ChalmersBooks

DAT076

Group 3:

Joakim Mattsson Thorell, joamatt@student.chalmers.se
Jakob Erlandsson, erjakob@student.chalmers.se
Sanjin Slavnic, slavnic@student.chalmers.se

March 2019

Description **3**

Design **3**

Architecture 3

Backend 3

Model 3

Services 4

Authentication 4

Frontend 4

JavaServer Faces 4

XHTML 4

CSS 4

Database 5

Frameworks 5

Bootstrap 5

Primefaces 5

Omnifaces 5

Use Cases **6**

Codeflow **6**

Register 6

Responsibilities **7**

Application structure **8**

Package structure 8

Appendix **9**

Use Cases 9

Login 9

Register 9

List ads 10

Search ads 10

Sort ads 11

Edit ad(s) 11

Create ad(s) 12

Remove ad(s) 12

Change password 13

Forgot password 13

Edit profile info 14

Logout 14

Description

ChalmersBooks is a platform mainly for students, but in general everyone with a CID (Chalmers ID), to buy and sell course literature.

An unregistered user has the ability to list all ads, sort ads by ascending and descending date or price and search ads. However, a user has to be registered with a valid CID mail in order to create ads. An registered user can edit his personal information, remove and edit personal ads. There is also an option to recover an account if password is forgotten.

Design

Architecture

An MVC pattern was implemented throughout the application. The model is updated by the controller. As a rule, the view handles giving output to the user, showing the new states of the model, while the controller handles taking input from the user.

This structure is represented in our package structure and conventional naming of the individual classes; keeping in mind to make our code modular for future development, but also emphasizing good practices and well-engineered OO design that we've picked up and learned from the lectures, assignments and our supervisor.

Backend

Model

The model represents the applications dynamic data structure. It holds required information, independent of the user interface, and is supposed to resemble necessary digitized information in order to sell and/or buy course literature. It manages the data, logic and rules of the application.

In addition, in order to maintain single responsibility principles and encapsulation, a component-driven UI design-model and a component based MVC framework was implemented throughout the application.

Services

The partly involuntary choice of services for the application was JavaServer Faces; a MVC web framework that simplifies construction of user interfaces for server-based applications. It uses XML files often called Facelets. JSF provides FacesServlet's, a request-response Controller that processes requests, events and renders the response to the client, among other things, and is major component in the application.

Authentication

Specific requests made to the server are authenticated by Java EE 8 Security API. This means basically that a user has to login in order to get access to create ad(s) site and (of course) access to personal profile page.

Passwords are stored hashed using SHA256 with help of Pbkdf2PasswordHash, a built in function in Java Security API, which provides security and builds trust for the end users.

Frontend

JavaServer Faces

Is actually a Java-based web application framework intended to simplify development integration of web-based user interfaces. It was used to build server-side user interface and made the development significantly easier

XHTML

XHTML is basically HTML, the language in which Web pages are formulated, written as XML. Whereas XML is a markup language.

CSS

Separate css files were created in order to add additional styling the different XHTML documents.

Database

User	Book
Email - @Id Password - @NotNull Name Phone Number Address Ads - @OneToMany	ISBN - @Id Name Edition Author
Course Code	**Ad**
Name - @Id Books - @ManyToMany	Id - @Id Book - @ManyToOne CourseCodes - @ManyToOne Price Date ImageURL User - @ManyToOne

Frameworks

We implemented various frameworks, mainly for frontend use, enabling smooth layout support, pretty buttons and other basic visual components.

Bootstrap

An open-source front-end web framework. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components.

Primefaces

An open-source user interface component library for JavaServer Faces-based applications

Omnifaces

Another open source utility library for the JavaServer Faces framework, making JSF life easier by providing a set of artifacts to improve the functionality of the JSF framework.

Use Cases

See Appendix.

