Khwaja Yunus Ali University

Quest for Knowledge

Module Title:
Object Oriented Programming and Modeling

Module Coursework:
Event Management

Course Code: MIS-403(Lab)

Submitted By:

MD. MILTON HOSSAIN

ID: 2015302020001

MD. Tofael Ahmed

ID: 2014302020002

MOST. Morium Khatun

ID:2015202020004
Batch: 5th

3nd Year 7st Semester

Department of MIS

Khwaja Yunus Ali University

Submitted to:

Mohammad Taherul Alam

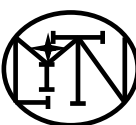Assistance Professor

Department of MIS

Khwaja Yunus Ali University

Submission Date: 20-Sep-2017

Table of Contents

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed overleaf. Please sign below to say that you have read and understand these extracts:

(Signature :) _____Actual number of hours: _____

<div align="center">

Report on
Object Oriented Programming and Modeling
MIS-403(Lab)

</div>

We gratefully acknowledge for the assistance, cooperation, guidance and clarifications provided by (MIS Team) during the development of the Event Management System. Our extreme gratitude to Mohammad Taherul Alam (Assistance Professor) who guided us throughout the project. Without his willing disposition, spirit of accommodation, frankness, timely clarification and above all faith in us, this project could not have been completed in due time. His readiness to discuss all important matters at work deserves special attention. We would also like to thank whole of the teachers and students of the University Cooperation and important support.

This is to certify that the project entitled "Event Management System" has been developed by "Milton Khan(Leader), Tofael Ahmed, Morium Khtun". We have worked on the project Event Management System Software. We have used JAVA, Netbeans IDE and MySQL as Database for the project. Our work is satisfactory.

1. Analysis Stage

INTRODUCTION

Event Management System is being launched because a need for a destination that is beneficial for both Event management institutes and people. With this system, Event management institutes can easily manage different kinds of Functional event such as Marriage event, Birth Day party, Public Function etc. Event Management Institute can manage all the function by using "Event Management System". This System will help them to manage and maintain all functional programs.

PURPOSE

The main aim to develop "**Event Management System"** is to storage event data and maintain functional events efficiently and effectively. It helps to make a good financial statement and also analysis the cost and benefits of events. It will store all event information of a whole year.

Event Management System fulfills the requirements of the institutes to conduct the all functional events program. They do not have to go to any software developer to make a separate system for being able to conduct program event management. They just have to login on the **Event Management System** and enter the all function of **Event Management System** details and the lists of the programs which can appear on the System.

Users can store information about different kinds of events or program by using the **Event Management System.** User can view the result and all information at the same time. Thus the purpose of the software is to provide a system that saves the efforts and time of both the Event Management Functional institutes and the users.

WHAT IS EVENT MANAGEMENT SYSTEM ALL ABOUT?

Event Management System is an offline based application that establishes a connection between the Event Management Institutes and the functional events. Institutes enter on the System can manage storage and control the all events information. The system storages the all functional programs' information and help to view the details all information at same time. User can easily save the information. It also helps to calculate the due automatically. User can search specific program's information by search Program ID. User also can see the **Order Receive Program** details in table and also can see the **Implementation Program Event** Information in **Implementation Program table**. This system also shows the holds events (Which event do not completed). User can update the event by pressing update button and can also delete events by pressing delete button.

There are six tabbed Panel in main Interface.

- ❖ First is **Order Receive Program**,
- ❖ Second is **Implementation Program**,
- ❖ Third is **Hold Programs**,
- ❖ Fourth is **Order Receive Program table**,
- ❖ Fifth is **Implementation Programs table** and
- ❖ Last is **Search Program ID**.

The **search Program ID** tabbed panel includes Delete and Update Function. By using this system user or Institute can take decision how many events will be implemented in a year. They can measure the receive order events with implementation events. **Event Management System** provides the platform but does not directly participate in, nor is it involved in any tests conducted. All events result can see in this system by pressing **Search Program ID Tabbed Panel**. The system entitled "**Event Management System**" is application software, which aims at providing services to the **Event Management Institutes**. It is developed by using J2EE technology and related database (mysql). It is used to develop two tools (TextPad and Netbeans IDE8.2)

PROBLEM DEFINITION

Event Management System is to design for controlling and maintaining different functional events. This system will help users to manage events. Some problems reorganization are given below:

- ➢ Reduce the using papers for storing data about different events.
- ➢ Reduce the time to mange event management.
- ➢ Increase the big electronic data store about event management.
- ➢ First searching desired data about different program events.
- ➢ Show incomplete events.
- ➢ Provide auto Financial Statement.
- ➢ Auto Financial calculation.
- ➢ Shows order receive program storage table at same time.
- ➢ Shows implementation program storage table at same time.
- ➢ Helps to take quick decision by measuring financial statement.

WHERE WE USED THIS SYSTEM

Event Management System is offline event management software. That is used to in different Event Management Organization. It will help to manage different kinds of functional events easily.

