PROJECT REPORT

VIT®

UNIVERSITY

(Estd. u/s 3 of UGC Act 1956)

VELLORE ■ CHENNAI

www.vit.ac.in

COURSE: PROGRAMMING IN JAVA
COURSE CODE:SWE1007
SLOT : B2+TB2

FIFA AUCTIONS

16MIS0240 – M.SAI DARSHINI
16MIS0244 – HARRIS JOHN
16MIS0319 – P.KOMAL VAISHNAVI
16MIS0404 – POOJITHA
16MIS0496 – KOWSIK B

CONTENTS:

CHAPTER-1
Software Requirements Specification

Document

Table of Contents

1. Introduction

FIFA is a famous series of football video games,released annually by Electronic Arts under the EA Sports label and played by millions of football heads in the world.As in real life the game tasks the player with controlling a football team.The player controls one of the eleven footballers on their team at a time, with the ability to switch players on command. The game allows up to four human players at the same time, each controlling a different footballer using wireless or wired controllers and keyboard. The players can choose to control a footballer on the same team or on opposing teams. The remaining footballers are controlled by the computer.The players are fictional.The objective of the game is to score by moving the ball using virtual players beyond the goal line into the opposing goal.The team that scores most goals by the end of the match wins.It features real-time 3D graphics that present players with real player names and positions, with ranking, transfer and team customisation tools.It have a refined graphics engine, team and player customisation options,stadiums, improved artificial intelligence with all FIFA-registered national teams,clubs and a licensed soundtrack featuring popular musical artists of the time.Each team has its

own detailed kit,club emblems and for players,their own unique faces.But the game can only be played with existing national squads or club squads.Even though here are team customization options and options for multiple users to choose same team,the game lack an auctioning system where the user can bid for his/her favorite player and make him play for his/her team alone.'FIFA AUCTIONS' is a standalone JAVA application which allows the gamers to conduct friendly auctions using virtual money 'outside' the game.Random players appear in the application and users can bid for their favorite ones.The data of players bought get stored in the database and using the report generated,team customisations can be made 'inside' the game.'FIFA AUCTIONS' helps gamers create their dream team and ease their winnings in multiplayer scenarios!

1.1 Purpose

The purpose of this project is to shed light on online auctioning system and how virtual currencies work.

1.2 Scope

This project involves creating a free standalone JAVA application for online auctioning.User can be anyone and the number also doesn't matter.The details of users are collected and already stored in the database.Every user will be provided with initial virtual amount which obviously cannot be cashed.The player listing which is obtained from FIFA official site is the manuscript.The auction starts when a random player is displayed in the application.The complete details,from personal to professional,is displayed.The player goes to the highest bidder.Not only the user improve his trading skills but also get familiarised with the football world,mainly who the kingpins are.Because football is a game which the world never misses.The application improves the management skills,both money and diplomacy of users who are involved in auctioning.Grab the best without losing the bet is the policy.The project has much more significance as the year has witnessed the word cup!

1.3 Definitions, Acronyms, and Abbreviations.

FIFA. : Fédération Internationale de Football Association(International Federation of Association Football in English)
EA : Electronic Arts

LB : Left Back Player
LWB : Left Wing Back Player
RB: : Right Back Player
RWB : Right Wing Back Player
CB : Centre Back player
LM : Left Mid Player
RM : Right Mid Player
CM : Centre Mid Player
CAM : Centre Attacking Mid Player
CDM : Centre Defensive Mid Player
CF : Centre Forward Player
LW : Left Wing Player
RW : Right Wing Player
LS : Left Footed Striker
RS : Right Footed Striker
ST : Striker

auction:a trade in which goods or property are sold to the highest bidder.
base value: thecurrent market value of a player
bidder:a person or organization making a formal offer for something, especially at an auction.
nationality:the status of belonging to a particular nation.
rating:a classification or ranking of someone or something based on a comparative assessment of their quality, standard, or performance.

1.5 Overview

In our architecture, there are 3 main elements:
• The client tier, that is responsible for the presentation of data, receiving user
elements and controlling the user interface.
• The application server tier, that is responsible for the business logic of the system. In fact, business-objects that implement the business rules "live" here, and
are available to the client-tier. This tier protects the data from direct access by the
clients.

• The data server tier, that is responsible for data storage

2. The Overall Description

With the implementation of 'FIFA AUCTIONS' there will be no noisy crowds like conventional system where users have to sit and bid.It excludes general frustration that usually happens while bidding in conventional system.There will be no schedule constraint that means bidder can bid any time and from anywhere.

2.1 Product Perspective

The product is independent and totally self-contained.

2.2 Functional Requirements

Administrator: He is the super user responsible for managing clients of the system,taking system backup,generating reports,maintaining organiztion details

Manage Clients:The Administrator assigns new users when a new cient joins the online auctioning.Also he can delete an account when any of the user leave the auctioning organization

Maintain Corporate Details:The Administrator maintains entire details of the Corporate that includes details of the Corporate Admin and its sub user's details etc

Take System backup:The Administrator Backup the database inorder to prevent loss of data on system crashes.He can backup entire database or a particular section

Generate Reports: Responsible for checking the logs of different system users for auditing and maintaining he integrity of the system

2.3 Non-Functional Requirements

2.3.1 Hardware Interfaces

Several hardware platforms support FIFA AUCTIONS .An online auctioning system is effective when the correct software application and hardware configuration is used.

-Processor : Intel Pentium 4 or above
-Clock speed:800 MHz
-System bus: 32 bit
-RAM : 256 MB or above
-Hard disk : 160 GB or above
-Monitor: SVGA color
-Keyboard: 101 keys
-Modem:56 Kbps/ADSL broadband
-Mouse: PS2/Serial
-FDD:1.44 MB

Mid range systems are small,powerful and easy to maintain.Mid-range system hardware is less expensive than high-range equipment,has built-in,preventive maintenance utilities which automatically monitor central processing unit(CPU) usage and other hardware functions.Workstations can be integrated into the system and configured to process regular and Web auctioning made easy.

2.3.2 Software Interfaces

Development tools : Swings,JDBC
The system must use SQL Server as its database component. i.e.2.1.4.1 Microsoft SQL Server 7. Communication with the DB is through JDBC connections. The system must provide SQL data table definintions for setup.

2.3.3 Communications Interfaces

Client on Internet will be using HTTP/HTTPS protocol
Client on Intranet will be using TCP/IP protocol
A Web Browser such as IE 6.0 or equivalent

2.4 Constraints

- Regulatory policies
- Hardware limitations (for example, signal timing requirements)
- Interface to other applications
- Parallel operation
- Audit functions
- Control functions
- Higher-order language requirements
- Signal handshake protocols
- Reliability requirements
- Criticality of the application
- Safety and security considerations

3. Specific Requirements

- Users may participate in an auction in different roles:the auctioneer,the bidders and guests shall be admitted

- Different roles get different information at hand.Only the auctioneer can co-relate the bids to their bidders during the auction.Bidders appear to each other anonymously,but know how many competitors there are.

- The auction times may vary.Very short auctions may have an auction time as short as 15 minutes.Typical auction times are 1-3 hours,consisting of a main part and an extension part

- The auction time is extended whenever a bid arrives shortly before the auction end.This allows all other bidders to react.The provided reaction time may vary.e.g.starting from 3 minutes as an initial extension down to a few seconds at the very end.

- A report on the auction result is provided for all users

3.1 External interfaces

The application is accessible from any operating system provided it has a web browser and a connection to the web server.No special hardware is required by the end-user.The client browser must be W3C XHTML compatible Communication between the users and the database will be through HTTP communication using TCP/IP port 80.If an error occurs during a request,the user should receive a clear error message.These errors also be logged

3.2 Functions

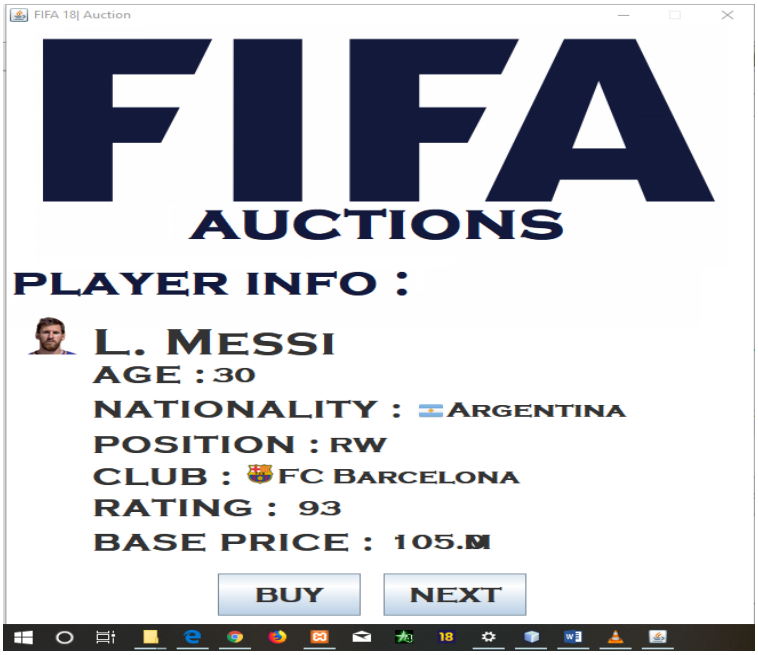
Auctions have a name, a description, possibly a photo (of the related item) uploaded by
users and an end period: users cannot place bids when the auction interval (start - end
period) ends, but, in case there were no offers for an item, there is the possibility to
extend the interval..Moreover, administrators have the possibility to accept or refuse auctions proposed by users, to view information about users and items and to create, modify and delete the categories of auctions.The system is realized with a 3-tier architecture: a relational database that store the information regarding items, users, auctions and categories of auction; an application server that cares about the business logic of the system and the presentation layer that consists in the web browser where users can interact with the system.

4. Future Works

The online auction portal works very well in all of its functionality. However, some
future works can be done on the existing system:

• Add a chat room. It would be nice for a user to enter in a chat room

to talk with other users about auctions or any other topic. This chat can be realized
using the Java Applet technology.
• Add a more attractive graphics



5. Remarks(if any)

6. Document Approvals

CHAPTER – 2

UML DIAGRAMS:

CONTENTS:

14.RATIONAL ROSE CODE

1.INTRODUCTION :

FIFA is a famous series of football video games,released annually by Electronic Arts under the EA Sports label and played by millions of football heads in the world.As in real life the game tasks the player with controlling a football team.The player controls one of the eleven footballers on their team at a time, with the ability to switch players on command. The game allows up to four human players at the same time, each controlling a different footballer using wireless or wired controllers and keyboard. The players can choose to control a footballer on the same team or on opposing teams. The remaining footballers are controlled by the computer.The players are fictional.The objective of the game is to score by moving the ball using virtual players beyond the goal line into the opposing goal.The team that scores most goals by the end of the match wins.It features real-time 3D graphics that present players with real player names and positions, with ranking, transfer and team customisation tools.It have a refined graphics engine, team and player customisation options,stadiums, improved artificial intelligence with all FIFA-registered national teams,clubs and a licensed soundtrack featuring popular musical artists of the time.Each team has its own detailed kit,club emblems and for players,their own unique faces.But the game can only be played with existing national squads or club squads.Even though here are team customization options and options for multiple users to choose same team,the game lack an auctioning system where the user can bid for his/her favorite player and make him play for his/her team alone.'FIFA AUCTIONS' is a standalone JAVA application which allows the gamers to conduct friendly auctions using virtual money 'outside' the game.Random players appear in the application and users can bid for their favorite ones.The data of players bought get stored in the database and using the report generated,team customisations can be made 'inside' the game.'FIFA AUCTIONS' helps gamers create their dream team and ease their winnings in multiplayer scenarios.

2.UNIFIED MODELING LANGUAGE (UML) DIAGRAMS

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 diagram, which is as follows.

