GUESSBID

Design documentation

Author: Nguyen Van Minh

May,13, 2015

CONTENTS

1 INTRODUCTION

1.1 PURPOSE

The design documentation is written on purpose of representing the architecture and detail design of Guess-bid project. During this, the reader can find out how the design organizes the project in terms of satisfying RASD document provided before.

1.2 SCOPE

As a normal software product, this project will have a design from high-level (sub-systems) to low-level (details in user interface, database models…).

Concerning RASD documentation, Guess-bid project provides general functionalities:

- Profiles:
  - Profile will be considered in some different ways as: guests, users, administrators.
- Auctions:
  - Users are likely to join in other users' auctions and manage their own auctions in the meantime.
- Bids:
  - Users are able to make some bids on an auctions along with the cost of each bids.
- Users:
  - Guess-bid project will manage the state logging in/logging out of any users.

1.3 DEFINITIONS AND ACRONYMS

Guess-bid project will use some definitions and acronyms accordingly.

1.3.1 Definitions

Keyword	Definition
Auction (considered as well as reverse auction in Guess-bid project)	is a process of selling and buying goods or services (abstract items in this project) by offering them up for bid, taking bids with the lowest unique bid.
Administrator	Super users who has right to manage other users

1.3.2 Acronyms

Acronym or abbreviation	Definition
JavaEE	Java Platform Enterprise Edition
EJB	Enterprise Java Bean
RASD	Requirements Analysis and Specification Document
JSP	Java Server Pages
JSF	Java Server Faces
FR	Functional Requirements

1.4 OVERVIEW

2 DESIGN OVERVIEW

This part will present the overall look on functionalities of software system as well as overall design

2.1 DESIGN CONTEXT

2.1.1 Functionalities

The following functionalities were identified by RASD. They are divided into groups.

2.1.1.1 Managing Profiles

Functional requirements

- [**FR1**]: View personal information
- [**FR2**]: Modify personal information

2.1.1.2 Managing Users

Functional requirements

- [**FR3**]: Register to the system
- [**FR4**]: Login
- [**FR5**]: Logout
- [**FR6**]: Modify password

2.1.1.3 Managing Auctions

Functional requirements

- [**FR7**]: Create auction
- [**FR8**]: Modify auction
- [**FR9**]: Close auction

- [**FR10**]: View auction
- [**FR11**]: View list of auctions
- [**FR12**]: Notify closed auction to users