SOFTWARE DEVELOPMENT METHODOLOGY

The establishment and use of sound engineering principles in order to obtain economically developed software that is reliable and works efficiently on real machines is called software engineering.

Software engineering is the discipline whose aim is:

1. Production of quality software

2. Software that is delivered on time

3. Cost within the budget

 4. Satisfies all requirements.

Software process is the way in which we produce the software. Apart from hiring smart, knowledgeable engineers and buying the latest development tools, effective software development process is also needed, so that engineers can systematically use the best technical and managerial practices to successfully complete their projects. A software life cycle is the

series of identifiable stages that a software product undergoes during its lifetime .A software lifecycle model is a descriptive and diagrammatic representation of the software life cycle .A life cycle model represents all the activities required to make a software product transit through its lifecycle phases .It also captures the order in which these activities are to be taken.
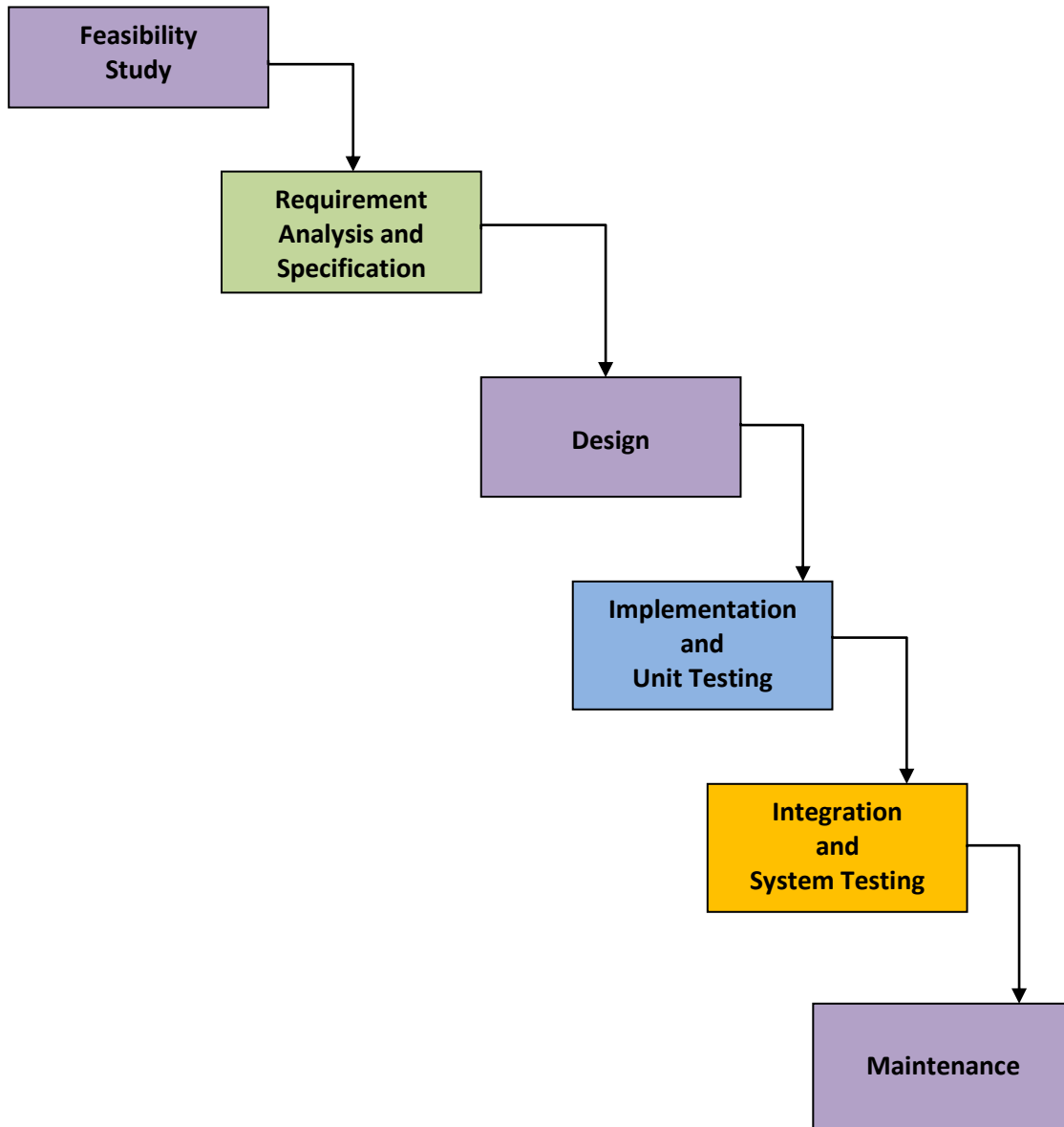
Life Cycle Models

There are various life cycle models to improve the software processes.

- WATERFALL MODEL
- PROTOTYPE MODEL
- ITERATIVE ENHANCEMENT MODEL
- EVOLUTIONARY MODEL
- SPIRAL MODEL

In the project, Waterfall model is followed.

