Final Documentation

CREWSADERS IT SOLUTIONS

INFORMATION SYSTEM DEVELOPMENT PROJECT

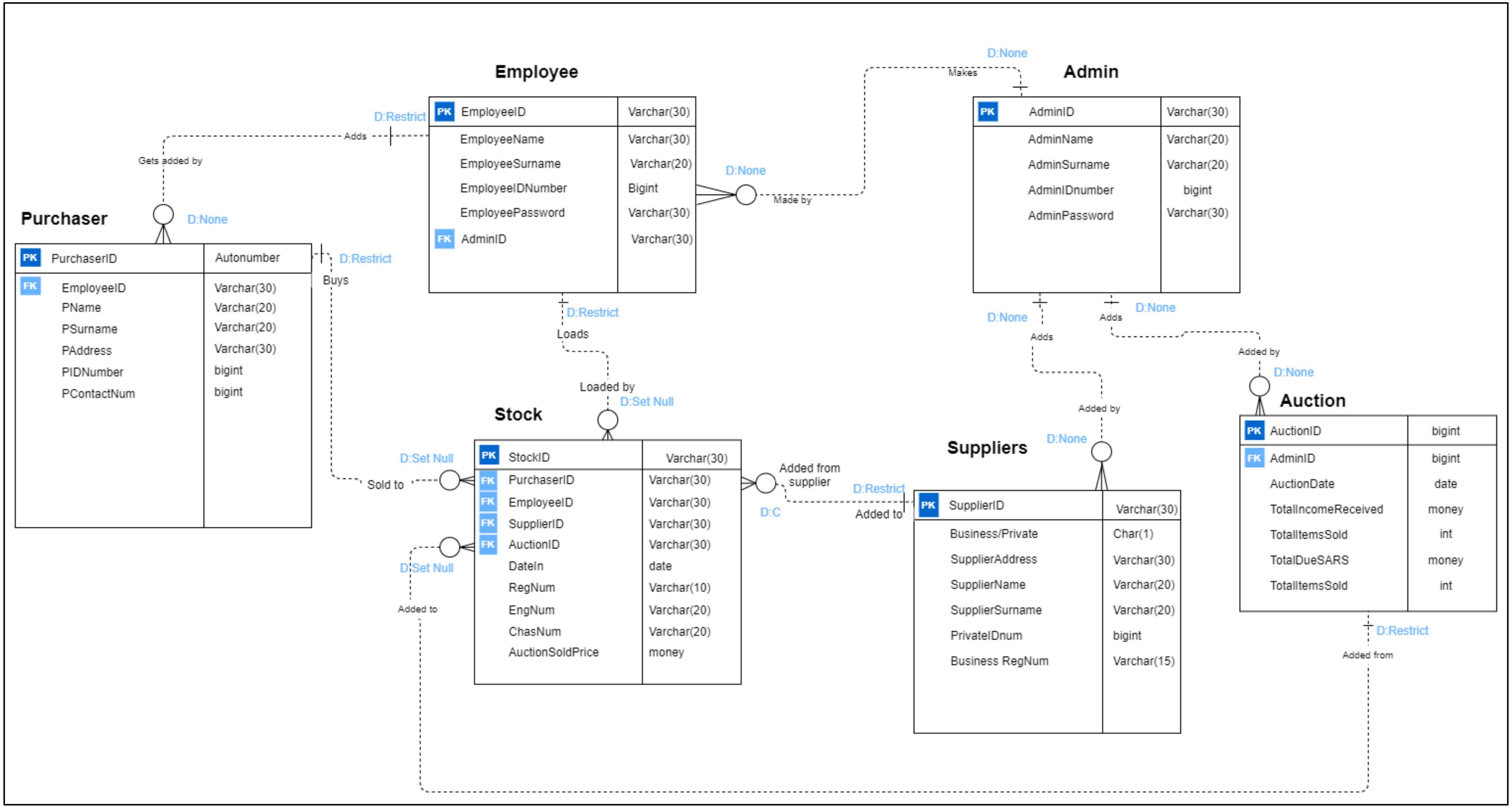
- FOR JMPPT VEHICLE AUTIONEERS

CREWSADERS GROUP MEMBERS:

- Barnard, J – 35796316
- Lategan, R – 34956905
- Rademeyer, M – 34551344
- Van der Merwe, P – 35244119

Table of Contents

Physical Data Model.



Employee

PK	EmployeeID	Varchar(30)
	EmployeeName	Varchar(30)
	EmployeeSurname	Varchar(20)
	EmployeeIDNumber	Bigint
	EmployeePassword	Varchar(30)
FK	AdminID	Varchar(30)

Admin

PK	AdminID	Varchar(30)
	AdminName	Varchar(20)
	AdminSurname	Varchar(20)
	AdminIDnumber	bigint
	AdminPassword	Varchar(30)

Purchaser

PK	PurchaserID	Autonumber
FK	EmployeeID	Varchar(30)
	PName	Varchar(20)
	PSurname	Varchar(20)
	PAddress	Varchar(30)
	PIDNumber	bigint
	PContactNum	bigint

Stock

PK	StockID	Varchar(30)
FK	PurchaserID	Varchar(30)
FK	EmployeeID	Varchar(30)
FK	SupplierID	Varchar(30)
FK	AuctionID	Varchar(30)
	DateIn	date
	RegNum	Varchar(10)
	EngNum	Varchar(20)
	ChasNum	Varchar(20)
	AuctionSoldPrice	money

Suppliers

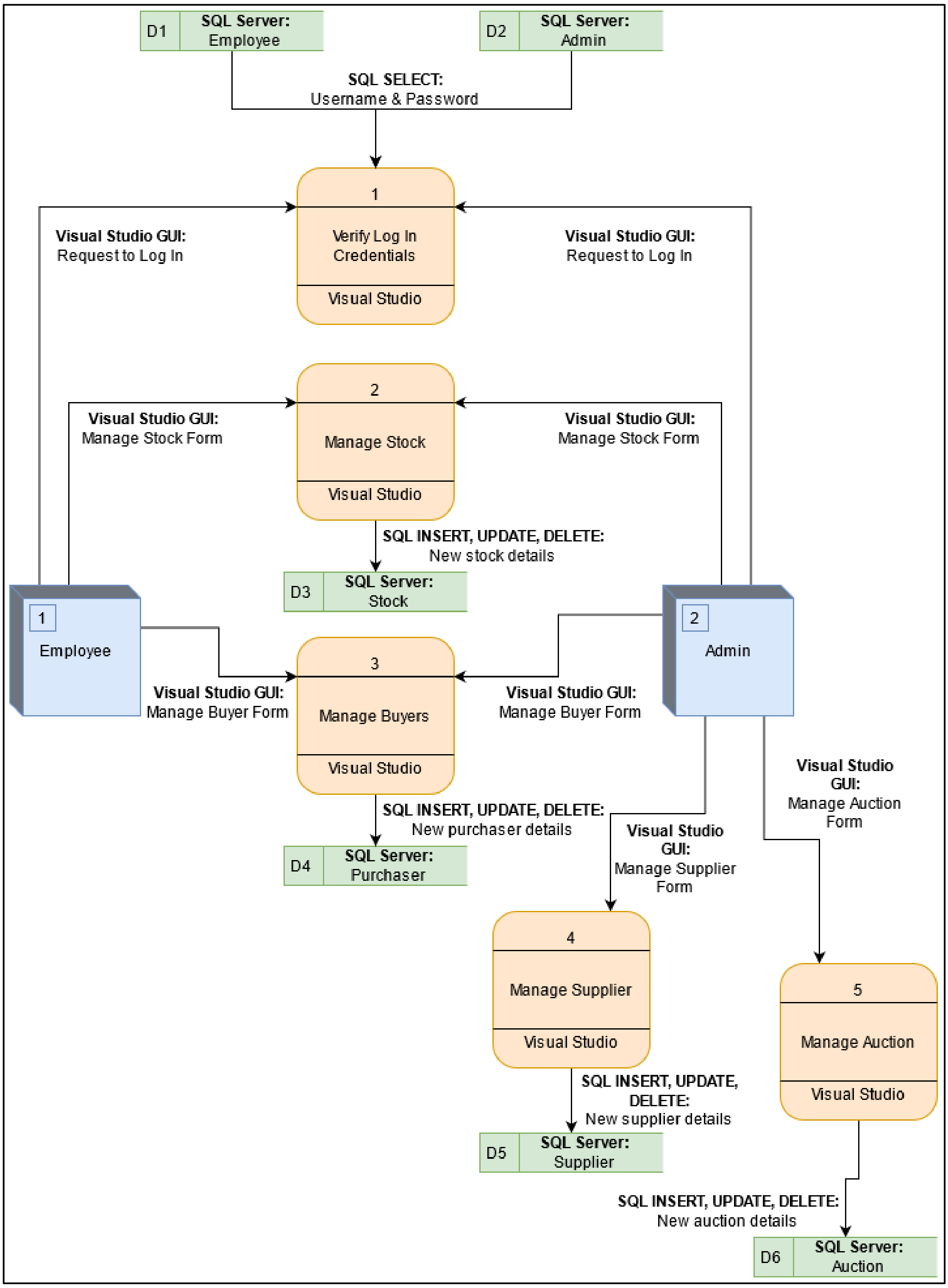
PK	SupplierID	Varchar(30)
	Business/Private	Char(1)
	SupplierAddress	Varchar(30)
	SupplierName	Varchar(20)
	SupplierSurname	Varchar(20)
	PrivateIDnum	bigint
	Business RegNum	Varchar(15)