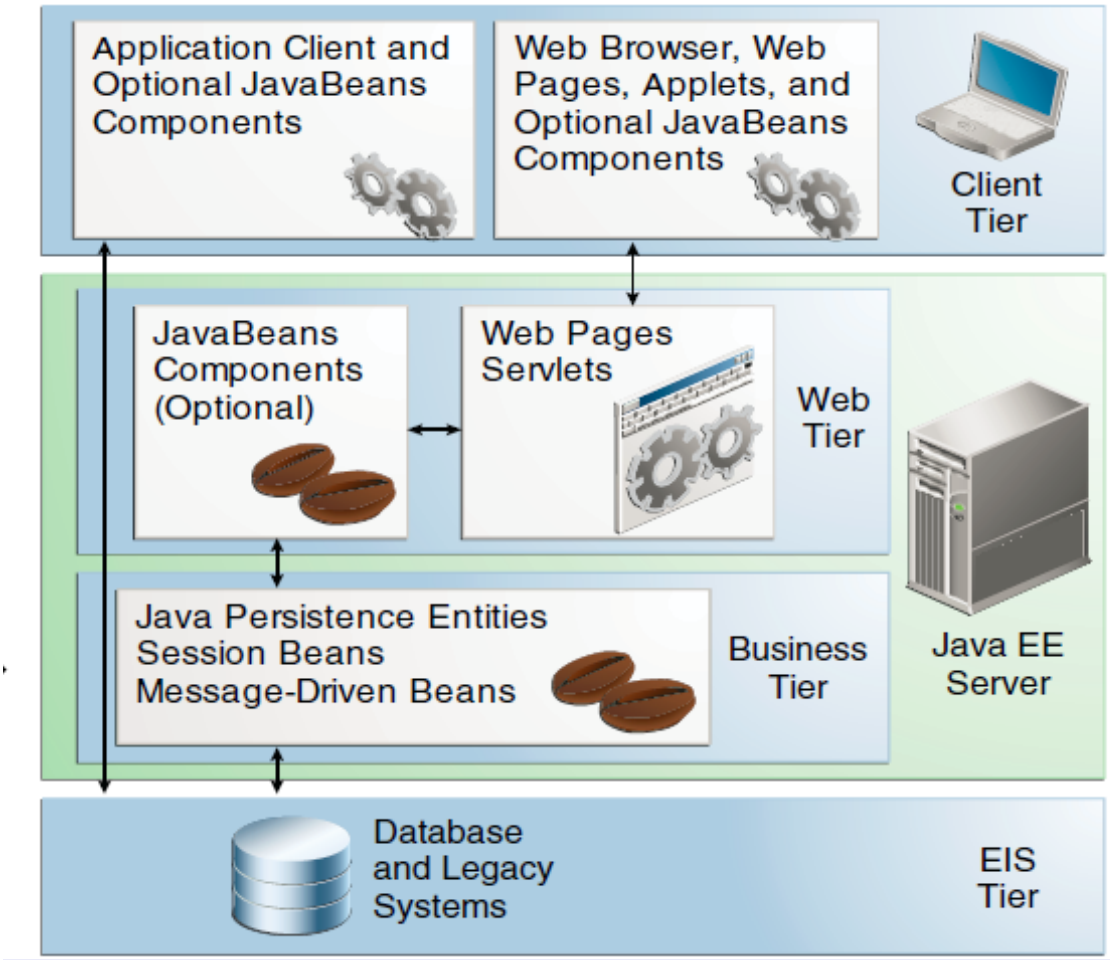
2.1.1.4 Managing Bids

Functional requirements

- [**FR13**]: View status of bids on an auction
- [**FR14**]: Do multiple bids on an auction

2.1.2 System technologies

This Guess-bid project goes along with multi-tier technologies of JavaEE/EJB application. Here is a short overview of system technologies:



This system architecture tends to give the developers the convenience to work with. The whole final software product can be divided into 4 part: client-tier, web-tier, business, EIS tier. Therefore, the system looks clean from the point of developer's view. After having a good design, I can work separately on individual components (business tier – apply functionalities, Web pages servlets – create dynamic interaction user interface).

2.2 GENERAL DESIGN DESCRIPTIONS

2.2.1 Design approach

The design approach is based on client-server 3-tier distributed system model. Each tier is described below:

- Client tier: This tier provides users friendly interface in order to intuitively work on. In this project, client tier is web interface.
- Business Logic tier: This tier coordinates the application, processes commands, does logical decisions and evaluations, and performs calculations. It also moves and processes data between the client and persistence tiers.
- Persistence tier: This tier holds the information of the system data model, and is in charge of storing and retrieving information from a database.

The design process followed a top-down process approach, so the outermost tiers were first identified and then broken into components that encapsulate the functionality.