WATERFALL MODEL

```
┌──────────────┐
│ Feasibility  │
│    Study     │
└──────────────┘
        │
        ▼
   ┌──────────────┐
   │ Requirement  │
   │ Analysis and │
   │Specification │
   └──────────────┘
           │
           ▼
      ┌──────────────┐
      │    Design    │
      └──────────────┘
              │
              ▼
         ┌──────────────────┐
         │ Implementation   │
         │       and        │
         │   Unit Testing   │
         └──────────────────┘
                  │
                  ▼
            ┌──────────────────┐
            │   Integration    │
            │       and        │
            │  System Testing  │
            └──────────────────┘
                      │
                      ▼
               ┌──────────────┐
               │  Maintenance │
               └──────────────┘
```

WATERFALL MODEL

This model contains 6 phases:

- **Feasibility study:**

The feasibility study activity involves the analysis of the problem and collection of the relevant information relating to the product. The main aim of the feasibility study is to determine whether it would be financially and technically feasible to develop the product.

- **Requirement analysis and specification:**

The goal of this phase is to understand the exact requirements of the customer and to document them properly.(SRS)

- **Design:**

The goal of this phase is to transform the requirement specification into a structure that is suitable for implementation in some programming language.

- **Implementation and unit testing:**

During this phase the design is implemented. Initially small modules are tested in isolation from rest of the software product.

- **Integration and system testing:**

In this all the modules are integrated and then tested altogether.

- **Operation and maintenance:**

Release of software inaugurates the operation and life cycle phase of the operation. The phases always occur in this order and do not overlap.

PARTS OF THE SOFTWARE

- ❖ **A Database:** I have used mysql database to store data in this system. We can easily search data from the system and we get our desired data quickly.
- ❖ **Automed Event Management System:** here, I have developed a holds event system which will allow showing to current exit program. I have also developed an auto calculation fee management system. It helps to minus from total cost fee and give due fee automatically.
- ❖ **A User Friendly Interface:** I have developed a user friendly interface which will allow the user to use the software effectively and efficiently.
- ❖ **Access the information:** for access the information, I have also developed some views in the interface with Standard Query Language. This will allow user to access the information with just one click.
- ❖ **Security:** I have developed a login form that will help you to access authorized only. So other unauthorized person cannot access this system.
- ❖ **Improve Decision Support:** This system will help manager to take effective decision by measuring or analyzing event information. Whereas, this system will store all events information. It will help to show information firstly.

