



MEDICAL LAB AUTOMATION SYSTEM

TEST SUITE RESULTS

By

Naresh R (13CS30024)
Barnopriyo Barua (13CS30009)
Software Engineering
Assignment

TABLE OF CONTENTS

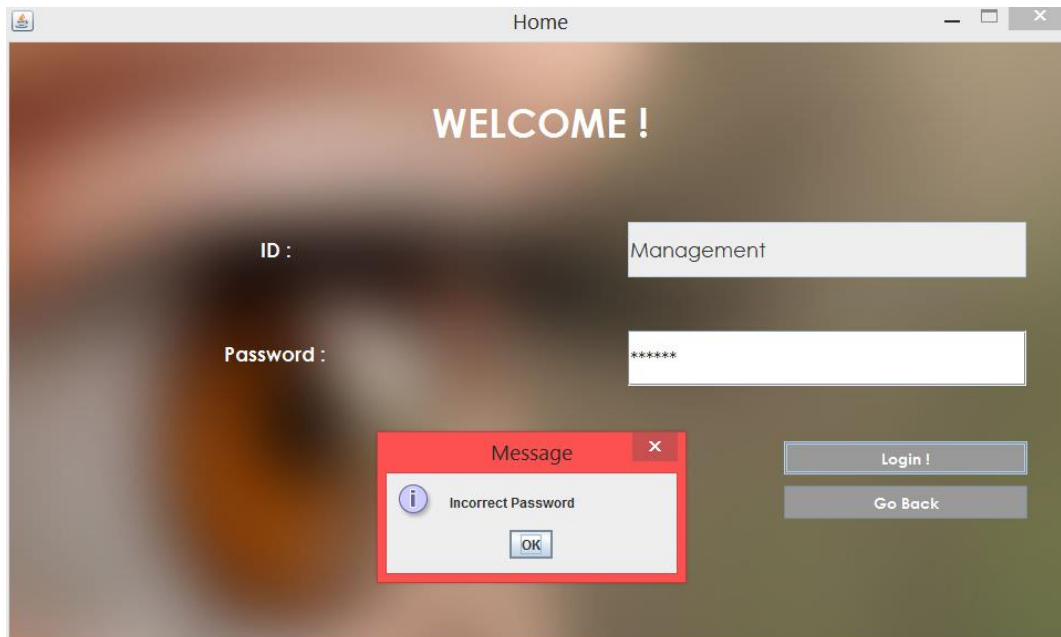
Integration Testing Results (Black Box Testing)	3
Login Page	3
Login Error for Management	3
Login error for unregistered user	3
Invalid Patient Name during Registration	4
Management Page	5
Test to be added already exists	5
No input while editing Tests	5
Test Charges containing non-digits are given as input	6
Adding an already existing Stock	6
Generation of Test Report	7
No Notifications Check	8
Patient Page	8
Test Report Collection earlier than specified Date	8
Unit Testing (White Box Testing)	9
Management_Page	9
Adding a Test	9
Removing a Test	11
Edit Test	13
Adding a Stock Item	14
Add Patient	16
Generation Of Bill	17
Generate Test Report	20
Low Stock Notification	22
Patient Page	23

Take Test	23
Collect Bill	24
Collect Test Report	26

INTEGRATION TESTING RESULTS (BLACK BOX TESTING)

LOGIN PAGE

LOGIN ERROR FOR MANAGEMENT



LOGIN ERROR FOR UNREGISTERED USER

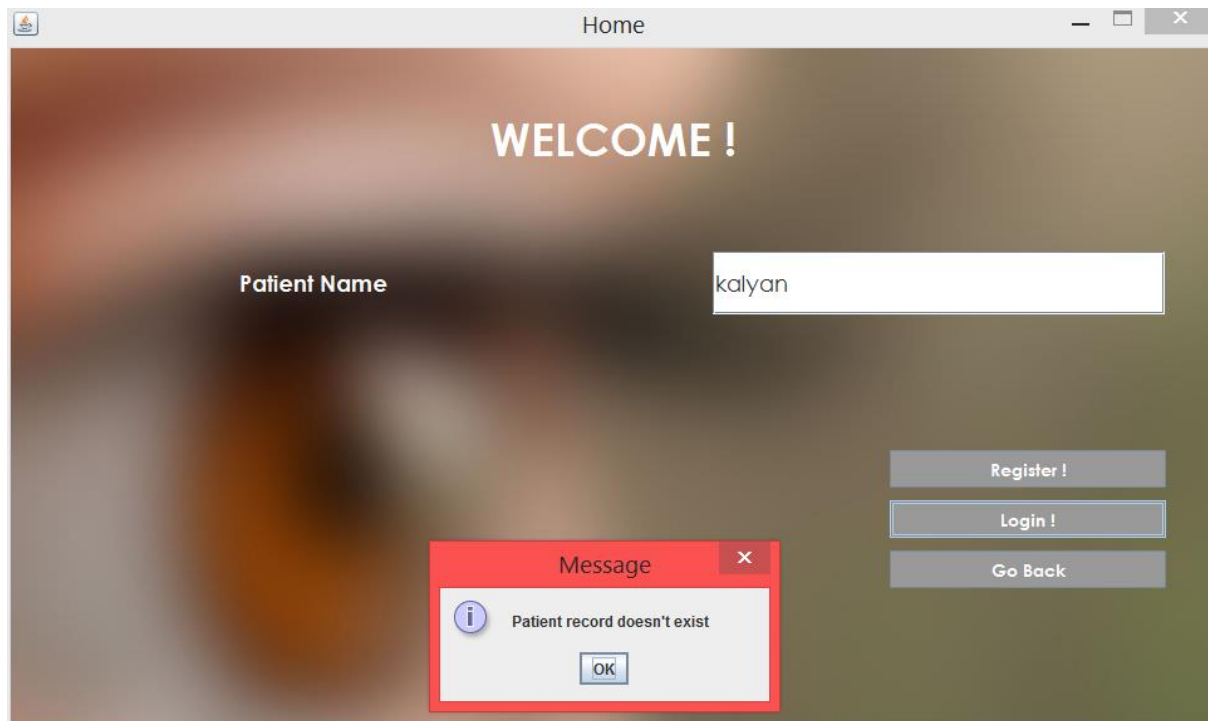
Database before login attempt

The screenshot shows a SQL query result in a web browser. The query is 'select * from patients;'. The result is a table with 4 rows and 3 columns: '#', 'id', and 'pname'. The table is titled 'select * from patients'. The data is as follows:

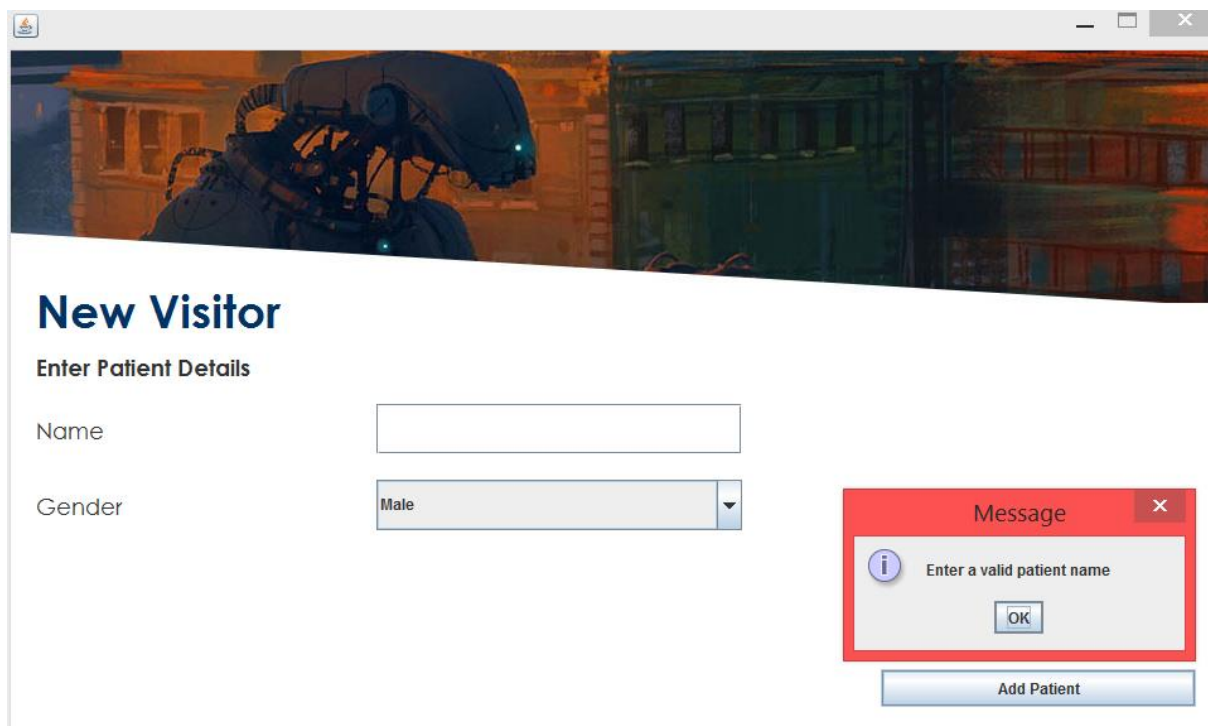
#	id	pname
1	1	naresh
2	2	pryanka
3	3	barno
4	4	vikas

The table has a 'Matching Rows' column on the right, which shows the count of rows for each row. The counts are 0, 1, 0, and 0 respectively.

Error Displayed



INVALID PATIENT NAME DURING REGISTRATION



MANAGEMENT PAGE

TEST TO BE ADDED ALREADY EXISTS

The screenshot shows the 'Add Test' form. At the top is a header image of a child with a balloon. Below it, the title 'Add Test' is centered. The form has two input fields: 'Test Name' with the value 'pressure' and 'Charges' with the value '350'. Below these is a table titled 'Stocks Required' with columns 'Item' and 'Quantity'. The table contains three rows: 'syringes' with quantity 0, 'cotton' with quantity 5, and 'medicine' with quantity 0. At the bottom of the form are two buttons: 'Add' and 'Go Back'. A modal message box is open on the right, displaying an information icon, the text 'Sorry the test already exists!!', and an 'OK' button.

Item	Quantity
syringes	0
cotton	5
medicine	0

NO INPUT WHILE EDITING TESTS

The screenshot shows the 'Edit Test' form. At the top is the same header image as the previous form. Below it, the title 'Edit Test' is centered. The form has a 'Select Test' dropdown menu with 'pressure' selected and an empty 'Charges' input field. Below these is a table titled 'Stocks Required' with columns 'Item' and 'Quantity'. The table contains three rows: 'syringes' with quantity 30, 'cotton' with quantity 40, and 'medicine' with quantity 25. At the bottom of the form are two buttons: 'Update' and 'Go Back'. A modal message box is open on the right, displaying an information icon, the text 'Enter all the details and then press Update.', and an 'OK' button.

Item	Quantity
syringes	30
cotton	40
medicine	25

TEST CHARGES CONTAINING NON-DIGITS ARE GIVEN AS INPUT

The screenshot shows a web application titled "Edit Test". It features a header image of a child with a balloon. Below the header, there is a "Select Test" dropdown menu with "pressure" selected, and a "Charges" text input field containing "haha". A table titled "Stocks Required" lists items and their quantities: syringes (30), cotton (40), and medicine (25). Below the table are "Update" and "Go Back" buttons. A red error message box is displayed, stating "Charges can contain only numbers(0 to 9)!" with an "OK" button.

Edit Test

Select Test: pressure Charges: haha

Stocks Required

Item	Quantity
syringes	30
cotton	40
medicine	25

Update Go Back

Message
Charges can contain only numbers(0 to 9)!

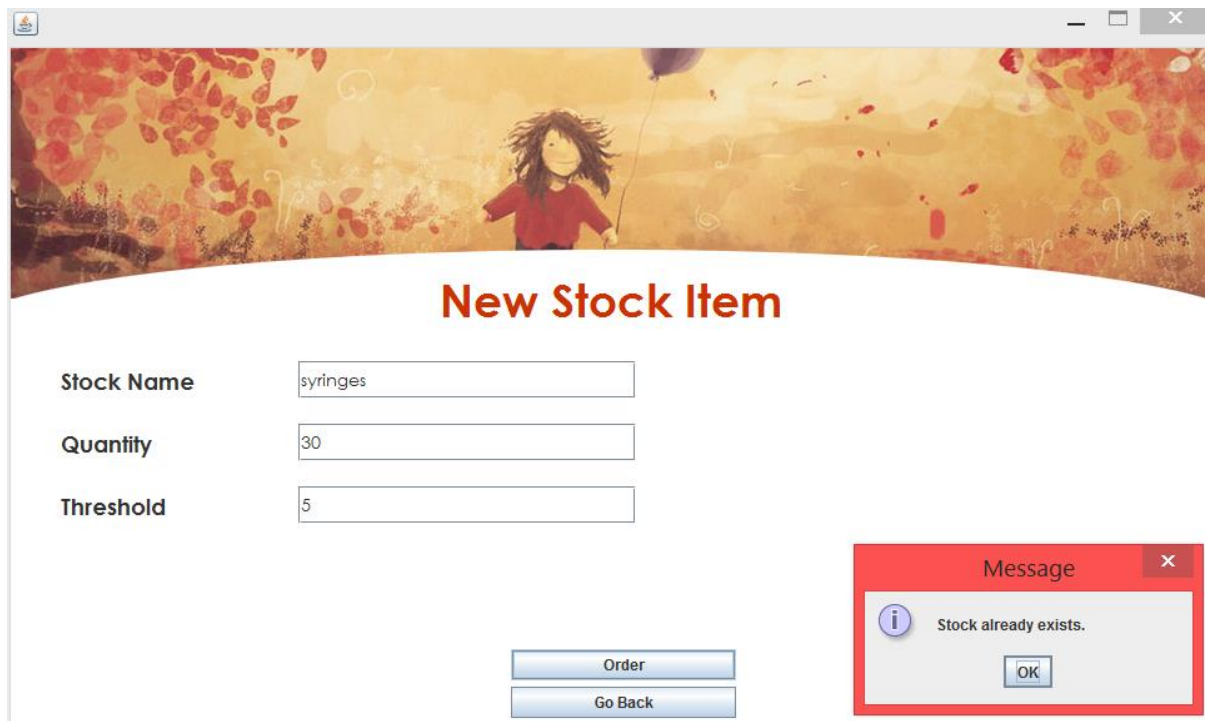
ADDING AN ALREADY EXISTING STOCK

Stocks Table before Adding

The screenshot shows a SQL query result in a web application. The query is "select * from stocks;". The result is displayed in a table with columns: #, id, Stock_Name, Quantity, and Threshold. The table contains three rows of data.

#	id	Stock_Name	Quantity	Threshold
1	1	syringes	30	3
2	2	cotton	40	5
3	3	medicine	25	5

Error Message Displayed



The screenshot shows a web application window titled "New Stock Item". The form contains three input fields: "Stock Name" with the value "syringes", "Quantity" with the value "30", and "Threshold" with the value "5". Below the form are two buttons: "Order" and "Go Back". A red error message box is displayed on the right side of the form, containing the text "Message" and "Stock already exists." with an "OK" button.

New Stock Item

Stock Name:

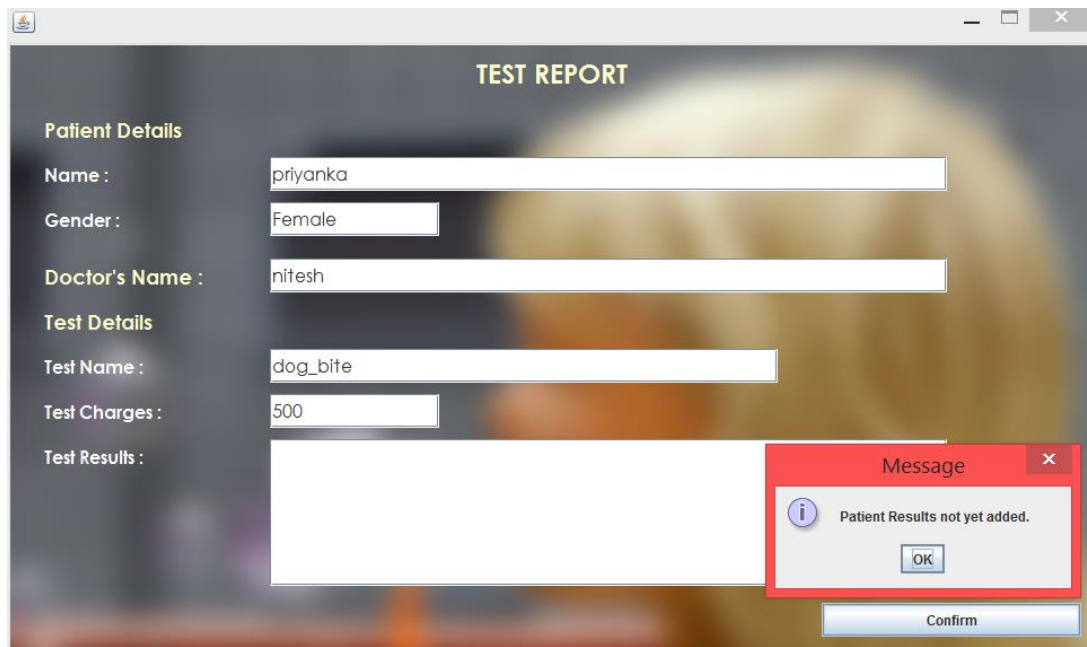
Quantity:

Threshold:

Message
Stock already exists.

GENERATION OF TEST REPORT

Patient Results not added



The screenshot shows a web application window titled "TEST REPORT". The form is divided into two sections: "Patient Details" and "Test Details". The "Patient Details" section includes fields for "Name" (priyanka), "Gender" (Female), and "Doctor's Name" (nitesh). The "Test Details" section includes fields for "Test Name" (dog_bite), "Test Charges" (500), and "Test Results" (empty). A red error message box is displayed on the right side of the form, containing the text "Message" and "Patient Results not yet added." with an "OK" button. A "Confirm" button is located at the bottom right of the form.

TEST REPORT

Patient Details

Name:

Gender:

Doctor's Name:

Test Details

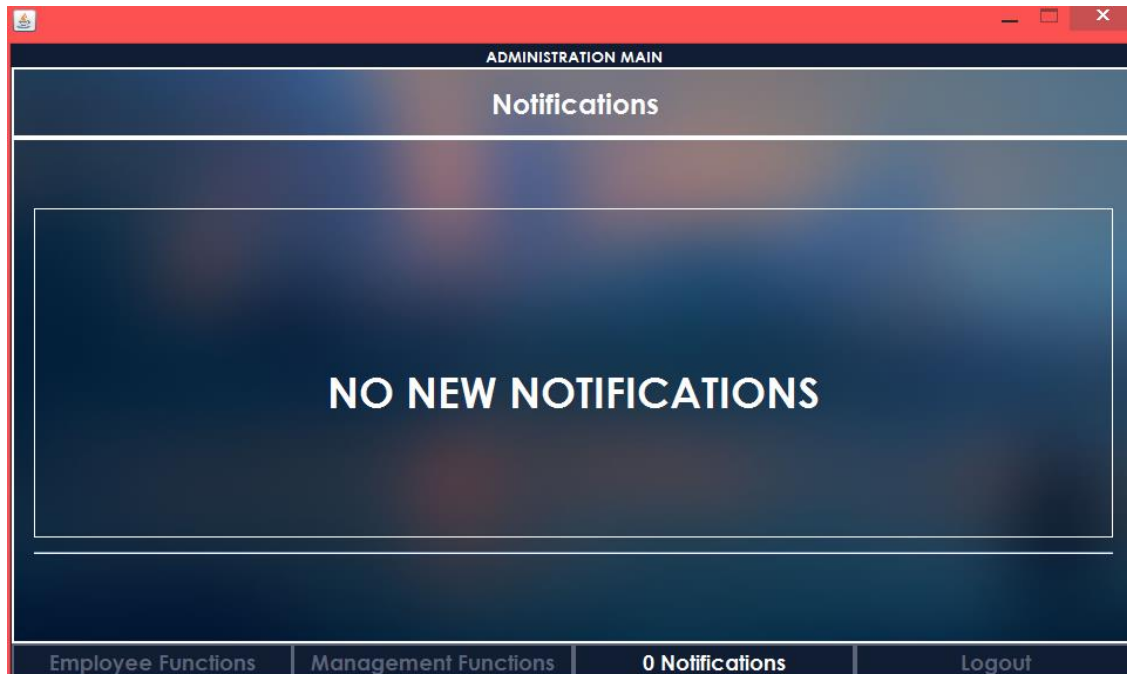
Test Name:

Test Charges:

Test Results:

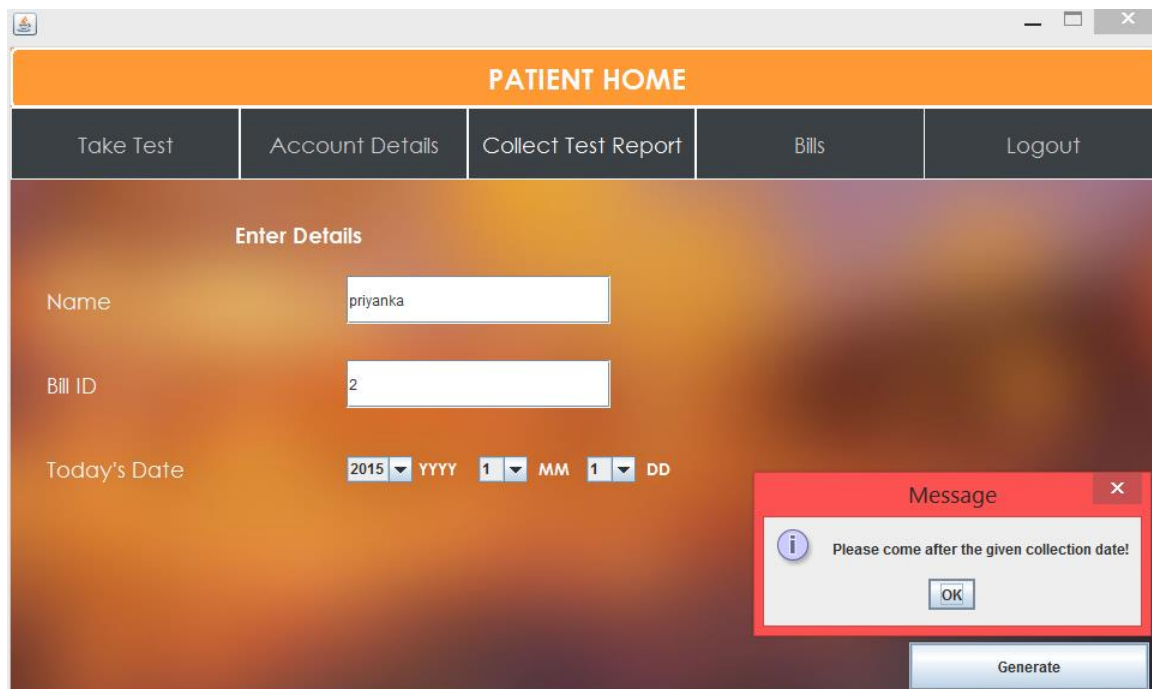
Message
Patient Results not yet added.

NO NOTIFICATIONS CHECK



PATIENT PAGE

TEST REPORT COLLECTION EARLIER THAN SPECIFIED DATE

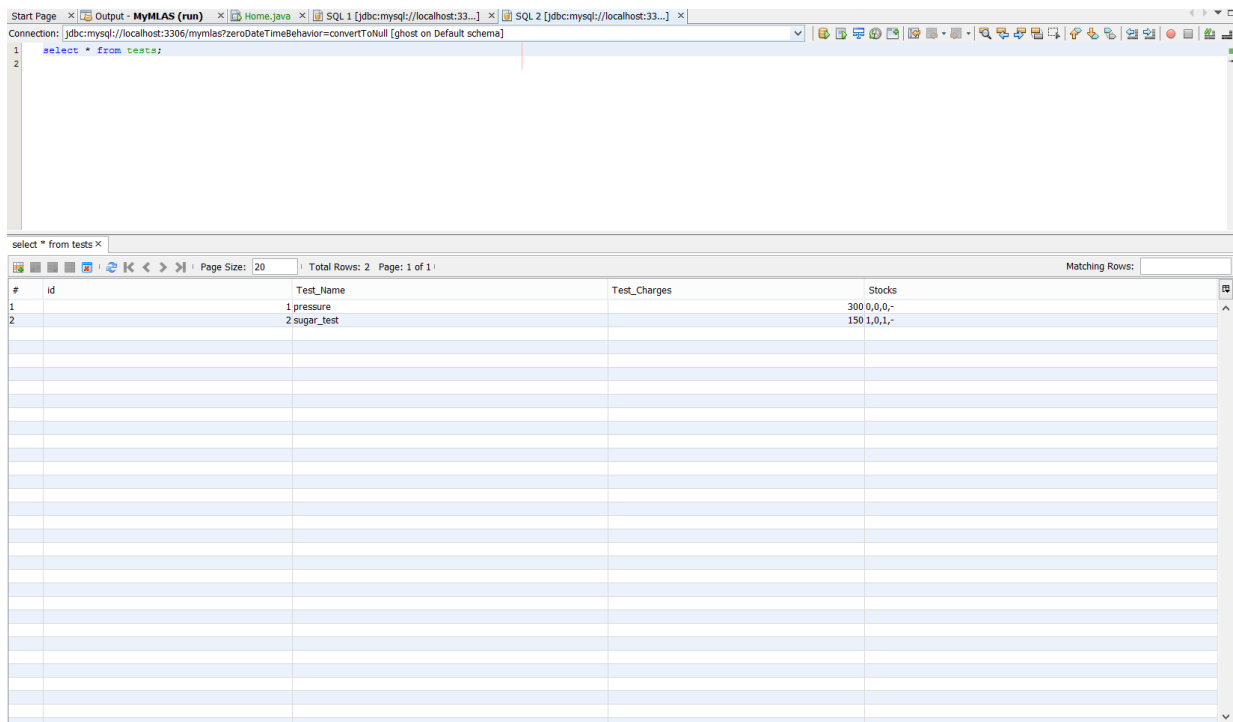


UNIT TESTING (WHITE BOX TESTING)

MANAGEMENT_PAGE

ADDING A TEST

Database before Adding the test



The screenshot shows a SQL IDE interface. The top window displays the query `select * from tests;`. The bottom window shows the results of this query as a table with 5 columns: #, id, Test_Name, Test_Charges, and Stocks. There are 2 rows of data.

#	id	Test_Name	Test_Charges	Stocks
1		1.pressure		300.0,0,0,-
2		2.sugar_test		150.1,0,1,-

Success Result

Add Test

Test Name: Charges:

Stocks Required

Item	Quantity
syringes	3
cotton	1
medicine	1

Message
The test has been successfully added!!

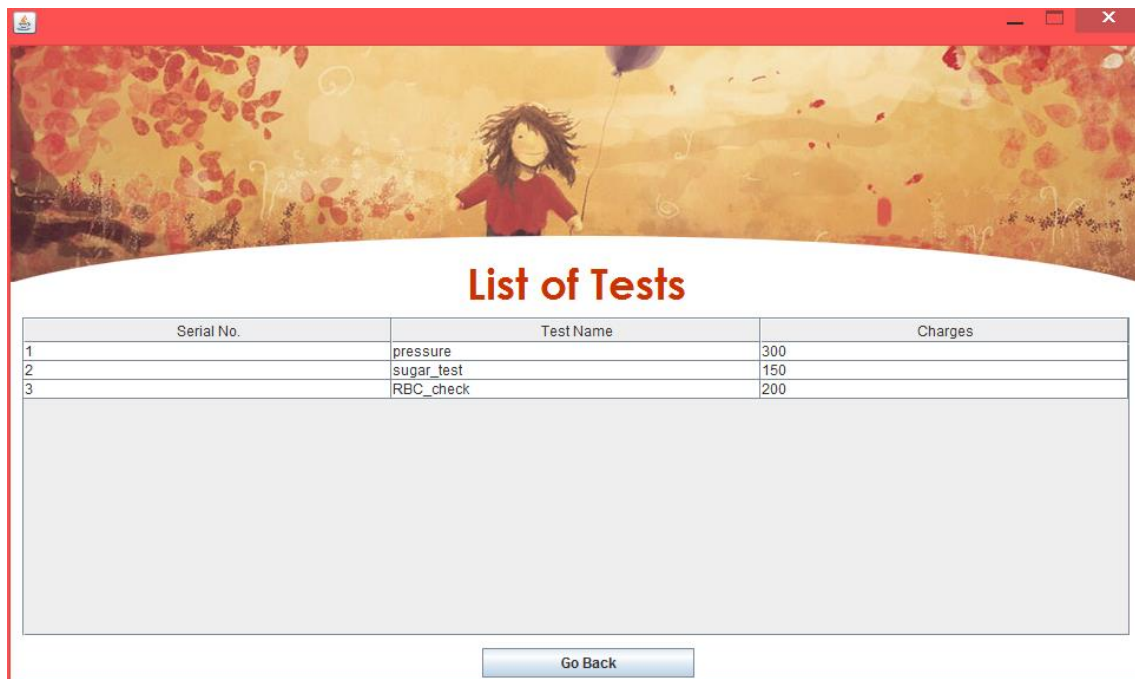
Database After adding Test

Connection: jdbc:mysql://localhost:3306/mymlas?zeroDateTimeBehavior=convertToNull [ghost on Default schema]

```
1 select * from tests;
```

#	id	Test_Name	Test_Charges	Stocks
1	1	pressure	300.0,0.0,-	
2	2	sugar_test	150.1,0.1,-	
3	3	RBC_check	200.3,1.1,-	

Tests listed after Adding

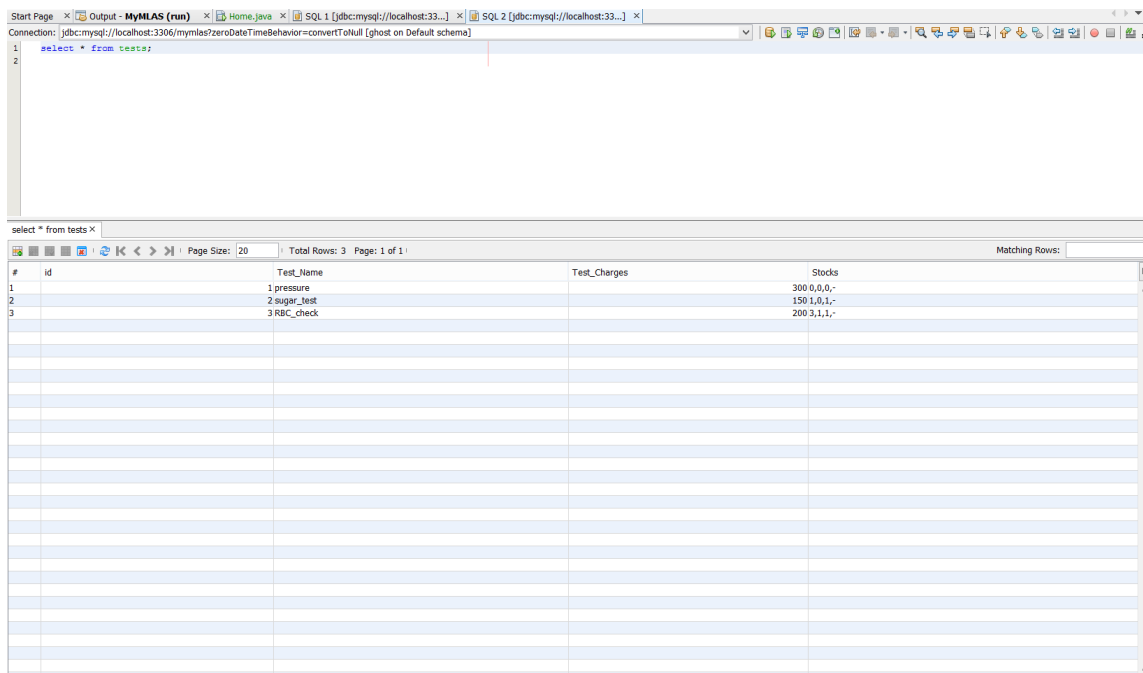


Serial No.	Test Name	Charges
1	pressure	300
2	sugar_test	150
3	RBC_check	200

Go Back

REMOVING A TEST

Database of Tests before removing



#	id	Test_Name	Test_Charges	Stocks
1	1	pressure	300	0.0.0.
2	2	sugar_test	150	1.0.1.
3	3	RBC_check	200	3.1.1.

Remove Test (Success)

Remove Test

Available Tests

Test Name
pressure
sugar_test
RBC_check

Select Test to Remove

sugar_test

Confirm

Go Back

Message

Done.

OK

Database after removing a test

Start Page x Output - MyMLAS (run) x Home.java x SQL 1 [jdbc:mysql://localhost:3306/] x SQL 2 [jdbc:mysql://localhost:3306/] x

Connection: jdbc:mysql://localhost:3306/mymlas?zeroDateTimeBehavior=convertToNull [ghost on Default schema]

```
1 select * from tests;
```

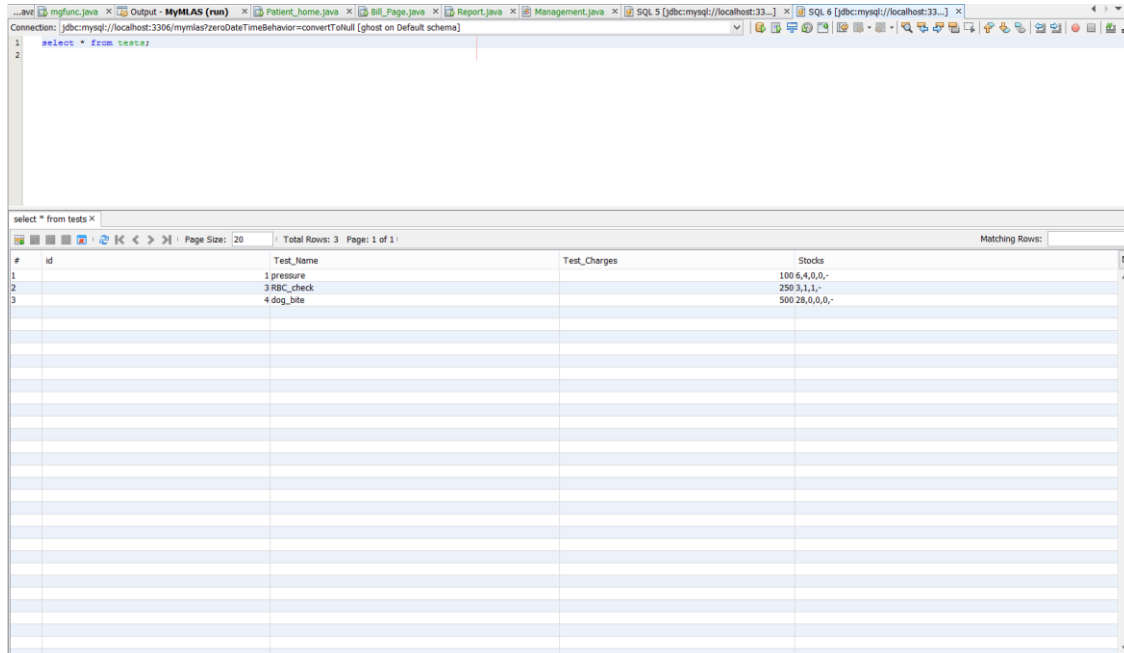
select * from tests x

Page Size: 20 Total Rows: 2 Page: 1 of 1 Matching Rows:

#	id	Test_Name	Test_Charges	Stocks
1	1	pressure		300.0,0,-
2	3	RBC_check		200.3,1,1,-

EDIT TEST

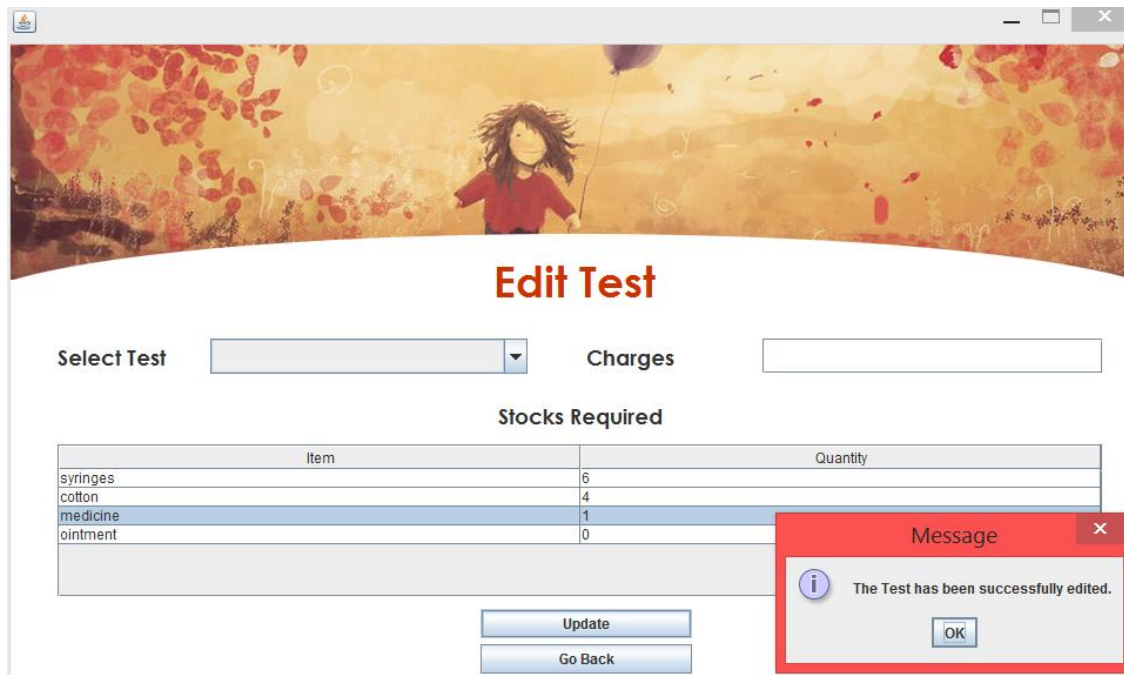
Database before Editing Test



The screenshot shows a MySQL database interface with a table named 'tests'. The table has four columns: 'id', 'Test_Name', 'Test_Charges', and 'Stocks'. There are three rows of data.

#	id	Test_Name	Test_Charges	Stocks
1		1 pressure		100 6,4,0,0,-
2		3 RBC_check		250 3,1,1,-
3		4 dog_bite		500 28,0,0,0,-

Edit Test (Success)



The screenshot shows the 'Edit Test' web application interface. It features a header with a child holding a balloon. Below the header, there is a form with a 'Select Test' dropdown menu, a 'Charges' input field, and a 'Stocks Required' table. The table has two columns: 'Item' and 'Quantity'. The items listed are 'syringes', 'cotton', 'medicine', and 'ointment'. The quantities are 6, 4, 1, and 0 respectively. At the bottom of the form, there are 'Update' and 'Go Back' buttons. A red message box is displayed on the right side of the form, indicating that the test has been successfully edited.

Edit Test

Select Test: Charges:

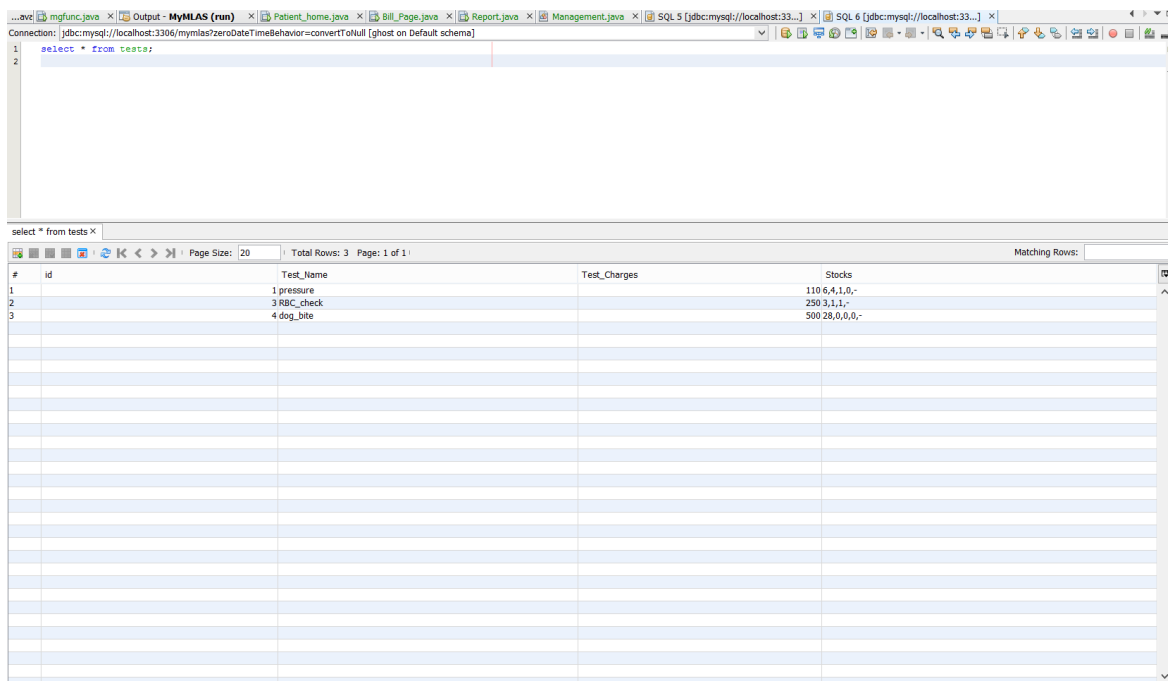
Stocks Required

Item	Quantity
syringes	6
cotton	4
medicine	1
ointment	0

Message

The Test has been successfully edited.

Database after Editing of Test

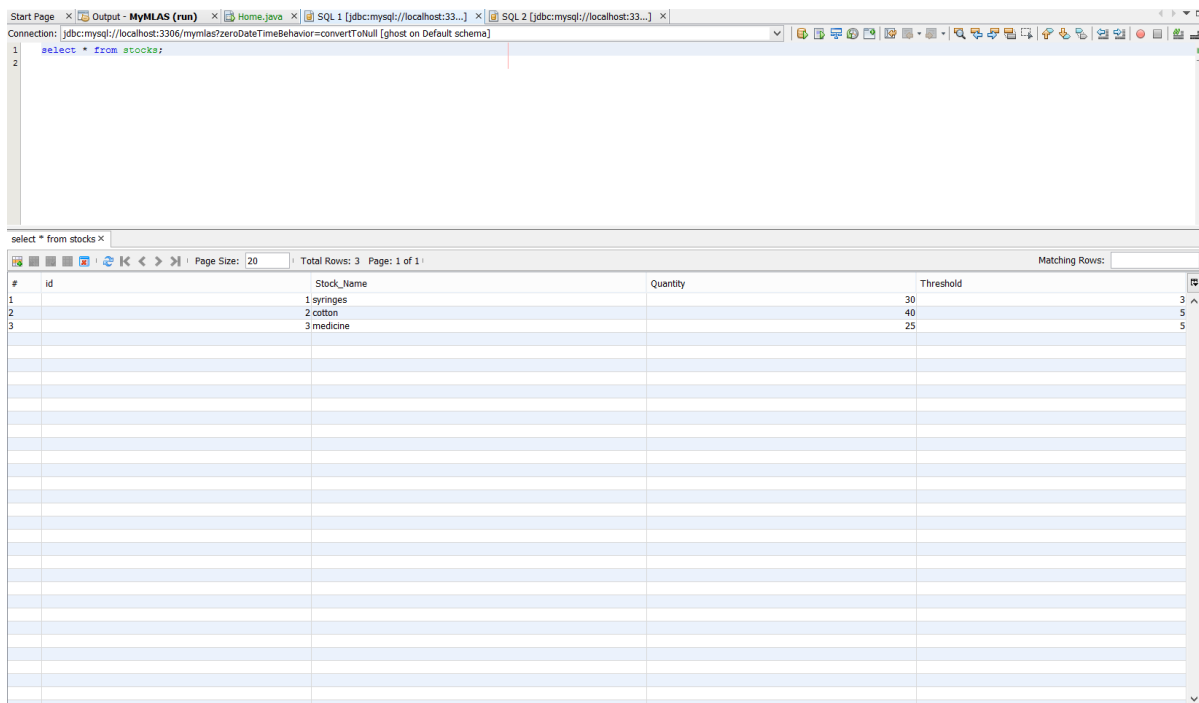


The screenshot shows a Java IDE with several tabs open. The active tab displays a SQL query: `select * from tests;`. Below the query editor, a table titled "select * from tests" shows the results of the query. The table has five columns: #, id, Test_Name, Test_Charges, and Stocks. There are three rows of data.

#	id	Test_Name	Test_Charges	Stocks
1		1 pressure		110 6,4,1,0,-
2		3 RBC_check		250 3,1,1,-
3		4 dog_bite		500 28,0,0,0,-

ADDING A STOCK ITEM

Database (Stocks) before Adding



The screenshot shows a Java IDE with several tabs open. The active tab displays a SQL query: `select * from stocks;`. Below the query editor, a table titled "select * from stocks" shows the results of the query. The table has five columns: #, id, Stock_Name, Quantity, and Threshold. There are three rows of data.

#	id	Stock_Name	Quantity	Threshold
1		1 syringes		30
2		2 cotton		40
3		3 medicine		25

Add Stock (Success)

New Stock Item

Stock Name:

Quantity:

Threshold:

Message
Done.

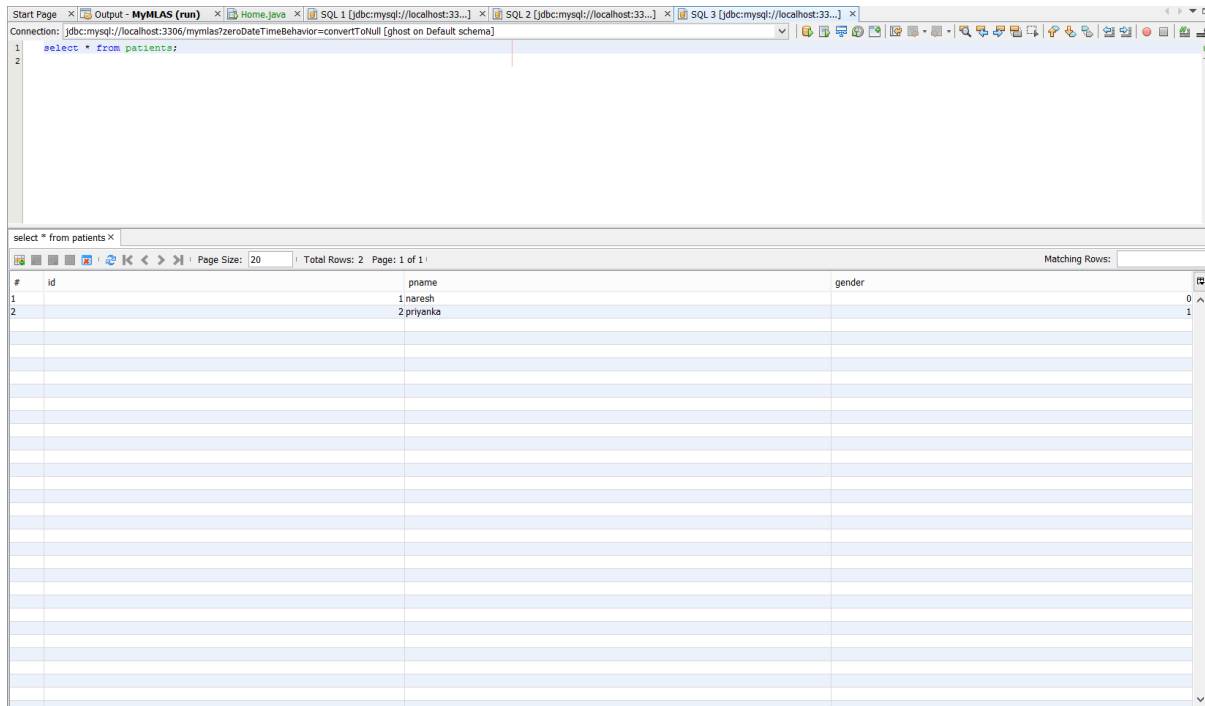
Database after Stock Added

```
select * from stocks;
```

#	id	Stock_Name	Quantity	Threshold
1		1 syringes	30	3
2		2 cotton	40	5
3		3 medicine	25	5
4		5 ointment	60	10

ADD PATIENT

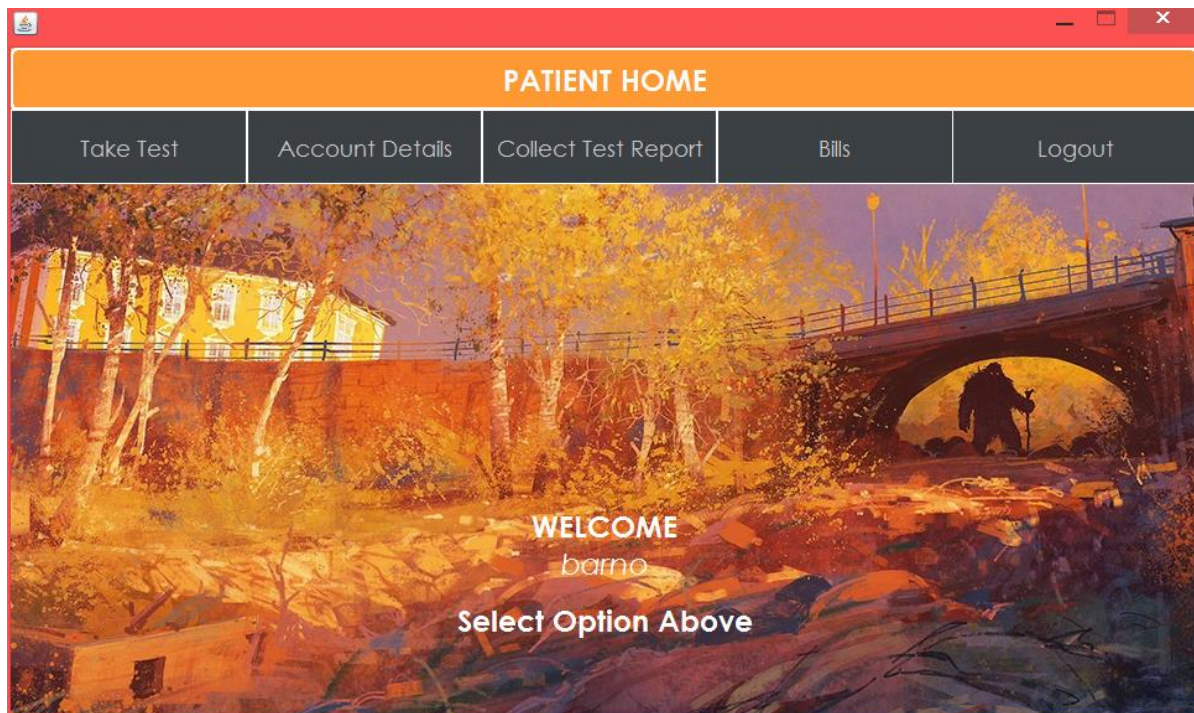
Database before addition of Patient



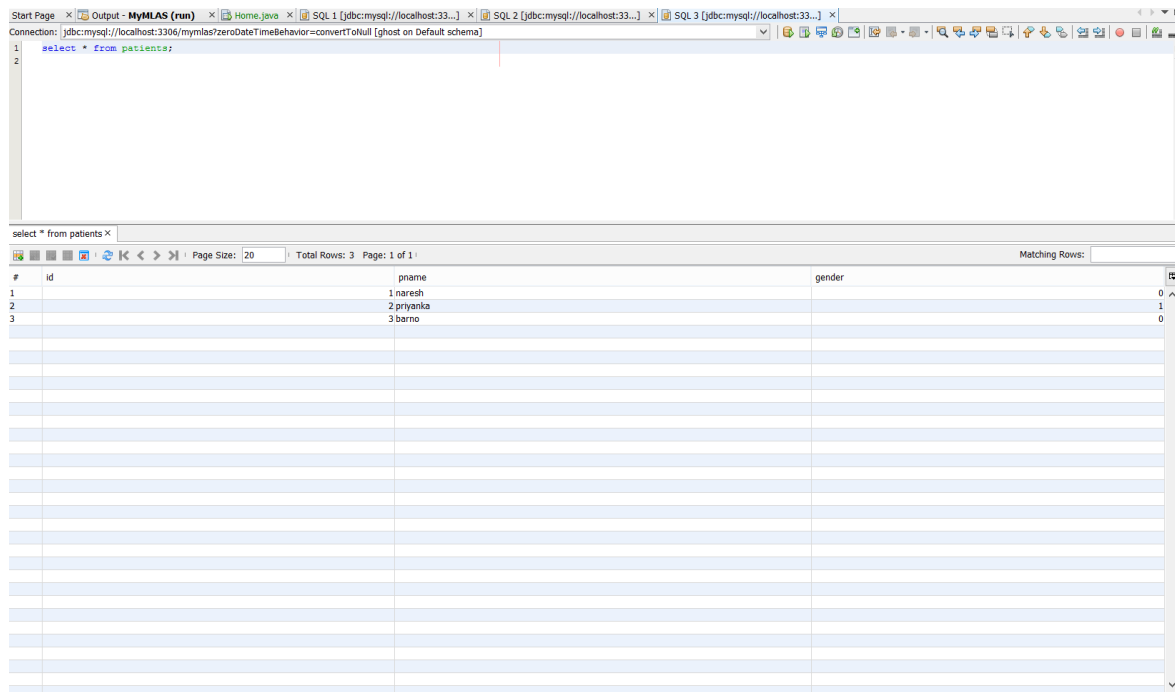
The screenshot shows a MySQL database interface with a query window and a results table. The query window contains the SQL statement: `select * from patients;`. The results table displays two rows of data:

#	id	pname	gender
1	1	naresh	0
2	2	priyanka	1

Add Patient (Success)



Database after addition of Patient

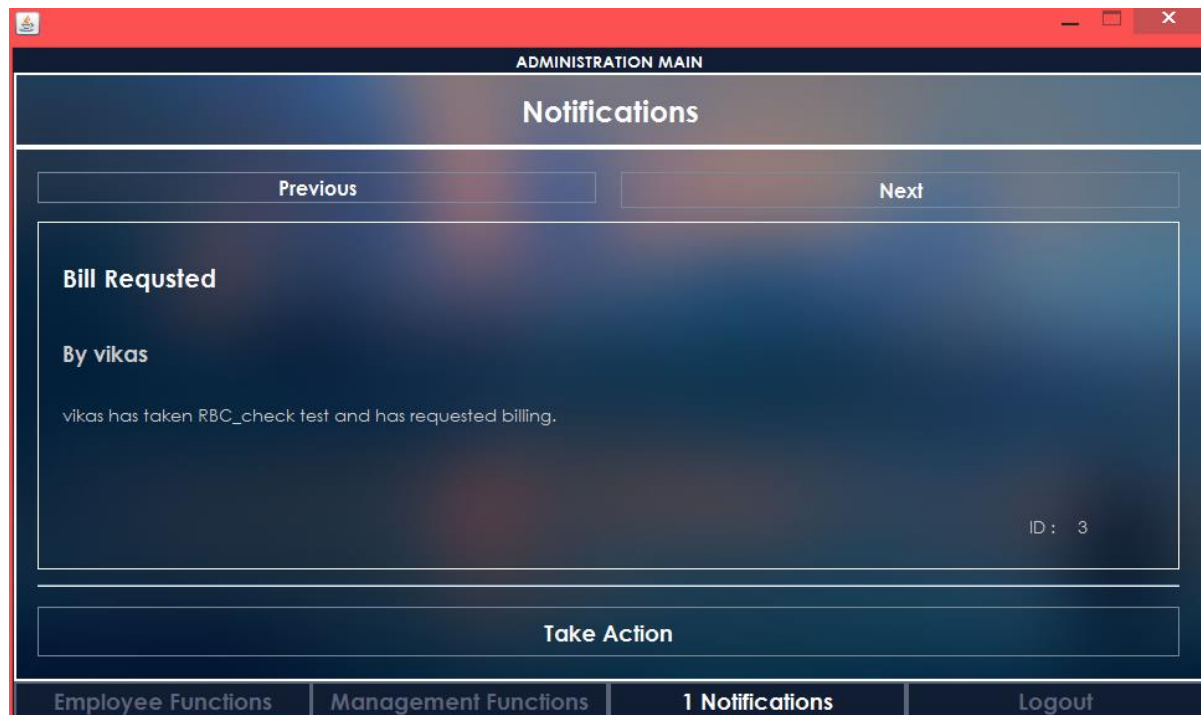


The screenshot shows a MySQL IDE window with a query editor at the top containing the SQL statement `select * from patients;`. Below the editor, the results pane displays a table with 3 rows and 4 columns: `#`, `id`, `pname`, and `gender`. The data rows are as follows:

#	id	pname	gender
1	1	naresh	0
2	2	priyanka	1
3	3	barno	0

GENERATION OF BILL

Notification for Generation of Bill



The screenshot displays the 'ADMINISTRATION MAIN' interface. At the top, there is a 'Notifications' section with 'Previous' and 'Next' buttons. The notification content is as follows:

Bill Requested

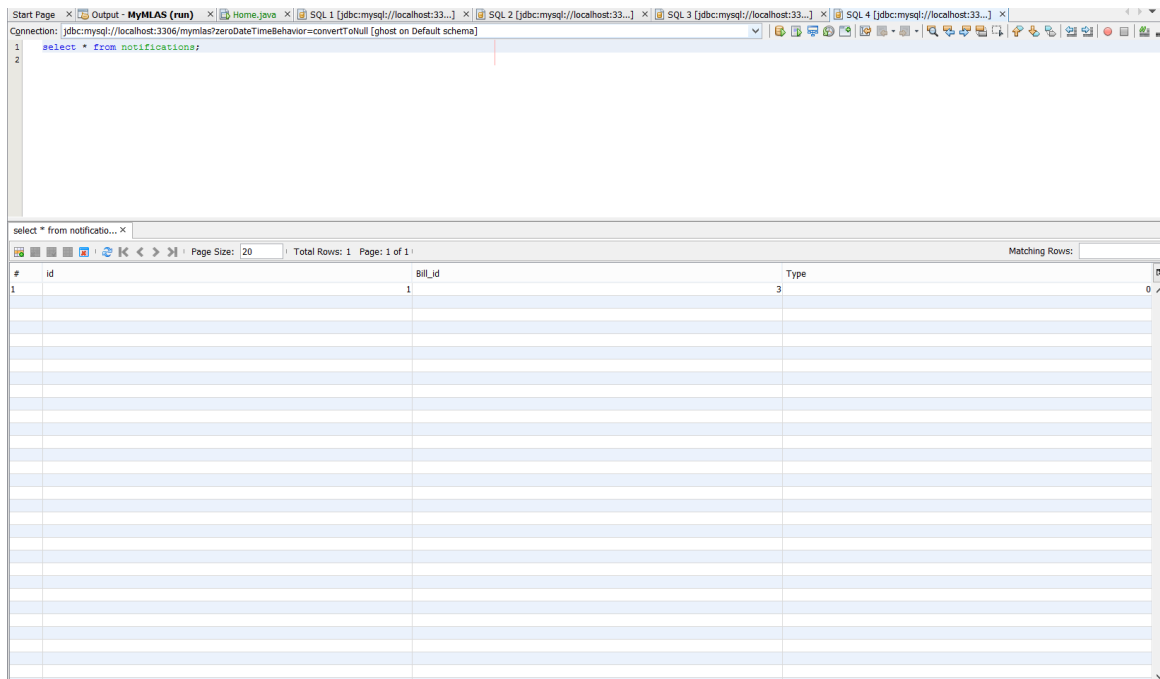
By vikas

vikas has taken RBC_check test and has requested billing.

ID : 3

At the bottom of the notification area is a 'Take Action' button. The footer of the application shows four tabs: 'Employee Functions', 'Management Functions', '1 Notifications', and 'Logout'.

Database before Bill Generation



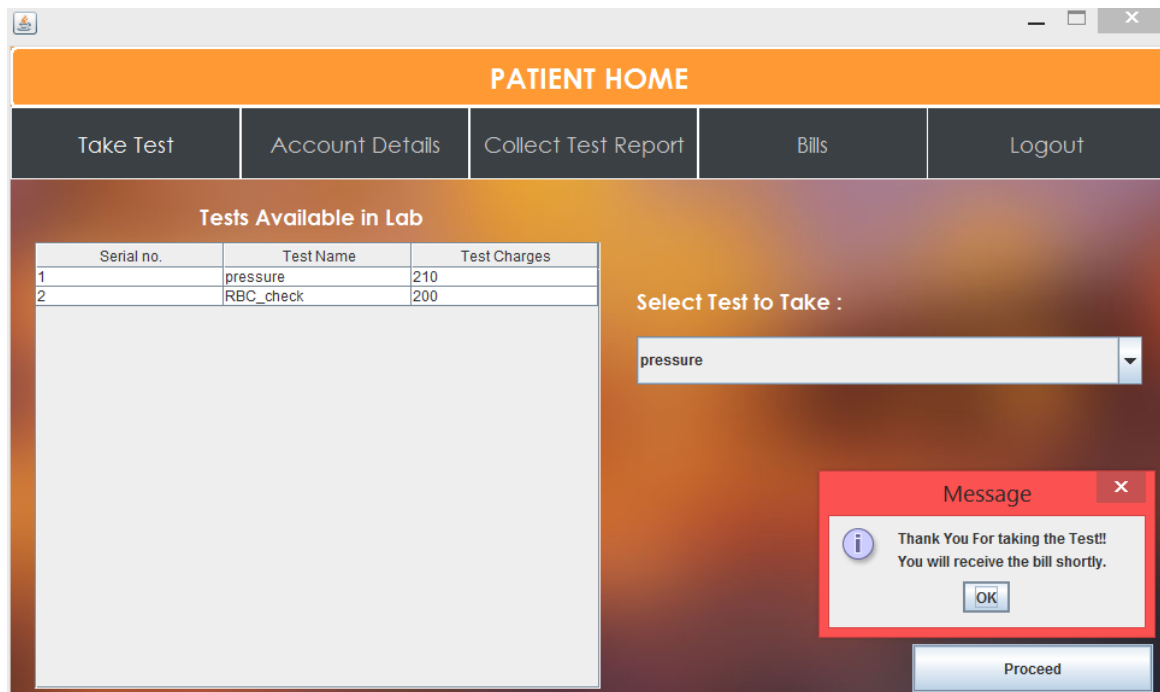
The screenshot shows a SQL IDE with a query window and a results window. The query window contains the following SQL statement:

```
1 select * from notifications;
2
```

The results window displays the output of the query, showing a table with 4 columns: #, id, Bill_id, and Type. The table contains 1 row of data.

#	id	Bill_id	Type
1		1	3

Generate Bill



The screenshot shows the 'PATIENT HOME' interface. The top navigation bar includes links for 'Take Test', 'Account Details', 'Collect Test Report', 'Bills', and 'Logout'. The main content area is titled 'Tests Available in Lab' and displays a table with 3 columns: Serial no., Test Name, and Test Charges.

Serial no.	Test Name	Test Charges
1	pressure	210
2	RBC_check	200

Below the table, there is a 'Select Test to Take :' dropdown menu with 'pressure' selected. A message box is displayed with the text: 'Thank You For taking the Test!! You will receive the bill shortly.' and an 'OK' button. A 'Proceed' button is also visible at the bottom right.

Bill Generation (Success)

BILL

Unique Bill ID :

3

Patient's Name :

vikas

Doctor's Name :

moneish

Test Details

Test Name : RBC_check

Charges : 200

Date of Test :

2015

YYYY

10

MM

2

DD

Date of Collection of Test Report :

2015

YYYY

10

MM

7

DD

Time of Collection of Test Report :

09

HH

00

MM

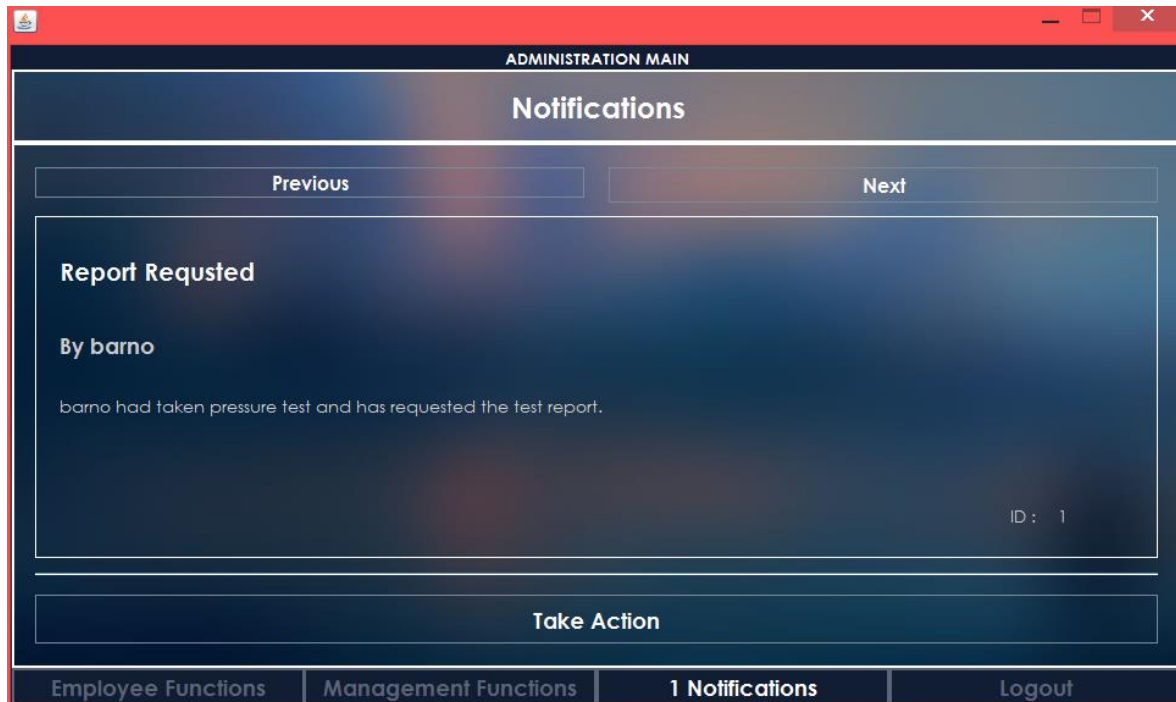
Confirm

Database after Bill Generation

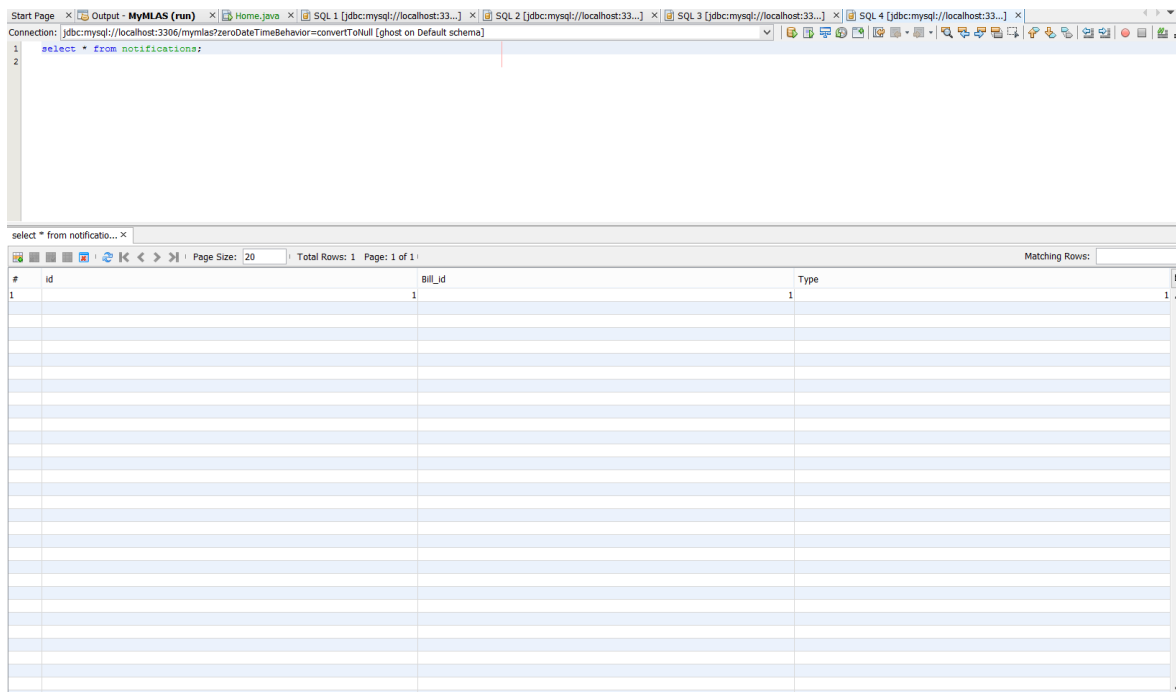
[illegible]

GENERATE TEST REPORT

Test Report notification



Database before Test Report Generation



Generate Test Report

The screenshot shows a web application window titled "PATIENT HOME". It features a navigation bar with five buttons: "Take Test", "Account Details", "Collect Test Report", "Bills", and "Logout". The main content area is titled "Enter Details" and contains three input fields: "Name" with the value "barno", "Bill ID" with the value "1", and "Today's Date" with a date picker set to "2015" for the year, "4" for the month, and "25" for the day. A "Generate" button is located at the bottom right. A red message box is overlaid on the right side, containing the text "Thank You For your time!! You will receive the report shortly." and an "OK" button.

PATIENT HOME

Take Test Account Details Collect Test Report Bills Logout

Enter Details

Name barno

Bill ID 1

Today's Date 2015 YYYY 4 MM 25 DD

Message

Thank You For your time!!
You will receive the report shortly.

OK

Generate

Test Report Generated (Success)

The screenshot shows a web application window titled "TEST REPORT". It displays the following details:

- Patient Details**
 - Name : barno
 - Gender : Male
 - Doctor's Name : vikas
- Test Details**
 - Test Name : pressure
 - Test Charges : 210
 - Test Results : Blood pressure is normal.

A "Confirm" button is located at the bottom right.

TEST REPORT

Patient Details

Name : barno

Gender : Male

Doctor's Name : vikas

Test Details

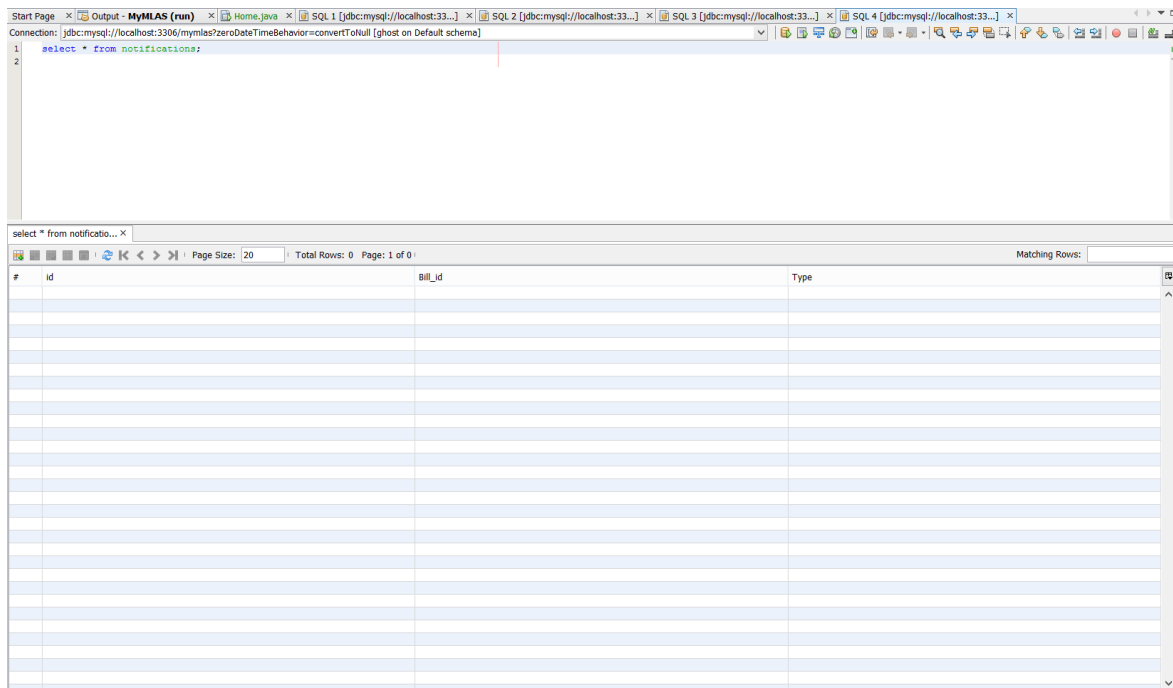
Test Name : pressure

Test Charges : 210

Test Results : Blood pressure is normal.

Confirm

Database after Test Report Generation

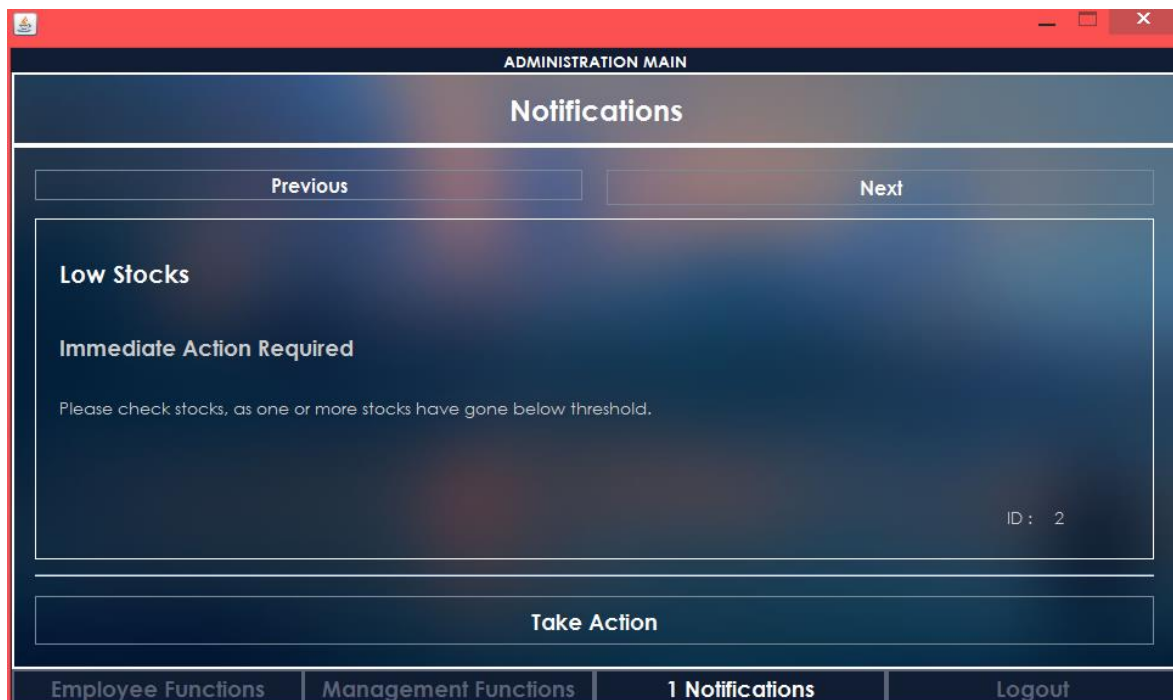


The screenshot shows a database client window with a query editor at the top containing the SQL statement: `select * from notifications;`. Below the editor, a results pane displays a table with the following structure:

#	id	Bill_id	Type
1	2		

The table has 4 columns: #, id, Bill_id, and Type. The first row of data shows id=2. The interface includes a toolbar with navigation and formatting icons, and a status bar at the bottom indicating 'Total Rows: 0' and 'Page: 1 of 0'.

LOW STOCK NOTIFICATION

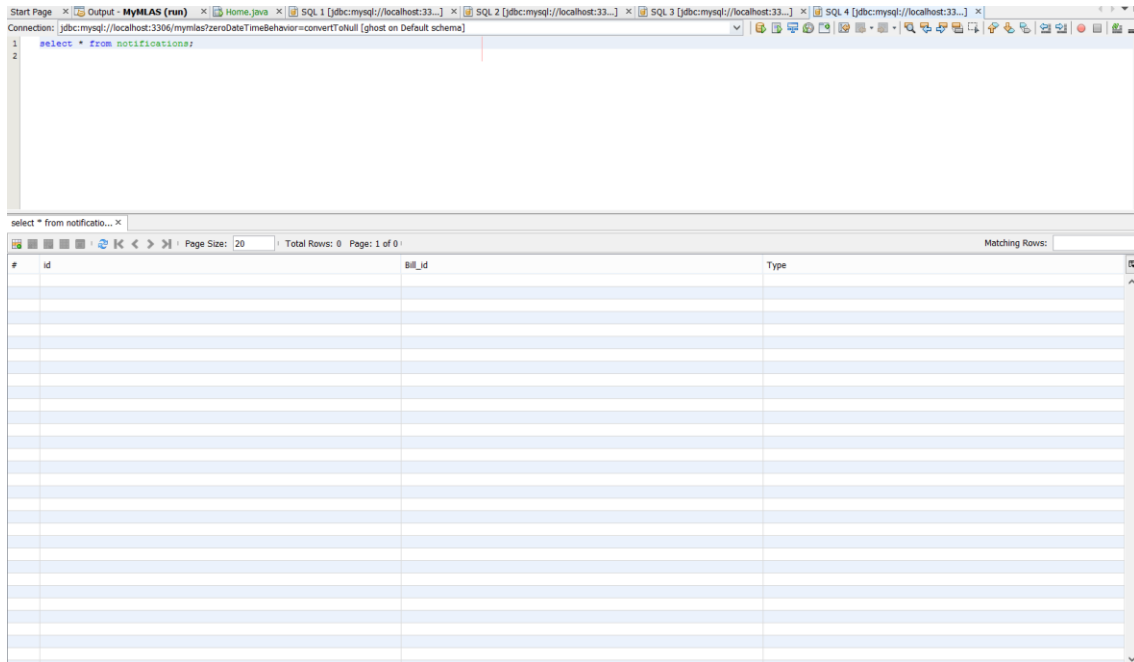


The screenshot displays a web application interface titled 'ADMINISTRATION MAIN'. A 'Notifications' section is active, showing a 'Low Stocks' alert. The notification text reads: 'Immediate Action Required' and 'Please check stocks, as one or more stocks have gone below threshold.' The notification ID is '2'. Below the notification, there is a 'Take Action' button. At the bottom of the page, a navigation bar contains links for 'Employee Functions', 'Management Functions', '1 Notifications', and 'Logout'.

PATIENT PAGE

TAKE TEST

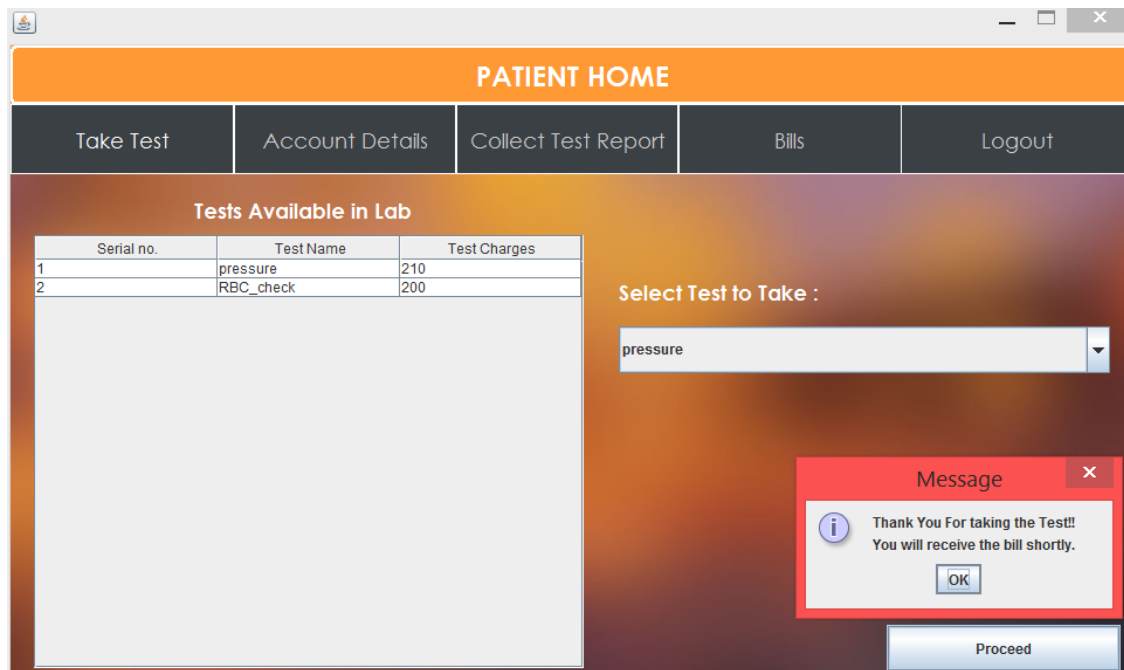
Database before Taking Test



The screenshot shows a database client interface. The top window displays a SQL query: `select * from notifications;`. The bottom window shows the results of this query as an empty table with the following columns: #, id, Bill_id, and Type. The table has 20 rows, all of which are empty.

#	id	Bill_id	Type

Take Test (Success)

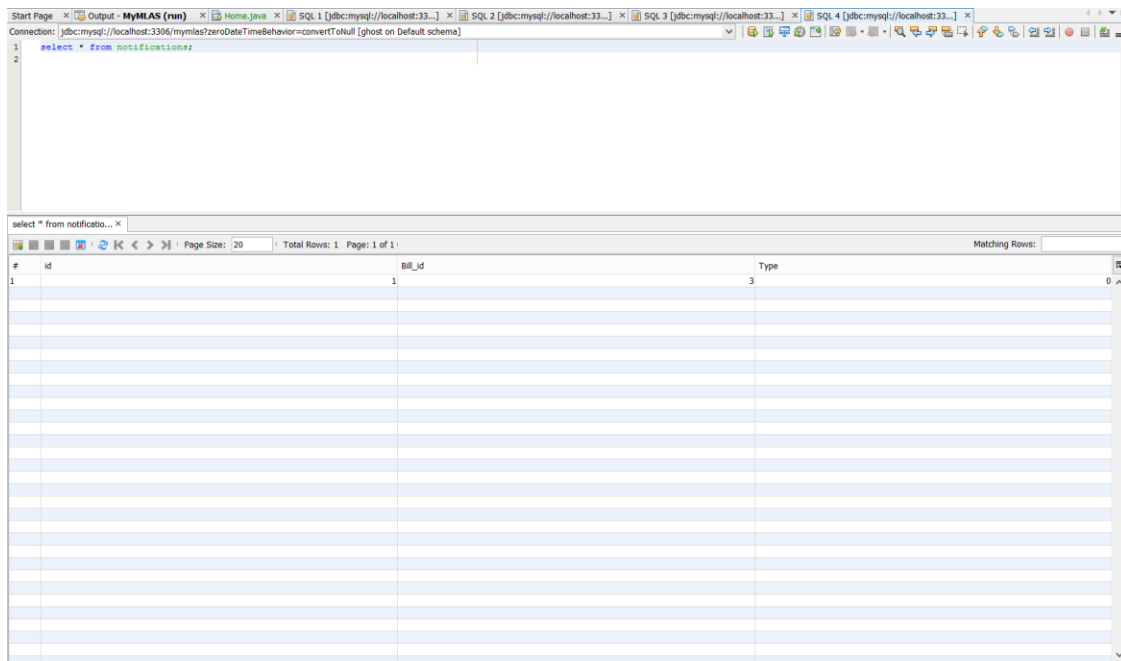


The screenshot shows the 'PATIENT HOME' interface. It has a navigation bar with buttons: 'Take Test', 'Account Details', 'Collect Test Report', 'Bills', and 'Logout'. Below the navigation bar, there is a section titled 'Tests Available in Lab' containing a table with the following data:

Serial no.	Test Name	Test Charges
1	pressure	210
2	RBC_check	200

To the right of the table, there is a label 'Select Test to Take :' followed by a dropdown menu currently showing 'pressure'. At the bottom right, there is a 'Proceed' button. A red message dialog box is open, displaying the text: 'Thank You For taking the Test!! You will receive the bill shortly.' with an 'OK' button.

Database after Taking Test

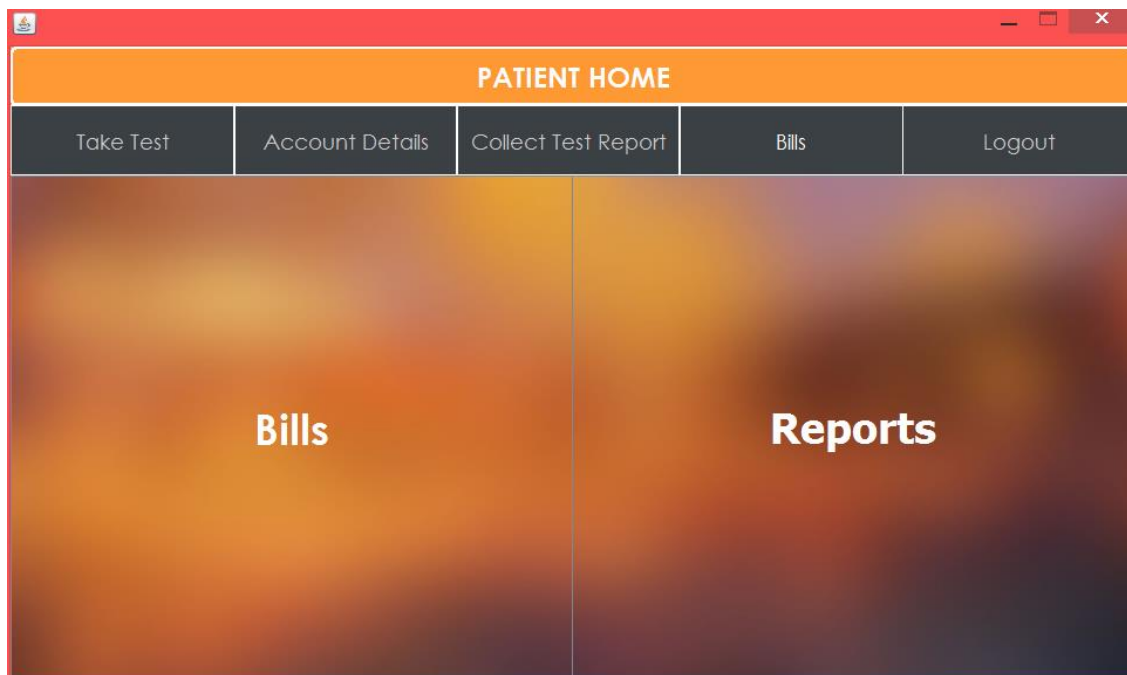


The screenshot shows a MySQL IDE window with a query editor at the top containing the SQL statement `select * from notification;`. Below the editor, the results of the query are displayed in a table. The table has three columns: `#`, `id`, and `Type`. The first row contains the values `1`, `1`, and `3` respectively. The table is titled `select * from notificatio...` and shows `Page Size: 20`, `Total Rows: 1`, and `Page: 1 of 1`.

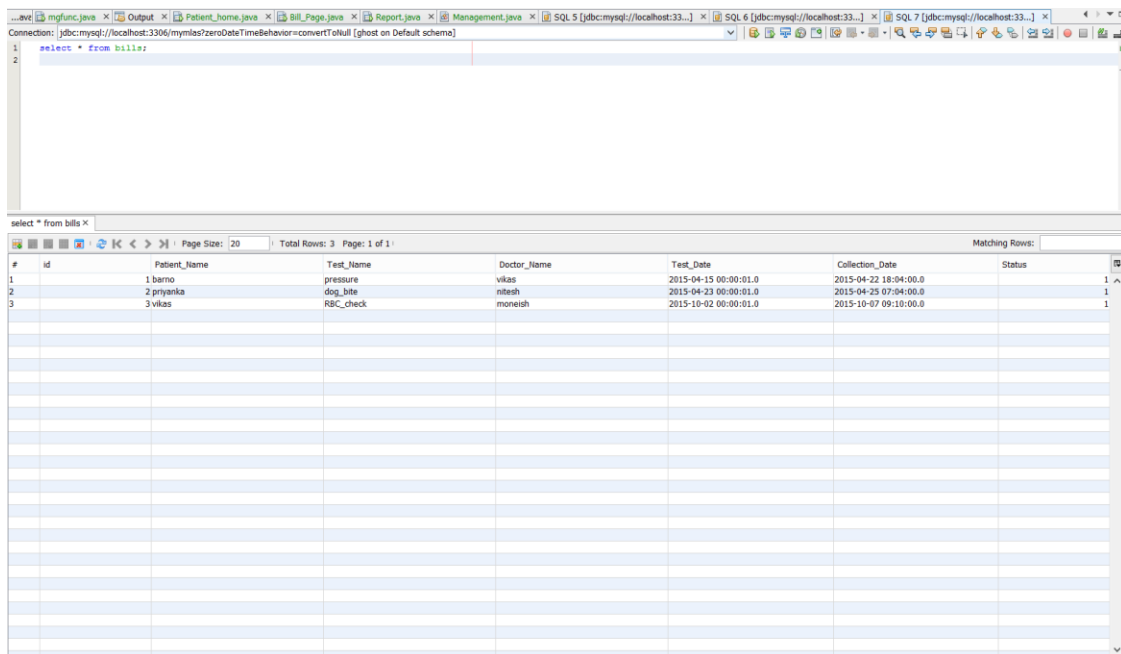
#	id	Type
1	1	3

COLLECT BILL

Collect Bill



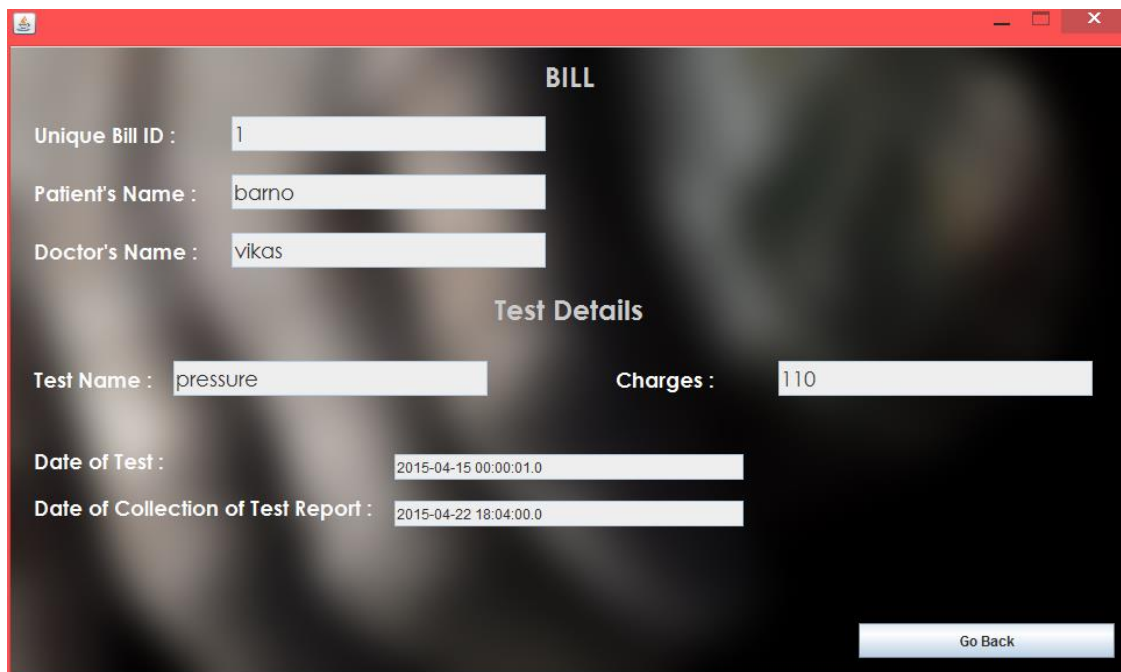
Bill Table



The screenshot shows a Java IDE with several tabs open. The active tab displays a SQL query: `select * from bills;`. Below the query, a table titled "select * from bills" shows the results. The table has 7 columns: #, id, Patient_Name, Test_Name, Doctor_Name, Test_Date, Collection_Date, and Status. There are 3 rows of data.

#	id	Patient_Name	Test_Name	Doctor_Name	Test_Date	Collection_Date	Status
1	1	barno	pressure	vikas	2015-04-15 00:00:01.0	2015-04-22 18:04:00.0	1
2	2	priyanka	dog_bite	nitesh	2015-04-23 00:00:01.0	2015-04-25 07:04:00.0	1
3	3	vikas	RBC_check	moneish	2015-10-02 00:00:01.0	2015-10-07 09:10:00.0	1

Bill Generated (Success)



The screenshot shows a Java application window titled "BILL". It contains a form with the following fields:

- Unique Bill ID : 1
- Patient's Name : barno
- Doctor's Name : vikas

Below these fields is a section titled "Test Details" with the following fields:

- Test Name : pressure
- Charges : 110
- Date of Test : 2015-04-15 00:00:01.0
- Date of Collection of Test Report : 2015-04-22 18:04:00.0

A "Go Back" button is located at the bottom right of the form.

COLLECT TEST REPORT

Collect Test Report

The screenshot shows a web application window titled "PATIENT HOME". The window has a red header bar with standard window controls (minimize, maximize, close) on the right. Below the header is a navigation bar with five buttons: "Take Test", "Account Details", "Collect Test Report", "Bills", and "Logout". The main content area is divided into two large, blurred rectangular sections. The left section is labeled "Bills" and the right section is labeled "Reports". The "Collect Test Report" button in the navigation bar is highlighted, indicating it is the active selection.

Test Report Collected (Success)

The screenshot shows a web application window titled "TEST REPORT". The window has a red header bar with standard window controls (minimize, maximize, close) on the right. The main content area is a form with two sections: "Patient Details" and "Test Details".

Patient Details

- Name : barno
- Gender : Male
- Doctor's Name : vikas

Test Details

- Test Name : pressure
- Test Charges : 110
- Test Results : Blood pressure is normal.

At the bottom right of the form, there is a "Go Back" button.