

# Hemlagat

A marketplace for home-cooked food

Nurhusein Abdulkader  
Daniel Cebe  
Rickard Gyllensten  
Roman Melnik

# Introduction

When one is looking to create a web application, there are plenty of options to choose from. Our group considered several ideas: a webshop, a new type of message board, and a marketplace. We decided to proceed with the latter. While pondering on different concepts for a marketplace, a seemingly unoccupied niche with great potential was identified. The idea for our application is a marketplace for home-cooked food sold by private persons. The project is aptly called Hemlagat (home-cooked) and offers all the basic features one would expect and a few rather advanced ones, such as rating and picture uploading.

The git repo can be found at: <https://github.com/cebed/WebApplications-DAT076.git>

## Use Cases

1. Login/logout, Register new account
2. Displaying My profile page containing username, email. Changing password
3. Find all ads, purchases or sales for a certain user (by entering email)
4. Creating an ad including uploading a photo
5. Finding all ads from a certain city
6. Viewing detailed info about an ad
7. Buying an item after confirming your order
8. Submit/view rating for an ad or a seller, including text comments
9. Sending a message to customer service which is connected to gmail.

## User Manual

On the main page users can either sign in/register right away or view the ads that are available for a certain city. When a user who has not signed in tries to buy an item, he/she will be redirected to the login/register page.

A user who has registered and signed in can view their profile and change their password. When the user wants to find ads, he/she enters a location, and all the ads from that city are displayed. Then the user can click on an ad and see a detailed view of it. Clicking the “Köp” (Buy) button will take the user to a Confirm Order page, where the purchase can be completed. After completed purchase, a user can submit a rating and a text comment for the item that was purchased. The total rating for each seller based on their ads is shown in the ad.

Users who wish to sell food can create ads containing a title of their dish, as well as a description, picture, quantity, weight, price, and allergy information. A photo of the dish can be uploaded.

# Design

The application is built on Java EE platform using Netbeans 8.2 IDE. The front end of the application is coded in HTML. Component tags are used to obtain components from libraries. Expression language is used to connect to objects in the back end. Several component libraries are used: JavaServer Faces, PrimeFaces and BootsFaces.

General layout of the site is handled by using templates, including separate templates for header and footer. The templates refer to Cascaded Style Sheets (CSS) that the group designed.

The back end of the application is a database mapped to objects by Object-relational Mapping (ORM). This is implemented by using Java Persistence API (JPA). The database is generated locally on each computer by using the Apache Derby database management system, which is embedded in Netbeans.

Facades are a design pattern used to achieve loose coupling. Lombok library is used to generate getters, setters and other methods automatically.

## Application Flow

### Use case: Login

When a user signs in, he/she enters email and password and a *login.login()* request is sent to the server. The email and password parameters are handled by LoginController, which injects UserDBFacade and UserBean. LoginController calls method *findUser()* in Model class UserDBFacade. UserDBFacade calls method *equals()* in Model class UserDB to match email and password to the database. If login is successful, LoginController sets email and username of the logged in user in the named session-scoped bean UserBean and redirects the user to a new view, Locationpage.

Unauthorized access is filtered by SessionFilter, which redirects one to LoginPage, if a user who is not signed in tries to navigate to pages that require login.