THE TECHNOLOGIES USED IN DEVELOPED THIS SYSTEM

- ➢ Mysql :
  MYSQL database have used for storing, querying, updating, deleting data in this system.
- ➢ Java:
  Object Oriented Programming Language. It is a platform based programming language.
- ➢ TextPad
  It is used to write code. It is like a tool.
- ➢ Netbeans IDE 8.2:
  It is used to write code. It has also helped to design manually.
- ➢ Microsoft office :
  For drawing ER Diagram, DFD and other documentation, I have used this application.
- ➢ Launch4j:
  I have used Launch4j for making Exe File.
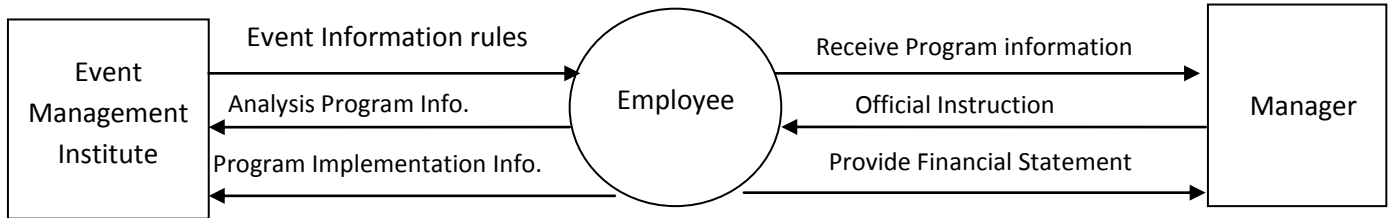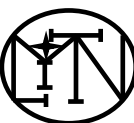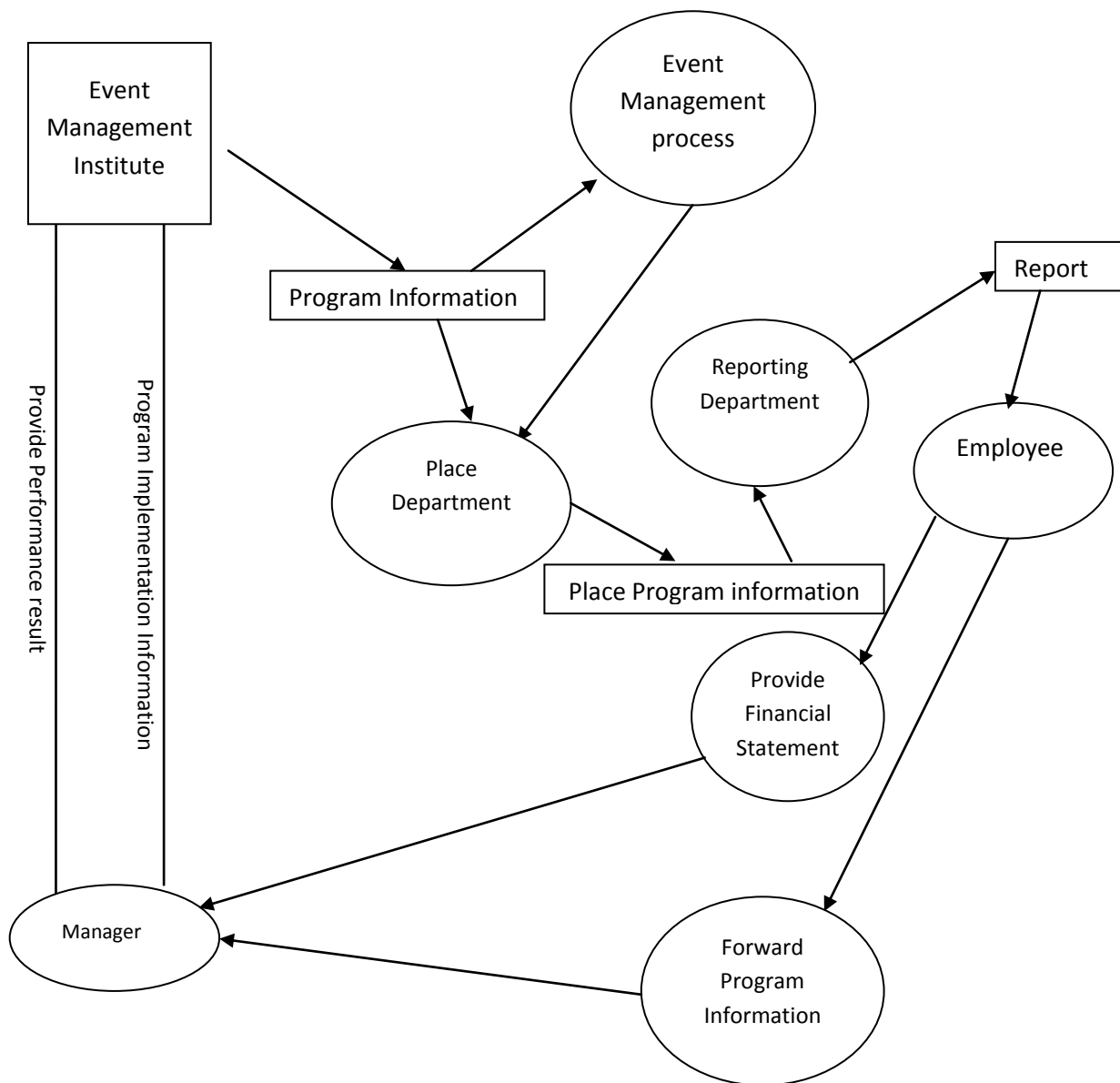
2. Design Stage
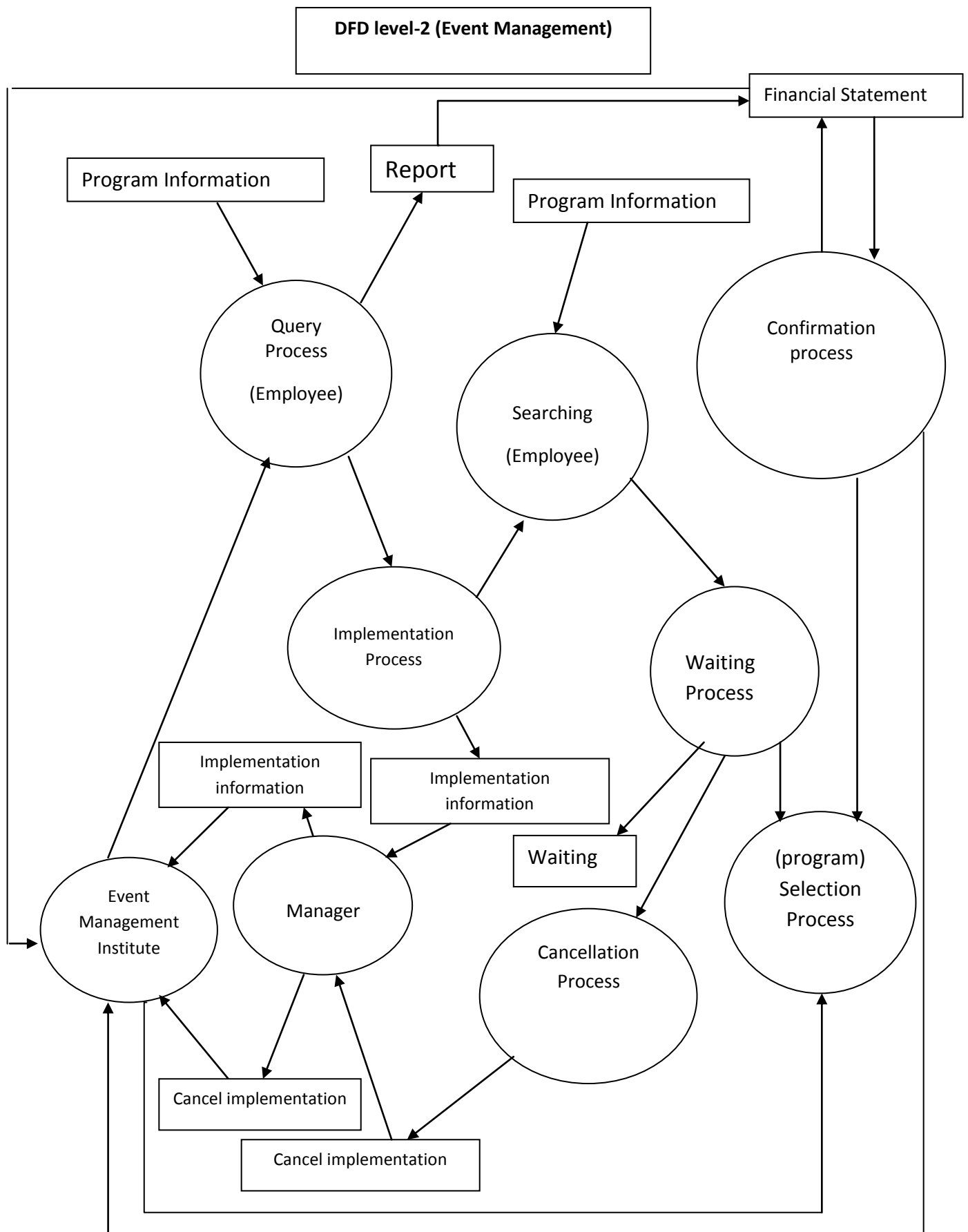
Entity Diagram of Event Management System

