PROJEKT - WEBAPPLIKATIONER

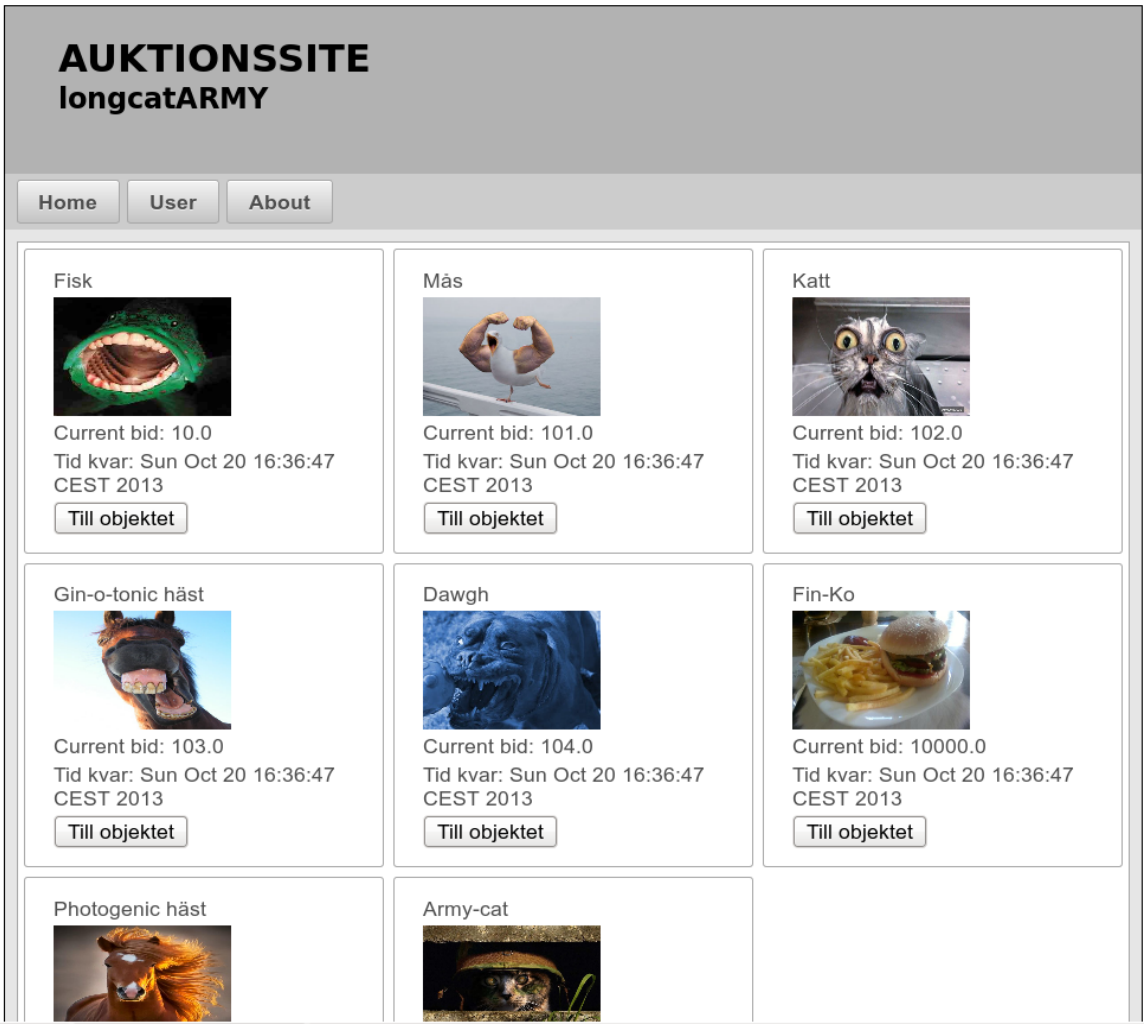
LONGCATARMY - GROUP 16

Alexander Lissenko, 910403-5895,cid: lissal

Emelie Svensson, 920227-2044, cid: emesven

Henrik Ståhlsparre, 820917-5994, cid: k02sthe (stahlsparre)

William Axhav Bratt 920227-1897, cid: axhav



General overview over the system.

Our project is meant to represent an auction site, where different users can add objects and place bids on each others objects. More or less, this is a simple web shop with a bit different functionality (bids instead of direct buys, *although the bid method does not work*).

Our main idea is that we have two different types of users: admin and customer – each with different permissions.

- Customer is allowed to *view auctions, add new auctions and bid on auctions*.
- Admin is allowed to *edit existing auctions and customers*.

However, login functionality is not yet implemented so we basically just have an admin and one default logged in customer.

