CHAPTER I
INTRODUCTION

An Auction is a work which means augment. Auction is a bid, a process of selling, buying and services offered take place. There are several different types of auctions and certain rules exist for each auction. There are variations for an auction which may include minimum price limit, maximum price limit and time limitations etc. Depending upon the auction method bidder can participate remotely or in person. Remote auction includes participating through telephone, mail, and internet. Online auction is becoming more and more popular in electronic commerce and hence it should system must increase its quality and security.

The online auction system is a model where we participate in a bid for products and service. This auction is made easier by using online software which can regulate processes involved. There are several different auction methods or types and one of the most popular methods is English auction system. This system has been designed to be highly-scalable and capable of supporting large numbers of bidders in an active auction.

Online Auctioning System has several other names such as e-Auctions, electronic auction etc. The requirement for online auction or online bidding can be more accurately specified by the client. It should be healthy and will be a good practice when it is made more transparent as a matter of fact.

Online bidding has become a standard method for procurement process. Bidders can be maintained in a single database according to the preference, and they can be monitored. User's data can be maintained in a confidential way for validity and integrity of contractual documentation. Neat reporting reduces paperwork, postage, photocopying and time beneficial. Multiple bidders can be communicated with a great ease. This system allows multiple bids by single users.

1.1 Objective:

The Objective is to develop a user-friendly auctioning site where any kind of product can be auctioned and provide value-added services to the bidders and the sellers. The products will be authenticated and the site provides a safe environment for online users:

- Secure registration of all users including a personal profile Administrators would authorize the product to auction, set auction dates and Minimum auction amount for that product.
- Complete Search/Site Map of the entire site for easy access.
- Discussion forums for users to interact with other users to know about the product's value and originality.
- Online Legal Documentation to avoid disputes. Guidance to the users about the same must be available.
- Rare articles may be withheld by owner on the advice of the administrator to be thrown open in special auctions held by the site so as to increase the bid-values

1.2 Scope:

- The Online Auctioning System (OAS) addresses the bid product on online without any commission anywhere any product.
- The system has been designed to be highly-scalable and capable of supporting large numbers of bidders in an active auction.
- The payment gateway is very secure from buyer to seller and vice versa
- This SRS describes only required functionality of OAS, not the functionality of external systems like data storage, change management or version control systems.
- This document does not divide the OAS into subsystems; it describes only requirements for the whole system functionality which will be defined in the design model.

1.3 Organization of Report:

The report represents different aspects about Online Auction System is the flaws available in existing system followed by explaining the idea regarding the proposed system for better way of handling projects. The next chapter provides a detailed explanation about hardware and software requirements, requirement analysis and specification with SRS. The subsequent chapters' deals with system design at various levels followed by the implementation of design in terms of coding and finally codes are tested to get the result as specified in the SRS. The last two chapters deals with conclusion, future scope and references for project and report preparation.

CHAPTER II
HARDWARE AND SOFTWARE REQUIREMENTS

2.1 Hardware Requirements:

1	**Processor**	core i3/i5/i7
2	**Hard Disk**	80 GB or above
3	**RAM**	1GB or above

Table 2.1 Hardware Requirements.

2.2 Software Requirement:

1	**Operating system**	Windows 7/8/10
2	**User-Interface**	HTML, CSS
3	**Client-side scripting**	JavaScript
4	**Programming language**	Java
5	**Technologies**	JDBC, Servlets, JSP
6	**IDE/Workbench**	Eclipse 8.0
7	**Database**	Oracle11g
8	**Server Deployment**	Tomcat8.5x or Higher
9	**Frame Work**	Struts

Table 2.2 Software Requirements.

CHAPTER III
REQUIRMENT ANALYSIS AND SPECIFICATION

3.1 SDLC Methodology:

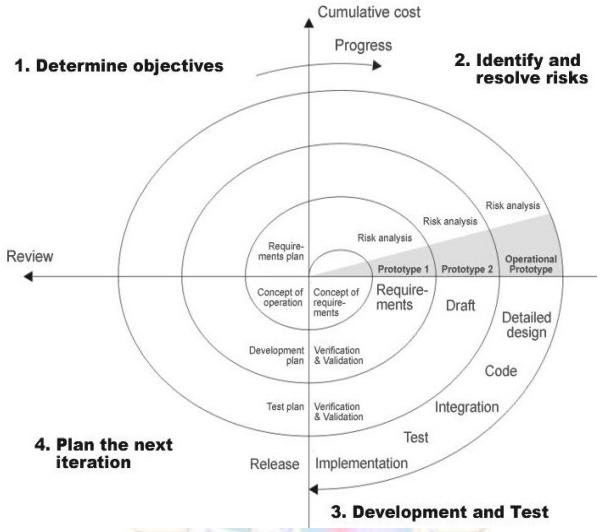

Figure 3.1 Spiral Model.

This diagrammatic representation refers to a basic model that appears like a spiral with many loops. The exact number of loops in the spiral is not fixed. Each loop of the spiral represents a phase of the software process. For example, the innermost loop might be concerned with feasibility study. The next loop with requirements specification, the next one with design, and so on. Each phase in this model is split into four sectors (or quadrants). The following activities are carried out during each phase of a spiral model.

First quadrant (Objective Setting)

- During the first quadrant, it is needed to identify the objectives of the phase.
- Examine the risks associated with these objectives.

Second Quadrant (Risk Assessment and Reduction)

- A detailed analysis is carried out for each identified project risk.
- Steps are taken to reduce the risks. For example, if there is a risk that the requirements are inappropriate, a prototype system may be developed.

Third Quadrant (Development and Validation)

- Develop and validate the next level of the product after resolving the identified risks.

Fourth Quadrant (Review and Planning)

- Review the results achieved so far with the customer and plan the next iteration around the spiral.
- Progressively more complete version of the software gets built with each iteration around the spiral.

3.2 System Analysis:

3.2.1 Limitation of Existing System:

- This existing system is not providing secure registration and profile management of all the users.
- The manual system gives us very less security for saving data some data may be lost due to mismanagement.
- This system doesn't provide proper authorization for the products to be auctioned.
- Existing system is not having the facility of sharing data among the users.
- This system doesn't provide managing of the accounts and the payments

3.2.2 Proposed System:

The development of this new system contains the following activities, which try to develop the web-application entire process keeping in the view of database integration approach.

- This system will generate team progress and also provides secure registration and profile management of the users.
- Administrators would authorize the product to auction, set auction dates & minimum auction amount for that product.
- Prior to each bid, the user's bank or credit account must be authenticated for available balance required for the bid.
- Users can select their interested fields for bidding and periodic Mail alerts must be sent in case an article in that field goes on auction
- Complete Search/Site Map of the entire site for easy access.
- Discussion forums for users to interact with other users to know about the product's value and originality.
- Online Legal Documentation to avoid disputes. Guidance to the users about the same must be available.
- Rare articles may be withheld by owner on the advice of the administrator to be thrown open in special auctions held by the site so as to increase the bid-values.

3.3 Software Requirement Specification:

3.3.1 Introduction:

The following subsections of the SRS document provide an overview of the entire software requirement specification.

i) Purpose:

This document specifies the Software Requirements Specification (SRS) for Online Auctioning System (OAS). It describes scope of the system, both functional and non-functional requirements for the software to be developed, design constraints and system interfaces.

ii) Scope:

- The Online Auctioning System (OAS) addresses the bid product on online without any commission anywhere any product.
- The system has been designed to be highly-scalable and capable of supporting large numbers of bidders in an active auction.
- The payment gateway is very secure from buyer to seller and vice versa
- This SRS describes only required functionality of OAS, not the functionality of external systems like data storage, change management or version control systems.
- This document does not divide the OAS into subsystems; it describes only requirements for the whole system functionality which will be defined in the design model.

iii) Abbreviations:

JSP stands for Java Server Pages.

HTTP stands for Hyper Text Transfer Protocol.

iv) References:

IEEE Recommended Practice for Software Requirements Specification-IEEE Std. 830-1993.

v) Overview:

The rest of this SRS document describes the various system requirements, interfaces, features and functionalities in detail.

3.3.2 Overall Description:

Product Perspective:

a) User interfaces:

The application will have a user friendly and menu-based interface. Following screens will be provided:

- A login screen for entering the user name, password.
- Access to different screens will be based upon the user
- There is a screen for displaying information regarding the items.
- There is an interface for Seller.
- There is an interface for Buyer.
- There is a screen for providing information about Auction.
- There is a screen for giving information about Domains.

b) Hardware interfaces:

- Support for printer for printing results then and there.
- Screen resolution of at least 1200X800 is required for proper and complete viewing of screens. Higher resolution will be accepted.

c) Software interfaces:

- Any windows-based operating system.
- Oracle 10g as the DBMS-for database.

d) Communications interfaces: None

e) Memory Constraints:

- At least 1 GB space on hard disk will be required for running the application.

f) Site Adaptation Requirements:

- Web browser with cookies enabled.

1) User Characteristics:

- Educational level: Users should be comfortable with the English language.
- Experience: Users should have prior information regarding the online feedback.
- Skills: Users should have basic knowledge and should be comfortable using general purpose applications on computers.

2) Constraints:

- The DBMS being used is Oracle 11g, using which all the constraints that are needed for proper database management are truly imposed.
- Normalization considered.

3) Apportioning of Requirements:

- The future versions of the web application will be having a better database to handle larger number of records.

3.3.3 Specific Requirements:

This section provides software requirements to a level of detail sufficient to enable designers to design the system and testers to test the system.

1) External Interface Requirements

a) **Hardware interfaces:**

- Support for printer for printing results then and there.
- Screen resolution of at least 1200 X 800 is required for proper and complete viewing of screens. Higher resolution will be accepted.

b**) Software interfaces:**

- Any windows-based operating system.
- Oracle 11g as the DBMS-for database.
- Tomcat 8.x or higher for deployment

c) Communications interfaces: None

2) Software Product Features:

Validity Checks:

JavaScript provides validity checks for various fields in the forms.

- Sequencing Information: All the information regarding project details, employee list, question details, display of result should be handled sequentially that is data should be stored only in a particular sequence to avoid any inconvenience
- Error Handling: If any of the validations or sequencing flows does not hold true then appropriate error messages will be prompted to the user for doing the needful.

3) Performance Requirements:

This subsection specifies numerical requirements placed on the software or on the human interaction with the software, as a whole. Numerical requirements will include:

- 300 terminals will be supported at a time
- Only text information will be supported (HTTP)
- All the transactions will be processed within seconds.

4) Software System Attributes:

- Security: Only authorized users will be able to access the web app by entering the correct login name and corresponding password.
- Maintainability: The web app can be maintained in present or future. It will be easy to incorporate new requirements in the individual modules.

3.3.4 Module detail description:

The system after careful analysis has been identified to be presented with the following modules:

The Modules involved are

1. Admin Module.
2. Seller Module.
3. Buyer Module
4. Security and authentication
5. Reports

Admin Module:

This module provides the complete information related to products for sale and the buyers can bid for the products and can own them. All this has to be provided and maintained by the admin because the complete auction process is to be kept under control till the product sale gets confirmed. It has to verify the details given by the buyer and seller then it has to confirm all the things furnished by the both buyer and seller.

Seller Module:

Sellers want a place where seller can sale their products at a higher price and get maximum benefit out of that. This is the place where seller can display all his products and sell them. Seller can display all the possible products for sale and can call the people for the auction then after receiving the final bidding whichever is the highest that highest bidder owns the product. Seller can have the benefits directly without any third people involvement.

Buyer Module:

The people always want different things to purchase but in the local market they can have local products only. But in this application buyer can buy any product from any part of the world at a very best competitive price and own the product. Buyer has to just furnish their details and can participate in the bidding to acquire the product, which is for sale.

Security and Authentication:

- Login as buyer or seller or administrator
- Change password
- Forgot Password
- Registration for buyer / seller

Reports:

In this module, different actors can generate the different types of Reports according to their access.

CHAPTER IV
SOFTWARE DESIGN

4.1 UML Diagram:

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

User Model View:

- This view represents the system from the user's perspective.
- The analysis representation describes a usage scenario from the end-user's perspective.

Structural model view:

- In this model the data and functionality are arrived from inside the system.
- This model view models the static structures.

Behavioral Model View:

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View:

In this the structural and behavioral as parts of the system are represented as they are to be built.

Environmental Model View:

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

4.1.1 Use Case Diagrams:

4.1.1.1 Admin Use Case Diagram

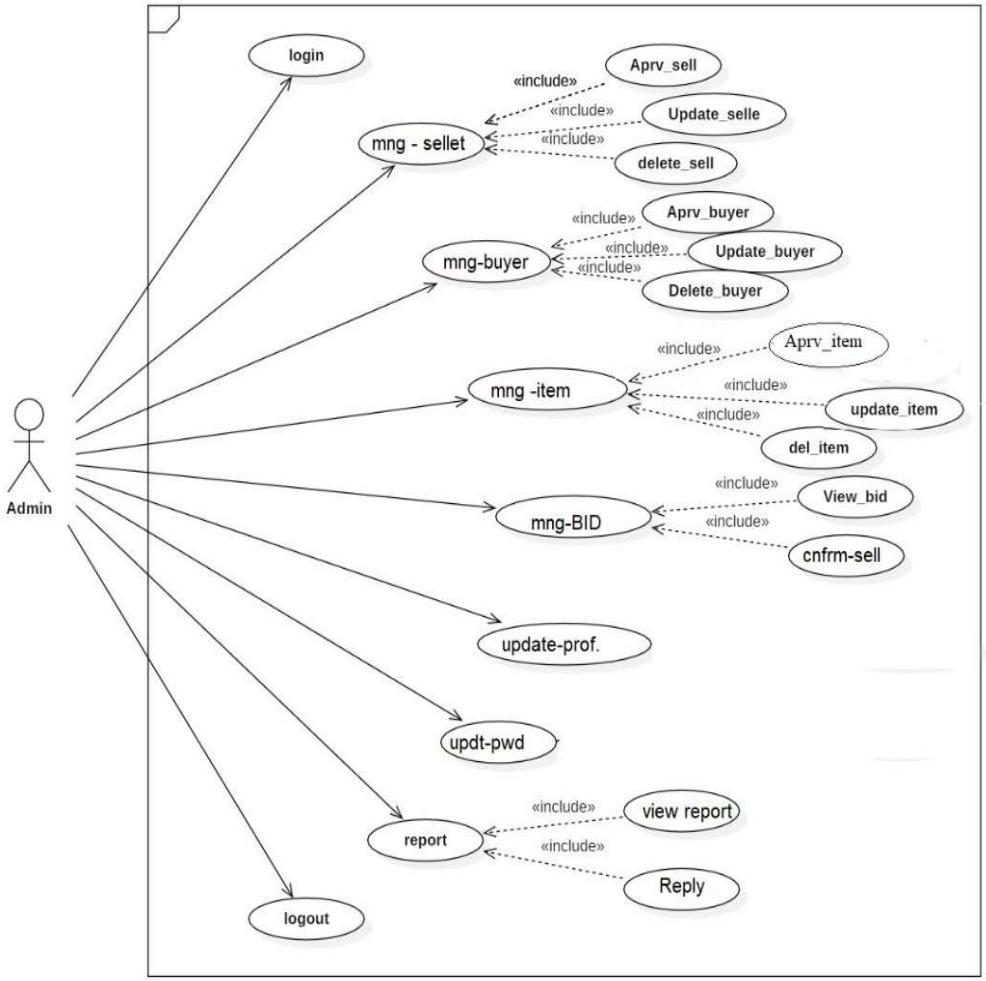

Figure 4.2 Admin Use Case Diagram.