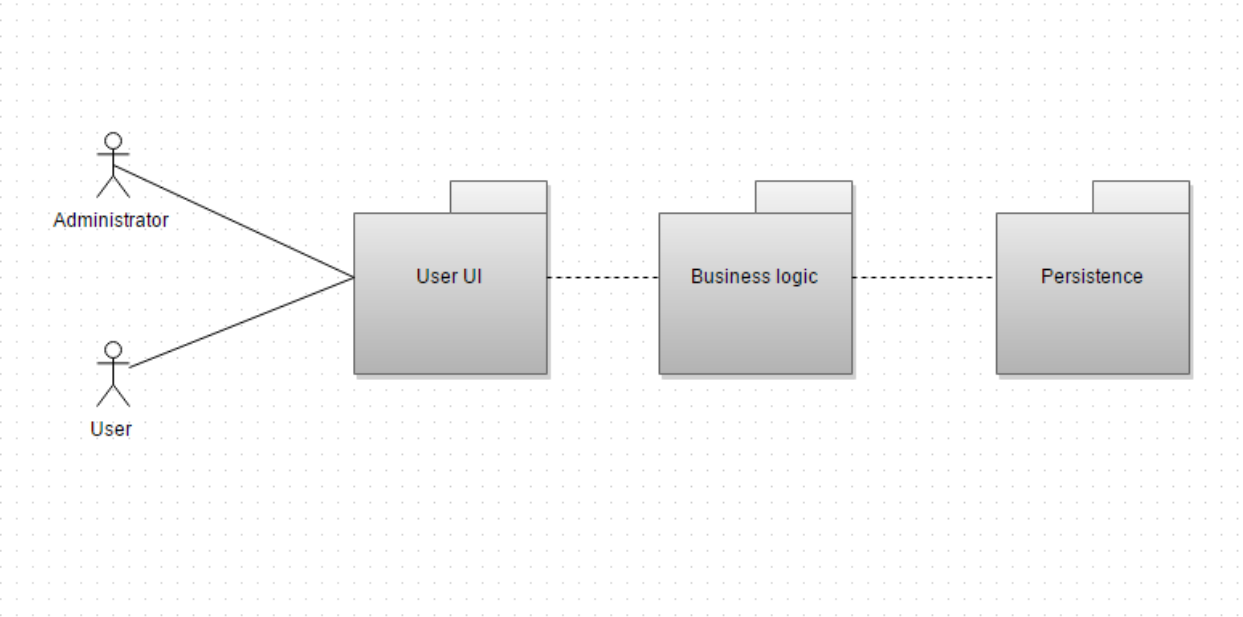
2.2.2 Overall design

Since each tier is broken into components and each component is responsible for a set of functionalities that fulfill the requirements. Here I point out 3 packages in relation to use cases

- User UI: This package contains the user interfaces. It is responsible for the interaction with the user such as getting UI requests, referring them to the Business Logic package and retrieving the data back for displaying.
- Business Logic: This package contains the business logic components. This package is responsible for handling the User UI package requests, processing them and accessing the Persistence package if required to provide a response.
- Persistence: This package is responsible for managing the data requests from the Business Logic package.

Guess bid project 2015

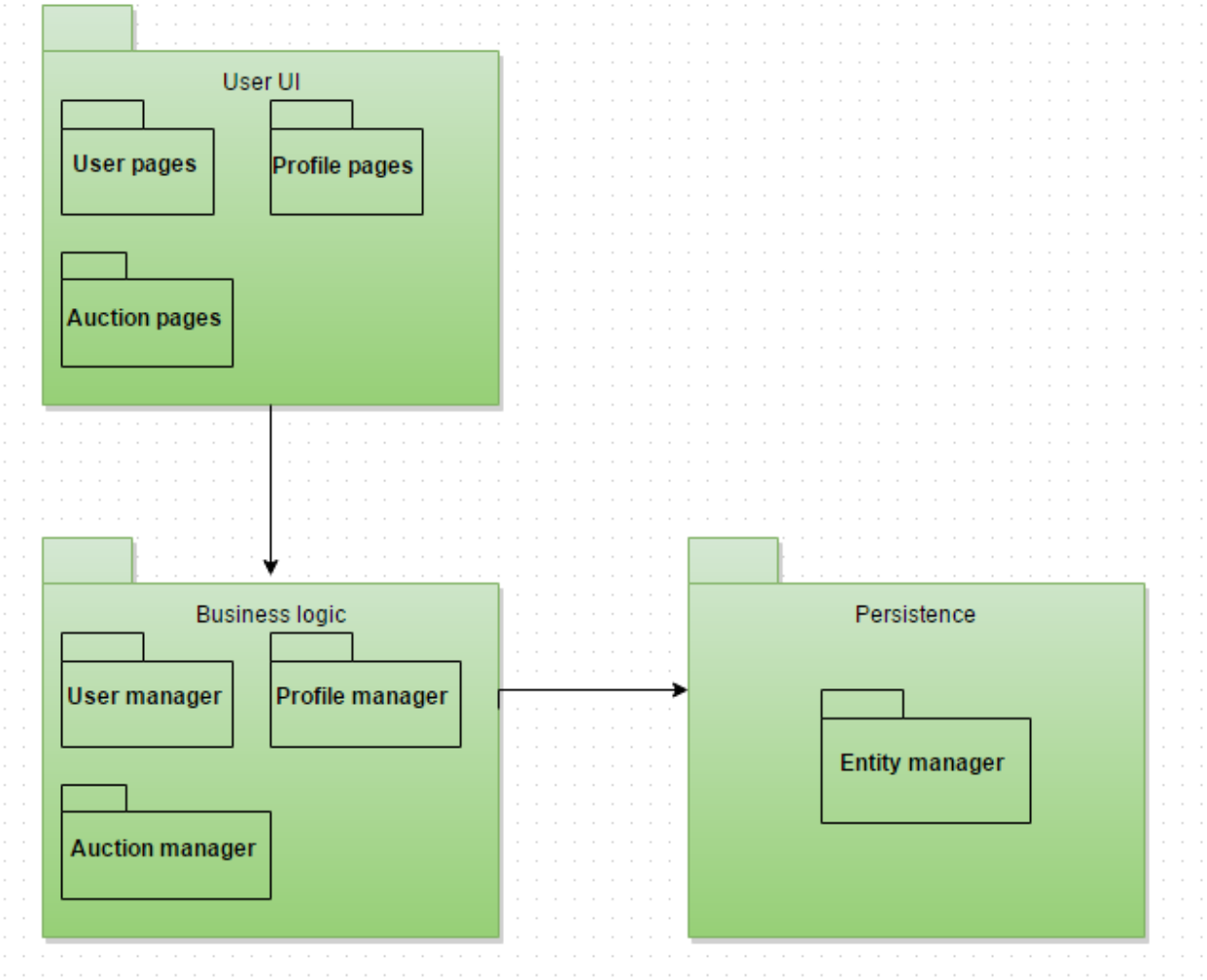
Basic package can be drawn as picture below



