POLITECNICO DI MILANO

DEPARTMENT OF ELECTRONICS, INFORMATION AND BIOENGINEERING

SOFTWARE ENGINEERING II

Professor Elisabetta Di Nitto

Professor Mirandola Raffael

"GuessBid" Project

Design Document

Authors:

-Nery, Abel Sebsebe (816863)

-Beshir, Addisalem Wondie (816955)

May 15, 2015

Table of Contents

Diagrams

1 INTRODUCTION

1.1 Purpose

This purpose is attained through a detailed study of the system requirement and a proposed solution as specified in the RAD. To meet the proposed system this system design document will provide a detailed design of the software structure, software components, interface interaction, and type of data. This document will also give insight on how the system will provide a mechanism for storing, retrieving and exchanging basic data within the system component.

1.2 Definitions, acronyms and abbreviations

Term	Definition
Buyer_ID	Buyer Identification
Seller_ID	Seller Identification
Good_ID	Good Identification (Product Identification)
Bid_ID	Bidding Identification
Auction_ID	Auction Identification
Initial_Price	The starting price for bidding on a specific auction of association good/product
ER	Entity Relationship

1.3 Reference

- DD Example from past year.pdf
- Software Engineering II Course Materials
- http://www.tutorialspoint.com/dbms/

- http://dev.mysql.com/
- http://www.learndatamodeling.com/

1.4 Overview of Document

This Design document is organized of three parts:

- The First part is the Introduction Part of the design document.
- The Second Part covers the database design which includes identification of entities with their attributes, and the relationship between each entity, that is shown by depicting entity relationship diagrams and a conceptual database diagram. Following the ER diagram, we have presented the translated logical schema. The GuessBid web Application Database Tables are also included in this Document.
- The Third Part includes the system Architecture Design. It specifies the design entities that collaborate to perform the functionality of the system. The Component Diagrams, Deployment Diagram that shows the physical nodes on which the system resides are also included in the third part of this document. This allows a clear explanation of where each design entity will reside. And at the last part we have included the Assumption we made.

2 DATABASE DESIGN

2.1 Entity Relationship (ER) Diagram

In our system, the main entities are: BUYER, SELLER, GOOD, AUCTION, and BID.

Sellers are allowed to create, modify and delete auctions for each list of goods they have already created. Buyers can participate on auctions by selecting goods and guessing and providing the lowest bid value. Both users can receive auction notification such as the winner of a specific auction so as to confirm the purchasing process and payment completion.

Entities and Attributes

To minimize the complication of the ER Diagram, we have drawn a diagram for each entity with their attributes and we have included only primary keys of attributes in our ER Diagram.