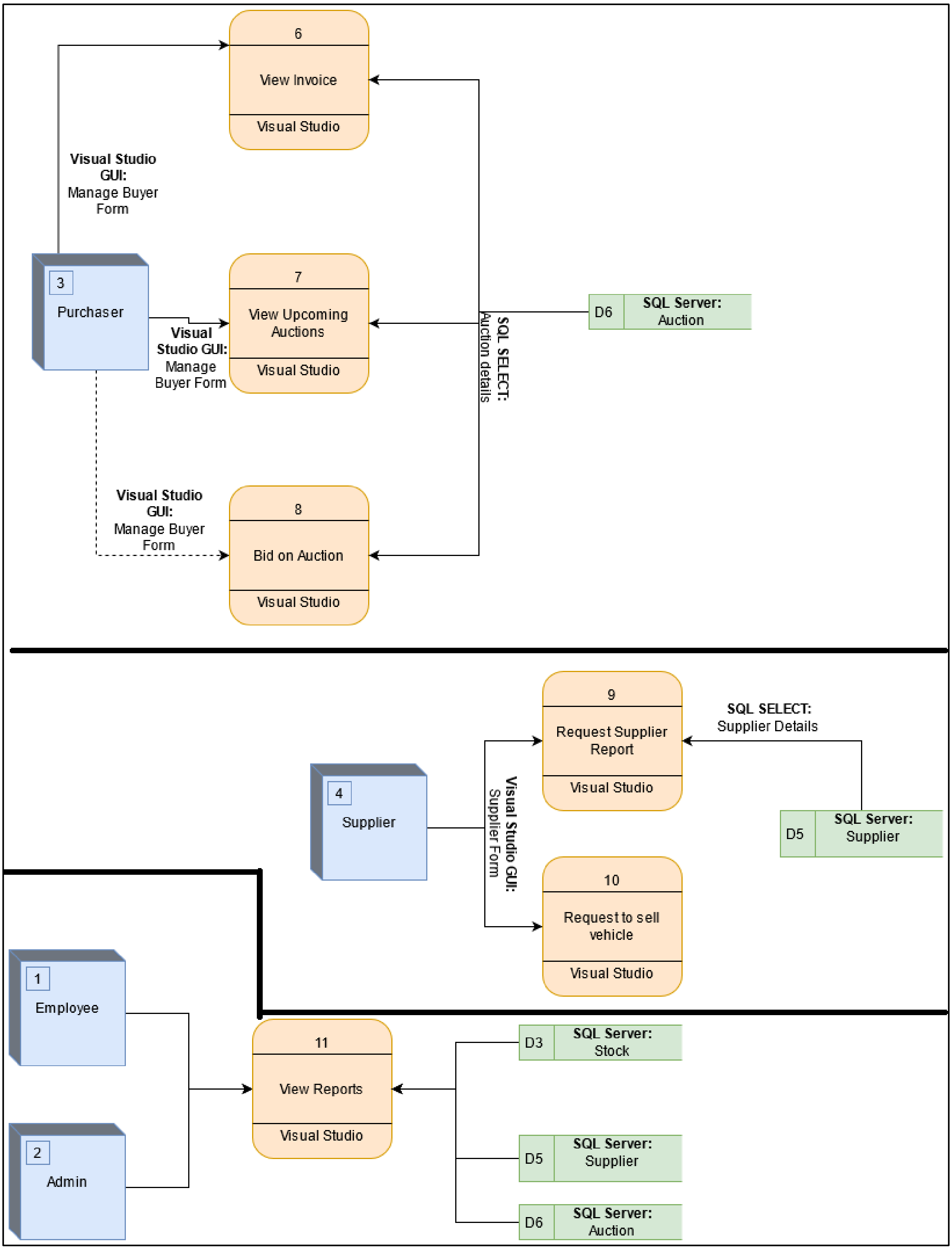
Auction

PK	AuctionID	bigint
FK	AdminID	bigint
	AuctionDate	date
	TotalIncomeReceived	money
	TotalItemsSold	int
	TotalDueSARS	money
	TotalItemsSold	int

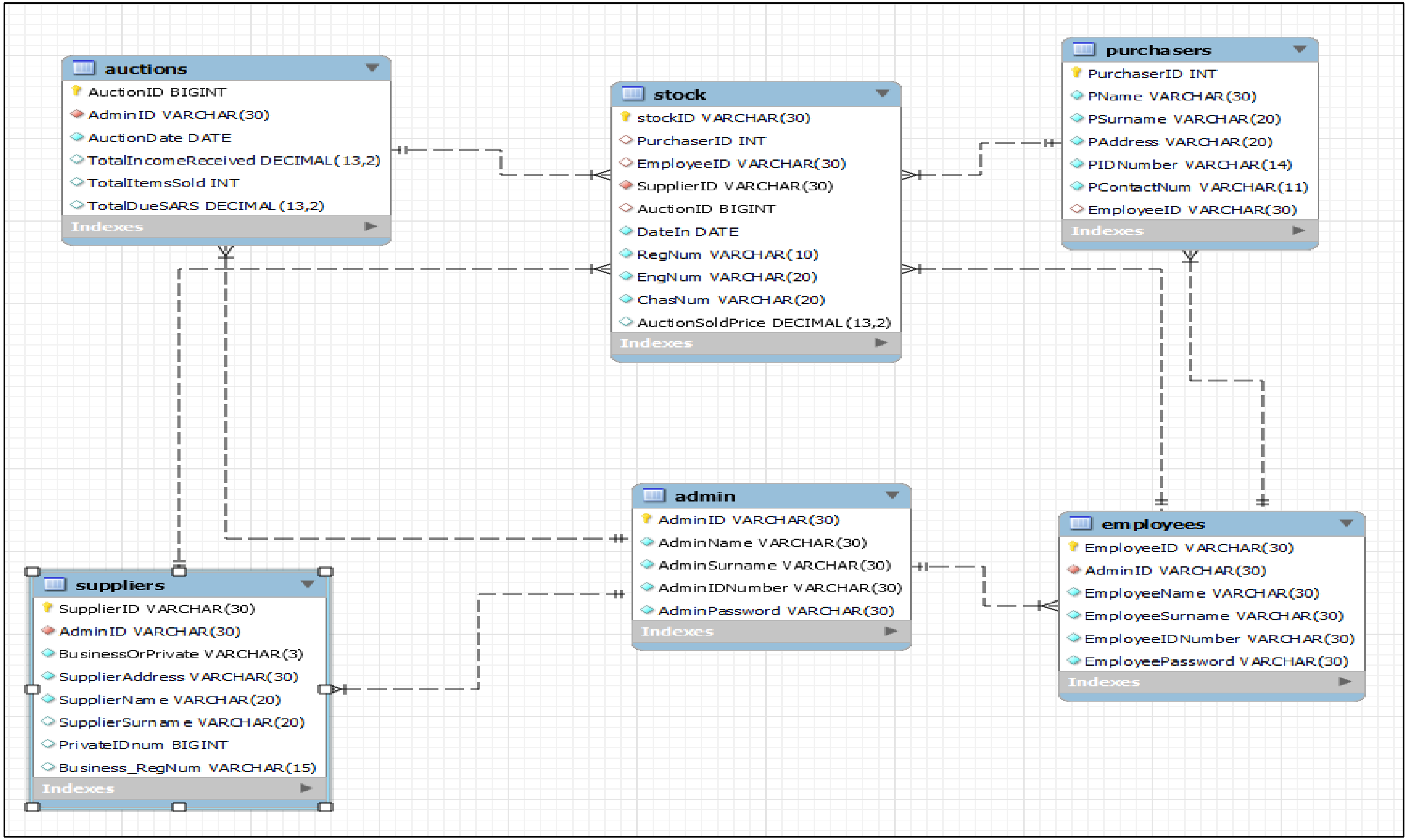
Physical Process Model.

NOTE: The diagram was split to better read and understand it.




6
View Invoice
Visual Studio
Visual Studio GUI: Manage Buyer Form
3
Purchaser
7
View Upcoming Auctions
Visual Studio
Visual Studio GUI: Manage Buyer Form
SQL SELECT: Auction details
D6
SQL Server: Auction
Visual Studio GUI: Manage Buyer Form
8
Bid on Auction
Visual Studio
9
Request Supplier Report
Visual Studio
SQL SELECT: Supplier Details
4
Supplier
Visual Studio GUI: Supplier Form
D5
SQL Server: Supplier
10
Request to sell vehicle
Visual Studio
1
Employee
11
View Reports
Visual Studio
D3
SQL Server: Stock
2
Admin
D5
SQL Server: Supplier
D6
SQL Server: Auction

Database Schema.



SQL Code.

