**GLASS CUTTING, BEVELING AND SALE MANAGEMENT SYSTEM**

**GROUP MEMBERS:**

**Maaz Sabahuddin [02-134171-131]**

**Muhammad Shahrukh Mahmood [02-134171-119]**

**Momal Qadir [02-134171-030]**

**Azka Khan [02-134171-039]**

**A Project Report submitted for**

**DATABASE MANAGEMENT SYSTEM**



**Department of Computer Science**

**Bahria University, Karachi Campus**

**TABLE OF CONTENTS:**

**Introduction:** **………………………………………………….…1**

**Problem Statement: ……………………………………………..2**

**Problem Scope: …………………………………………………. 3**

**Functionality: ………………………………………………..….. 3**

**Code: ………………………………….…………………………. 4**

**Normalization: …………………………………………….……..**

**ERD: ……………………………………………………...............INTRODUCTION:**

Glass manufacturing has become mature business in the developing countries of the world. Like other businesses, glass manufacturers are also looking for ways to make thing easier. Beveled glass is usually made by taking thick glass and creating an angled surface cut (bevel) around the entire periphery. Bevels act as prisms in the sunlight creating an interesting color diffraction. There are many types of glass some of them are as follows:

**TEMPERED GLASS:**

This product is used most often for shelves, some fire places and table tops. Tempered Glass breaks into small pieces when broken and usually cracks.

**ANNEALED GLASS:**

Often used in small pieces. This product is not considered a safety glass. Annealed Glass doesn’t shatter into small pieces. If broken, the piece will crack into large shards. Maximum annealed glass size is 28x28.For pieces larger then this please select Tempered Glass.

**MIRROR GLASS:**

This product is used to replace any mirror. Mirrors are not tempered and if broken will break into large shards.

**CERAMIC GLASS:**

These products are made to withstand very high temperature and are often used in woodstoves, glass stoves, ovens, halogen lamps and laboratories. Ceramic glasses do not shatter. If broken, the piece will crack into large shards.

Our project will cover all the aspects of business, from analyzing the market, confirming availability of various necessities. Our project Glass Cutting, Beveling and Sale Management System is basically about selling and purchasing of glass. It is designed in a way that it can be accessed for two purposes. One purpose is for Admin use and the other purpose is for User.

If you are user and want to purchase glass then firstly you will provide all your details, then you will select glass type, shape, dimension and finally the quantity. After that you will be able to see your payment receipt.

If you are an admin then firstly you will login to your account then an admin panel will be opened in which there will be 3 buttons(Sales, Records, Vendor). If you want to view all the records then you should click on Records’ button. By clicking on Vendor button u will be able to see vendor details. You can view one day sale and monthly sale by clicking on sales’ button.

**PROBLEM STATEMENT:**

* **Lack of immediate retrievals: -** The information is very difficult to retrieve and to find particular information like- E.g. - To find out about any purchaser’s order details, the user has to go through various registers. This results in inconvenience and wastage of time.
* **Lack of immediate information storage: -** The information generated by various transactions takes time and efforts to be stored at right place.
* **Error prone manual calculation: -** Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of bills based on various orders.
* **Preparation of accurate and prompt reports: -** This becomes a difficult task as information is difficult to collect from various registers.

**Project Scope:**

• In our Glass Cutting Project and the database will be maintain

• The main functionality of our project is to retrieve and store data customer and their order .

• We will generate Bill of the customer and issue the delivery date and status.

• Our interface will be user free

• We will maintain the Vendors manage our Sales annually

• We will use JOINS and TRIGGERS tor managing our database

**Functionality:**

1. Super Admin User: Super Admin User will give access to other users as well as can create the tables, view tables, can insert data

2. Mid-Level User: Mid-Level User can only view his table, update table and insert into table.

3. End User: End User can only order.

**Code:**

**TOP GLASS:**

CREATE OR REPLACE PROCEDURE topglass

a number;

b number;

c number;

d number;

BEGIN

SELECT count(glasstype) into a from glass\_details where glasstype='Tempered Glass';

SELECT count(glasstype) into b from glass\_details where glasstype='Ceramic Glass';