3.DESCRIPTION FOR MODULES:

The following sections include the descriptions for modules.

ADMIN Module: This module provides the complete information related to players for bid and the buyers can bid for the players 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 player gets confirmed.

PLAYER 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 players and sell them.

PAYMENT Module: In this application buyer can buy any player from any part of the world at a very best competitive price and own the product.

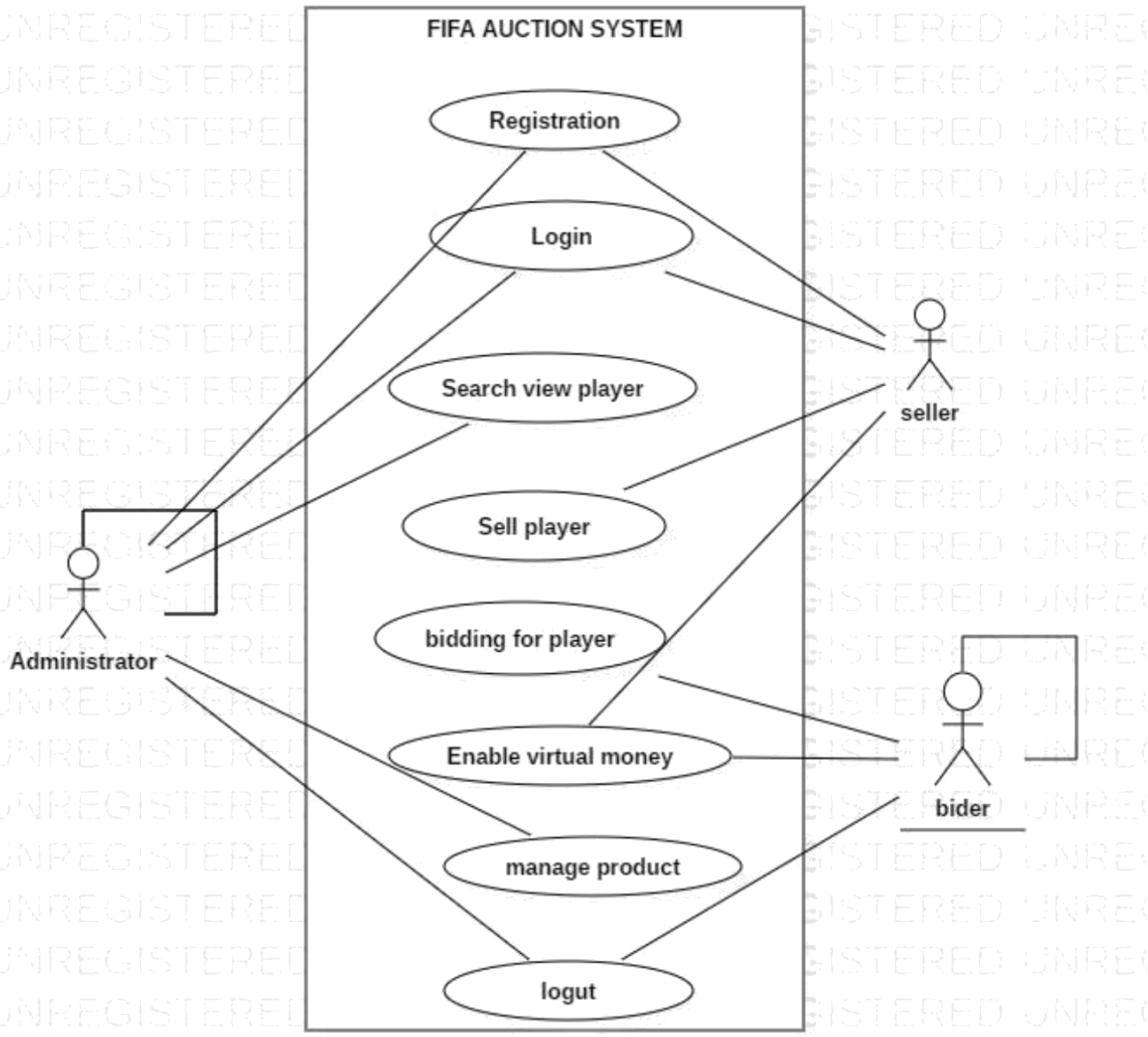
USER Module: Visitor is nothing but all the people who visits this application online. They can know the information of all the players, which are for bid under this application.

4.USE CASE DIAGRAM:

Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, buyer, seller…etc., or another system like central database. System here refers to the Online Auctioning system and the actors that are using the system are Admin and Customer.

Admin can view all the customers, view the bid items, view today and end day auction items and view the profiles of the customers. Customer registers, and upon successful login can buy or sell the items, can view his profile, can view the new auction items today and end day.


FIFA AUCTION SYSTEM
Registration
Login
Search view player
Sell player
bidding for player
Enable virtual money
manage product
logut
Administrator
seller
bider

5.ACTIVITY DIAGRAM:

From the admin home the following are the tasks that admin can perform .

 View Customers: Admin can view information of all the registered customers from this page.

 View Bid Items: Admin can view all the items that are up for bidding through this page.

 View Auction Items Today: Admin can view all the items that are up for auction for that particular day.

 View Auction Items End Day: Admin can view all the auction items whose auction date ends that day.

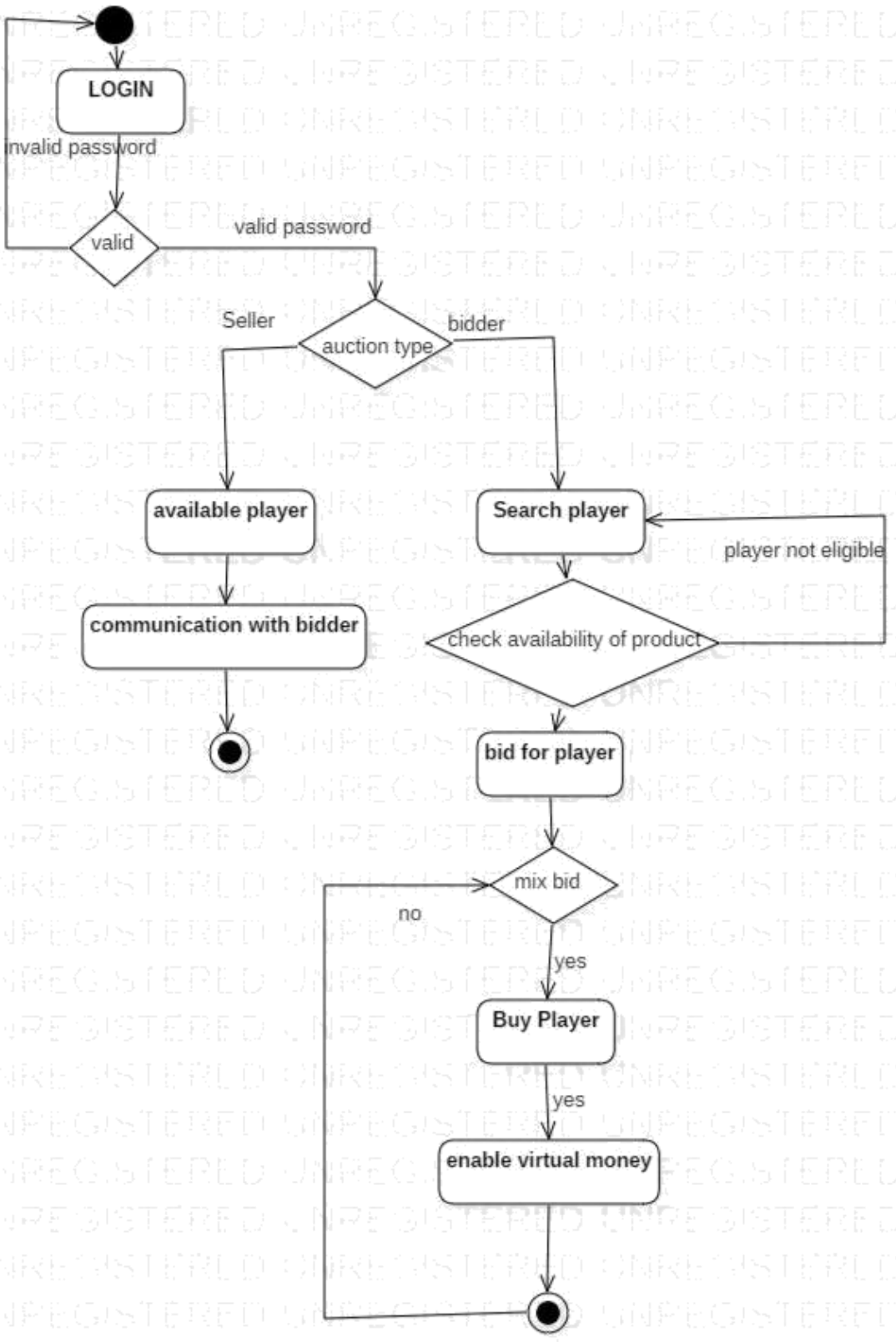
After providing login details for customer login the details will be validated and after successful validation it will be redirected to the customer home page. From the customer home the following are the tasks that customer can perform:

 Bid Item: customer can bid for the items available from this page.

 Sell Item: Customer can sell the items to aany of the registered customer from this page.

 View Auction Items Today: Admin can view all the items that are up for auction for that particular day.

 View Auction Items End Day: Admin can view all the auction items whose auction date ends that day.


LOGIN
invalid password
valid
valid password
auction type
Seller
bidder
available player
communication with bidder
Search player
player not eligible
check availability of product
bid for player
mix bid
no
yes
Buy Player
yes
enable virtual money

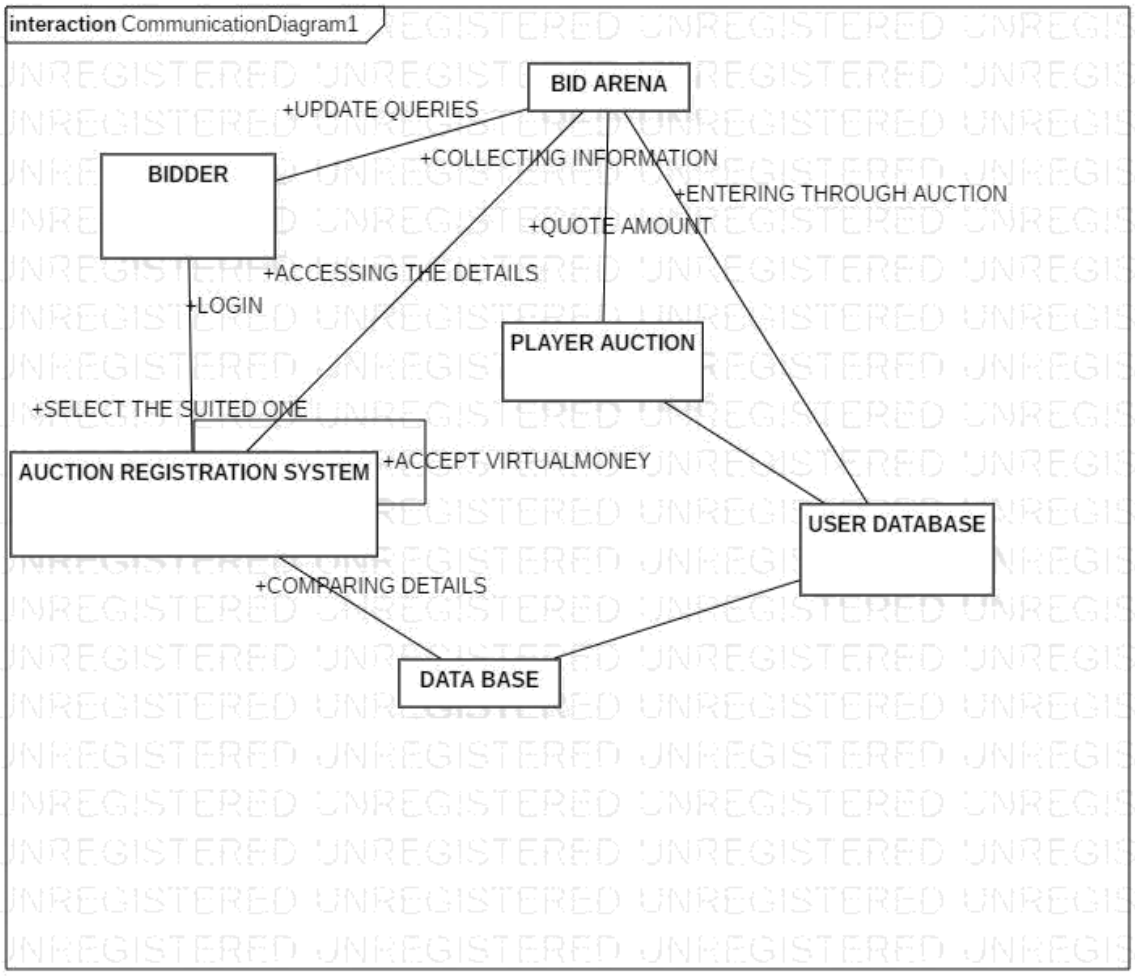
After successful validation of login credentials Admin will be redirected to the home page and from the home page he can add items for auction, view items that are up for auction, add categories, view categories, add bidding info and view bidding.