4.1.1.2 Seller Use Case Diagram

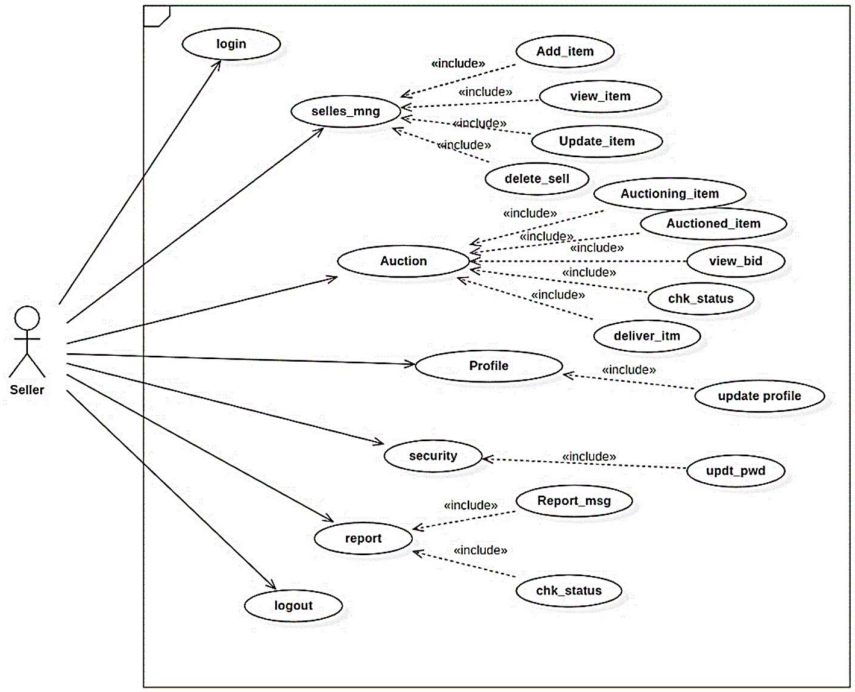

Figure 4.3 Seller Use Case Diagram.

4.1.1.3 Buyer Use Case Diagram

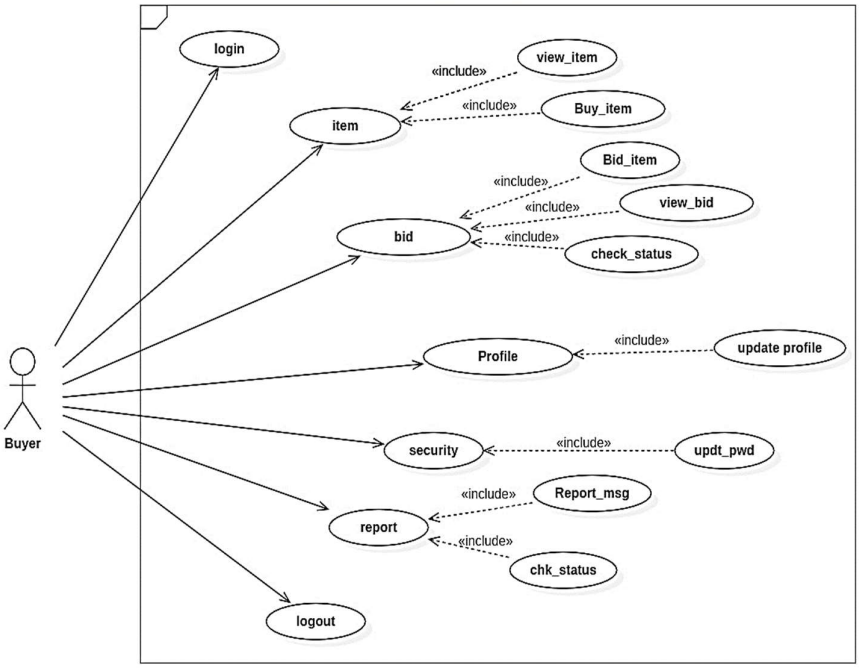

Figure 4.4 Buyer Use Case Diagram.

4.2 Activity Diagram:

4.2.1 Activity Diagram of Admin:

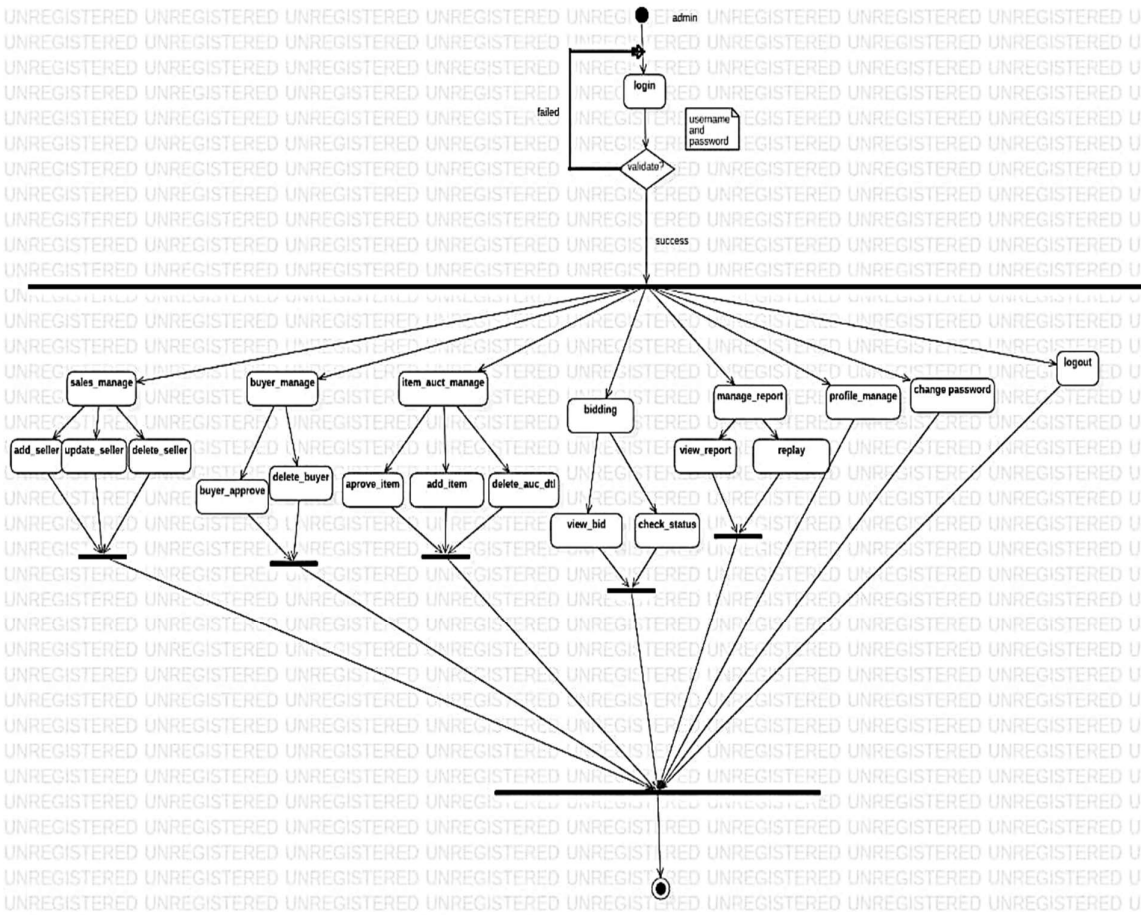

Figure 4.5 Admin Activity Diagram.

4.2.2 Activity Diagram of Seller:

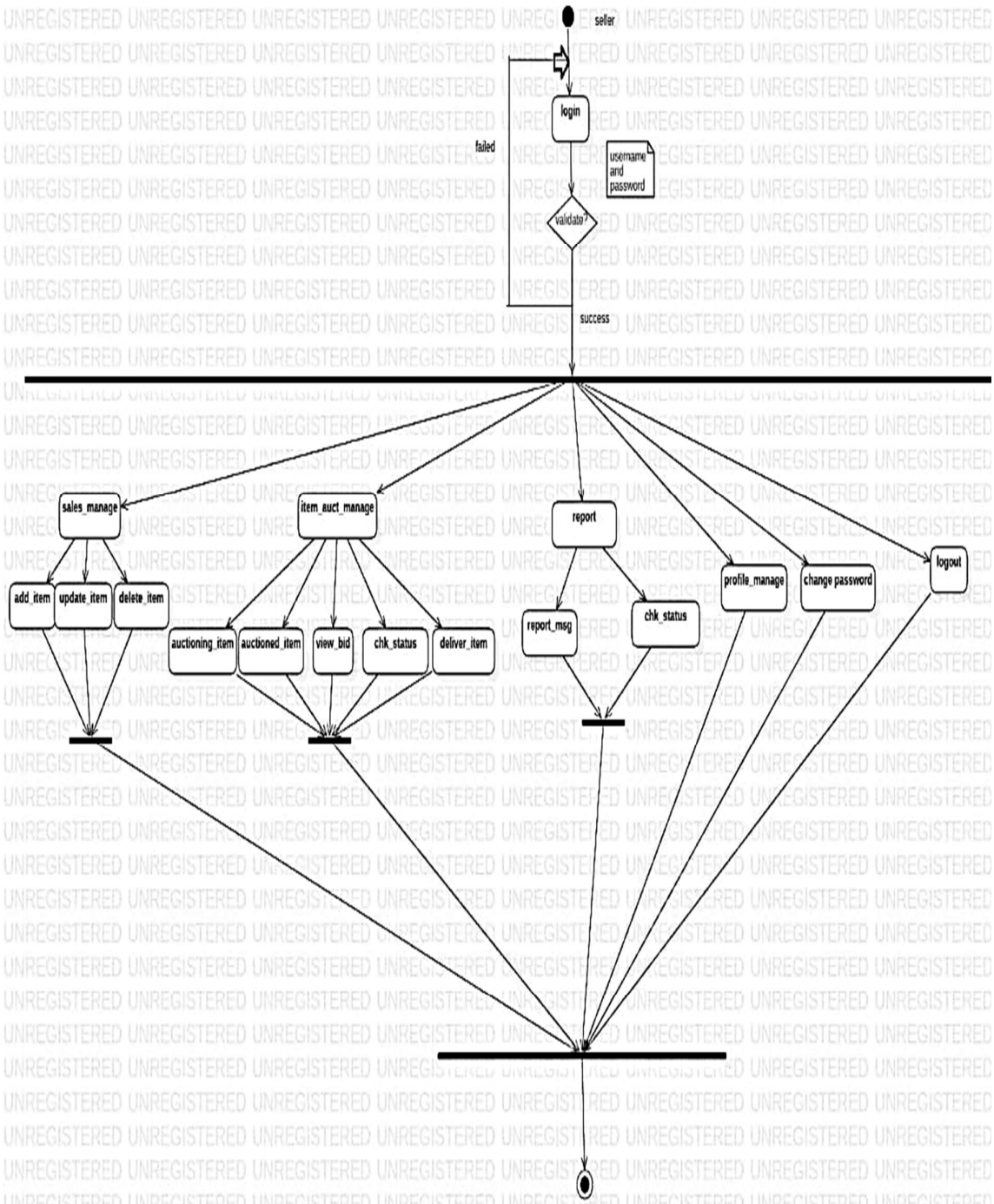

Figure 4.6 Seller Activity Diagram.

4.2.3 Activity Diagram of Buyer:

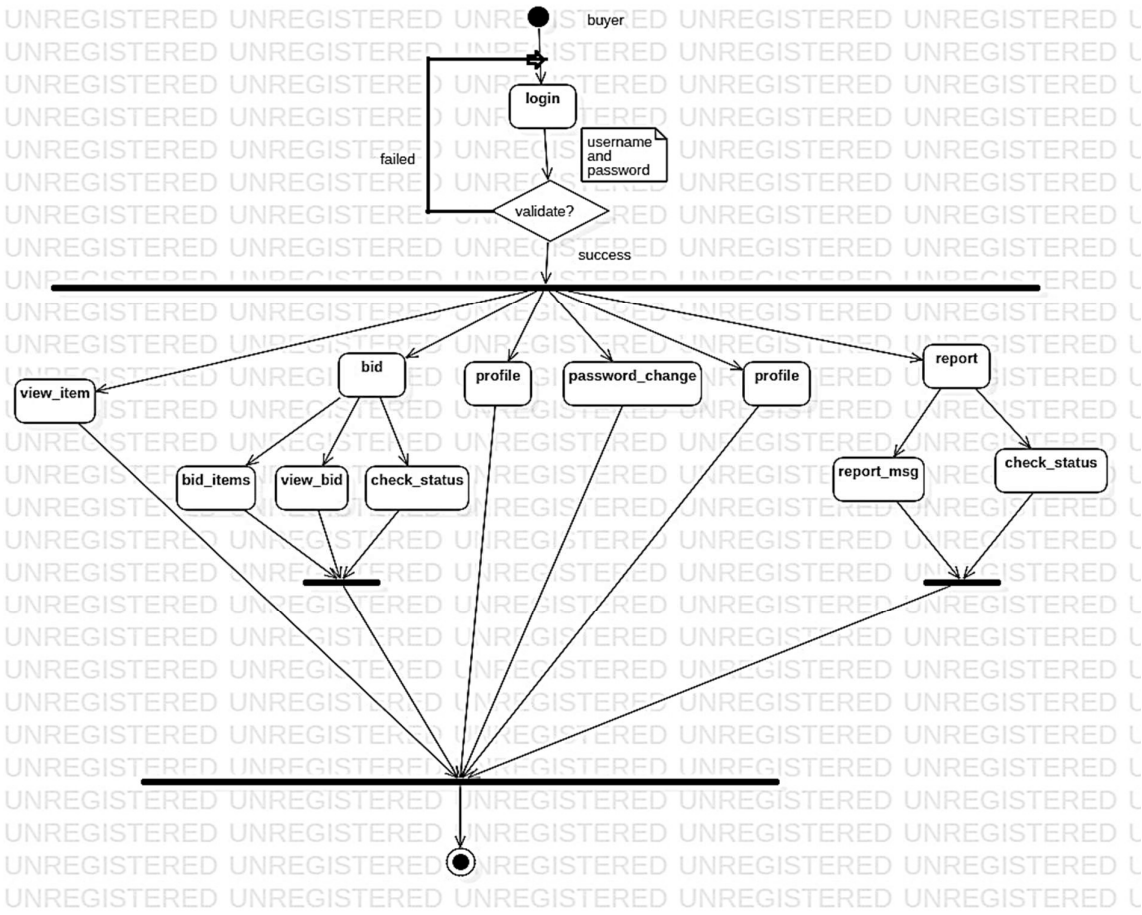

Figure 4.7 Buyer Activity Diagram.