SELECT count(glasstype) into c from glass\_details where glasstype='Mirror Glass';

SELECT count(glasstype) into d from glass\_details where glasstype='Annealed Glass';

if a>b and a>c and a>d then dbms\_output.put\_line('Tempered Glass');

else if b>a and b>c and b>d then dbms\_output.put\_line('Ceramic Glass');

else if c>a and c>b and c>d then dbms\_output.put\_line('Mirror Glass');

else if d>a and d>b and d>c then dbms\_output.put\_line('Annealed Glass');

end if;

end if;

end if;

end if;

END;

/

**Method 2:**

SELECT a.glasstype

from(select count(\*) as TOTAL, glasstype from glass\_details

group by glasstype) a,

(select MAX(TOTAL) as MAXTOTAL

from (select count(\*) as TOTAL, glasstype from glass\_details

group by glasstype)) b

where a.TOTAL = b.MAXTOTAL

**PROCEDURE FOR DISPLAY ORDER OF A PARTICULAR USER:**

CREATE OR REPLACE PROCEDURE showOrderDetails

is

CURSOR abc

is

SELECT a.order\_id "O\_ID",a.user\_id "U\_ID",b.glasstype "GT" ,b.glassshape "GS",b.glassthickness "GTH",

b.glassdimension "GD",a.order\_date "OD",d.quantity "Q",d.amount "A"

FROM glass\_user\_order a inner join glass\_details b

ON a.glass\_id=b.glass\_id

INNER JOIN glass\_price d

ON a.price\_id=d.price\_id AND b.glasstype='Mirror Glass';

BEGIN

dbms\_output.put\_line(' O\_ID '||' U\_ID '||' GT '||' GS '||' GTH '

||' GD '||' OD '||' Q '||' A ');

for cur in abc

LOOP

dbms\_output.put\_line(cur.O\_ID||' '|| cur.U\_ID||' '|| cur.GT||' '|| cur.GS||' '|| cur.GTH

||' '||cur.GD||' '|| cur.OD||' '|| cur.Q||' '||cur.A);

END LOOP;

END;

/

**VIEW**

CREATE VIEW TopProduct

as

SELECT a.glasstype

from(select count(\*) as TOTAL, glasstype from maaz.glass\_details

group by glasstype) a,

(select MAX(TOTAL) as MAXTOTAL

from (select count(\*) as TOTAL, glasstype from maaz.glass\_details

group by glasstype)) b

where a.TOTAL = b.MAXTOTAL

**VIEW top customer**

CREATE VIEW topcustomer

as

SELECT name from maaz.user\_details

where USER\_ID = (SELECT a.USER\_ID

from(SELECT count(\*) as TOTAL, user\_id from maaz.glass\_user\_order

group by USER\_ID)a, (SELECT max(total) as MAXTOTAL

from (SELECT COUNT(\*) as TOTAL, USER\_ID from maaz.glass\_user\_order group by USER\_ID))b

where a.TOTAL=b.MAXTOTAL)

/

**FUNCTION userappear**

CREATE OR REPLACE FUNCTION userappear(id number) return number

is

a number;

BEGIN

select count(user\_id) into a from glass\_user\_order where user\_id=id;

return a;

END;

/

**TRIGGER myuser**

CREATE OR REPLACE TRIGGER user\_trigger

before delete on user\_details

for each row

BEGIN

INSERT into log VALUES(sysdate, :old.user\_id, :old.name, :old.cnic, :old.cno, :old.city, :old.dateofreg);

END;