After successful validation of login credentials seller/owner will be redirected to the home page and from the home page he can view items, update or delete items, view categories, view bidding info, update bidding info and delete bidding info.

After successful validation of login credentials registered bidder will be redirected to the home page and from the home page he can view items, view categories and view bidding info.

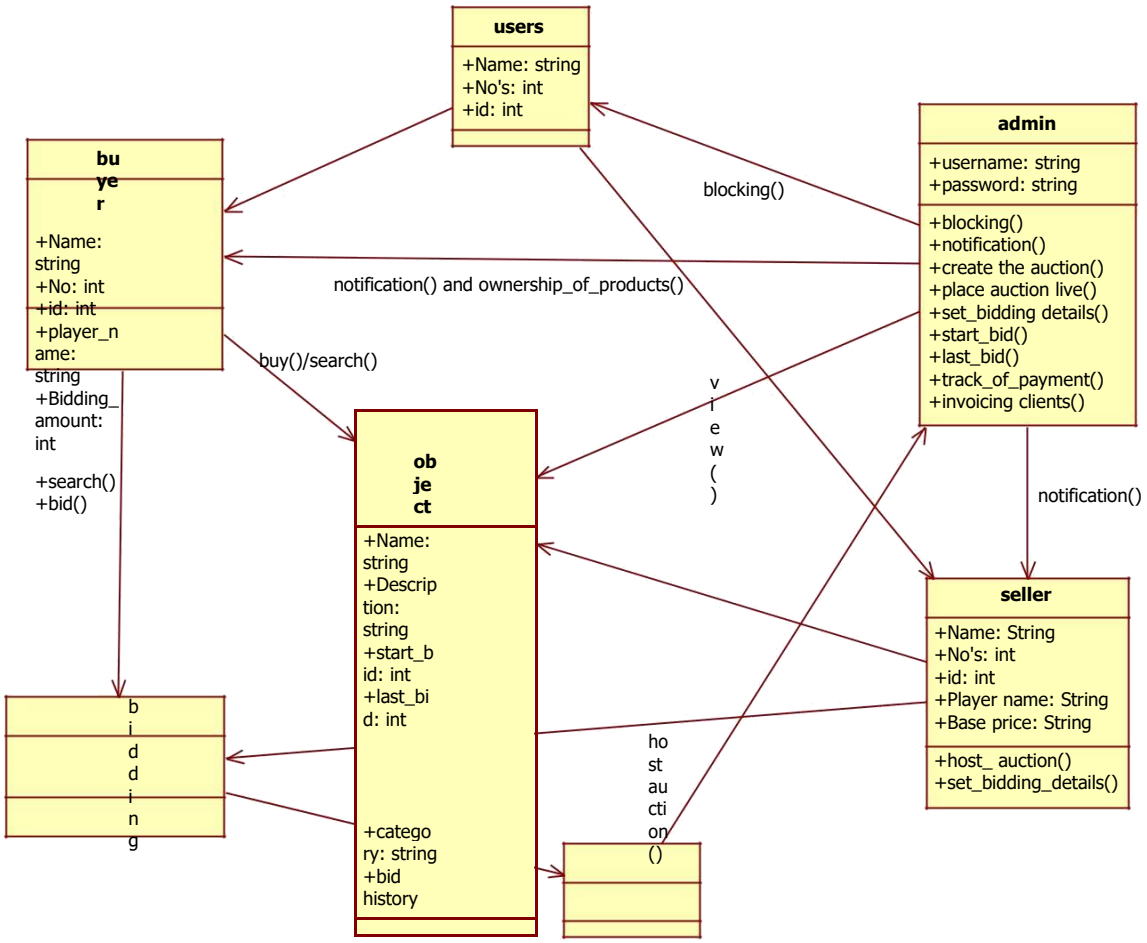
6.COLLABORATION DIAGRAM:

Admin after validating login information will be taken to the admin home page and from the home page he can view the customers info, bid items, auction items today and end day. Customer after login validation will be taken to the customer home page and from the home page he can bid for items, sell items, view auction items today and end day.



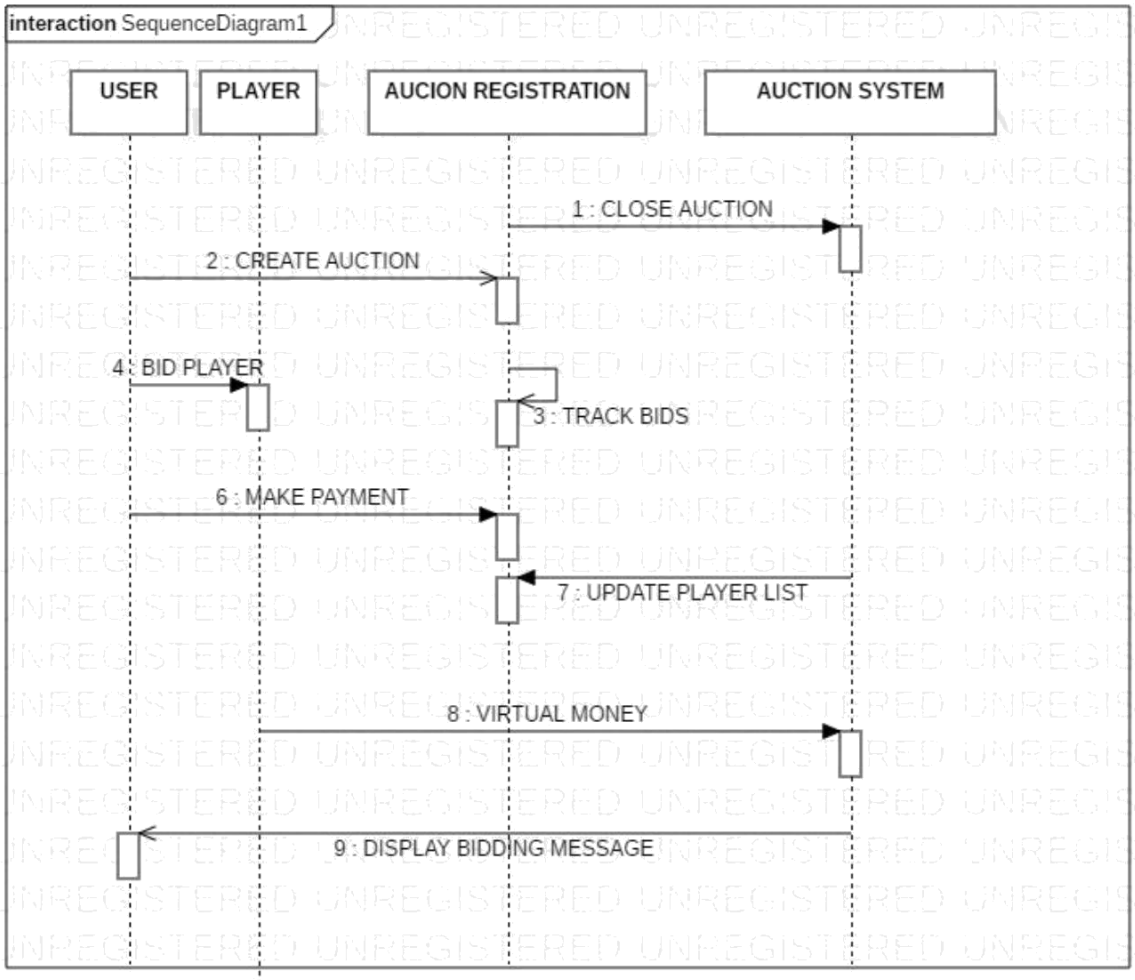
7.CLASS DIAGRAM:

The class User contains several parameters that are information about a normal user that browses the system (username, password, name, surname…); the parameter administrator is a flag that indicates whether the user is an administrator or not. This class performs the operations of a normal user (search for an Player, get the items on auction, post a new auction, send messages to other users) and those of a system administrator (create a new category, delete a category, delete an item, delete an user). The class Message has a text, a topic and the user_id of the sender and of the receiver. The class Item contains several information about the item on auction (the name of the player, its photo, a textual description of it..) and about the category it belongs to.


users
+Name: string
+No's: int
+id: int
admin
+username: string
+password: string
+blocking()
+notification()
+create the auction()
+place auction live()
+set_bidding details()
+start_bid()
+last_bid()
+track_of_payment()
+invoicing clients()
buyer
+Name: string
+No: int
+id: int
+player_name: string
+Bidding_amount: int
+search()
+bid()
blocking()
notification() and ownership_of_products()
buy()/search()
view()
notification()
object
+Name: string
+Description: string
+start_bid: int
+last_bid: int
+category: string
+bid history
seller
+Name: String
+No's: int
+id: int
+Player name: String
+Base price: String
+host_ auction()
+set_bidding_details()
bidding
host auction()
bid
+current_bid: int
+maximum_bid: int
+sort_bidding_amount()
track_of_payment()
mark bid as amount
payment
+amount_int

8.SEQUENCE DIAGRAM:

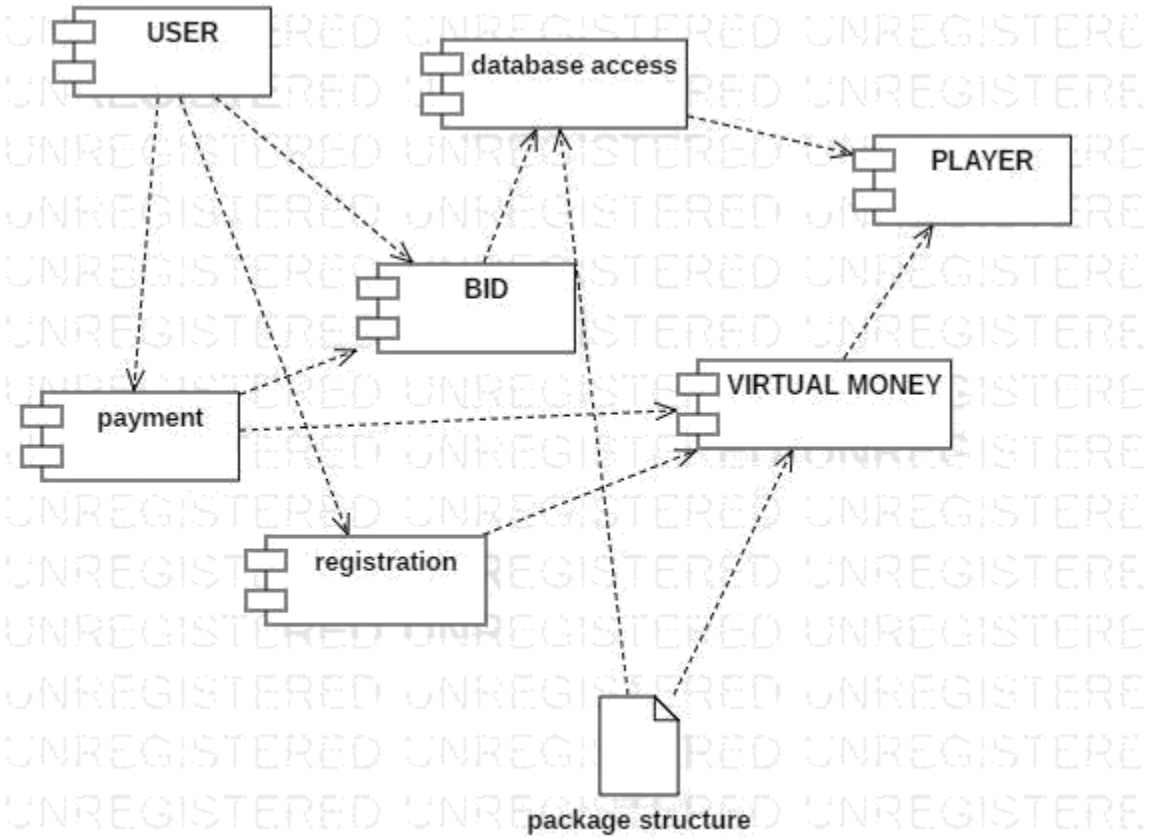
The Sequence Diagram depicts the overall flow of control in an object-oriented program. The top of the diagram shows the objects as boxes and the functions that define the behavior of the objects that belong to each class. The messages sent between objects for dynamic communication are represented as arrows from one object's lifeline to another. Time progresses downward and the sequence of interaction between different objects by passing of messages is shown. Thus, the Sequence Diagram represents the interactive relationship among objects visually and sequentially. Typically, the sequence diagram captures the behavior of a single use-case. Figure shows the Sequence Diagram for the Purchaser in the Auction System. The Purchaser logs into the Auction Mall with ID and Password to purchase products. If the user is successfully certified, the auction system displays the players on auction to the user. Next, if the Purchaser selects players and enters the Negotiation room for buying, he can select the bidding method to purchase the products. Further, the Purchaser decides the bidding price by viewing the history of the other bidder's previous auction records and the price that is suggested by the selected bidding method. Finally, the Purchaser that bids the highest price buys the players and the auction result is stored in the database. After that, the auction manager informs all bidders of the auction result and concludes the product's auction.



9.COMPONENT DIAGRAM:

The Component Diagram is a simple high-level diagram that is used to show how the code is actually divided between the different modules. It shows the dependencies among the software components. Figure illustrates the interactive relationship among the software components of the auction system. In our implementation, we made the entities that are used in the auction as components using EJB (Enterprise Java Beans). It is clear from the diagram that it is possible to modify relevant parts/components without affecting the entire system. The configuration of the whole system is easily understood looking at the Component

Diagram because it graphically depicts the interrelationships and dependencies (if any) between the various software components.



10.STATE CHART DIAGRAM:

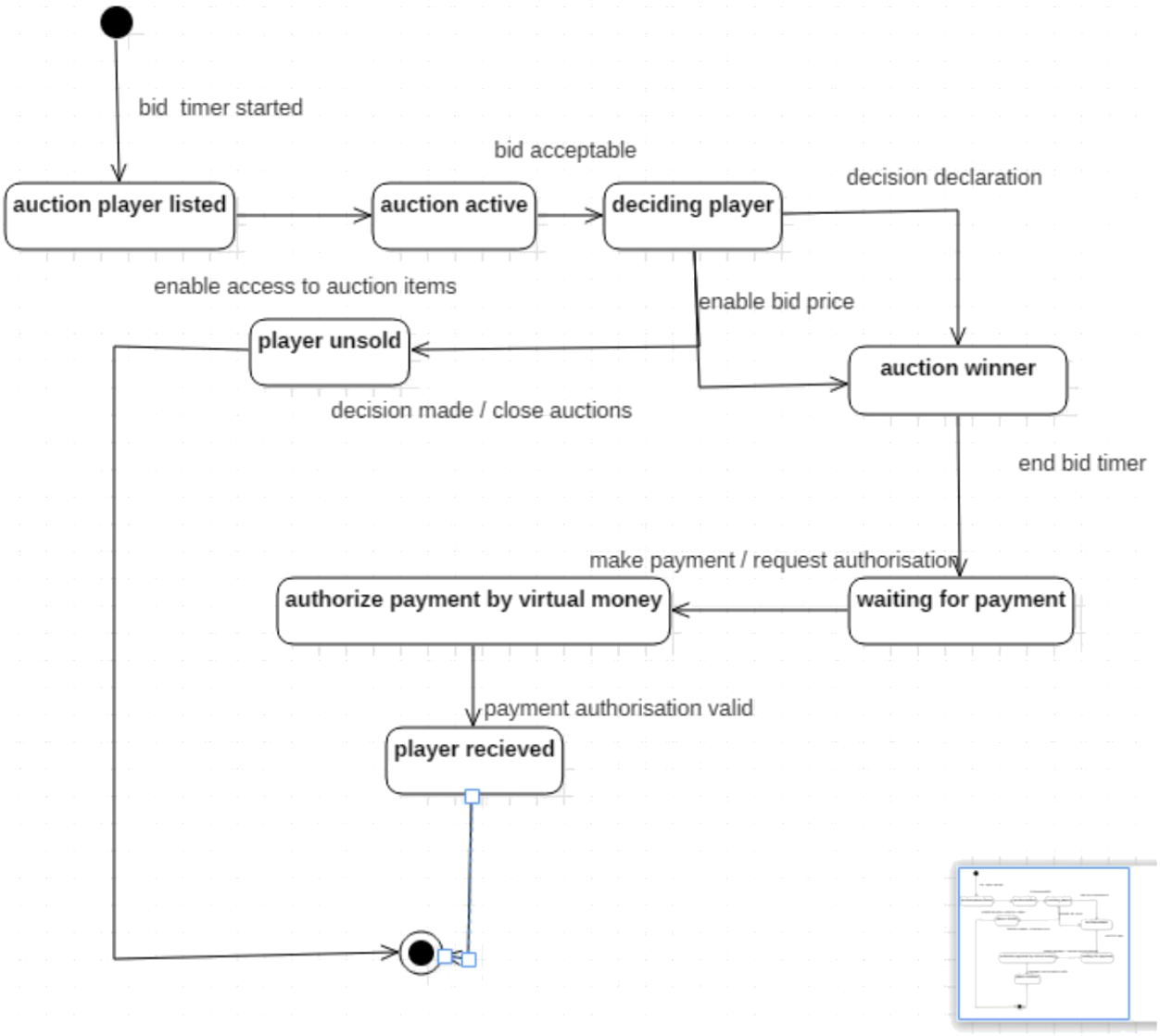
Statechart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination.

Statechart diagrams are also used for forward and reverse engineering of a system.
However, the main purpose is to model the reactive system.

Following are the main purposes of using Statechart diagrams −

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

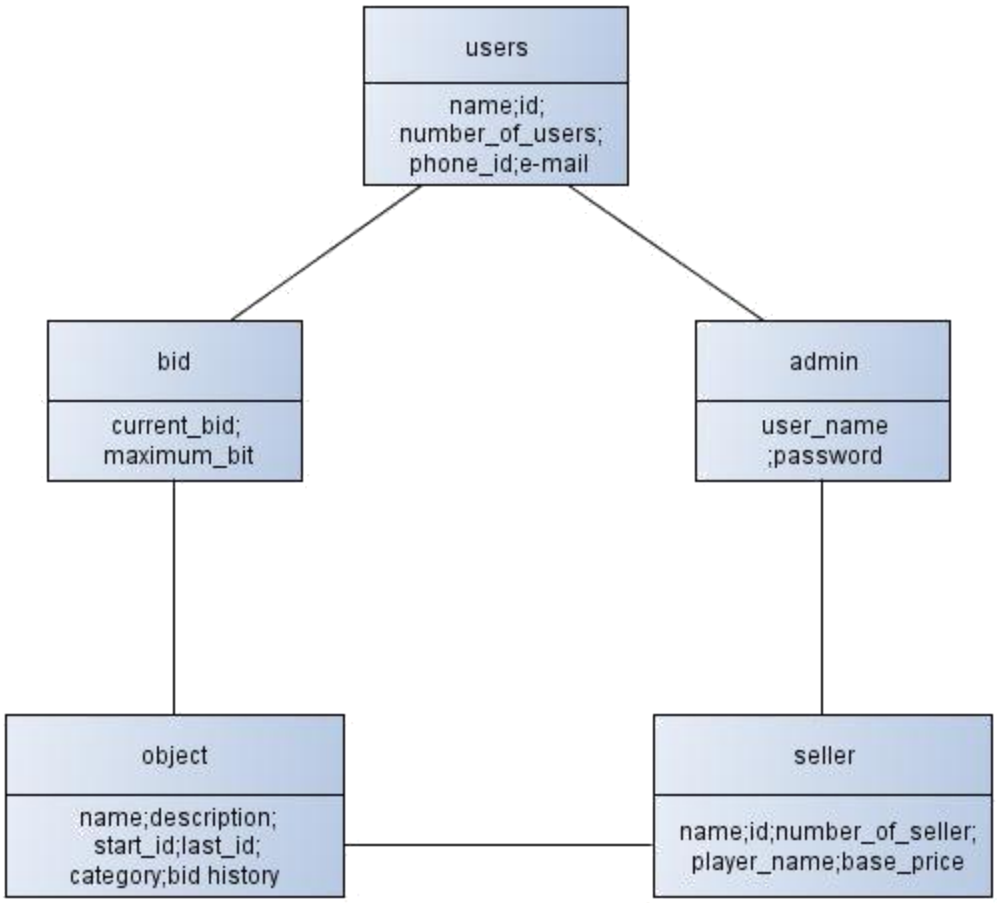

bid timer started
auction player listed
bid acceptable
auction active
deciding player
decision declaration
enable access to auction items
enable bid price
player unsold
auction winner
decision made / close auctions
end bid timer
make payment / request authorisation
authorize payment by virtual money
waiting for payment
payment authorisation valid
player recieved

11.OBJECT DIAGRAM: Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.

Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

Object diagrams are used to render a set of objects and their relationships as an instance.

