# Software Engineering Group 2 (2009-2010)
# Software Design Document

Bart Maes

November 30, 2009

**Abstract**

This document, the Software Design Description lays down the architecture
and detailed design of the Salesmen software, to be conceived in the context of
the annual project for the *Software Engineering* course. It took place under
guidance of Prof Dirk Vermeir and assistant Yoeri De Koster during the
2009-2010 study year at the Vrije Universiteit Brussel (Brussels, Belgium).

*Document history:*

| version | date | author | description |
|---|---|---|---|
| v1 | 27 Nov 2009 | Bart Maes | Draft initialization |
| v2 | 29 Nov 2009 | Bart Maes | Added Appendix A, added Technical Design |
| v3 | 29 Nov 2009 | Bart Maes | Added Appendix B, initial QA check |

# Contents

# Chapter 1

# Introduction

## 1.1  Purpose

This document specifies the entire software architecture and design for the Salesmen software. These design decisions directly relate to the functionalities, performances, constraints, attributes and interfaces of the system.

## 1.2  Scope

This document describes the software architecture and design for the initial release of the Salesmen software, version 1.0. The intended audience of this document exclusively includes the designers, the developers and the testers of the software.

This SDD document mainly follows the standards of the IEEE 1016-1998 standard.

## 1.3  Reference materials

- IEEE Std. 1016-1998 - Software Design Descriptions

- `http://tinf2.vub.ac.be/~dvermeir/courses/software_engineering/slides.pdf` [1 Nov 2009].

## 1.4 Definitions and acronyms

| | |
|---|---|
| **CSS** | Cascading Style Sheets |
| **HTML** | HyperText Markup Language |
| **EJB3** | Enterprise JavaBeans v3 |
| **J2EE** | Java 2 Platform, Enterprise Edition |
| **jBoss** | An application server |
| **JSP** | Java Server Pages |
| **RDBMS** | Relational DataBase Management System |
| **Servlet** | (Java) program that runs in a J2EE webcontainer |
| **SCMP** | Software Configuration Management Plan |
| **SDD** | Software Design Document |
| **SMTP** | Single Mail Transfer Protocol |
| **SRS** | Software Requirements Specification |
| **SQL** | Structured Query Language |

# Chapter 2

# System Architecture

## 2.1 Overview

System architecture diagram: see Appendix A, 8.1

## 2.2 Three-Tier Architecture

Three-tier is a client-server architecture in which the user interface, functional process logic, computer data storage and data access are developed and maintained as independent modules, most often on separate platforms.

Apart from the usual advantages of modular software with well defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently as requirements or technology change. For example, a change of operating system in the presentation tier would only affect the user interface code.

Typically, the user interface runs on a desktop PC or workstation and uses a standard graphical user interface, functional process logic may consist of one or more separate modules running on a workstation or application server, and an RDBMS on a database server or mainframe contains the computer data storage logic.

Three-tier architecture has the following three tiers:

- **Presentation tier** This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.

- **Application tier (Business Logic/Logic Tier)** The logic tier is pulled out from the presentation tier and, as its own layer, it controls an applications functionality by performing detailed processing.

- **Data tier** This tier consists of Database Servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.

Benefits of using this architectural style include increased performance, flexibility, maintainability, reusability, and scalability.

## 2.3 Discussion of Alternative Designs

**MVC** At first glance, the three tiers may seem similar to the MVC (Model View Controller) concept; however, topologically they are different. A fundamental rule in a three-tier architecture is the client tier never communicates directly with the data tier; in a three-tier model all communication must pass through the middleware tier. Conceptually the three-tier architecture is linear. However, the MVC architecture is triangular: the View sends updates to the Controller, the Controller updates the Model, and the View gets updated directly from the Model.

From a historical perspective the three-tier architecture concept emerged in the 1990s from observations of distributed systems (e.g., web applications) where the client, middleware and data tiers ran on physically separate platforms. Whereas MVC comes from the previous decade and is based on observations of applications that ran on a single graphical workstation; MVC was applied to distributed applications much later in its history.