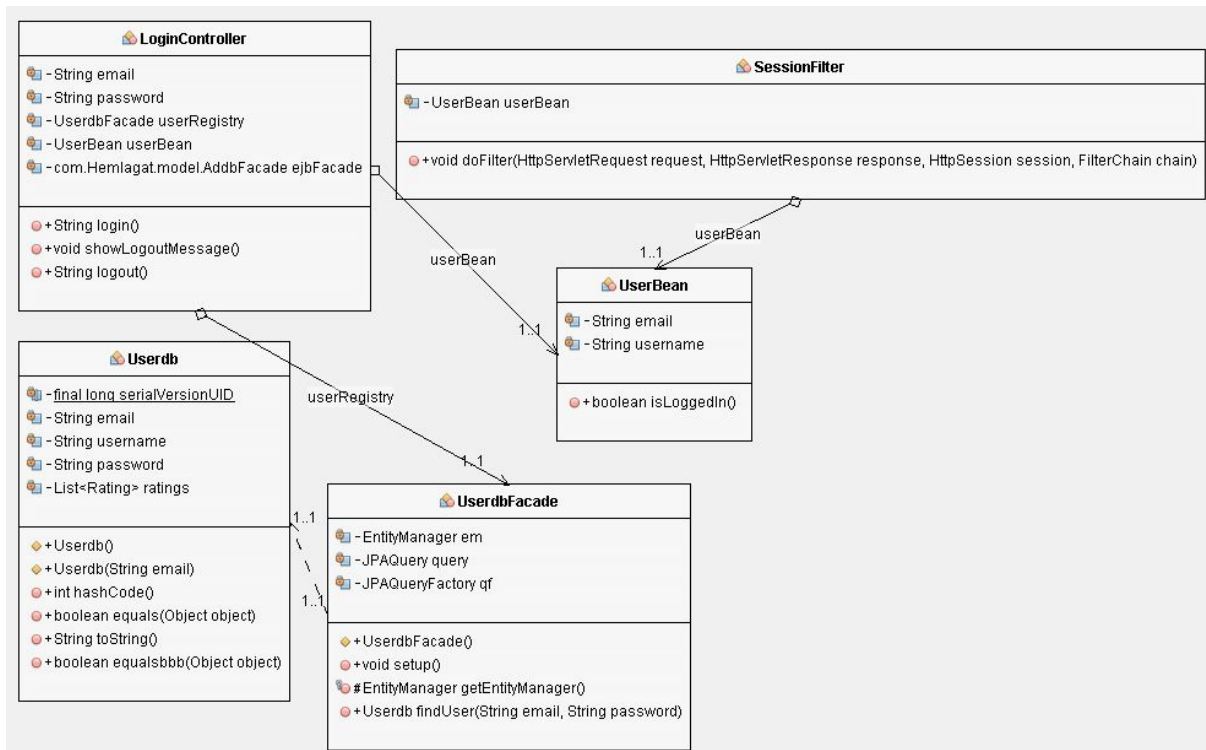


Figure 1. UML diagram for Use case: Login

### Use case: Submitting a rating

When a user is logged in and bought a dish he can leave a rating to the seller, a `ratingController.create()` request is sent to the server. The request is handled by RatingController, which calls the method `create()` in Model class RatingFacade, which in turn adds the rating to the database.

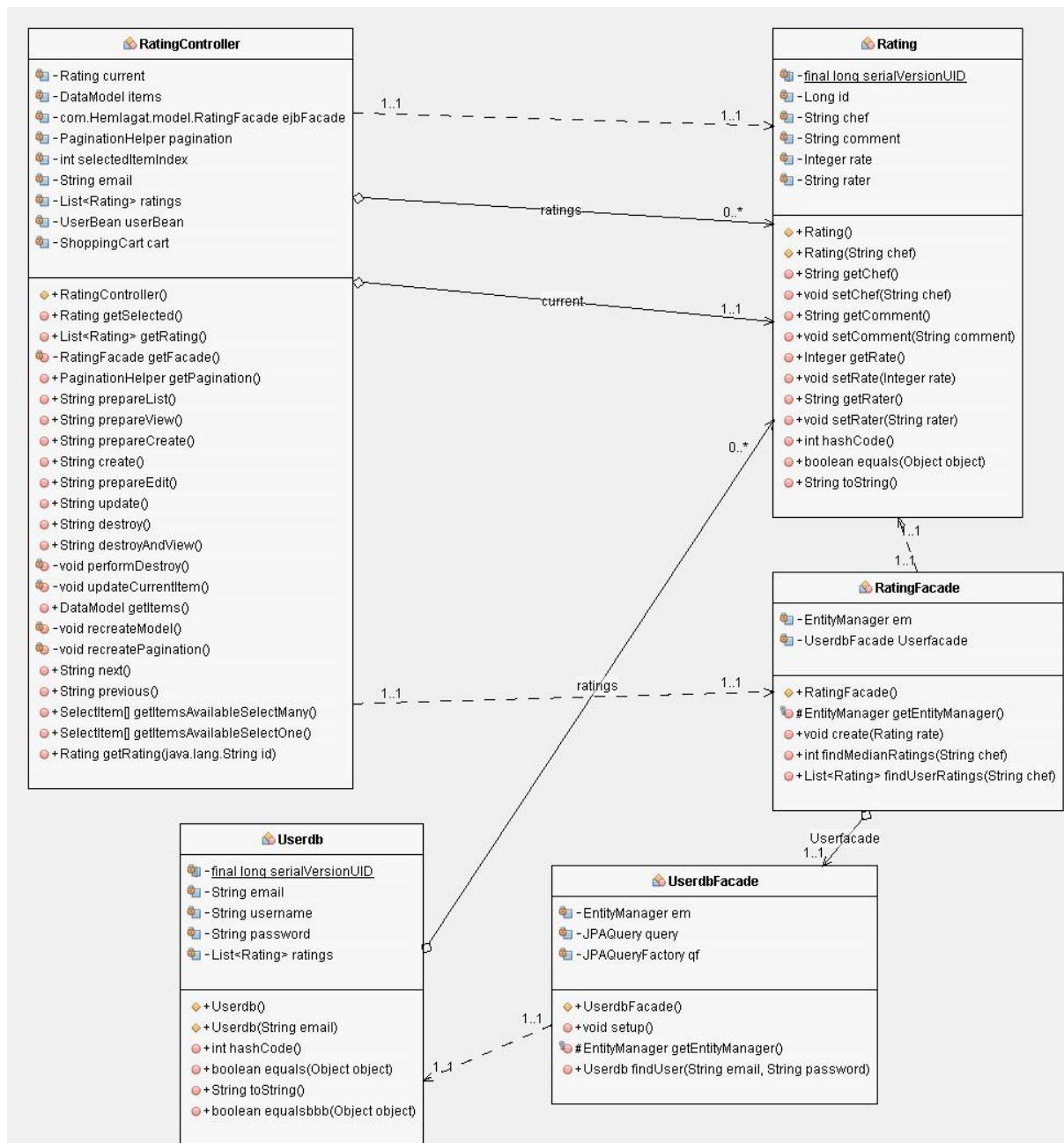


Figure 2. UML diagram for Use case: Submitting a rating

The complete UML diagram for the project is available as a separate file on Github in root directory.

## Responsibilities

Daniel - Login/Logout/Register, session filters, uploading photos, Contact us page

Nur - Creating ads, finding ads from a certain city, viewing your purchases/sales/active ads, detailed view of ads, visual design, templates, CSS

Rickard - Everything about rating

Roman - My profile page, Change password, a few visual design elements, the entire project report, presentation slides