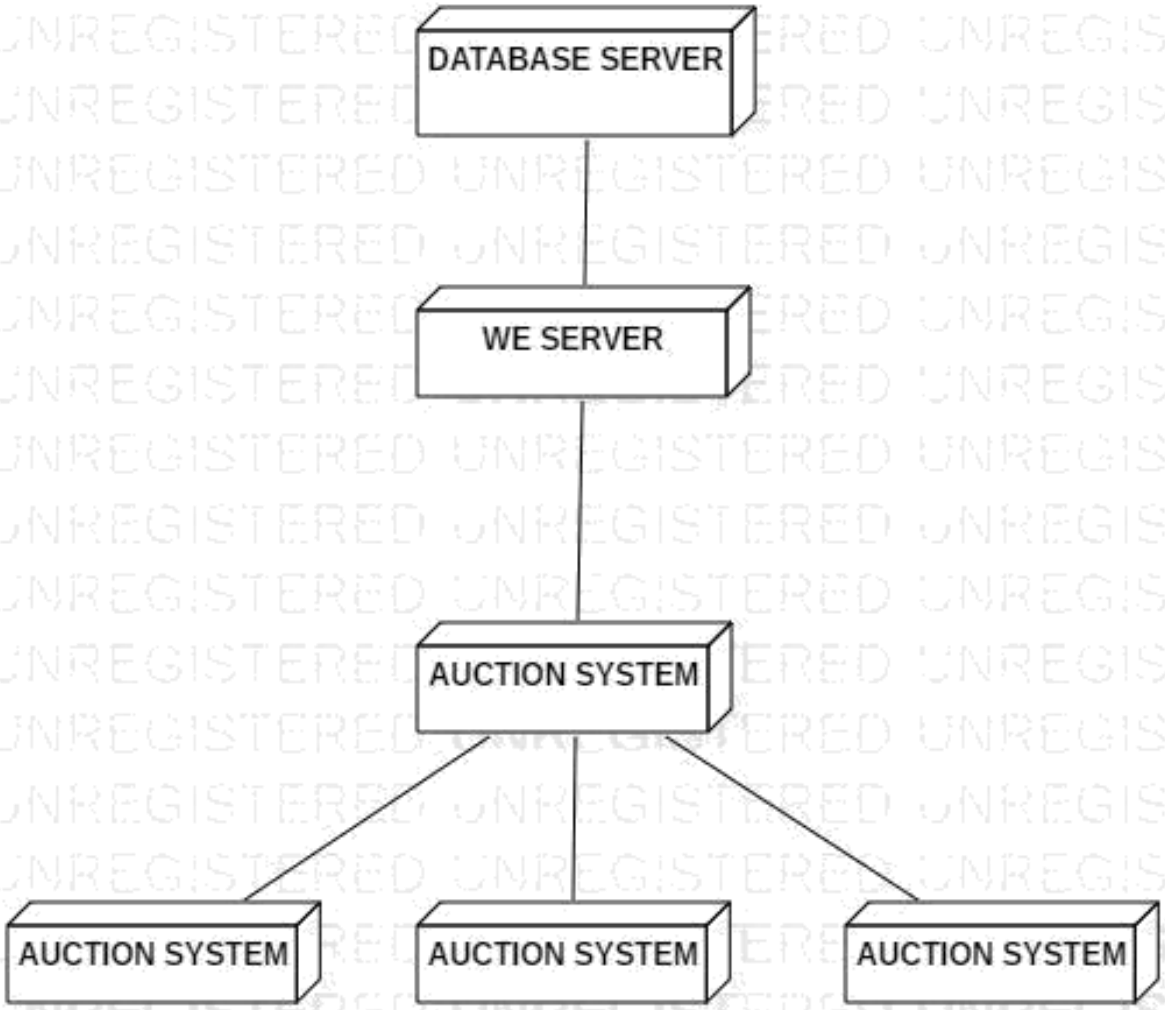
.



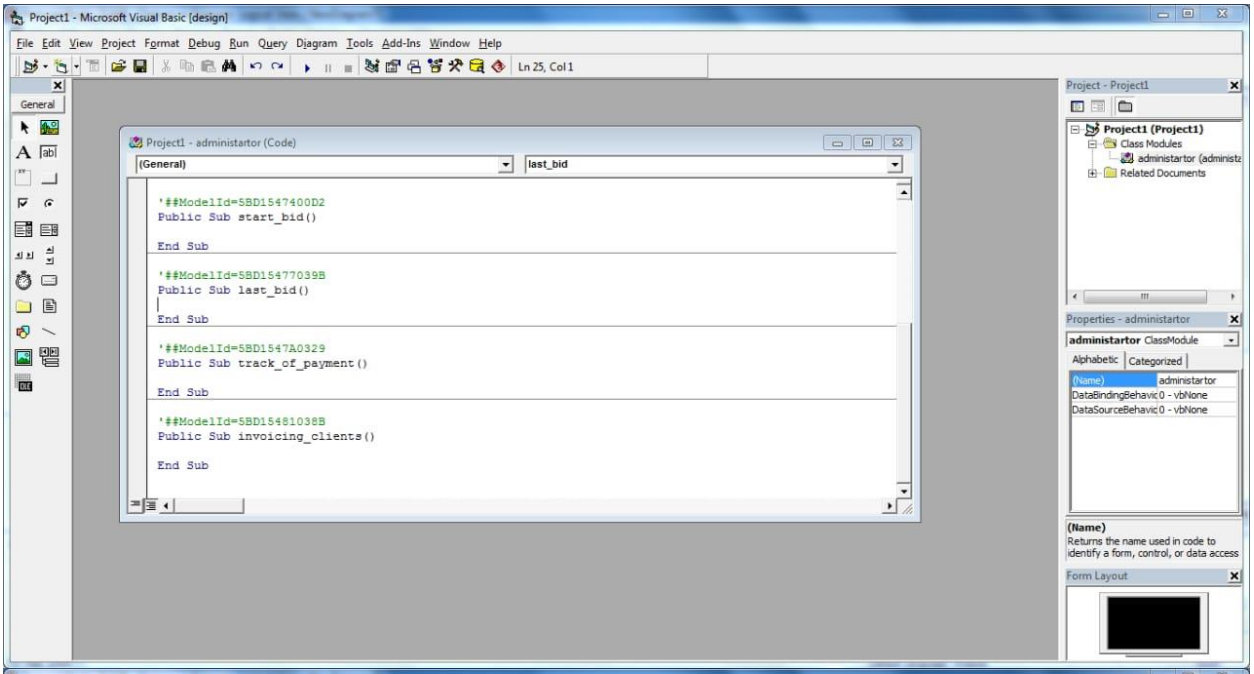
12.DEPLOYMENT DIAGRAM:

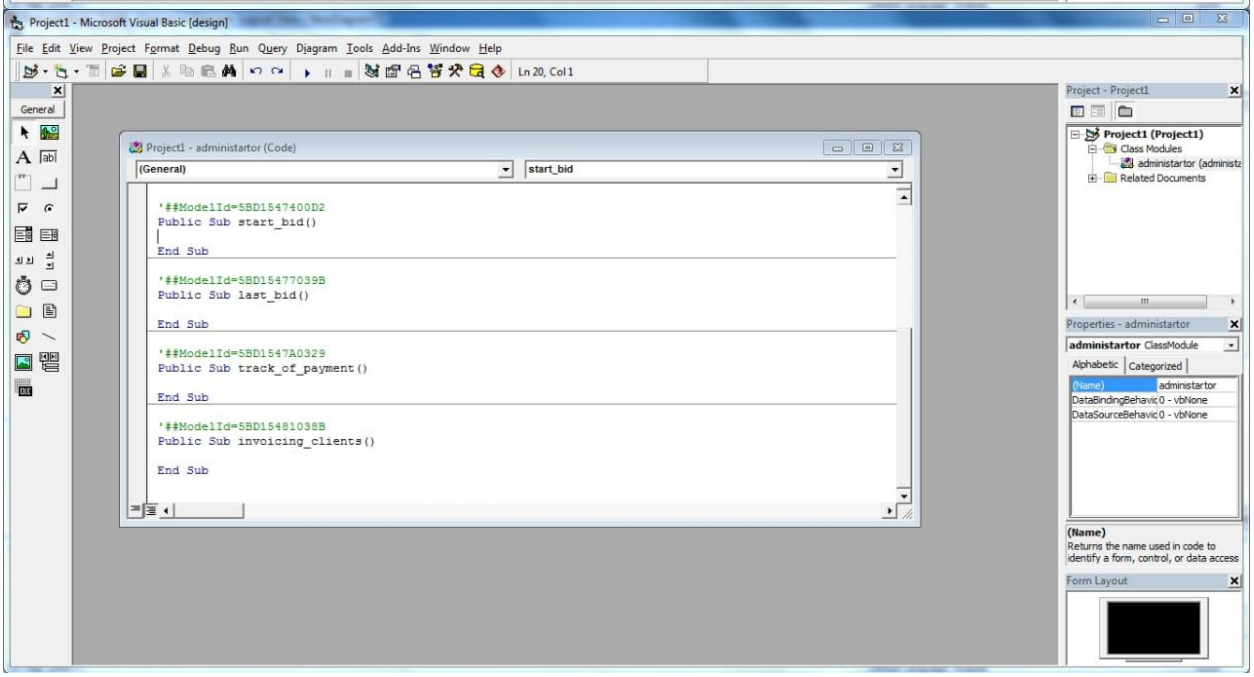
Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

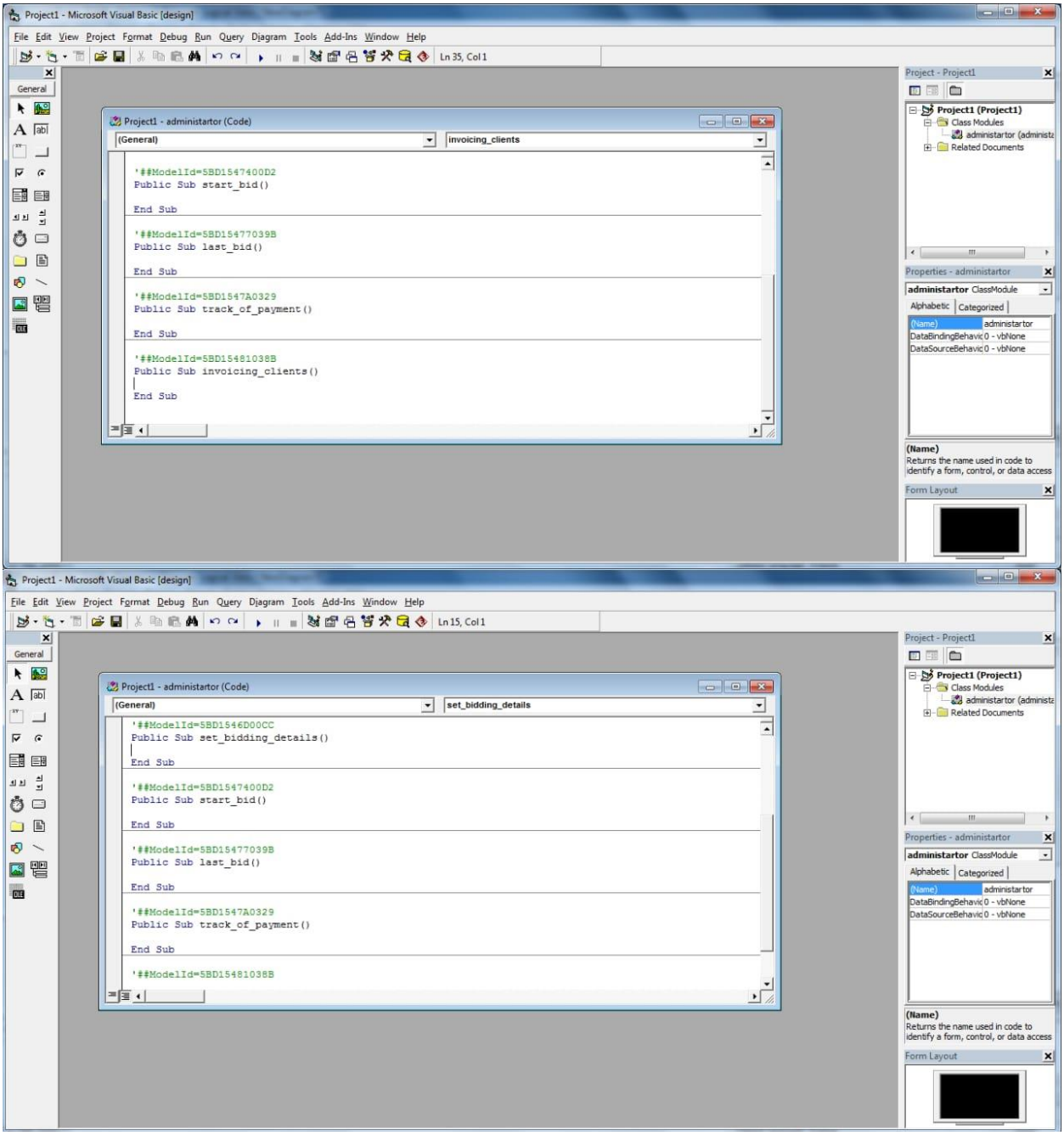
Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.