Event Management

(1 . *)

Work

Admin

(1 . *)

Employee

(* , *)   **take**   Employee ID(PK)   Employee_Name   (* , *)   **finished**

(* , *)

(* , *)

Receive Program

(* , *)   **incomplete**   (* , *)

Implementation Program

Program ID(PK)   Gmail

P_Name   Event Address

Client Name   Advance Paid

Mobile No   Receive Date

Program ID(PK)   Advance Paid

Program_Name   Paid Fee

Employee_Name   Due Fee

Total Fee   P_FinishedDate

(* , *)   (* , *)

Hold Program

(* , *)   **forward**   (* , *)

Program ID(PK)   P_Name

Client Name   Mobile No

DFD level-0 (Event Management)

Event Management Institute → Employee

- Event Information rules →
- ← Analysis Program Info.
- ← Program Implementation Info.

Employee → Manager

- Receive Program information →
- ← Official Instruction
- Provide Financial Statement →

DFD level-1 (Event Management)

Event Management Institute

Event Management process

Program Information

Report

Reporting Department

Provide Performance result

Program Implementation Information

Place Department

Employee

Place Program information

Provide Financial Statement

Manager

Forward Program Information

DFD level-2 (Event Management)

Financial Statement

Program Information

Report

Program Information

Query Process (Employee)

Searching (Employee)

Confirmation process

Implementation Process

Waiting Process

Implementation information

Implementation information

Waiting

(program) Selection Process

Event Management Institute

Manager

Cancellation Process

Cancel implementation

Cancel implementation

Database (SQL Commands)

Order Receive Event Table:

```
1  Create Table OrderReceiveEvent
2  (
3         ProgramID int(20) not null,
4         ProgramName varchar(50) not null,
5         ClientName varchar(50) not null,
6         MobileNo int(50) not null,
7         Gmail varchar(50) not null,
8         EventAddress varchar(50) not null,
9         AdvancePaid int(10) not null,
10        ReceiveDate varchar(20) not null,
11        Primary Key(OrderID)
12 );
13 |
```

Implementation Event Table:

```
1  Create Table ImplementationEvent
2  (
3         ProgramID int(20) not null,
4         ProgramName varchar(50) not null,
5         EmployeesName varchar(50) not null,
6         TotalFee int(10) not null,
7         AdvancePaid int(10) not null,
8         PaidFee int(10) not null,
9         DueFee int(10) not null,
10        EventAddress varchar (50) not null,
11        ProgramFinishedDate varchar(20) not null,
12        Primary Key(ProgramID)
13 );
14 |
```
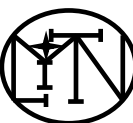
Hold Events Table:

```
1  Create Table HoldsEvent
2  (
3          ProgramID int(20) not null,
4          ProgramName varchar(50) not null,
5          ClientName varchar(50) not null,
6          MobileNo int(50) not null,
7
8  );
```