4.3 Sequence Diagram:

4.3.1 Login Sequence Diagram

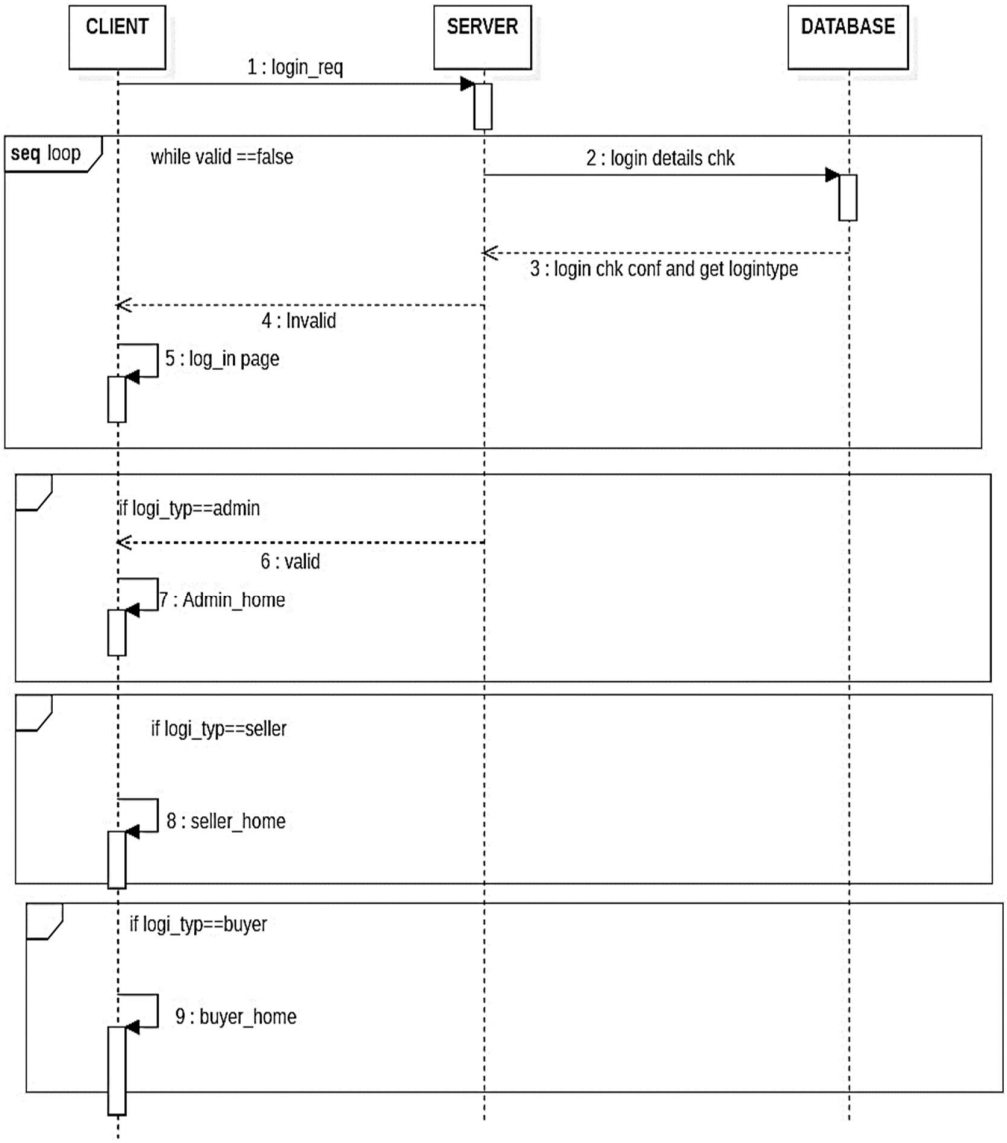

Figure 4.8 Login Sequence Diagram.

4.3.2 Admin-profile sequence diagram

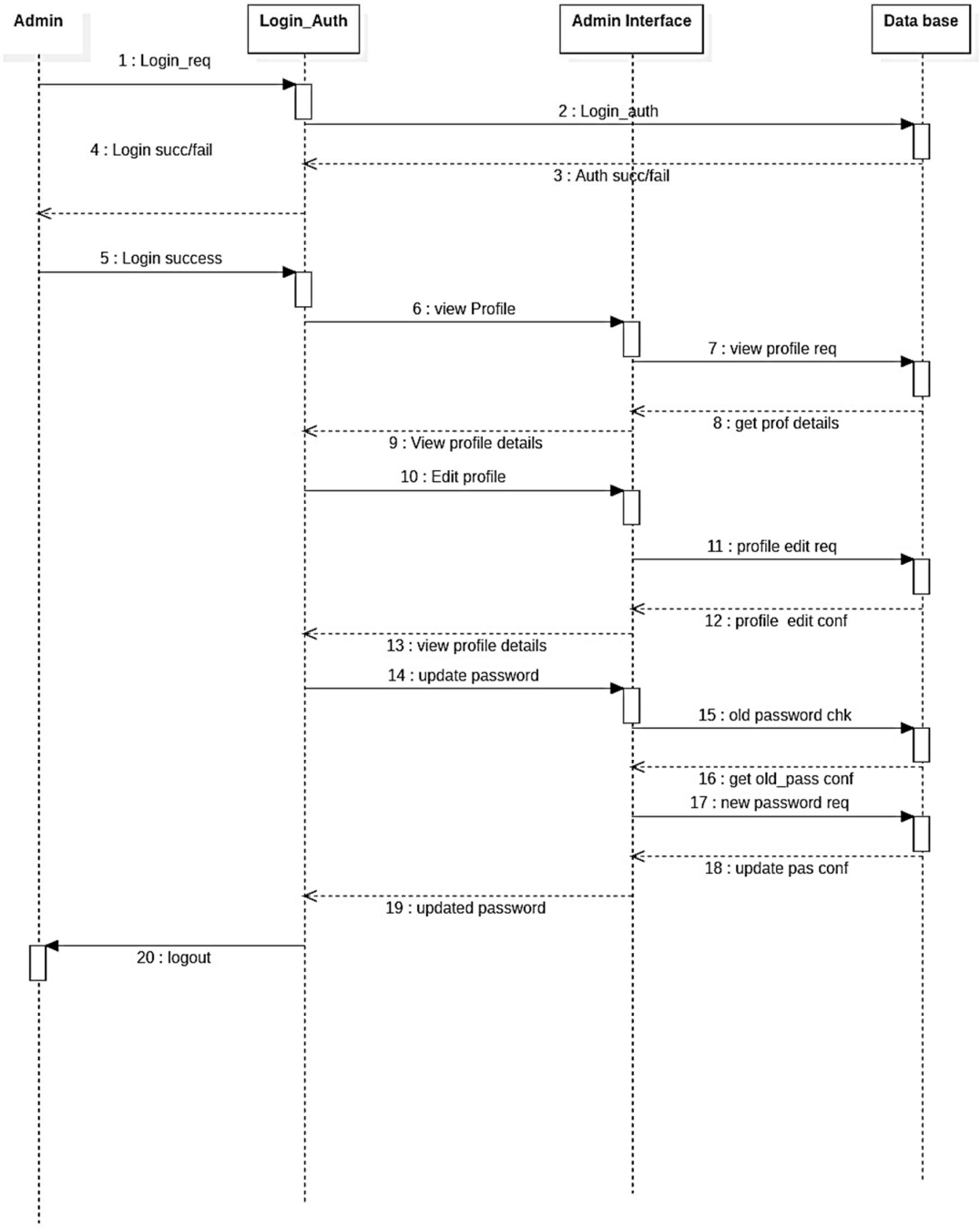

Figure 4.9 Admin-profile Sequence Diagram.