RATIONAL ROSE CODE:






Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Ln 35, Col 1
General
Project1 - administartor (Code)
(General)
invoicing_clients
'##ModelId=5BD1547400D2
Public Sub start_bid()
End Sub
'##ModelId=5BD15477039B
Public Sub last_bid()
End Sub
'##ModelId=5BD1547A0329
Public Sub track_of_payment()
End Sub
'##ModelId=5BD15481038B
Public Sub invoicing_clients()
End Sub
Project - Project1
Project1 (Project1)
Class Modules
administartor (administ
Related Documents
Properties - administartor
administartor ClassModule
Alphabetic
Categorized
(Name)
administartor
DataBindingBehavic 0 - vbNone
DataSourceBehavic 0 - vbNone
(Name)
Returns the name used in code to identify a form, control, or data access
Form Layout
Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Ln 15, Col 1
General
Project1 - administartor (Code)
(General)
set_bidding_details
'##ModelId=5BD1546D00CC
Public Sub set_bidding_details()
End Sub
'##ModelId=5BD1547400D2
Public Sub start_bid()
End Sub
'##ModelId=5BD15477039B
Public Sub last_bid()
End Sub
'##ModelId=5BD1547A0329
Public Sub track_of_payment()
End Sub
'##ModelId=5BD15481038B
Project - Project1
Project1 (Project1)
Class Modules
administartor (administ
Related Documents
Properties - administartor
administartor ClassModule
Alphabetic
Categorized
(Name)
administartor
DataBindingBehavic 0 - vbNone
DataSourceBehavic 0 - vbNone
(Name)
Returns the name used in code to identify a form, control, or data access
Form Layout

CHAPTER-3

IMPLEMENTATION:

TABLE OF CONTENTS:

CODE:

1.Auction.java

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
import java.awt.image.BufferedImage;
import java.net.*;
import java.io.InputStream;
import javax.imageio.ImageIO;
import java.util.Random;
class Auction extends JFrame implements ActionListener
{
   boolean status=false;
   int Age,Overall,z;
   Float Value;
   String Name,Photo,Nationality,Flag,Club,CLogo,Preff;
   Random randomNum = new Random();
   int x= randomNum.nextInt(205);
   JButton Next=new JButton("NEXT");
   JButton Buy=new JButton("BUY");
   int M(int x)throws Exception
   {System.out.println("Inside M: "+x);
      Boolean flag;flag=Exclude(x);
      if(flag==true)
      {
         x=randomNum.nextInt(205);
         x=M(x);
      }
      else if(flag==false)
      {    swap(x);
      }
      System.out.println("Returning : "+x);
      return x;
   }
   void swap(int x)throws Exception
   {
```

```
        System.out.println("Swapping "+x+" with 999");
        Class.forName("com.mysql.jdbc.Driver");
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/auction","
root","");
        Statement stmt;
        String query="update count set C ="+999+" where id="+(x);
        stmt=con.createStatement();
        stmt.execute(query);
           stmt.close();
           con.close();
    }
    Boolean Exclude(int x)throws Exception
    {
        int Array[]=new int[205];
        Class.forName("com.mysql.jdbc.Driver");
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/auction","
root","");
        PreparedStatement stmt=con.prepareStatement("Select C from count");
        ResultSet rs;
        rs=stmt.executeQuery();
        status=rs.next();
           for(int i=0;rs.next();i++)
           {
              Array[i]=(rs.getInt("C"));
           }
           for(int i=0;i<205;i++)
           {
              if(x==Array[i])
              {
                 return false;
              }
           }
        return true;
    }
    void Extract(int y)throws Exception
    {
           Class.forName("com.mysql.jdbc.Driver");
```

```
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/auction","
root","");
        PreparedStatement stmt=con.prepareStatement("select * from player
where ID=\""+y+"\"");
        ResultSet rs=stmt.executeQuery();
        status=rs.next();
        if(status==true)
                                                        {

       Name=rs.getString("Name");
                    Age=rs.getInt("AGE");
                    Photo=rs.getString("Photo");
                    Nationality=rs.getString("Nationality");
                    Flag=rs.getString("Flag");
                    Overall=rs.getInt("Overall");
                    Club=rs.getString("Club");
                    CLogo=rs.getString("Club Logo");
                    Value=rs.getFloat("Value");
                    Preff=rs.getString("PREFERED POSITION");
                                                        }
        con.close();
   }
   void Extract()throws Exception
   {System.out.println("Displaying player :"+z);
        Class.forName("com.mysql.jdbc.Driver");
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/auction","
root","");
        PreparedStatement stmt=con.prepareStatement("select * from player
where ID=\""+z+"\"");
        ResultSet rs=stmt.executeQuery();
        status=rs.next();
        if(status==true)
                                                        {

       Name=rs.getString("Name");
                    Age=rs.getInt("AGE");
                    Photo=rs.getString("Photo");
```

```
                    Nationality=rs.getString("Nationality");
                    Flag=rs.getString("Flag");
                    Overall=rs.getInt("Overall");
                    Club=rs.getString("Club");
                    CLogo=rs.getString("Club Logo");
                    Value=rs.getFloat("Value");
                    Preff=rs.getString("PREFERED POSITION");
                                                        }
        con.close();
    }
    Auction(int X) throws Exception
    {
       Extract(X);
       setTitle("FIFA 18| Auction ");
         setSize(660, 760);
       setBackground(Color.white);
       setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
      // setUndecorated(true);
       setResizable(false);
       JPanel p=new JPanel();
       ImageIcon Q=new
ImageIcon("C:\\Users\\harri\\Desktop\\Stuff\\Auction\\Back.png");
       JLabel J=new JLabel(Q);
       J.setBounds(0,0,655,750);
       p.setLayout(null);
       URL url_Photo = new URL(Photo);
       URL url_Flag = new URL(Flag);
       URL url_CLogo = new URL(CLogo);
       Image image_Photo = ImageIO.read(url_Photo.openStream());
       Image image_Flag = ImageIO.read(url_Flag.openStream());
       Image image_CLogo = ImageIO.read(url_CLogo.openStream());
       JLabel label_Photo = new JLabel(new ImageIcon(image_Photo));
       JLabel label_Flag = new JLabel(new ImageIcon(image_Flag));
       JLabel label_CLogo = new JLabel(new ImageIcon(image_CLogo));
       JLabel label_Name =new JLabel(Name);
       JLabel label_Club =new JLabel(Club);
       JLabel label_Value =new JLabel(Float.toString(Value));
       JLabel label_Preff=new JLabel(Preff);
       JLabel label_Overall=new JLabel(Integer.toString(Overall));
```

```
JLabel label_Age=new JLabel(Integer.toString(Age));
JLabel label_Nationality=new JLabel(Nationality);
JLabel M=new JLabel("M");
JLabel a=new JLabel("AGE :");
JLabel n=new JLabel("NATIONALITY :");
JLabel P=new JLabel("POSITION :");
JLabel c=new JLabel("CLUB :");
JLabel b=new JLabel("BASE PRICE :");
JLabel o=new JLabel("RATING :");
Font font1 = new Font("Copperplate Gothic Bold",Font.BOLD,45);
Font font = new Font("Copperplate Gothic Bold",Font.BOLD,30);
Font font2 = new Font("Copperplate Gothic Bold",Font.BOLD,25);
label_Name.setFont(font1);
label_Age.setFont(font2);
label_Preff.setFont(font2);
label_Club.setFont(font2);
label_Nationality.setFont(font2);
M.setFont(font2);
a.setFont(font);
n.setFont(font);
P.setFont(font);
c.setFont(font);
b.setFont(font);
o.setFont(font);
Next.setFont(font2);
Buy.setFont(font2);
label_Value.setFont(font2);
label_Overall.setFont(font2);
label_Photo.setBounds(15,342,48,48);
label_Flag.setBounds(360,448,23,17);
label_CLogo.setBounds(210,520,24,24);
label_Name.setBounds(75,240,615,270);
label_Club.setBounds(237,472,300,125);
label_Value.setBounds(335,550,148,125);
M.setBounds(400,550,148,125);
label_Overall.setBounds(255,510,148,125);
label_Preff.setBounds(282,410,250,175);
label_Age.setBounds(180,340,48,148);
label_Nationality.setBounds(385,381,250,148);
```

```
    a.setBounds(75,327,148,175);
    n.setBounds(75,368,290,175);
    P.setBounds(75,410,200,175);
    c.setBounds(75,448,148,175);
    b.setBounds(75,526,348,175);
    o.setBounds(75,485,348,175);
    Next.setBounds(330,650,125,50);
    Buy.setBounds(185,650,125,50);
    p.add(M);
    p.add(label_Photo);
    p.add(label_CLogo);
    p.add(label_Name);
    p.add(label_Club);
    p.add(label_Value);
    p.add(label_Preff);
    p.add(label_Age);
    p.add(label_Nationality);
    p.add(label_Flag);
    p.add(label_Overall);
    p.add(a);
    p.add(n);
    p.add(P);
    p.add(c);
    p.add(o);
    p.add(b);
    p.add(Next);
    p.add(Buy);
      p.add(J);
    add(p);
    setVisible(true);
    Next.addActionListener(this);
    Buy.addActionListener(this);
}
Auction() throws Exception
{
    try{z=M(x);}catch(Exception e){System.out.println("ERROR");}
    System.out.println("Z :" +z);
    Extract();
    setTitle("FIFA 18| Auction ");
```

```
          setSize(660, 760);
        setBackground(Color.white);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
       // setUndecorated(true);
        setResizable(false);
        JPanel p=new JPanel();
        ImageIcon Q=new
ImageIcon("C:\\Users\\harri\\Desktop\\Stuff\\Auction\\Back.png");
        JLabel J=new JLabel(Q);
        J.setBounds(0,0,655,750);
        p.setLayout(null);
        URL url_Photo = new URL(Photo);
        URL url_Flag = new URL(Flag);
        URL url_CLogo = new URL(CLogo);
        Image image_Photo = ImageIO.read(url_Photo.openStream());
        Image image_Flag = ImageIO.read(url_Flag.openStream());
        Image image_CLogo = ImageIO.read(url_CLogo.openStream());
        JLabel label_Photo = new JLabel(new ImageIcon(image_Photo));
        JLabel label_Flag = new JLabel(new ImageIcon(image_Flag));
        JLabel label_CLogo = new JLabel(new ImageIcon(image_CLogo));
        JLabel label_Name =new JLabel(Name);
        JLabel label_Club =new JLabel(Club);
        JLabel label_Value =new JLabel(Float.toString(Value));
        JLabel label_Preff=new JLabel(Preff);
        JLabel label_Overall=new JLabel(Integer.toString(Overall));
        JLabel label_Age=new JLabel(Integer.toString(Age));
        JLabel label_Nationality=new JLabel(Nationality);
        JLabel M=new JLabel("M");
        JLabel a=new JLabel("AGE :");
        JLabel n=new JLabel("NATIONALITY :");
        JLabel P=new JLabel("POSITION :");
        JLabel c=new JLabel("CLUB :");
        JLabel b=new JLabel("BASE PRICE :");
        JLabel o=new JLabel("RATING :");
        Font font1 = new Font("Copperplate Gothic Bold",Font.BOLD,45);
        Font font = new Font("Copperplate Gothic Bold",Font.BOLD,30);
        Font font2 = new Font("Copperplate Gothic Bold",Font.BOLD,25);
        label_Name.setFont(font1);
        label_Age.setFont(font2);
```

