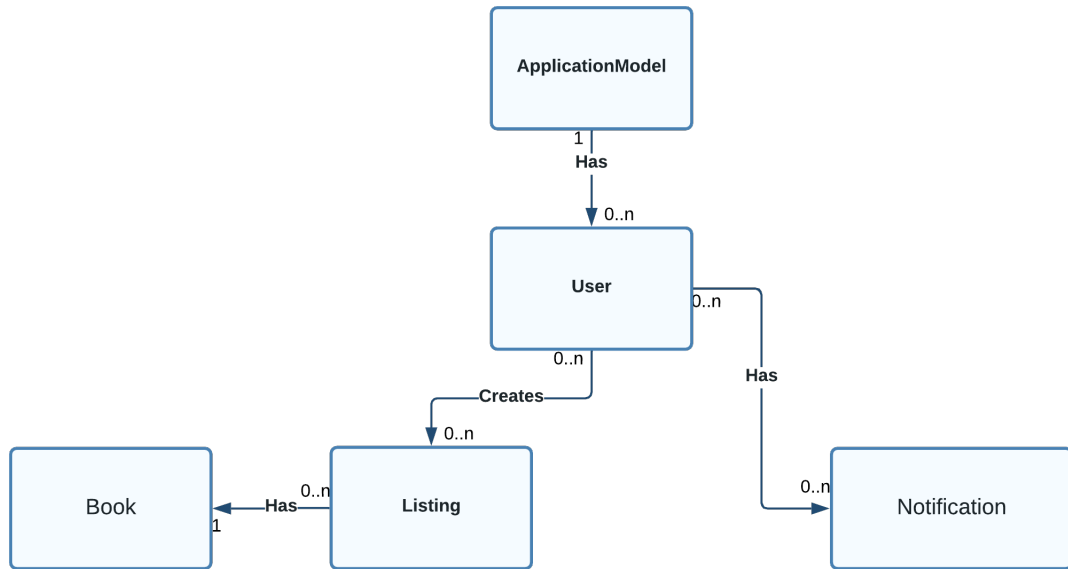


Figure 11: The application's domain model.

3.1 Class responsibilities

The classes are divided to induce modularity and affirm the desired object-oriented design.

3.1.1 ApplicationModel

This class contains the core functionality of the application's model. Superior application tasks such as: creating new listings, updating views through observers and has multiple databases containing users, listings and books.

3.1.2 User

The User class holds all data as well as represents an instance of a specific user.

3.1.3 Listing

The Listing class holds all data pertaining to a specific book listing.

3.1.4 Book

The Book class serves as a container of information of a specific book type. The differentiation from the Listing class is that a Book object is not for sale nor does it have a condition or a seller.

3.1.5 Notification

The Notification class holds and manages all types of different notifications.

3.1.6 Empty classes: EmptyUser, EmptyBook and EmptyListing

The *Empty* classes are placeholders, intended to avoid managing null values in the model. They are not purposed for use, but rather skeletons imitating an actual object.

4 References

The following platforms, build tools, libraries and design tools were used to develop the application.

4.1 IDE

JetBrains IntelliJ IDEA

4.2 Build Tools

Maven 3.8.1

4.2.1 Imported libraries

- JUnit
- JavaFX 16

4.2.2 Structure analysis

JDepend

4.2.3 Quality analysis

- SpotBugs
- PMD

4.3 Design Tools

- LucidChart for UML and domain model design
- Figma for GUI design
- SceneBuilder to construct the GUI and connect it to the code