Use cases

LongcatArmyview:

1. Home (index)
    - Browsing the homepage.
    - By clicking an auction viewing it.
    - Display all current auctions.
2. User
    - Browse users objects put up for auctioning and the objects that user has placed a bid on.
    - View an auction by clicking it.
    - Create a new auction.
    - Edit the profile.
3. View auction
    - View the auction.
4. About
    - Viewing information about creators, with collapsable infofields.

Admin view:

- Browse through the tables by clicking Prev and Next buttons
- By clicking a row in the table, an Edit/Delete-dialog pops up.
- Editing and deleting a customer or auction object is connected to the database

Technical design

We're using two types of approaches from the workshops - "*component based*" and "*service based*" approach. Most of the application (web application **LongcatArmy**) is written in the component based approach, since we find this approach most satisfying working with.
We've also made a web application in the service based approach (**AuctionAdmin**), which is built up as a one page restful application.

For GUI, we've mostly used standard jsf with some implementations of PrimeFaces-components together with our own custom made CSS.

In the Admin-view we have two "tables", one that contains all customers and one containing all auction objects.

To implement this application from a physical point of view you need to have a setup that consists of a database and a web server.

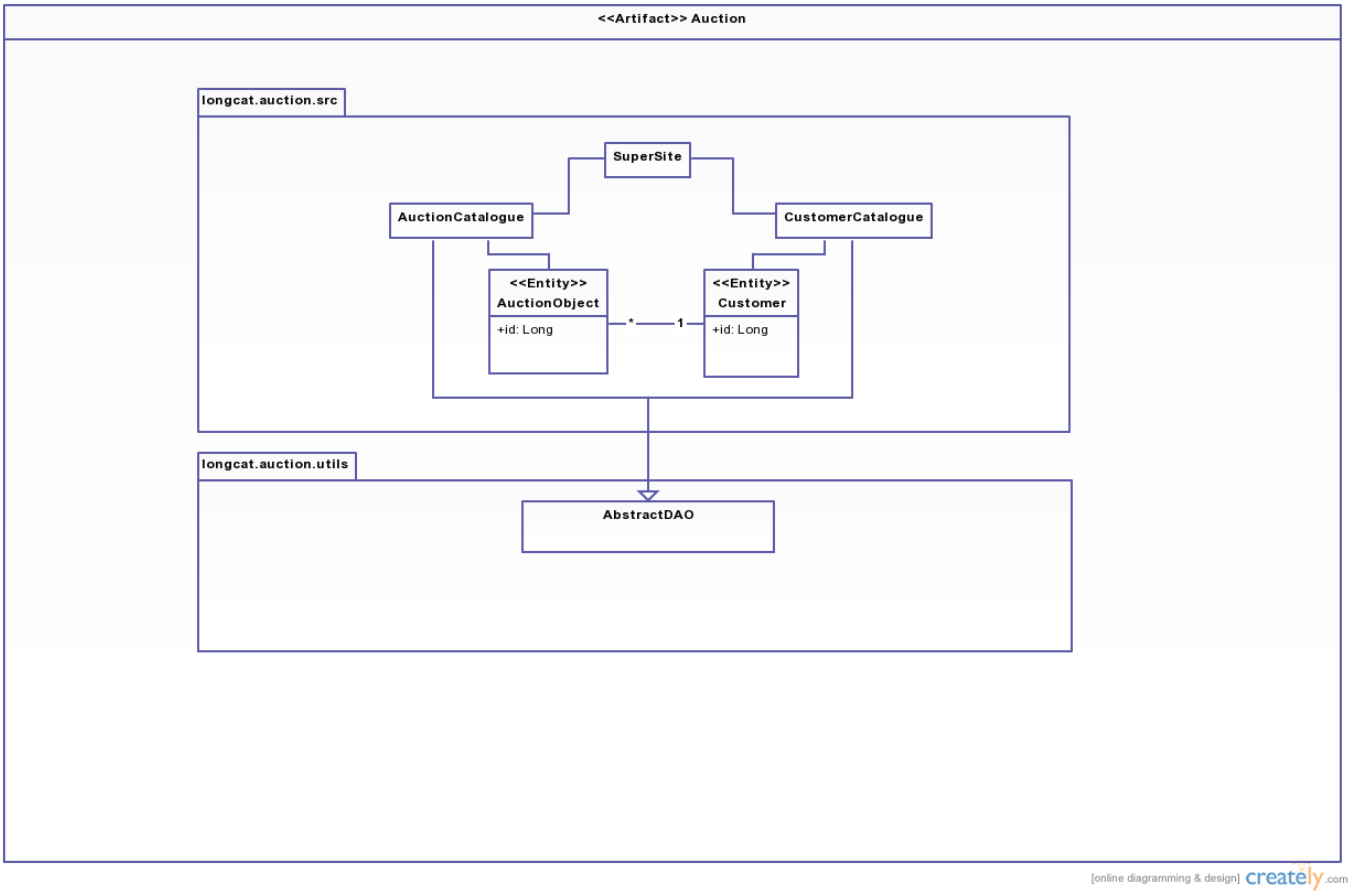
The application is running the standard jsf design, with built-in support of primefaces. It is directly connected to a java Derby database. The main component "**LongcatArmy**" displays a user friendly view of the auctions contained in the database. Here you have the possibility to add and bid (not fully implemented) on auctions and view and edit your account profile.
"**AuctionAdmin**" on the other hand is an admin interface created with REST-functionallity where you easily can edit and delete entities managed by the database.