## 2.4 Survey of Technologies Used

### 2.4.1 Introduction

This section describes briefly what software and technologies will be used to implement the Salesmen software. For more information about these technologies, the reader is kindly referred to the SCMP.

### 2.4.2 Presentation Layer

Java Server Pages (JSP) is a technology from Sun that allows easy development of GUIs using webbrowser clients. JSP helps in making a clear distinc-

tion of layout and code concepts, so that a developer and graphical designer can independently work on the same webpages with little interference.

### 2.4.3 Business Layer

JBoss Application Server (or JBoss AS) is a free software/open-source Java EE-based application server. Because it is Java-based, the JBoss application server operates cross-platform: usable on any operating system that Java supports. As part of the jBoss software, the Apache Tomcat servlet container will allow us to serve dynamically parsed webpages (JSP).

### 2.4.4 Data Layer

Two database platforms were considered: MySQL and PostgresSQL.

Our programmers will not need to write SQL statements themselves, as these are generated with the use of the EJB3 standard and Java annotations. Based on Hibernate concepts, EJB3 allows for persistant objects without the need for platform dependant SQL statements and other 'dirty' constructs.

## 2.5 Physical Configuration

The software is to be installed on the Wilma server at the VUB datacenter. Wilma is accessible over the internet through web browsers and command line clients. The software relies on the availability of an SMTP server on the local LAN of the VUB, so that we do not have to include MDA (Mail Delivery Agent) functionality in the software.

System infrastructure diagram: see Appendix A, 8.2

## 2.6 System Interface Description

The web-interface is the only interface available to users of the Salesmen software. However, depending of the access-rights provided to a given user, more options may become available. Website administrators, for example, can close auctions or ban other members while 'normal' members cannot access these facilities. A more detailed description of the access-rights provided to each group of users will be provided in the SRS. Prototypes of the webinterface can be found in Appendix B.

# Chapter 3

# Technical Design

## 3.1 Presentation layer

The presentation merely consists of static (x)HTML files that contain calls to Java Servlets to obtain dynamicly changing information. These (x)HTML files will use CSS to seperate the static information from the actual presentation of the website. Javascript will be used to provide the user with a friendly and smooth interface. This facilitates the following goals:

- Presentation is abstracted from (dynamicly changing) information, coming from the business layer

- Support for multiple languages (1 set of html templates per language), as required in the SRS

- 'Smooth' interface ith a minimum of required page refreshes, as required in the SRS

Prototypes of the webinterface can be found in Appendix B.

## 3.2 Business layer

The business layer consists of the Java Servlets that will be able to receive and process requests, and return proper responses. From the class diagram it becomes clear that two classes will play a central role in our application: User and Auction.

Class diagram, see Appendix A, 8.3

## 3.3 Data layer

The data layer consists of the database and the EJB3 standard for persistant objects. EJB3 allows, through the use of annotations in the class, to abstract away from SQL-statements as EJB3 will generate these. An alternative for the use of EJB3 would be Hibernate. Hibernate however, is not a standard, in contrary to EJB3. Futhermore, EJB3 is newer technology based on the concepts of Hibernate and is now seamlessly supported by the jBoss AS. Because of these reasons, Hibernate was discarded.

The database design supports all current requirements and is designed with expansion in mind.