4.3.3 Admin-seller Sequence Diagram

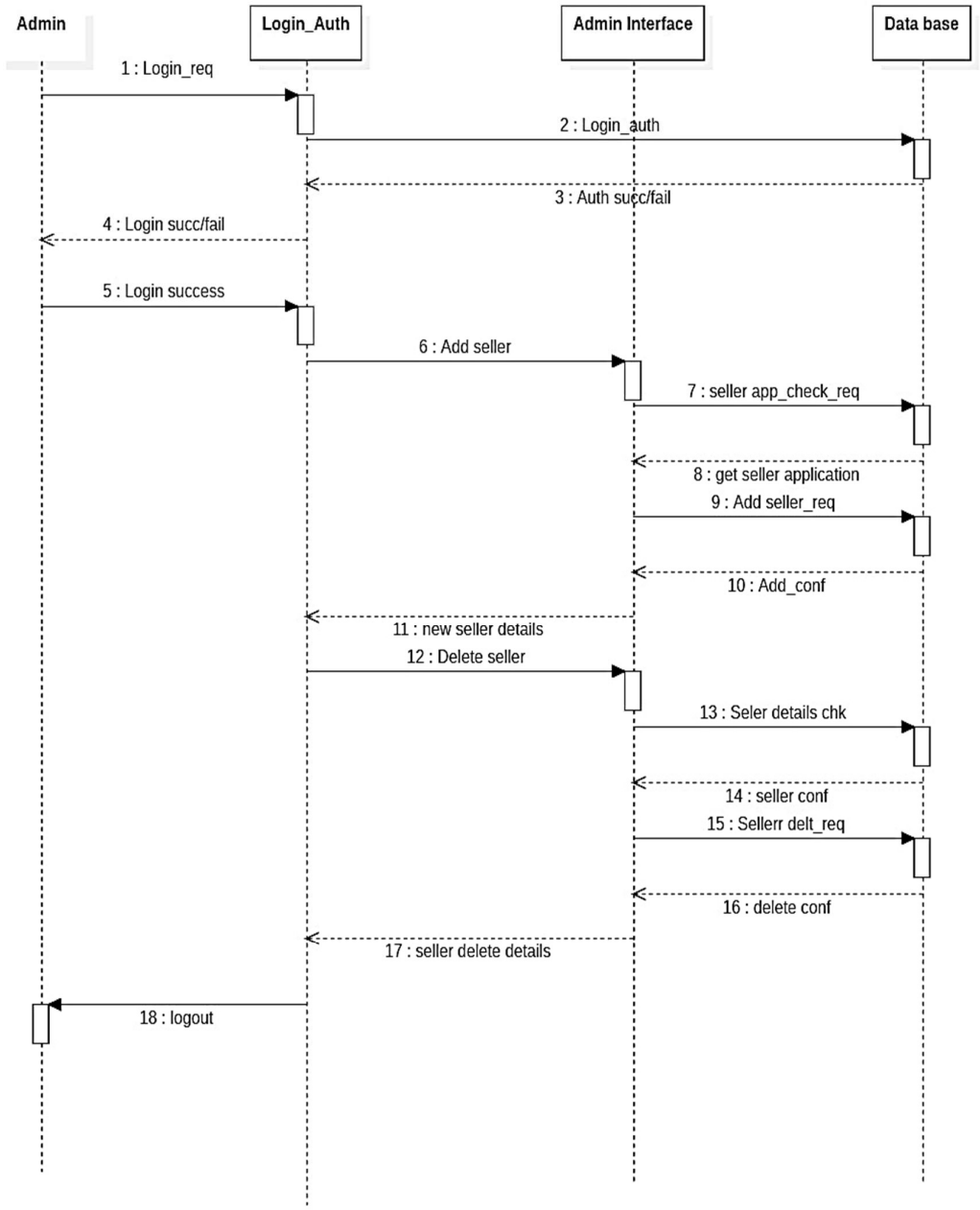

Figure 4.10 Admin-Seller Sequence Diagram

4.3.4 Admin-Buyer Sequence Diagram

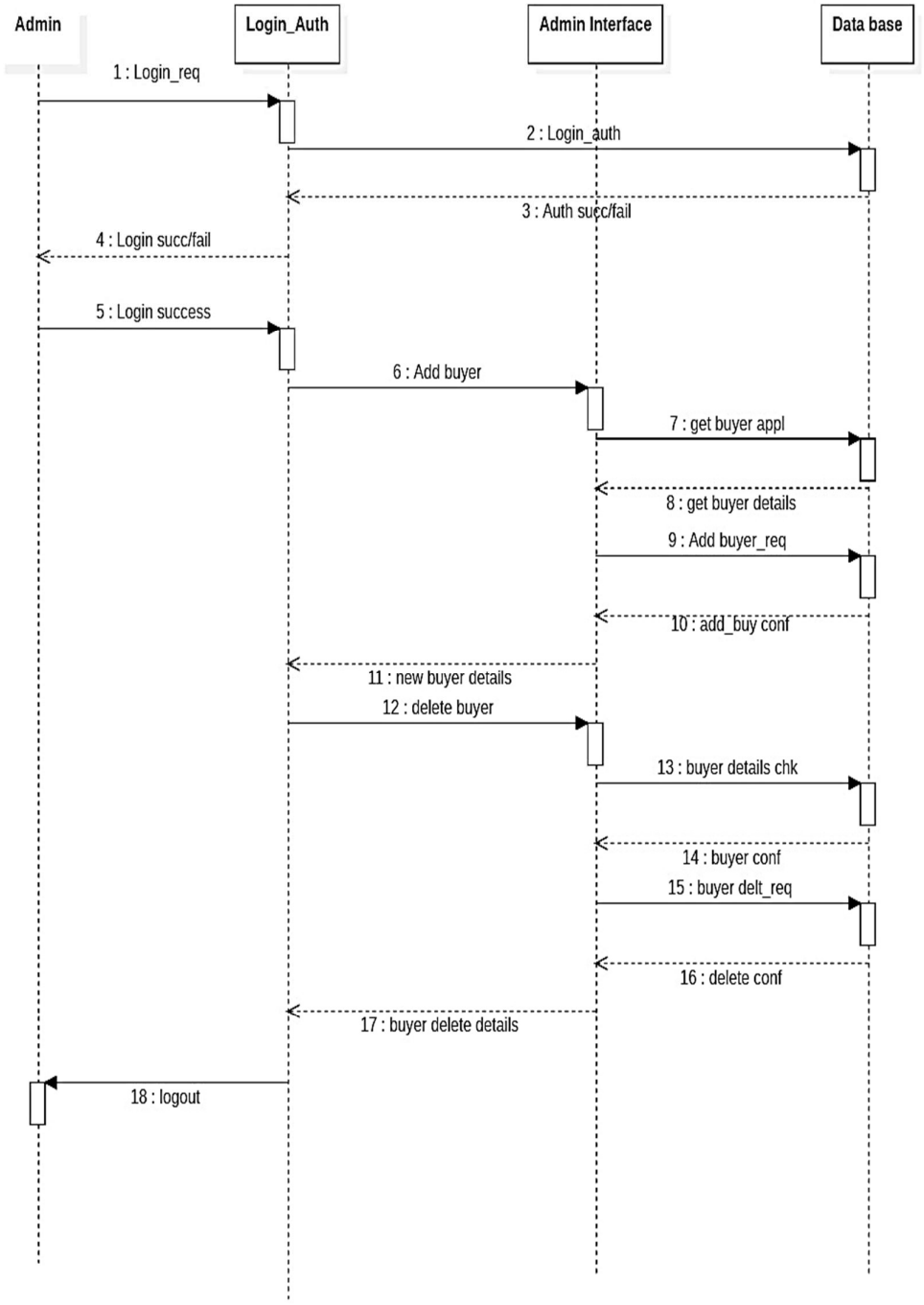

Figure 4.11 Admin-Buyer Sequence Diagram

4.3.5 Admin-Item Sequence Diagram

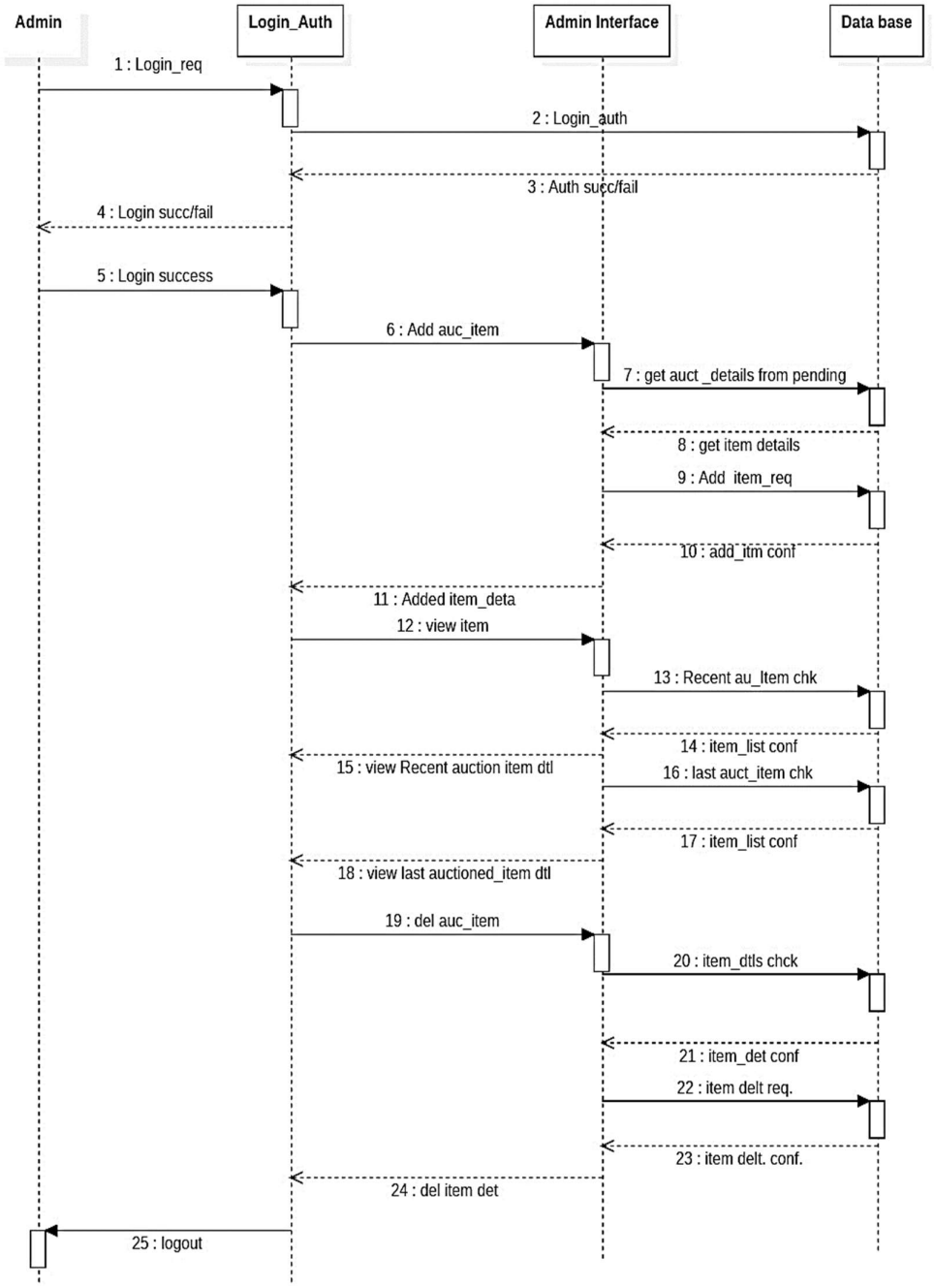

Figure 4.12 Admin-Item Sequence Diagram

4.3.6 Seller-profile Sequence Diagram

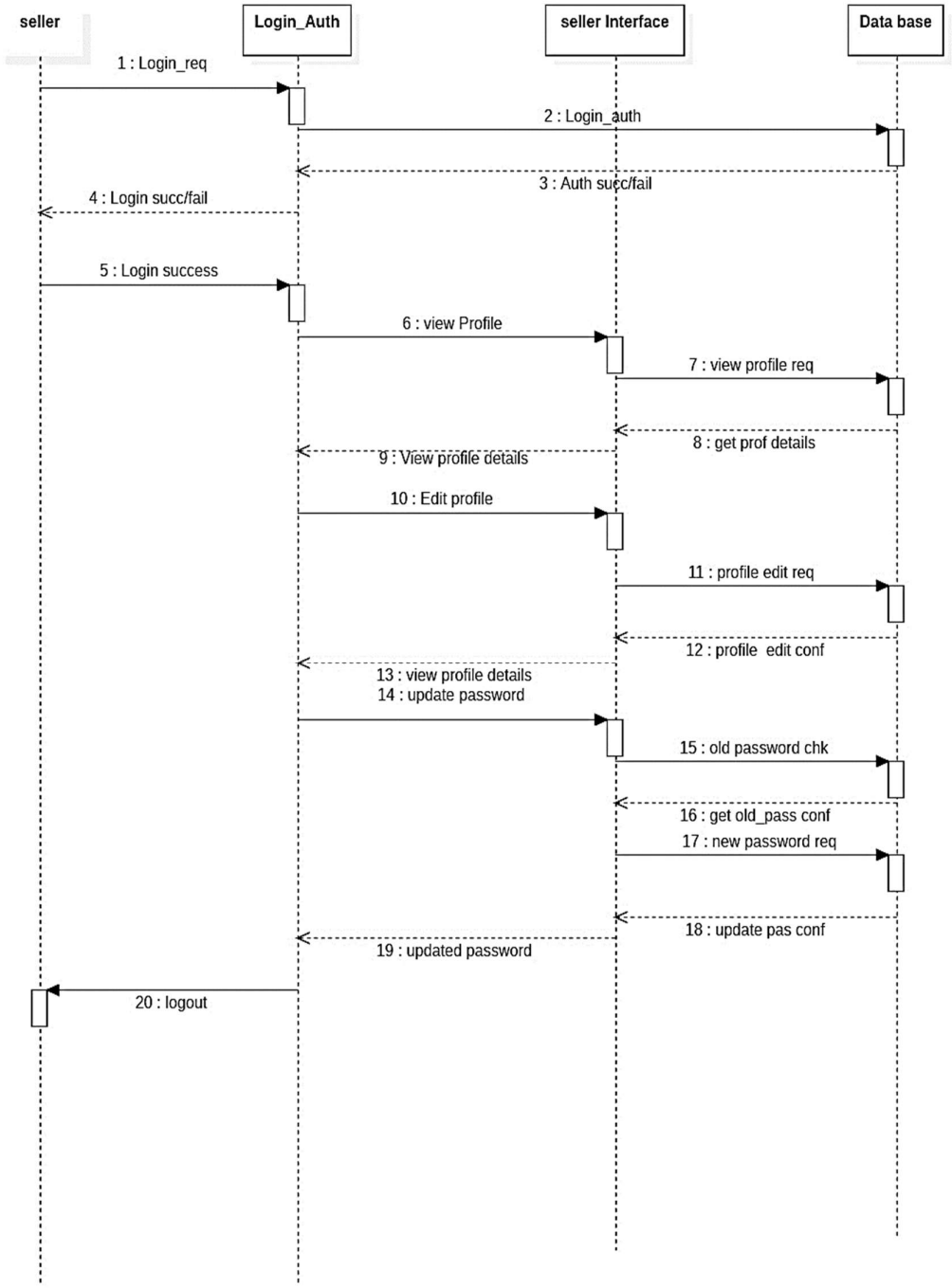

Figure 4.13 Seller-profile Sequence Diagram

4.3.7 Seller-item Sequence Diagram

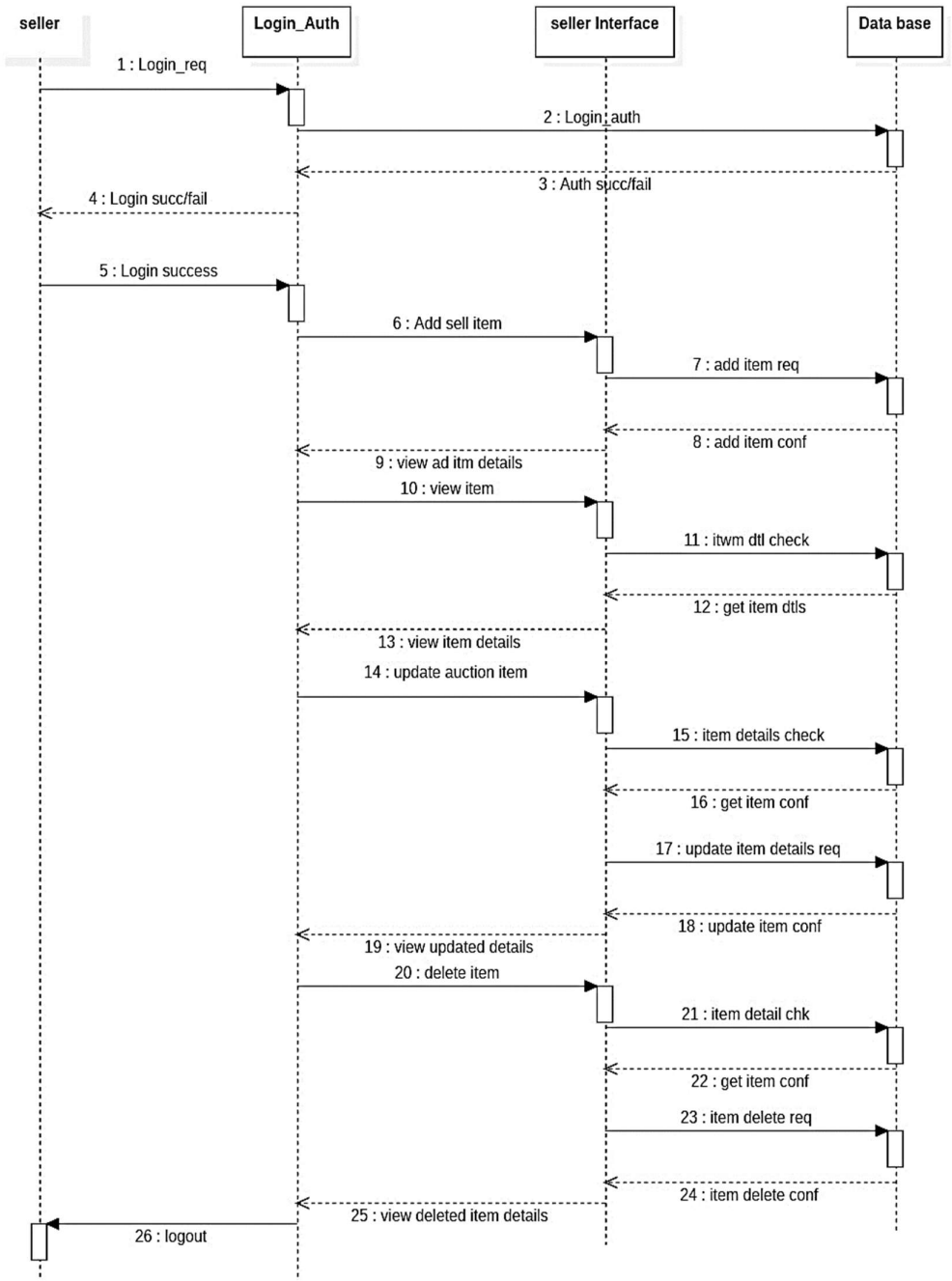

Figure 4.14 Seller-item Sequence Diagram

4.3.8 Seller-Auction Sequence Diagram

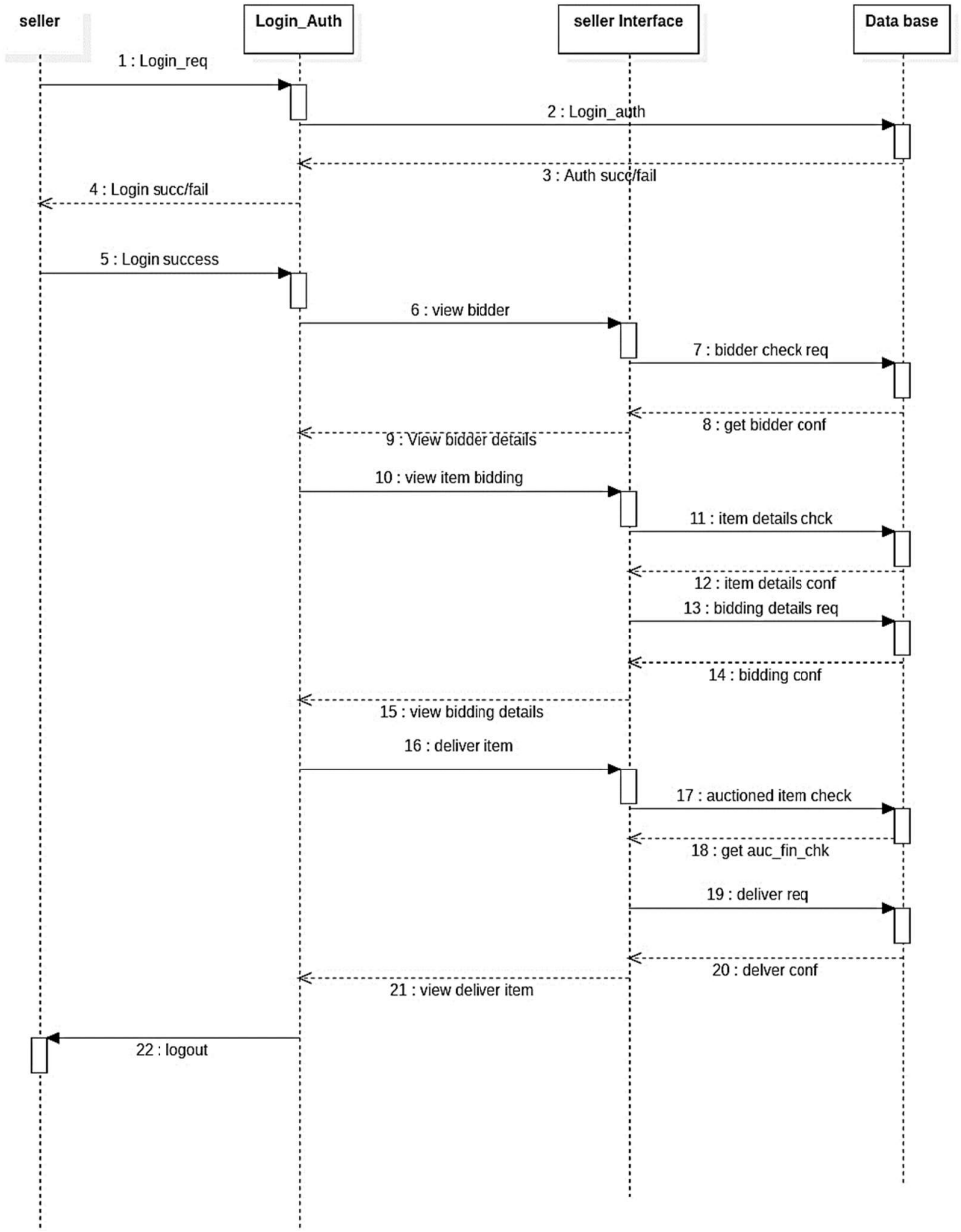

Figure 4.15 Seller-auction Sequence Diagram

4.3.9 Buyer-Profile Sequence Diagram

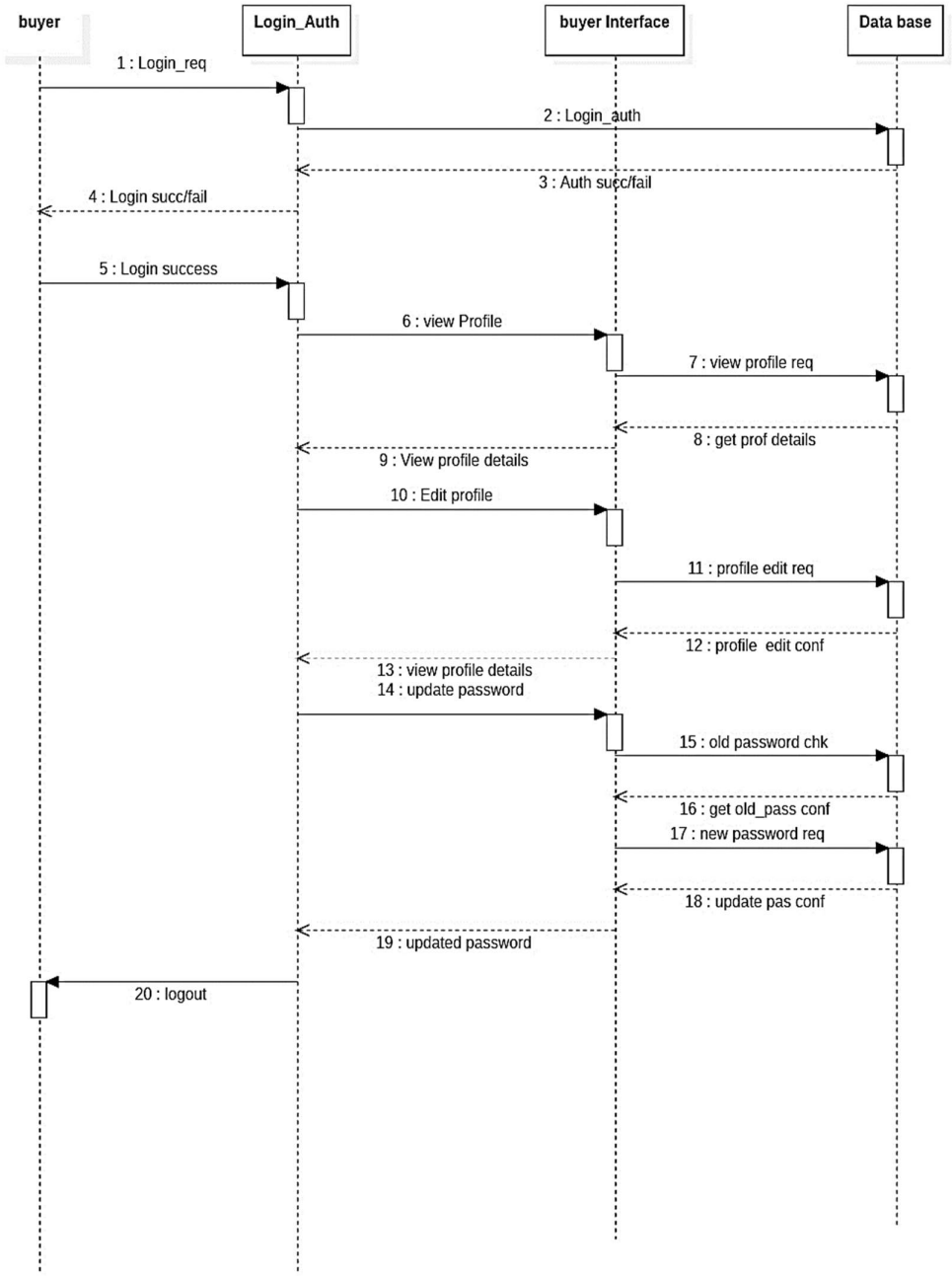

Figure 4.16 Buyer-Profile Sequence Diagram

4.3.10 Buyer-Auction Sequence Diagram

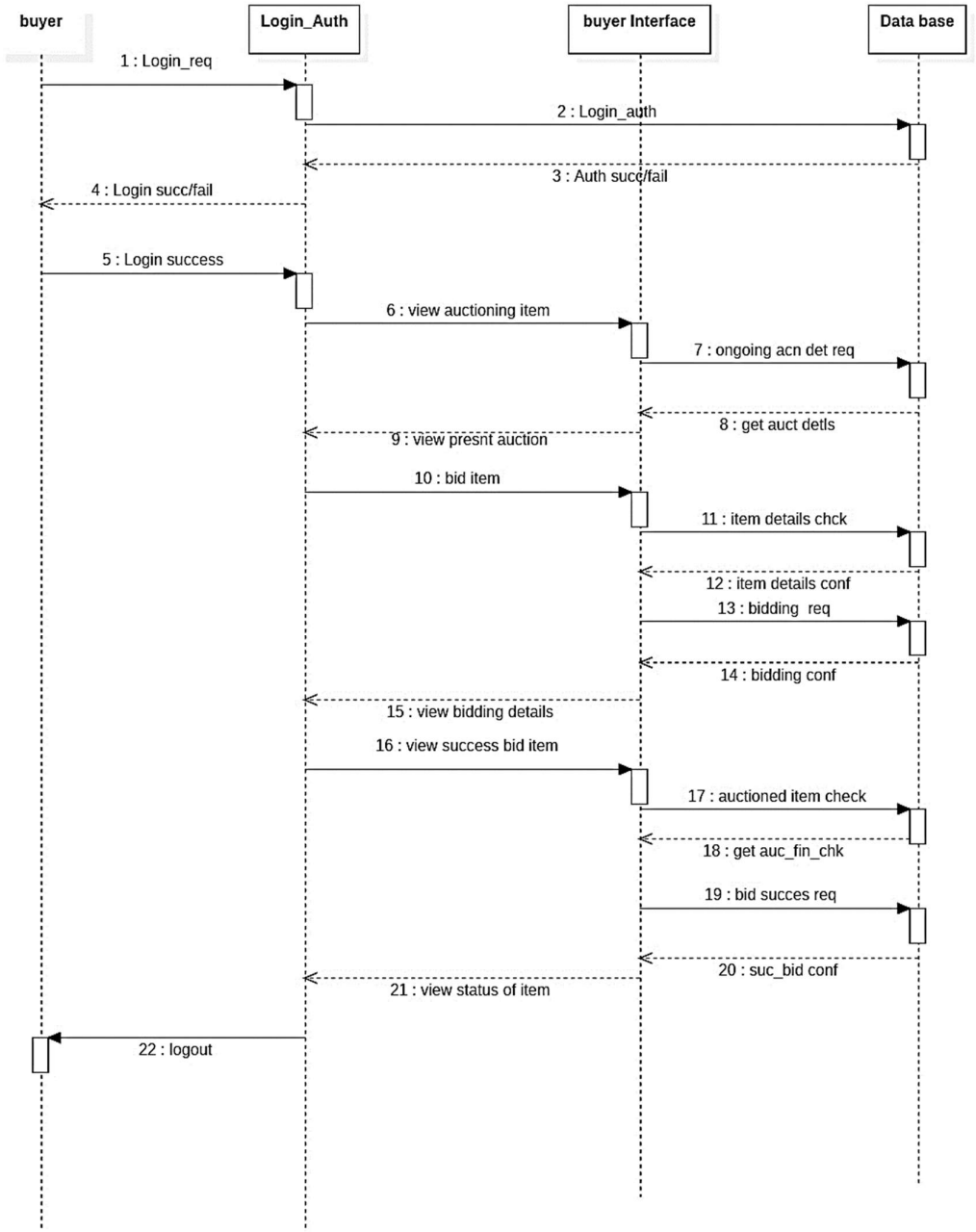

Figure 4.17 Buyer-Auction Sequence Diagram

CHAPTER V
BACKEND DESIGN

5.1 E-R Diagram:

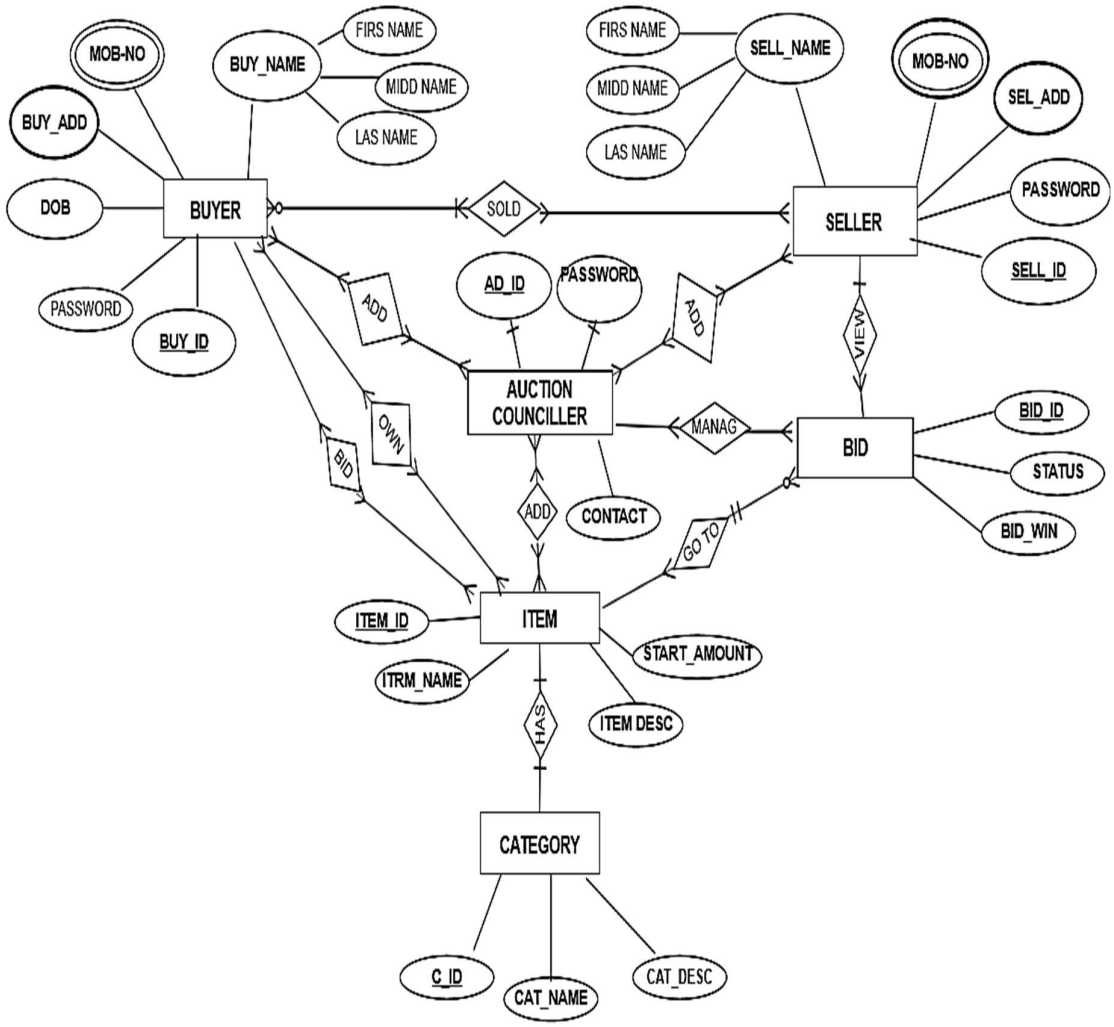

Figure 5.1 E-R Diagram

5.2 DATA DICTIONARY

5.2.1 LoginDetails:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	LOGINID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	PWD	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	FNAME	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)
4	MNAME	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)
5	LNAME	VARCHAR2(50 BYTE)	Yes	(null)	5	(null)
6	DOB	DATE	Yes	(null)	6	(null)
7	EMAIL	VARCHAR2(50 BYTE)	Yes	(null)	7	(null)
8	CNO	VARCHAR2(50 BYTE)	Yes	(null)	8	(null)
9	LTYPE	VARCHAR2(50 BYTE)	Yes	(null)	9	(null)

Table 5.1 Logindetails Table

5.2.2 AdminProfile:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	LOGINID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	PWD	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	FNAME	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)
4	MNAME	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)
5	LNAME	VARCHAR2(50 BYTE)	Yes	(null)	5	(null)
6	DOB	DATE	Yes	(null)	6	(null)
7	EMAIL	VARCHAR2(50 BYTE)	Yes	(null)	7	(null)
8	CNO	VARCHAR2(50 BYTE)	Yes	(null)	8	(null)
9	LTYPE	VARCHAR2(50 BYTE)	Yes	(null)	9	(null)

Table 5.2 AdminProfile Table

5.2.4 SellerProfile:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	LOGINID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	PWD	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	FNAME	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)
4	MNAME	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)
5	LNAME	VARCHAR2(50 BYTE)	Yes	(null)	5	(null)
6	DOB	DATE	Yes	(null)	6	(null)
7	GENDER	VARCHAR2(10 BYTE)	Yes	(null)	7	(null)
8	EMAIL	VARCHAR2(50 BYTE)	Yes	(null)	8	(null)
9	CNO	VARCHAR2(50 BYTE)	Yes	(null)	9	(null)
10	LTYPE	VARCHAR2(50 BYTE)	Yes	(null)	10	(null)