3. Development State:

THE SOME SOURCE CODE OF EVENT MANAGEMENT:

package eventmanagement;

import java.awt.HeadlessException;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.sql.SQLException;

import javax.swing.*;

import net.proteanit.sql.DbUtils;

import java.awt.event.*;

/**

 *

 * @author MILTON KHAN

 */

public class MainInterfaceEventManagement extends javax.swing.JFrame implements KeyListener,MouseListener

{
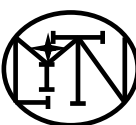
  dbConnection dcc = new dbConnection();

  public MainInterfaceEventManagement()

  {

    this.setTitle("Event Management System");

    initComponents();

```java
        showTableData();

        showOrderTableInfo();

        showImplementationTableInfo();

        txtProgramID.addKeyListener(this);

        txtProgID.addKeyListener(this);

        txtDueFee.addMouseListener(this);

        txtPaidFee.addKeyListener(this);

    }
//*********************************Insert          Order          Receive          Data
Insert****************
public void insertOrderReceiveData()

    {

        try

        {

                dc.statement.executeUpdate("INSERT     INTO     `orderreceiveevent`(`OrderID`,
                `OrderName`,`ClientName`, `MobileNo`, `Gmail`, `EventAddress`, `AdvancePaid`,
                `ReceiveDate`)                                                      VALUES
                ('"+txtID.getText()+"','"+txtName.getText()+"','"+txtClientName.getText()+"','"+tx
                tMobileNo.getText()+"','"+txtGmail.getText()+"','"+txtEventAddress.getText()+"','
                "+txtAdvancePaid.getText()+"','"+txtReceiveDate.getText()+"')");

                dc.statement.executeUpdate("INSERT          INTO          `holdevents`(`OrderID`,
                `OrderName`,                             `ClientName`,`MobileNo`)VALUES
                ('"+txtID.getText()+"','"+txtName.getText()+"','"+txtClientName.getText()+"','"+tx
                tMobileNo.getText()+"')");

                 JOptionPane.showMessageDialog(null,"Data Inserted Successfully");

                 refreshTextFileds();

    }
```
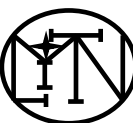
```java
catch(HeadlessException | SQLException ae)

{

    JOptionPane.showMessageDialog(null,ae);

}

showTableData();

showOrderTableInfo();

}
```

//************end   of   Order   Receive   Data insert**************************************

//*********Insert Implementation Program Data*****************************//

```java
public void insertImplementationData()

{

    try

    {

        dc.statement.executeUpdate("INSERT      INTO      `implementationevent`(`ProgramID`,
        `ProgramName`, `EmployeesName`, `TotalFee`,`AdvancePaid`, `PaidFee`, `DueFee`,
        `EventAddress`,          `ProgramFinishedDate`,          `PaidDate`)          VALUES
        ('"+txtProgramID.getText()+"','"+txtProgramName.getText()+"','"+txtEmployeesName.ge
        tText()+"','"+txtTotalFee.getText()+"','"+txtAdvancePaidFee.getText()+"','"+txtPaidFee.ge
        tText()+"','"+txtDueFee.getText()+"','"+txtEventAddress2.getText()+"','"+txtProgramFinis
        hedDate.getText()+"','"+txtPaidDate.getText()+"')");

        dc.statement.executeUpdate("DELETE    FROM    `holdevents`    WHERE    `OrderID`    =
        '"+txtProgramID.getText()+"'");

                JOptionPane.showMessageDialog(null,"Data Inserted Successfully");

                refreshImplementationTextFields();

    }

    catch(HeadlessException | SQLException ae)
```

```java
        {

            JOptionPane.showMessageDialog(null,ae);

        }

        showImplementationTableInfo();

        showTableData();

    }
```

//******************* End of Implementation Program*********************************