/

**Vendor Trigger**

CREATE OR REPLACE TRIGGER vendor\_trigger

before delete on vendor\_details

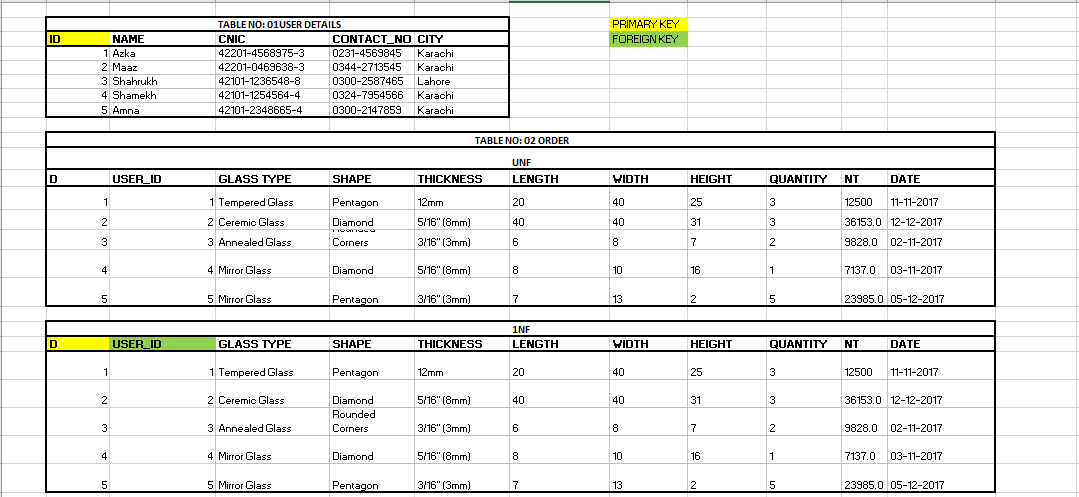
for each row

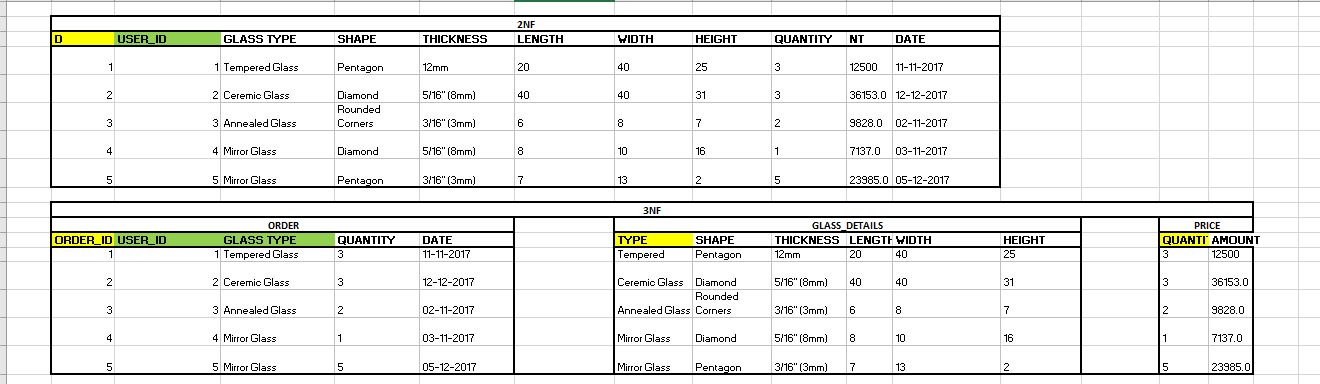
BEGIN

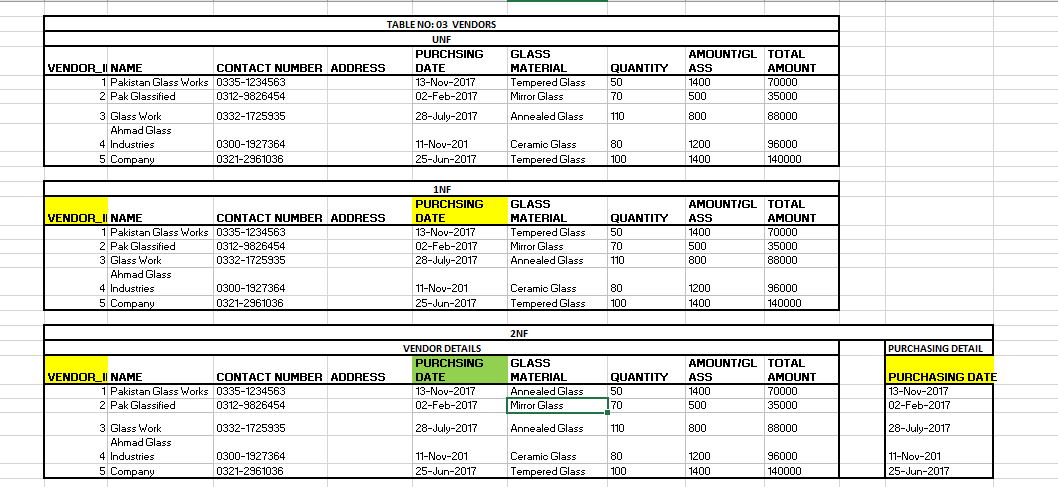
INSERT into log\_v VALUES(sysdate, :old.v\_ID, :old.v\_name, :old.v\_cno);