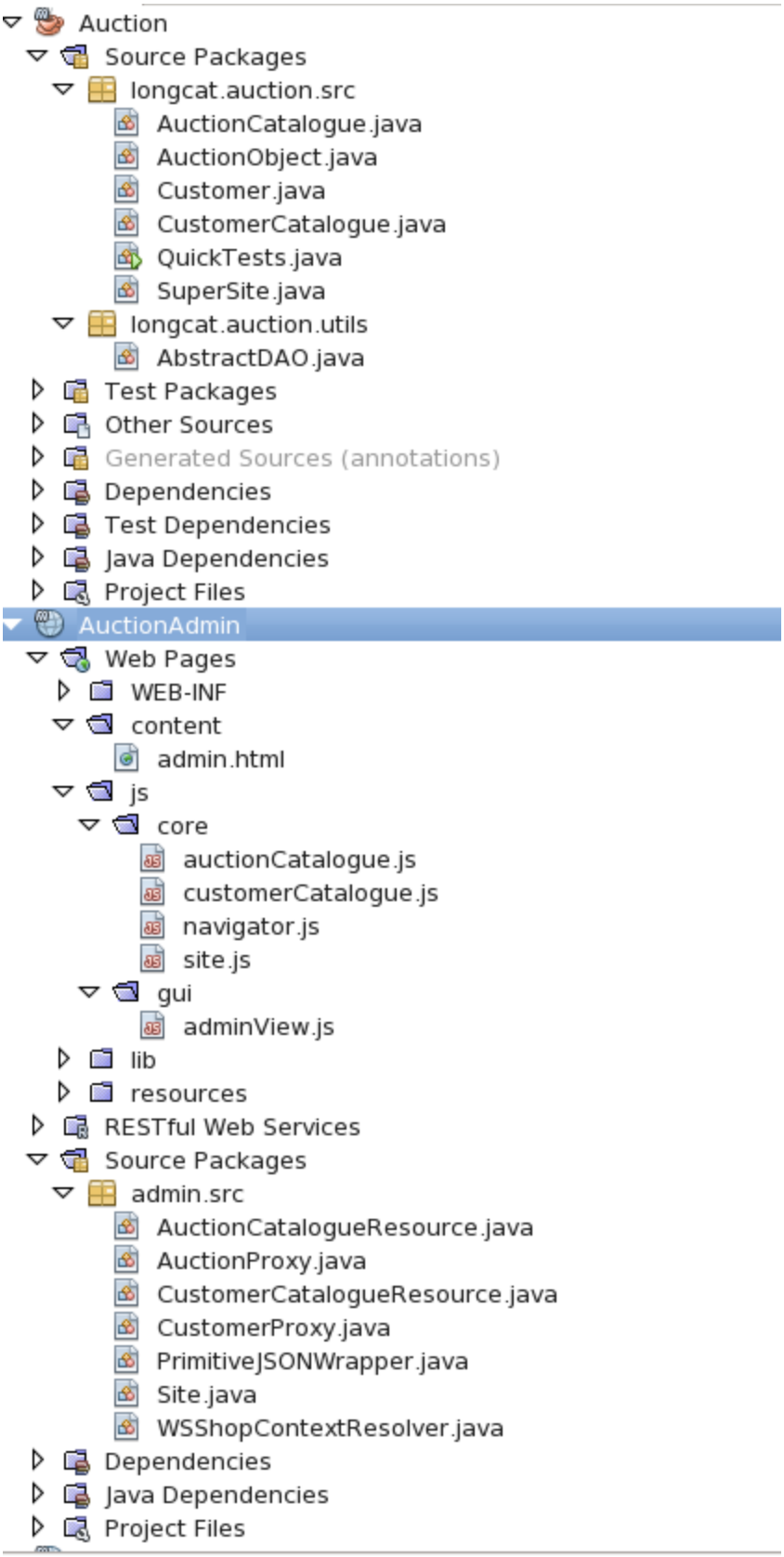
The responsiblities for each module:
AuctionAdmin : REST with AJAX, JSON, JQuery
LongcatArmy : JSF with Primefaces
Auction: Backend with JPA