//********show Implementation Event Information Table*******************//

```java
public void showImplementationTableInfo()

    {

        try

        {

            String sql = "SELECT * FROM `implementationevent`";

            dc.resultSet = dc.statement.executeQuery(sql);

            tableImplementationInfo.setModel(DbUtils.resultSetToTableModel(dc.resultSet));
```

```java
        }

    catch(SQLException ae)

    {

        JOptionPane.showMessageDialog(null,ae);

    }

  }
//**************end of Implementation Event Information Table*****************

//**************Delete All table Data //*********************************

public void DeleteData()

  {

    try

    {

        dcc.stat.executeUpdate("DELETE FROM `orderreceiveevent` WHERE `OrderID` =
        '"+txtProgID.getText()+"'");

        dcc.stat.executeUpdate("DELETE FROM `implementationevent` WHERE `ProgramID` =
        '"+txtProgID.getText()+"'");

        dcc.stat.executeUpdate("DELETE FROM `holdevents` WHERE `OrderID` =
        '"+txtProgID.getText()+"'");

      JOptionPane.showMessageDialog(null,"Data Delete Successfully");

    }

    catch(Exception ae)

    {

     JOptionPane.showMessageDialog(null,ae);

    }

    showImplementationTableInfo();
```

```java
        showTableData();

        showOrderTableInfo();

    }


//*****************End        code        of        Delete        All        table
data***********************************


//*******************                        Update                all                table
data*******************************************

public void updateData()

    {

        try

        {

            dcc.stat.executeUpdate("UPDATE            `holdevents`            SET            `OrderName`=
'"+txtProgName.getText()+"',`ClientName`=                '"+txtClName.getText()+"',`MobileNo`=
'"+txtMbNo.getText()+"' WHERE `OrderID` = '"+txtProgID.getText()+"'");

            dcc.stat.executeUpdate("UPDATE                    `orderreceiveevent`                    SET
`OrderName`='"+txtProgName.getText()+"',`ClientName`='"+txtClName.getText()+"',`MobileNo`
=        '"+txtMbNo.getText()+"',`Gmail`=        '"+txtEmailAddress.getText()+"',`EventAddress`=
'"+txtEventAdd.getText()+"',`AdvancePaid`=        '"+txtAdvancePd.getText()+"',`ReceiveDate`=
'"+txtRvDate.getText()+"' WHERE `OrderID` = '"+txtProgID.getText()+"'");

            dcc.stat.executeUpdate("UPDATE        `implementationevent`        SET        `ProgramName`=
'"+txtProgName2.getText()+"',`EmployeesName`='"+txtEmployeeName2.getText()+"',`TotalFee`
='"+txtTotalFee2.getText()+"',`AdvancePaid`=
'"+txtAdvancePaid2.getText()+"',`PaidFee`='"+txtPaidFee2.getText()+"',`DueFee`=
'"+txtDueFee2.getText()+"',`EventAddress`='"+txtEA.getText()+"',`ProgramFinishedDate`=
'"+txtProgFDate.getText()+"' WHERE `ProgramID` = '"+txtProgID.getText()+"'");

            JOptionPane.showMessageDialog(null,"Data Updated Successfully");

        }
```
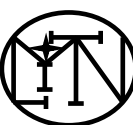
```java
    catch(Exception ae)

    {

      JOptionPane.showMessageDialog(null,ae);

    }

    showImplementationTableInfo();

    showTableData();

    showOrderTableInfo();

  }
//**************End          code          of          Update          all          table
Data*******************************
```

```java
//**********Get                              data                              in
TextFields**********************************************
public void getTextFieldInSearchID()

  {

    try

    {

      String sql = ("SELECT `OrderName`, `ClientName`, `MobileNo`, `Gmail`, `EventAddress`,
`AdvancePaid`,    `ReceiveDate`    FROM    `orderreceiveevent`    WHERE    `OrderID`    =
'"+txtProgID.getText()+"'");

      dcc.rs = dcc.stat.executeQuery(sql);

      while(dcc.rs.next())

      {

        txtProgName.setText(dcc.rs.getString("OrderName"));

        txtClName.setText(dcc.rs.getString("ClientName"));
```
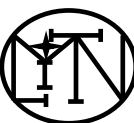
```java
            txtMbNo.setText(dcc.rs.getString("MobileNo"));

            txtEmailAddress.setText(dcc.rs.getString("Gmail"));

            txtEventAdd.setText(dcc.rs.getString("EventAddress"));

            txtAdvancePd.setText(dcc.rs.getString("AdvancePaid"));

            txtAdvancePaid2.setText(dcc.rs.getString("AdvancePaid"));

            txtRvDate.setText(dcc.rs.getString("ReceiveDate"));


        }

    }

    catch(SQLException ae)

    {

      JOptionPane.showMessageDialog(null,ae);

    }

  }
```

//**********end the code of Get data in TextField ************************************