Buyer Entity

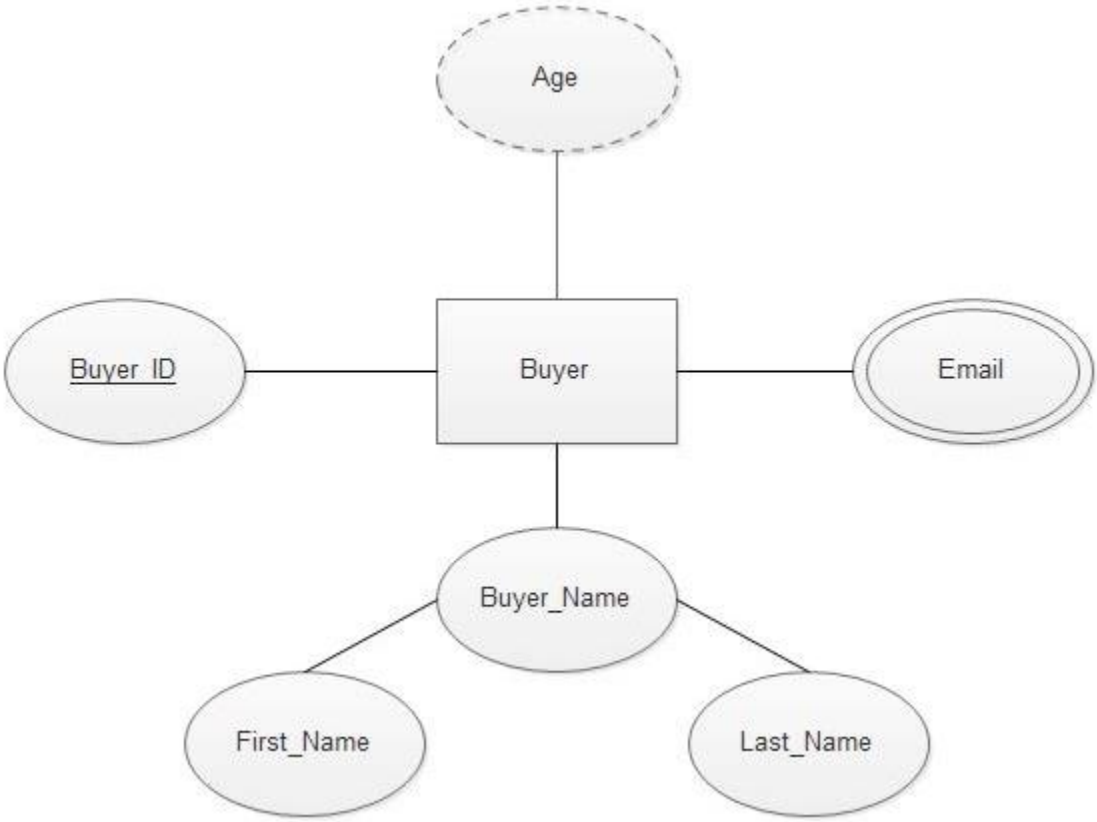


Figure 1 Buyer Entity diagram

Seller Entity

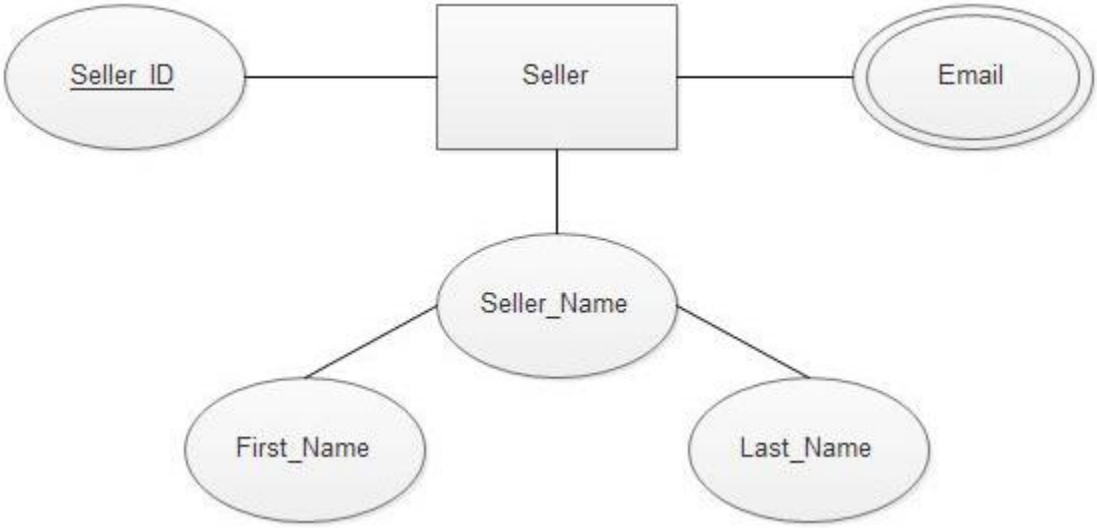


Figure 2 Seller Entity Diagram

Good Entity

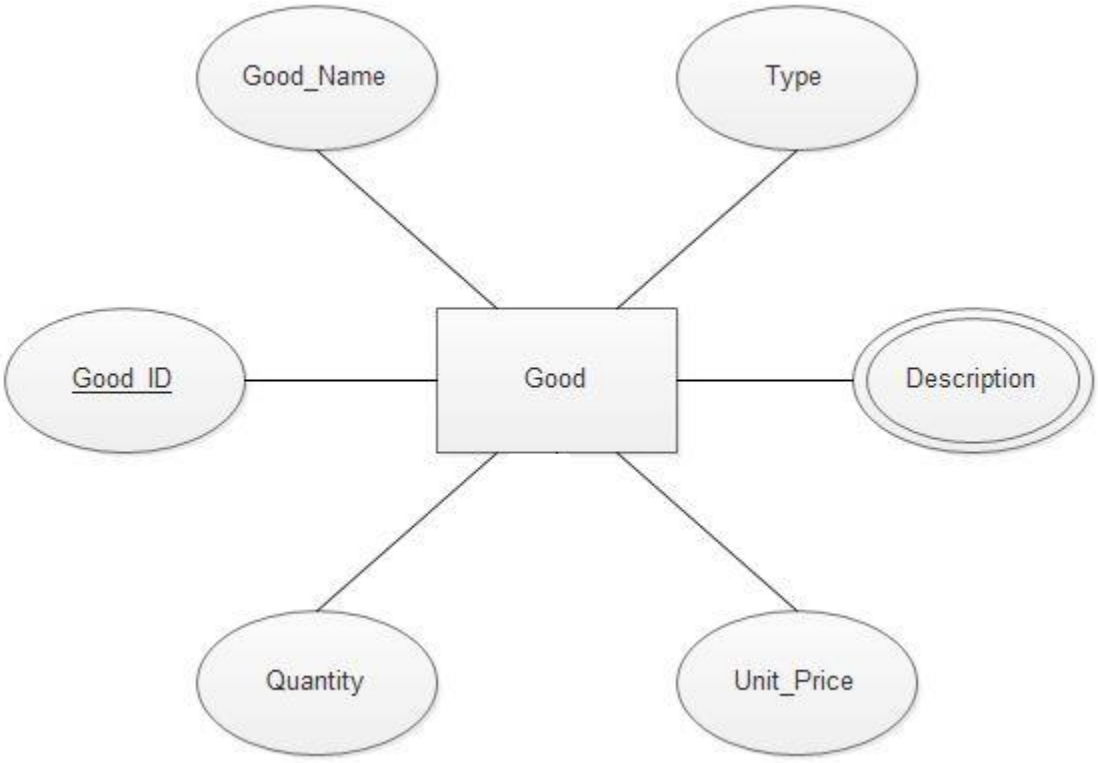


Figure 3 Good Entity Diagram

Auction Entity

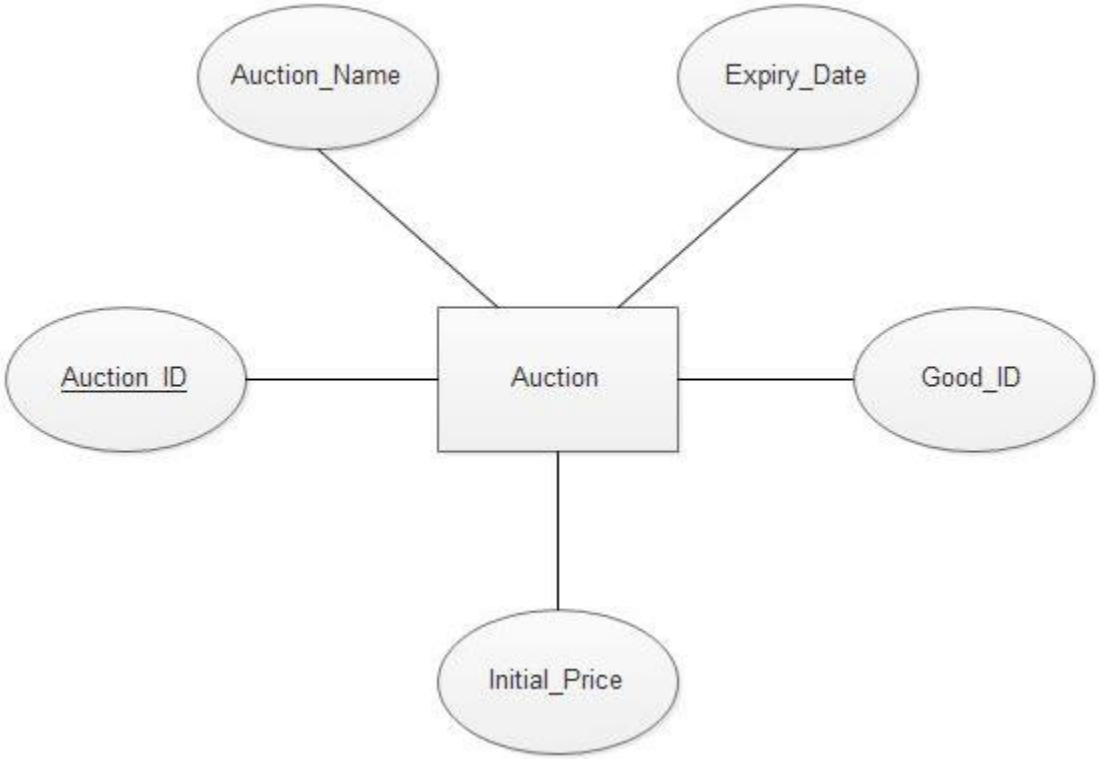


Figure 4 Auction Entity Diagram

Bid Entity

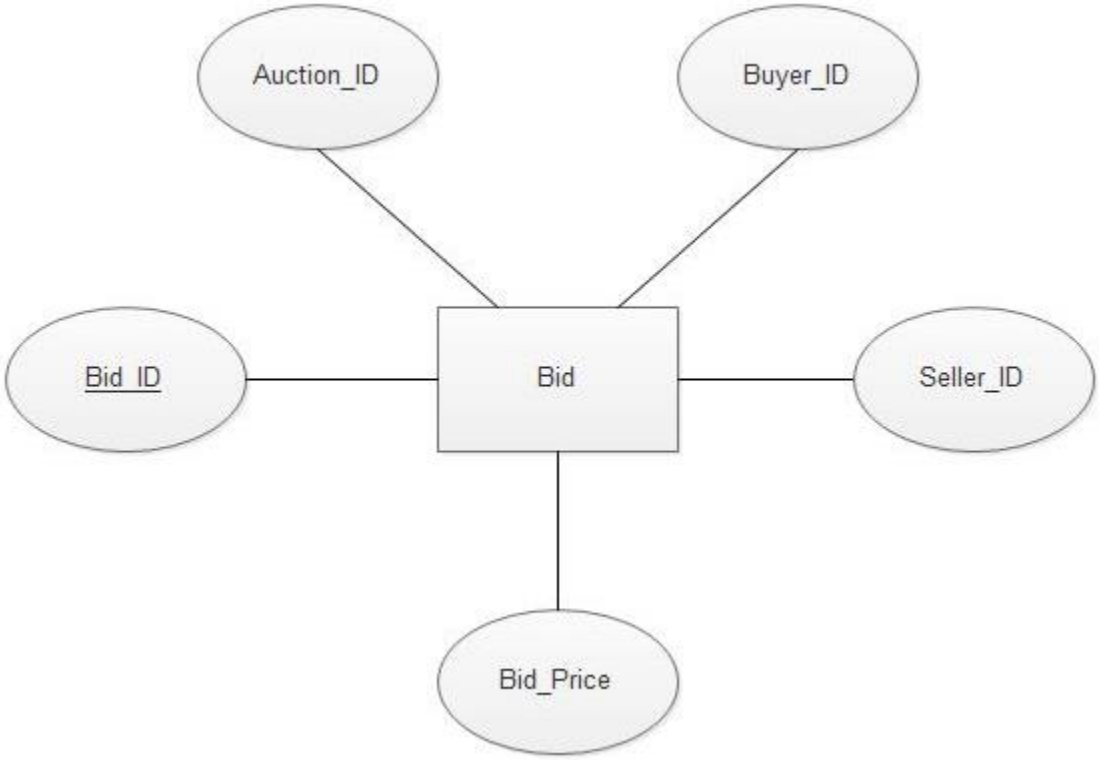


Figure 5 Bid Entity Diagram

The ER-Diagram

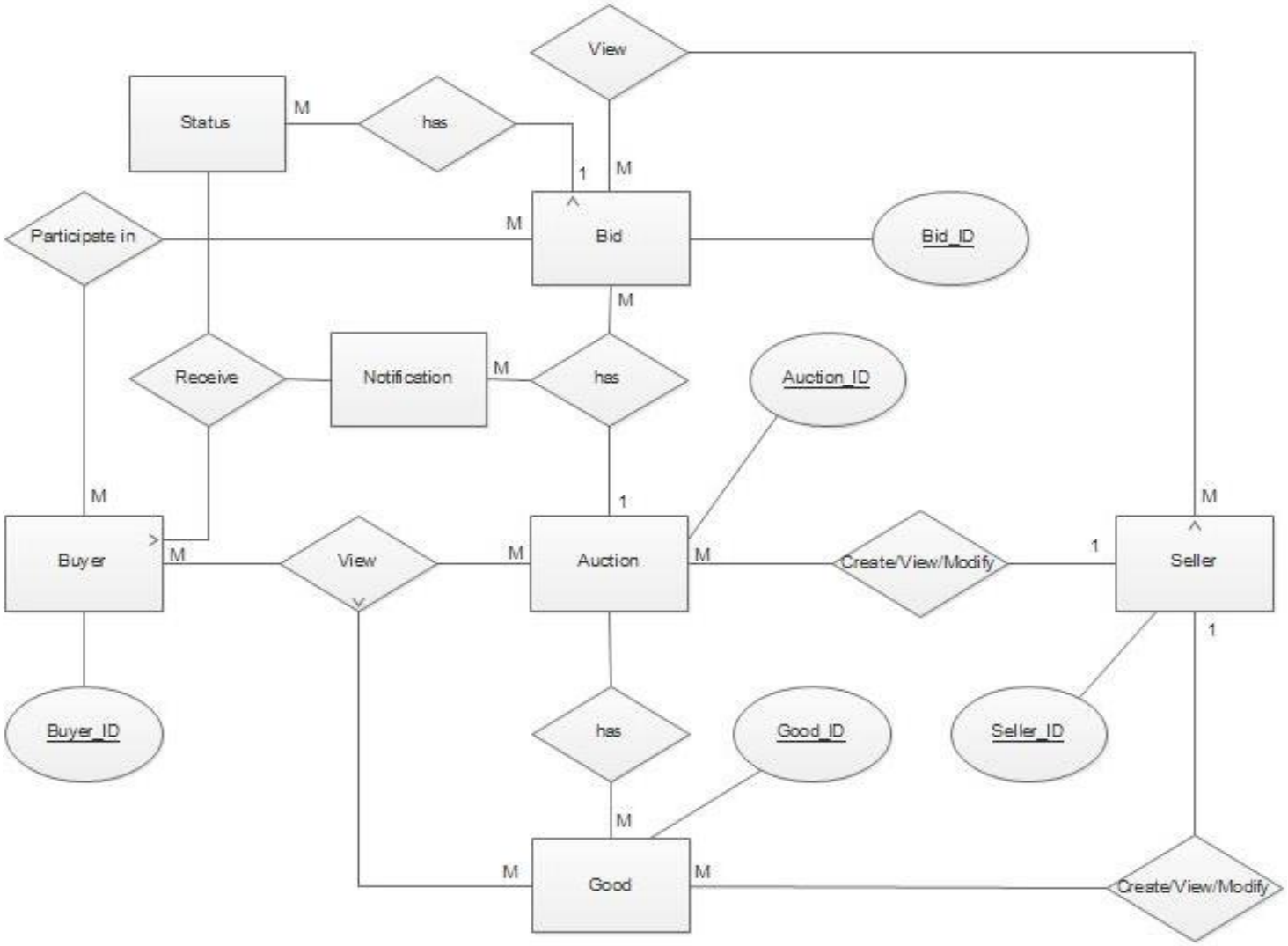


Figure 6 System ER-Diagram