We hereby intuitively get the overall view of our basic design.

In the next step, I divide each basic package into several sub-packages:

- User UI: User pages, Profile pages, Auction pages
- Business logic: User manager, Profile manager, Auction manager
- Persistence: Entity manager



3 DESIGN CONSIDERATIONS

3.1 DEPENDENCIES

Dependency	Impact
Java virtual machine that supports JEE7 is already installed on OS.	GuessBid only runs on OS that support the JEE7 platform.
The supported browsers will be Firefox and Chrome	GuessBid web-client XHTML code that require mostly standard browsers.

3.2 GENERAL CONSTRAINTS

This section shows the requirement to ensure installed software will run properly as an application server.

Element	Requirement
Memory	2Gb RAM
Database server	MySQL
Network	Internet Access
Hard drive	10Gb

3.3 PERFORMANCE REQUIREMENTS

3.3.1 Standard compliance

This software product does not have to meet any standard compliance.

3.3.2 Reliability

It is mandatory to back up database to assure the reliability of this software product.

3.3.3 Availability

An application server is used to guarantee availability of the software product.

3.3.4 Security

This software offers the hashing of password according to SHAL algorithm in database. Thereby anyone who even have the rights to access to the database can't decode the password of user.

3.3.5 Maintainability

This design tries to both separate and divide package into components, sub-components, and use interfaces to interact between each sub components. It contributes to low coupling and high cohesion of the software product.