11	ADDR	VARCHAR2(50 BYTE)	Yes	(null)	11	(null)
12	CITY	VARCHAR2(50 BYTE)	Yes	(null)	12	(null)
13	PINC	VARCHAR2(50 BYTE)	Yes	(null)	13	(null)
14	COUNTRY	VARCHAR2(50 BYTE)	Yes	(null)	14	(null)
15	STATE	VARCHAR2(50 BYTE)	Yes	(null)	15	(null)
16	STATUS	VARCHAR2(20 BYTE)	Yes	(null)	16	(null)

Table 5.3 SellerProfile Table

5.2.4 BuyerProfile:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	LOGINID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	PWD	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	FNAME	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)
4	MNAME	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)
5	LNAME	VARCHAR2(50 BYTE)	Yes	(null)	5	(null)
6	DOB	DATE	Yes	(null)	6	(null)
7	GENDER	VARCHAR2(10 BYTE)	Yes	(null)	7	(null)
8	EMAIL	VARCHAR2(50 BYTE)	Yes	(null)	8	(null)
9	CNO	VARCHAR2(50 BYTE)	Yes	(null)	9	(null)
10	LTYPE	VARCHAR2(50 BYTE)	Yes	(null)	10	(null)
11	ADDR	VARCHAR2(50 BYTE)	Yes	(null)	11	(null)
12	CITY	VARCHAR2(50 BYTE)	Yes	(null)	12	(null)
13	PINC	VARCHAR2(50 BYTE)	Yes	(null)	13	(null)
14	COUNTRY	VARCHAR2(50 BYTE)	Yes	(null)	14	(null)
15	STATE	VARCHAR2(50 BYTE)	Yes	(null)	15	(null)
16	STATUS	VARCHAR2(20 BYTE)	Yes	(null)	16	(null)

Table 5.4 BuyerProfile Table

5.2.5 ItemMaster:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ITEMID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	ITEMNAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	CATEGORYNAME	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4	ITEMDESC	VARCHAR2(100 BYTE)	Yes	(null)	4	(null)
5	STARTPRICE	NUMBER	Yes	(null)	5	(null)
6	BIDINCREME...	NUMBER	Yes	(null)	6	(null)
7	SELLERID	VARCHAR2(50 BYTE)	Yes	(null)	7	(null)
8	BUYERID	VARCHAR2(50 BYTE)	Yes	(null)	8	(null)
9	BIDCOUNT	NUMBER	Yes	(null)	9	(null)
10	STATUS	VARCHAR2(50 BYTE)	Yes	(null)	10	(null)
11	IMAGE	BLOB	Yes	(null)	11	(null)

Table 5.5 ItemMaster Table

5.2.6 CategoryMaster:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CATEGORYNAME	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	CATEGORYABBR	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	CATEGORYDESC	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)

Table 5.6 CategoryMaster Table

5.2.7 BiddingInformation:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BIDDERID	NUMBER	No	(null)	1	(null)
2	LOGINNAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	BAMOUNT	NUMBER	Yes	(null)	3	(null)
4	ITEMID	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)
5	BIDDERDATE	DATE	Yes	(null)	5	(null)
6	STATUS	VARCHAR2(50 BYTE)	Yes	(null)	6	(null)
7	BIDCOUNT	VARCHAR2(20 BYTE)	Yes	(null)	7	(null)

Table 5.7 BiddingInofrmation Table

5.2.8 LoginAudit:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	LOGINID	VARCHAR2(20 BYTE)	Yes	(null)	1	(null)
2	LOGINDATE	DATE	Yes	(null)	2	(null)
3	LOGINDESC	VARCHAR2(200 BYTE)	Yes	(null)	3	(null)