Creating the database.

```
CREATE DATABASE auctiondb;
USE auctiondb;

CREATE TABLE Admin(
AdminID Varchar(30) NOT NULL,
AdminName Varchar(30) NOT NULL,
AdminSurname Varchar(30) NOT NULL,
AdminIDNumber Varchar(30) NOT NULL,
AdminPassword Varchar(30) NOT NULL,
PRIMARY KEY (AdminID)
);

CREATE TABLE employees(
EmployeeID Varchar(30) NOT NULL,
AdminID Varchar(30) NOT NULL,
EmployeeName Varchar(30) NOT NULL,
EmployeeSurname Varchar(30) NOT NULL,
EmployeeIDNumber Varchar(30) NOT NULL ,
EmployeePassword Varchar(30) NOT NULL,
PRIMARY KEY (EmployeeID),
FOREIGN KEY (AdminID ) REFERENCES admin(AdminID )
);

CREATE TABLE purchasers(
PurchaserID INT NOT NULL AUTO_INCREMENT ,
PName Varchar(30) NOT NULL,
PSurname Varchar(20) NOT NULL,
PAddress Varchar(20) NOT NULL,
PIDNumber Varchar(14) NOT NULL,
PContactNum Varchar(11) NOT NULL,
EmployeeID Varchar(30),
PRIMARY KEY (PurchaserID ),
FOREIGN KEY (EmployeeID) REFERENCES employees(EmployeeID)
);

CREATE TABLE auctions(
AuctionID bigint NOT NULL AUTO_INCREMENT ,
AdminID Varchar(30) NOT NULL,
AuctionDate date NOT NULL,
TotalIncomeReceived DECIMAL(13,2),
TotalItemsSold int,
TotalDueSARS DECIMAL(13,2),
PRIMARY KEY (AuctionID),
FOREIGN KEY (AdminID) REFERENCES admin(AdminID)
);

CREATE TABLE suppliers(
SupplierID Varchar(30) NOT NULL,
AdminID Varchar(30) NOT NULL,
BusinessOrPrivate Varchar(3) NOT NULL,
SupplierAddress Varchar(30) NOT NULL,
SupplierName Varchar(20) NOT NULL,
SupplierSurname Varchar(20),
PrivateIDnum bigint,
Business_RegNum Varchar(15),
PRIMARY KEY (SupplierID),
FOREIGN KEY (AdminID)REFERENCES admin(AdminID)
);

CREATE TABLE stock(
stockID Varchar(30) NOT NULL,
PurchaserID INT ,
EmployeeID Varchar(30),
SupplierID Varchar(30) NOT NULL,
AuctionID bigint,
DateIn date NOT NULL ,
RegNum Varchar(10) NOT NULL,
EngNum Varchar(20) NOT NULL,
ChasNum Varchar(20) NOT NULL,
AuctionSoldPrice DECIMAL(13,2),
PRIMARY KEY (stockID),
FOREIGN KEY (EmployeeID) REFERENCES employees(EmployeeID),
FOREIGN KEY (PurchaserID) REFERENCES purchasers(PurchaserID),
FOREIGN KEY (SupplierID) REFERENCES suppliers(SupplierID),
FOREIGN KEY (AuctionID) REFERENCES auctions(AuctionID)
);

ALTER TABLE `auctiondb`.`auctions`
DROP FOREIGN KEY `auctions_ibfk_1`;
ALTER TABLE `auctiondb`.`auctions`
ADD CONSTRAINT `auctions_ibfk_1`
  FOREIGN KEY (`AdminID`)
  REFERENCES `auctiondb`.`admin` (`AdminID`)
  ON DELETE NO ACTION;

ALTER TABLE `auctiondb`.`suppliers`
DROP FOREIGN KEY `suppliers_ibfk_1`;
ALTER TABLE `auctiondb`.`suppliers`
ADD CONSTRAINT `suppliers_ibfk_1`
  FOREIGN KEY (`AdminID`)
  REFERENCES `auctiondb`.`admin` (`AdminID`)
  ON DELETE NO ACTION;

ALTER TABLE `auctiondb`.`employees`
DROP FOREIGN KEY `employees_ibfk_1`;
ALTER TABLE `auctiondb`.`employees`
ADD CONSTRAINT `employees_ibfk_1`
  FOREIGN KEY (`AdminID`)
  REFERENCES `auctiondb`.`admin` (`AdminID`)
  ON DELETE NO ACTION;

ALTER TABLE `auctiondb`.`purchasers`
DROP FOREIGN KEY `purchasers_ibfk_1`;
ALTER TABLE `auctiondb`.`purchasers`
ADD CONSTRAINT `purchasers_ibfk_1`
  FOREIGN KEY (`EmployeeID`)
  REFERENCES `auctiondb`.`employees` (`EmployeeID`)
  ON DELETE NO ACTION;

ALTER TABLE `auctiondb`.`stock`
DROP FOREIGN KEY `stock_ibfk_1`;
ALTER TABLE `auctiondb`.`stock`
ADD CONSTRAINT `stock_ibfk_1`
  FOREIGN KEY (`EmployeeID`)
  REFERENCES `auctiondb`.`employees` (`EmployeeID`)
  ON DELETE SET NULL;

ALTER TABLE `auctiondb`.`stock`
DROP FOREIGN KEY `stock_ibfk_2`;
ALTER TABLE `auctiondb`.`stock`
ADD CONSTRAINT `stock_ibfk_2`
  FOREIGN KEY (`PurchaserID`)
  REFERENCES `auctiondb`.`purchasers` (`PurchaserID`)
  ON DELETE SET NULL;

ALTER TABLE `auctiondb`.`stock`
DROP FOREIGN KEY `stock_ibfk_3`;
ALTER TABLE `auctiondb`.`stock`
ADD CONSTRAINT `stock_ibfk_3`
  FOREIGN KEY (`SupplierID`)
  REFERENCES `auctiondb`.`suppliers` (`SupplierID`)
  ON DELETE CASCADE;

ALTER TABLE `auctiondb`.`stock`
DROP FOREIGN KEY `stock_ibfk_4`;
ALTER TABLE `auctiondb`.`stock`
ADD CONSTRAINT `stock_ibfk_4`
  FOREIGN KEY (`AuctionID`)
  REFERENCES `auctiondb`.`auctions` (`AuctionID`)
  ON DELETE SET NULL;

CREATE USER 'jmppt'@'%' IDENTIFIED BY 'Auction123!';

GRANT ALL PRIVILEGES ON auctiondb.* TO 'jmppt'@'%';

USE auctiondb;
INSERT INTO admin(AdminID,AdminName,AdminSurname,AdminIDNumber,AdminPassword) VALUES ('ATest123','TestAccA','TestAccA','123456789123','Test123');
INSERT INTO employees(EmployeeID,AdminID,EmployeeName,EmployeeSurname,EmployeeIDNumber,EmployeePassword) VALUES ('ETest123','ATest123','TestAccE','TestAccE','123456789123','Test123');
```

Maintaining tables.

Purchasers.

```
string query = "INSERT INTO purchasers (PName,PSurname,PAddress,PIDNumber, PContactNum)
VALUES(@buyer_name, @buyer_surname, @buyer_address,@buyer_ID , @buyer_contactNum)";

string query = "UPDATE purchasers SET PName=@buyer_name WHERE PurchaserID=@buyer_ID";

string query = "UPDATE purchasers SET PSurname=@buyer_surname WHERE
PurchaserID=@buyer_ID";

string query = "UPDATE purchasers SET PAddress=@buyer_address WHERE
PurchaserID=@buyer_ID";

string query = "UPDATE purchasers SET PurchaserID=@buyer_ID WHERE PurchaserID=@buyer_ID";

string query = "UPDATE purchasers SET PConactNum=@buyer_contactNum WHERE
PurchaserID=@buyer_ID";

string query = "DELETE FROM purchasers WHERE PurchaserID=@buyer_ID";
```

Employees.

```
string query = "INSERT INTO employees
(EmployeeID,AdminID,EmployeeName,EmployeeSurname,EmployeeIDNumber,EmployeePassword)
VALUES(@empID,@adminID,@empName,@empSurn,@empIdnum,@empPass)";

string query = "UPDATE employees SET EmployeeName=@empName WHERE
EmployeeID=@empID";

string query = "UPDATE employees SET EmployeeSurname=@empSurname WHERE
EmployeeID=@empID";

string query = "UPDATE employees SET EmployeeIDNumber=@empIDNum WHERE
EmployeeID=@empID";

string query = "UPDATE employees SET EmployeePassword=@empPass WHERE
EmployeeID=@empID";

string query = "DELETE FROM employees WHERE EmployeeID=@empID";
```

Stock.

```
string query = "INSERT INTO stock
(stockID,EmployeeID,SupplierID,DateIn,RegNum,EngNum,ChasNum)
VALUES(@stkID,@userID,@suppID,@dateIn,@rNum,@eNum,@chsNum)";

string query = "UPDATE stock SET RegNum=@rNum WHERE stockID=@stkID";

string query = "UPDATE stock SET EngNum=@eNum WHERE stockID=@stkID";

string query = "UPDATE stock SET ChasNum=@chsNum WHERE stockID=@stkID";
string query = "UPDATE stock SET DateIn=@dateIn WHERE stockID=@stkID";

string query = "UPDATE stock SET SupplierID=@suppID WHERE stockID=@stkID";

string query = "DELETE FROM stock WHERE stockID=@stkID";
```

Suppliers.

```
string query = "INSERT INTO suppliers
(SupplierID,AdminID,BusinessOrPrivate,SupplierAddress,SupplierName,SupplierSurname,
PrivateIDnum, Business_RegNum)
VALUES(@supID,@adminID,@supBoP,@supAddress,@supName,@supSurn,@supIdnum,@supBusRe
gNum)";

string query = "UPDATE suppliers SET SupplierName=@supName WHERE SupplierID=@supID";

string query = "UPDATE suppliers SET SupplierSurname=@supSurname WHERE SupplierID=@supID";

string query = "UPDATE suppliers SET PrivateIDnum=@supIDNum WHERE SupplierID=@supID";

string query = "UPDATE suppliers SET SupplierAddress=@supAddress WHERE SupplierID=@supID";

string query = "UPDATE suppliers SET Business_RegNum=@supBusRegNum WHERE
SupplierID=@supID";

string query = "DELETE FROM suppliers WHERE SupplierID=@supID";
```

Auctions.

```
string query = "INSERT INTO auctions(AdminID,AuctionDate) VALUES(@adminID,@aucDate)";

string query = "UPDATE auctions SET
TotalIncomeReceived=@totalIncRec,TotalItemsSold=@totalItems,TotalSars=@totalSars WHERE
AuctionID=@aucID";

string query = "UPDATE stock SET AuctionID=@aucID WHERE stockID=@stockID";

string query = "UPDATE employees SET EmployeeIDNumber=@empIDNum WHERE
EmployeeID=@empID";

string query = "UPDATE employees SET EmployeePassword=@empPass WHERE
EmployeeID=@empID";

string query = "DELETE FROM employees WHERE EmployeeID=@empID";
```

Admins.

```
string query = "INSERT INTO admin
(AdminID,AdminName,AdminSurname,AdminIDNumber,AdminPassword)
VALUES(@adminID,@adminName,@adminSurn,@adminIdnum,@adminPass)";

string query = "UPDATE admin SET AdminName=@adminName WHERE AdminID=@adminID";
string query = "UPDATE admin SET AdminSurname=@adminSurname WHERE AdminID=@adminID";

string query = "UPDATE admin SET AdminIDNumber=@adminIDNum WHERE AdminID=@adminID";

string query = "UPDATE admin SET AdminPassword=@adminPass WHERE AdminID=@adminID";
```

Select Statements.