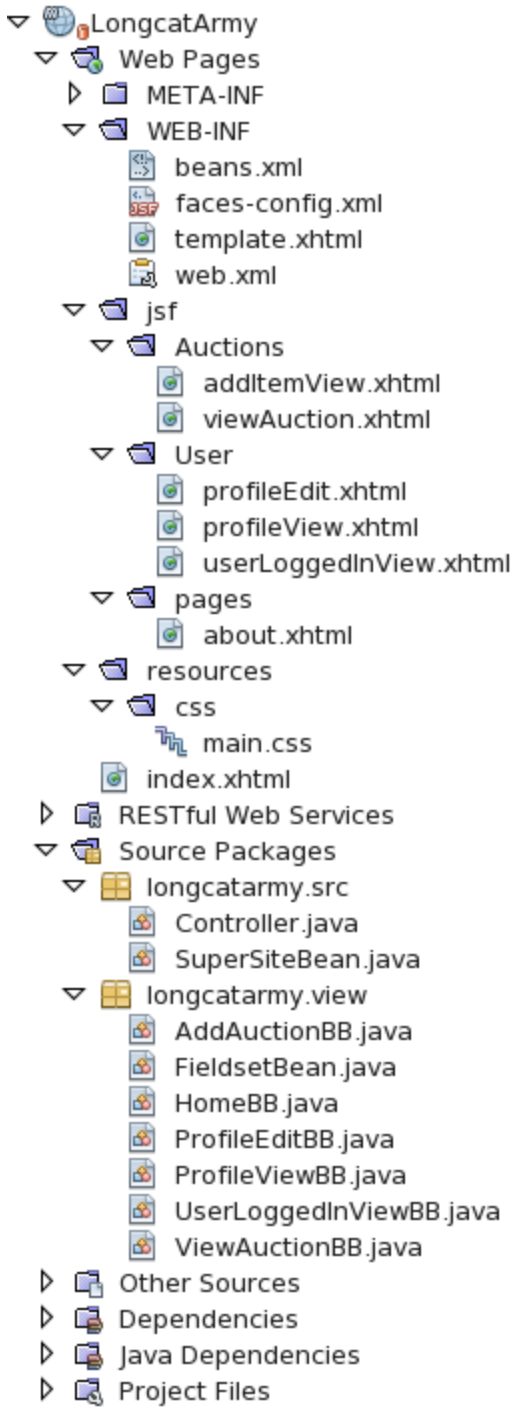
A layered view:

GUI		Application Layer		Model		Persistence		Services
LongcatArmy.view		LongcatArmy.src		Auction		AuctionObject		No Services
AuctionAdmin/js		AuctionAdmin resource				Customer		

The Object-oriented model as an UML-diagram.




Auction
Source Packages
longcat.auction.src
AuctionCatalogue.java
AuctionObject.java
Customer.java
CustomerCatalogue.java
QuickTests.java
SuperSite.java
longcat.auction.utils
AbstractDAO.java
Test Packages
Other Sources
Generated Sources (annotations)
Dependencies
Test Dependencies
Java Dependencies
Project Files
AuctionAdmin
Web Pages
WEB-INF
content
admin.html
js
core
auctionCatalogue.js
customerCatalogue.js
navigator.js
site.js
gui
adminView.js
lib
resources
RESTful Web Services
Source Packages
admin.src
AuctionCatalogueResource.java
AuctionProxy.java
CustomerCatalogueResource.java
CustomerProxy.java
PrimitiveJSONWrapper.java
Site.java
WSShopContextResolver.java
Dependencies
Java Dependencies
Project Files


LongcatArmy
Web Pages
META-INF
WEB-INF
beans.xml
faces-config.xml
template.xhtml
web.xml
jsf
Auctions
addItemView.xhtml
viewAuction.xhtml
User
profileEdit.xhtml
profileView.xhtml
userLoggedInView.xhtml
pages
about.xhtml
resources
css
main.css
index.xhtml
RESTful Web Services
Source Packages
longcatarmy.src
Controller.java
SuperSiteBean.java
longcatarmy.view
AddAuctionBB.java
FieldsetBean.java
HomeBB.java
ProfileEditBB.java
ProfileViewBB.java
UserLoggedInViewBB.java
ViewAuctionBB.java
Other Sources
Dependencies
Java Dependencies
Project Files