In order to translate a logical conceptual model is necessary in a restructuring that involves the resolution of generalizations and processing entities (with associated links) of some relationships, according to databases principles.

The changes applied to the ER scheme are as follows:

- According to the translation rules all relationships, with multiple cardinality in both directions have been converted into tables.
- For all entities we added unique identifier that represents the primary key. This decision proves to be necessary in some cases in which no combination of attributes is a key demand, as in the case of the entity, it was decided to apply it to any other entity for easier management and greater uniformity.

2.2 Translated Logical Schema

BUYER (Buyer_ID, First_Name, Last_Name, Age, Email)

SELLER (Seller_ID, First_Name, Last_Name, Email)

GOOD (Good_ID, Good_Name, Type, Unit_Price, Quantity, Description, Picture)

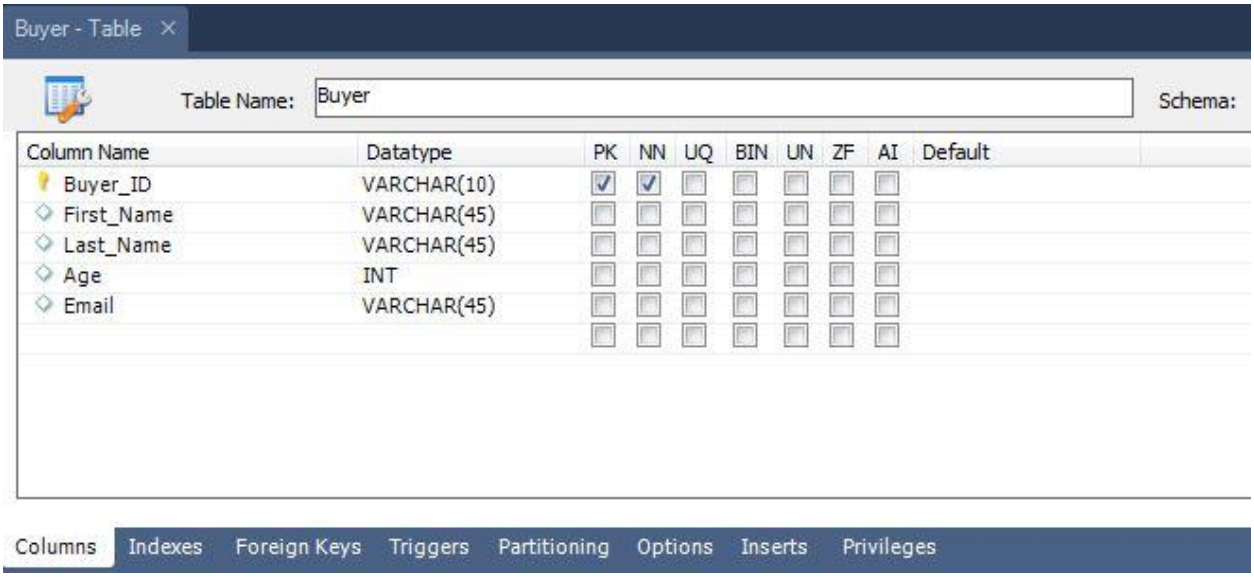
AUCTION (Auction_ID, Auction_Name, Initial_Price, Expiry_Date, Good_ID)

BID (Bid_ID, Bid_Price, Auction_ID, Buyer_ID, Seller_ID)

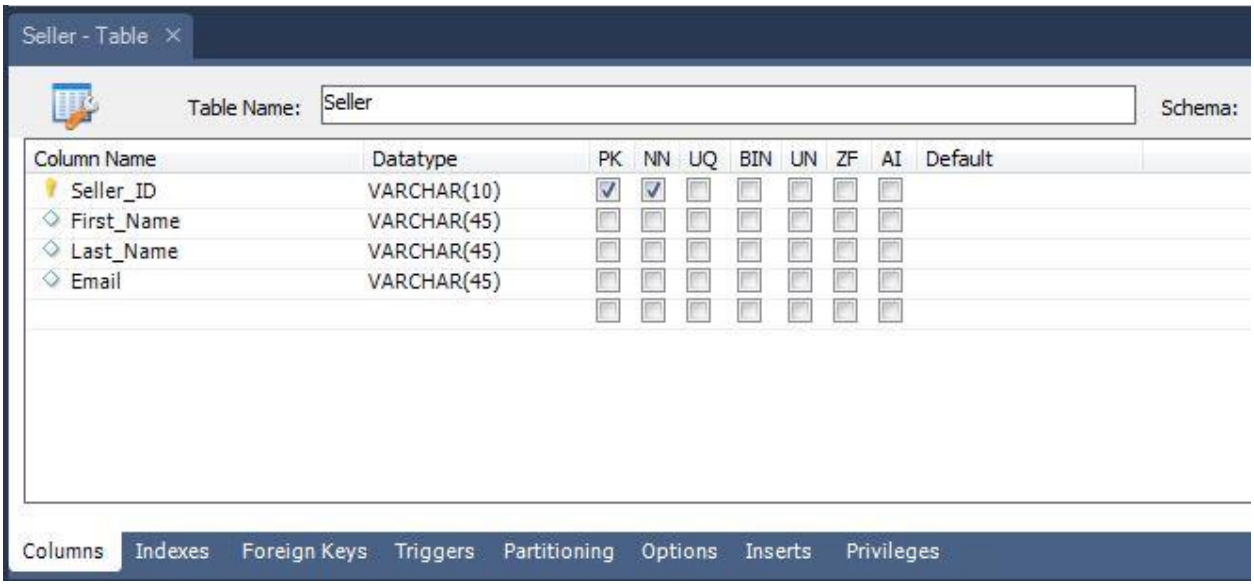
2.3 Database Tables

The Database tables designed for this system are shown here below:

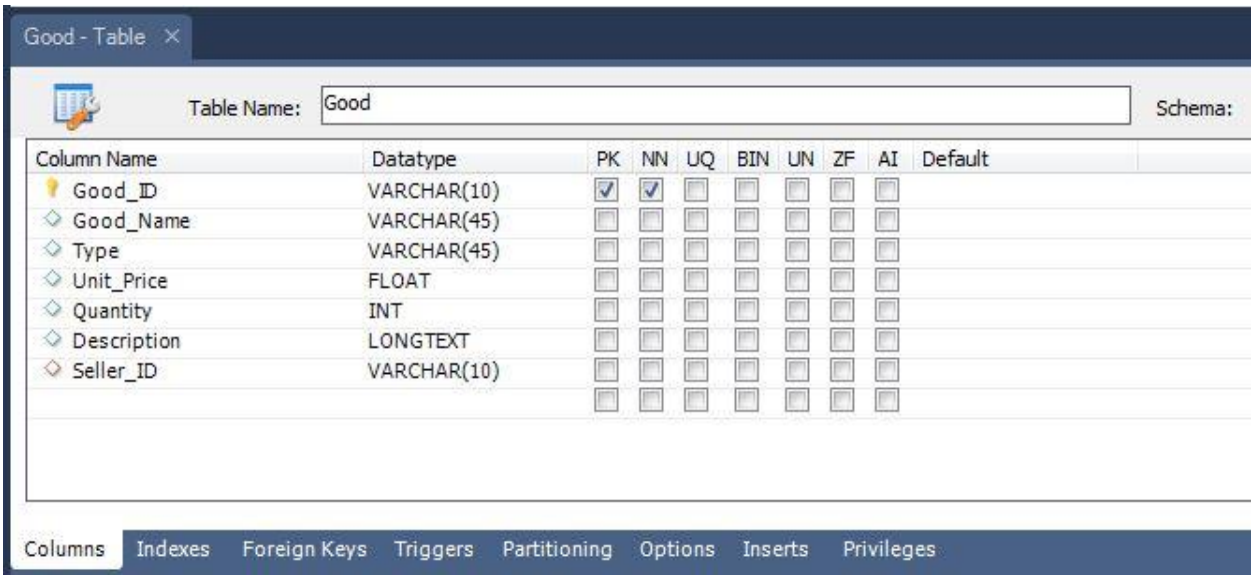
Buyer Table



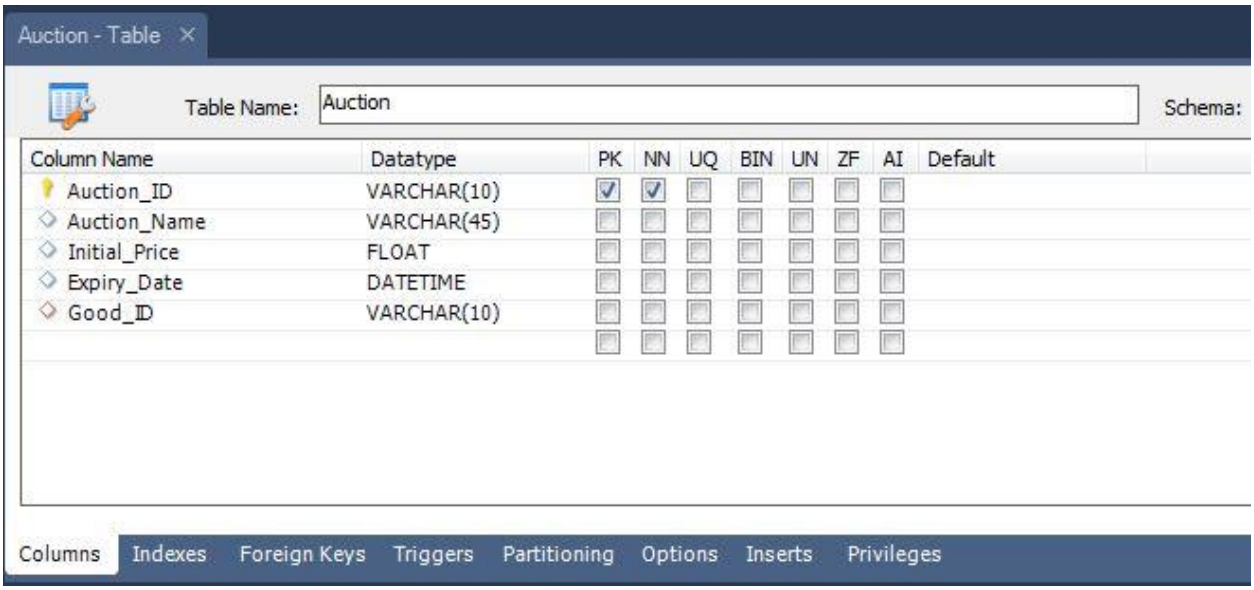
Seller Table



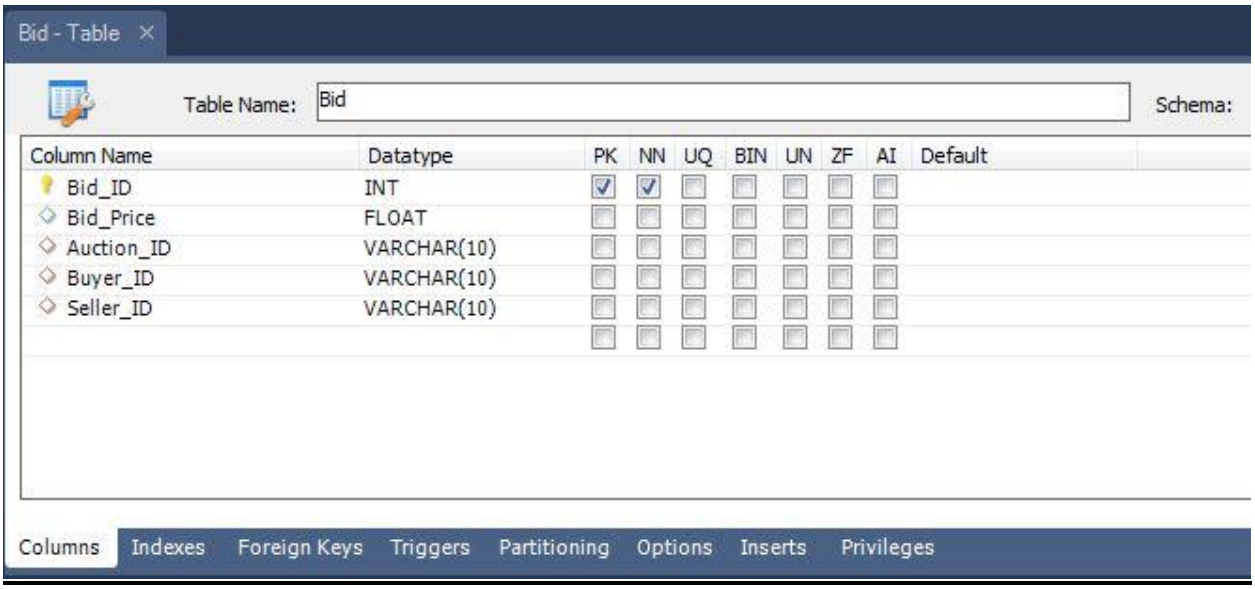
Good Table



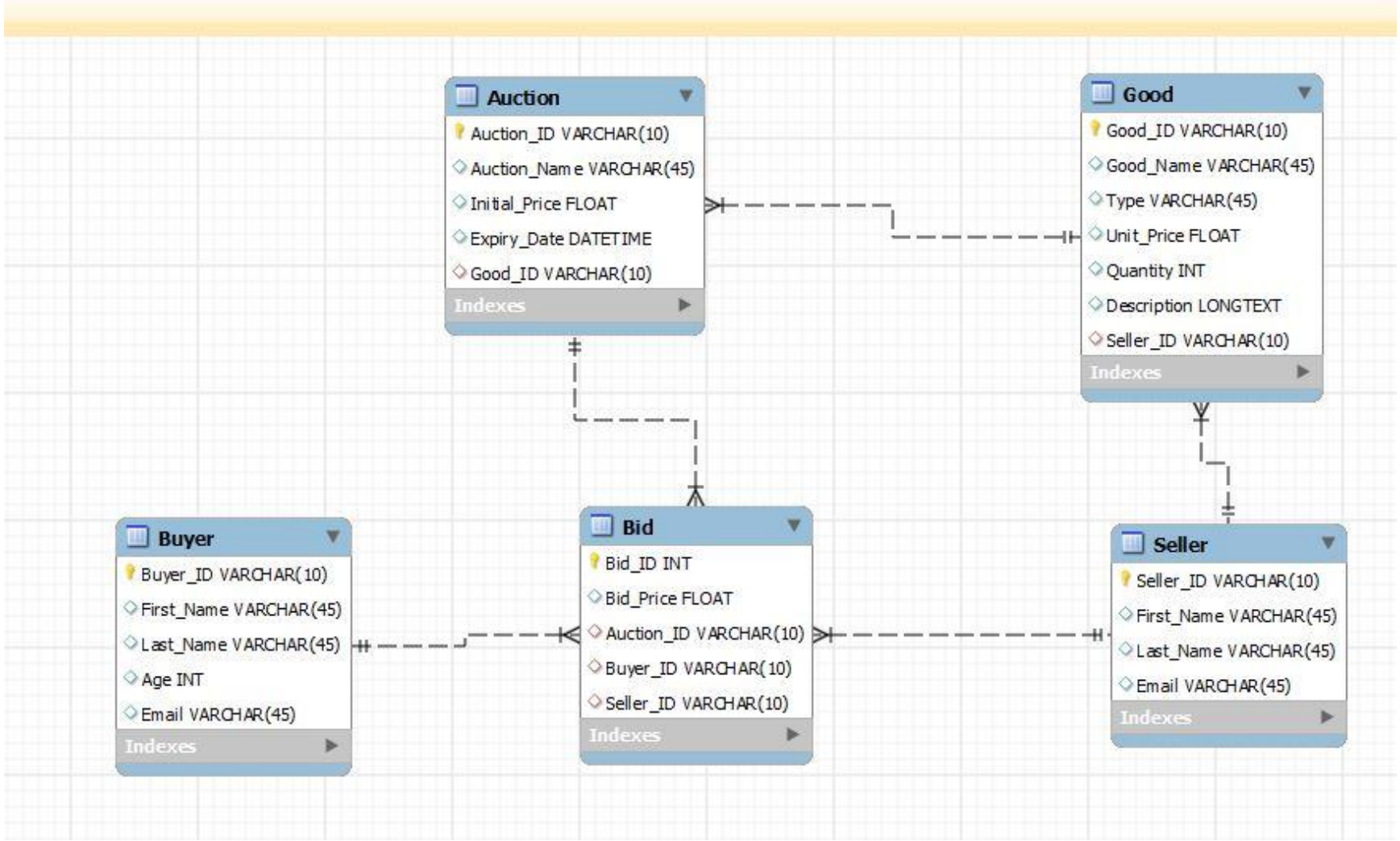
Auction Table



Bid Table


Bid - Table
Table Name: Bid
Schema:
Column Name
Datatype
PK
NN
UQ
BIN
UN
ZF
AI
Default
Bid_ID
INT
Bid_Price
FLOAT
Auction_ID
VARCHAR(10)
Buyer_ID
VARCHAR(10)
Seller_ID
VARCHAR(10)
Columns
Indexes
Foreign Keys
Triggers
Partitioning
Options
Inserts
Privileges

2.4 Logical Database Design



3 SYSTEM ARCHITECTURE DESIGN

3.1 Design Approach

The GuessBid web application design approach is based on a client-server with 3-tier software architecture, which are the Client tier, Business logic tier and the persistence tier. Each tier is described as follows:

Client tier

The client tier is where the user accesses the application. This tier is responsible for translating user actions and presenting the output of tasks and results into something the user can understand.

Business Logic tier

The business logic layer typically contains deployed EJB components that encapsulate business rules and other business functions in Session beans, Entity beans, Message driven beans. This tier coordinates the application, processes commands, makes 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 follows a top-down process approach, so the outermost tiers were first identified and then broken into components that encapsulate the functionality. Hence each component is responsible for certain functionalities and interacts with others.

3.2 General Package Design

Since each tier is broken into components and each component is responsible for a set of functionalities that fulfill the requirements. There is a correlation between use cases (functionality) and package design.

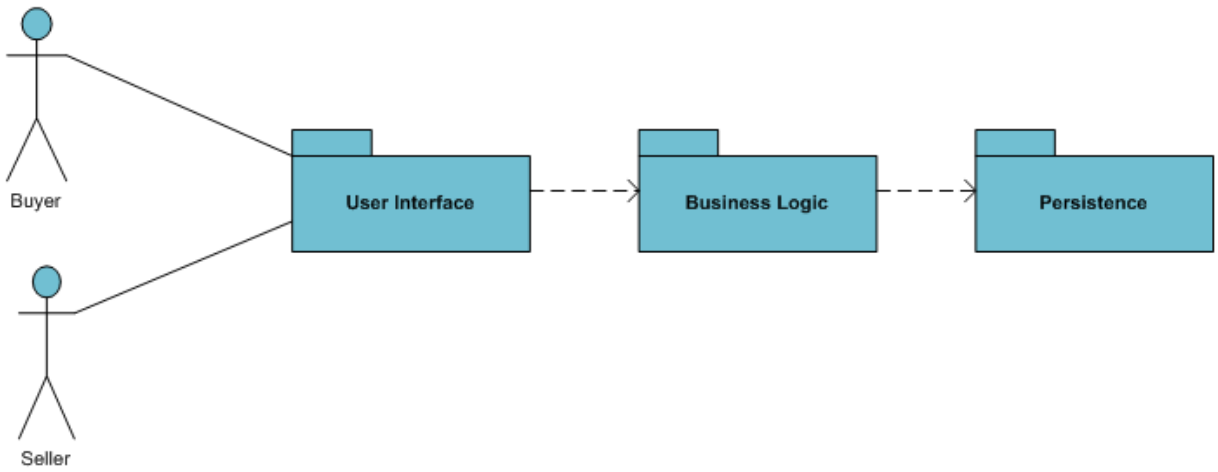


Figure 8 General Package Diagram

User Interface 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.

3.3 System Architecture

GuessBid web application shall be developed by using a general 3-tier J2EE Architecture, as mentioned earlier .We have conceptually identified the components of each, by dividing them in tiers which represent logically a clearer view of what each of them is responsible for. It should be noted that:

- Client Tier and Web Tier represent the Web Component.
- Business Logic Tier represents the Business Logic component.
- Persistence Tier represents Persistence Component.
- Database represents the data model.