Codeflow

Register

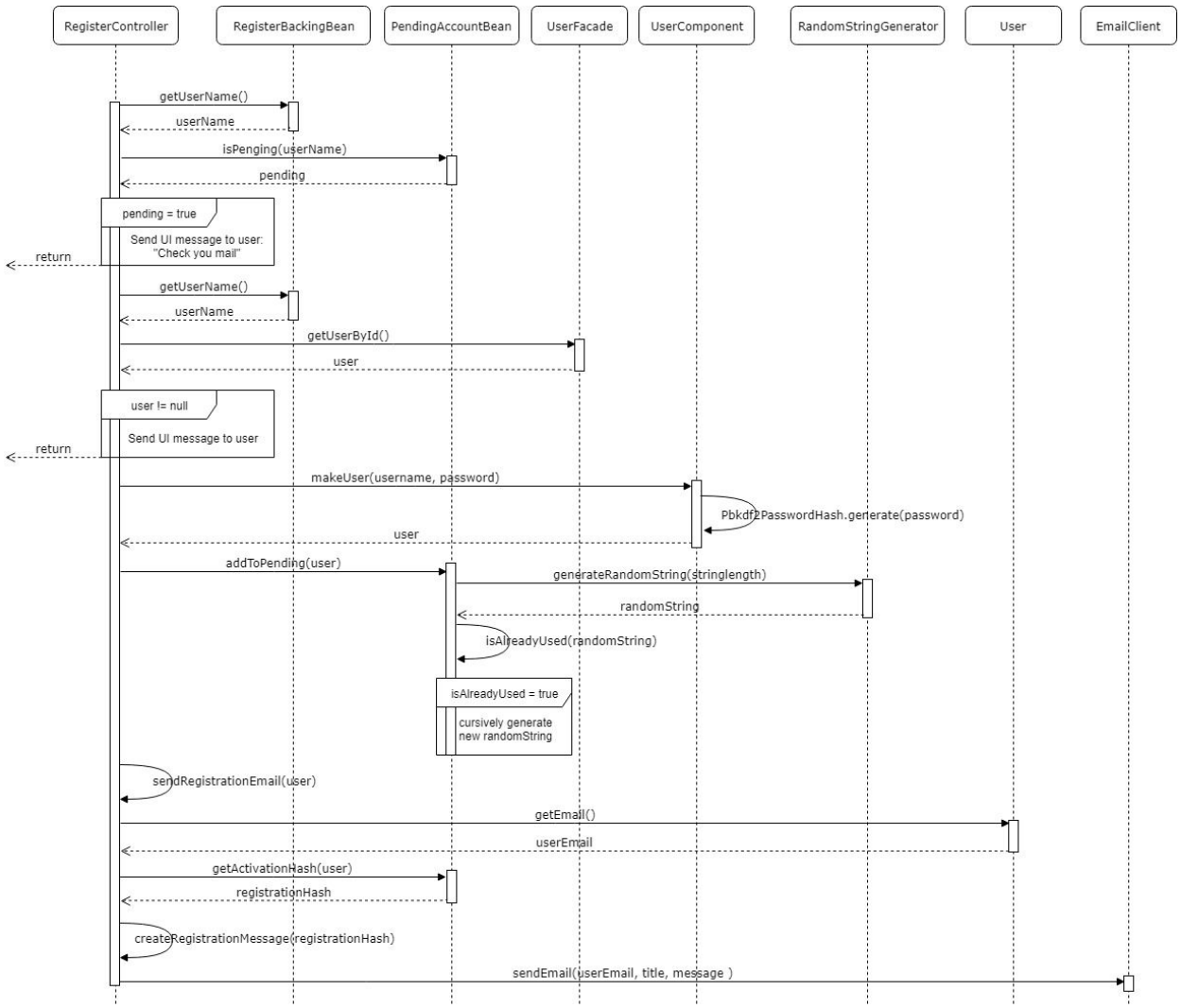

Figure 1.

Figure 1 shows a sequence diagram when a user submits a registration. This event calls the register() method in RegisterController class. If user is not already registered or hasn't already submitted an email the sequence will call sendEmail() method in the EmailClient class and sends an email to the email the user entered in the registration field with a confirmation link.

