NAME: EMMANUEL DARKWAH AMPOFO

AMALITECH ORGANIZATION

PROJECT TITLE: HOTEL MANAGEMENT SYSTEM

**ACKNOWLEDGEMENT**

I would like to express my earnest gratitude to the Almighty God for the gift of life and seeing me through this program successfully.

I would also like to express my profound gratitude to Amalitech for the opportunities to learn a new skill and enhance my programming skills.

Lastly, my heartfelt thanks to Ahmad Mehmood, the course instructor for his swift response and crystal-clear explanations to questions related to the program of study

**TABLE OF CONTENTS**

Acknowledgement…………………………………………………………………….2

**Introduction**…………………………………………………………………………..4

Background……………………………………………………………………………4

**Statement of the Problem**……………………………………………………………4

**System Description**…………………………………………………………………..5

Guest Information Table………………………………………………………………5

Accommodation Table………………………………………………………………..6

Meals………………………………………………………………………………….7

Receipts……………………………………………………………………………….8

**System Implementation**……………………………………………………………..9

**Conclusion & Recommendations**………………………………………………….10

**Java Codes**…………………………………………………………………………..11

Connection to Database………………………………………………………………11

Login as Admin………………………………………………………………………12

Add Guest…………………………………………………………………………….12

Calculate Bill………………………………………………………………………….13

Receipt…………………………………………………………………………………14

Search by ComboBox………………………………………………………………….15

Reset……………………………………………………………………………………15

Load Guest Information………………………………………………………………..16

Images…………………………………………………………………………………..17

**List of Figures**…………………………………………………………………………18

**INTRODUCTION**

**Background.**

Genius Arena was developed following development steps for smooth running and management of Hill View Hotel. After careful information gathering from different sections of the hotel, it was well established the hotel needed a computerized management system.

Genius Arena was carefully designed to ensure maximum efficiency of the system at the hotel. The system was skillfully and carefully coded to seal any loopholes in the system. Genius Arena was developed using Java Programming Language in Eclipse Integrated Development Environment.

The system will indeed help the hotel management and the esteemed staff members to manage and steer the hotels functionalities and transactions to realize its maximum potential in addition to its competence in the hotel business.

**STATEMENT OF THE PROBLEM**

Hill View Hotel offers accommodation and meals as their primary services. Although the hotel is well known in Accra and its environs, it is not realizing its maximum potential due to delay of activities by the current redundant manual system.

Due to huge losses suffered by the hotel from frequent errors in the current. The current manual system has some known advantages such as,

* Staff do not require special computer skills to run the manual system
* No relevance on computer devices which may fail

It has lots of disadvantages too, namely

* Guest files can easily get lost or mixed up with other guest files
* Files occupy large storage space
* Unnecessary duplication of data
* Files are prone to unauthorized modification due to low security levels

**SYSTEM DESCRIPTION**

The system will cover taking guest information upon booking or arrival,Accommodation(rooms), Meals(restaurant), Check guest out when leaving, Reset fields to check next guest out, Generate Receipt.

The proposed system will

* Enable automated data entry methods
* Avoid data entry errors by use of input masks
* Enable only authorized data modifications
* Enable fast and retrieval of guest records and data for fast reference activities

***Guest Information Table***

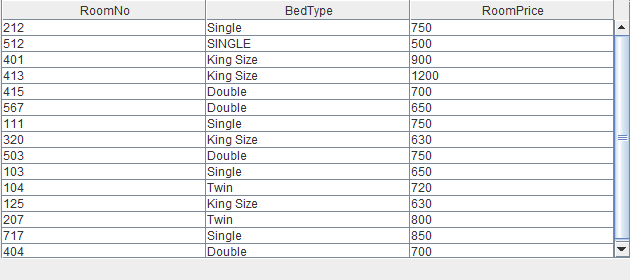
The table contains guest details that will be input when the guest books in the hotel. For phone in and walk in, details are taken and entered into the system by the staff. The table has the fields *GuestID, FirstName, SurName, Address, Phone, Sex, Room Type, RoomNumber, Room Price, InitialDeposit, NumberOfDays*.

*fig 1.1*

Guest Information Table

***Accommodation (Room) Table***

Contains information about rooms availability. This helps uniquely identify the guest in a particular room. These fields include *RoomNo, BedType, RoomPrice, RoomCondition*

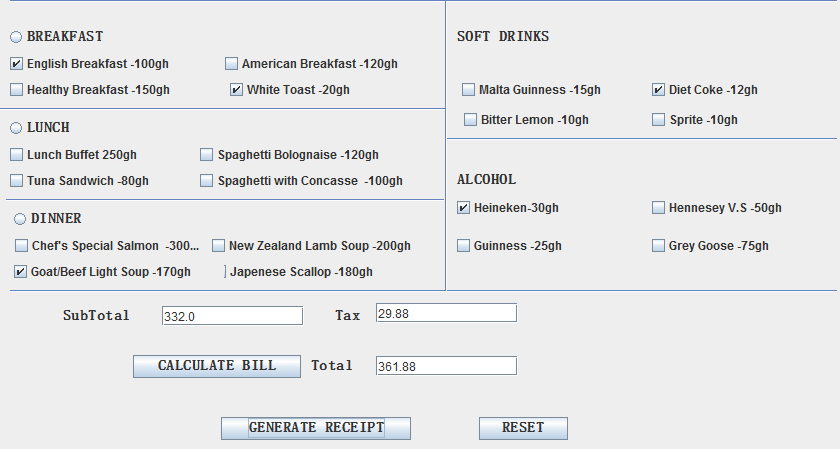
*fig 1.2*

*Accommodation Table*

***Meals (Restaurant) Table***

This contains the hotels catering transactions/information. This information is vital as this department is the backbone of any hotel aspiring to achieve its goals and realize its maximum potential.

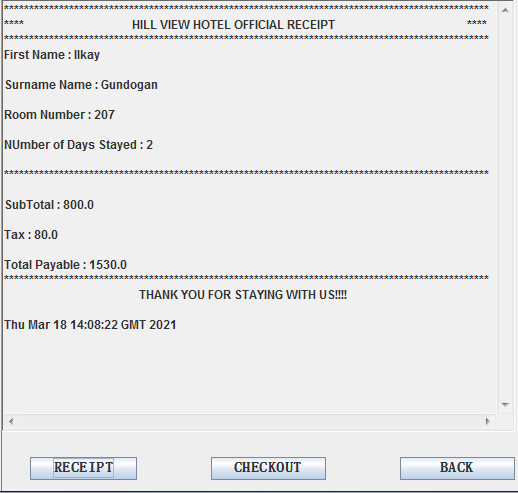
The system allows restaurant staff to pick from a wide range of options according to what the guest orders. The system enables automatic calculations of the *SubTotal, Tax, Total amount* charged for their orders or choice of meal

* fig 1.3*

*Meals*

***Receipts***

This generates official receipts on bills guest have made necessary payments to, be it meals or accommodation.

*fig 1.4*

*Receipts*

**SYSTEM IMPLEMENTATION**

After several careful consultations with management, staff and development team regarding the best changeover practices. A decision was taken on using the Parallel changeover method.

Parallel method is whereby the computerized system will run concurrently with the manual system before discarding the manual system.

Although expensive, the changeover method will prove to be the most efficient because

* Parallel changeover provides time for the database administrator to update all guest files before a total changeover to the new system.
* Provides time for staff to learn and adapt to the new system
* Lower the risk to the management in case od a technical hitch or breakdown as the manual system will still be in place as the development team fixes the technical hitch
* It is possible to troubleshoot any errors arising from loading process without affecting the hotels transactions as the manual system will still be in place to carry out the hotel activities smoothly.

**CONCLUSION AND RECOMMENDATIONS**

With the mandate from taking Amalitech to take a Java Programming Course and develop a real world working system, I have been able to come to a successful end of a journey that has seen me through ups and downs that seemed impossible but the outcome has been the birth of a computerized hotel management system named ***GENIUS ARENA***

It accomplishes the objectives I set out to attain earlier when I wrote the system objectives. The system will be able to serve its intended purpose and meet its objectives to the satisfaction of the hotel management, staff and guest.

**Recommendations**

I would like to recommend the following for the system and hotel to function to its best capabilities

* Introduce a website or an online portal to be integrated with the system to enable guest make bookings online
* The system should be troubleshooted and updated to maintain the high competence standards of the system
* As system enlarges, every staff of the hotel is identified uniquely by a username and password
* High security measures should be upheld.

**JAVA CODES**

**Connection to Database**

import java.sql.Connection;

import java.sql.DriverManager;

import java.util.\*;

public class hotelDbConnection {

static Connection connection ;

public static Connection getConnection() {

try {

Class.forName("org.sqlite.JDBC");

Connection connection = DriverManager.getConnection("jdbc:sqlite:C:\\Users\\Poff\\eclipse-workspace\\HotelManagementSystem\\src\\database\\hoteldata")

return connection;

}catch(Exception e) {

e.printStackTrace();

}

return null;

}

**Login as Admin**

**public** **void** actionPerformed(ActionEvent e) {

String username = textField.getText();

String password = passwordField.~~getText~~();

**if**(username.equals("admin") && password.equals("admin")) {

MainPage mainpage = **new** MainPage();

mainpage.setVisible(**true**);

frmGeniusArena.dispose();

}

**else** {

JOptionPane.*showMessageDialog*(btnAdminLogin, "Invalid Credentials");

}

}

**Add Guest**

**public** **void** actionPerformed(ActionEvent e) {

**try** {

String query = "Insert into guestinfo (Guestid, FirstName,Surname, Address, Phone, Sex, Roomtype,RoomNo,RoomPrce InitialDeposit,NumberOfDays) values (?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement pst = connection.prepareStatement(query);

pst.setString(1, textFieldGuestID.getText());

pst.setString(2, textFieldName.getText());

pst.setString(3, textFieldSurname.getText());

pst.setString(4, textFieldAddress.getText());

pst.setString(5, textFieldPhone.getText());

pst.setString(6, textFieldType.getText());

pst.setString(7,textFieldRoomtype.getText());

pst.setString(8,textFieldrn.getText());

pst.setString(9,textFieldrpx.getText());

pst.setString(10, textFieldAmount.getText());

pst.setString(11, textFieldnDays.getText());

pst.execute();

JOptionPane.*showMessageDialog*(btnNewButton, "GUEST CHECK\_IN SUCCESSFUL");

pst.close();

textFieldGuestID.setText("");

textFieldName.setText("");

textFieldSurname.setText("");

textFieldAddress.setText("");

textFieldAddress.setText("");

textFieldPhone.setText("");

textFieldType.setText("");

textFieldRoomtype.setText("");

textFieldAmount.setText("");

textFieldnDays.setText("");

textFieldrn.setText("");

textFieldrpx.setText("");

} **catch** (SQLException e1) {

e1.printStackTrace();

}

}

**Calculate Bill**

**public** **void** actionPerformed(ActionEvent e) {

**double** total;

**double** subtotal = 0;

**double** tax;

**final** **double** TAXRATE = 0.1;

subtotal = subtotal + Double.*parseDouble*(textFieldRPrice.getText());

textFieldsubtotal.setText(Double.*toString*(subtotal));

tax = subtotal \* TAXRATE;

total = tax + subtotal;

textFieldtax.setText(Double.*toString*(tax));

textFieldtotal.setText(Double.*toString*((total \* Double.*parseDouble*(textFieldDays.getText()) - Double.*parseDouble*(textFieldDepo.getText()))));

}

**Receipt**

**public** **void** actionPerformed(ActionEvent e) {

Date date1 = **new** Date();

String date = date1.toString();

textArea.setText("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

textArea.setText(textArea.getText()+ "\*\*\*\* HILL VIEW HOTEL OFFICIAL RECEIPT \*\*\*\*\n");

textArea.setText(textArea.getText()+ "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

textArea.setText(textArea.getText()+ "First Name : "+textFieldFName.getText()+"\n");

textArea.setText(textArea.getText()+ "\n");

textArea.setText(textArea.getText()+ "Surname Name : "+textFieldSName.getText()+"\n");

textArea.setText(textArea.getText()+ "\n");

textArea.setText(textArea.getText()+ "Room Number : "+textFieldRNo.getText()+"\n");

textArea.setText(textArea.getText()+ "\n");

textArea.setText(textArea.getText()+ "NUmber of Days Stayed : "+textFieldDays.getText()+"\n");

textArea.setText(textArea.getText()+ "\n");

textArea.setText(textArea.getText()+ "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

textArea.setText(textArea.getText()+ "\n");

textArea.setText(textArea.getText()+ "SubTotal : "+textFieldsubtotal.getText()+"\n");

textArea.setText(textArea.getText()+ "\n");

textArea.setText(textArea.getText()+ "Tax : "+textFieldtax.getText()+"\n");

textArea.setText(textArea.getText()+ "\n");

textArea.setText(textArea.getText()+ "Total Payable : "+textFieldtotal.getText()+"\n");

textArea.setText(textArea.getText()+ "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

textArea.setText(textArea.getText()+ " THANK YOU FOR STAYING WITH US!!!!\n");

textArea.setText(textArea.getText()+ "\n"+date+"\n");

}

**Search By ComboBox**

**public** **void** actionPerformed (ActionEvent e) {

**try** {

String query = "Select \* from guestinfo where GuestID=?";

PreparedStatement pst = connection.prepareStatement(query);

pst.setString(1, (String)comboBoxName.getSelectedItem());

ResultSet rs = pst.executeQuery();

**while**(rs.next()) {

textFieldGuest.setText(rs.getString("GuestID"));

textFieldFName.setText(rs.getString("FirstName"));

textFieldSName.setText(rs.getString("Surname"));

textFieldDepo.setText(rs.getString("InitialDeposit"));

textFieldRNo.setText(rs.getString("RoomNo"));

textFieldRType.setText(rs.getString("RoomType"));

textFieldRPrice.setText(rs.getString("RoomPrice"));

textFieldDays.setText(rs.getString("NumberOfDays"));

}

pst.close();

}**catch**(Exception e1) {

}

}

**Reset**

**public** **void** actionPerformed(ActionEvent e) {

textFieldtotal.setText("");

textFieldtax.setText("");

textFieldsubtotal.setText("");

textFieldDays.setText("");

textFieldGuest.setText("");

textFieldFName.setText("");

textFieldSName.setText("");

textFieldDepo.setText("");

textFieldRNo.setText("");

textFieldRType.setText("");

textFieldRPrice.setText("");

textArea.setText("");

}

**Check-Out**

**public** **void** actionPerformed(ActionEvent e) {

**int** action = JOptionPane.*showConfirmDialog*(**null**, "ARE YOU SURE YOU WANT CHECK THIS GUEST OUT?","CheckOut Confirmation", JOptionPane.***YES\_NO\_OPTION***);

**if** (action == 0) {

**try** {

String query = "Delete from guestinfo where GuestID='"+textFieldGuest.getText()+" '";

PreparedStatement pst = connection.prepareStatement(query);

pst.execute();

JOptionPane.*showMessageDialog*(**null**, "GUEST CHECKED\_OUT SUCCESSFUL");

}**catch**(Exception e1) {

e1.printStackTrace();

}

}

textFieldtotal.setText("");

textFieldtax.setText("");

textFieldsubtotal.setText("");

textFieldDays.setText("");

textFieldGuest.setText("");

textFieldFName.setText("");

textFieldSName.setText("");

textFieldDepo.setText("");

textFieldRNo.setText("");

textFieldRType.setText("");

textFieldRPrice.setText("");

textArea.setText("");

}

**Load Guest Information**

**public** **void** actionPerformed(ActionEvent e) {

**try** {

String query = "Select GuestID,FirstName,Surname,Address,Phone,Sex,RoomType,InitialDeposit,NumberOfDays from guestinfo";

PreparedStatement pst = connection.prepareStatement(query);

ResultSet rs = pst.executeQuery();

table.setModel(DbUtils.*resultSetToTableModel*(rs));

}**catch**(Exception e1) {

e1.printStackTrace();

}

}

**Images**

lblpicture = **new** JLabel("");

Image img1 = **new** ImageIcon(**this**.getClass().getResource("Images\\restaurant.jpg")).getImage();

lblpicture.setIcon(**new** ImageIcon(img1));

lblpicture.setBounds(10, 11, 371, 220);

contentPane.add(lblpicture);

**LIST OF FIGURES**

Figure 1.1: Guest Information Table

Figure1.2: Accommodations Table

Figure 1.3: Meals

Figure 1.4: Receipts