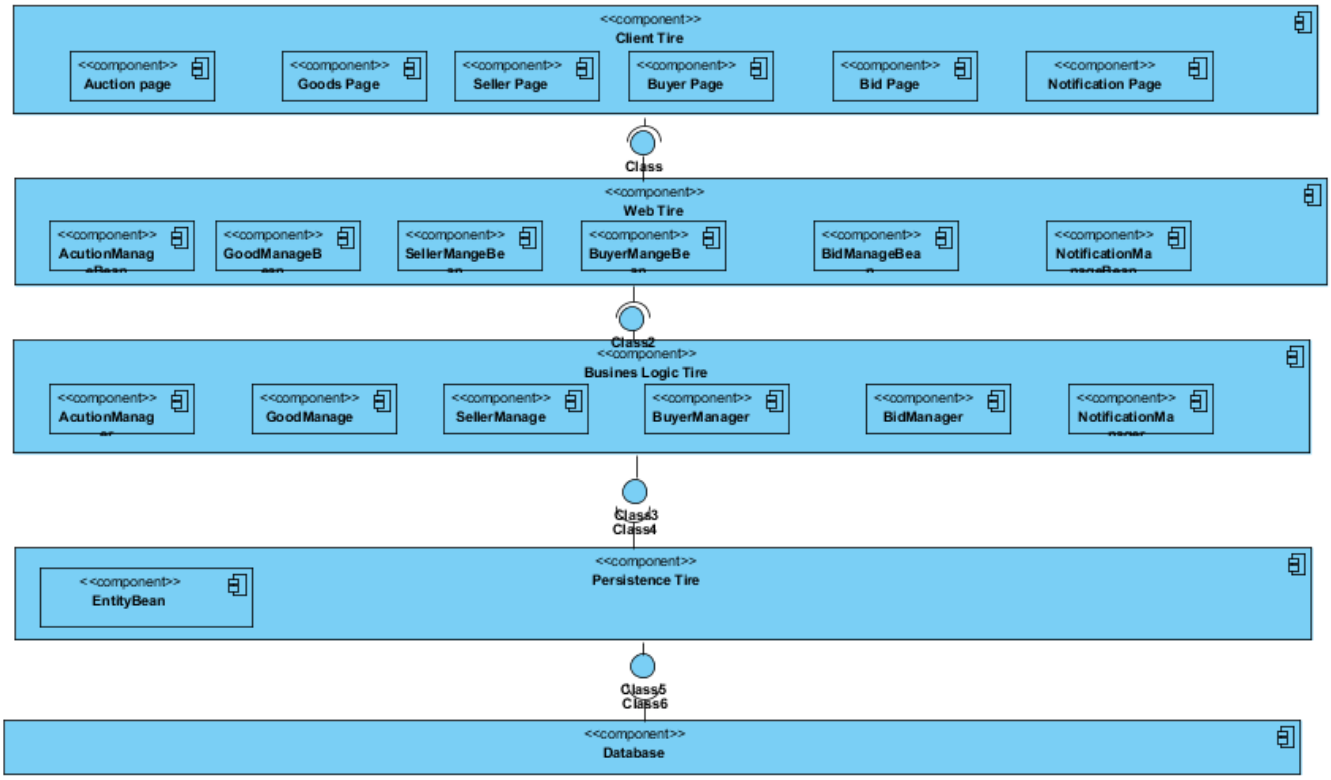


Figure 9 System Architecture Diagram

Even though we will build our application based on the 3-tier architecture, it is clear that the user will interact with the XHTML pages, which in their side

are implemented with beans to manage user interaction. This web interaction is then supported by the business tier, which holds on the information provided by the persistence layer. The persistence layer is the one in charge of the connection to the database and managing all the queries needed from the above layers.

3.4 Detailed Design

Implementation modules/components

Our project is composed of 3 main components, which shall be implemented during implementation phase.

- Web component
- Business logic component
- Persistence component

Web component

In this section we provide the useful information for what needs to be implemented regarding this component. We have 6 main Subcomponents and their related Managed Beans.

- Buyer page
- Seller page
- Goods page
- Auction Page
- Bid page
- Notification page

We here by remind that we are using JSF and that user events in the pages are managed by Managed Beans divided in 6 sections as well. The related beans Managed Beans are:

- BuyerManageBean
- SellerManageBean
- GoodsManageBean
- AuctionManageBean
- BidManageBean
- NotificationManageBean

These beans represent the conceptual idea of the Managed Beans, as there will be more managed beans that will be needed during implementation.

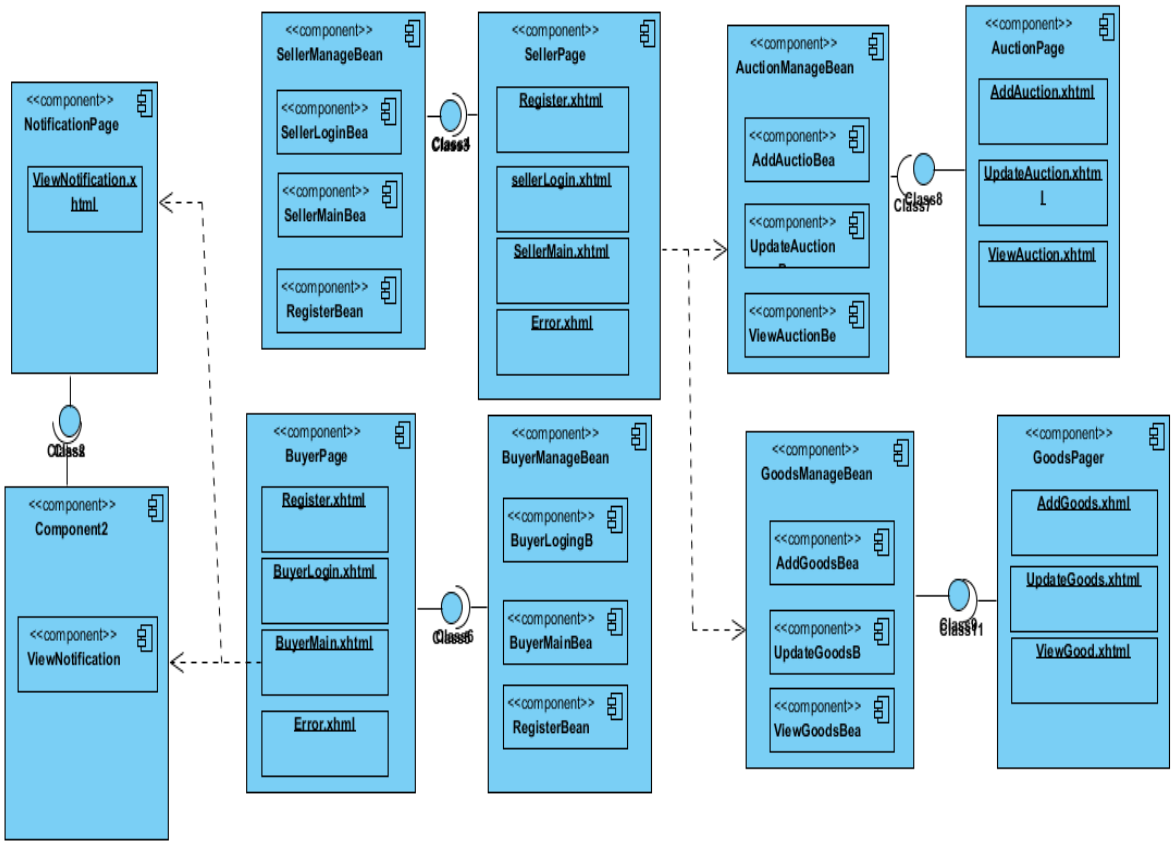


Figure 10 System component Interaction

The Component feature available for Seller

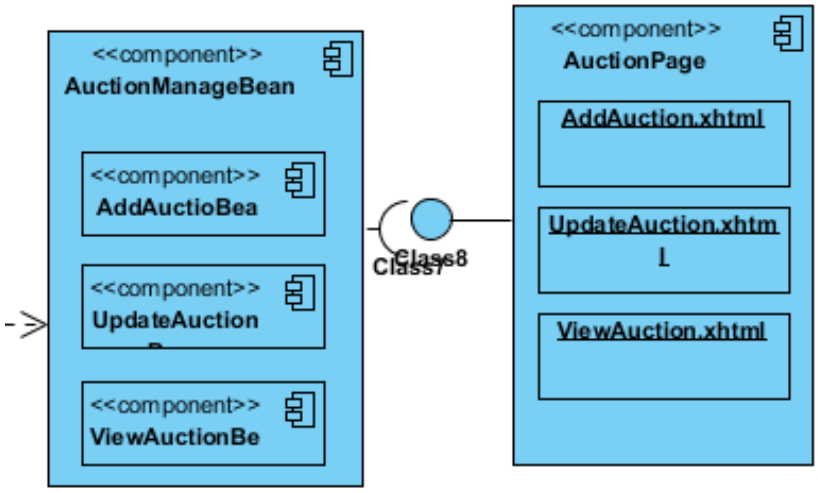


Figure 11 Auction module component Interaction

Component name	**AddAuction.xhtml**
Definition	The UI used to implement Auction registration form
Responsibility	Display Auction registration form that is used insert new Auction information