Auctions.

```
public DataTable GetAuctionList(string query, string aucID)

    {

      if (this.OpenConnection() == true)

      {

        if (string.IsNullOrEmpty(query))

        {

          query = "SELECT * FROM auctions";

        }

        MySqlCommand cmd = new MySqlCommand(query, connection);

        if (query.Contains("LIKE") && !(string.IsNullOrEmpty(aucID)) && query.Contains("@aucID"))

        {

          cmd.Parameters.AddWithValue("@aucID", aucID);

        }

        if (!(string.IsNullOrEmpty(aucID)) && query.Contains("@date"))

        {

          cmd.Parameters.AddWithValue("@date", aucID);

        }

        DataTable dt = new DataTable();

        MySqlDataReader dataReader = cmd.ExecuteReader();          //Create a data reader and
Execute the command

        if (dataReader.HasRows)

        {

          dt.Load(dataReader);


          dataReader.Close();
```

```
                    this.CloseConnection();
                    return dt;
                }
                else
                {
                    dataReader.Close();
                    this.CloseConnection();
                    return null;
                }
            }
            else
            {
                return null;
            }
        }
```

Purchasers.

```
public DataSet SelectBuyer(string sqlQuery, string BuyerID)
        {
            string query = sqlQuery;
            if (sqlQuery.Contains("LIKE"))
            {
                if (this.OpenConnection() == true)
                {
                    MySqlCommand cmd = new MySqlCommand(query, connection);
                    cmd.Parameters.AddWithValue("@BuyerID", BuyerID);

                    MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
                    DataSet ds = new DataSet();
                    adap.Fill(ds, "Source Table");

                    this.CloseConnection();    //close Connection
                    return ds;          //return dataset for the datagridview
                }
                else
                {
                    return null;
                }

            }
            else
            {
                if (this.OpenConnection() == true)
                {
                    MySqlCommand cmd = new MySqlCommand(query, connection);
                    MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
                    DataSet ds = new DataSet();

                    adap.Fill(ds, "Source Table");

                    this.CloseConnection();    //close Connection

                    return ds;          //return dataset for the datagridview
                }
                else
                {
                    return null;
                }
            }
        }
```

Employee.

```
string query = sqlQuery;
            if (sqlQuery.Contains("LIKE"))
            {
                if (this.OpenConnection() == true)
                {
                    MySqlCommand cmd = new MySqlCommand(query, connection);
                    cmd.Parameters.AddWithValue("@empID", empID);

                    MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
                    DataSet ds = new DataSet();
                    adap.Fill(ds,"Source Table");

                    this.CloseConnection();    //close Connection

                    return ds;              //return dataset for the datagridview
                }
                else
                {
                    return null;
                }

            }
            else
            {
                if (this.OpenConnection() == true)
                {
                    MySqlCommand cmd = new MySqlCommand(query, connection);
                    MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
                    DataSet ds = new DataSet();
                    adap.Fill(ds, "Source Table");


                    this.CloseConnection();    //close Connection


                    return ds;             //return dataset for the datagridview
                }
                else
                {
                    return null;
                }
            }
            //Open connection


        }
```

Stock.

```
string query = sqlQuery;
        if (sqlQuery.Contains("LIKE"))
        {
            if (this.OpenConnection() == true)
            {
                MySqlCommand cmd = new MySqlCommand(query, connection);
                cmd.Parameters.AddWithValue("@stockID", stockID);


                MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
                DataSet ds = new DataSet();
                adap.Fill(ds, "Source Table");


                this.CloseConnection();    //close Connection


                return ds;             //return dataset for the datagridview
            }
            else
            {
                return null;
            }

        }
        else
        {
            if (this.OpenConnection() == true)
            {
                MySqlCommand cmd = new MySqlCommand(query, connection);
                MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
                DataSet ds = new DataSet();

                adap.Fill(ds, "Source Table");

                this.CloseConnection();    //close Connection

                return ds;            //return dataset for the datagridview
            }
            else
            {
                return null;
            }
        }

    }
```

Suppliers.

```
string query = sqlQuery;
      if (sqlQuery.Contains("LIKE"))
      {
        if (this.OpenConnection() == true)
        {
          MySqlCommand cmd = new MySqlCommand(query, connection);
          cmd.Parameters.AddWithValue("@supID", supID);

          MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
          DataSet ds = new DataSet();
          adap.Fill(ds, "Source Table");

          this.CloseConnection();    //close Connection

          return ds;          //return dataset for the datagridview
        }
        else
        {
          return null;
        }

      }
      else
      {
        if (this.OpenConnection() == true)
        {
          MySqlCommand cmd = new MySqlCommand(query, connection);
          MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
          DataSet ds = new DataSet();
```

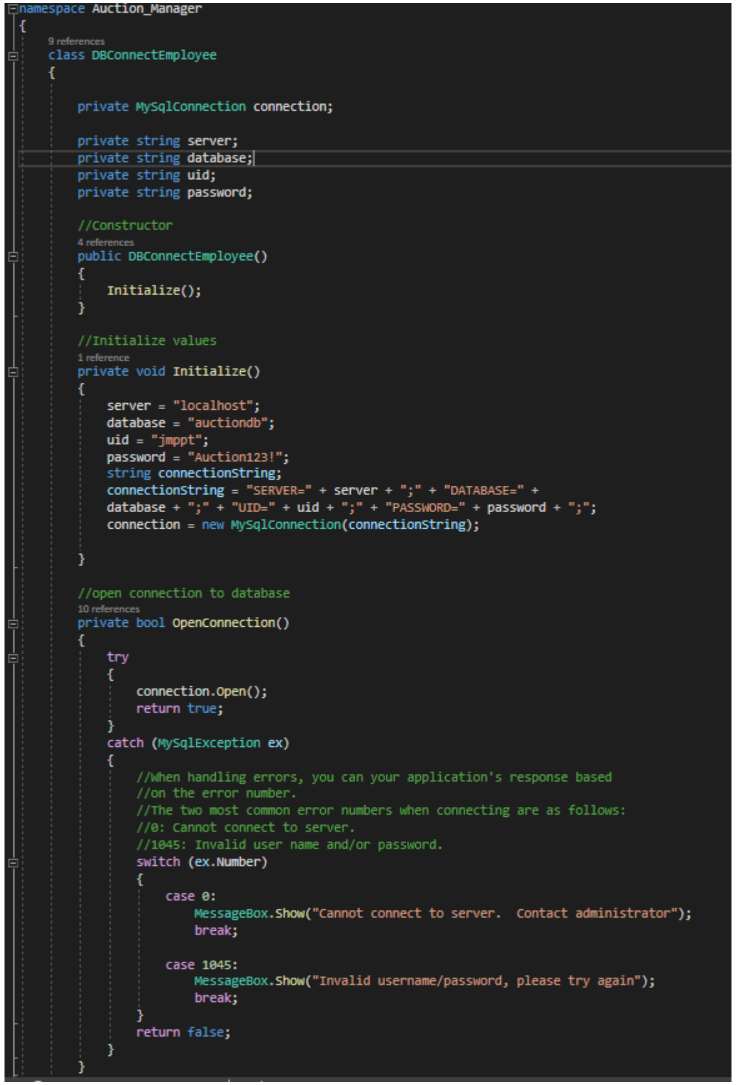
```
                    adap.Fill(ds, "Source Table");


                    this.CloseConnection();    //close Connection


                    return ds;          //return dataset for the datagridview
                }
                else
                {
                    return null;
                }
            }
            //Open connection


        }
```

Admins.

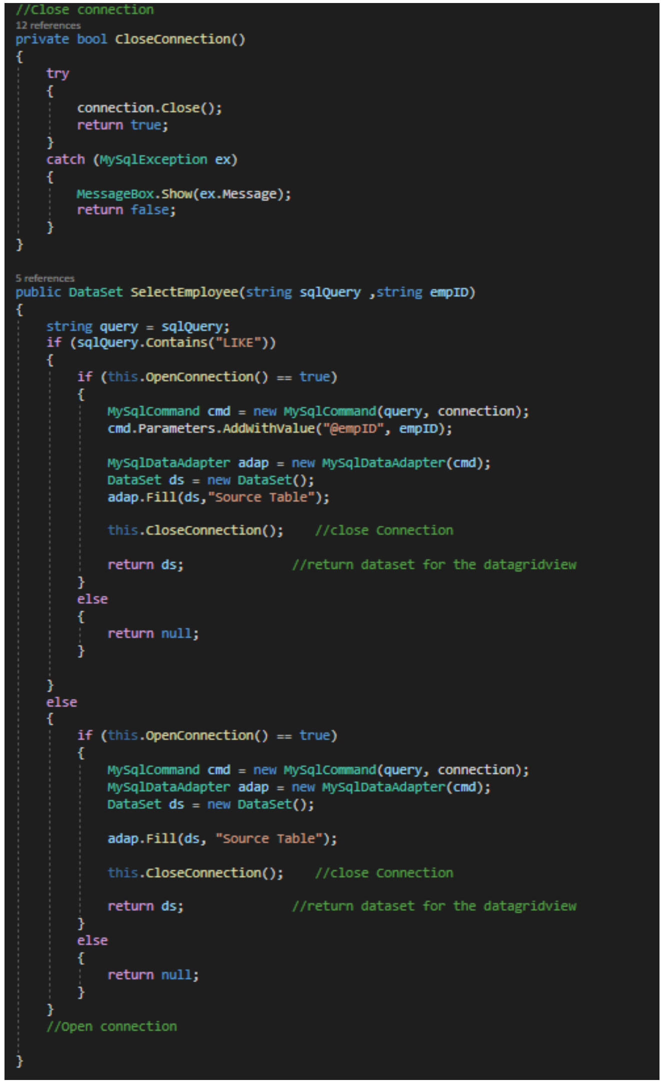
```
string query = sqlQuery;
            if (sqlQuery.Contains("LIKE"))
            {
                if (this.OpenConnection() == true)
                {
                    MySqlCommand cmd = new MySqlCommand(query, connection);
                    cmd.Parameters.AddWithValue("@adminID", adminID);
                    MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
                    DataSet ds = new DataSet();
                    adap.Fill(ds, "Source Table");


                    this.CloseConnection();


                    return ds;
                }
                else
```

```
    {
        return null;
    }

}
else
{
    if (this.OpenConnection() == true)
    {
        MySqlCommand cmd = new MySqlCommand(query, connection);
        MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
        DataSet ds = new DataSet();
        adap.Fill(ds, "Source Table");

        this.CloseConnection();

        return ds;
    }
    else
    {
        return null;
    }
}
```