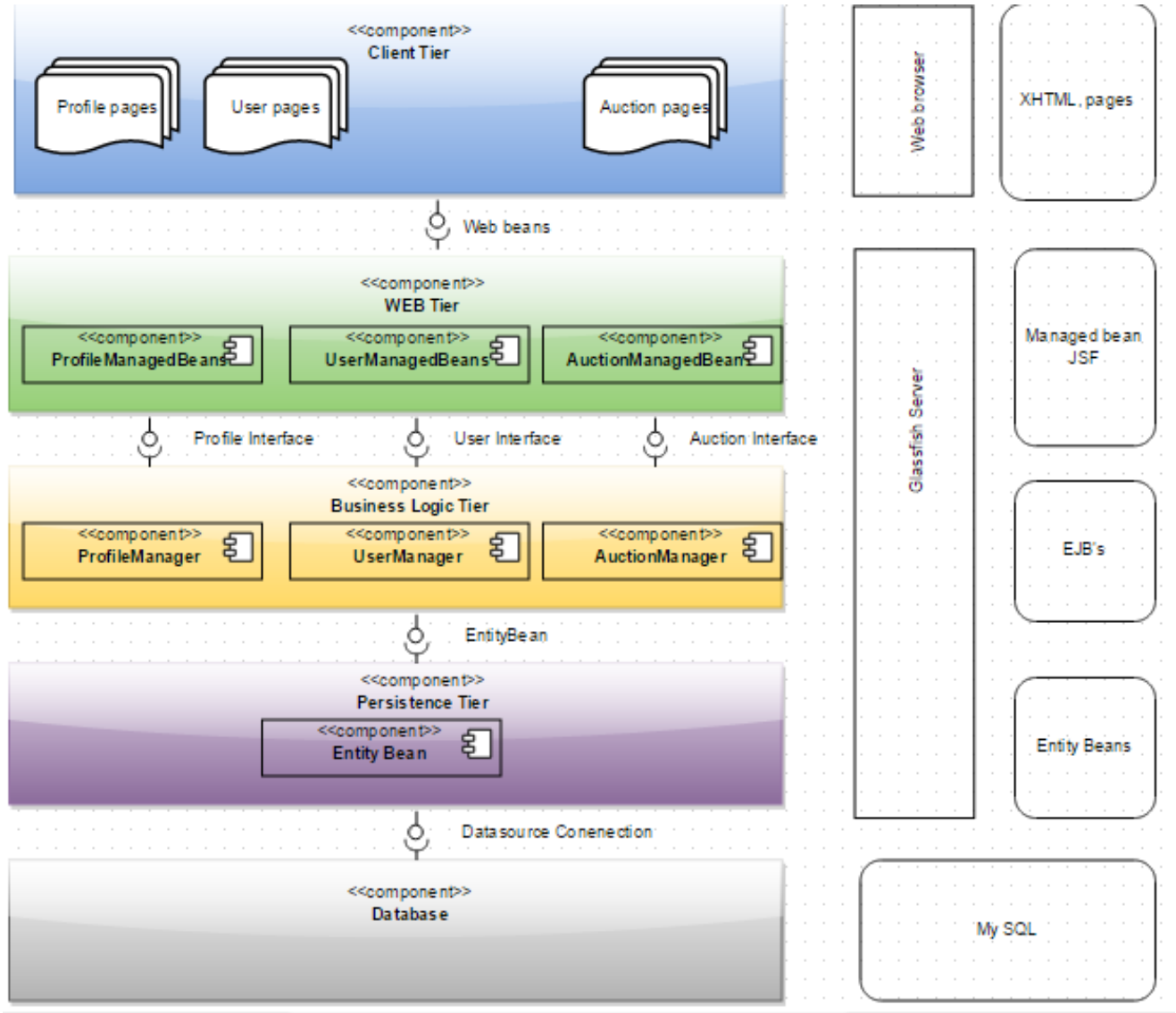
3.3.6 Portability

Java platform has an advantage that after installing on OS, application run over the platform independently.

4 SOFTWARE ARCHITECTURE

As aforementioned, GuessBid project uses 3 tier architecture to build up the design. So the Client-tier and Web-tier are considered as Web components, the remaining components are Business logic tier, Persistence tier (this one connects directly to database by JEE functionality).

4.1 CONCEPTUAL DESIGN



4.1.1 Client tier

This tier is constituted by XHTML pages that user can see. It strictly relates to Web tier.

4.1.2 Web tier

This tier is composed of beans which receive user request and deal with BusinessLogicTier before sending the appropriate respond to user.

4.1.3 Business logic tier

This tier is basically the logic of this application where most of requirements, rules will be applied into functions.

4.1.4 Persistence tier

This tier is composed of beans which are entities. These entities should be adapted to RASD and intuitively related to database entities.

4.1.5 Database

This tier represents the real entities (tables, indexes) need for software product.

4.2 SYSTEM SPECIFICATION

The following table shows the technologies I use in Guess-bid project

Component name	Technology
Client – Web tier	XHTML, JSF, JSP
Business Tier	JEE with Glassfish Server
Persistence Tier – Database	MySQL