Table 5.8 LoginAudit

CHAPTER VI
CODE SAMPLES

6.1 Index page:

Index.jsp:

```
<HTML>
        <HEAD>
                <TITLE>Online Auctioning Home </TITLE>
                <link  href="css/home.css" rel="stylesheet" type="text/css">
</HEAD>
        <BODY>
        <header>
          <div class="main">
           <div class="search">
             <form>
              <input type="text" placeholder="search....">
              <button>search</button>
             </form>  </div>
           <div class="menu">
            <ul>
        <li class="active"><a href='index.jsp'>Home</a></li>
        <li><a href='LoginForm.jsp'>Login</a></li>
        <li><a href='Registerform.jsp'>Register</a></li>
        <li><a href='product.jsp'>Product</a>
          <ul>
             <li><a href="#">Electronic </a></li>
             <li><a href="#">Fashion & style</a></li>
             <li><a href="#">House & Furniture </a></li>
             <li><a href="#">Vechile </a></li>
             <li><a href="#">Art</a></li>
           </ul>          </li>
        <li><a href='help.jsp'>Help</a></li>
      </ul>
      <div class="tittle">
          <h1>ONLINE AUCTION</h1>
          <th><h2>Buy and Bid your product here</h2></th>
      </div>
           </div>   </div>  </header></BODY> </HTML>
```

Home.css:

```
*{
        margin: 0px;
        padding: 0px;
        font-family:arial;
        }
body{
        background-image:linear-
gradient(rgba(0,0,0,0.6),rgba(0,0,0,0.4)),url(../images/back.jpg);
        height: 100vh;
        background-size: cover;
        background-position: center;
}
ul{
        float:left;
        list-style-type: none;
        list-style: none;
        margin-right: 0px;
}
ul li
{
 display: inline-block;
 width:170px;
 height:35px;
 text-align: center;
 opacity:1.8;
}
ul li a
{
        text-decoration: none;
        color: white;
        padding: 5px 20px;
        border: 1px solid transparent;
        transition:0.6s ease;`
}
ul li a:hover
{
        background-color:white;
        color: black;
}
   ul li ul li{
        display:none;
}
ul li:hover ul li
{
        display:block;
   padding: 5px 20px;
```

```
}
.search{
        float:right;
}
.search input[type=text],select{
  padding: 12px 20px;
  margin: 8px ;
  margin-top:8px;
  display: inline;
  border: 1px none;
 font-size: 17px;
 box-shadow: blue;
 color:white;
 background-color:transparent;
}
.search button{
         padding: 12px 20px;
  margin: 8px ;
  margin-top:8px;
  display: inline;
  border: 1px none;
 font-size: 17px;
 box-shadow: blue;
 background-color:transparent;
 color: white;
 transition:0.6s ease;`
}
.search button:hover{
        background-color: white;
        color: black;
}
 ul li.active a{
        background-color: white;
        color: black;
}
.main{
        max-width: 1200px;
        height:720px;
        margin:auto;
}
.tittle{
        position: absolute;
        top: 50%;
        left: 50%;
        transform:translate(-50%,-50%);
}
```

```
.tittle h1{
   font-size:70px;
   color: rgb(220,220,220);
background: transparent;
text-shadow: 1px 0px 1px rgb(0,206,209), 0px 1px 1px rgb(0,206,209),
2px 1px 1px rgb(0,206,209), 1px 2px 1px #EEEEEE, 3px 2px 1px #CCCCCC,
 2px 3px 1px #EEEEEE, 4px 3px 1px #6495ED, 3px 4px 1px #EEEEEE,
 5px 4px 1px #6495ED, 4px 5px 1px #EEEEEE, 6px 5px 1px #6495ED,
 5px 6px 1px #6495ED, 7px 6px 1px rgb(0,206,209);

 }
 .tittle h2{
  color:rgb(0,206,209);
  text-align:center;
  font-size:25px;
 }
```

6.2 Login page:

Login.jsp:

```
<HTML>
        <HEAD>
                <TITLE>Login page </TITLE>
                <link href="css/login.css"  rel="stylesheet" type="text/css">

</HEAD>

        <body>
        <header>
     <div class="head">
       <ul>
         <li><a href='index.jsp'>Home</a></li>
         <li><a href='Registerform.jsp'>Register</a></li>
         <li><a href='help.jsp'>Help</a></li>
      </ul><br><br><br><br>
      </div>
      <div class="tittle">
     <h1>Login</h1>
     <h2>Please enter correct USER Id and PASSWORD</h2>
    </div>
    <!--  --<div class="image">
     <img  src="images/aa.jpg" >
    </div>-->
        </header>
        <form   action="loginAction.do"   name   ="login"   onsubmit="return
validate()">
        <center>
```

```
        <div class="table">
          <p>User Name</p>
        <div class="textbox1">
       <input type="text" name=username placeholder="**user id**">
      </div>
      <p>Password</p>
      <div class="textbox2">
       <input type="password" name=password placeholder="**password**">
      </div>
      <div class="btn">
        <input type="submit" name=send value=Login>
       <div id="l"> <a href="ForgetPassword.jsp">Forget password</a>
     </div>
    </div>
    </center>
    </form>
    <%if(request.getParameter("status")!=null) {%>
                                <font                          color="white"><b>
<%=request.getParameter("status") %></b></font>
                                <%} %>
                                <script                  language="JavaScript"
type="text/javascript">
//You should create the validator only after the definition of the HTML form
  var frmvalidator  = new Validator("login");


  frmvalidator.addValidation("username","req","Please    enter    your    User
Name");

  frmvalidator.addValidation("password","req","Please enter your Password");
  </script>
  </body>
  </HTML>
```

Login.css:

```
*{
        margin: 0px;
        padding: 0px;
        font-family: century Gothic;
}
body{
        background-color: rgba(0.1,0.1,0.1,0.1);

}
ul{
        float:left;
        list-style-type: none;
        margin-top: 20px;
        list-style: none;
```

```
}
ul li
{
 display: inline-block;
 width:100px;
 height: auto;
}
ul li a
{
        text-decoration: none;
        color: white;
        padding: 5px 20px;
        border: 1px solid transparent;
        transition:0.6s ease;`
}
ul li a:hover
{
        background-color:white;
        color: black;
}

.textbox1 input[type=text], select {
  padding: 12px 20px;
  margin: 8px ;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}
.textbox2 input[type=password], select {
  padding: 12px 20px;
  margin: 8px ;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}
.btn input[type=submit], select {
  color:white;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
   background-color: mediumseagreen;
}
```

```
 input[type=submit]:hover{
  background-color: blue;
 }
.head{
        background-color: rgba(0,0,0,0.4);
}
.tittle h1{
        text-align: center;
        color:aqua;
        font-size: 50px;
}
.tittle h2{
        text-align: center;
        color: red;
        font-size: 25px;
        font-family: cursive;
}
.image img{
        width:25%;
        height:25%;
        margin-left:35%;
        background-size: cover;
        align-content: center;

        }
.table p{
        color: white;
}
center{
        background-image:linear-
gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(../images/aa.jpg);
        width:50%;
        height:50%;
        margin-left:25%;
        background-size: cover;
        align-content: center;
        position:fixed;
}
.table{
        margin-top: 15%;
}
```

LoginAction.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
   <%@ page
import="com.dts.dae.dao.ProfileDAO,com.dts.dae.model.Profile,com.dts.dae.d
ao.SecurityDAO"%>

<%
        String path = request.getContextPath();
String basePath =
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPor
t()+path+"/";
%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Auction</title>
<meta http-equiv="pragma" content="no-cache">
                <meta http-equiv="cache-control" content="no-cache">
                <meta http-equiv="expires" content="0">
                <meta http-equiv="keywords"
content="keyword1,keyword2,keyword3">
                <meta http-equiv="description" content="This is my page">
</head>
<body>
<br>
                <%
                        String target = "LoginForm.jsp?ls=Invalid username and
password";
                                try{
                                 Profile  rb=new Profile();
                                 String username =
request.getParameter("username");
                        rb.setLoginid(username);
                        rb.setPassword(request.getParameter("password"));

                        String role=new
SecurityDAO().loginCheck(username,request.getParameter("password"));
                        if(role.equals("Admin"))
                        {
                          target = "AdminHome.jsp?status=Welcome "+username;
                          session.setAttribute("user",username);
                          session.setAttribute("role",role);
                        }
                        else if(role.equals("seller"))
                        {
```

```
                    target = "SellerHome.jsp?status=Welcome "+username;
                        session.setAttribute("user",username);
                        session.setAttribute("role",role);
                }


  else if(role.equals("buyer"))
                {
                  target = "CustomerHome.jsp?status=Welcome
"+username;
                  session.setAttribute("user",username);
                  session.setAttribute("role",role);
                }
                else{
                  target = "LoginForm.jsp?ls=Invalid username and
password";
                }
                        }
                catch(Exception e){e.printStackTrace();
                out.println("I have some error please login again");
            %>
            <jsp:include page="LoginForm.jsp"></jsp:include>
            <%
                  }
                RequestDispatcher rd =
request.getRequestDispatcher(target);
                rd.forward(request,response);
                %>
        </body>
</html>
```

LoginAction.java:

```
package com.dts.project.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.dts.dae.dao.SecurityDAO;


public class LoginAction extends Action {
```

```
	public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request,
		HttpServletResponse response) {



		try {
			String username = request.getParameter("username");

			String password = request.getParameter("password");
			HttpSession session = request.getSession();
			String role = new SecurityDAO().loginCheck(username, password);

			System.out.println("this is Role=" + role);
			if (role.equalsIgnoreCase("Admin")) {

				// target = "AdminHome.jsp?status=Welcome "+username;
				session.setAttribute("user", username);
				session.setAttribute("role", role);
				return mapping.findForward("admin");
			} else if (role.equals("seller")) {
				// target = "SellerHome.jsp?status=Welcome "+username;
				session.setAttribute("user", username);
				session.setAttribute("role", role);

				return mapping.findForward("seller");
			}

			else if (role.equals("buyer")) {
				session.setAttribute("user", username);
				session.setAttribute("role", role);

				return mapping.findForward("buyer");
			} else {
				return mapping.findForward("invalid");
			}

		} catch (Exception e) {
			e.printStackTrace();

			return mapping.findForward("invalid");
		}
	}
}
```

SecurityDAO.java:

```
package com.dts.dae.dao;
import com.dts.dae.model.Profile;
import com.dts.core.dao.AbstractDataAccessObject;
import com.dts.core.util.DateWrapper;
import com.dts.core.util.LoggerManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Date;
public class SecurityDAO extends AbstractDataAccessObject {
	Connection con;
	private String desc;
	private boolean flag;

	/** Creates a new instance of SecurityDAO */
	public SecurityDAO() {
		// getting Database Connection
		con = getConnection();

	}

	// Compare Password
	public boolean checkPassword(Profile regbean) {
		String loginid = regbean.getLoginid();
		String oldpassword = regbean.getPassword();

		try {
			PreparedStatement pst = con.prepareStatement("select *
from logindetails where password=? and loginid=?");
			pst.setString(1, oldpassword);
			pst.setString(2, loginid);
			ResultSet rs = pst.executeQuery();
			if (rs.next()) {
				flag = true;
				LoggerManager.writeLogInfo("Login Success");
			} else {
				flag = false;
			}
		} catch (SQLException ex) {
			LoggerManager.writeLogSevere(ex);
```

```
                LoggerManager.writeLogInfo("Database Connection
problem");
                flag = false;
            }
            return flag;
        }

        // Compare Password
        public int checkFirstLogin(String loginname) {
            int fstatus = 2;
            try {

                con.setAutoCommit(true);
                PreparedStatement pst = con.prepareStatement("select
firstlogin from logindetails where loginid=?");
                pst.setString(1, loginname);
                ResultSet rs = pst.executeQuery();

                if (rs.next()) {
                    fstatus = rs.getInt(1);
                }
                con.close();
            } catch (SQLException ex) {
                LoggerManager.writeLogSevere(ex);
                LoggerManager.writeLogInfo("Database Connection
problem");
                flag = false;
            }
            return fstatus;
        }

        // Login Check
        public String loginCheck(String loginid, String password) {
            // String loginid=regbean.getLoginID();
            // String password=regbean.getPassword();
            String role = "";
            ResultSet rs = null;
            try {
                con = getConnection();

                PreparedStatement pst = con
                        .prepareStatement("select logintype from
logindetails where password=? and loginname=? ");
                pst.setString(2, loginid);
                pst.setString(1, password);

                rs = pst.executeQuery();
                if (rs.next()) {
```

```
                        role = rs.getString(1);
                        desc = "Login Success";
                    } else {
                        flag = false;
                        desc = "Entered Invalid password";
                        role = rs.getString(0);
                    }

                } catch (SQLException ex) {
                    LoggerManager.writeLogSevere(ex);
                    desc = "Database Connection problem";
                    flag = false;
                    role = "invalid";
                }
                loginaudit(loginid, desc);
                return role;
            }

            // Method for login audit
            public void loginaudit(String loginid, String desc) {
                try {
                    String newdate = DateWrapper.parseDate(new Date());
                    PreparedStatement pst = con.prepareStatement("insert
into loginaudit values(?,?,?)");
                    pst.setString(1, loginid);
                    pst.setString(2, newdate);
                    pst.setString(3, desc);
                    int i = pst.executeUpdate();
                    if (i == 1)
                        con.commit();
                    else
                        con.rollback();
                    con.close();
                } catch (Exception e) {
                    try {
                        con.rollback();
                    } catch (SQLException ex) {
                        LoggerManager.writeLogSevere(ex);
                    }
                    e.printStackTrace();
                }
            }

            // Change Password
            public boolean changePassword(Profile regbean) {
                String loginid = regbean.getLoginid();
                String oldpassword = regbean.getPassword();
                String newpassword = regbean.getNewpassword();
```

```
		try {
			con.setAutoCommit(false);
			String newdate = DateWrapper.parseDate(new Date());
			PreparedStatement pst = con.prepareStatement(
				"UPDATE logindetails SET
password=?,passmodifieddate=? WHERE loginname=? and password=?");

			pst.setString(1, newpassword);
			pst.setString(2, newdate);
			pst.setString(3, loginid);
			pst.setString(4, oldpassword);

			int i = pst.executeUpdate();
			if (i == 1) {
				flag = true;
				con.commit();
			} else {
				flag = false;
				con.rollback();
			}
			con.close();
		} catch (SQLException ex) {
			LoggerManager.writeLogSevere(ex);
			flag = false;
			try {
				con.rollback();
			} catch (SQLException sex) {
				LoggerManager.writeLogSevere(sex);
			}

		} catch (Exception e) {
			e.printStackTrace();
			flag = false;
			try {
				con.rollback();
			} catch (SQLException sex) {
				LoggerManager.writeLogSevere(sex);
			}
		}
		return flag;
	}
}
```

6.3 Registration Page:

RegisterForm.jsp:

```
<HTML>
        <HEAD>
                <TITLE>Registration page </TITLE>
     <link href="css/reg.css" rel="stylesheet" type="text/css">
</HEAD>

<body>
 <div class=head>
<ul><br>
  <li><a href='index.jsp'>Home</a></li>
  <li><a href='LoginForm.jsp'>Login</a></li>
  <li><a href='help.jsp'>Help</a></li>
</ul>
<br>
 <br>
  <h1>Registration Form</h1>
  <h2>Please Register to our familar</h2>
</div>
   <form action="reg" method="post" name="register" onSubmit="return validation()">
   <!--  --<form action="registerAction.do" method="post" name="register" onSubmit="return validate()" >-->
   <center>
    <table>
    <br><br>
    <div class="h">Personal Details</div>
    <br><br>
      <span id="massage" style="color: red ;font-weight: bold"></span>
    <tr>
     <th><td>Login Id</td><td><input type="text" name="loginid" id="lid" value="" autocomplete="off"
     pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}" title="Login Id must be contain One upper case, one lower case and one number" required="This Should be filled" style="width: 173px; "></td>
      <td style="width: 240px; "><h3 style="color:red">**Please Enter One Upercase,one lowecase and one number</h3></span></td>
     <tr>
    <tr>
     <th><td>Password</td><td><input type="password" name="password" id="pwd" value="" autocomplete="off" placeholder="password" required="This Should be filled" style="width: 173px; "
      pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,16}" title="Password most contain atleast one number,one Upercase and lowercase letter and its contain min-8 and max-16 more character"></td>
      <td style="width: 240px; "><h4 style="color:red">**Please Enter One Upercase,one lowecase and one number</h4></span>
```