```
label_Preff.setFont(font2);
label_Club.setFont(font2);
label_Nationality.setFont(font2);
a.setFont(font);
n.setFont(font);
P.setFont(font);
c.setFont(font);
b.setFont(font);
o.setFont(font);
M.setFont(font2);
Next.setFont(font2);
Buy.setFont(font2);
label_Value.setFont(font2);
label_Overall.setFont(font2);
label_Photo.setBounds(15,342,48,48);
label_Flag.setBounds(360,448,23,17);
label_CLogo.setBounds(210,520,24,24);
label_Name.setBounds(75,240,615,270);
label_Club.setBounds(237,472,300,125);
label_Value.setBounds(335,550,148,125);
M.setBounds(400,550,148,125);
label_Overall.setBounds(255,510,148,125);
label_Preff.setBounds(282,410,250,175);
label_Age.setBounds(180,340,48,148);
label_Nationality.setBounds(385,381,250,148);
a.setBounds(75,327,148,175);
n.setBounds(75,368,290,175);
P.setBounds(75,410,200,175);
c.setBounds(75,448,148,175);
b.setBounds(75,526,348,175);
o.setBounds(75,485,348,175);
Next.setBounds(330,650,125,50);
Buy.setBounds(185,650,125,50);
p.add(M);
p.add(label_Photo);
p.add(label_CLogo);
p.add(label_Name);
p.add(label_Club);
p.add(label_Value);
```

```
        p.add(label_Preff);
        p.add(label_Age);
        p.add(label_Nationality);
        p.add(label_Flag);
        p.add(label_Overall);
        p.add(a);
        p.add(n);
        p.add(P);
        p.add(c);
        p.add(o);
        p.add(b);
        p.add(Next);
        p.add(Buy);
          p.add(J);
        add(p);
        setVisible(true);
        Next.addActionListener(this);
        Buy.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae)
         {
            if(ae.getSource()==Next)
            {
               try{new Auction();}
               catch(Exception e){}
               dispose();
            }
            if (ae.getSource()==Buy)
            {
               try{new Bidder(Name,x,Value);}
               catch(Exception e){}
               dispose();
            }
        }
    public static void main(String args[])throws Exception
         {
                                                        //Auction a=new
Auction();
        }
```

```
}
```

2.Bidder.java

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
class Bidder extends JFrame implements ActionListener
{
   Float amt;
   String
N[]={"prajith","abdu","harris","saji","prabat","chindan","samuel","sourav",
"akhildev"};
   boolean status=false;
   JTextField BidderName;
   JTextField Price;
   JButton Btn=new JButton("Buy");
   JButton Back=new JButton("Back");
   String P_Name;
   int a=1;
   int X;
   Float V;
   Bidder(String Name,int x,Float Value)
   {
      P_Name=Name;
      X=x;
      V=Value;
      setTitle("FIFA 18| Auction ");
        setSize(660, 760);
      setBackground(Color.white);
      setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
      JPanel p=new JPanel();
      ImageIcon Q=new
ImageIcon("C:\\Users\\harri\\Desktop\\Stuff\\Auction\\Bid.png");
      JLabel J=new JLabel(Q);
      JLabel B=new JLabel("Bidder Name");
      JLabel P=new JLabel("Bid Amount");
```

```
        J.setBounds(0,0,655,750);
        BidderName=new JTextField(20);
        Price=new JTextField(20);
          BidderName.setBorder(javax.swing.BorderFactory.createEmptyBord
er());
        Price.setBorder(javax.swing.BorderFactory.createEmptyBorder());
        Font font = new Font("Copperplate Gothic Bold",Font.BOLD,34);
        Font font1 = new Font("Copperplate Gothic Bold",Font.BOLD,25);
          BidderName.setFont(font);
        Price.setFont(font);
        B.setFont(font);
        P.setFont(font);
        Btn.setFont(font1);
        Back.setFont(font1);
        BidderName.setBounds(328,350,300,50);
        Price.setBounds(328,435,300,50);
        B.setBounds(30,240,280,270);
        P.setBounds(30,325,280,270);
        Btn.setBounds(185,650,125,50);
        Back.setBounds(330,650,125,50);
        p.add(Btn);
        p.add(Back);
        p.setLayout(null);
        p.add(P);
        p.add(Price);
        p.add(B);
        p.add(BidderName);
        p.add(J);
        add(p);
        setVisible(true);
        getRootPane().setDefaultButton(Btn);
        Btn.addActionListener(this);
        Back.addActionListener(this);
    }
    void funcG(int i)throws Exception
    {
        Class.forName("com.mysql.jdbc.Driver");
```

```
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/auction","
root","");
        PreparedStatement stmt;
        String query="select * from bank where id="+(i+1);
        ResultSet rs;
              stmt=con.prepareStatement(query);
                 rs=stmt.executeQuery();
                 status=rs.next();
                 if(status==true)
                 {
                       amt=Float.parseFloat(rs.getString("AMT"));
                 }
                  con.close();
    }
    void funcI(int i,Float A)throws Exception
    {
        Class.forName("com.mysql.jdbc.Driver");
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/auction","
root","");
        Statement stmt;
        ResultSet rs;
        String query="update bank set AMT ="+A+" where id="+(i+1);
        stmt=con.createStatement();
        stmt.execute(query);
        String st=P_Name+" brought by "+BidderName.getText()+" for
"+Price.getText()+" M";

JOptionPane.showMessageDialog(this,st,"Succes",JOptionPane.INFORMA
TION_MESSAGE);
        try{new
Buy(BidderName.getText(),P_Name,Float.parseFloat(Price.getText()));}
                 catch(Exception e){}
                 dispose();
           stmt.close();
           con.close();

    }
```

```
   void check() throws Exception
   {
      Float price=Float.parseFloat(Price.getText());
      int flag=0;
      String Name=BidderName.getText();
      for(int i=0;i<9;i++)
      {
         if(Name.equalsIgnoreCase(N[i]))
         {
               try{funcG(i);}catch(Exception e){}
               if(price<=amt&& price>=V)
               {
                  amt-=price;a=0;
                  try{funcI(i,amt);}catch(Exception e){}
               }
               else if(price<V)    JOptionPane.showMessageDialog(this,"the
entered amount is less than base price","There was a
problem",JOptionPane.ERROR_MESSAGE);
               else
               {
                  JOptionPane.showMessageDialog(this,"the entered amount is
greater than available amount","There was a
problem",JOptionPane.ERROR_MESSAGE);a=1;
               }flag=1;
               break;
         }
      }
      if(flag==0)
         JOptionPane.showMessageDialog(this,"Bidder Name Not
Identified","There was a problem",JOptionPane.ERROR_MESSAGE);a=1;
   }

   public void actionPerformed(ActionEvent ae)
        {
          if(ae.getSource()==Btn)
          {
             if(a==1){try{ check();}catch(Exception e){}}
             else{
```

```
                try{new
Buy(BidderName.getText(),P_Name,Float.parseFloat(Price.getText()));}
                catch(Exception e){}
                dispose();}
            }
            if(ae.getSource()==Back)
            {
                try{new Auction(X);}catch(Exception e){}
                dispose();
            }
    }
    public static void main(String args[])throws Exception
         {
        }
}
```

3.Buy.java

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
class Buy extends JFrame //implements ActionListener
{
   JTextField txt;
   Boolean status=false;
   String Player[]=new String[11];
   int n;
   JLabel B;
   PreparedStatement stmt=null;
   void Extract(String Bidder,String Name,Float Price)throws Exception
   {
         Class.forName("com.mysql.jdbc.Driver");
         Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/auction","
root","");
         String Query="insert into "+Bidder+" values(?,?)";
         stmt=con.prepareStatement(Query);
         stmt.setString(1,Name);
         stmt.setFloat(2,Price);
         stmt.executeUpdate();
         stmt.close();
         con.close();
         dispose();
         new Auction();
   }
   Buy(String BidderName,String Name,Float Price)
   {
      try{Extract(BidderName,Name,Price);}catch(Exception e){}
   }
   public static void main(String args[])throws Exception
        {