Entity-Relationship diagram: see Appendix A, 8.4

# Chapter 4

# Software Features

## 4.1 Functional Requirements

### 4.1.1 Guest

**Create a new account**

**Requirement ID** 1

**Priority** Must have

**Actor** Guest

**Preconditions** User is not logged in

**Description** A guest can create a new account so that he or she can use the full functionality of the site

**Main path**   1. Guest selects *register*

2. Guest fills in *registration form*

3. Guest submits form

4. System checks form and if valid saves it

5. System sends a confirmation mail

**Exceptional paths**   4a  Username is already in use

4a1  System informs the user that the username is already taken

4a2  User returns to step 2, with the correct information still entered in the form field

4b  E-mail address is already in use

    4b1 System informs the user that the e-mail address is already in use and asks if the user is sure he or she wants to use this e-mail address

    4b2 The account is created or the user returns to step 2, with the correct information still entered in the form field

  4c Incorrect information in the registration form

    4c1 System informs the user that there is some incorrect information in the form

    4c2 User returns to step 2, with the correct information still entered in the form field

  4d Incomplete form

    4d1 System informs the user that there are some fields in the form that are not filled in

    4d2 User returns to step 2, with the correct information still entered in the form field

**Result** An account is created for the user

**Log in**

**Requirement ID** 2

**Priority** Must have

**Actor** Guest

**Preconditions** The user is registered on the site, user is not logged in

**Description** If a user is registered but not logged in, he or she can log in to use the full functionality of the site.

**Main path**    1. Guest fills in login form

    2. Guest submits form

    3. System checks if the form is valid and if so logs the user in

    4. System redirect the user to the page he or she was on before the log in

**Exceptional paths**    3a Incorrect username and/or password

    4a1 System informs the user that the he or she has entered an incorrect information

4a2　User returns to step 1

　　3b　Incomplete login form

　　　　4a1　System informs the user that some information is missing

　　　　4a2　User returns to step 1

**Result** The user is logged in

## 4.1.2　Unconfirmed user

**Confirm account**

**Requirement ID** 41

**Priority** Want to have

**Actor** Unconfirmed user

**Preconditions** User is at confirmation page

**Description** Unconfirmed users can confirm their account so they become
　　regular users

**Main path**　1. Users follows link in the confirmation mail he or she has
　　　　received

　　　2. User selects confirm account on the confirmation page

　　　3.

**Exceptions** None

**Result** The user becomes a regular user and can make use of all the func-
　　tionality of the site

**Request new confirmation mail**

**Requirement ID** 42

**Priority** Want to have

**Actor** Unconfirmed user

**Preconditions** User is at control panel

**Description** Unconfirmed users can request a new confirmation mail

**Main path**    1. Users selects "send new confirmation mail"

        2. System send a new confirmation mail to the user's e-mail address

        3.