Component name	**UpdateAuction.xhtml**
Definition	The UI used to implement Auction modification form
Responsibility	Display Auction modifying form that is used update Auction information

Component name	DeleteAuction.xhtml
Definition	The UI used to remove Auction information
Responsibility	Display Auction information form that is used to select Auction and remove

Component name	AddAuctionBean
Definition	Manage bean to add new Auction
Responsibility	• To load form of Auction registration • To validate the input information of the Seller • To redirect to SellerMain pager or Error page

Component name	Update AuctionBean
Definition	Manage bean to update existing Auction
Responsibility	• To load the selected Auction information • To validate the input information of the Seller • To redirect to SellerMain page or Error page

Component name	DeleteAuctionBean
Definition	Manage bean to delete Auction information
Responsibility	• To load the selected Auction information • To redirect to SellerMain page

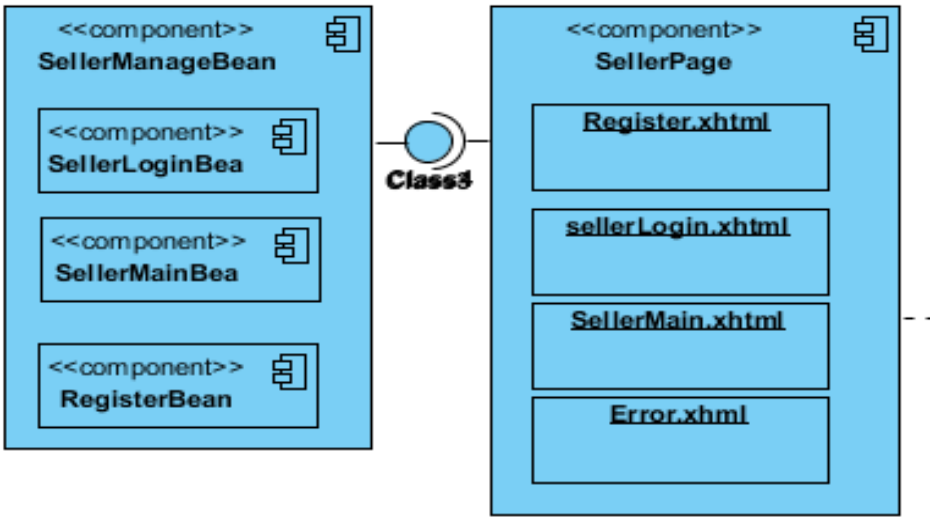


Figure 12 Seller module component Interaction

Component name	SellerMain.xhtml
Definition	The UI implemented for displaying Seller main page
Responsibility	Display Seller main page Display list of Auction and invitation the Seller manage

Component name	SellerLogin .xhtml
Definition	The UI implemented for displaying Seller login page
Responsibility	Display Seller login page

Component name	Error.xhtml
Definition	The UI implemented for displaying error page
Responsibility	Display error message for invalid operation

Component name	SellerMainBean
Definition	Manage bean for providing main page for Seller
Responsibility	• To load Seller main page • To load a list of Auction the Seller manage

Component name	SellerLoginBean
Definition	Manage bean for providing login page for Seller
Responsibility	• To load login for Seller • To validate login information • To redirect to Seller main page

Component name	RegisterBean
Definition	Manage bean for providing register page for Seller
Responsibility	• To load Seller registration page • To validate input • To redirect to Seller main page

The Component feature available for Buyer and Seller

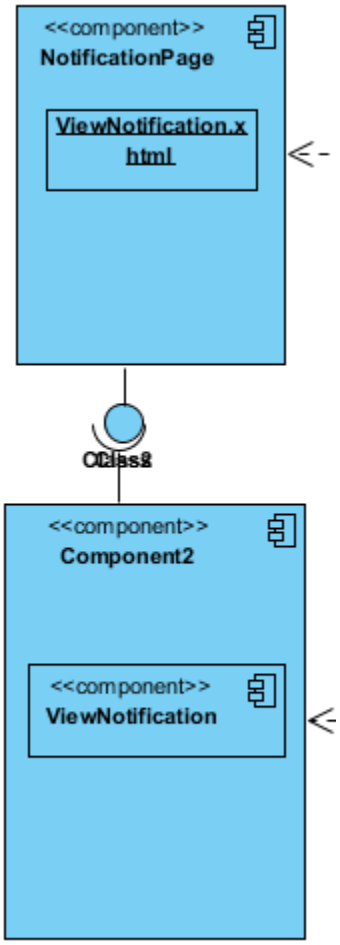


Figure 4 Notification module components Interaction

Component name	ViewNotification.xhtml
Definition	The UI used to view list of Auction invitation
Responsibility	Display a list of invitation that are sent form the Seller with a response button on it

Component name	SendNotification.xhtml
Definition	The UI used to send invitation to the Buyere
Responsibility	Display a list available Buyere and send button on it

Component name	ViewNotificationBean
Definition	Manage bean to view Auction invitation
Responsibility	To load the Auction invitaion To redirect to BuyereMain page

The Component feature available for Buyer

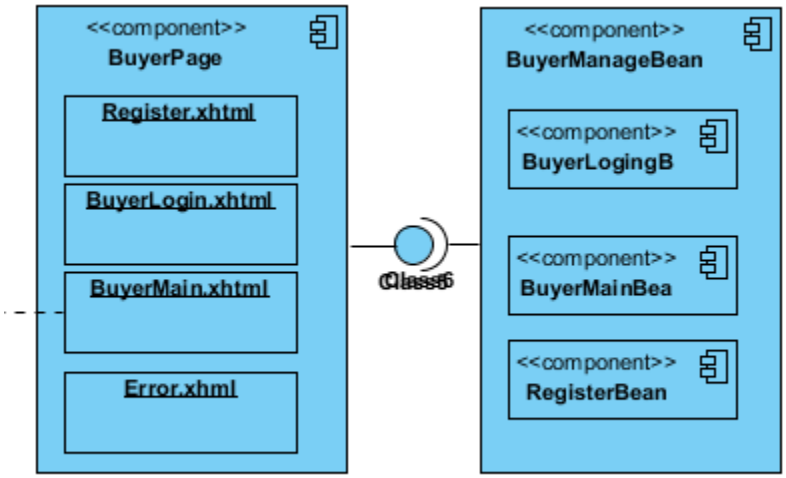


Figure 14 Buyer module component Interaction

Component name	BuyerMain.xhtml
Definition	The UI implemented for displaying Buyer main page
Responsibility	Display Buyere main page Display list of Auction and invitation the Buyere participated and accepted respectively

Component name	BuyerLogin .xhtml
Definition	The UI implemented for displaying Buyere login page
Responsibility	Display Seller login page

Component name	Error.xhtml

Definition	The UI implemented for displaying error page
Responsibility	Display error message for invalid operation

Component name	**BuyerMainBean**
Definition	Manage bean for providing main page for Buyere
Responsibility	• To load Buyere main page • To load a list of Auction the Buyere participating

Component name	**BuyereLoginBean**
Definition	Manage bean for providing login page for Buyere
Responsibility	• To load login for Buyere • To validate login information • To redirect to Buyere main page

Component name	**RegisterBean**
Definition	Manage bean for providing register page for Buyere
Responsibility	• To load Buyere registration page • To validate input • To redirect to Seller main page