5 DETAIL SOFTWARE DESIGN

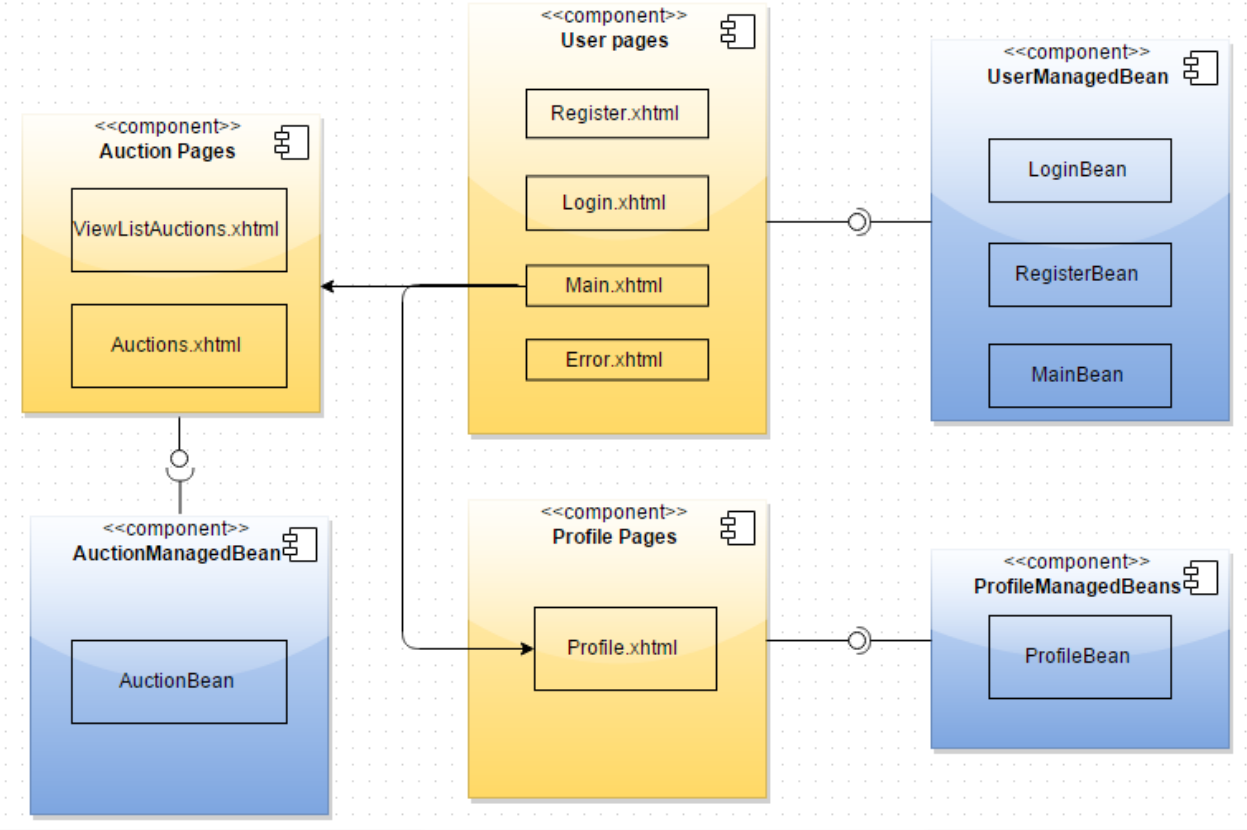
Here comes the detail of software design on implementation, database, runtime view…

5.1 IMPLEMENTATION MODULES/COMPONENTS

This project considers 3 main components:

- Web component
- Business logic component
- Persistence component

5.1.1 Web components



Component name	
Classification	Login.xhtml
Definition	User interface for user login
Responsibilities	• Display login form

Component name	
Classification	Register.xhtml
Definition	User interface for user registration
Responsibilities	• Load registration form

Component name	
Classification	Main.xhtml
Definition	User interface for showing main menu for user
Responsibilities	• Display user main homepage • Display list of auctions on his/her own • Display list of auctions of other users • Display admin main homepage

Component name	
Classification	Error.xhtml
Definition	User interface for displaying warning message
Responsibilities	• Display warning message for improper input