**Exceptions**  None

**Result**  The user receives a new confirmation mail

### 4.1.3   Guest and User

**Search for an auction**

**Requirement ID**  3

**Priority**  Must have

**Actor**  User or Guest

**Preconditions**  User is at the home page

**Description**  Guests and members can search for auctions

**Main path**    1. Users enters a search term in the search field

        2. User submits the search

        3. System redirects the user to a page with all found auctions that match this search query

**Exceptions**  Empty search field

**Result**  A page with the found auctions for the search term

**View auctions of a certain category**

**Requirement ID**  4

**Priority**  Must have

**Actor**  User or Guest

**Preconditions**  User is at the home page

**Description**  Guests and users can view all the auctions of a certain category

**Main path**    1. Users select a category from the category list

2. System redirects the user to a page with all the auctions of the selected category

**Exceptions** None

**Result** A page displaying all the auctions of a certain category

**View auctions with a certain tag**

**Requirement ID** 5

**Priority** Want to have

**Description** Guests and users can view all the auctions with a certain tag

**Change language**

**Requirement ID** 6

**Priority** Must have

**Description** Guests and users can change the language of the website to one of the available languages

**Change currency**

**Requirement ID** 7

**Priority** Nice to have

**Description** Guests and users can change the currency in which auctions are displayed

### 4.1.4 User

**Log out**

**Requirement ID** 8

**Priority** Must have

**Actor** User

**Preconditions** User is logged in

**Description** Members who are logged in can log out

**Main path**   1. User selects *log out*

    2. System logs the user out

    3. System redirects the user to the site's homepage

**Exceptions** None

**Result** The user is logged out

**Search a user**

**Requirement ID** 9

**Priority** Want to have

**Actor** User

**Preconditions** User is at advanced search page

**Description** Members can search for other members

**Main path**   1. User enters search term in search user field

    2. User submits search

    3. System redirects the user to page containing all users corresponding to that search query

**Exceptions** Incorrect or incomplete form

**Result** A page displaying the found members for the search term

**Place an auction**

**Requirement ID** 10

**Priority** Must have

**Actor** User

**Preconditions** User is logged in. User is at home page or user home

**Description** Members of the site can create a new auction on which other members can bid

**Main path**   1. User selects *place auction*

    2. User fills in *new auction form*

3. User submits the form

4. System checks the form and if valid creates the auction

**Exceptions** Incorrect information in the form, incomplete form

**Result** Auction is placed

**View placed auctions**

**Requirement ID** 11

**Priority** Want to have

**Actor** User

**Preconditions** User is logged in. User is at control panel

**Description** Users can view the auctions they have placed

**Main path**   1. User selects *placed auctions*

2. System redirects user to user's placed auctions page

**Exceptions** None

**Result** The user views the auctions he or she has placed

**Bid on an auction**

**Requirement ID** 12

**Priority** Must have

**Actor** User

**Preconditions**   1. User is logged in

2. User is on an auction page

3. Auction is not of the user

**Description** Members can bid on auctions of other members

**Main path**   1. User select *bid*

2. User fills in bidding form

3. User submits form

4. System checks form and if valid places the bid

**Exceptions**  Wrong bid value in the form, incomplete form

**Result**  The bid is placed on the auction

**View active auctions**

**Requirement ID**  13

**Priority**  Want to have

**Actor**  User

**Preconditions**  User is logged in. User is at control panel

**Description**  Users can view the auctions they have bid on

**Main path**     1. User selects *active auctions*

        2. System redirects user to user's active auctions page

**Exceptions**  None

**Result**  The user can view the auctions he or she had bid on

**Modify account information**

**Requirement ID**  14

**Priority**  Must have

**Actor**  User

**Preconditions**  User is logged in. User is at control panel

**Description**  Members can modify their account information

**Main path**     1. User selects *modify account*

        2. User changes account information form

        3. User submits the form

        4. System checks the form and if valid, makes the changes

**Exceptions**  Incorrect information, incomplete form

**Result**  The account information of the user is changed

**Send a personal message**

**Requirement ID** 15

**Priority** Nice to have

**Actor** User

**Preconditions** User is logged in. User is at user page

**Description** Users can send messages to other users

**Main path**   1. User selects *send message*

2. User fills in personal message form

3. User submits the form

4. System checks the form and if valid, sends the message

**Exceptions** Incorrect information, incomplete form, users send message to him or herself

**Result** A message is sent to another user

**View personal messages**

**Requirement ID** 16

**Priority** Nice to have

**Actor** User

**Preconditions** User is logged in. User is at control panel

**Description** Users can view the personal messages

**Main path**   1. User selects *personal messages*

2. System redirects the user the user's personal messages page

**Exceptions** None

**Result** The user will view a page with his or her personal messages

**Delete a personal message**

**Requirement ID** 17

**Priority** Nice to have

**Actor** User

**Preconditions** User is logged in. User is at personal messages page

**Description** Users can delete personal messages they have received

**Main path**   1. User selects a message

2. User selects *delete message*