      }
```

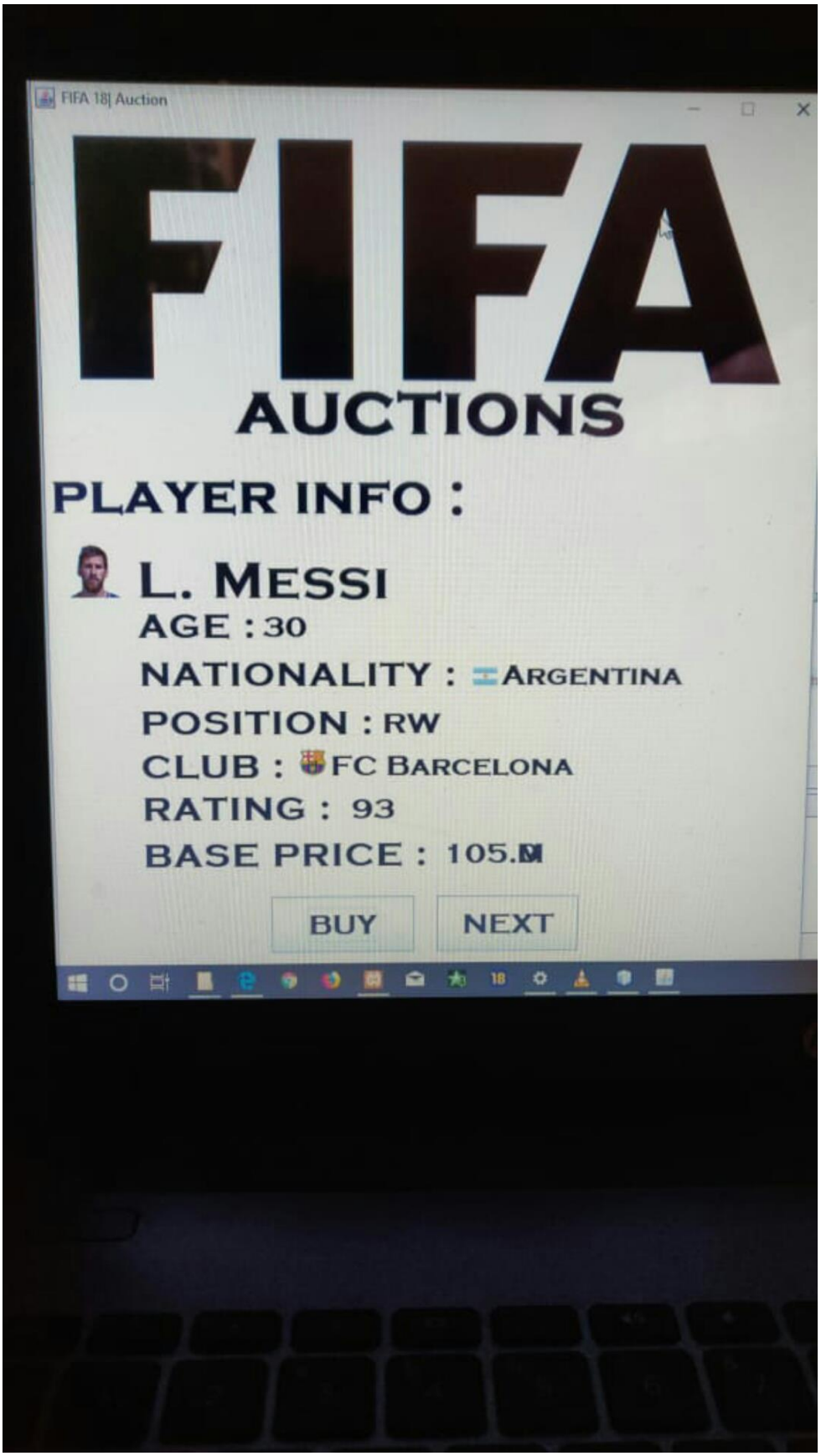
```
}
```

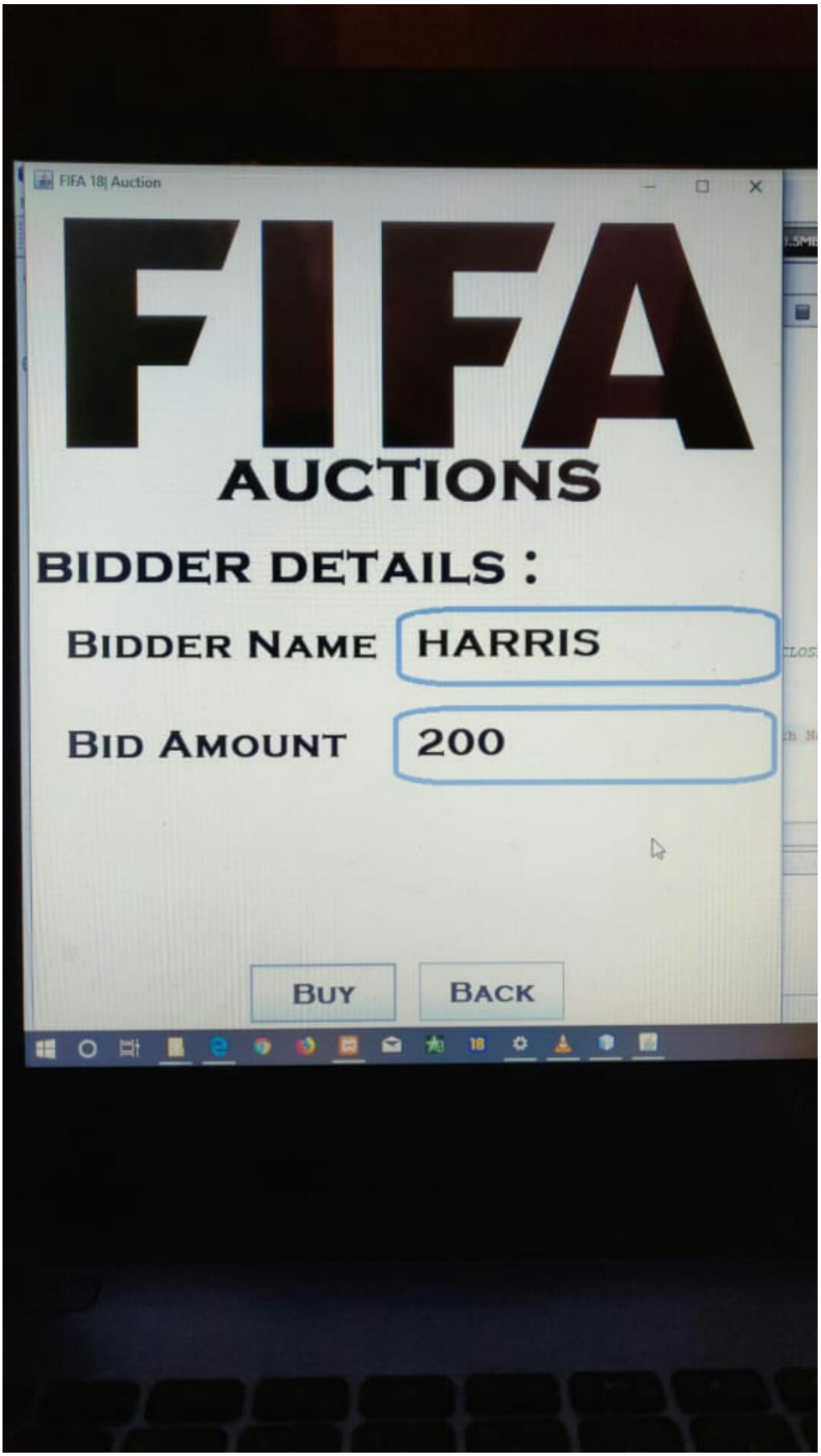
4.Logo.java

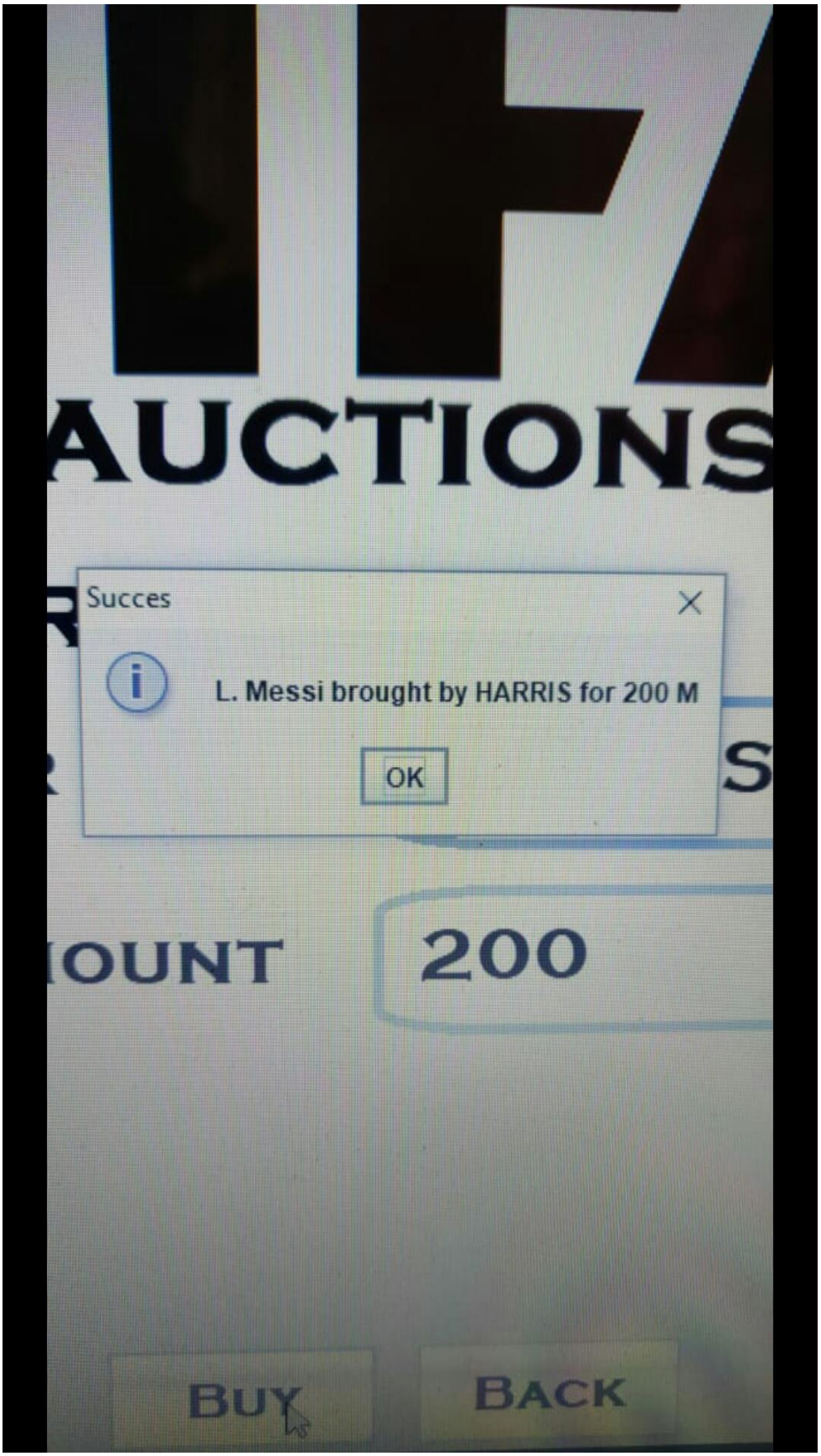
```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
import java.awt.image.BufferedImage;
import java.net.*;
import java.io.InputStream;
import javax.imageio.ImageIO;
import java.util.Random;
class Logo extends JFrame
{
   Logo()
   {
        setSize(781, 449);
      setBackground(Color.white);
      setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
      setUndecorated(true);
      setResizable(false);
      JPanel p=new JPanel();
      ImageIcon Q=new
ImageIcon("C:\\Users\\harri\\Desktop\\Stuff\\Auction\\Logo.jpg");
      JLabel J=new JLabel(Q);
      J.setBounds(0,0,781, 449);
      p.setLayout(null);
      p.add(J);
      add(p);
      setVisible(true);
   }
public static void main(String args[])throws Exception
       {
                                                    Logo l= new Logo();
            new Auction();
            l.dispose();
      }
```

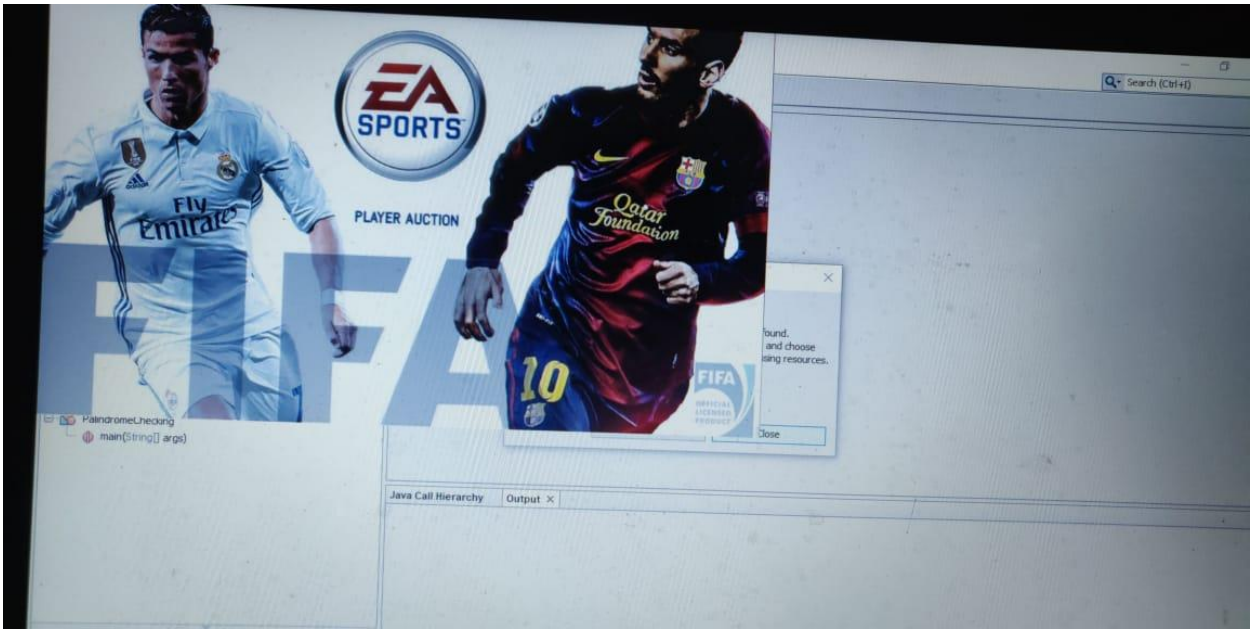
} CHAPTER_ 4

SCREENSHOTS:


FIFA 18| Auction
FIFA
AUCTIONS
PLAYER INFO :
L. MESSI
AGE : 30
NATIONALITY : ARGENTINA
POSITION : RW
CLUB : FC BARCELONA
RATING : 93
BASE PRICE : 105.M
BUY
NEXT


FIFA 18| Auction
FIFA
AUCTIONS
BIDDER DETAILS :
BIDDER NAME
HARRIS
BID AMOUNT
200
BUY
BACK


AUCTIONS
Succes
L. Messi brought by HARRIS for 200 M
OK
200
BUY
BACK



REFERENCES

1.Fred Barwell,Richard Blair,..(2004)'Professional VB.NET 2nd Edition',Wrox Press Ltd.

1. www.auction.indiatimes.com
2. www.ebay.com
3. www.msdn. com

4.Fifa EA sports