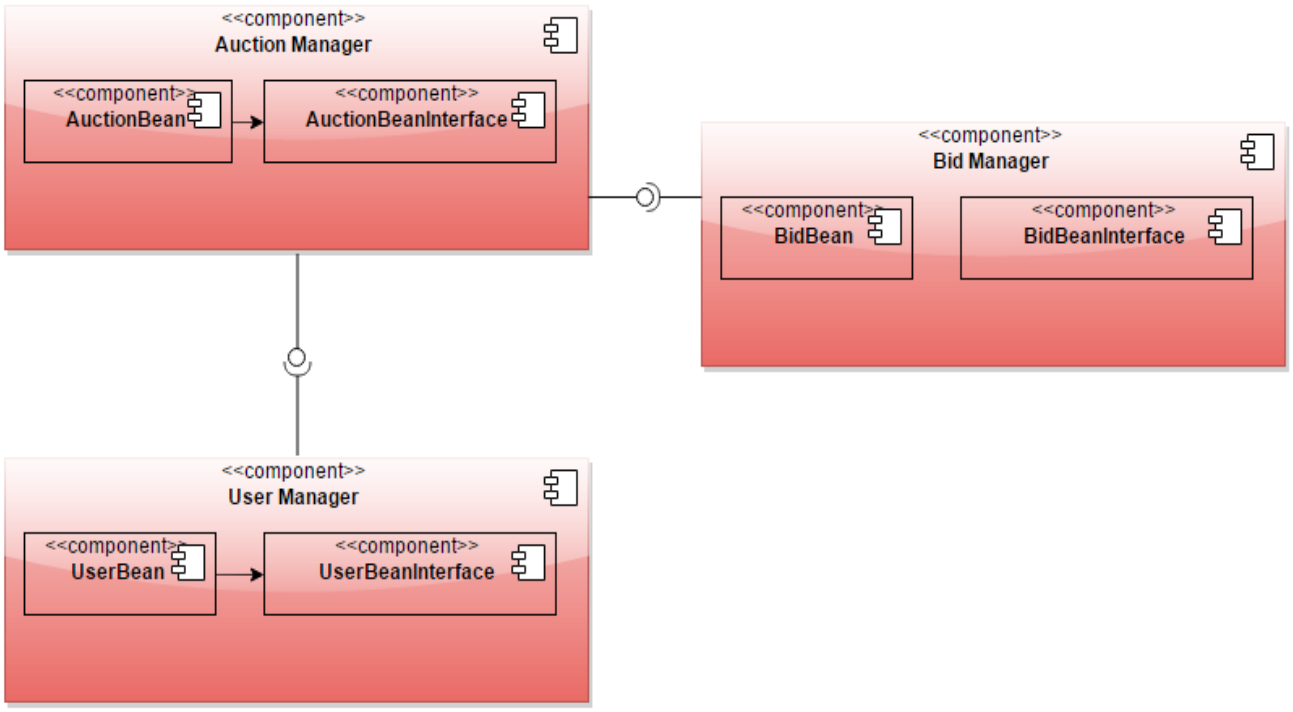
Component name	
Classification	Profile.xhtml
Definition	User interface for showing profile for user
Responsibilities	• Display profile user page

Component name	
Classification	ViewListAuction.xhtml
Definition	User interface for showing list of auctions (of current user and another ones)
Responsibilities	• Display list of auctions which can be divided into 2 sections: belong to current user, belong to another users

Component name	
Classification	Auction.xhtml
Definition	User interface for displaying all information in an auction
Responsibilities	• Display information of an auction: title, description, due date, current status of bid • Allow owner to create, modify, cancel

Component name	
Classification	Auction.xhtml
Definition	User interface for displaying all information in an auction
Responsibilities	• Display information of an auction: title, description, due date, current status of bid • Allow owner to create, modify, cancel

5.1.2 Business logic components



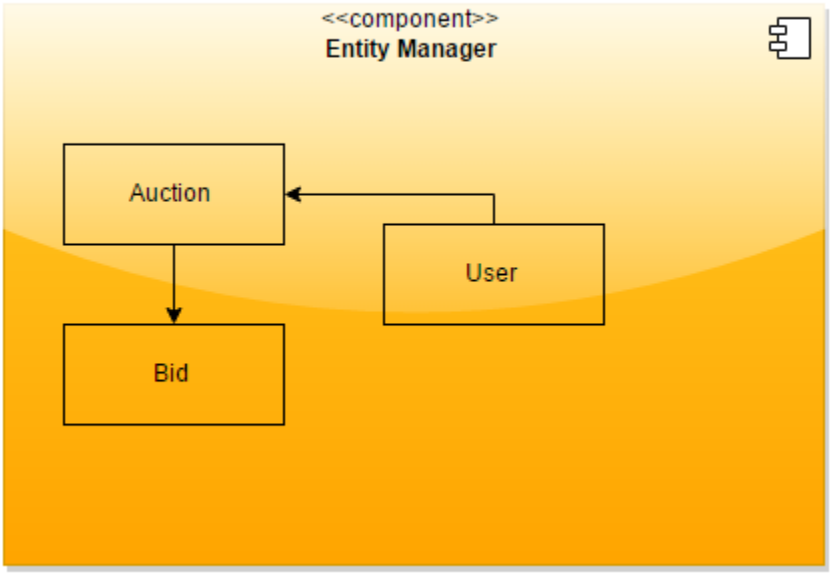
Component name	
Classification	UserBeanInterface
Definition	Interface instantiated every time communication between beans is needed.
Responsibilities	• Communicate with AuctionBeanInterface • Allow owner to create, modify, cancel
Component name	
Classification	UserBean
Definition	Beans in charge of all functionalities related to users.
Responsibilities	• Login • Logout • Register • Load profile information