3. System deletes the message

**Exceptions** None

**Result** A message is deleted from the user's inbox

**Follow an auction**

**Requirement ID** 18

**Priority** Want to have

**Actor** User

**Preconditions** User is logged in. User is at auction page

**Description** Users can follow an auction, i.e. they put it in their following list

**Main path**   1. User selects *follow auction*

2. System adds auction to the users followed auctions list

**Exceptions** None

**Result** An auction is added to the user's follow auctions list

**View followed auctions**

**Requirement ID** 19

**Priority** Want to have

**Actor** User

**Preconditions** User is logged in. User is at control panel

**Description** Users can view the auctions they are following

**Main path**   1. User selects *followed auctions*

   2. System redirects user to user's followed auctions page

**Exceptions** None

**Result** The user can view the auctions he or she is following

**View Transaction**

**Requirement ID** 20

**Priority** Want to have

**Actor** User

**Preconditions** User is logged in. User is at auction page. User bought the item.

**Description** Users can view the transaction of an item they bought

**Main path**   1. User selects *view transaction*

   2. System forwards user to transaction page of auction

**Exceptions** None

**Result** The user views the transaction of the auction

**Pay transaction**

**Requirement ID** 21

**Priority** Want to have

**Actor** User

**Preconditions** User is logged in. User is at transaction page

**Description** Users can pay auctions on the transaction page of an auction

**Main path**     1. User selects *pay item*

      2. User selects a payment method

      3. User performs the payment

      4. System notifies the seller that the auction is paid for

**Exceptions** Incorrect information, incomplete form

**Result** The transaction is paid for

**Rate transaction**

**Requirement ID** 22

**Priority** Want to have

**Actor** User

**Preconditions** User is logged in. User is at transaction page, user paid transaction

**Description** Users can send rate a transaction after it is paid

**Main path**     1. User selects *rate transaction*

      2. User fills in rate transaction form

      3. User submits the form

      4. System checks the form and if valid, rates the transaction

      5. System updates the ratings of the user

**Exceptions** Incorrect information, incomplete form

**Result** The transaction is rated

**Add seller to favourites**

**Requirement ID** 23

**Priority** Nice to have

**Actor** User

**Preconditions** User is logged in. User is at user page

**Description** Users can add a seller to their favourites so they can their auctions easily

**Main path**   1. User selects *add seller to favourites*

2. System adds the seller to the favourite seller list of the user

**Exceptions** Seller is already in the favourite seller list

**Result** The seller is added to the favourite seller list of the user

**View favourite sellers**

**Requirement ID** 24

**Priority** Nice to have

**Actor** User

**Preconditions** User is logged in. User is at control panel

**Description** Users can view his or her favourite sellers

**Main path**   1. User selects *favourite sellers*

2. System redirects user to user's favourite sellers page

**Exceptions** None

**Result** The user will view a page with his or her favourite sellers

**Delete a favourite seller**

**Requirement ID** 25

**Priority** Nice to have

**Actor** User

**Preconditions** User is logged in. User is at control panel

**Description** Users can remove a seller from his or her favourite sellers list

**Main path**   1. User selects a seller from the list

      2. User selects *delete seller*

      3. System removes the selected user from the list

**Exceptions** None

**Result** A seller is removed from user's favourite sellers list

**View salespal**

**Requirement ID** 26

**Priority** Want to have

**Actor** User

**Preconditions** User is logged in. User is at control panel

**Description** Users can view their own personal "bank acount" on the site

**Main path**   1. User selects *view salespal*

      2. System redirects the user to the user's salespal page

**Exceptions** None

**Result** The user is on his or her salespal page

**Top up salespal**

**Requirement ID** 27

**Priority** Want to have

**Actor** User

**Preconditions** User is logged in. User is at salespal page

**Description** Users can add more money on their salespal account

**Main path**   1. User selects *top up salespal*
2. Users fills in top up salespal form
3. System checks the form and if valid tops up the account

**Exceptions** Incorrect or incomplete form

**Result** The user tops up his or her salespal account

**View recommended auctions**

**Requirement ID** 28

**Priority** Nice to have

**Actor** User

**Preconditions** User is logged in. User is at control panel

**Description** Users can view recommended auctions for him or her. This list is generated through tags and categories the user frequently uses

**Main path**   1. User selects *view recommendations*
2. System redirects user to user's recommended auctions page

**Exceptions** None

**Result** The user views a page with recommended auctions for the user

**View buyer's assistant**

**Requirement ID** 29

**Priority** Want to have

**Description** A user can check the auctions the buyer's assistant has found for his or her preferences

**Comment on an auction**

**Requirement ID** 38

**Priority** Want to have

**Description** A user can comment on an auction. This comment can be viewed by anyone viewing the auction page

## 4.1.5 Administrator

**Remove a user**

**Requirement ID** 30

**Priority** Must have

**Actor** Administrator

**Preconditions** User is at control panel

**Description** User selects *manage users* and selects a user from the users list. After the user is selected, *remove user* is selected.