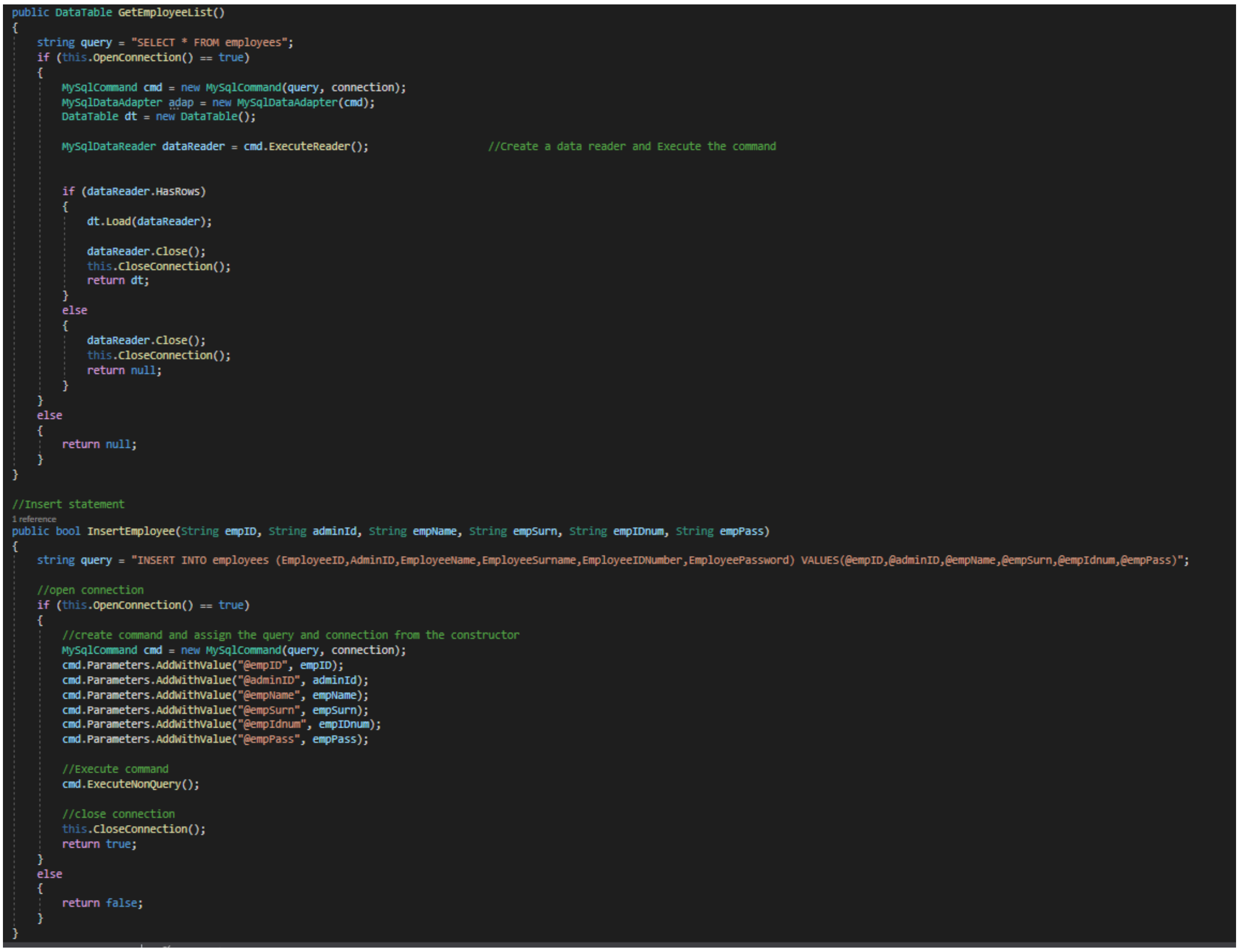
Screen Print of example Programming Code.



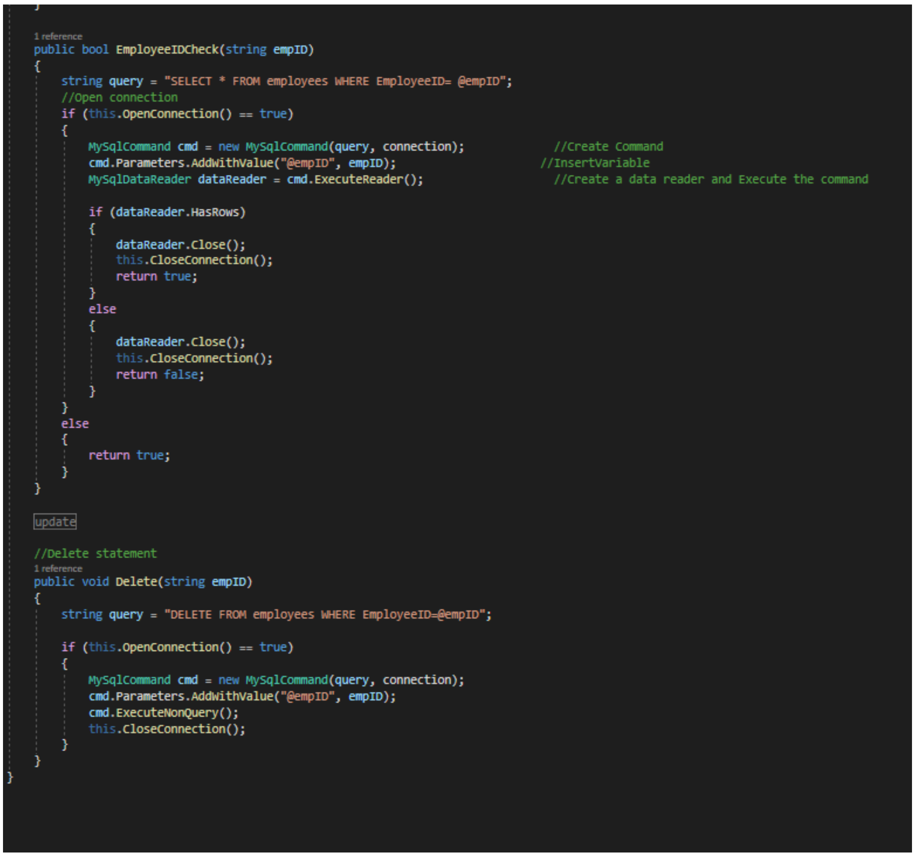
```
namespace Auction_Manager
{
    9 references
    class DBConnectEmployee
    {

        private MySqlConnection connection;

        private string server;
        private string database;
        private string uid;
        private string password;

        //Constructor
        4 references
        public DBConnectEmployee()
        {
            Initialize();
        }

        //Initialize values
        1 reference
        private void Initialize()
        {
            server = "localhost";
            database = "auctiondb";
            uid = "jmppt";
            password = "Auction123!";
            string connectionString;
            connectionString = "SERVER=" + server + ";" + "DATABASE=" +
            database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";
            connection = new MySqlConnection(connectionString);

        }

        //open connection to database
        10 references
        private bool OpenConnection()
        {
            try
            {
                connection.Open();
                return true;
            }
            catch (MySqlException ex)
            {
                //When handling errors, you can your application's response based
                //on the error number.
                //The two most common error numbers when connecting are as follows:
                //0: Cannot connect to server.
                //1045: Invalid user name and/or password.
                switch (ex.Number)
                {
                    case 0:
                        MessageBox.Show("Cannot connect to server.  Contact administrator");
                        break;

                    case 1045:
                        MessageBox.Show("Invalid username/password, please try again");
                        break;
                }
                return false;
            }
        }
```



```
//Close connection
12 references
private bool CloseConnection()
{
    try
    {
        connection.Close();
        return true;
    }
    catch (MySqlException ex)
    {
        MessageBox.Show(ex.Message);
        return false;
    }
}

5 references
public DataSet SelectEmployee(string sqlQuery ,string empID)
{
    string query = sqlQuery;
    if (sqlQuery.Contains("LIKE"))
    {
        if (this.OpenConnection() == true)
        {
            MySqlCommand cmd = new MySqlCommand(query, connection);
            cmd.Parameters.AddWithValue("@empID", empID);

            MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
            DataSet ds = new DataSet();
            adap.Fill(ds,"Source Table");

            this.CloseConnection();    //close Connection

            return ds;              //return dataset for the datagridview
        }
        else
        {
            return null;
        }

    }
    else
    {
        if (this.OpenConnection() == true)
        {
            MySqlCommand cmd = new MySqlCommand(query, connection);
            MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
            DataSet ds = new DataSet();

            adap.Fill(ds, "Source Table");

            this.CloseConnection();    //close Connection

            return ds;              //return dataset for the datagridview
        }
        else
        {
            return null;
        }
    }
    //Open connection

}
```