Component name	
Classification	AuctionBeanInterface
Definition	Interface instantiated every time communication between beans is needed.

Responsibilities	• Communicate with UserBeanInterface, BidBeanInterface

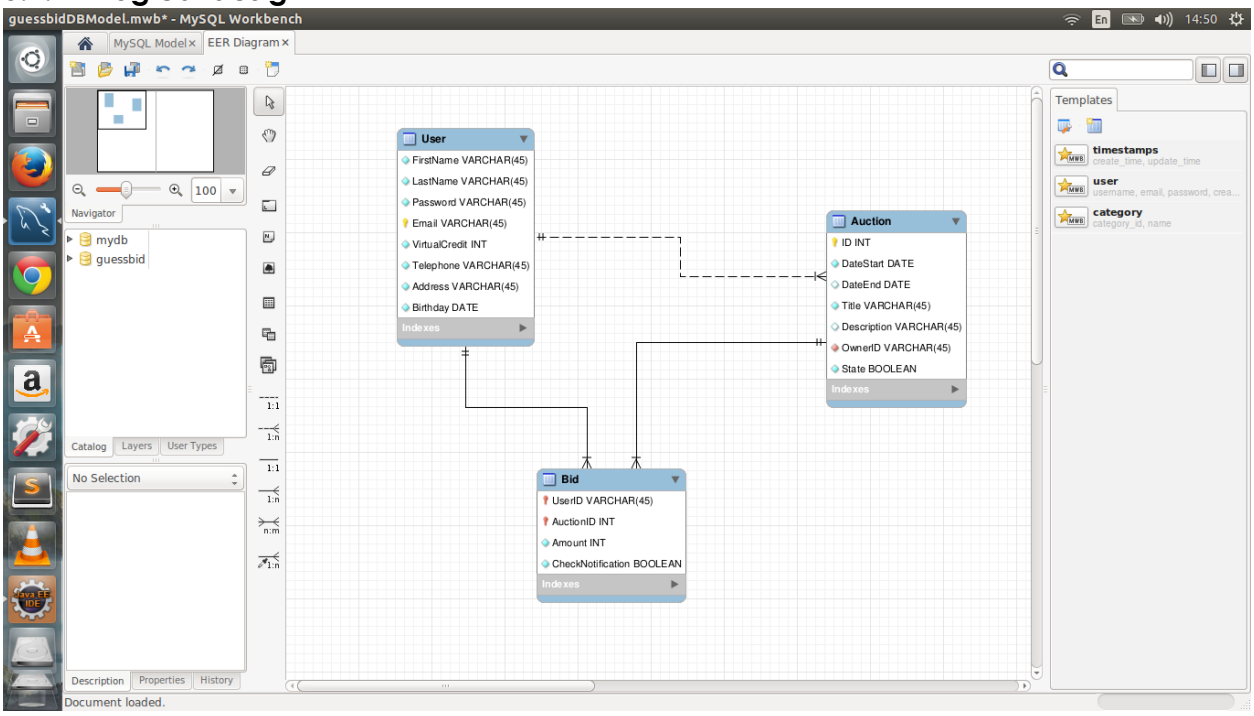
Component name	
Classification	AuctionBean
Definition	Beans in charge of all functionalities related to auctions.
Responsibilities	• List • View • Edit • Create • Cancel

5.1.3 Persistence components



5.2 DATABASE MODEL

5.2.1 Logical design



5.3 RUNTIME VIEW

5.4 DEPLOYMENT VIEW

5.5 MODULE VIEW