**Exceptions** None

**Result** A user is removed from the system

**Remove an auction**

**Requirement ID** 31

**Priority** Must have

**Actor** Administrator

**Preconditions** User is at control panel

**Description** User selects *manage auctions* and selects an auction from the auctions list. After the auction is selected, *manage auction* is selected.

**Exceptions** None

**Result** Auction is removed

**Retract bid**

**Requirement ID** 32

**Priority** Want to have

**Description** An administrator can retract a bid from a user when that user has e.g. made a high bid because of a typo

## 4.1.6 Security

**Encrypted password**

**Requirement ID** 33

**Priority** Must have

**Description** Password must be stored encrypted

**CAPTCHA**

**Requirement ID** 34

**Priority** Must have

**Description** When a user registers, he or she has to fill in a CAPTCHA

**Limited login attempts**

**Requirement ID** 35

**Priority** Must have

**Description** A guest may only try to try to log in with a wrong password a fixed number of times

**Extra site for banned users**

**Requirement ID** 39

**Priority** Nice to have

**Description** When a banned user visits the site, he or she will view a special site for banned users

**Cookies**

**Requirement ID** 40

**Priority** Want to have

**Description** Cookies can be created for the user to let the user log in automatically

### 4.1.7 Other

**Basic e-mail notification**

**Requirement ID** 36

**Priority** Want to have

**Description** User can receive an e-mail when he or she has won an auction, when the item is shipped, when an auction is paid for

**Advanced e-mail notification**

**Requirement ID** 37

**Priority** Nice to have

**Description** User can receive an e-mail when he or she is overbid, when an auction is almost done, with recommended auctions for the user

## 4.2 Non-functional Requirements

### 4.2.1 Browser compatibility

**Requirement ID** 39

**Priority** Must have

**Description** The site should be fully compatible with the three most popular browsers. At the time of writing these are: Internet Explorer (version 7 or higher), Firefox (version 3 or higher) and Safari (version 4 or higher)[**?**].

### 4.2.2 Secure connection

**Requirement ID** 40

**Priority** Must have

**Description** Whenever sensitive transactions are made, the HTTPS protocol should be used.

### 4.2.3 Advertisments through Google AdWords

**Requirement ID** 43

**Priority** Want to have

**Description** There should be space for advertisements from Google AdWords

# Chapter 5

# User Interface Design

## 5.1 Web pages tree

### 5.1.1 Overview

(pagetree image)

### 5.1.2 Page Description

- index.jsp: welcome page of the website, invites guests to register, shows advertising, features auctions,..

- login.jsp: allows a guest to log in. Every page will feature a login-form, login.jsp will handle the business logic (check password etc)

- logout.jsp: destroys the current session, logs the member out

- auctionlist.jsp: shows both an overview of subcategories and a list of auctions within this category. It features a 'filter' form that allows to shrink the current auctionlist according to provided keywords ( search within current list). When no category is specified, the top-level category is shown.

- auctiondetail.jsp: shows details of the given auction. When no (or an illegal) auction is specified, an error is shown. It also features bidding-facilities, show a history of bids, an interface to view pictures of the items sold and comments added by the sellers of potential buyers.

- newauction.jsp: allows a member to add a new auction to the website. The provided form will request vital information like the auctionname, the duration of the auction, a description of the items sold, the required

minimum amount for which the item will be sold and facilities to upload pictures of the given item.

- profile.jsp: features the user control panel, to allow a member to change his personal information. It also allows a member to manage favorite sellers, the buyers assistant and followed auctions.