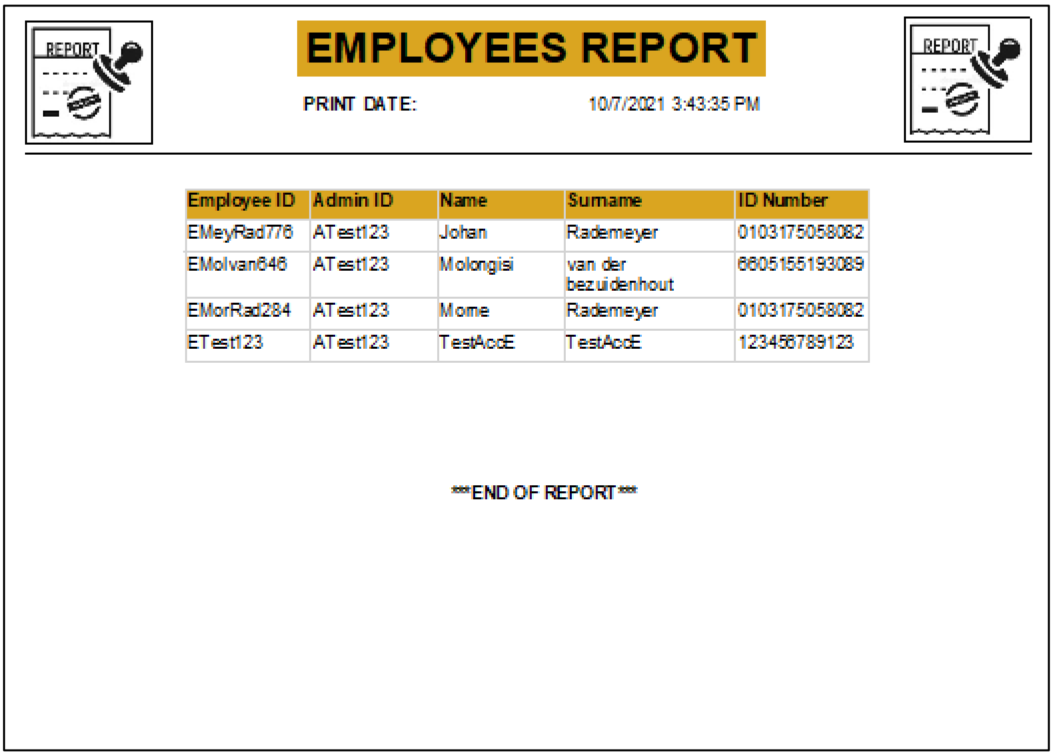


```
public DataTable GetEmployeeList()
{
    string query = "SELECT * FROM employees";
    if (this.OpenConnection() == true)
    {
        MySqlCommand cmd = new MySqlCommand(query, connection);
        MySqlDataAdapter adap = new MySqlDataAdapter(cmd);
        DataTable dt = new DataTable();

        MySqlDataReader dataReader = cmd.ExecuteReader();                   //Create a data reader and Execute the command


        if (dataReader.HasRows)
        {
            dt.Load(dataReader);

            dataReader.Close();
            this.CloseConnection();
            return dt;
        }
        else
        {
            dataReader.Close();
            this.CloseConnection();
            return null;
        }
    }
    else
    {
        return null;
    }
}

//Insert statement
1 reference
public bool InsertEmployee(String empID, String adminId, String empName, String empSurn, String empIDnum, String empPass)
{
    string query = "INSERT INTO employees (EmployeeID,AdminID,EmployeeName,EmployeeSurname,EmployeeIDNumber,EmployeePassword) VALUES(@empID,@adminID,@empName,@empSurn,@empIdnum,@empPass)";

    //open connection
    if (this.OpenConnection() == true)
    {
        //create command and assign the query and connection from the constructor
        MySqlCommand cmd = new MySqlCommand(query, connection);
        cmd.Parameters.AddWithValue("@empID", empID);
        cmd.Parameters.AddWithValue("@adminID", adminId);
        cmd.Parameters.AddWithValue("@empName", empName);
        cmd.Parameters.AddWithValue("@empSurn", empSurn);
        cmd.Parameters.AddWithValue("@empIdnum", empIDnum);
        cmd.Parameters.AddWithValue("@empPass", empPass);

        //Execute command
        cmd.ExecuteNonQuery();

        //close connection
        this.CloseConnection();
        return true;
    }
    else
    {
        return false;
    }
}
```



```
    }
    1 reference
    public bool EmployeeIDCheck(string empID)
    {
        string query = "SELECT * FROM employees WHERE EmployeeID= @empID";
        //Open connection
        if (this.OpenConnection() == true)
        {
            MySqlCommand cmd = new MySqlCommand(query, connection);            //Create Command
            cmd.Parameters.AddWithValue("@empID", empID);                    //InsertVariable
            MySqlDataReader dataReader = cmd.ExecuteReader();                  //Create a data reader and Execute the command

            if (dataReader.HasRows)
            {
                dataReader.Close();
                this.CloseConnection();
                return true;
            }
            else
            {
                dataReader.Close();
                this.CloseConnection();
                return false;
            }
        }
        else
        {
            return true;
        }
    }

    update

    //Delete statement
    1 reference
    public void Delete(string empID)
    {
        string query = "DELETE FROM employees WHERE EmployeeID=@empID";

        if (this.OpenConnection() == true)
        {
            MySqlCommand cmd = new MySqlCommand(query, connection);
            cmd.Parameters.AddWithValue("@empID", empID);
            cmd.ExecuteNonQuery();
            this.CloseConnection();
        }
    }
}
```

Screen prints of two reports generated.

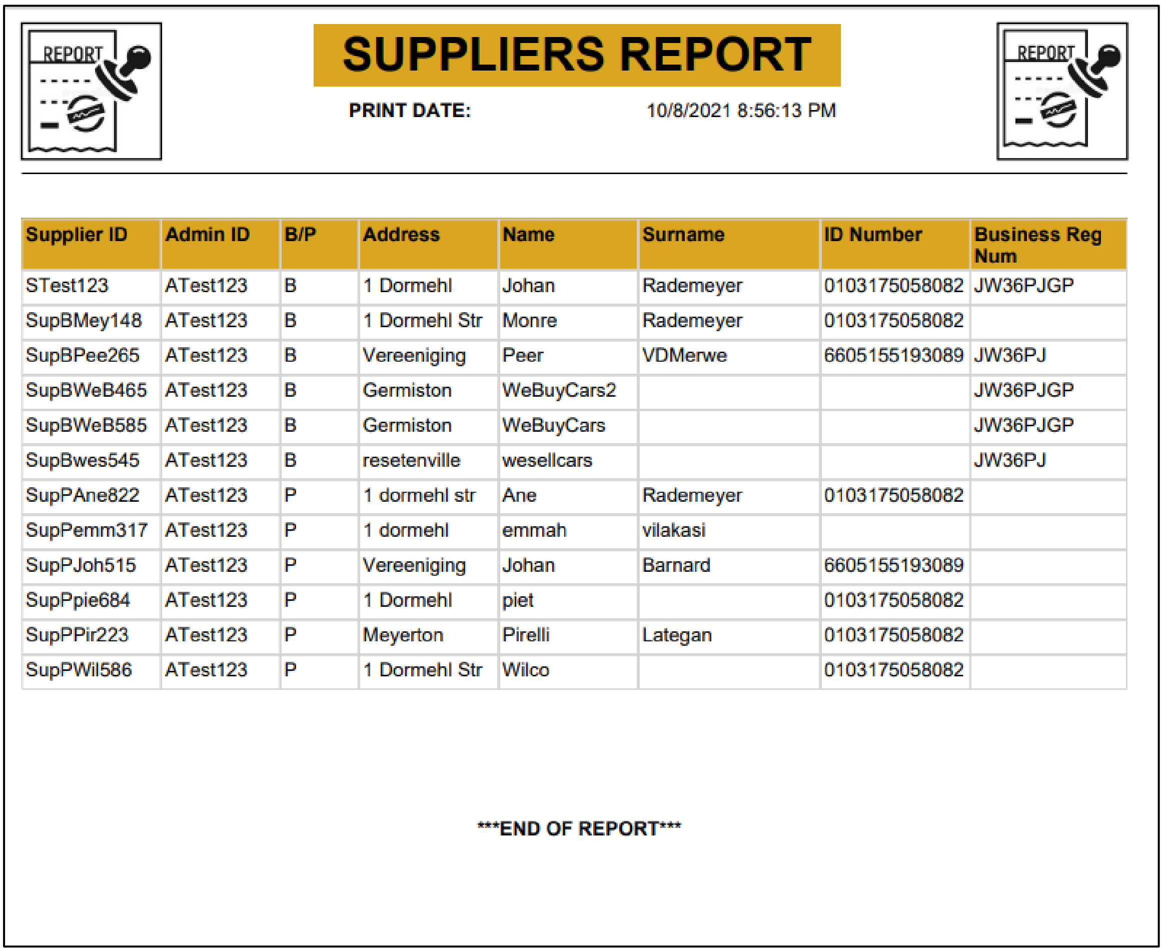


EMPLOYEES REPORT

PRINT DATE: 10/7/2021 3:43:35 PM

Employee ID	Admin ID	Name	Surname	ID Number
EMeyRad776	ATest123	Johan	Rademeyer	0103175058082
EMolvan646	ATest123	Molongisi	van der bezuidenhout	6605155193089
EMorRad284	ATest123	Morne	Rademeyer	0103175058082
ETest123	ATest123	TestAccE	TestAccE	123456789123