//***************Auto   Calculate
Due***

```java
public void Totaldue()

  {

    String totalFee = txtTotalFee.getText();

    int storageTotalFee = Integer.parseInt(totalFee);


    String txtAdvanceFee = txtAdvancePaidFee.getText();

    int storageTxtAdvanceFee = Integer.parseInt(txtAdvanceFee);
```

```java
    String txtPaidFeeTest = txtPaidFee.getText();

    int storagetxtPaidFee = Integer.parseInt(txtPaidFeeTest);


    int totalPaid = storageTxtAdvanceFee + storagetxtPaidFee;


    int totalDueResult = storageTotalFee - totalPaid;

    txtDueFee.setText(String.valueOf(totalDueResult));

  }
//*********************end          code          of          Auto          calculate          due
***************************
```
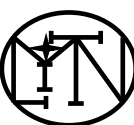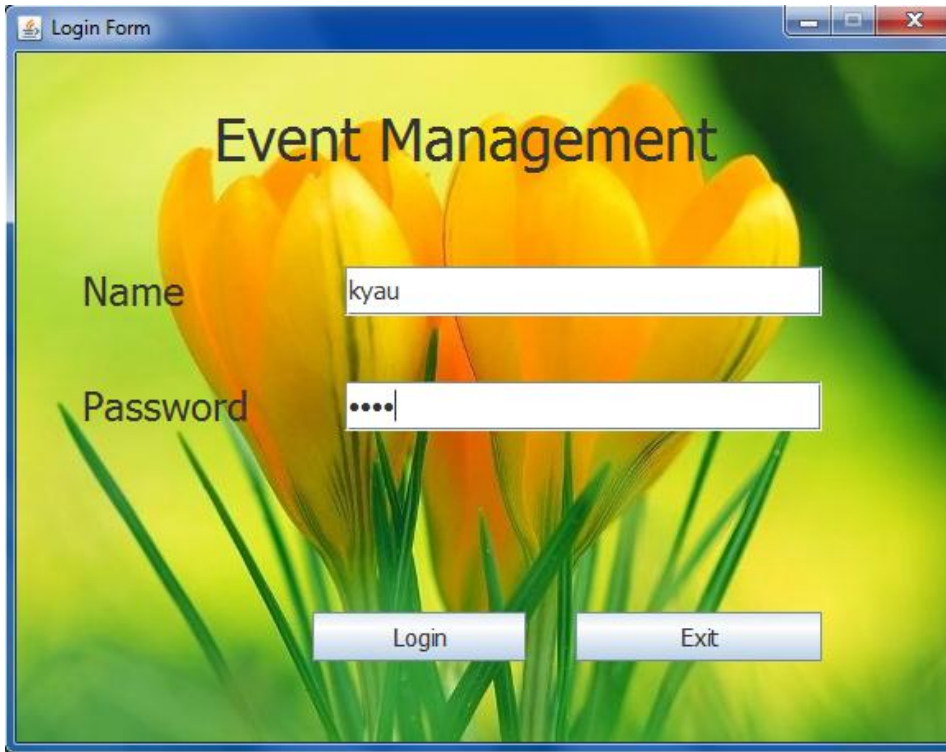
The screen Shoot of Event Management System:

Click on Login Button

Name: kyau

Password: 1234

Click on Login Button. It will open " Event Management System"
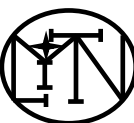
Click "Order Receive Tabbed Panel"

Click Save button and Save Information.

Click "Order Receive Program Table" on Tabbed Panel

Now you can also click on "Search Program ID" tabbed Panel

Fill Program ID by using previous Order or Implementation Program ID and click Delete Button

Click Update Button

Now Click on "Implementation Program" Tabbed Panel

Fill the Program ID by previous Order Program ID in Textfield and you will see automatically to show the

Previous data.

Then fill the other TextFields.

And click the Save Button

Click the "Implementation Table" tabbed panel

Click the "Hold Programs" tabbed panel

We have tested every single frames and function of this **Event Management System**. We Promise you it will work well. There is no bug. So it may help the organization to run more efficiently and effectively.

LACKING ANALYSIS:

I think that there are some lacks in my project. Some lacks are given below:

- I want to implemented all functional facilities such as tools cost, furniture cost etc.
- I want to implement the message system which shows the finished function Message ("Program is finished").
- I want to implement print system of functional programs.
- I want to this system online.
- I want to implement dynamic system.

CRITICAL ANALYSIS:

During Development, I have to face different kinds of problems. Such as:

- The System analysis and Design
- The System specification.
- The internal relationship among database tables.
- The internal logic of the proposed System.
- The prototype design of the proposed System.
- I had a little knowledge about the system requirement.
- I had to face problem to call internal class in project.
- I had to face to create Data Flow Diagram of the proposed system.

At last, We have been able to implement the Event Management System. During Development this project. We went to our teacher Mohammad Taherul Alam for taking tips about ER Diagram, Internal logic of project, DFD and Relation among different tables and we also had gone to our senior brothers to collect information to implement this project. We follow Java Complete Reference and Java_How_to_Program of Paul Deitel. We have taken the help of online websites.

Finally, we are very happy and delighted to implement The **Event Management System**.
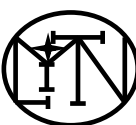
References:

- Java Complete Reference (9th edition)
- Java_How_to_Program_9th_Edition_Paul_Deitel_Harvey_Deitel(www.ebook-dl.com)
- http://www.w3schools.com/sql
- http://www.javawizerd.com
- www.mysql.com/sql
- https://www.javatpoint.com/java-tutorial
- http://www.java2s.com/
- https://docs.oracle.com/javase/7/docs/api/
- https://www.tutorialspoint.com/java/
- https://www.udemy.com/java-tutorial/
- www.learnjavaonline.org/
- https://iconverticons.com/online/
- www.learnjavaonline.org/
- https://www.oracle.com/
- https://www.youtube.com
- https://www.iftmuniversity.ac.in/iftmu/pages/coe/downloads/OnlineExam
- freeproject.org/get-full-project-on-online-test-in-java/