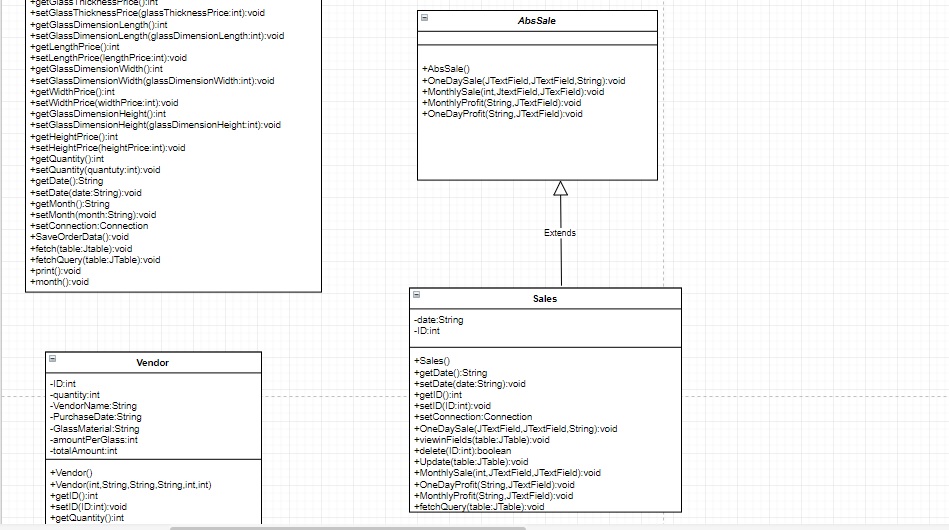
END;

