IS Assignment 2

Integração Sistemas

Francisco Barão 2016238992

Pedro Ribeiro 2016218753

Introduction

In this project we had to build an enterprise application. We were suppose to build MyBay which is a web application to manage an online store of secondhand items. MyBay contains information of users and their items.

As we started the project, we learned early that the most complicated step would be the configuration of the application.

Connecting the database to wildfly and configuring the maven project were our biggest challenges, after that, the actual building of the application went pretty smoothly.

Dividing the project in three clear layers helped us build it in an easier and cleaner way.

This report is done to explain this 3 different layers.

Presentation Tier

When we first get to our implementation of MyBay, we encounter a login screen with two text boxes and a submit button and also a login and register button to allow the user to go to these two Urls.

After registering and logging in, the user goes to home where the user visualizes a welcoming message, a search button and a new header where he can go to his profile, logout and return to Home-Refer to the structure of webpage (with layout).

If the user clicks on the Logout button he will logout and redirected to the Login page.

If the user clicks on the Home Button he will be redirected to the Home Page.

If the user clicks on the search button he will be redirected to the search form where he will be faced with a few optional parameters for him to fill. After Filling these out, the user will either be presented a message warning him that the search is empty or he will be shown some filtered search results on a new page.
This new page will contain the mentioned items, these being clickable to better visualize all their information in a new page, and a form containing a dropdown button with different sorting methods, a radio button for the user to choose descending or ascending order and a submit button.

If the user chooses a method and an order, the items will be re-presented in a new sorted way.

If the user clicks on the Profile Button he will be redirected to the Profile page where he will be presented all his information, an edit and delete button for the user, a create item button, all his items sorted by date of insertion and edit/view/delete buttons for each his items.

Password storage

The user inputs the password on a “password” type input field which is then sent to the Business

layer where it is hashed before being stored on the database.

Hard to implement

Photo for item was the only aspect we considered difficult mostly due to the lack of updated information.

Business tier

In this tier we perform Independent operations where we receive information from the Presentation tier and we use JPQL to formulate queries to access our database(CRUD).
We decided to have all our EJBs @stateless and all our Interfaces @Local since we have no need to maintain a state with our clients.
Our transaction management was automated by JPA Persistence Context and entity manager.

UserEJB - EJB where we perform user operations.

Login

Function that receives an email and password and after hashing the password, creates the query and a User which will be returned if found on the database.
Data transferred: String Email | Hashed String Password
Data collected: User Object containing all his user information.

Register

Function that receives user info and after checking that the email is valid, the password is hashed and a User is created and persisted to the database.
Data transferred: String email | Hashed String password | String country | String name

Read

Function that receives an email, creates the query and retrieves the User from the database if it exists. Otherwise returns Null.
Data transferred: String Email
Data collected: User Object containing all his user information.

SelectAllUsers

Function used on sendEmail function of EmailEJB which creates a query that retrieves all users from the database.
Data collected: All users on Database

Update

Function that receives User parameters and after checking if the email is valid (Non existing on database or is the same email that User had) the query is created and the update is executed.
Data transferred: String email | Hashed String password | String country | String name

Delete

Function that receives User id, and starts by calling DeleteAll service from ItemEJB to erase all user's items and moves on to creating the JPQL query that will be deleting the user from the database.
Data transferred: String id

Hash - Function that receives a String password and uses a Hash to encrypt it.

Interface - All the above methods are on the interface except the Hash Function since there is no need to call it from the other tiers.

ItemsEJB - EJB where we perform item operations.

Create

Function that receives item parameters and owner, creates the item and persists it into the database.

Data transferred: Item containing String name | String category | String country | int price | Date date | blob photo | User user

Read

Function that receives an item id, creates a typedQuery and gets a single result from the database.

Data collected: Item item

Search

Function that receives Search parameters and checks each of the parameters to see if they were filled to filtrate the user search and modify the query accordingly. For example, if the user chose a name we add the clause where “name Like ‘&name&’ ” to the search.

Data transferred: Search parameters - String name | String category | int minPrice | int maxPrice | String inCounty | String afterDate

Data collected: Items that fit into the criteria

SearchRecentItems

Function that creates a query that returns the last three added items.

Data collected: Last three items that were added

SearchByUser

Function that receives a User ID and returns all items that belong to it.

Data transferred String userId

Data collected: All Items that belong to that User

CheckUserItem

Function that receives a user Id and an item Id and creates a query that will search the database to check if the item belongs to the user returning true or false depending on what is found.

Data transferred String userId | String itemId

Data collected: Item if it belongs to the user

Sort

Function that receives a List of items to sort, a method and a boolean for the order. Depending on the method, the list of items will be sorted by a different parameter and returned to the user.
No communication is done with the data layer.

Update

Function that receives all item parameters, constructs the query and updates the item information.
Data transferred Item parameters

Delete

Function that receives an item Id and deletes it from the database.
Data transferred: String itemId

Delete All

Function that receives a String User ID and constructs a query that will delete all items on the database containing that id.
Data transferred: String userId

EmailEJB

EJB where we send the emails to the users.
This EJB contains the tag @startup and @schedule so as to begin when the server launches and automatically run in the background.

sendMail

Function that runs every 5 minutes and that gets all users emails. After having all emails on the address list, we get the latest three items added to MyBay and formulate the message. In the end we send the mails to the users.
Since we use two of the above mentioned functions there is no direct contact with the data layer, but indirectly:

Data collected: All users and the three most recent added items.

Logging

For logging we used three layers. INFO, DEBUG and ERROR

Data Tier

-Entity Relationship Diagram

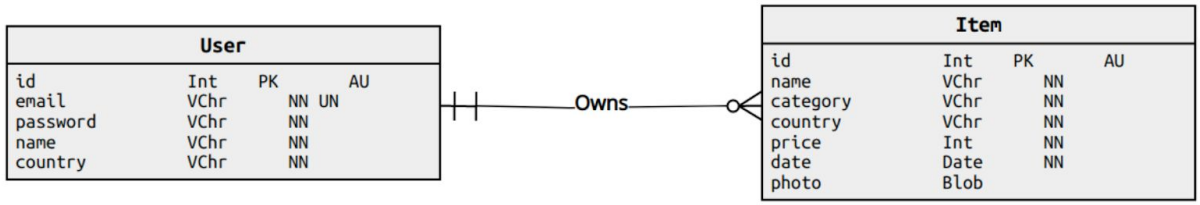

Fig 1. Entity Relationship Diagram

Project Management and Packaging

The project is divided in four sub projects, each one with each “pom.xml”.

First we have the data project, which has all the models necessary to run the project, in this case Item model and User model.

Then we have the business project that has all the EJB, User EJB, Item EJB and Email EJB , it also has some POJO (Plain Old Java Objects) like the sorting objects. This project is responsible for communication with the data layer.

In the third we have the web project, which deals with the human interaction via the browser, has all the views and calls the respective EJBs to allow interaction with the database.

Last but not least the ear project.

We then have the maven project that ties everything together, and allow us to deploy the application to wildfly.