END OF REPORT



SUPPLIERS REPORT

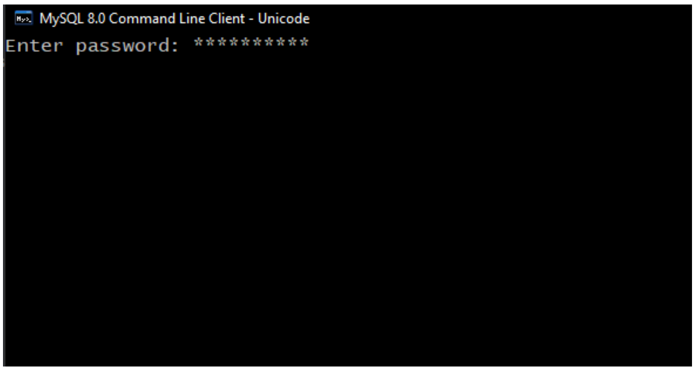
PRINT DATE: 10/8/2021 8:56:13 PM

Supplier ID	Admin ID	B/P	Address	Name	Surname	ID Number	Business Reg Num
STest123	ATest123	B	1 Dormehl	Johan	Rademeyer	0103175058082	JW36PJGP
SupBMey148	ATest123	B	1 Dormehl Str	Monre	Rademeyer	0103175058082	
SupBPee265	ATest123	B	Vereeniging	Peer	VDMerwe	6605155193089	JW36PJ
SupBWeB465	ATest123	B	Germiston	WeBuyCars2			JW36PJGP
SupBWeB585	ATest123	B	Germiston	WeBuyCars			JW36PJGP
SupBwes545	ATest123	B	resetenville	wesellcars			JW36PJ
SupPAne822	ATest123	P	1 dormehl str	Ane	Rademeyer	0103175058082	
SupPemm317	ATest123	P	1 dormehl	emmah	vilakasi		
SupPJoh515	ATest123	P	Vereeniging	Johan	Barnard	6605155193089	
SupPpie684	ATest123	P	1 Dormehl	piet		0103175058082	
SupPPir223	ATest123	P	Meyerton	Pirelli	Lategan	0103175058082	
SupPWil586	ATest123	P	1 Dormehl Str	Wilco		0103175058082	

END OF REPORT

User Manual.

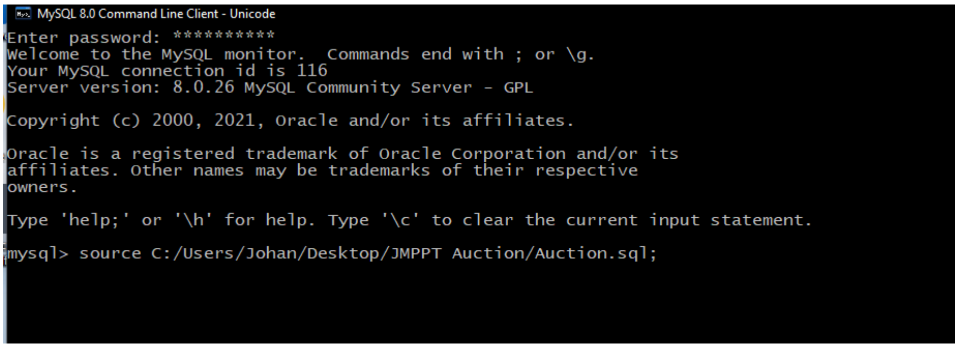
Ensure that the Programs Files which are contained within the Debug folder of the project for end Users, I would create a folder and copy the contents of the debug folder into a Folder called JMPPT Auction. Within the folder is an Auction Manger.exe file that the user needs to launch to run the program. A shortcut can also be made of this executable file for ease of access. For the Initial installation on the host pc, MySQL Workbench will need to be installed with MySQL server. On the host computer you will have to open MySQL command line client and enter the password you're provided during installation.



```
Enter password: **********
```

Thereafter, enter this in the following format:

source C:/Users/YourUserName/Desktop/JMPPT Auction/Auction.sql;



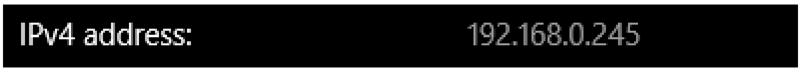
```
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 116
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> source C:/Users/Johan/Desktop/JMPPT Auction/Auction.sql;
```

The command line may vary depending on the location of the program files.

Once this has completed the database server is now up and running.

Next you will need to check what the host computer's Local IP Address is.

On Windows 10 this can be done by going to Settings > Network & Internet > Properties > then scroll all the way down till you see this:



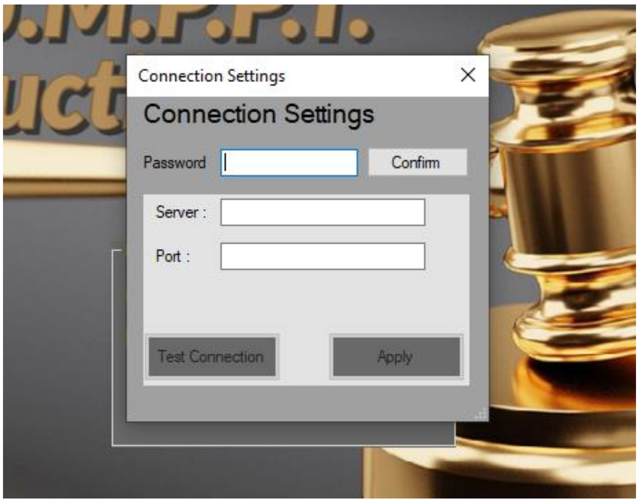
IPv4 address: 192.168.0.245

Next you will need to open the Auction Mananger.exe file.

You will be greeted with a log in window.

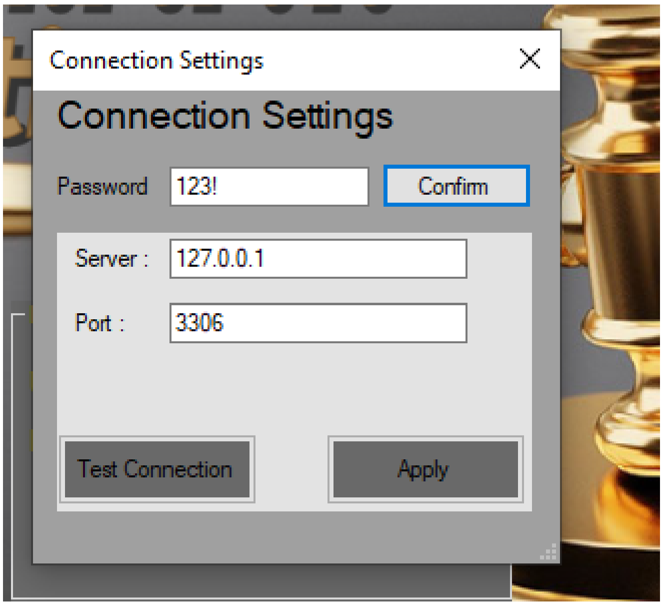
In the bottom left there is a button called "Database Connection", Click on it.

You will now have a small window in front of you:



Enter "123!" into the password field then click the “Confirm” button.

The window will now show an IP Address in the server field and a port number in the port field.



Password 123!

Server : 127.0.0.1

Port : 3306

Delete the contents of the Server field and enter your local IP Address e.g., 192.168.0.245 in previous picture.

Then click on “Apply” then close the window.

At first use Username: ATest123, Password: Test123. Once logged in you can proceed to create a new User - Manage Users > Add > then enter the necessary details and specify whether you want to make this user an admin or not .

Setting up second user computer

Copy the files onto the computer.

Open the executable file.

You will be greeted with a log in window in the bottom left. There is a button called database connection.

Click it.

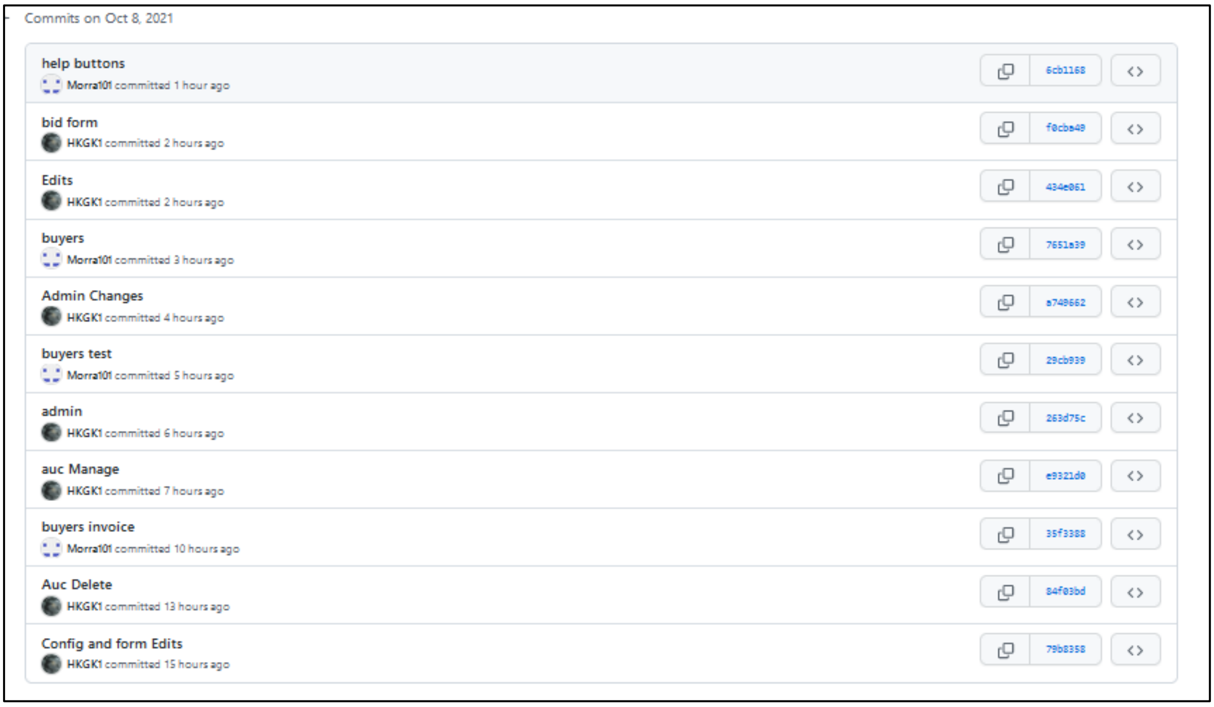
You will now have a small window in front of you.