Responsibilities

Initially, we designed the entire core (model and database) and mock ups together. We thereafter divided work and had specific responsibilities. Nonetheless, we built most of the core of the App together. We met each week and helped each other by pair programming to solve tasks. Unfortunately, no fair view is given by the statistics in the git repo (Gitinspector).

We had two large refactorings of the code structure and packages, class responsibilities and good practices, which we discussed and arranged together.

Sanjin Slavnic

- Login
- Register
- Sort ads

Joakim Mattsson

- Login
    - Forgot password
- Register
- Create ad(s)

Jakob Erlandsson

- Profile
    - Edit
        - Password
        - Profile info
    - List users ad(s)
        - Remove users ad(s)
        - Edit ad(s)
- List ad(s)
- Search ad(s)

Application structure

Package structure

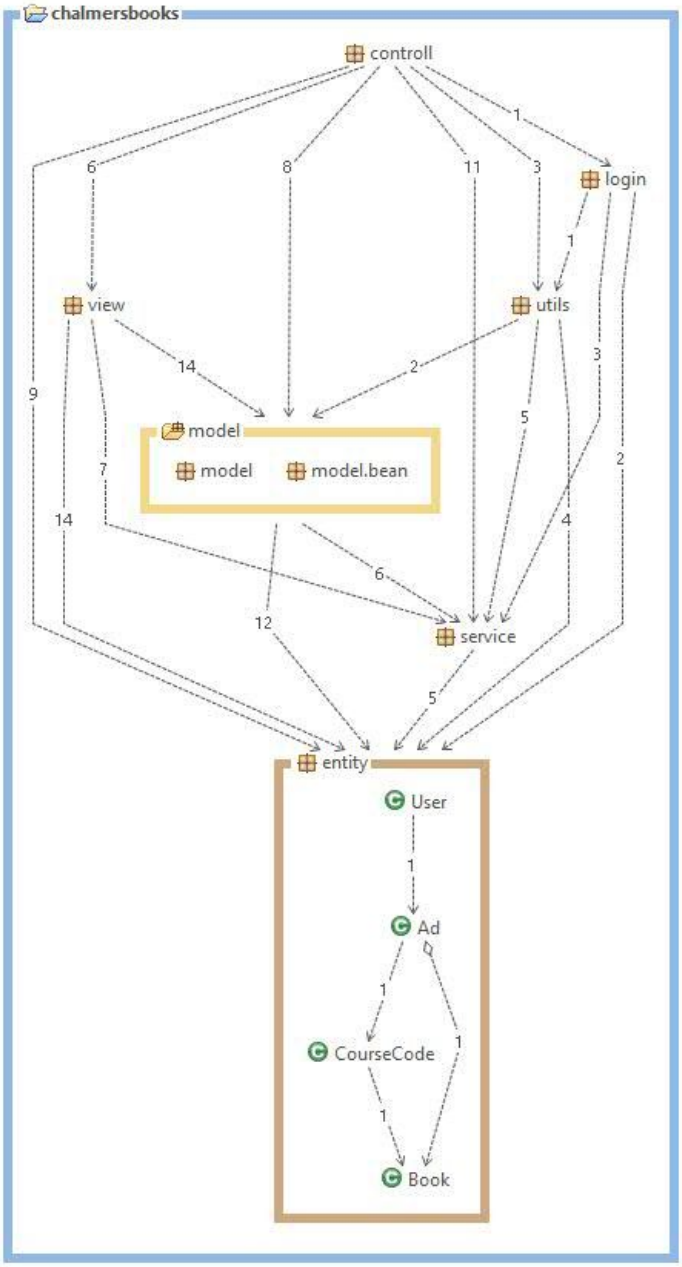

Figure 2.

Figure 2 shows application package structure and displays dependencies.