```
      </td>
     <td>confirm  pasword</td><td><input  type="password"  name="cpwd"
id="cpwd"  value=""  autocomplete="off"   placeholder="confirm  password"
required="This Should be filled" style="width: 173px;"></td></th><th></th>
       <td style="width: 240px; "></td>
    </tr>
   <tr>
     <th><td>First   name</td><td><input   type="text"   name="firstname"
placeholder="firstname" autocomplete="off" required="This Should be filled"
style="width: 173px; "></td>
       <td style="width: 240px; "></td>
        <td>Middle   name</td><td><input   type="text"   name="middlename"
placeholder="middlename" autocomplete="off" style="width: 173px;"></td>
          <td style="width: 240px;"></td>
          <td>last    name</td><td><input    type="text"    name="lastname"
placeholder="lastname" autocomplete="off" required="This Should be filled"
style="width: 173px;"></td>
           <td style="width: 240px; "></td>
   </tr>
   <tr>
    <th><td>DOB</td><th><input  type="date"  name="bdate"    id="dob"
value="" autocomplete="off" style="width: 173px; " required="This Should be
filled" min="1919-01-01" max="2000-12-31">
      <td style="width: 240px; ">
       </td>
    <td>Gender</td>
    <td>
      <input type="radio"name="gender" value="M" required="Please select
gender">male
      <input  type="radio"name="gender"value="F"  required="Please  select
gender">female
    </td>
     <td style="width: 240px; "></td>
    <td>E-Mail</td>
    <td><input    type="email"    name="email"    id="email"    value=""
autocomplete="off" style="width: 173px;"
    pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$" ></td>
     <td style="width: 240px; "></td>
   </tr>
   <tr>
    <th><td>Contact number</td>
    <td><input   type="tel"   name="mobnumber"      id="cno"   value=""
autocomplete="off"  required="This  Should  be  filled"  pattern=".{10,10}"
title="Please Enter 10 digt number" style="width: 173px;">
   <td style="width: 240px; "></td>
    <td>Registration Type</td>
    <td>
    <select name="type" required="please select the type" id="type" >
```

```
        <option value="" >(Selcet)</option>
        <option >buyer</option>
        <option >seller</option>
    </select>
    </td>
   </tr>
  </table>
  <br><br>

  <table align="center" >
  <div class="h">Adress Details</div>
 <br><br>
   <tr>
    <th><td>Address</td><td><textarea rows="2" name="address" required="required"style="width: 173px;" required="This Should be filled"></textarea></td>
    <td>City</td><td><input type="text" name="city" style="width: 173px;" required="This Should be filled"></td>
    <td>Pincode</td><td><input type="text" name="pincode"style="width: 173px;" id="pin" required="This Should be filled" min="100000" max="999999"></td>
   </tr>
   <tr>
    <th><td>Country</td>
       <td> <input type="text" readonly="readonly" name="country" value="India"> </td>
          <td>State</td>
          <td>
          <select name="state" required="required">
            <option value="">select</option>
            <option> Andhra Pradesh</option>
            <option>Assam</option>
            <option>Gujarat</option>
            <option>Haryana</option>
            <option>Jharkhand </option>
            <option>Karnataka</option>
            <option>Kerala</option>
            <option>Madhya Pradesh</option>
            <option>Maharashtra </option>
            <option>Meghalaya</option>
            <option>Odisha</option>
            <option>Punjab</option>
            <option>Tamil Nadu</option>
            <option>Telangana </option>
            <option>Uttar Pradesh</option>
            <option>West Bengal</option>
            </select>
          </td>
```

```
            </tr>
            <div class="btn">
                <tr align="right">
    <td style="width: 90px; height:90px; "><input name="Input" type="submit"
value="Register">
    </td>
    </tr>
      </div>
    </table>
    </center>
        </form>
<script type="text/javascript">
  function validation(){

            var pwd=document.register.password.value;
            var cpwd=document.register.cpwd.value;

            }
            if (cpwd == pwd) {
                    return true;
          }
            else{
                    alert("Password and confirmpassword should be mattched");
                    return false;
            }

  }
</script>
</BODY>
</HTML>
```

Registration.java:

```
package com.auction.reg;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import javax.jws.WebService;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebService
public class Registration extends HttpServlet {
        private static final long serialVersionUID = 1L;
```

```
protected void doPost(HttpServletRequest req, HttpServletResponse
res) throws ServletException, IOException {

        res.setContentType("text/html");
        int i;
        RequestDispatcher rd = null;
        Connection con;
        PreparedStatement ps;
        PrintWriter out = res.getWriter();
        String loginid = req.getParameter("loginid");
        String password = req.getParameter("password");
        String firstname = req.getParameter("firstname");
        String middlename = req.getParameter("middlename");
        String lastname = req.getParameter("lastname");
        String dob = req.getParameter("bdate");
        String gender = req.getParameter("gender");
        String email = req.getParameter("email");
        String cno = req.getParameter("mobnumber");
        String type = req.getParameter("type");
        String addr = req.getParameter("address");
        String city = req.getParameter("city");
        String pincode = req.getParameter("pincode");
        String country = req.getParameter("country");
        String state = req.getParameter("state");
        String status = "n";
        try {

            Class.forName("oracle.jdbc.driver.OracleDriver");
            con                                                    =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"eauction", "prajwal");
            if (type.equalsIgnoreCase("seller")) {
                ps = con.prepareStatement(
                        "insert into SELLERPROFILE
values(?,?,?,?,?,TO_DATE(?,'yyyy-MM-dd'),?,?,?,?,?,?,?,?,?,?)");
                ps.setString(1, loginid);
                ps.setString(2, password);
                ps.setString(3, firstname);
                ps.setString(4, middlename);
                ps.setString(5, lastname);
                ps.setString(6, dob);
                ps.setString(7, gender);
                ps.setString(8, email);
                ps.setString(9, cno);
                ps.setString(10, type);
                ps.setString(11, addr);
                ps.setString(12, city);
```

```
                        ps.setString(13, pincode);
                        ps.setString(14, country);
                        ps.setString(15, state);
                        ps.setString(16, status);
                        i = ps.executeUpdate();
                } else {
                        ps = con.prepareStatement(
                                "insert into BUYERPROFILE
values(?,?,?,?,?,?,TO_DATE(?,'yyyy-MM-dd'),?,?,?,?,?,?,?,?,?,?,?)");
                        ps.setString(1, loginid);
                        ps.setString(2, password);
                        ps.setString(3, firstname);
                        ps.setString(4, middlename);
                        ps.setString(5, lastname);
                        ps.setString(6, dob);
                        ps.setString(7, gender);
                        ps.setString(8, email);
                        ps.setString(9, cno);
                        ps.setString(10, type);
                        ps.setString(11, addr);
                        ps.setString(12, city);
                        ps.setString(13, pincode);
                        ps.setString(14, country);
                        ps.setString(15, state);
                        ps.setString(16, status);
                        i = ps.executeUpdate();

                }
                if (i == 0) {
                        rd                                              =
req.getRequestDispatcher("unsucessreg.jsp");
                        rd.forward(req, res);
                }

                else {
                        rd = req.getRequestDispatcher("Successreg.jsp");
                        rd.forward(req, res);

                }

        } catch (Exception e) {
                // TODO: handle exception
                 rd = req.getRequestDispatcher("ServerError.jsp");
                rd.forward(req, res);
        }

    }
}
```

6.4 AmdinHome:

AdminHome.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>ADMIN HOME</title>
</head>
<style>
#wel{
text-align:center;
color: #A7DD3C;
background:transparent;
text-shadow: 2px 0 0px #800040, 3px 2px 0px rgba(77,0,38,0.5),
3px 0 3px #FF002B, 5px 0 3px #800015, 6px 2px 3px rgba(77,0,13,0.5),
6px 0 9px #FF5500, 8px 0 9px #802A00, 9px 2px 9px rgba(77,25,0,0.5),
9px 0 18px #FFD500, 11px 0 18px #806A00, 12px 2px 18px rgba(77,66,0,0.5),
12px 0 30px #D4FF00, 14px 0 30px #6A8000, 15px 2px 30px rgba(64,77,0,0.5),
15px 0 45px #80FF00,
 17px 0 45px #408000, 17px 2px 45px rgba(38,77,0,0.5);
}
</style>
<body>
<%
String id=(String)session.getAttribute("user");
%>
<jsp:include page="AdminOption.jsp"></jsp:include>
<center>
<marquee direction="left" >

<div id="wel"><h1> WEL-COME TO ADMIN HOME </h1></div>
</marquee>
<jsp:include page="PendingBuyer.jsp"></jsp:include>
<jsp:include page="PendingSeller.jsp"></jsp:include>
</center>
</body>
</html>
```

AdminOption.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
```

```
<title>Insert title here</title>
        <link  href="css/adm.css" rel="stylesheet" type="text/css">
</head>
<body>
<ul ><br>
 <li><a href="AdminHome.jsp">Home</a></li>
 <li><a href="#">User</a>
     <ul>
     <li><a href="#">Seller</a>
       <ul>
        <li><a href="PendingSeller.jsp">Requests</a></li>
        <li><a href="Update.jsp">Update</a></li>
       </ul>
     </li>
     <li><a href="#">Buyer</a>
     <ul>
        <li><a href="PendingBuyer.jsp">Requests</a></li>
        <li><a href="Update.jsp">Update</a></li>
       </ul>
     </li>
     </ul>
 </li>
   <li><a>Auction</a>
   <ul>
   <li><a href="PendingItem.jsp">Requests</a></li>
   <li><a href="UpdateItem.jsp">Update</a></li>
   <li><a href="AdminAuctionItemsToday.jsp">View Auction</a></li>
   <li><a href="AdminAuctionItemsEndday.jsp">View Auction</a></li>
   </ul>
   </li>
   <li><a href='AdminProfile.jsp'>Profile</a></li>
    <li><a href='AdminPasswordUpdate.jsp'>Security</a>
          <ul>
          <li><a href="AdminPasswordUpdate.jsp">Update password</a></li>
          </ul>
         </li>
   <li><a href="LogoutAction.jsp">Logout</a></li>
</ul>
</body>
</html>
```

6.5 SellerHome:

SellerHome.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
```

```
<head>
<meta charset="ISO-8859-1">
<title>Seller home</title>
</head>
<style>
#wel{
text-align:center;
color: #A7DD3C;
background:transparent;
text-shadow: 2px 0 0px #800040, 3px 2px 0px rgba(77,0,38,0.5),
3px 0 3px #FF002B, 5px 0 3px #800015, 6px 2px 3px rgba(77,0,13,0.5),
6px 0 9px #FF5500, 8px 0 9px #802A00, 9px 2px 9px rgba(77,25,0,0.5),
9px 0 18px #FFD500, 11px 0 18px #806A00, 12px 2px 18px rgba(77,66,0,0.5),
12px 0 30px #D4FF00, 14px 0 30px #6A8000, 15px 2px 30px rgba(64,77,0,0.5),
15px 0 45px #80FF00,
 17px 0 45px #408000, 17px 2px 45px rgba(38,77,0,0.5);
}
</style>
<body>
<jsp:include page="Selleroption.jsp"></jsp:include>
<center>
<marquee>
<div id="wel"><h2>WEL-COME TO SELLER </h2></div>
</marquee>
</center>
</body>
</html>
```

SellerOption.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Seller page</title>
<link  href="css/seler.css" rel="stylesheet" type="text/css">
</head>
<body>
            <ul><br>
         <li class="active"><a href='SellerHome.jsp'>Home</a></li>
          <li><a href='Item.jsp'>Item</a>
            <ul>
              <li><a href="SellItem.jsp">Sell item</a></li>
              <li><a href="UpdateItem.jsp">Your item</a></li>
              <li><a href="UpdateItem.jsp">update item</a></li>
              <li><a href="DeleteItem.jsp">delete item</a></li>
            </ul>
         </li>
```

```
        <li><a href='#'>Auction</a>
            <ul>
             <li><a href="AuctionItemsToday.jsp">Recent Auction </a></li>
             <li><a href="LastAuction.jsp">Finished Auction</a></li>
             <li><a href="#">ViewBid</a></li>
             <li><a href="#">view status</a></li>
             <li><a href="Recipt.jsp">Deliver item</a></li>
            </ul>
        </li>
        <li><a href='SellerProfile.jsp'>Profile</a></li>
        <li><a href='SellerPasswordUpdate.jsp'>Security</a>
          <ul>
          <li><a href="SellerPasswordUpdate.jsp">Update password</a></li>
          </ul>
         </li>
        <li><a href='help.jsp'>Help</a></li>
        <li><a href="LogoutAction.jsp">Logout</a></li>
        </ul>
</body>
</html>
```

6.6 AddItem:

AddItem.jsp:

```
<%@page import="com.dts.core.util.CoreList"%>
<%@page import="java.util.Enumeration"%>
<%@page import="com.dts.project.dao.CategoryDAO"%>
<%@page import="com.dts.project.model.CategoryModel"%>
<HTML>
        <HEAD>
           <link  href="css/item.css" rel="stylesheet" type="text/css">
                <TITLE>Online Auctioning </TITLE>

</HEAD>
        <body background="white">
        <jsp:include page="Selleroption.jsp"></jsp:include>
                <div class="table">
                <center><h1>Add Your item</h1></center>
                <br><br>
                        <form action="ireg" method="post">
                        <center>
                        <label>Item name</label>
                        <div><input          type="text"          name="iname"
required="required"/></div>
                        <label>Item Description</label>
                        <div><input          type="text"          name="idsc"
required="required"/></div>
```

```
                        <label>Category</label>
                        <div><select name="category">
                        <option value="">select</option>
                        <option>Electronic</option>
                        <option>Tech.</option>
                        <option>Fashion</option>
                        <option>Style</option>
                        <option>House</option>
                        <option>Furniture</option>
                        <option>Vechile</option>
                        <option>Art</option>
                        </select></div>
                        <label>Starting Price</label>
                        <div><input type="text" name="isr"
required="required"/></div>
                        <label>Bid Increment price</label>
                        <div><input type="text" name="icr"
required="required"/></div>
                        <label><img id="blah" alt="plaese upload item pic"
width="120" height="120" /></label>
                        <div><input type="file" name="image"
accept="image/*" onchange="document.getElementById('blah').src =
window.URL.createObjectURL(this.files[0])"></div>
                        <br><br>
                        <div><input type="submit" name="submit"
value="Add"></div>
                        <div><input type="hidden" name="loginid" hidden=""
value="<%=(String)session.getAttribute("user") %>"/></div>
                        </center>
                        </div>
                        </form>
        </body>
</HTML>
```

6.7 BuyerHome: BuyerHome.jsp:

```
<HTML>
        <HEAD>
                <TITLE>Online Auctioning </TITLE>
</HEAD>
<style>
#wel{
text-align:center;
color: #A7DD3C;
background:transparent;
text-shadow: 2px 0 0px #800040, 3px 2px 0px rgba(77,0,38,0.5),
3px 0 3px #FF002B, 5px 0 3px #800015, 6px 2px 3px rgba(77,0,13,0.5),
```