Enter the local IP of your host computer.

And that’s it .

Detail diary.

NOTE: Too many screenshots to give full diary.



Commits on Oct 8, 2021

help buttons
Morra101 committed 1 hour ago — 6cb1168

bid form
HKGK1 committed 2 hours ago — f0cba49

Edits
HKGK1 committed 2 hours ago — 434e061

buyers
Morra101 committed 3 hours ago — 7651a39

Admin Changes
HKGK1 committed 4 hours ago — a749662

buyers test
Morra101 committed 5 hours ago — 29cb939

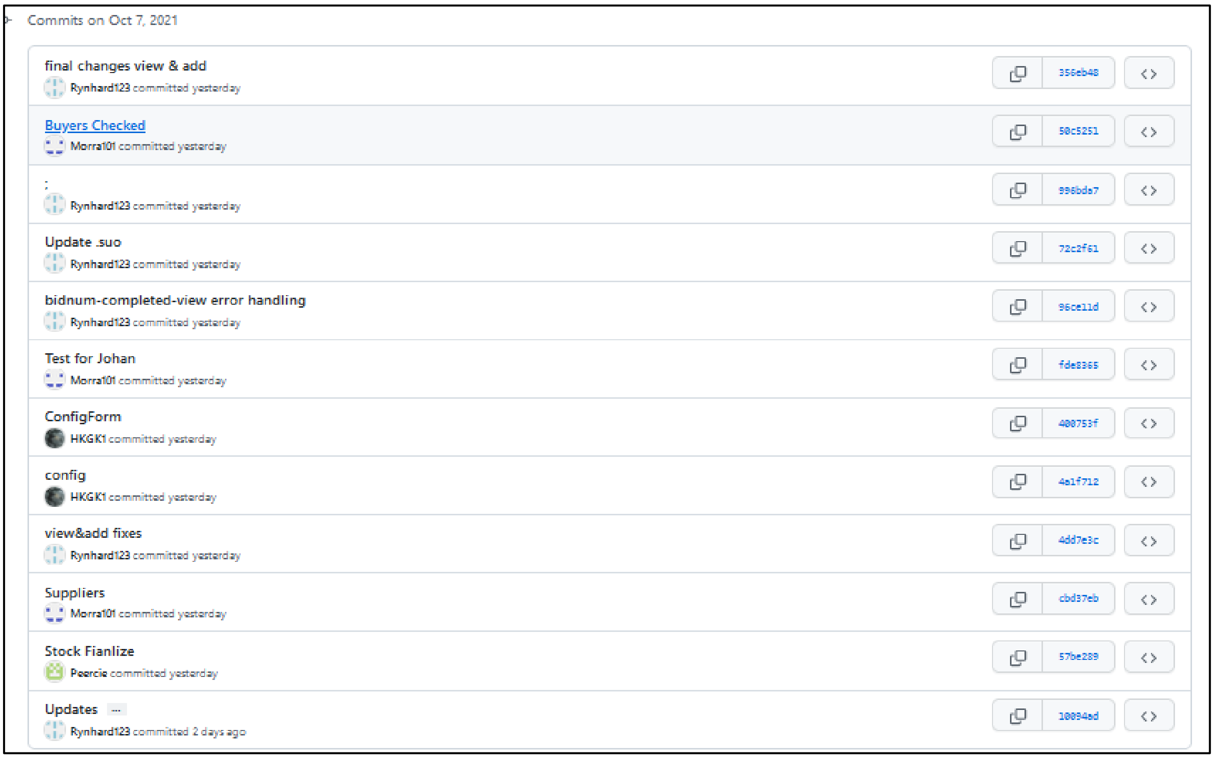
admin
HKGK1 committed 6 hours ago — 263d75c

auc Manage
HKGK1 committed 7 hours ago — e9321d0

buyers invoice
Morra101 committed 10 hours ago — 35f3388

Auc Delete
HKGK1 committed 13 hours ago — 84f03bd

Config and form Edits
HKGK1 committed 15 hours ago — 79b8358



Commits on Oct 7, 2021

final changes view & add
Rynhard123 committed yesterday
356eb48

Buyers Checked
Morra101 committed yesterday
50c5251

;
Rynhard123 committed yesterday
996bda7

Update .suo
Rynhard123 committed yesterday
72c2f61

bidnum-completed-view error handling
Rynhard123 committed yesterday
96ce11d

Test for Johan
Morra101 committed yesterday
fde8365

ConfigForm
HKGK1 committed yesterday
400753f

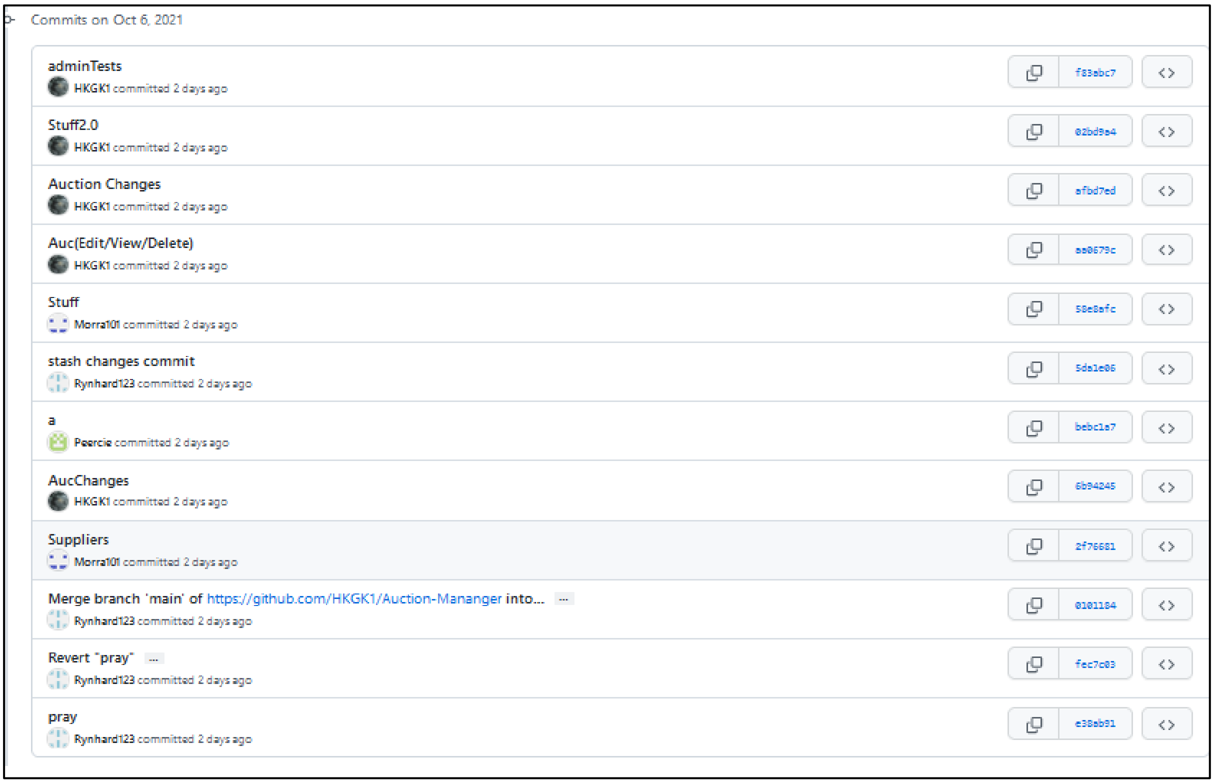
config
HKGK1 committed yesterday
4a1f712

view&add fixes
Rynhard123 committed yesterday
4dd7e3c

Suppliers
Morra101 committed yesterday
cbd37eb

Stock Fianlize
Peercie committed yesterday
57be289

Updates …
Rynhard123 committed 2 days ago
10894ad



Commits on Oct 6, 2021

adminTests
HKGK1 committed 2 days ago
f83abc7

Stuff2.0
HKGK1 committed 2 days ago
02bd9a4

Auction Changes
HKGK1 committed 2 days ago
afbd7ed

Auc(Edit/View/Delete)
HKGK1 committed 2 days ago
aa0679c

Stuff
Morra101 committed 2 days ago
58e8afc

stash changes commit
Rynhard123 committed 2 days ago
5da1e06

a
Peercie committed 2 days ago
bebc1a7

AucChanges
HKGK1 committed 2 days ago
6b94245

Suppliers
Morra101 committed 2 days ago
2f76681

Merge branch 'main' of https://github.com/HKGK1/Auction-Mananger into… …
Rynhard123 committed 2 days ago
0101184

Revert "pray" …
Rynhard123 committed 2 days ago
fec7c03

pray
Rynhard123 committed 2 days ago
e38ab91