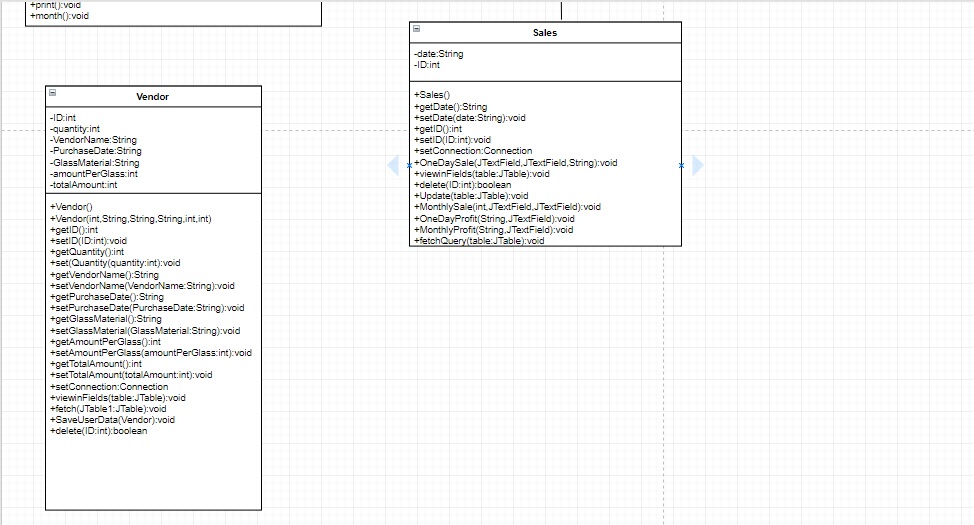
**Normalization:**





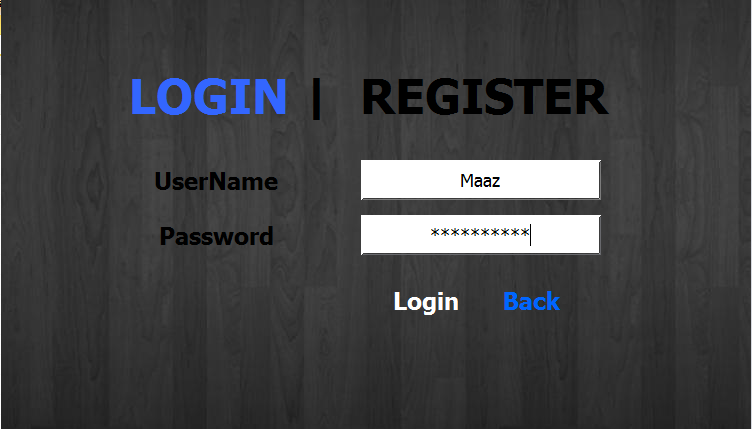


****

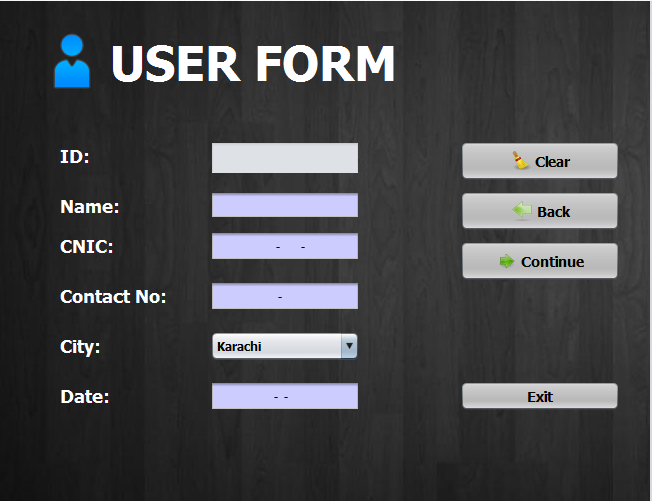