```
6px 0 9px #FF5500, 8px 0 9px #802A00, 9px 2px 9px rgba(77,25,0,0.5),
9px 0 18px #FFD500, 11px 0 18px #806A00, 12px 2px 18px rgba(77,66,0,0.5),
12px 0 30px #D4FF00, 14px 0 30px #6A8000, 15px 2px 30px rgba(64,77,0,0.5),
15px 0 45px #80FF00,
 17px 0 45px #408000, 17px 2px 45px rgba(38,77,0,0.5);
}
</style>
<body>
<jsp:include page="BuyerOption.jsp"></jsp:include>
<center>
<marquee>
  <div id="wel"><h1>WEL-COME TO BUYER HOME </h1></div>
</marquee>
</center>
</body>
</HTML>
```

BuyerOption.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title></title>
<link  href="css/cus.css" rel="stylesheet"   type="text/css">
</head>
<body>
<ul><br>
   <li><a href="CustomerHome.jsp">Home</a> </li>
   <li><a href="Item.jsp">Item</a> </li>
    <li><a href="BuyerProfile.jsp">profile</a> </li>
       <li><a href='BuyerPasswordUpdate.jsp'>Security</a>
           <ul>
           <li><a href="BuyerPasswordUpdate.jsp">Update password</a></li>
           </ul>
          </li>
    <li><a href="LogoutAction.jsp">Logout</a> </li>
</ul>
</body>
</html>
```

CHAPTER VII
SCREENSHOT

7.1Home Page:

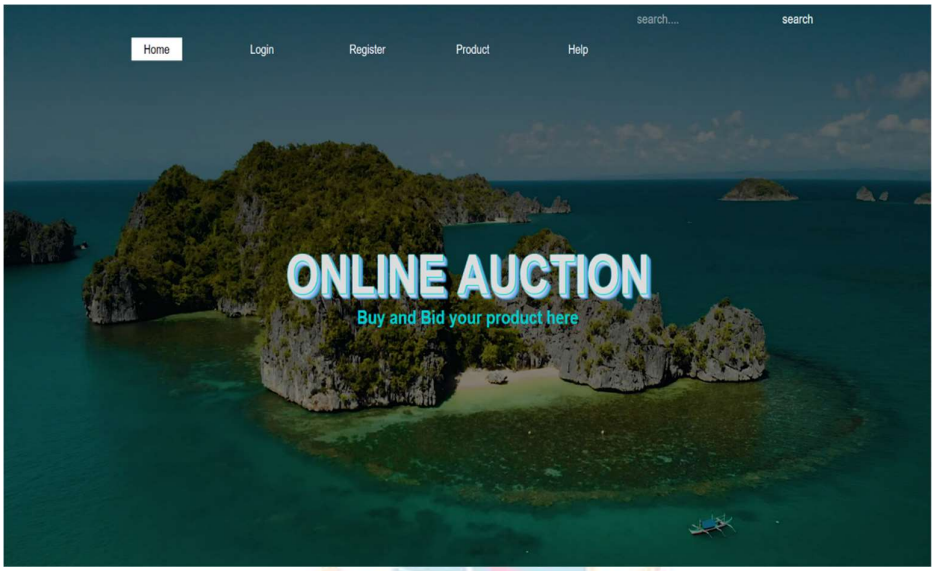

Figure 7.1 Home Page.

7.2 Login Page:

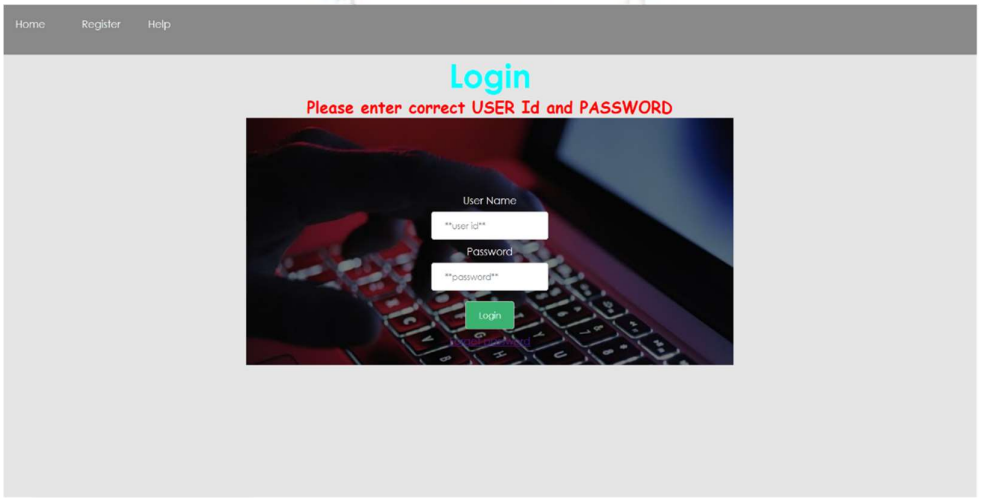

Figure 7.2 Login Page

7.3 Registration Page:

Figure 7.3 Registration Form.

7.4Admin Home:

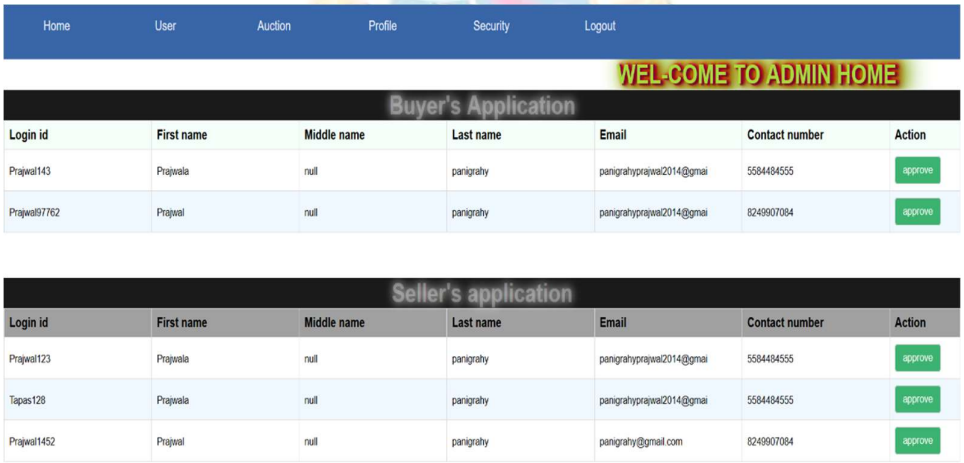

Figure 7.4Admin Home.

7.5Seller Home Page:

Figure 7.5 Seller Home Page.

7.6 Add Item:

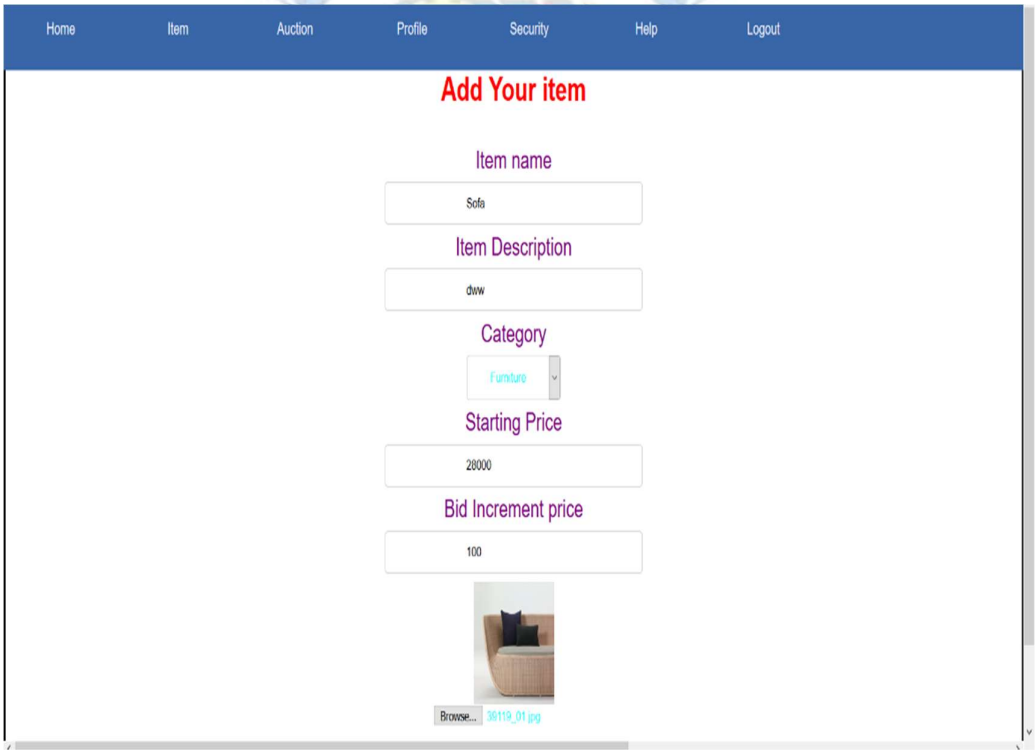

Figure 7.6 Add Item.

7.7 Buyer Home Page:

Figure 7.7 Buyer Home Page:

CHAPTER VIII
TESTING

Software Testing is the process used to help identify the correctness, completeness, security, and quality of developed computer software. Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind, testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behaviour of the product against a specification. An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (SQA), which encompasses all business process areas, not just testing.

At SDEI, 3 levels of software testing are done at various SDLC phases

- **Unit Testing**: in which each unit (basic component) of the software is tested to verify that the detailed design for the unit has been correctly implemented
- **Integration testing**: in which progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a whole.
- **System testing**: in which the software is integrated to the overall product and tested to show that all requirements are met

Test cases, suites, scripts and scenarios

A test case is a software testing document, which consists of event, action, input, output, expected result and actual result. Clinically defined (IEEE 829-1998) a test case is an input and an expected result. This can be as pragmatic as 'for condition x your derived result is y', whereas other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps (but often steps are contained in a separate test procedure that can be exercised against multiple test cases, as a matter of economy) but with one expected result or expected outcome. The optional fields are a test case ID,

test step or order of execution number, related requirement(s), depth, test category, author, and check boxes for whether the test is automatable and has been automated. Larger test cases may also contain prerequisite states or steps, and descriptions. A test case should also contain a place for the actual result. These steps can be stored in a word processor document, spreadsheet, database or other common repository. In a database system, you may also be able to see past test results and who generated the results and the system configuration used to generate those results. These past results would usually be stored in a separate table.

8.1. Test Cases

8.1.1 Positive Test Cases:

Test case-1: Admin Login

Field	Input	Output	Remark
Username	Admin	Launched to admin Portal	Test Case Working Successfully
Password	admin		

Table 8.1 Admin Login. (Positive Test Case)

Test case-2: Seller Login

Field	Input	Output	Remark
Username	Prajwal976	Launched to Seller Portal	Test Case Working Successfully
Password	Prajwal143		

Table 8.2 Seller Login. (Positive Test Case)

Test case-3: Buyer Login

Field	Input	Output	Remark
Username	Prajwal143	Launched to buyer Portal	Test Case Working Successfully
Password	Prajwal123		

Table 8.3 Buyer Login. (Positive Test Case)

Test case 4: Seller Register

Field	Input	Output	remark
Login id	Ayush@143	Stored in the database	Test case worked successfully
Password	Pradhan977		
First Name	Ayush		
Middle name	Kumar		
Last name	Pradhan		
DOB	10/04/1998		
Email	ayush143@gmail.com		
Cno	9776208048		
Gender	Male		
Registration Type	Seller		

Table 8.4 Seller Register (positive test case)

Test case 5: Buyer Register

Field	Input	Output	remark
Login id	Tapas@1550	Stored in the database	Test case worked successfully
Password	Patra1455		
First Name	Tapas		
Middle name	Ranjan		
Last Name	Patra		
DOB	10/04/1999		
Email	patratapas123@gmail.com		
Cno	9776208048		
Gender	Male		
Registration Type	Seller		

Table 8.5 Buyer Register (positive test case)

8.1.2 Negative Test Cases:

Test case-1: Admin Login

Field	Input	Output	Remark
Username	Amann	Password incorrect	Test Case Successfully
Password	Aaa		

Table 8.6 Admin Login (Negative Test Case)

Test case-2: Seller Login

Field	Input	Output	Remark
Username	Prajwal97	Password incorrect	Test Case Successfuly
Password	72ddhvh		

Table 8.7 Seller Login .(Negative Test Case)

Test case-3: Buyer Login

Field	Input	Output	Remark
Username	Prajwal143	Password incorrect	Test Case Successfuly
Password	Jad6s16ff		

Table 8.8 Buyer Login .(Negative Test Case)

Test case 4: Seller Register

Field	Input	Output	remark
Login id	Ayush@143	Duplicate found, can't store in database	Test case worked successfully
Password	Pradhan977		
First Name	Ayush		
Middle name	Kumar		
Last name	Pradhan		
DOB	10/04/1998		
Email	Ayush143@gmail.com		
Cno	9776208048		
Gender	Male		
Registration Type	Seller		

Table 8.9 Seller Register (negative test case)

Test case 5: Buyer Register

Field	Input	Output	remark
Login id	Tapas@1550	Duplicate found , can't store in database	Test case worked successfully
Password	Patra1455		
First Name	Tapas		
Middle name	Ranjan		
Last Name	Patra		
DOB	10/04/1999		
Email	patratapas123@gmail.com		
Cno	9776208048		
Gender	Male		
Registration Type	Seller		

Table 8.10 Buyer Register (positive test case)

CHAPTER IX
CONCLUSION AND FUTURE DIRECTIONS

Conclusion:

The online auction system, that we build is a model, where the user can participate in a bid for products and service. This auction is made easier by using online software, what we have developed in this project. which can regulate processes involved in the basic activity of bidding, selling and buying. There are several different auction methods and user actually decide, easily to buy a product through our bidding system services. This system has been designed to be highly-scalable and capable of supporting large numbers of bidders in an active auction.

It should be healthy and will be a good practice when it is made more transparent as a matter of fact. Online Bidding has become more wide spread in all sorts of industrial usage mostly at a commercially profitable work. It not only includes the product or goods to be sold, it also has services which can be provided to user to carry out the operation smoothly. Due to their low cost this expansion made the system to grow and prosper. Online bidding has become a standard method for procurement process. Bidders can be managed to gain optimal solution for selling and buying process.

Future Scope Enhancement:

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

1. It can be possible, to make transactions through secure payment gateways.
2. Payment gateway, security, authentication and confidential issues must be handled.
3. As the technology emerges, it is possible to upgrade the system and proper maintenance activities to be implemented for long run of system.
4. Code optimization, database normalization and some other areas still require heavy modifications.
5. Guest module can be added, just to view some features.
6. An in-built web browser can be added.

CHAPTER X
REFERENCES

1. J. BLOCH, Effective Java–Programming Language Guide, Addison-Wesley, Boston, Massachusetts, 2018.
2. E. FREEMAN AND E. FREEMAN, Head First HTML with CSS & XHTML, O'Reily Media, Sebastopol, California, 2015.
3. C.S. HORTSMANN AND G. CORNELL, Core Java Volume II–Advanced Features, Prentice Hall, Upper Saddle River, New Jersey, 2018.
4. F. MARCHIONI, JBoss AS 5 Development, Packt Publishing, Birmingham, England, 2019.
5. H. MARTY, More Servlets and Java Server Pages, Pearson Education, Upper Saddle River, New Jersey, 2011.
6. N. RICHARDS AND S. GRIFFITH, JBoss–A Developer's Notebook, O'Reilly Media, Sebastopol, California, 2016.
7. D.THAU, The Book of JavaScript 2nd Edition: A Practical Guide to Interactive Web Pages, No Starch Press, San Francisco, California, 2016.
8. J. TREVATHAN AND W. READ, Detecting collusive shill bidding, in Proceedings of the Fourth International Conference on Information Technology: New Generations, Las Vegas, Nevada, 2017.
9. W. WANG, H. ZOLTÁN, AND A.B. WHINSTON, Shill bidding in multi-round online auctions, in Proceedings of the 35th Hawaii International Conference on System Sciences, Big Island, Hawaii, 2012.