- search.jsp: provides powerful search-options using keywords, tags and (multiple) categories.

## 5.2  User Interface

Prototypes of the webinterface can be found in Appendix B.

# Chapter 6

# Additional Material

# Chapter 7

# Requirements Traceability Matrix

# Chapter 8

# Appendix A

## 8.1 Architecture



Figure 8.1: Salesmen Architecture

## 8.2   Infrastructure



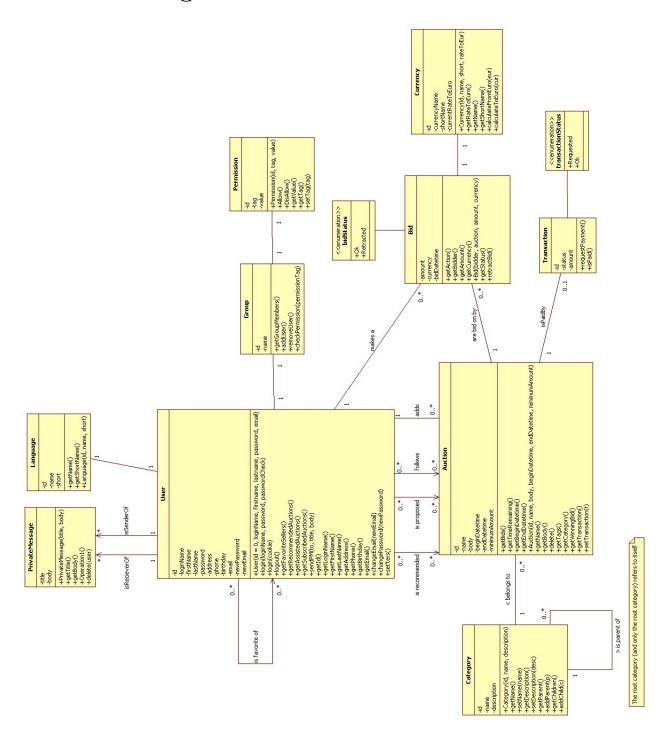Figure 8.2: Salesmen infrastructure

## 8.3 Class Diagram



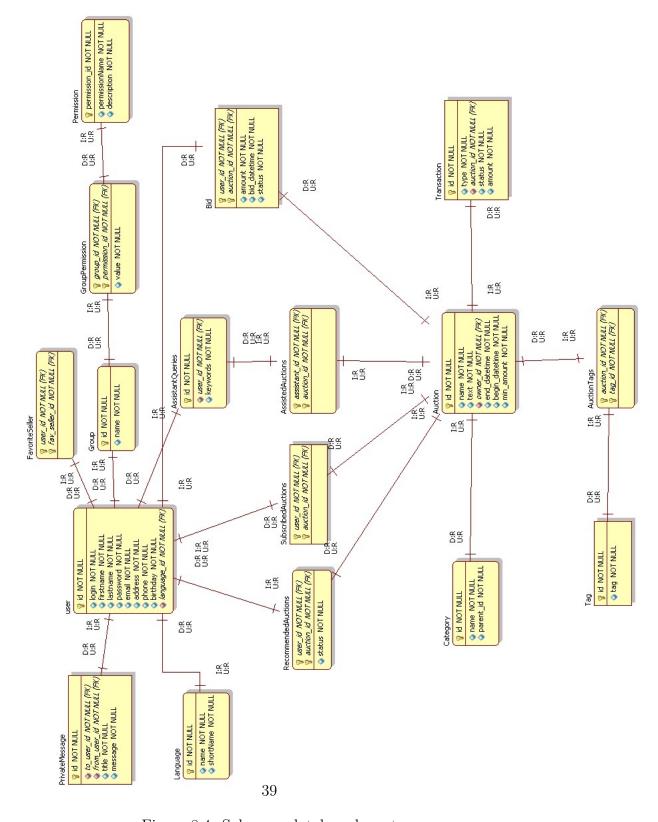Figure 8.3: Salesmen class diagram

## 8.4 Database diagram



Figure 8.4: Salesmen database layout