Appendix

Use Cases

Login

Description
As an user I want to login

Actor
Registered user

Preconditions
Actor must be a registered user

Standard path
User is redirected to main page

Alternate path 1
Actor enters wrong email or password

Alternate path 2
Actor presses *forgot password*

Postcondition
Give user permission to enter whole site

Register

Description
As an user I want to register

Actor
New user

Preconditions
All fields must have correct input

- Password must contain at least:
  - Eight characters
  - One uppercase letter
  - One special character
  - One number

Standard path

1. User submits registration
2. User gets redirected to confirmation showing an email has been sent with a link
3. User has to use link in order to complete the registration

Alternate path 1
Actor enters wrong password(s)

Alternate path 2
Actor presses *Cancel* and gets redirected to main page

Postcondition
Backend stores users email and awaits for the confirmation link to get used in order to store the user in the database.

List ads

Description
As a user I want to see all ads

Actor
Website visitors

Preconditions
Ads must exist in order to see listed ads

Standard path
Ads are listed in a listview

Alternate path 1
Ads are listed in a gridview

Alternate path 2
Actor can see detailed info of each ad, such as seller's mail or phone number.

Postcondition
Ads exist in the database, ready for querying

Search ads

Description
As a user I want to search course literature

Actor
Website visitor

Preconditions
Ads must exist in order to search for ads

Standard path
Actor enters for example:

- Book name
- Course code

which in that case will list all ads that contain and match the input.

Alternate path 1
Actor enters non-existing input, in which case no books are shown

Alternate path 2
Actor enters nothing in the fields and presses search in which case every books is shown

Postcondition
A query is executed to find a match of the input.

Sort ads

Description
As a user I want to sort listed ads

Actor
Website visitor

Preconditions
Ads must exist in order to sort and list ads

Standard path
User sorts listed ads by date added

Alternate path 1
User sorts listed ads by book title

Postcondition
A query is made to the database which gets a sorted list of ads, either by book title or date added

Edit ad(s)

Description
As a user I want to edit my ad(s):

- Price

Actor
Registered, logged in user

Preconditions
User must be logged in and have created ad(s)

Standard path
User lists ads on Profile site and edits available listed ad(s)

Postcondition
Price of ad is updated in the database

Create ad(s)

Description
As a user I want to create an ad

Actor
Registered user

Preconditions
User has to be registered and login in in order to create an ad

Standard path
A wizard guides the user through creating a new ad making sure all information is correct

Alternate path 1
The user manually inputs the parameters needed to create an ad

Postcondition
The new ad is added to the database and displayed in the list of ads

Remove ad(s)

Description
As an actor I want to remove an ad I've created

Actor
Registered user

Preconditions
User must be logged in and have created an ad

Standard path
User goes to profile page, lists ads and removes chosen ad

Postcondition
Ad is removed from database

Change password

Description
As a user I want to change my password

Actor
Registered user

Preconditions
User has to be registered and logged in

Standard path
User goes to profile page and changes password

Alternate path 1
New passwords don’t match

Postcondition
New password is hashed and updated

Forgot password

Description
As a user I want to recover access to my registered account

Actor
Registered user who has forgotten their password

Preconditions
User must have a registered CID mail and exist in the database

Standard path
User enters registered CID mail in order to get a new password

Alternate path 1
User enters wrong or non-existing CID mail

Alternate path 2
User abort recovery

Postcondition
An mail is sent to the user with a new random-generated password

Edit profile info

Description
As an user I want to edit personal info:

- Name
- Phone number
- Address

Actor
Registered user

Preconditions
User must have a registered CID mail and exist in the database

Standard path
User opens dialog and edit information with new valid information

Alternate path 1
User enters a phone number that is not only numbers

Alternate path 2
User cancels the dialog without changing any information

Postcondition
Database is updated with the new information

Logout

Description
As a user I want to log out of the application

Actor
Logged in user

Precondition
User must be logged in

Standard path
User presses logout button

Postcondition
User can no longer reach authorized pages or edit any information there