The Component feature available for Goods

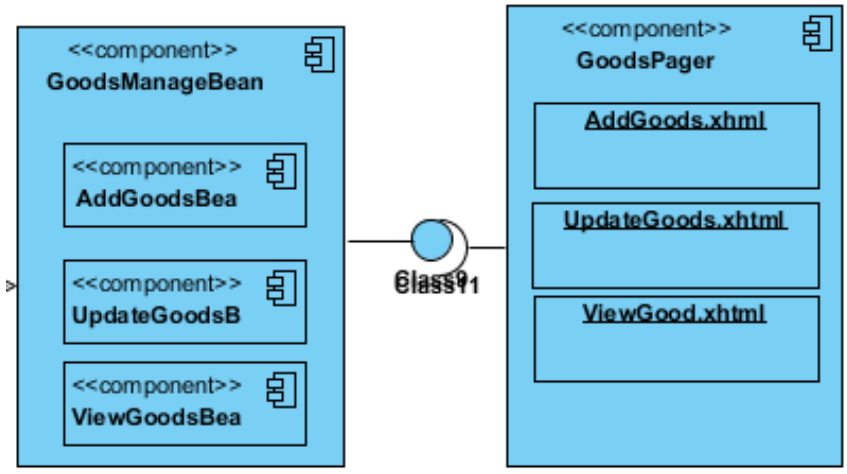


Figure 15 Goods module component Interaction

Component name	**AddGood.xhtml**
Definition	The UI used to implement goods registration form
Responsibility	Display event registration form that is used insert new goods information

Component name	**UpdateGood.xhtml**
Definition	The UI used to implement goods modification form
Responsibility	Display event modifying form that is used update goods information

Component name	DeleteGood.xhtml
Definition	The UI used to remove goods information
Responsibility	Display event information form that is used to select goods and remove

Component name	AddGoodBean
Definition	Manage bean to add new good
Responsibility	• To load form of goods registration • To validate the input information of the good • To redirect to Good Main pager or Error page

Component name	Update GoodBean
Definition	Manage bean to update existing good
Responsibility	• To load the selected good information • To validate the input information of the good • To redirect to GoodMain page or Error page

Component name	DeleteGoodBean
Definition	Manage bean to delete good information
Responsibility	• To load the selected good information • To redirect to GoodMain page

Component name	ViewNotificationBean
Definition	Manage bean to view Auction invitation
Responsibility	• To load the Auction invitation • To redirect to BuyereMain page

3.5 Website organization

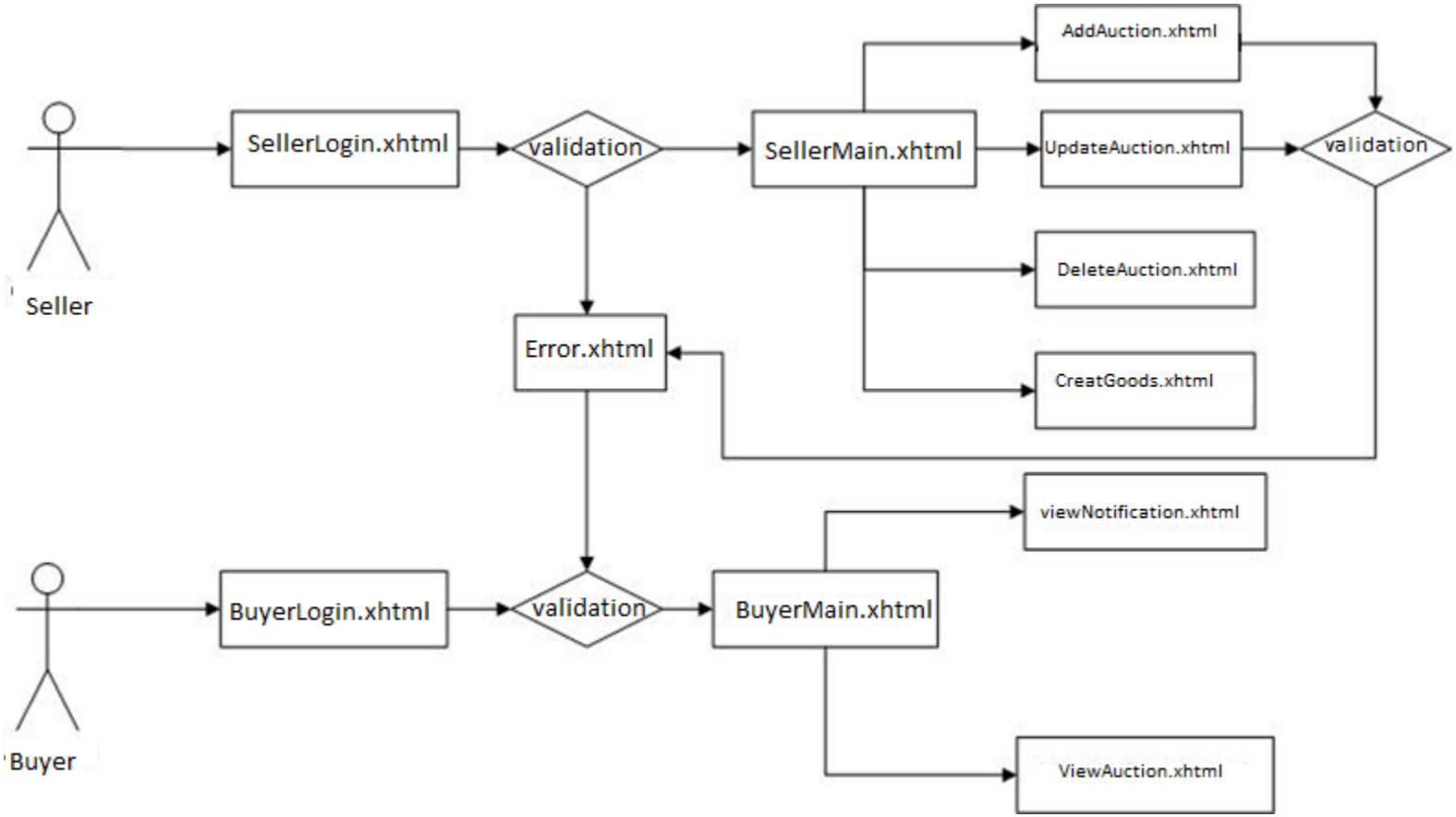


Figure 16 Website Organization

3.6 Deployment diagram

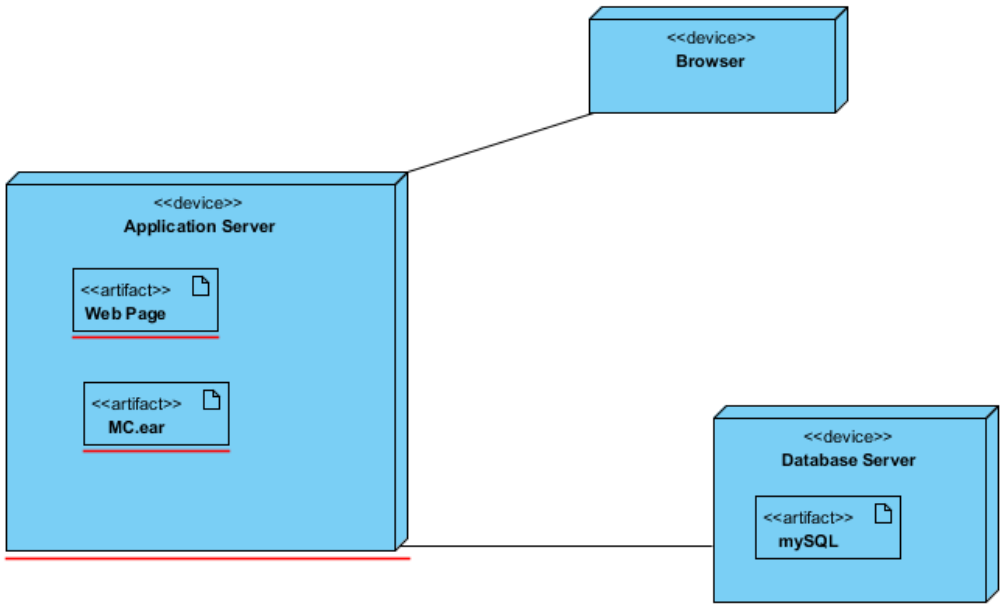


Figure 17 Deployment diagram

3.7 Assumption

There are few assumptions we made here:

- We assume that there are Pre registered Seller that are managing Auction information.
- The Password will be stored as MD5, then a sequence of 32 digits hexadecimal, so the type of the attribute will be a string of 32 characters.
- There are lookup tables on our system database witch are not included on this design document we only include entities of the functional requirement