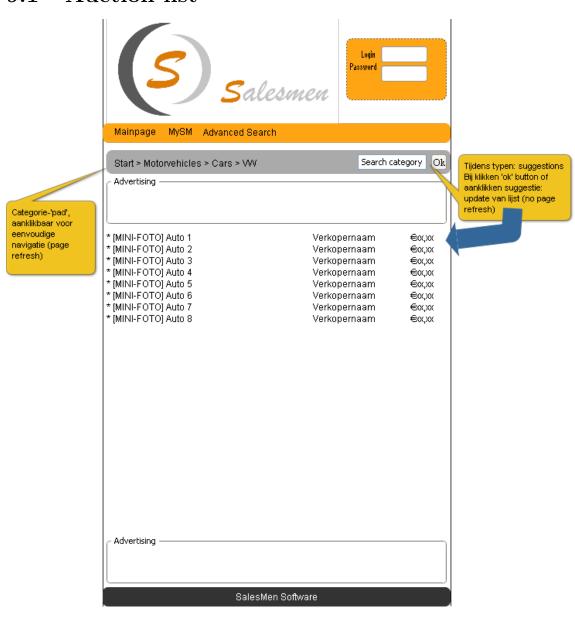
# Chapter 9

# Appendix B

## 9.1 Auction list



Figure 9.1: auctionlist.jsp

## 9.2   Auction detail



Figure 9.2: auctiondetail.jsp

## 9.3 Profile - Auctions



Figure 9.3: profile.jsp - auctions

## 9.4 Profile - Buyers assistant



Figure 9.4: profile.jsp - buyers assistant

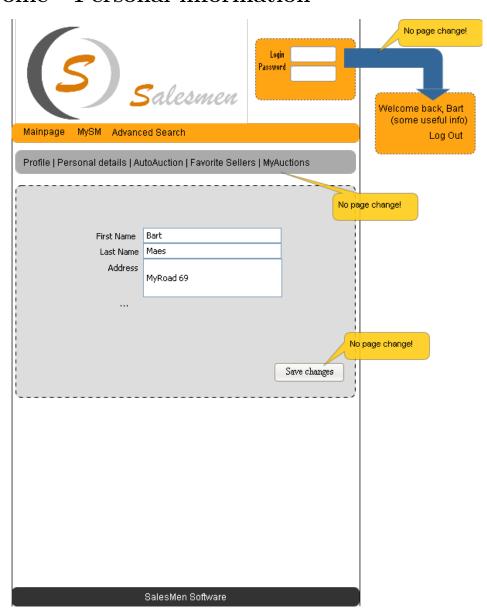## 9.5 Profile - Personal information



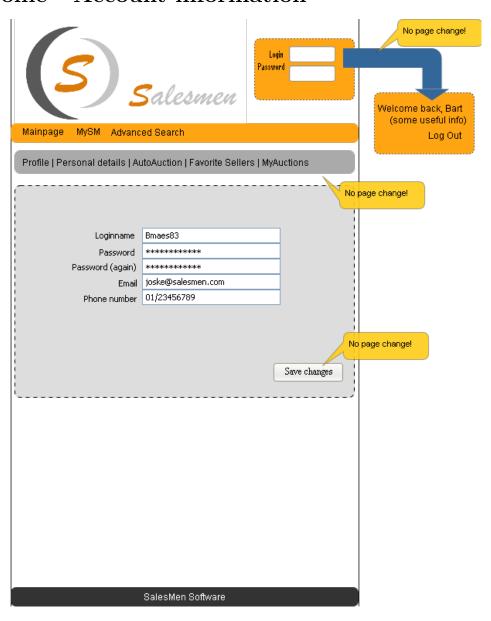Figure 9.5: profile.jsp - personal information

## 9.6 Profile - Account information



Figure 9.6: profile.jsp - website information