****

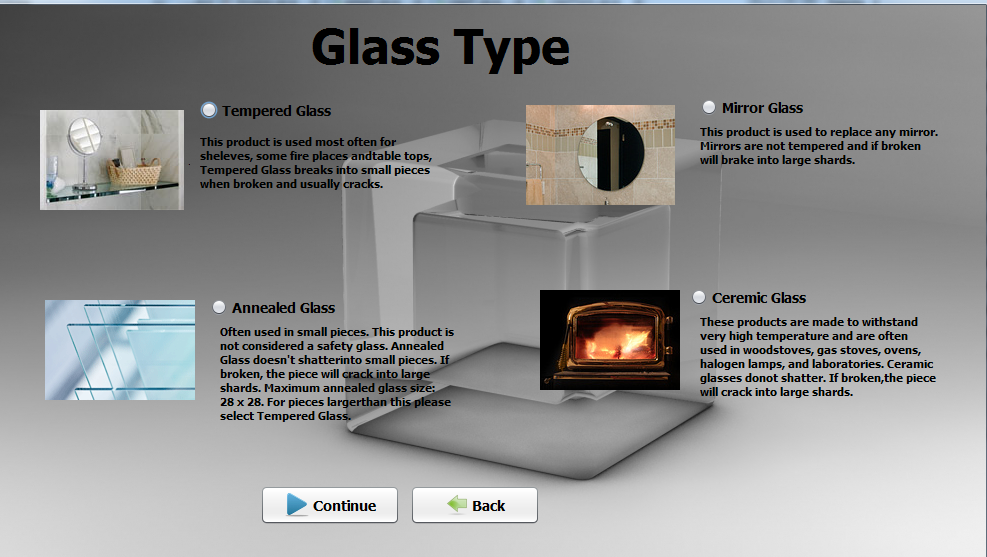
**SCREEN SHOTS:**

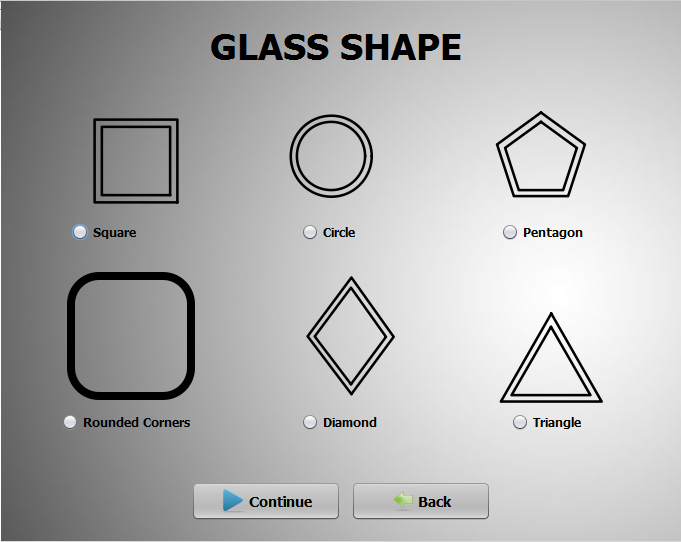
MAIN FORM:

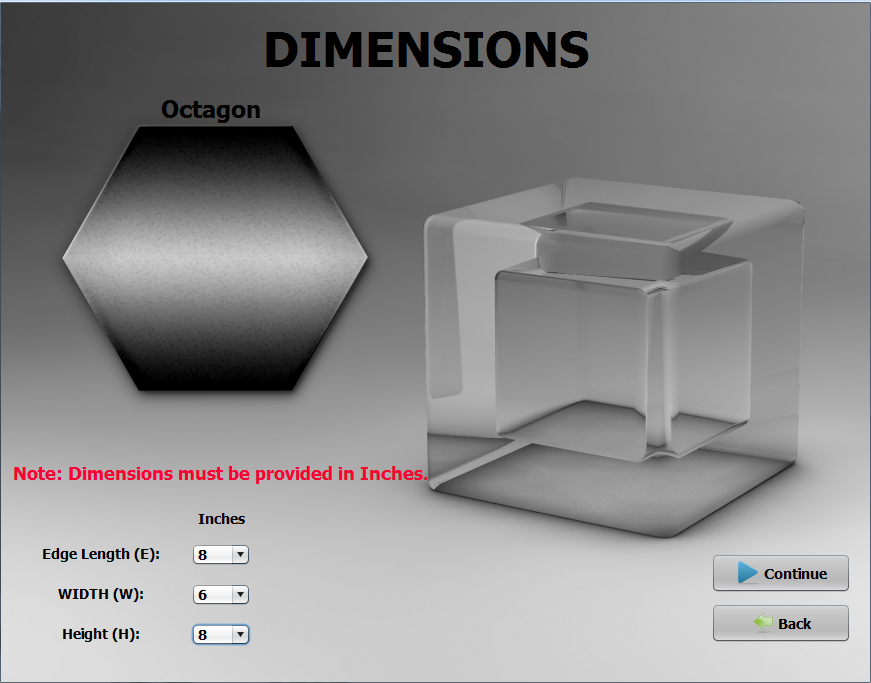
LOGINFORM:

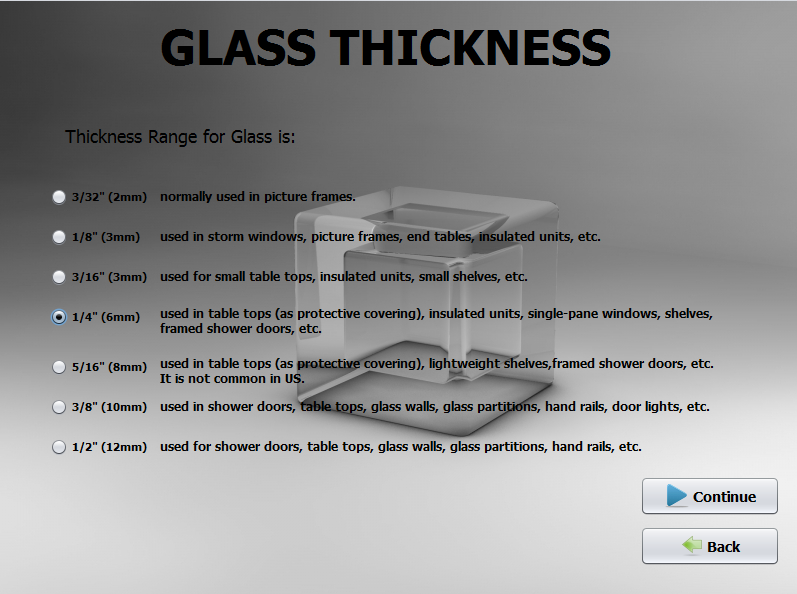
USERFORM:

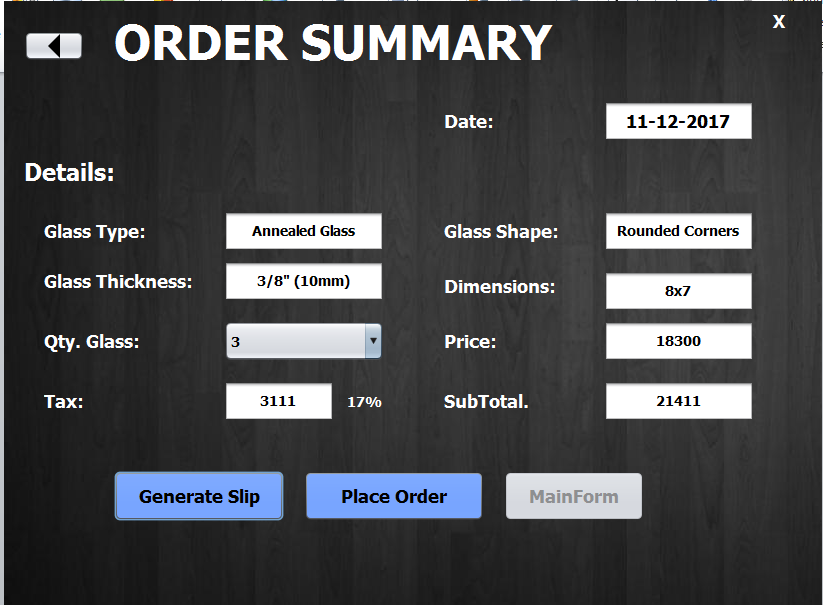


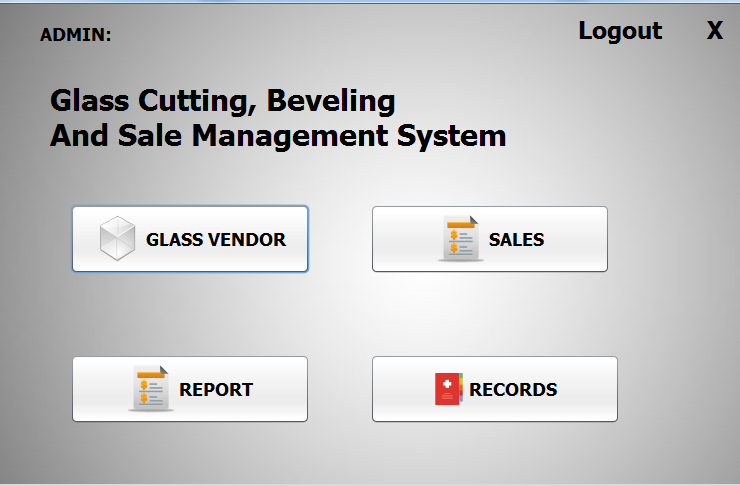
GLASS TYPE FORM:

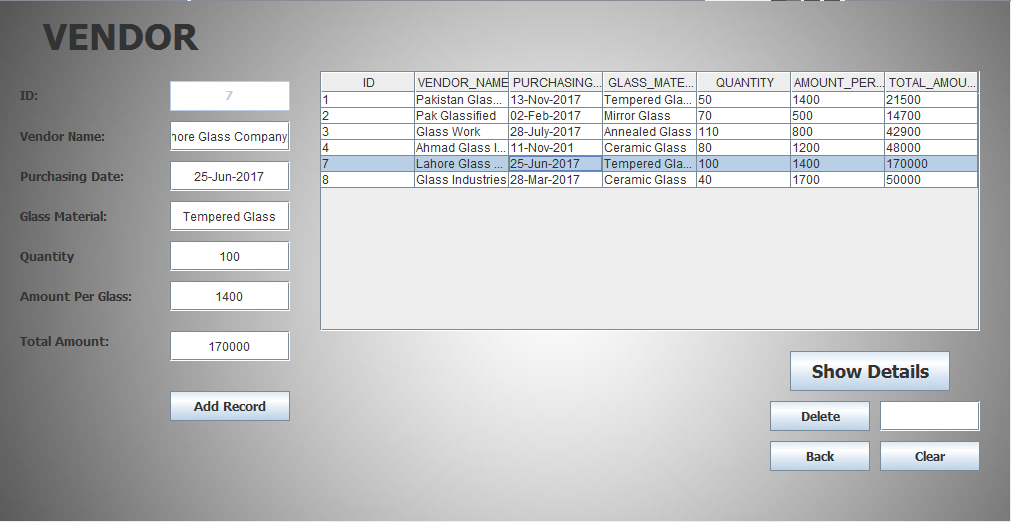
GLASS SHAPE FORM:

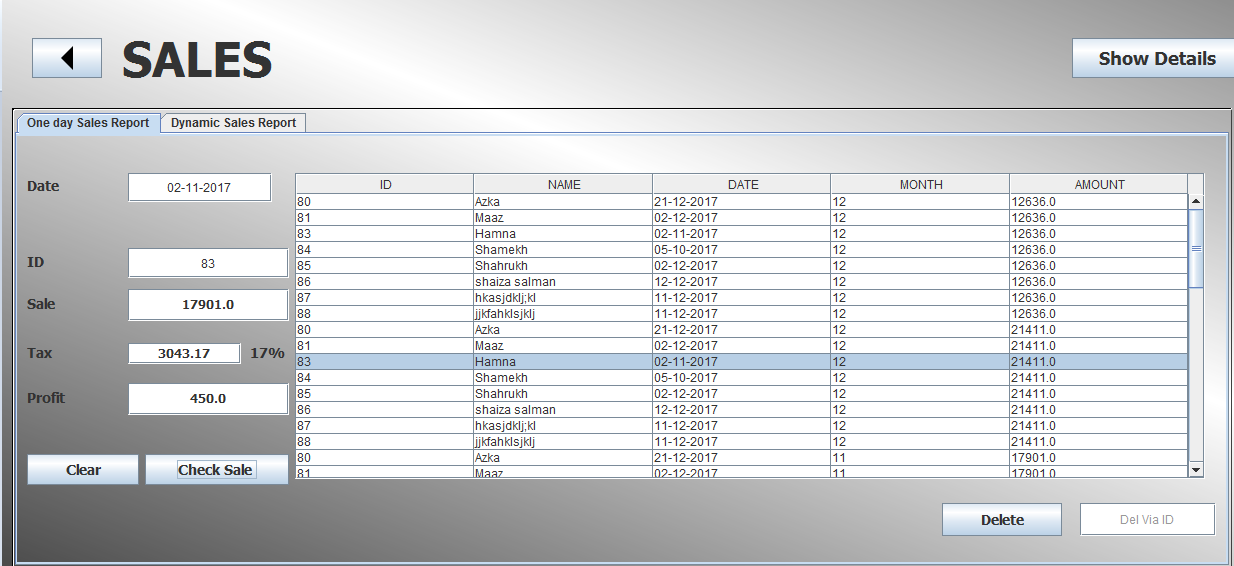
DIMENSIONS FORM:

GLASS THICKNESS FORM:PAYMENT RECEIPT FORM:



ADMIN PANEL FORM:

VENDOR FORM:

SALES FORM:

ORDER DETAILS FORM: