

A PROJECT REPORT

ON

EVENT MANAGEMENT SYSTEM

DONE BY

Pooja Prathmesh Warang

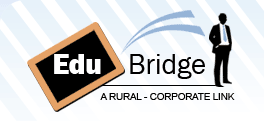
(EBEON0920321994)

Under the Guidance of,

**Mr. V JAYANTH**

**Technical Trainer**

**SOFTWARE DEVELOPER PROJECT**

**­**

**ARISE**

**ONCAMPUS**

**SOFTWARE DEVELOPER TRAINEE**

(Thane)­­

2020

**­­­­**

**ABSTRACT**

Currently, couples face frustrating situation in looking for wedding products and services here and there, and vendors and planners have difficulties to expand their customer base.

The Event Management System designed to be a one-stop web-based platform used by couples, vendors and planners. Each user has different perspective towards the system, depends on the needs in terms of their roles in the system. Couples use the system on their wedding design, plan and management. Vendors use the system to market and sell their services and products. Planners use the systems to assist in interaction between couples and vendors.

Event Management System is a business platform that aims to help organize successful wedding event. The system assists couples in the decision making and planning processes associated with all aspects of a wedding organization.

The system offers features that couples can retrieve information for wedding products and services as well as information of vendors and planners in the shortest possible time. Vendors and planners can also gain benefits of getting more recognition from clients and generating more revenue.

**TABLE OF CONTENT**

**S. TITLE PAGE NO**

ABSTRACT 2

**1. INTRODUCTION TO THE STUDY**

Introduction 5

Objective of the System 5

**2. Established the need of new System** 6

Proposed system 7

**3. Project Category** 8

**4. RESEACH METHOD** 9

**5. SYSTEM ANALYSIS** 10

Principles of System Analysis 10

Feasibility Study 11

Cost and Benefit Analysis 11

Technical Feasibility 12

Operational Feasibility 12

**6.System Life Cycle** 13

**7.Scope of Work** 14

Identification of Need 15

ER Diagram 16

**8.Data Flow Diagram** 17

**9. SYSTEM DESIGN**

System Design 24

Scheduling 25

Proposed enhancement of the project 25

**10. CONCLUSION 27**

**11. REFERENCES 28**

**APPENDIX-I 29**

* **SNAPSHOTS 29**
* **SOURCE CODE 33**

**CHAPTER – 1**

**INTRODUCTION TO THE STUDY**

**Event Management** System is a web-based project which act as a bridge between potential users and wedding planners. As people spend lots of money on Weddings, Parties and MICE, but they involve their selves in each and every affair in such a way that at the end of the day they feel that they have not seen the wedding of their only Son/Daughter or they have not been able to enjoy the functions. That is why, a Wedding Planner-Co-coordinator is required to make people comfortable on the day of the Wedding /Event.

So that I have tried to make a web project so that all the planning’s for the food, venues, invitation cards, bookings can be done on time. This project is developed to make the users comfortable to enjoy each and every function along with their family, friends and relatives.

**OBJECTIVES OF THE SYSTEM**

**The objective of this development effort is: -**

* To provide the information regarding **Wedding.**
* To provide the information about the **Booking of user’s choice.**
* Online Users can submit their queries by giving the wedding destination, tentative date, number of guests, contact number etc.
* Interested Users can book accommodations for their guests in Hotels. It can be a 5 Star,4 Star or 3 Star hotel. Online users can provide their requirements like no. of rooms, no. of nights, accommodation budget etc.
* Wedding themes are available in the websites.
* There is a facility to book caterers according to the budget. Different plans are available by the caterers; these are Economy, Gold, Platinum and Silver.
* Mailing facility to reply clients.
* Different venues are available for the choices.
* Availability of Wedding cards, Wedding cakes, Wedding dresses, Transport, Flower Decoration etc.
* To provide the functionality of **online Booking and Cancellation.**
* To provide the information of **status of catering, facilities available.**
* To provide the facility to maintain the records of users.
* Information can be easily accessed.
* This system will save the time of the Users.

**CHAPTER – 2**

**ESTABLISH THE NEED OF NEW SYSTEM**

The existing system is manual /machine systems where the users must have to perform their manually. It will take more time and this whole procedure is very tedious and takes a lot of time.

Problems of the existing system

* It is limited to a single system.
* It has lot of manual work (Manual work does not mean working with pen or paper, rather includes working on spread sheets and other simple software.)
* The present system is very less secure.
* It unable to generate different kinds of reports.
* It is difficult to maintain phone bookings and cancellations.
* Does not provide global support. User cannot do the reservation from remote place.
* Retrieval of information is difficult and time consuming.
* A lot of time and manual effort is required

**AIM:**

The aim of the study to fully related with Event Management system.

· The Software is for the automation of Event Management System.

· It maintains two levels of users:-

\_ Administrator Level

\_ User Level

· The Software includes:-

\_ Maintaining Patient details.

\_ Providing Prescription, Precautions and Diet advice.

\_ Providing and maintaining all kinds of tests for a patient.

\_ Billing and Report generation

**PROPOSED SYSTEM**

**l. Employee Details:** The new proposed system stores and maintains all the employees

details.

**2. Calculations:** The new proposed system calculates salary and income tax

automatically and it is very fast and accurate.

**3. Registers:** There is no need of keeping and maintaining salary and employee register

manually. It remembers each and every record and we can get any report related to

employee and salary at any time.

**4. Speed:** The new proposed system is very fast with 100% accuracy and saves time.

**5. Manpower:** The new proposed system needs less manpower. Less people can do the

large work.

**6. Efficiency:** The new proposed systems complete the work of many salesperson in

less time.

**7. Past details:** The new proposed system contains the details of every past doctor and

patients for future assistance.

**8. Reduces redundancy:** The most important benefit of this system is that it reduces

the redundancy of data within the data.

**9. Work load:** Reduces the work load of the data store by helping in easy updates of

the products and providing them with the necessary details together with financial

transactions management.

**10. Easy statements:** Month-end and day-end statement easily taken out without

getting headaches on browsing through the day end statements.

**NEED:**

I have designed the given proposed system in the JSP to automate the process of day to

day activities of Wedding Planning like Room activities, select the venue, book venue, select location, select decoration, costume etc., online facilities to the multiple users etc.

**Online Event Management System is a** Management system with Automation System this project is aimed at automation of a Wedding Planning and thus to develop customize software package.

Efficiency: The basic need of the project is efficiency. The project should be efficient so

that whenever a new customer is added, and automatically a venue is assigned and also a

decoration is assigned to the customers requirement according to the customers’ design.

Control: The complete control of the project is under the hands of authorized person who

has the password to access this project and illegal access is not supposed to deal with. All

the control is under the administrator and the other members have the rights to just see

the records not to change any transaction or entry.

Security: Security is the main criteria for the proposed system. Since illegal access may

corrupt the database and it will affect not only the hospital but also it also affects the

patient’s life. So security has to be given in this project.

**CHAPTER – 3**

**PROJECT CATEGORY**

This project is basically categorized as RDBMS. The project is based on three tier architecture**.** The three tier architecture where the application is divided into three logical constituents.

* Presentation layer – In this layer mainly following pages contained:
  + Web Pages
  + User Control
  + Master Pages
* Business Layer – Business Logic,
  + Result Engine
  + User permissions logic
  + Access Rights
* Data Layer – Provide handling and validation of data (MS SQL Server in this case).
  + MY SQL Workbench

**CHAPTER – 4**

**RESEARCH METHODOLOGY**

The project ‘Wedding Planner Management System’ is based on the database, object oriented and

networking techniques. As there are many areas where we keep the records in database

for which we are using MY SQL software which is one of the best and the easiest

software to keep our information. This project uses JAVA as the front-end software which

is an Object Oriented Programming and has connectivity with MY SQL. It is a web based

application in which number of clients can also access with a server.

**HARDWARE**

**Processor :** Pentium 2.4 GHz or above

**Memory :** 256 MB RAM or above

**Cache Memory :** 128 KB or above

**Hard Disk :** 3 GB or above [at least 3 MB free space required]

**Pen Drive : 5 GB**

**SOFTWARE**

**Operating System :** Windows XP (Professional).

**Font-End Tool :** JSP, Servlets, Java.

**Back-End :** My Sql

**FRONT END**

We have implemented **Java** for all the Client side validations. Client side

Java is designed to reside inside HTML document & ensure they run properly. It is

object based, event driven, platform independent. These are important parts of any Web

application to implement Client side Validations and the invalid data is not submitted.

The form is not submitted until user fills in correct data. It is extremely useful to restrict

mistakes by user.

**BACK END**

We have used MySQL. MySQL provides efficient/effective solution for major database

tech.

- Large database and space management.

- Many concurrent database users.

- High transaction processing requirement

- High Availability

- Industry accepted standards

- Manageable security

- Portability

**CHAPTER – 5**

**SYSTEM ANALYSIS**

**PRINCIPLES OF SYSTEM ANALYSIS:**

**PRINCIPLES:**

Understand the problem before you begin to create the analysis model.

Develop prototypes that enable a user to understand how human machine

interaction will occur.

Record the origin of and the reason for every requirement.

Use multiple views of requirements like building data, function and behavioral

models.

Work to eliminate ambiguity

System Analysis is a separation of a substance into parts for study and their

implementation and detailed examination.

Before designing any system it is important that the nature of the business and the way

it currently operates are clearly understood. The detailed examination provides the

specific data required during designing in order to ensure that all the client's

requirements are fulfilled. The investigation or the study conducted during the analysis

phase is largely based on the feasibility study. Rather it would not be wrong to say that

the analysis and feasibility phases overlap. High-level analysis begins during the

feasibility study. Though analysis is represented as one phase of the system

development life cycle (SDLC), this is not true. Analysis begins with system

initialization and continues until its maintenance. Even after successful implementation

of the system, analysis may play its role for periodic maintenance and up gradation of

the system. One of the main causes of project failures is inadequate understanding, and

one of the main causes of inadequate understanding of the requirements is the poor

planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three

questions:

• What type of information is required?

• What are the constraints on the investigation?

• What are the potential problems that may make the task more difficult?

Keeping the above questions in mind and considering the survey conducted to

determine the need of the system; the total system was designed and can be described

as under:

**The three major parts of the system are:**

 **Providing Information:**

The system is effectively used to provide large variety of information to the interested

customer. The major purpose of the site is to easily provide access to records of various

Job seekers & users of matrimonial such as resume & profile of boys and girls those

who want to search a life partner with quick update to latest modifications in the

records. This thing is not at all possible in printed material, which are updated only

once a few weeks. It also gives information about the general usage of the system for

first time visitors. The system itself works as a information provider for company & life

partner seekers.

**Preliminary Investigation**

System development, a process consisting of two major steps of system analysis and

design, start when management or sometimes system development personnel feel that a

new system or an improvement in the existing system is required. The system

development life cycle is classically thought of as the set of activities that analysts,

designers and users carry out to develop and implement an information system. The

system development life cycle consists of the following activities:

 Preliminary investigation

 Determination of system requirements

 Design of system

 Development of software

 System testing

 Implementation, evaluation, and maintenance

**A request to take assistance from information system can be made for many reasons,**

**but in each case someone in the organisation initiates the request is made, the first**

**system activity the preliminary investigation begins. This activity has three parts:**

1) Request clarification

2) Feasibility study

3) Request approval

**Request clarification**: Many requests from employees and users in the organisations are

not clearly defined, Therefore it becomes necessary that project request must be eximined

and clarified properly before considering systems investigation.

**FEASIBILITY STUDY:**

The feasibility study proposes one or more conceptual solution to the problem set of the

project. In fact, it is an evaluation of whether it is worthwhile to proceed with project or

not.

1. Evaluation of feasibility of such solutions. Such evaluation often indicates

shortcomings in the initial goals. This step is repeated as the goals are adjusted and

the alternative solutions are evaluated.

Feasibility analysis usually considers a number of project alternatives, one that is chosen

as the most satisfactory solution. These alternatives also need to be evaluated in a broad

way without committing too many resources. Various steps involved in feasibility

analysis are:

2. To propose a set of solution that can realize the project goal. These solutions are

usually descriptions of what the new system should look like.

Four primary areas of interest in feasibility study are:

**Economic Feasibility**: An evaluation of development cost weighed against the ultimate

income of benefit derived from the development system of product. In economic

feasibility, cost benefit analysis is done in which expected cost and benefits are evaluated.

**COST AND BENEFIT ANALYSIS:**

Developing an IT application is an investment. Since after developing that application it

provides the organization with profits. Profits can be monetary or in the form of an

improved working environment. However, it carries risks, because in some cases an

estimate can be wrong. And the project might not actually turn out to be beneficial.

Cost benefit analysis helps to give management a picture of the cost, benefits and risks. It

usually involves comparing alternate investments.

Cost benefit determines the benefits and savings that are expected from the system and

compares them with the expected costs.

In performing cost and benefit analysis it is important to identify cost and benefits

factors. Cost and benefits can be categorized into the following categories:

1. **Development Costs –** Development costs is the costs that are incurred during the

development of the system. It is one time investment.

2. **Operating Costs –** Operating Costs are the expenses required for the day to day

running of the system. Examples of Operating Costs are Wages, Supplies and

Overheads.

3. **Hardware/Software Costs –** It includes the cost of purchasing or leasing of

computers and it’s peripherals. Software costs involves required S/W costs.

4. **Personnel Costs –** It is the money spent on the people involved in the development

of the system.

5. **Facility Costs –** Expenses that are incurred during the preparation of the physical site

where the system will be operational. These can be wiring, flooring, acoustics,

lightning, and air-conditioning.

6. **Supply Costs –** These are variable costs that are very proportionately with the

amount of use of paper, ribbons, disks, and the like.

 **BENEFITS**

We can define benefits as

**Profit or Benefit = Income – Costs**

Benefits can be accrued by:

 Increasing income, or

 Decreasing costs, or

 Both

**TECHNICAL FEASIBILITY:**

Technical Feasibility includes existing and new H/W and S/W requirements that are

required to operate the project using JSP. The basic S/W requirement is J2EE in which

the front end of the online hospital management project has been done. The basic entry

forms are developed in JSP and the data is stored in the MY SQL.

**OPERATIONAL FEASIBILITY:**

Operational feasibility is mainly concerned with issues like whether the system will be

used if it is developed and implemented. Whether there will be resistance from users that

will affect the possible application benefits? The essential questions that help in testing

the technical feasibility of a system are following:

 Does management support the project?

 Are the users not happy with current business practices? Will it reduce the time

considerably? If yes, then they will welcome the change and the new system.

 Have the users involved in the planning and development of the project? Early

involvement reduced the probability of resistance towards the new system.

 Will the proposed system really benefit the organization? Does the overall response

increase? Will accessibility of information be lost? Will the system affect the

customers in considerable way?

**Legal Feasibility**:

A determination of any infringement, violation, or liability that could result from

development of the system. Legal feasibility tells that the software used in the project

should be original purchased from the legal authorities and they have the license to use it

or the software are pirated.

**Alternatives**:

An evaluation of alternative approaches to the development of system or product.

**CHAPTER – 6**

**SYSTEM LIFE CYCLE:**

System life cycle is an organizational process of developing and maintaining systems.

The different phase of software development life cycle is shown below.

PRELIMINARY

**DETERMINATION OF**

**REQUIREMENTS**

**REVIEW RUNNING**

**SYSTEM AND SYSTEM**

**MAINTENANCE**

**DEVELOPMENT OF**

**PROTOTYPE SYSTEM**

**SYSTEM**

**IMPLEMENTATION**

**DESIGN OF SYSTEM**

**DEVELOPMENT OFSOFTWARE**

**AND INVESTIGATION**

**INVESTIGATION**

**CODING**

**SYTEM TESTING**

**FIG: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME**

**DEVOTED**

**CHAPTER – 7**

**SCOPE OF WORK**

The scope of the system is defined on the basis of various functionalities provided by the system. The scope can be explained as:-

* Build a web based system.
* Information handling of the booking of the Wedding Planners i.e. new records can be created, data retrieval, update and cancellation of the booking functionality.
* The scope of the project is to develop customize software package for reducing the manual problems.
* It should deals with the online Wedding Planning as well as cancellation.
* This system never decreases the manpower but helps the development of available manpower and optimizes the manpower by which banquet’s standards and capabilities can be scaled to higher dimension.

**IDENTIFICATION OF NEED:**

I have designed the given proposed system in the JSP to automate the process of news

sites. Many different people use Usenet, for many different reasons, ranging from

discussions of various topics, requests for help or to trade media, which is often seen as

the ‘dark side’ to Usenet — the abuse of public news groups to trade copyrighted or

offensive material. Thankfully however, those that use Usenet responsibly far out weight

the few who use it as a tool for piracy etc.

Many of the people using and contributing to Usenet are people who work with computer

technology. These people often use Usenet to seek help with a piece of software or

suggest improvements, indeed one of the early functions of Usenet was to help users

identify and report bugs found in software.

Unfortunately, the massive growth seen within Usenet has also made it a difficult

medium to keep track of, the great variety and number of articles can take considerable

time to filter through in order to find those of interest. There have been a variety of

software applications created to help deal with this problem and many are freely available

for personal use.

The following steps that give the detailed information of the need of proposed system are:

**Performance:** During past several decades, the records are supposed to be manually

handled for all activities. The manual handling of the record is time consuming and

highly prone to error. To improve the performance of the Company system, the

computerized system is to be undertaken. The computerized project is fully computerized

and user friendly even that any of the members can see the report and status of the

company.

**Efficiency:** The basic need of this website is efficiency. The website should be efficient

so that whenever a new user submit his/her details the website is updated automatically.

This record will be useful for other users instantly.

**Control:** The complete control of the project is under the hands of authorized person who

has the password to access this project and illegal access is not supposed to deal with. All

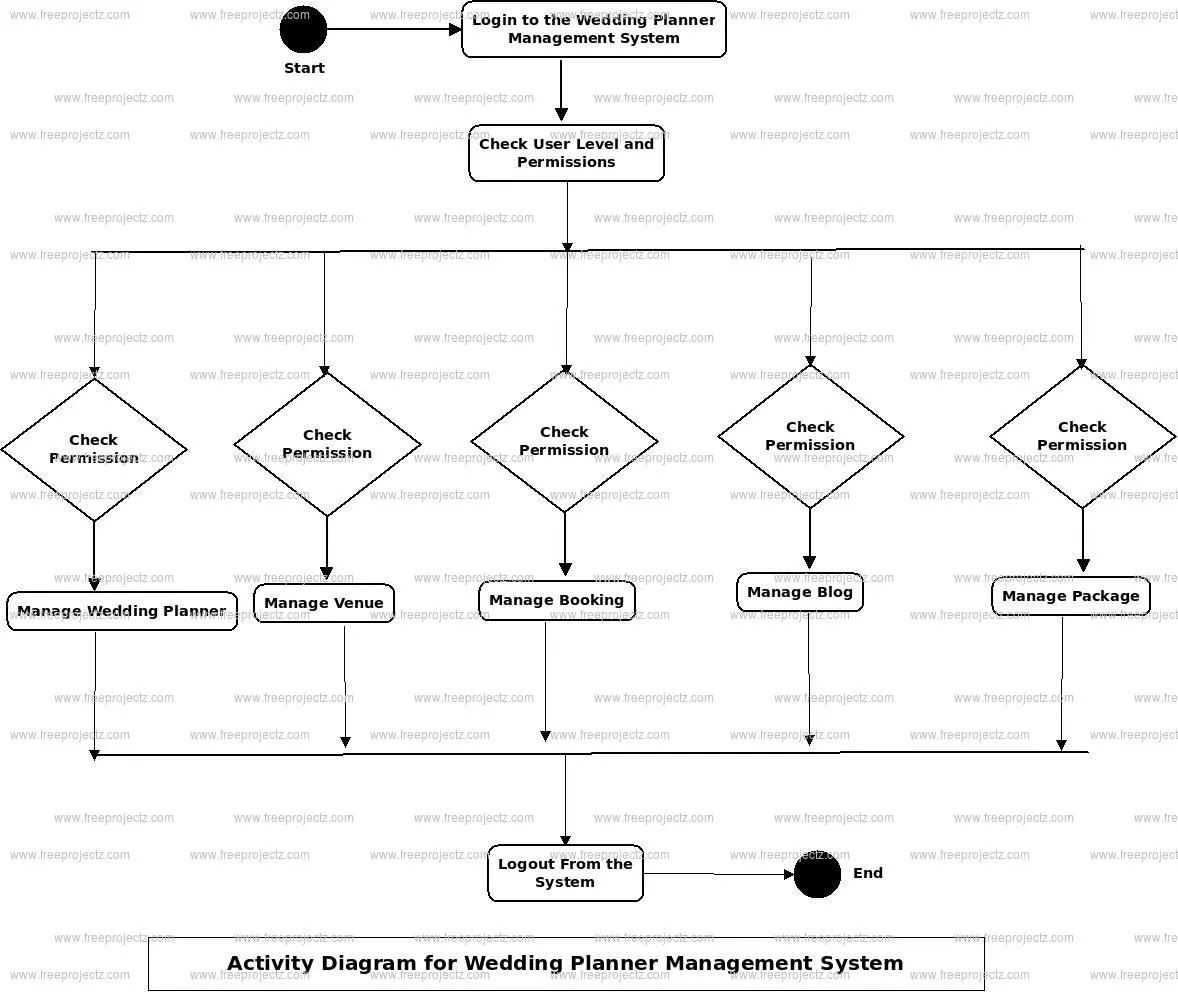
the control is under the administrator and the other members have the rights to just see

the records not to change any transaction or entry.

**Security:** Security is the main criteria for the proposed system. Since illegal access may

corrupt the database. So security has to be given in this project.

**E-R DIAGRAM**



**8. DATA FLOW DIAGRAM**

### Context Level DFD for Online Wedding Planning System

**3.**

**Cancellation**

**7.**

**Availability**

**&**

**Facilities**

**2.**

**Booking**

**6.**

**Party Halls**

**5.**

**Photographers**

**Online Event Management System**

**1.**

**Admin**

**4.**

**Other Arrangements**

**DFD for Login Module**

**USER**

**Online Event Management System**

**Screen**

**Admin Panel**

**Login Module**

**DFD for Booking Module**

**Customer**

**Invoice**

**Accounts**

**Caterers**

**Booking Hall**

**Customer**

**Online Event Management System**

**Theme**

DFD for Booking Inquiries Form

**Customer**

**Online Event Management System**

**Booking**

**Status**

**Party Hall**

**Booking**

**DFD For Cancellation Module**

**Customer**

**Invoice**

**Online Event Management System**

**DFD for Entering the Transport Details**

**Accounts**

**Booking**

**ADMIN**

**Online Event**

**Manageent System**

**Transport**

**DFD for Entering the Photographer Details**

**ADMIN**

**Online Event Hall Booking**

**Photographer**

**CHAPTER – 9**

**SYSTEM DESIGN**

**System Design:**

The design document that we will develop during this phase is the blueprint of the

software. It describes how the solution to the customer problem is to be built. Since

solution to complex problems isn’t usually found in the first try, iterations are most likely

required. This is true for software design as well. For this reason, any design strategy,

design method, or design language must be flexible and must easily accommodate

changes due to iterations in the design. Any technique or design needs to support and

guide the partitioning process in such a way that the resulting sub-problems are as

independent as possible from each other and can be combined easily for the solution to

the overall problem. Sub-problem independence and easy combination of their solutions

reduces the complexity of the problem. This is the objective of the partitioning process.

**SCHEDULING:**

Scheduling of a software project does not differ greatly from scheduling of any multitask

engineering effort. Therefore, generalized project scheduling tools and techniques

can be applied with little modification to software projects.

Program evaluation and review technique (PERT) and critical path method (CPM) are

two project scheduling methods that can be applied to software development. Both

techniques are driven by information already developed in earlier project planning

activities.

**Estimates of Effort**

 A decomposition of the product function.

 The selection of the appropriate process model and task set.

 Decomposition of tasks.

Interdependencies among tasks may be defined using a task network. Tasks, sometimes

called the project Work Breakdown Structure (WBS) are defined for the product as a

whole or for individual functions. Both PERT and CPM provide quantitative tools that allow the software planner to (1) determine the critical path-the chain of tasks that determines the duration of the project; (2) establish "most likely" time estimates for individual tasks by applying statistical

models; and (3) calculate "boundary times" that define a time window" for a particular task.

Boundary time calculations can be very useful in software project scheduling. Slippage in

the design of one function, for example, can retard further development of other functions. It describes important boundary times that may be discerned from a PERT or CPM network: (I) the earliest time that a task can begin when preceding tasks are completed in the shortest possible time, (2) the latest time for task initiation before the minimum project completion time is delayed, (3) the earliest finish-the sum of the earliest start and the task duration, (4) the latest finish- the latest start time added to task duration, and (5) the total float-the amount of surplus time or leeway allowed in scheduling tasks so that the network critical path maintained on schedule. Boundary time calculations lead to a determination of critical path and provide the manager with a quantitative method for evaluating progress as tasks are completed.

Both PERT and CPM have been implemented in a wide variety of automated tools that

are available for the personal computer. Such tools are easy to use and take the scheduling methods described previously available to every software project manager.

**PROPOSED ENHANCEMENT OF THE PROJECT**

**DRAWBACKS:**

1. No automatic backup facilities available.

2. To run the application Internet Explorer 5.0 and above is required.

3. High bandwidth is required for as the transaction rate is high and third party gateway.

**PROPOSED ENHANCEMENT:**

1. Data backup facility will be introduced.
2. Graphical richness is required for the more user interactivity.

**10. SECURITY**

**The security measures have been taken in a bid to make the software full proof in terms of various activities: -**

1. Authorization of the user as well as administrator
2. Input Validations of the forms
3. Output Validations of the result
4. Login timing to maintain the Session management
5. Logout timing of the user
6. Data Security
7. Software Security
8. Hardware Security
9. Illegal copying should be restricted

These are the few measures that are taken into account to meet the security measures while developing a project for an organization.

**CONCLUSIONS**

This project has been a rewarding experience in more than one way. The entire project

work has enlightened us in the following areas.

The Event Management System is a wedding services provider that offers wedding plans and services.

Online Event Management System deals with the information System for Bookings. This project explains about the booking of the Party Places from a short event to a complete combination of activities, including meetings, conferences, exhibitions etc.

This website helps any authorized person to book the places, decoration, dresses, photographers; flower decorators etc. from internet and also shows the availability of a particular requirement at the chosen date and time.

**11. REFERENCE**

**BOOKS:**

1. Mathew MacDonal Beninning ASP.Net 2.0 in C# 2005: From Novice to Professional, First edition Apress Publication – 2006
2. James Foxall: Sams Teach Yoursel Microsoft Visual C# 2005 in 24 hours, complete starter kit, First edition
3. Damon Armstrong: Pro ASP.Net 2.0: Website programming, First edition
4. Alison Batter: Sams Teach Yourself Microsoft SQL Server 2005 express in 24 hours, First edition
5. Joseph Schmuller: Sams Teach Yourself UML in 24 Hours, Third edition
6. Ron, Patton, “ Software Testing”, Second Edition, Pearson

**WEB REFERENCES:**

1. http:// [www.asp.net/](http://www.asp.net/)
2. <http://www.csharphelp.com/>
3. [http://www.google.com](http://www.google.com/)
4. <http://msdn.microsoft.com/net/quickstart/aspplus/default.com>
5. <http://www.aspfree.com/>

**APPENDIX-I**

**SNAPSHOTS:**

**SOURCE CODE**

**FrontPage.java**

**/\*\* This class displays the Home Page of Wedding Planner. It consists of control flow for all the major functionalities of this application**

**\* like log in, sign up, plan a wedding and send invitation.**

**\* @author Vipul Patil \*/**

**package com.proj;**

**import java.awt.Button;**

**import java.awt.Color;**

**import java.awt.Cursor;**

**import java.awt.Dimension;**

**import java.awt.Font;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.awt.event.WindowAdapter;**

**import java.awt.event.WindowEvent;**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.ResultSet;**

**import java.sql.SQLException;**

**import javax.swing.BorderFactory;**

**import javax.swing.ImageIcon;**

**import javax.swing.JButton;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import javax.swing.JPanel;**

**public class FrontPage{**

**final static Button loginButton = new Button("Log In");**

**final static Button signupButton = new Button("Sign Up");**

**final static Button logoutButton = new Button("Log Out");**

**final static JFrame window = new JFrame("Wedding Planner - Welcome");**

**final static JPanel panel = new JPanel();**

**final static JButton planButton = new JButton("Plan a Wedding");**

**final static JButton invitationsButton = new JButton("Wedding Invitations");**

**/\*\* This class contains the main function which indicates that the control flow of the whole application begins from this page.**

**\* @param args**

**\* @throws ClassNotFoundException \*/**

**public static void main(String[] args) throws ClassNotFoundException{**

**window.setSize(630, 600); // (width, height) /\*\* Front Page Frame: window \*/**

**window.setResizable(false);**

**/\*\* As this is the main window, if the user clicks to close this window then they should be asked once again whether he or she has to exit for sure. \*/**

**window.addWindowListener(new WindowAdapter(){**

**public void windowClosing(WindowEvent e){**

**int selectedOption = JOptionPane.showConfirmDialog(window, "Exit Wedding Planner?", "Confirm Exit", JOptionPane.YES\_NO\_OPTION, JOptionPane.PLAIN\_MESSAGE);**

**if(selectedOption == JOptionPane.YES\_OPTION)**

**window.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**else if(selectedOption == JOptionPane.NO\_OPTION)**

**window.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);**

**else**

**window.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);**

**}**

**});**

**/\*\* panel is contained on top of the window, on which various components can be added \*/**

**panel.setLayout(null);**

**panel.setPreferredSize(new Dimension(560,540)); // (width, height)**

**panel.setBackground(new Color(255,229,219)); // (r,g,b)**

**window.add(panel);**

**/\*\* The title of the project: Wedding Planner is displayed using a simple label \*/**

**JLabel titleLabel = new JLabel("Wedding Planner");**

**titleLabel.setFont(new Font("Curlz MT", Font.BOLD, 45));**

**titleLabel.setForeground(new Color(225,0,0));**

**titleLabel.setBounds(30, 20, 350, 55); // (x, y, width, height)**

**panel.add(titleLabel);**

**/\*\* This button is for a returning user who wants to log into the application \*/**

**loginButton.setFont(new Font("Arial", Font.BOLD, 15));**

**loginButton.setBounds(410, 37, 80, 30);**

**loginButton.setForeground(new Color(225,0,0));**

**loginButton.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**panel.add(loginButton);**

**loginButton.addActionListener(new ActionListener(){ // When this button is clicked log in window should be opened.**

**public void actionPerformed(ActionEvent arg0){**

**window.setVisible(false);**

**new LoginClass();**

**}**

**});**

**/\*\* This button is for a new user who wants to sign up for the application \*/**

**signupButton.setFont(new Font("Arial", Font.BOLD, 15));**

**signupButton.setBounds(500, 37, 80, 30);**

**signupButton.setForeground(new Color(225,0,0));**

**signupButton.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**panel.add(signupButton);**

**signupButton.addActionListener(new ActionListener(){ // When this button is clicked sign up window should be opened.**

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

**window.setVisible(false);**

**new SignUpClass();**

**}**

**});**

**/\*\* A cover image is loaded and displayed on the front page of the application \*/**

**ImageIcon frontPageImageIcon = new ImageIcon(FrontPage.class.getResource("cover-frontpage.jpg"));**

**JLabel frontPageImageLabel = new JLabel(frontPageImageIcon);**

**frontPageImageLabel.setBounds(25, 100, 550, 325);**

**frontPageImageLabel.setBorder(BorderFactory.createLineBorder(Color.BLACK, 15));**

**panel.add(frontPageImageLabel);**

**/\*\* This button is for planning the wedding which is disabled initially until user logs in \*/**

**planButton.setEnabled(false);**

**planButton.setBounds(125, 475, 150, 30);**

**planButton.setToolTipText("First log into the application.");**

**panel.add(planButton);**

**planButton.addActionListener(new ActionListener(){ // On click of this button planner page should be displayed.**

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

**window.setVisible(false);**

**new PlannerPage();**

**}**

**});**

**/\*\* This button is for sending invitations which is also disabled initially until user logs in \*/**

**invitationsButton.setEnabled(false);**

**invitationsButton.setBounds(335, 475, 150, 30);**

**invitationsButton.setToolTipText("First log into the application.");**

**panel.add(invitationsButton);**

**invitationsButton.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent e){ // This button should open the invitation page.**

**window.setVisible(false);**

**new InvitationsPage();**

**}**

**});**

**JLabel copyrightLabel = new JLabel("© Wedding Planner - Created by Pooja Warang");**

**copyrightLabel.setBounds(185,550,250,20);**

**copyrightLabel.setForeground(new Color(60,60,60));**

**panel.add(copyrightLabel);**

**window.setVisible(true);**

**Class.forName("com.mysql.cj.jdbc.Driver");**

**Connection connection;**

**try{**

**connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\",\"root\",\"root");**

**java.sql.DatabaseMetaData dbmd = connection.getMetaData();**

**ResultSet rs=dbmd.getTables(null, null, "usertable", null);**

**connection.close();**

**}**

**catch (SQLException e1){**

**System.out.println("Tables already exists");**

**System.exit(0);**

**}**

**}**

* **}**

**InvitationPage.java**

**/\*\* This page will allow the user to send e-invitations of his/her wedding to the guests. This feature is completely independent of the**

**\* planning process. So the user can simultaneously plan and invite the guests.**

**\* @author Vipul Patil \*/**

**package com.proj;**

**import java.awt.Color;**

**import java.awt.Font;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.awt.event.WindowAdapter;**

**import java.awt.event.WindowEvent;**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.ResultSet;**

**import java.sql.Statement;**

**import java.util.Properties;**

**import javax.mail.AuthenticationFailedException;**

**import javax.mail.Authenticator;**

**import javax.mail.Message;**

**import javax.mail.MessagingException;**

**import javax.mail.PasswordAuthentication;**

**import javax.mail.Session;**

**import javax.mail.Transport;**

**import javax.mail.internet.InternetAddress;**

**import javax.mail.internet.MimeMessage;**

**import javax.swing.JButton;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import javax.swing.JPanel;**

**import javax.swing.JPasswordField;**

**import javax.swing.JScrollPane;**

**import javax.swing.JTextField;**

**import javax.swing.JTextPane;**

**public class InvitationsPage{**

**JFrame invitationWindow = new JFrame("Wedding Planner - Invitation");**

**final JPanel invitationPanel = new JPanel();**

**JLabel lineLabel = new JLabel("Enter the following details for sending email invitation...");**

**JLabel emailAddressLabel = new JLabel("Your email address: ");**

**final JTextField emailAddressTextField = new JTextField();**

**JLabel passwordLabel = new JLabel("Your account's password: ");**

**final JPasswordField passwordTextField = new JPasswordField();**

**JLabel toLabel = new JLabel("Recipients: ");**

**final JTextField toTextField = new JTextField();**

**JLabel subjectLabel = new JLabel("Subject: ");**

**final JTextField subjectTextField = new JTextField();**

**JLabel bodyLabel = new JLabel("Message: ");**

**final JTextPane bodyTextPane = new JTextPane();**

**/\*\* Constructor which will open the invitations page for the user. \*/**

**public InvitationsPage(){**

**invitationWindow.setSize(600, 600);**

**invitationWindow.setResizable(false);**

**invitationPanel.setSize(600, 600);**

**invitationPanel.setLayout(null);**

**invitationPanel.setBackground(new Color(255,229,219));**

**lineLabel.setBounds(10, 10, 500, 20);**

**lineLabel.setFont(new Font("comic sans ms",Font.BOLD ,15));**

**invitationPanel.add(lineLabel);**

**emailAddressLabel.setBounds(10, 40, 200, 20); // Simple label asking user to enter his/her accounts email address.**

**emailAddressLabel.setFont(new Font("comic sans ms",Font.BOLD ,13));**

**invitationPanel.add(emailAddressLabel);**

**emailAddressTextField.setBounds(180, 40, 250, 24); // User will enter his/her email address here.**

**invitationPanel.add(emailAddressTextField);**

**passwordLabel.setBounds(10, 70, 200, 20); // Label asking password of that account.**

**passwordLabel.setFont(new Font("comic sans ms",Font.BOLD ,13));**

**invitationPanel.add(passwordLabel);**

**passwordTextField.setBounds(180, 70, 250, 24); // User will enter the password of his/her email account here.**

**invitationPanel.add(passwordTextField);**

**toLabel.setBounds(10, 100, 200, 20); // Label asking user to enter the recipients email addresses**

**toLabel.setFont(new Font("comic sans ms",Font.BOLD ,13));**

**invitationPanel.add(toLabel);**

**toTextField.setBounds(180, 100, 250, 24); // User will enter email address of the guests. Multiple addresses can be entered using a ',' separator.**

**invitationPanel.add(toTextField);**

**toTextField.setToolTipText("Enter multiple addresses separated by a comma.");**

**subjectLabel.setBounds(10, 130, 200, 20); // Label asking user to enter some subject for the email.**

**subjectLabel.setFont(new Font("comic sans ms",Font.BOLD ,13));**

**invitationPanel.add(subjectLabel);**

**subjectTextField.setBounds(180, 130, 250, 24); // User will enter the subject of the invitation email here.**

**invitationPanel.add(subjectTextField);**

**bodyLabel.setBounds(10, 160, 200, 20); // Label asking user to enter main content, body of the email.**

**bodyLabel.setFont(new Font("comic sans ms",Font.BOLD ,13));**

**invitationPanel.add(bodyLabel);**

**JScrollPane scroll = new JScrollPane(bodyTextPane, JScrollPane.VERTICAL\_SCROLLBAR\_AS\_NEEDED, JScrollPane.HORIZONTAL\_SCROLLBAR\_NEVER);**

**scroll.setBounds(180, 160, 400, 300); // User will frame and construct the invitation email in this space.**

**invitationPanel.add(scroll);**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root"); // Get a connection**

**Statement myStmt = myConn.createStatement(); // Create a statement**

**// This SQL query will retrieve all the details of this page, if the user has saved them previously.**

**ResultSet retrieveRs = myStmt.executeQuery("Select EmailAddress, Password, EmailTo, EmailSubject, EmailBody from invitationtable where USER\_ID = " + LoginClass.loginUserID);**

**if(retrieveRs.next()){**

**emailAddressTextField.setText(retrieveRs.getString("EmailAddress").toString());**

**passwordTextField.setText(retrieveRs.getString("Password").toString());**

**toTextField.setText(retrieveRs.getString("EmailTo").toString());**

**subjectTextField.setText(retrieveRs.getString("EmailSubject").toString());**

**bodyTextPane.setText(retrieveRs.getString("EmailBody").toString());**

**}**

**else{ // If not saved anything previously then leave all the fields blank.**

**emailAddressTextField.setText("");**

**passwordTextField.setText("");**

**toTextField.setText("");**

**subjectTextField.setText("");**

**bodyTextPane.setText("");**

**}**

**myConn.close();**

**}**

**catch(Exception e){**

**e.printStackTrace();**

**}**

**// Before closing the invitation window warn the user and ask whether he/she has to save all this data which would be helpful in the future.**

**invitationWindow.addWindowListener(new WindowAdapter(){**

**public void windowClosing(WindowEvent e){**

**int selectedOption = JOptionPane.showConfirmDialog(invitationWindow, "Save any changes from this page?", "Confirm Save", JOptionPane.YES\_NO\_OPTION, JOptionPane.PLAIN\_MESSAGE);**

**if(selectedOption == JOptionPane.YES\_OPTION){**

**saveInvitationData(); // if user says 'Yes' then save all the data in the database table.**

**invitationWindow.setVisible(false);**

**FrontPage.window.setVisible(true);**

**}**

**else if(selectedOption == JOptionPane.NO\_OPTION){**

**invitationWindow.setVisible(false); // If user selects 'No' then simple close the invitation window and open the Home page of the application.**

**FrontPage.window.setVisible(true);**

**}**

**else**

**invitationWindow.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);**

**}**

**});**

**// Separate button for allowing user to save the data in the database. (Same functionality like ctrl+S)**

**JButton saveButton = new JButton("Save Template");**

**saveButton.setBounds(50, 500, 120, 30);**

**invitationPanel.add(saveButton);**

**saveButton.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent arg0){**

**saveInvitationData();**

**JOptionPane.showMessageDialog(invitationPanel, "Email template saved.", "Invitation - Template", JOptionPane.INFORMATION\_MESSAGE);**

**}**

**});**

**// This button click will send the email to all the recipients.**

**JButton sendButton = new JButton("Send Email");**

**sendButton.setBounds(250, 500, 100, 30);**

**invitationPanel.add(sendButton);**

**sendButton.addActionListener(new ActionListener(){**

**@SuppressWarnings("deprecation")**

**public void actionPerformed(ActionEvent arg0){**

**if(emailAddressTextField.getText().length() == 0){ // Validation 1: Email address not entered.**

**JOptionPane.showMessageDialog(invitationPanel, "Enter your email address.", "Invitation - Email", JOptionPane.ERROR\_MESSAGE);**

**emailAddressTextField.requestFocus();**

**}**

**else if(!(emailAddressTextField.getText().toLowerCase().contains("@yahoo.com") || emailAddressTextField.getText().toLowerCase().contains("@gmail.com"))) {**

**JOptionPane.showMessageDialog(invitationPanel, "Only Gmail and Yahoo accounts addresses.", "Invitation - Email", JOptionPane.ERROR\_MESSAGE);**

**emailAddressTextField.requestFocus();**

**}**

**else if(passwordTextField.getText().length() == 0){ // Validation 2: Password not entered.**

**JOptionPane.showMessageDialog(invitationPanel, "Enter your password.", "Invitation - Password", JOptionPane.ERROR\_MESSAGE);**

**passwordTextField.requestFocus();**

**}**

**else if(toTextField.getText().length() == 0){ // Validation 3: Recipients email address is not entered.**

**JOptionPane.showMessageDialog(invitationPanel, "Enter recipients email address.", "Invitation - Recipient", JOptionPane.ERROR\_MESSAGE);**

**toTextField.requestFocus();**

**}**

**else if(!(toTextField.getText().toLowerCase().contains("@yahoo.com") || toTextField.getText().toLowerCase().contains("@gmail.com"))) {**

**JOptionPane.showMessageDialog(invitationPanel, "Only Gmail and Yahoo accounts addresses.", "Invitation - Recipient", JOptionPane.ERROR\_MESSAGE);**

**toTextField.requestFocus();**

**}**

**else if(subjectTextField.getText().length() == 0){ // Validation 4: Subject field is empty.**

**JOptionPane.showMessageDialog(invitationPanel, "Enter some subject.", "Invitation - Subject", JOptionPane.ERROR\_MESSAGE);**

**subjectTextField.requestFocus();**

**}**

**else if(bodyTextPane.getText().length() == 0){ // Validation 5: Empty email, no body constructed.**

**JOptionPane.showMessageDialog(invitationPanel, "Enter some content.", "Invitation - Body", JOptionPane.ERROR\_MESSAGE);**

**bodyTextPane.requestFocus();**

**}**

**else{ // Set all the properties and put all the required fields.**

**Properties properties = new Properties();**

**properties.put("mail.smtp.auth", "true");**

**properties.put("mail.smtp.starttls.enable", "true");**

**if(emailAddressTextField.getText().toLowerCase().contains("@gmail.com"))**

**properties.put("mail.smtp.host", "smtp.gmail.com");**

**else if(emailAddressTextField.getText().toLowerCase().contains("@yahoo.com"))**

**properties.put("mail.smtp.host", "smtp.mail.yahoo.com");**

**properties.put("mail.smtp.port", "587");**

**final String username = emailAddressTextField.getText();**

**final String password = passwordTextField.getText();**

**String toEmail = toTextField.getText();**

**final String toNewEmail = toEmail.replaceAll("\\s+", "");**

**final String subject = subjectTextField.getText();**

**final String message = bodyTextPane.getText();**

**// Confirm the account's email address and password is valid.**

**try{**

**Session session = Session.getInstance(properties, new Authenticator() {**

**protected PasswordAuthentication getPasswordAuthentication(){**

**return new PasswordAuthentication(username, password);**

**}**

**});**

**Message textMessage = new MimeMessage(session); // Construct the whole email**

**textMessage.setFrom(new InternetAddress(username));**

**textMessage.setRecipients(Message.RecipientType.TO, InternetAddress.parse(toNewEmail));**

**textMessage.setSubject(subject);**

**textMessage.setText(message);**

**Transport.send(textMessage); // send the email too all the recipients.**

**JOptionPane.showMessageDialog(invitationPanel, "Invitation sent successfully.", "Invitation - Success", JOptionPane.INFORMATION\_MESSAGE);**

**}**

**catch (AuthenticationFailedException e) {**

**JOptionPane.showMessageDialog(invitationPanel, "Username or password invalid.", "Invitation - Invalid", JOptionPane.ERROR\_MESSAGE);**

**emailAddressTextField.requestFocus();**

**}**

**catch(MessagingException e){**

**JOptionPane.showMessageDialog(invitationPanel, "Couldn't connect to the server. Check the internet connection.", "Connection problem", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**}**

**});**

**JButton backButton = new JButton("Back"); // Simply close the invitation page without saving any data.**

**backButton.setBounds(430, 500, 100, 30);**

**invitationPanel.add(backButton);**

**backButton.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent arg0){**

**int selectedOption = JOptionPane.showConfirmDialog(invitationWindow, "Save any changes from this page?", "Confirm Save", JOptionPane.YES\_NO\_OPTION, JOptionPane.PLAIN\_MESSAGE);**

**if(selectedOption == JOptionPane.YES\_OPTION){**

**saveInvitationData(); // if user says 'Yes' then save all the data in the database table.**

**invitationWindow.setVisible(false);**

**FrontPage.window.setVisible(true);**

**}**

**else if(selectedOption == JOptionPane.NO\_OPTION){**

**invitationWindow.setVisible(false); // If user selects 'No' then simple close the invitation window and open the Home page of the application.**

**FrontPage.window.setVisible(true);**

**}**

**else**

**invitationWindow.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);**

**}**

**});**

**invitationWindow.add(invitationPanel);**

**invitationWindow.setVisible(true);**

**}**

**/\*\* This method will help user by saving all the data on invitation page in the database table. \*/**

**@SuppressWarnings("deprecation")**

**public void saveInvitationData(){**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**/\*\* Execute SQL Query -> Select from Table: To check whether User ID already exists in the table or not \*/**

**ResultSet userRsCheck = myStmt.executeQuery("Select USER\_ID from invitationtable where USER\_ID = " + LoginClass.loginUserID);**

**if(userRsCheck.next()) // If the user already exists then simply update the data.**

**myStmt.executeUpdate("Update invitationtable set USER\_ID = " + LoginClass.loginUserID + ", EmailAddress = '" + emailAddressTextField.getText() + "', Password = '" + passwordTextField.getText() + "', EmailTo = '" + toTextField.getText() + "', EmailSubject = '" + subjectTextField.getText() + "', EmailBody = '" + bodyTextPane.getText() + "' where USER\_ID = " + LoginClass.loginUserID);**

**else // If the user has not save any data previously then insert it for the first time.**

**myStmt.executeUpdate("Insert into invitationtable(USER\_ID, EmailAddress, Password, EmailTo, EmailSubject, EmailBody) values(" + LoginClass.loginUserID + ", '" + emailAddressTextField.getText().toString() + "', '" + passwordTextField.getText().toString() + "', '" + toTextField.getText().toString() + "', '" + subjectTextField.getText().toString() + "', '" + bodyTextPane.getText().toString() + "')");**

**myConn.close();**

**}**

**catch(Exception e){**

**e.printStackTrace();**

**JOptionPane.showMessageDialog(invitationPanel, "Database problem. Data not saved.", "Invitation - Database", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**}**

**LoginClass.java**

**/\*\* This class consists of design and logic about how the user will log into the Wedding Planner Application. User will be asked to enter his**

**\* or her user name and password. After validating those details the user will be allowed to log into the application.**

**\* @author Vipul Patil \*/**

**package com.proj;**

**import java.awt.Color;**

**import java.awt.Cursor;**

**import java.awt.Dimension;**

**import java.awt.Font;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.awt.event.KeyEvent;**

**import java.awt.event.KeyListener;**

**import java.awt.event.WindowAdapter;**

**import java.awt.event.WindowEvent;**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.ResultSet;**

**import java.sql.Statement;**

**import javax.swing.JButton;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import javax.swing.JPanel;**

**import javax.swing.JPasswordField;**

**import javax.swing.JTextField;**

**public class LoginClass{**

**public static int loginUserID;**

**final JFrame loginWindow = new JFrame();**

**final JPanel loginPanel = new JPanel();**

**JLabel nameLabel = new JLabel("Enter username: ");**

**final JTextField nameTextField = new JTextField();**

**JLabel passwordLabel = new JLabel("Enter password: ");**

**final JPasswordField passwordTextField = new JPasswordField();**

**JButton okButton = new JButton("OK");**

**/\*\* Constructor which generates a separate frame for user log in. Elements like labels, textfields and buttons are added to this frame. \*/**

**public LoginClass(){**

**loginWindow.setTitle("Wedding Planner - Log In"); // frame for user log in.**

**loginWindow.setVisible(true);**

**loginWindow.setSize(300, 300);**

**loginWindow.setResizable(false);**

**loginWindow.addWindowListener(new WindowAdapter(){**

**public void windowClosing(WindowEvent e){**

**loginWindow.setVisible(false);**

**FrontPage.window.setVisible(true);**

**}**

**});**

**loginPanel.setLayout(null); // panel on top of that frame.**

**loginPanel.setPreferredSize(new Dimension(250, 250));**

**loginPanel.setBackground(new Color(255,229,219));**

**loginWindow.add(loginPanel);**

**nameLabel.setFont(new Font("comic sans ms",Font.BOLD ,15)); // simple label asking user to enter his or her user name.**

**nameLabel.setBounds(10, 10, 200, 25);**

**loginPanel.add(nameLabel);**

**nameTextField.setFont(new Font("comic sans ms",Font.BOLD ,15)); // element where the user will enter the user name.**

**nameTextField.setBounds(10, 40, 250, 25);**

**loginPanel.add(nameTextField);**

**nameTextField.addKeyListener(new KeyListener() {**

**public void keyTyped(KeyEvent e){}**

**public void keyReleased(KeyEvent e){}**

**public void keyPressed(KeyEvent e){**

**if(e.getKeyCode() == KeyEvent.VK\_ENTER){**

**login();**

**}**

**}**

**});**

**passwordLabel.setFont(new Font("comic sans ms",Font.BOLD ,15)); // simple label asking user to enter his or her password.**

**passwordLabel.setBounds(10, 80, 200, 25);**

**loginPanel.add(passwordLabel);**

**passwordTextField.setBounds(10, 110, 250, 25); //space for user to enter his or her hidden password.**

**loginPanel.add(passwordTextField);**

**passwordTextField.addKeyListener(new KeyListener() {**

**public void keyTyped(KeyEvent e){}**

**public void keyReleased(KeyEvent e){}**

**public void keyPressed(KeyEvent e){**

**if(e.getKeyCode() == KeyEvent.VK\_ENTER){**

**login();**

**}**

**}**

**});**

**okButton.setBounds(20, 170, 100, 30); // On click of this OK button user will be validated and logged in accordingly.**

**loginPanel.add(okButton);**

**okButton.addActionListener(new ActionListener(){**

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

**login();**

**}**

**});**

**/\*\* This button should close the log in window and open the Home page of the application without logging in the user. \*/**

**JButton cancelButton = new JButton("Cancel");**

**cancelButton.setBounds(150, 170, 100, 30);**

**loginPanel.add(cancelButton);**

**cancelButton.addActionListener(new ActionListener(){**

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

**loginWindow.setVisible(false);**

**FrontPage.window.setVisible(true);**

**}**

**});**

**}**

**@SuppressWarnings("deprecation")**

**public void login()**

**{**

**if(nameTextField.getText().length() == 0){ // Validation 1: user did not enter the user name.**

**JOptionPane.showMessageDialog(loginPanel, "Enter the user name.", "Log In - User Name", JOptionPane.ERROR\_MESSAGE);**

**nameTextField.requestFocus();**

**}**

**else if(passwordTextField.getText().length() == 0){ // Validation 2: user did not enter the password**

**JOptionPane.showMessageDialog(loginPanel, "Enter the password.", "Log In - Password", JOptionPane.ERROR\_MESSAGE);**

**passwordTextField.requestFocus();**

**}**

**else{**

**// Validation 3: Checking whether user name and password are correct.**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**/\*\* Execute SQL Query -> Select from Table: To check whether same user name already exists in the table \*/**

**ResultSet myRsCheck = myStmt.executeQuery("Select USER\_ID, USER\_NAME from usertable where USER\_NAME = '" + nameTextField.getText().toString() + "' AND USER\_PASSWORD = '" + passwordTextField.getText().toString() +"'");**

**if(myRsCheck.next()){**

**/\*\* Entered credentials are valid and the user should be logged in and directed to the home page of the application. \*/**

**loginUserID = myRsCheck.getInt("USER\_ID");**

**nameTextField.setText("");**

**passwordTextField.setText("");**

**loginWindow.setVisible(false);**

**FrontPage.window.setVisible(true);**

**FrontPage.panel.remove(FrontPage.loginButton);**

**FrontPage.panel.remove(FrontPage.signupButton);**

**FrontPage.planButton.setEnabled(true);**

**FrontPage.invitationsButton.setEnabled(true);**

**FrontPage.planButton.setToolTipText(null);**

**FrontPage.invitationsButton.setToolTipText(null);**

**/\*\* On successfully logging in, instead of log in and sign up buttons log out button should appear on Home page. \*/**

**FrontPage.logoutButton.setBounds(410, 37, 160, 30);**

**FrontPage.logoutButton.setFont(new Font("Arial", Font.BOLD, 15));**

**FrontPage.logoutButton.setForeground(new Color(225,0,0));**

**FrontPage.logoutButton.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**FrontPage.panel.add(FrontPage.logoutButton);**

**FrontPage.logoutButton.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent arg0){**

**FrontPage.panel.remove(FrontPage.logoutButton);**

**FrontPage.panel.add(FrontPage.loginButton);**

**FrontPage.panel.add(FrontPage.signupButton);**

**FrontPage.planButton.setEnabled(false);**

**FrontPage.invitationsButton.setEnabled(false);**

**FrontPage.planButton.setToolTipText("First log into the application.");**

**FrontPage.invitationsButton.setToolTipText("First log into the application.");**

**}**

**});**

**FrontPage.window.setVisible(true);**

**}**

**else**

**// Validation 4: Entered credentials are invalid and the user should be asked to re-enter the details.**

**JOptionPane.showMessageDialog(loginPanel, "Invalid credentials, enter again.", "Login - Invalid", JOptionPane.ERROR\_MESSAGE);**

**myConn.close();**

**}**

**catch(Exception e1){**

**e1.printStackTrace();**

**}**

**}**

**}**

**}**

**PlannerPage.java**

**/\*\* This class displays the Planner Page of Wedding Planner Application. It mainly consists selection of location, bride's dress, groom's dress**

**\* decoration theme and catering. Based on all these selections of the user, total estimation amount of the wedding will also be displayed on**

**\* this page so that according to his/her budget they can manage the selection.**

**\* @author Vipul Patil \*/**

**package com.proj;**

**import java.awt.Color;**

**import java.awt.Cursor;**

**import java.awt.Dimension;**

**import java.awt.FlowLayout;**

**import java.awt.Font;**

**import java.awt.GridLayout;**

**import java.awt.Image;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.awt.event.KeyEvent;**

**import java.awt.event.KeyListener;**

**import java.awt.event.MouseAdapter;**

**import java.awt.event.MouseEvent;**

**import java.awt.event.WindowAdapter;**

**import java.awt.event.WindowEvent;**

**import java.sql.\*;**

**import java.util.ArrayList;**

**import javax.swing.BorderFactory;**

**import javax.swing.ButtonGroup;**

**import javax.swing.ImageIcon;**

**import javax.swing.JButton;**

**import javax.swing.JCheckBox;**

**import javax.swing.JComboBox;**

**import javax.swing.JComponent;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import javax.swing.JPanel;**

**import javax.swing.JRadioButton;**

**import javax.swing.JScrollPane;**

**import javax.swing.JTextField;**

**public class PlannerPage{**

**JLabel decorationImageLabel1 = new JLabel(), decorationImageLabel2 = new JLabel(), decorationImageLabel3 = new JLabel(), decorationImageLabel4 = new JLabel();**

**int totalEstimationAmount = 0, locationCost = 0, brideDressCost = 0, groomDressCost = 0, decorationThemeCost = 0, cateringCost = 0;**

**JLabel totalEstimationLabel = new JLabel();**

**final JFrame plannerWindow = new JFrame("Wedding Planner - Plan");**

**public int decorationID = 1;**

**String locations[] = {"Select Wedding Location","Santa Clara","Udaipur","Fremont","Mountain View","Sunnyvale","San Francisco","San Jose"};**

**final JComboBox<?> locationComboBox = new JComboBox<>(locations);**

**String brideDresses[] = {"Select Bride's Dress Designer", "Maggie Sottero","Deepika Padukone","Mark Zunino","Alvina Valenta","Augusta Jones", "Danielle Caprese", "Pnina Tornai", "Pallas Couture", "Ian Stuart", "Lazaro", "Austin Scarlett"};**

**final JComboBox<?> brideDressesComboBox = new JComboBox<>(brideDresses);**

**String groomDresses[] = {"Select Groom's Dress Designer", "Pronto Uomo", "Ranbir Kapoor", "Joseph Abboud", "Calvin Klein", "Armani", "Mon Cheri", "Rina Di Montella", "Paul Smith", "Vera Wang", "Alexander McQueen", "Joseph & Feiss"};**

**final JComboBox<?> groomDressesComboBox = new JComboBox<>(groomDresses);**

**final ImageIcon locationImageIcon = new ImageIcon();**

**final JLabel locationImageLabel = new JLabel();**

**final ImageIcon brideDressesImageIcon = new ImageIcon();**

**final JLabel brideDressesImageLabel = new JLabel();**

**final ImageIcon groomDressesImageIcon = new ImageIcon();**

**final JLabel groomDressesImageLabel = new JLabel();**

**final JRadioButton aquaRadioButton = new JRadioButton("Aqua");**

**final JRadioButton goldenRadioButton = new JRadioButton("Golden");**

**final JRadioButton redRadioButton = new JRadioButton("Red");**

**final JRadioButton lavenderRadioButton = new JRadioButton("Lavender");**

**final JButton estimationButton = new JButton("Calculate");**

**String locPhotoPath="", bridePhotoPath="", groomPhotoPath="";**

**public boolean buttonPressed = false;**

**int people = 0;**

**public PlannerPage(){**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**/\*\* Checking whether current user has already saved any data or not. \*/**

**ResultSet retrieveRs = myStmt.executeQuery("Select USER\_ID, LocationID, BrideDressID, GroomDressID, LocationPhoto, BrideDressPhoto, GroomDressPhoto, DecorationThemeID, TotalEstimation from savedatatable where USER\_ID = " + LoginClass.loginUserID);**

**int locId = 0, brideId = 0, groomId = 0;**

**if(retrieveRs == null){ // User has not saved any data on planner page previously.**

**locationComboBox.setSelectedIndex(0);**

**brideDressesComboBox.setSelectedIndex(0);**

**groomDressesComboBox.setSelectedIndex(0);**

**locationCost = 0; brideDressCost = 0; groomDressCost = 0; decorationThemeCost = 0; cateringCost = 0;**

**totalEstimationAmount = 0;**

**aquaRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**goldenRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**redRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**lavenderRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**}**

**else if(retrieveRs.next()){ // User have saved some data previously on planner page, so retrieve it and display on this page..**

**locId = retrieveRs.getInt("LocationID");**

**locationComboBox.setSelectedIndex(locId);**

**brideId = retrieveRs.getInt("BrideDressID");**

**brideDressesComboBox.setSelectedIndex(brideId);**

**groomId = retrieveRs.getInt("GroomDressID");**

**groomDressesComboBox.setSelectedIndex(groomId);**

**if(locId != 0){**

**Image img1 = new ImageIcon(PlannerPage.class.getResource(retrieveRs.getString("LocationPhoto").toString())).getImage();**

**locationImageIcon.setImage(img1);**

**locationImageLabel.setIcon(locationImageIcon);**

**locationImageLabel.setBorder(BorderFactory.createLineBorder(Color.black, 1));**

**}**

**if(brideId != 0){**

**Image img2 = new ImageIcon(PlannerPage.class.getResource(retrieveRs.getString("BrideDressPhoto").toString())).getImage();**

**brideDressesImageIcon.setImage(img2);**

**brideDressesImageLabel.setIcon(brideDressesImageIcon);**

**brideDressesImageLabel.setBorder(BorderFactory.createLineBorder(Color.black, 1));**

**}**

**if(groomId != 0){**

**Image img3 = new ImageIcon(PlannerPage.class.getResource(retrieveRs.getString("GroomDressPhoto").toString())).getImage();**

**groomDressesImageIcon.setImage(img3);**

**groomDressesImageLabel.setIcon(groomDressesImageIcon);**

**groomDressesImageLabel.setBorder(BorderFactory.createLineBorder(Color.black, 1));**

**}**

**int decorationID = retrieveRs.getInt("DecorationThemeID");**

**switch(decorationID){**

**case 1: aquaRadioButton.setSelected(true);**

**aquaRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**decorationImageLabel1.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**decorationThemeCost = 4500;**

**break;**

**case 2: goldenRadioButton.setSelected(true);**

**goldenRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**decorationImageLabel2.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**decorationThemeCost = 5250;**

**break;**

**case 3: redRadioButton.setSelected(true);**

**redRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**decorationImageLabel3.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**decorationThemeCost = 2750;**

**break;**

**case 4: lavenderRadioButton.setSelected(true);**

**lavenderRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**decorationImageLabel4.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**decorationThemeCost = 2000;**

**break;**

**default:aquaRadioButton.setSelected(true);**

**decorationImageLabel1.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**}**

**totalEstimationLabel.setText("");**

**}**

**myConn.close();**

**}**

**catch(SQLException e){**

**e.printStackTrace();**

**}**

**plannerWindow.setSize(627,730); // This frame is for opening new window for Planner page.**

**plannerWindow.setResizable(false);**

**JPanel plannerPanel = new JPanel(); // This panel will be placed on top of this frame.**

**plannerPanel.setLayout(null);**

**plannerPanel.setPreferredSize(new Dimension(600, 1000));**

**plannerPanel.setBackground(new Color(255,229,219));**

**plannerWindow.add(plannerPanel);**

**totalEstimationLabel.setBounds(350, 945, 225, 40); // At the bottom of the page, total estimation of whole wedding will be displayed using this label.**

**totalEstimationLabel.setFont(new Font("Times New Roman", Font.BOLD, 20));**

**locationComboBox.setBounds(10, 10, 175, 20); // User can select various locations according to his/her choice and budget.**

**plannerPanel.add(locationComboBox);**

**locationImageLabel.setBounds(10, 40, 550, 250); //Based on user's location selection, image of that particular location will be displayed here**

**locationImageLabel.setBorder(BorderFactory.createLineBorder(Color.black, 2));**

**plannerPanel.add(locationImageLabel);**

**locationComboBox.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent arg0){**

**// Retrieve data from locationtable depending upon the value selected in LocationComboBox**

**if(locationComboBox.getSelectedIndex() != 0){**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**/\*\* Execute SQL Query -> Select from Table: To retrieve image path and cost of the location from locationtable \*/**

**ResultSet locationRs = myStmt.executeQuery("Select LocationPhoto, LocationCost from locationtable where LocationID = " + locationComboBox.getSelectedIndex());**

**if(locationRs.next()){**

**locPhotoPath = locationRs.getString("LocationPhoto");**

**locationCost = locationRs.getInt("LocationCost");**

**}**

**myConn.close();**

**}**

**catch(Exception e1){**

**e1.printStackTrace();**

**}**

**// load the image of that location using that retrieved image path.**

**Image img = new ImageIcon(PlannerPage.class.getResource(locPhotoPath)).getImage();**

**locationImageIcon.setImage(img);**

**locationImageLabel.setIcon(locationImageIcon);**

**locationImageLabel.setBorder(BorderFactory.createLineBorder(Color.black, 1));**

**}**

**}**

**});**

**brideDressesComboBox.setBounds(10, 300, 200, 20); // User can select a bride's dress from a variety of collection from here.**

**plannerPanel.add(brideDressesComboBox);**

**brideDressesImageLabel.setBounds(10, 330, 270, 400); // Based on user's selection, bride's dress photo will be displayed here.**

**brideDressesImageLabel.setBorder(BorderFactory.createLineBorder(Color.black, 2));**

**plannerPanel.add(brideDressesImageLabel);**

**brideDressesComboBox.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent arg0){**

**// Retrieve data from bridedresstable depending upon the value selected in brideDressesComboBox**

**if(brideDressesComboBox.getSelectedIndex() != 0){**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**/\*\* Execute SQL Query -> Select from Table: To retrieve image path and cost of the bride's dress from bridedresstable \*/**

**ResultSet brideDressRs = myStmt.executeQuery("Select BrideDressPhoto, BrideDressCost from bridedresstable where BrideDressID = " + brideDressesComboBox.getSelectedIndex());**

**if(brideDressRs.next()){**

**bridePhotoPath = brideDressRs.getString("BrideDressPhoto");**

**brideDressCost = brideDressRs.getInt("BrideDressCost");**

**}**

**myConn.close();**

**}**

**catch(Exception e1){**

**e1.printStackTrace();**

**}**

**// load the image of that bride's dress using that retrieved image path.**

**Image img = new ImageIcon(PlannerPage.class.getResource(bridePhotoPath)).getImage();**

**brideDressesImageIcon.setImage(img);**

**brideDressesImageLabel.setIcon(brideDressesImageIcon);**

**brideDressesImageLabel.setBorder(BorderFactory.createLineBorder(Color.black, 1));**

**}**

**}**

**});**

**groomDressesComboBox.setBounds(290, 300, 210, 20); // User can select a groom's dress from a variety of collection from here.**

**plannerPanel.add(groomDressesComboBox);**

**groomDressesImageLabel.setBounds(290, 330, 270, 400); // Based on user's selection, groom's dress photo will be displayed here.**

**groomDressesImageLabel.setBorder(BorderFactory.createLineBorder(Color.black, 2));**

**plannerPanel.add(groomDressesImageLabel);**

**groomDressesComboBox.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent arg0){**

**// Retrieve data from groomdresstable depending upon the value selected in groomDressesComboBox**

**if(groomDressesComboBox.getSelectedIndex() != 0){**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**/\*\* Execute SQL Query -> Select from Table: To retrieve image path and cost of the groom's dress from groomdresstable \*/**

**ResultSet groomDressRs = myStmt.executeQuery("Select GroomDressPhoto, GroomDressCost from groomdresstable where GroomDressID = " + groomDressesComboBox.getSelectedIndex());**

**if(groomDressRs.next()){**

**groomPhotoPath = groomDressRs.getString("GroomDressPhoto");**

**groomDressCost = groomDressRs.getInt("GroomDressCost");**

**}**

**myConn.close();**

**}**

**catch(Exception e1){**

**e1.printStackTrace();**

**}**

**// load the image of that bride's dress using that retrieved image path.**

**Image img = new ImageIcon(PlannerPage.class.getResource(groomPhotoPath)).getImage();**

**groomDressesImageIcon.setImage(img);**

**groomDressesImageLabel.setIcon(groomDressesImageIcon);**

**groomDressesImageLabel.setBorder(BorderFactory.createLineBorder(Color.black, 1));**

**}**

**}**

**});**

**JLabel decorationLabel = new JLabel("Select the decoration theme for wedding location...");**

**decorationLabel.setBounds(10, 740, 300, 20);**

**plannerPanel.add(decorationLabel);**

**JPanel decorationPhotoPanel = new JPanel(); // Separate panel to display various decoration color theme images.**

**decorationPhotoPanel.setBounds(10, 760, 550, 180);**

**decorationPhotoPanel.setBorder(BorderFactory.createLineBorder(Color.BLACK, 2));**

**decorationPhotoPanel.setBackground(new Color(255,229,219));**

**decorationPhotoPanel.setLayout(new FlowLayout());**

**// Loading first image theme - Aqua**

**ImageIcon decorationImageIcon1 = new ImageIcon(PlannerPage.class.getResource("/DecorationIcon\_Aqua.jpg"));**

**decorationImageLabel1.setIcon(decorationImageIcon1);**

**decorationImageLabel1.setPreferredSize(new Dimension(130, 135));**

**decorationImageLabel1.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**decorationImageLabel1.setToolTipText("Click to enlarge");**

**decorationPhotoPanel.add(decorationImageLabel1);**

**// On clicking on the first image, it should be enlarged.**

**decorationImageLabel1.addMouseListener(new MouseAdapter(){**

**public void mouseClicked(MouseEvent e){**

**decorationThemeCost = 4500;**

**decorationID = 1;**

**newFrame("Decoration Theme - Aqua", "/Decoration\_Aqua.jpg"); // this will enlarge the Aqua-image and open into new frame.**

**// Borders around the images are managed according to the user's selection**

**decorationImageLabel1.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**decorationImageLabel2.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel3.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel4.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**// Also the radio buttons below th images are managed according to the user's selection.**

**aquaRadioButton.setSelected(true);**

**aquaRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**goldenRadioButton.setSelected(false);**

**goldenRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**redRadioButton.setSelected(false);**

**redRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**lavenderRadioButton.setSelected(false);**

**lavenderRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**}**

**});**

**ImageIcon decorationImageIcon2 = new ImageIcon(PlannerPage.class.getResource("/DecorationIcon\_Golden.jpg"));**

**decorationImageLabel2.setIcon(decorationImageIcon2);**

**decorationImageLabel2.setPreferredSize(new Dimension(130, 135));**

**decorationImageLabel2.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**decorationImageLabel2.setToolTipText("Click to enlarge");**

**decorationPhotoPanel.add(decorationImageLabel2);**

**// On clicking the second image, it should be enlarged.**

**decorationImageLabel2.addMouseListener(new MouseAdapter(){**

**public void mouseClicked(MouseEvent e){**

**decorationThemeCost = 5250;**

**decorationID = 2;**

**newFrame("Decoration Theme - Golden", "/Decoration\_Golden.jpg"); // this will enlarge the Golden-image and open into new frame.**

**// Managing the borders of all the images according to the user's selection.**

**decorationImageLabel1.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel2.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**decorationImageLabel3.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel4.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**// Managing all the radio buttons placed below the images, based on the user's selection.**

**aquaRadioButton.setSelected(false);**

**aquaRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**goldenRadioButton.setSelected(true);**

**goldenRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**redRadioButton.setSelected(false);**

**redRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**lavenderRadioButton.setSelected(false);**

**lavenderRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**}**

**});**

**ImageIcon decorationImageIcon3 = new ImageIcon(PlannerPage.class.getResource("/DecorationIcon\_Red.jpg"));**

**decorationImageLabel3.setIcon(decorationImageIcon3);**

**decorationImageLabel3.setPreferredSize(new Dimension(130, 135));**

**decorationImageLabel3.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**decorationImageLabel3.setToolTipText("Click to enlarge");**

**decorationPhotoPanel.add(decorationImageLabel3);**

**// On clicking the third image, it should be enlarged.**

**decorationImageLabel3.addMouseListener(new MouseAdapter(){**

**public void mouseClicked(MouseEvent e){**

**decorationThemeCost = 2750;**

**decorationID = 3;**

**newFrame("Decoration Theme - Red", "/Decoration\_Red.jpg"); // this will enlarge the Red-image and open into new frame.**

**// Managing all the borders of the images based on the user's selection.**

**decorationImageLabel1.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel2.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel3.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**decorationImageLabel4.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**// Also setting correct radio buttons based on the user's selection.**

**aquaRadioButton.setSelected(false);**

**aquaRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**goldenRadioButton.setSelected(false);**

**goldenRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**redRadioButton.setSelected(true);**

**redRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**lavenderRadioButton.setSelected(false);**

**lavenderRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**}**

**});**

**ImageIcon decorationImageIcon4 = new ImageIcon(PlannerPage.class.getResource("/DecorationIcon\_Lavender.jpg"));**

**decorationImageLabel4.setIcon(decorationImageIcon4);**

**decorationImageLabel4.setPreferredSize(new Dimension(130, 135));**

**decorationImageLabel4.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**decorationImageLabel4.setToolTipText("Click to enlarge");**

**decorationPhotoPanel.add(decorationImageLabel4);**

**// On clicking on this fourth image, it should be enlarged.**

**decorationImageLabel4.addMouseListener(new MouseAdapter(){**

**public void mouseClicked(MouseEvent e){**

**decorationThemeCost = 2000;**

**decorationID = 4;**

**newFrame("Decoration Theme - Lavender", "/Decoration\_Lavender.jpg"); // The Lavender image will be enlarged and opened in a new frame.**

**// Managing borders of all the images according to the user's selection.**

**decorationImageLabel1.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel2.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel3.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel4.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**// Managing all the radio buttons based on the user's selection.**

**aquaRadioButton.setSelected(false);**

**aquaRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**goldenRadioButton.setSelected(false);**

**goldenRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**redRadioButton.setSelected(false);**

**redRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**lavenderRadioButton.setSelected(true);**

**lavenderRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**}**

**});**

**// Defining all the radio buttons which are placed under the decoration images on the decorationPanel.**

**aquaRadioButton.setPreferredSize(new Dimension(130, 20));**

**aquaRadioButton.setBackground(new Color(255,229,219));**

**aquaRadioButton.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**aquaRadioButton.setToolTipText("Click on the image to enlarge it");**

**goldenRadioButton.setPreferredSize(new Dimension(130, 20));**

**goldenRadioButton.setBackground(new Color(255,229,219));**

**goldenRadioButton.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**goldenRadioButton.setToolTipText("Click on the image to enlarge it");**

**redRadioButton.setPreferredSize(new Dimension(130, 20));**

**redRadioButton.setBackground(new Color(255,229,219));**

**redRadioButton.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**redRadioButton.setToolTipText("Click on the image to enlarge it");**

**lavenderRadioButton.setPreferredSize(new Dimension(130, 20));**

**lavenderRadioButton.setBackground(new Color(255,229,219));**

**lavenderRadioButton.setCursor(new Cursor(Cursor.HAND\_CURSOR));**

**lavenderRadioButton.setToolTipText("Click on the image to enlarge it");**

**//Group the radio buttons.**

**ButtonGroup group = new ButtonGroup();**

**group.add(aquaRadioButton);**

**group.add(goldenRadioButton);**

**group.add(redRadioButton);**

**group.add(lavenderRadioButton);**

**//Register a listener for all the radio buttons. Based on user's selection manage the borders and and all the radio buttons.**

**aquaRadioButton.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent arg0){**

**decorationThemeCost = 4500;**

**decorationID = 1;**

**decorationImageLabel1.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**decorationImageLabel2.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel3.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel3.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**aquaRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**goldenRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**redRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**lavenderRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**}**

**});**

**goldenRadioButton.addActionListener(new ActionListener(){**

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

**decorationThemeCost = 5250;**

**decorationID = 2;**

**decorationImageLabel2.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**decorationImageLabel1.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel3.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel4.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**aquaRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**goldenRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**redRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**lavenderRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**}**

**});**

**redRadioButton.addActionListener(new ActionListener(){**

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

**decorationThemeCost = 2750;**

**decorationID = 3;**

**decorationImageLabel3.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**decorationImageLabel1.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel2.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel4.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**aquaRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**goldenRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**redRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**lavenderRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**}**

**});**

**lavenderRadioButton.addActionListener(new ActionListener(){**

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

**decorationThemeCost = 2000;**

**decorationID = 4;**

**decorationImageLabel4.setBorder(BorderFactory.createLineBorder(Color.BLACK, 7));**

**decorationImageLabel1.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel2.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**decorationImageLabel3.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));**

**aquaRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**goldenRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**redRadioButton.setFont(new Font("Comic Sans MS", Font.PLAIN, 13));**

**lavenderRadioButton.setFont(new Font("Comic Sans MS", Font.BOLD, 13));**

**}**

**});**

**decorationPhotoPanel.add(aquaRadioButton);**

**decorationPhotoPanel.add(goldenRadioButton);**

**decorationPhotoPanel.add(redRadioButton);**

**decorationPhotoPanel.add(lavenderRadioButton);**

**plannerPanel.add(decorationPhotoPanel);**

**JButton cateringButton = new JButton("Catering"); // A button which will help the user decide the food menu for the wedding.**

**cateringButton.setBounds(10, 950, 100, 30);**

**cateringButton.addActionListener(new ActionListener(){**

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

**cateringEstimation();**

**plannerWindow.setVisible(false);**

**}**

**});**

**plannerPanel.add(cateringButton);**

**estimationButton.setBounds(120, 950, 100, 30); // This button will let the user to know total estimation of his/her wedding.**

**estimationButton.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent arg0){**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**// Retrieve the cost of all the selection of the user from the database.**

**ResultSet locCostRs = myStmt.executeQuery("Select LocationCost from locationtable where LocationID = " + locationComboBox.getSelectedIndex());**

**if(locCostRs.next())**

**locationCost = locCostRs.getInt("LocationCost");**

**ResultSet brideCostRs = myStmt.executeQuery("Select BrideDressCost from bridedresstable where BrideDressID = " + brideDressesComboBox.getSelectedIndex());**

**if(brideCostRs.next())**

**brideDressCost = brideCostRs.getInt("BrideDressCost");**

**ResultSet groomCostRs = myStmt.executeQuery("Select GroomDressCost from groomdresstable where GroomDressID = " + groomDressesComboBox.getSelectedIndex());**

**if(groomCostRs.next())**

**groomDressCost = groomCostRs.getInt("GroomDressCost");**

**ResultSet cateringRs = myStmt.executeQuery("Select CateringCost from savedatatable where USER\_ID = " + LoginClass.loginUserID);**

**if(cateringRs.next())**

**cateringCost = cateringRs.getInt("CateringCost");**

**myConn.close();**

**}**

**catch(Exception e){**

**e.printStackTrace();**

**}**

**// Add the all the retrieved values and then display it on the totalEstimationLabel.**

**totalEstimationAmount = locationCost + brideDressCost + groomDressCost + decorationThemeCost + cateringCost;**

**totalEstimationLabel.setText("Total Estimation: $" + totalEstimationAmount);**

**}**

**});**

**plannerPanel.add(estimationButton);**

**plannerPanel.add(totalEstimationLabel);**

**JButton backButton = new JButton("Back"); // A button which will help the user decide the food menu for the wedding.**

**backButton.setBounds(230, 950, 100, 30);**

**backButton.addActionListener(new ActionListener(){**

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

**saveData();**

**}**

**});**

**plannerPanel.add(backButton);**

**// Add a scroll bar to the panel to allow the user to scroll vertically.**

**JScrollPane scroll = new JScrollPane(plannerPanel, JScrollPane.VERTICAL\_SCROLLBAR\_AS\_NEEDED, JScrollPane.HORIZONTAL\_SCROLLBAR\_NEVER);**

**plannerWindow.add(scroll);**

**// Ask user before closing that whether he/she has to save the changes made on this page.**

**plannerWindow.addWindowListener(new WindowAdapter(){**

**public void windowClosing(WindowEvent e){**

**saveData();**

**}**

**});**

**plannerWindow.setVisible(true);**

**}**

**public void saveData(){**

**int selectedOption = JOptionPane.showConfirmDialog(plannerWindow, "Save any changes from this page?", "Confirm Save", JOptionPane.YES\_NO\_OPTION, JOptionPane.PLAIN\_MESSAGE);**

**if(selectedOption == JOptionPane.YES\_OPTION){**

**int locCost = 0, brideCost = 0, groomCost = 0;**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**int userID = LoginClass.loginUserID;**

**int locID = locationComboBox.getSelectedIndex();**

**ResultSet locPathRs = myStmt.executeQuery("Select LocationCost, LocationPhoto from locationtable where LocationID = " + locationComboBox.getSelectedIndex());**

**if(locPathRs.next()){**

**locPhotoPath = locPathRs.getString("LocationPhoto");**

**locCost = locPathRs.getInt("LocationCost");**

**}**

**int brideID = brideDressesComboBox.getSelectedIndex();**

**ResultSet bridePathRs = myStmt.executeQuery("Select BrideDressCost, BrideDressPhoto from bridedresstable where BrideDressID = " + brideDressesComboBox.getSelectedIndex());**

**if(bridePathRs.next()){**

**bridePhotoPath = bridePathRs.getString("BrideDressPhoto");**

**brideCost = bridePathRs.getInt("BrideDressCost");**

**}**

**int groomID = groomDressesComboBox.getSelectedIndex();**

**ResultSet groomPathRs = myStmt.executeQuery("Select GroomDressCost, GroomDressPhoto from groomdresstable where GroomDressID = " + groomDressesComboBox.getSelectedIndex());**

**if(groomPathRs.next()){**

**groomPhotoPath = groomPathRs.getString("GroomDressPhoto");**

**groomCost = groomPathRs.getInt("GroomDressCost");**

**}**

**if(aquaRadioButton.isSelected())**

**decorationID = 1;**

**else if(goldenRadioButton.isSelected())**

**decorationID = 2;**

**else if(redRadioButton.isSelected())**

**decorationID = 3;**

**else if(lavenderRadioButton.isSelected())**

**decorationID = 4;**

**else**

**decorationThemeCost = 0;**

**totalEstimationAmount = locCost + brideCost + groomCost + decorationThemeCost + cateringCost;**

**ResultSet userRs = myStmt.executeQuery("Select USER\_ID from savedatatable where USER\_ID = " + userID);**

**if(userRs.next()) // If user data already exists, then simply update the previous entry**

**myStmt.executeUpdate("Update savedatatable SET LocationID = " + locID + ", BrideDressID = " + brideID +", GroomDressID = " + groomID + ", LocationPhoto = '" + locPhotoPath + "', BrideDressPhoto = '" + bridePhotoPath + "', GroomDressPhoto = '" + groomPhotoPath + "', DecorationThemeID = " + decorationID + ", TotalEstimation = " + totalEstimationAmount + " where USER\_ID = " + userID);**

**else // If the user is new and there is no user entry, then insert it into the table**

**myStmt.executeUpdate("Insert into savedatatable(USER\_ID, LocationID, BrideDressID, GroomDressID, LocationPhoto, BrideDressPhoto, GroomDressPhoto, DecorationThemeID, TotalEstimation, TotalGuests, CateringCost) Values(" + userID + ", " + locID + ", " + brideID +", " +groomID + ", '" + locPhotoPath + "', '" + bridePhotoPath + "', '" + groomPhotoPath + "', " + decorationID + ", " + totalEstimationAmount + ", " + people + ", " + cateringCost + ")");**

**myConn.close();**

**}**

**catch(Exception e1){**

**e1.printStackTrace();**

**}**

**FrontPage.window.setVisible(true);**

**plannerWindow.setVisible(false);**

**}**

**else if(selectedOption == JOptionPane.NO\_OPTION){**

**FrontPage.window.setVisible(true);**

**plannerWindow.setVisible(false);**

**}**

**else**

**plannerWindow.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);**

**}**

**/\*\* This method is used when the user clicks on the decoration theme image. At that time a new window must appear with the enlarged version**

**\* of that particular image.**

**\* @param frameTitle To set the title of the new frame window.**

**\* @param iconPath To set the enlarged image using icon path.**

**\*/**

**public void newFrame(String frameTitle, String iconPath){**

**JLabel bigImageLabel = new JLabel();**

**// load the bigger image**

**Image img = new ImageIcon(PlannerPage.class.getResource(iconPath)).getImage();**

**ImageIcon bigImageIcon = new ImageIcon(img);**

**bigImageLabel.setIcon(bigImageIcon);**

**bigImageLabel.setBounds(10, 10, 325, 500);**

**bigImageLabel.setBorder(BorderFactory.createLineBorder(Color.BLACK, 3));**

**final JComponent[] inputs = new JComponent[] {bigImageLabel};**

**JOptionPane.showMessageDialog(plannerWindow, inputs, frameTitle, JOptionPane.PLAIN\_MESSAGE);**

**}**

**/\*\* This method is triggered when the user clicks on the catering button. It manages all the details related to catering menu and catering total cost. \*/**

**public void cateringEstimation(){**

**final JFrame cateringFrame = new JFrame("Wedding Planner - Catering"); // new window will be displayed for food menu selection.**

**cateringFrame.setVisible(true);**

**cateringFrame.setSize(627, 600);**

**cateringFrame.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);**

**final JPanel cateringPanel = new JPanel();**

**cateringPanel.setLayout(null);**

**cateringPanel.setBackground(new Color(255,229,219));**

**JLabel dishesLabel = new JLabel("Select the dishes according to your choice (Prices per person)");**

**dishesLabel.setBounds(10, 10, 450, 15);**

**dishesLabel.setFont(new Font("Times New Roman", Font.BOLD, 15));**

**cateringPanel.add(dishesLabel);**

**JPanel dishesPanel = new JPanel();**

**dishesPanel.setLayout(new GridLayout(15, 2, 60, 10));**

**dishesPanel.setBounds(10, 35, 590, 400);**

**dishesPanel.setBackground(new Color(255,229,219));**

**// Check boxes array, for assisting user to select from variety of food items according to his/her choice and budget.**

**final ArrayList<JCheckBox> checkboxes = new ArrayList<JCheckBox>();**

**int i = 0;**

**String dish = "";**

**for(i = 0; i < 30; ++i){**

**JCheckBox itemCheckBox = new JCheckBox();**

**checkboxes.add(itemCheckBox);**

**checkboxes.get(i).setSelected(false);**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root"); // Get a connection**

**Statement myStmt = myConn.createStatement(); // Create a statement**

**// Retrieving dish name from cateringtable from the database and then assigning it to each checkbox.**

**ResultSet dishRs = myStmt.executeQuery("Select dishName from cateringtable where dishID = " + (i+1));**

**if(dishRs.next()){**

**dish = dishRs.getString("dishName");**

**itemCheckBox.setText(" " + dish);**

**itemCheckBox.setBackground(Color.WHITE);**

**}**

**myConn.close();**

**}**

**catch(Exception e1){**

**e1.printStackTrace();**

**}**

**dishesPanel.add(itemCheckBox);**

**}**

**cateringPanel.add(dishesPanel);**

**JLabel personsLabel = new JLabel("Rough number of guests "); // Simple label asking user to enter rough number of guests for the wedding.**

**personsLabel.setBounds(10, 470, 175, 15);**

**personsLabel.setFont(new Font("Times New Roman", Font.BOLD, 15));**

**cateringPanel.add(personsLabel);**

**final JTextField personsTextField = new JTextField(); // User can enter rough number of guests in order to calculate the catering cost for the wedding.**

**personsTextField.setBounds(185, 468, 90, 20);**

**personsTextField.setFont(new Font("Times New Roman", Font.BOLD, 15));**

**personsTextField.addKeyListener(new KeyListener(){**

**public void keyTyped(KeyEvent e){**

**char vchar = e.getKeyChar(); // Validation: Only digits must be entered in the personsTextField.**

**if(!(Character.isDigit(vchar)) || (vchar == KeyEvent.VK\_BACK\_SPACE) || (vchar == KeyEvent.VK\_DELETE))**

**e.consume();**

**}**

**public void keyReleased(KeyEvent arg0) {}**

**public void keyPressed(KeyEvent arg0) {}**

**});**

**cateringPanel.add(personsTextField);**

**final JLabel amountLabel = new JLabel("Total catering amount: "); // Label displaying the total catering amount calculated.**

**amountLabel.setBounds(338, 468, 300, 15);**

**amountLabel.setFont(new Font("Times New Roman", Font.BOLD, 15));**

**cateringPanel.add(amountLabel);**

**/\*\* Initially, if the user has saved any catering data in the database then first retrieve it and display it during loading the page. So that the user can edit or make changes to previous selection. \*/**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**int dishNumber;**

**ResultSet dishesRs = myStmt.executeQuery("Select USER\_ID, dishID from savecatering where USER\_ID = " + LoginClass.loginUserID);**

**while(dishesRs.next()){ // If user has saved the selection of food items then retrieve those dishId's and check those particular boxes.**

**dishNumber = dishesRs.getInt("dishID");**

**checkboxes.get(dishNumber-1).setSelected(true);**

**}**

**// Also retrieve rough number of guests and previously calculated total catering amount for remembering the user to make any changes accordingly.**

**ResultSet cateringRs = myStmt.executeQuery("Select TotalGuests, CateringCost from savedatatable where USER\_ID = " + LoginClass.loginUserID);**

**if(cateringRs.next()){**

**personsTextField.setText(cateringRs.getInt("TotalGuests") + "");**

**amountLabel.setText("Total catering amount: $" + cateringRs.getInt("CateringCost"));**

**}**

**else{**

**personsTextField.setText(people + "");**

**amountLabel.setText("Total catering amount: $" + cateringCost);**

**}**

**myConn.close();**

**}**

**catch(Exception e){**

**e.printStackTrace();**

**}**

**/\*\* calculateButton should calculate the total catering cost for the user, depending on the food items selected and number of guests coming for the wedding. \*/**

**JButton calculateButton = new JButton("Calculate Catering Amount");**

**calculateButton.setBounds(10, 515, 188, 30);**

**cateringPanel.add(calculateButton);**

**calculateButton.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent arg0){**

**buttonPressed = true;**

**int i, totalAmount = 0, dishesAmount = 0, numberOfGuests;**

**boolean itemSelected = false;**

**for(i = 0; i < 30; ++i){**

**if(checkboxes.get(i).isSelected()){**

**itemSelected = true;**

**break;**

**}**

**else**

**itemSelected = false;**

**}**

**if(itemSelected == false) // Validation 1: If none food items are selected then tell the user to select some food items.**

**JOptionPane.showMessageDialog(cateringPanel, "Select some food items.", "Catering - Food Items", JOptionPane.INFORMATION\_MESSAGE);**

**else if(personsTextField.getText().length() == 0 || (Integer.parseInt(personsTextField.getText())<50)) // Validation 2: If number of guests is not entered or is less than 50 then tell user accordingly to correct the entry.**

**JOptionPane.showMessageDialog(cateringPanel, "Enter valid number of guests (minimum 50).", "Catering - Guests", JOptionPane.INFORMATION\_MESSAGE);**

**else{**

**people = Integer.parseInt(personsTextField.getText());**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**numberOfGuests = Integer.parseInt(personsTextField.getText());**

**for(i = 0; i < 30; ++i){ // Retrieve the cost of each dish which has been selected by the user.**

**if(checkboxes.get(i).isSelected()){**

**ResultSet amountRs = myStmt.executeQuery("Select dishCost from cateringtable where dishID = " + (i+1));**

**if(amountRs.next()){**

**dishesAmount = dishesAmount + amountRs.getInt("dishCost"); // Add all the dishes cost and store it in dishesAmount.**

**}**

**}**

**}**

**totalAmount = numberOfGuests \* dishesAmount; // Now calculate total catering estimation by multiplying dishesAmount by number of guests.**

**amountLabel.setText("Total catering amount: $" + totalAmount); // display this total Amount in the amountLabel.**

**cateringCost = totalAmount;**

**myConn.close();**

**}**

**catch(Exception e1){**

**e1.printStackTrace();**

**}**

**}**

**}**

**});**

**// This button will add the total catering cost to the planner window's total wedding's estimation.**

**JButton addToPlannerButton = new JButton("Save and add to Planner");**

**addToPlannerButton.setBounds(213, 515, 185, 30);**

**cateringPanel.add(addToPlannerButton);**

**addToPlannerButton.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent arg0){**

**if(buttonPressed){**

**int i = 0;**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**// Save all the dishes selection of the user in the savecateringtable**

**ResultSet checkRs = myStmt.executeQuery("Select USER\_ID from savecatering where USER\_ID = " + LoginClass.loginUserID);**

**if(checkRs.next()){**

**myStmt.executeUpdate("Delete from savecatering where USER\_ID = " + LoginClass.loginUserID);**

**for(i = 0; i < 30; ++i){**

**if(checkboxes.get(i).isSelected())**

**myStmt.executeUpdate("Insert into savecatering(USER\_ID, dishID) Values (" + LoginClass.loginUserID + ", " + (i+1) + ")");**

**}**

**}**

**else{**

**for(i = 0; i < 30; ++i){**

**if(checkboxes.get(i).isSelected())**

**myStmt.executeUpdate("Insert into savecatering(USER\_ID, dishID) Values(" + LoginClass.loginUserID + ", " + (i+1) + ")");**

**}**

**}**

**// Also save the number of guests and total catering amount**

**ResultSet checkDataRs = myStmt.executeQuery("Select USER\_ID from savedatatable where USER\_ID = " + LoginClass.loginUserID);**

**if(checkDataRs.next())**

**myStmt.executeUpdate("Update savedatatable set TotalGuests = " + personsTextField.getText() + ", CateringCost = " + cateringCost + " where USER\_ID = " + LoginClass.loginUserID);**

**else**

**people = Integer.parseInt(personsTextField.getText());**

**myConn.close();**

**}**

**catch(Exception e){**

**e.printStackTrace();**

**}**

**totalEstimationLabel.setText("");**

**cateringFrame.setVisible(false);**

**plannerWindow.setVisible(true);**

**buttonPressed = false;**

**}**

**else**

**JOptionPane.showMessageDialog(cateringPanel, "First calculate the catering amount.", "Catering - Estimation", JOptionPane.WARNING\_MESSAGE);**

**}**

**});**

**// This button will simply close the catering window and open the planner window without saving any changes on the catering page.**

**JButton cancelButton = new JButton("Cancel");**

**cancelButton.setBounds(410, 515, 185, 30);**

**cateringPanel.add(cancelButton);**

**cancelButton.addActionListener(new ActionListener(){**

**public void actionPerformed(ActionEvent arg0){**

**cateringFrame.setVisible(false);**

**plannerWindow.setVisible(true);**

**buttonPressed = false;**

**}**

**});**

**cateringFrame.add(cateringPanel);**

**}**

**}**

**SignUpClass.java**

**/\*\* This class consists of design and logic about how a new user will register for Wedding Planner Application. User's user name and password**

**\* will be stored in the database so that these details will be further used at the time of log in.**

**\* @author Vipul Patil \*/**

**package com.proj;**

**import java.awt.Color;**

**import java.awt.Dimension;**

**import java.awt.Font;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.awt.event.KeyEvent;**

**import java.awt.event.KeyListener;**

**import java.awt.event.WindowAdapter;**

**import java.awt.event.WindowEvent;**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.ResultSet;**

**import java.sql.Statement;**

**import javax.swing.JButton;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import javax.swing.JPanel;**

**import javax.swing.JPasswordField;**

**import javax.swing.JTextField;**

**public class SignUpClass{**

**public static boolean status;**

**final JFrame signupWindow = new JFrame();**

**final JPanel signupPanel = new JPanel();**

**JLabel nameLabel = new JLabel("Enter user name: ");**

**final JTextField nameTextField = new JTextField();**

**JLabel passwordLabel = new JLabel("Enter password: ");**

**final JPasswordField passwordTextField = new JPasswordField();**

**JLabel repasswordLabel = new JLabel("Re-enter password: ");**

**final JPasswordField repasswordTextField = new JPasswordField();**

**JButton okButton = new JButton("OK");**

**/\*\* This constructor will generate a separate window which will allow the user to enter details required for registering for this application. \*/**

**public SignUpClass(){**

**signupWindow.setTitle("Wedding Planner - Sign Up"); // frame for sign up.**

**signupWindow.setVisible(true);**

**signupWindow.setSize(300, 300);**

**signupWindow.setResizable(false);**

**signupWindow.addWindowListener(new WindowAdapter(){**

**public void windowClosing(WindowEvent e){**

**signupWindow.setVisible(false);**

**FrontPage.window.setVisible(true);**

**}**

**});**

**signupPanel.setLayout(null); // panel which is placed on top of this frame.**

**signupPanel.setPreferredSize(new Dimension(250, 250));**

**signupPanel.setBackground(new Color(255,229,219));**

**signupWindow.add(signupPanel);**

**nameLabel.setFont(new Font("comic sans ms",Font.BOLD ,15)); // simple label asking a new user to enter it's user name.**

**nameLabel.setBounds(10, 10, 200, 25);**

**signupPanel.add(nameLabel);**

**nameTextField.setFont(new Font("comic sans ms",Font.BOLD ,15)); // user will enter his or her user name here.**

**nameTextField.setBounds(10, 35, 250, 25);**

**signupPanel.add(nameTextField);**

**nameTextField.addKeyListener(new KeyListener() {**

**public void keyTyped(KeyEvent e){}**

**public void keyReleased(KeyEvent e){}**

**public void keyPressed(KeyEvent e){**

**if(e.getKeyCode() == KeyEvent.VK\_ENTER){**

**signup();**

**}**

**}**

**});**

**passwordLabel.setFont(new Font("comic sans ms",Font.BOLD ,15)); // simple label asking a new user to enter it's password.**

**passwordLabel.setBounds(10, 70, 200, 25);**

**signupPanel.add(passwordLabel);**

**passwordTextField.setBounds(10, 95, 250, 25); // user will enter it's password here.**

**signupPanel.add(passwordTextField);**

**passwordTextField.addKeyListener(new KeyListener() {**

**public void keyTyped(KeyEvent e){}**

**public void keyReleased(KeyEvent e){}**

**public void keyPressed(KeyEvent e){**

**if(e.getKeyCode() == KeyEvent.VK\_ENTER){**

**signup();**

**}**

**}**

**});**

**repasswordLabel.setFont(new Font("comic sans ms",Font.BOLD ,15)); // to confirm, user will be asked to re-enter the same password.**

**repasswordLabel.setBounds(10, 130, 200, 25);**

**signupPanel.add(repasswordLabel);**

**repasswordTextField.setBounds(10, 155, 250, 25); // user will enter the same password again.**

**signupPanel.add(repasswordTextField);**

**repasswordTextField.addKeyListener(new KeyListener() {**

**public void keyTyped(KeyEvent e){}**

**public void keyReleased(KeyEvent e){}**

**public void keyPressed(KeyEvent e){**

**if(e.getKeyCode() == KeyEvent.VK\_ENTER){**

**signup();**

**}**

**}**

**});**

**/\*\* After validating all the entered details, user will be either signed up for the application or else will be prompted with appropriate message. \*/**

**okButton.setBounds(20, 200, 100, 30);**

**signupPanel.add(okButton);**

**okButton.addActionListener(new ActionListener(){**

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

**signup();**

**}**

**});**

**/\*\* This button should close the sign up window and open the Home page of the application without registering the user. \*/**

**JButton cancelButton = new JButton("Cancel");**

**cancelButton.setBounds(150, 200, 100, 30);**

**signupPanel.add(cancelButton);**

**cancelButton.addActionListener(new ActionListener(){**

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

**signupWindow.setVisible(false);**

**FrontPage.window.setVisible(true);**

**}**

**});**

**}**

**@SuppressWarnings("deprecation")**

**public void signup(){**

**if(nameTextField.getText().length() == 0){ // Validation 1: User doesn't enters a user name.**

**JOptionPane.showMessageDialog(signupPanel, "Enter user name.", "Sign Up - User Name", JOptionPane.ERROR\_MESSAGE);**

**nameTextField.requestFocus();**

**}**

**else if(passwordTextField.getText().length() == 0){ // Validation 2: User doesn't enters the password.**

**JOptionPane.showMessageDialog(signupPanel, "Enter password.", "Sign Up - Password", JOptionPane.ERROR\_MESSAGE);**

**passwordTextField.requestFocus();**

**}**

**else if(passwordTextField.getText().length() < 8){ // Validation 3: The entered password is less than 8 characters.**

**JOptionPane.showMessageDialog(signupPanel, "Enter strong password. (minimum 8 characters)", "Sign Up - Password", JOptionPane.ERROR\_MESSAGE);**

**passwordTextField.requestFocus();**

**}**

**else if(repasswordTextField.getText().length() == 0){ // Validation 4: User doesn't re-enter the password.**

**JOptionPane.showMessageDialog(signupPanel, "Re-enter the password.", "Sign Up - Password", JOptionPane.ERROR\_MESSAGE);**

**repasswordTextField.requestFocus();**

**}**

**else if(!(passwordTextField.getText().equals(repasswordTextField.getText()))){ // Validation 5: Re-entered password doesn't match the previous entered password.**

**JOptionPane.showMessageDialog(signupPanel, "Password doesn't match, enter again.", "Sign Up - Password", JOptionPane.ERROR\_MESSAGE);**

**repasswordTextField.requestFocus();**

**}**

**else{**

**try{**

**Connection myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/event\\\",\\\"root\\\",\\\"root");**

**Statement myStmt = myConn.createStatement();**

**/\*\* Execute SQL Query -> Select from Table: To check whether same user name already exists in the table \*/**

**ResultSet myRsCheck = myStmt.executeQuery("Select USER\_NAME from usertable where USER\_NAME = '" + nameTextField.getText().toString() + "'");**

**if(!myRsCheck.next()){**

**/\*\* Validation 6: Entered user name is unique so save the details in the database.**

**\* Execute SQL Query -> Insert into Table \*/**

**myStmt.executeUpdate("Insert into usertable(USER\_NAME,USER\_PASSWORD) VALUES('" + nameTextField.getText().toString() + "', '" + passwordTextField.getText().toString() + "')");**

**JOptionPane.showMessageDialog(signupPanel, "Successfully signed up. Thank you.", "Sign Up - Success", JOptionPane.INFORMATION\_MESSAGE);**

**nameTextField.setText("");**

**passwordTextField.setText("");**

**repasswordTextField.setText("");**

**signupWindow.setVisible(false);**

**FrontPage.window.setVisible(true);**

**}**

**else // Entered user name is already taken.**

**JOptionPane.showMessageDialog(signupPanel, "User name already present. Enter some unique user name.", "Sign Up - User Name", JOptionPane.ERROR\_MESSAGE);**

**myConn.close();**

**}**

**catch(Exception e1){**

**e1.printStackTrace();**

**}**

**}**

**}**

**}**