SCTR's ,PUNE INSTIUTE OF COMPUTER TECHNOGY,PUNE-411043

DEPARTMENT OF COMPUTER ENGG.

DBMSL(310247)

ACADEMIC YEAR:2020-21(SEM 1)

DBMSL MINI PROJECT REPORT

BATCH:K1

GUIDED BY : Deepali Kadam
SUBMITTED BY:
1)Vrushali Adgale(31102)

2)Aditi Bhosle(31112

3)Aniruddha Garje(31120)

ONLINE EXAM SYSTEM

ABSTRACT

Online Examination System is an application to take online examination, test in an efficient manner and no time wasting for manually checking of the test paper. The main objective of this web based online examination system is to efficiently evaluate the student through a fully automated system that not only saves lot of time but also gives fast and accurate results. For students they give papers according to their convenience from any location by using internet and time and there is no need of using extra thing like paper, pen etc.

INTRODUCTION

Online examination system is multiple choice question(MCQ) based examination .It provides an easy to use environment for Test conductor and students appearing for the examination. Online examination system helps students to offer a quick and easy way to appear for the test. It also provides the results immediately after the examination with 100% accuracy and security. Student can enter to perform exam only with their valid username and password. This examination contains multiple choice questions and appropriate number of options . The user can see their results after completing the exam and also the answer key .Result will contain no of questions attempted ,time taken, score, percentage etc. This helps the students to write the exam from far distance and which can provide security and simplicity.

SCOPE

- The main purpose of the system is to efficiently evaluate the users through a fully automated system that not only saves a lot of time but also gives fast results and saves paper.
- It is a cost effective and popular means of mass-evalution system.
- The faculty prepares the tests and questions for each exam.
- Users can check their score after completion of exam.
- User can view answer key too.

REQUIREMENTS

Hardware Requirements

Processor : Pentium iv

Ram : 256MB

Minimum storage capacity : 15GB

Key board : normal

Mouse : normal

Software Requirements

Operating System : Windows XP

Front End : Java Swing

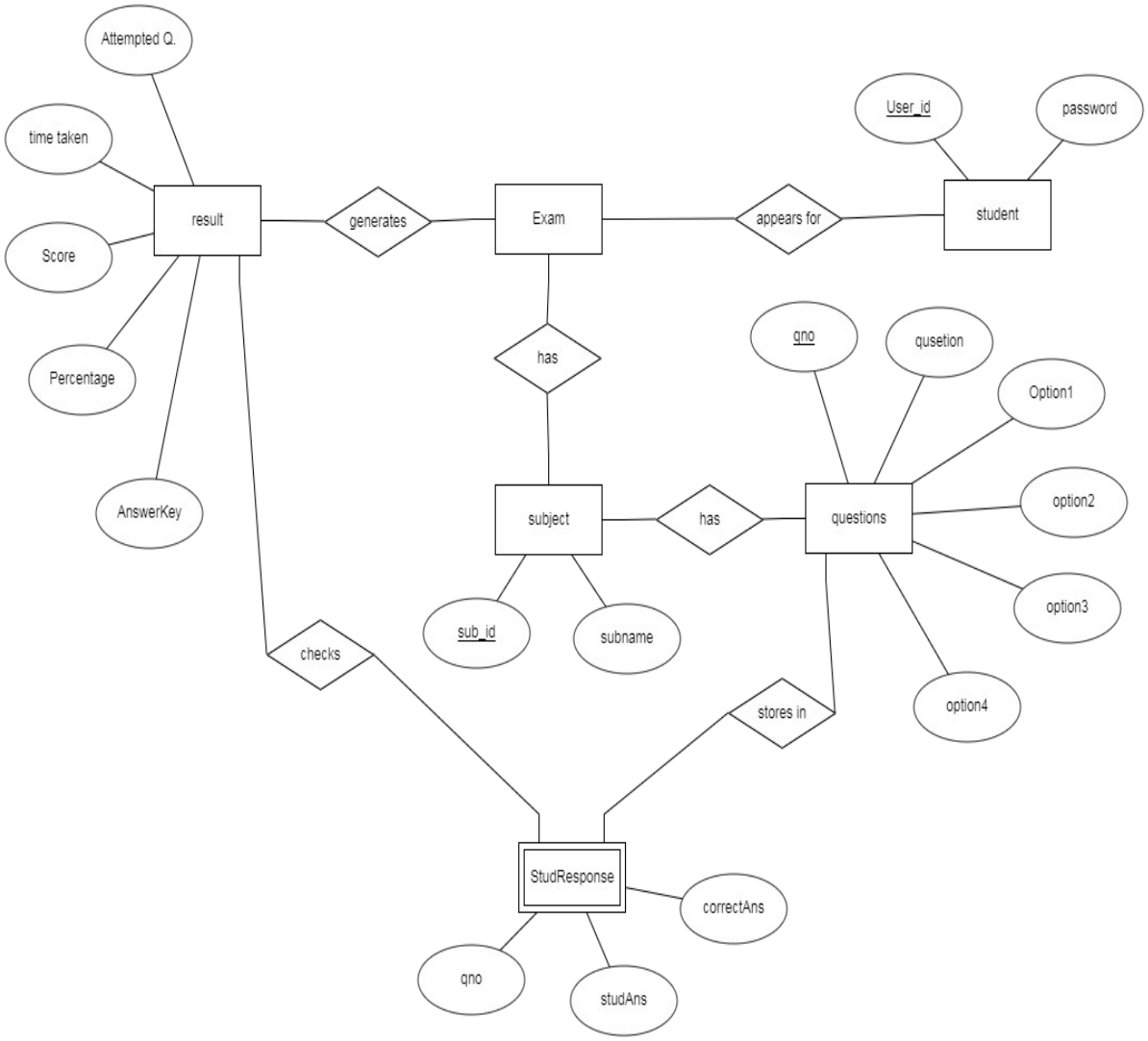
Back End : Java

Database: MySQL

DATA MODELING FEATURES

The real-world problems are more closely represented through the object-oriented data model. In this model, both the data and relationship are present in a single structure known as an object. We can store audio, video, images, etc in the database which was not possible in the relational model(although you can store audio and video in relational database, it is adviced not to store in the relational database). In this model, two are more objects are connected through links. We use this link to relate one object to other objects. All the data and relationships of each object are contained as a single unit. The two objects are connected through a common attribute.

ER DIAGRAM



DATABASE NORMALIZATION

The relation is in 1NF if it has no repeating groups. All tables has no repeating groups so they are in 1NF.

A relation is said to be in second normal form if it is already in first normal form and it has no partial dependency.

A relation is said to be in third normal form if it is already in 1st and 2nd NF and has no transitive dependency.

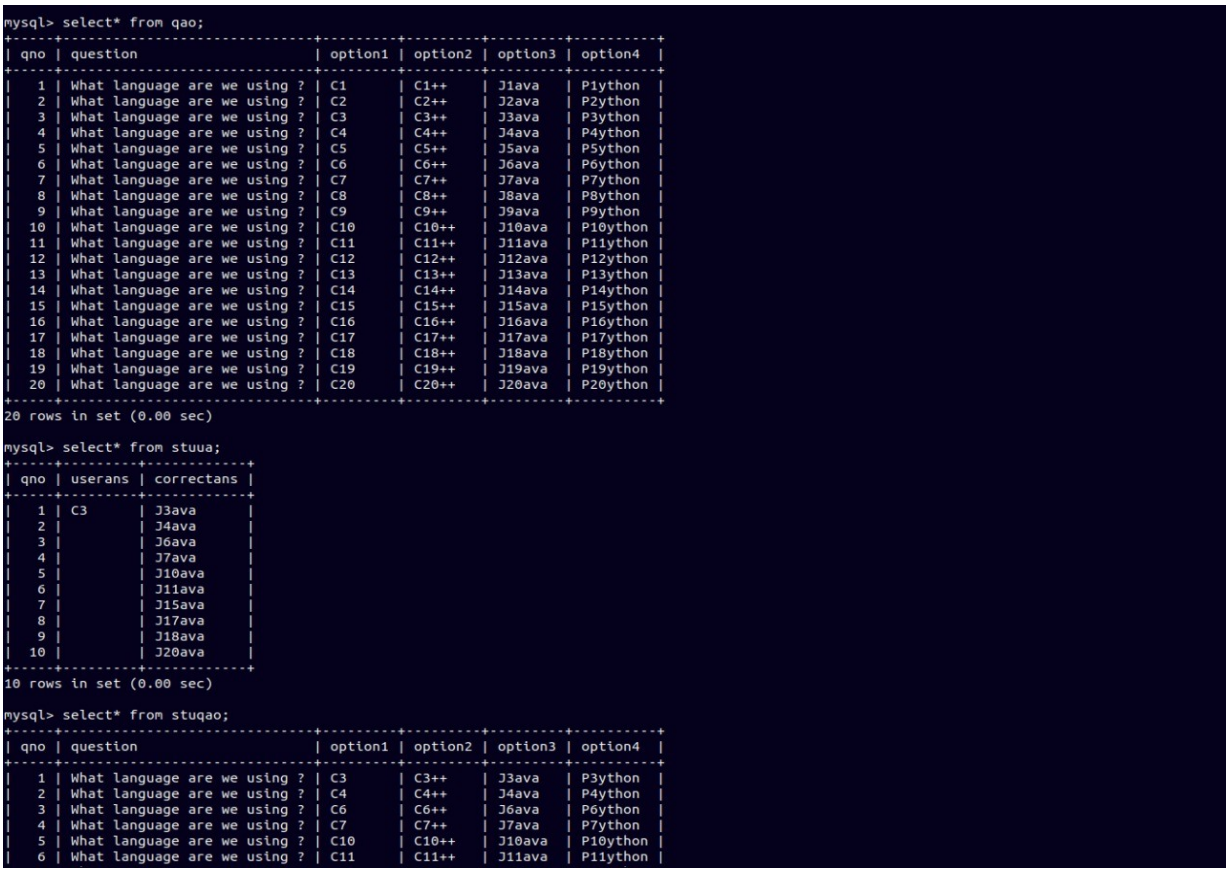
Tables:

```
mysql> use qa
Database changed
mysql> create table qao
    -> (
    -> qno int primary key,
    -> question varchar(250),
    -> option1 varchar(20),
    -> option2 varchar(20),
    -> option3 varchar(20),
    -> option4 varchar(20)
    -> );
Query OK, 0 rows affected (0.48 sec)

mysql> create table stuqao
    -> (
    -> qno int primary key,
    -> question varchar(250),
    -> option1 varchar(20),
    -> option2 varchar(20),
    -> option3 varchar(20),
    -> option4 varchar(20)
    -> );
Query OK, 0 rows affected (0.56 sec)

mysql> alter table stuqao add unique(qno);
Query OK, 0 rows affected (0.39 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> create table stuua
    -> (
    -> qno int primary key,
    -> ans varchar(20),
    -> foreign key (qno) references stuqao(qno)
    -> );
Query OK, 0 rows affected (0.47 sec)
```



```
mysql> select* from qao;
+-----+------------------------------+---------+---------+---------+----------+
| qno | question                     | option1 | option2 | option3 | option4  |
+-----+------------------------------+---------+---------+---------+----------+
|   1 | What language are we using ? | C1      | C1++    | J1ava   | P1ython  |
|   2 | What language are we using ? | C2      | C2++    | J2ava   | P2ython  |
|   3 | What language are we using ? | C3      | C3++    | J3ava   | P3ython  |
|   4 | What language are we using ? | C4      | C4++    | J4ava   | P4ython  |
|   5 | What language are we using ? | C5      | C5++    | J5ava   | P5ython  |
|   6 | What language are we using ? | C6      | C6++    | J6ava   | P6ython  |
|   7 | What language are we using ? | C7      | C7++    | J7ava   | P7ython  |
|   8 | What language are we using ? | C8      | C8++    | J8ava   | P8ython  |
|   9 | What language are we using ? | C9      | C9++    | J9ava   | P9ython  |
|  10 | What language are we using ? | C10     | C10++   | J10ava  | P10ython |
|  11 | What language are we using ? | C11     | C11++   | J11ava  | P11ython |
|  12 | What language are we using ? | C12     | C12++   | J12ava  | P12ython |
|  13 | What language are we using ? | C13     | C13++   | J13ava  | P13ython |
|  14 | What language are we using ? | C14     | C14++   | J14ava  | P14ython |
|  15 | What language are we using ? | C15     | C15++   | J15ava  | P15ython |
|  16 | What language are we using ? | C16     | C16++   | J16ava  | P16ython |
|  17 | What language are we using ? | C17     | C17++   | J17ava  | P17ython |
|  18 | What language are we using ? | C18     | C18++   | J18ava  | P18ython |
|  19 | What language are we using ? | C19     | C19++   | J19ava  | P19ython |
|  20 | What language are we using ? | C20     | C20++   | J20ava  | P20ython |
+-----+------------------------------+---------+---------+---------+----------+
20 rows in set (0.00 sec)

mysql> select* from stuua;
+-----+---------+------------+
| qno | userans | correctans |
+-----+---------+------------+
|   1 | C3      | J3ava      |
|   2 |         | J4ava      |
|   3 |         | J6ava      |
|   4 |         | J7ava      |
|   5 |         | J10ava     |
|   6 |         | J11ava     |
|   7 |         | J15ava     |
|   8 |         | J17ava     |
|   9 |         | J18ava     |
|  10 |         | J20ava     |
+-----+---------+------------+
10 rows in set (0.00 sec)

mysql> select* from stuqao;
+-----+------------------------------+---------+---------+---------+----------+
| qno | question                     | option1 | option2 | option3 | option4  |
+-----+------------------------------+---------+---------+---------+----------+
|   1 | What language are we using ? | C3      | C3++    | J3ava   | P3ython  |
|   2 | What language are we using ? | C4      | C4++    | J4ava   | P4ython  |
|   3 | What language are we using ? | C6      | C6++    | J6ava   | P6ython  |
|   4 | What language are we using ? | C7      | C7++    | J7ava   | P7ython  |
|   5 | What language are we using ? | C10     | C10++   | J10ava  | P10ython |
|   6 | What language are we using ? | C11     | C11++   | J11ava  | P11ython |
```

GRAPHICAL USER INTERFACE

Swing

Java Swing is a part of Java Foundation Classes (JFC) that is *used to create window-based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as **JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.**

Java JLabel

The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.

JavaTextArea

The object of a JTextArea class is a multi line region that displays text. It allows the editing of multiple line text. It inherits JTextComponent class

Java JCheckBox

The JCheckBox class is used to create a checkbox. It is used to turn an option on (true) or off (false). Clicking on a CheckBox changes its state from "on" to "off" or from "off" to "on ".It inherits JToggleButton class.

Java JRadioButton

The JRadioButton class is used to create a radio button. It is used to choose one option from multiple options. It is widely used in exam systems or quiz.

It should be added in ButtonGroup to select one radio button only.

DATABASE DESCRIPTION

SQL Server

SQL Server is a relational database management system (RDBMS) that uses Transact SQL to send request between a client and SQL Server.SQL Server is designed to be a client/server system. Client/Server systems are constructed so that the database can reside on a central computer, known as server, and be shared among several users. When users want to access data on the SQL Server, they run an application on their local computer, known as a client that connects over a network to the server running SQL Server. The following are the factors for which I have chosen SQL Server as the back end tool:

Advantages of using SQL Server:

1. Multi-user database
2. Supports RDBMS
3. It’s very fast.
4. It’s relatively easy to use.
5. It’s widely used
6. More secure

SOURCE CODE

```
1 //importing required packages
2 import java.awt.;
3 import java.awt.event.; //package to implement event response
4 import javax.swing.; //package to implement swing gui
```

```
5 import java.sql.; //package to connect to mysql database
6
7 class OnlineTest extends JFrame implements ActionListener
8 {
 9 JLabel l; //for setting question text
10 JRadioButton jb[]=new JRadioButton[5]; //for options of the question
 11 JButton b1,b2,b3; //for start/next, previous and result buttons
12 ButtonGroup bg;
13 int count=0,attempted=0,current=-1,x=1,y=1,now=0;
 14 long StartTime, EndTime,seconds,minutes,flag=0;;
 15 int a;
16 OnlineTest(String s)
17 {
18 super(s);
19 l=new JLabel();
20 add(l);
 21 bg=new ButtonGroup();
22 for(int i=0;i<5;i++)
23 {
 24 jb[i]=new JRadioButton();
25 add(jb[i]);
 26 bg.add(jb[i]);
```

```
27 }
28
 b1=new JButton("Start"); //Setting text of first button
 29 b1.addActionListener(this); //Added event listener, i.e., action to be taken on being clicked
30 add(b1); //added start button
 31 welcome(); //displays welcome message
32 b2=new JButton("Previous"); //Setting text of second button
 33 b3=new JButton("Result"); //Setting text of third button
34 b2.addActionListener(this);
35 b3.addActionListener(this);
36 add(b2);add(b3);
 37 // added previous and result buttons
 38 l.setBounds(30,40,450,20); //setting dimensions of question area
39 if(current !=-1)
 40 {
41 jb[0].setBounds(50,80,100,20); //setting dimensions and coordinates of radio button group
42 jb[1].setBounds(50,110,100,20);
43 jb[2].setBounds(50,140,100,20);
 44 jb[3].setBounds(50,170,100,20);
45 }
```

```
 46 b1.setBounds(100,240,100,30); //setting dimensions and
coordinates of start/next button

47 b2.setBounds(270,240,100,30); //setting dimensions and
coordinates of previous button

48 b3.setBounds(400,240,100,30); //setting dimensions and
coordinates of result button

49 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

50 setLayout(null);

 51 setLocation(250,100);

52 setVisible(true);

53 setSize(600,350);

54 }

 55 public void actionPerformed(ActionEvent e)

56 {

 57try

58 {

 59 Class.forName("com.mysql.jdbc.Driver");

 60 Connection con =
DriverManager.getConnection("jdbc:mysql:///qa","root","root"); //
connecting to database 'qa'

61 Statement stmt = con.createStatement();

 62 if(e.getSource()==b1 && current ==9 ) //if all 10 questions
have been displayed and user clicks on next, i.e., no more
questions are available to be displayed

63 {
```

64 adduserans(); //adding user's response to the 10th question

65 JOptionPane.showMessageDialog(this,"No more questions. Please go back to previous question or end test and see result.\ n");

66 }

67 else if(e.getSource()==b1) //if user clicks on start/next and there are more questions available to be displayed

68 {

69 if(current == -1) //if user hasn't started test yet, i.e., she/he clicks on "start" button

70 {

71 StartTime = System.currentTimeMillis(); //stores time when user starts test

72 b1.setText("Next"); //setting text of b1 button to "next"

73 }

74 else

75 adduserans(); //adding user's response to the question

76 current++; //incrementing counter of questions countered

77 setnext(); //setting next question

78 }

79 else if(e.getSource()==b2 && current ==0) //if user clicks on previous button and there are no more questions available to be displayed

80 {

81 adduserans();

82 JOptionPane.showMessageDialog(this,"No more questions. Please go back to next question or end test and see result.\n");

83 }

84 else if(e.getSource()==b2) //if user clicks on previous and there are more questions available to be displayed

85 {

86 current--; //decrementing counter of questions countered

87 adduserans();

88 setnext();

89 }

90 else if(e.getActionCommand().equals("Result")) //if user clicks on result button

91 {

92 EndTime = System.currentTimeMillis(); //stores time when user ends test 93 EndTime-=StartTime; //stores time taken by user to give test in milliseconds

94 EndTime/=1000; //stores time taken by user to give test in seconds

95 if(EndTime>=60) // if time can be expressed in minutes or hours

96 {

97 seconds = EndTime%60; //calculating seconds

98 EndTime/=60; //calculating minutes

99 flag=1;

100 if(EndTime>=60) //if time can be expressed in hours

```
101 {

102 flag=2;

103 minutes=EndTime%60; //calculating minutes

104 EndTime/=60; //calculating hours

105 }

106 }

107 current++;

108 check(); //checks user's responses against correct responses stored in database

109 if(flag==0)

110 a = JOptionPane.showConfirmDialog(this,"Attempted questions: "+attempted+" / 10\nTime taken: "+EndTime+" seconds\nYour Score: "+count+" / 10\nPercentage: "+(count*10)+" %\nDo you wish to see the answer key ?");

111 else if(flag==1)

112 a = JOptionPane.showConfirmDialog(this,"Attempted questions: "+attempted+" / 10\nTime taken: "+EndTime+" minutes "+seconds+" seconds\nYour Score: "+count+" / 10\nPercentage: "+(count*10)+" %\nDo you wish to see the answer key ?");

113 else

114 a = JOptionPane.showConfirmDialog(this,"Attempted questions: "+attempted+" / 10\nTime taken: "+EndTime+" hours "+minutes+" minutes "+seconds+" seconds\nYour Score: "+count+" / 10\nPercentage: "+(count*10)+" %\nDo you wish to see the answer key ?");
```

115 //displays number of attempted questions, total score and percentage 116 if(a==JOptionPane.YES_OPTION) //checks if user wants to see answer key or not

117 showAnswerKey();

118 else

119 {

120 stmt.executeUpdate("delete from ua");

121 stmt.executeUpdate("delete from stuua");

122 stmt.executeUpdate("delete from stuqao");

123 stmt.executeUpdate("delete from qao");

124 System.exit(0); //closes interface and exits

125 }

126 }

127 }

128 catch(Exception ex)

129 {

130 System.out.println("actionPerformed"+ex);

131 }

132 }

133 void welcome() //Welcome Message

134 {

135 l.setText("Welcome to the online examination. Click button to start with the test.") ;

136 }

```
137 void setnext() //function to set next/previous question

138 {

139 jb[4].setSelected(true);

140 try

141 {

142 Class.forName("com.mysql.jdbc.Driver");

143 Connection con =
DriverManager.getConnection("jdbc:mysql:///qa","root","root"); //
connecting to database 'qa'

144 Statement stmt = con.createStatement();

145 if(current==0)

146 {

147 String sql="select * from stuqao where qno=1"; //selects all
fields of table 'stuqao' with value of field qno equal to 1

148 ResultSet rs = stmt.executeQuery(sql); //executing mysql
query

149 rs.next(); //pointing to next row of result set

150 String s1 =rs.getString("question"); //getting value stored in
result set under field "question"

151 String s2 =rs.getString("option1"); //getting value stored in
result set under field "option1"

152 String s3 =rs.getString("option2"); //getting value stored in
result set under field "option2"

153 String s4 =rs.getString("option3"); //getting value stored in
result set under field "option3"
```

154 String s5 =rs.getString("option4"); //getting value stored in result set under field "option4"

155 l.setText("Q.1 "+s1); //setting question

156 jb[0].setText(s2);jb[1].setText(s3);jb[2].setText(s4);jb[3].setText(s5); //setting options

157 }

158 if(current==1)

159 {

160 String sql="select * from stuqao where qno=2";

161 ResultSet rs = stmt.executeQuery(sql);

162 rs.next();

163 String s1 =rs.getString("question");

164 String s2 =rs.getString("option1");

165 String s3 =rs.getString("option2");

166 String s4 =rs.getString("option3");

167 String s5 =rs.getString("option4");

168 l.setText("Q.2 "+s1);

169 jb[0].setText(s2);jb[1].setText(s3);jb[2].setText(s4);jb[3].setText(s5); 170 }

171 if(current==2)

172 {

173 String sql="select * from stuqao where qno=3";

```
174 ResultSet rs = stmt.executeQuery(sql);
175 rs.next();
176 String s1 =rs.getString("question");
177 String s2 =rs.getString("option1");
178 String s3 =rs.getString("option2");
179 String s4 =rs.getString("option3");
180 String s5 =rs.getString("option4");
181 l.setText("Q.3 "+s1);
182
jb[0].setText(s2);jb[1].setText(s3);jb[2].setText(s4);jb[3].setText(s
5); 183 }
184 if(current==3)
185 {
186 String sql="select * from stuqao where qno=4";
187 ResultSet rs = stmt.executeQuery(sql);
188 rs.next();
189 String s1 =rs.getString("question");
190 String s2 =rs.getString("option1");
191 String s3 =rs.getString("option2");
192 String s4 =rs.getString("option3");
193 String s5 =rs.getString("option4");
194 l.setText("Q.4 "+s1);
```

195
jb[0].setText(s2);jb[1].setText(s3);jb[2].setText(s4);jb[3].setText(s
5); 196 }

197 if(current==4)

198 {

199 String sql="select * from stuqao where qno=5";

200 ResultSet rs = stmt.executeQuery(sql);

201 rs.next();

202 String s1 =rs.getString("question");

203 String s2 =rs.getString("option1");

204 String s3 =rs.getString("option2");

205 String s4 =rs.getString("option3");

206 String s5 =rs.getString("option4");

207 l.setText("Q.5 "+s1);

208
jb[0].setText(s2);jb[1].setText(s3);jb[2].setText(s4);jb[3].setText(s
5); 209 }

210 if(current==5)

211 {

212 String sql="select * from stuqao where qno=6";

213 ResultSet rs = stmt.executeQuery(sql);

214 rs.next();

215 String s1 =rs.getString("question");

216 String s2 =rs.getString("option1");

217 String s3 =rs.getString("option2");

218 String s4 =rs.getString("option3");

219 String s5 =rs.getString("option4");

220 l.setText("Q.6 "+s1);

221
jb[0].setText(s2);jb[1].setText(s3);jb[2].setText(s4);jb[3].setText(s
5); 222 }

223 if(current==6)

224 {

225 String sql="select * from stuqao where qno=7";

226 ResultSet rs = stmt.executeQuery(sql);

227 rs.next();

228 String s1 =rs.getString("question");

229 String s2 =rs.getString("option1");

230 String s3 =rs.getString("option2");

231 String s4 =rs.getString("option3");

232 String s5 =rs.getString("option4");

233 l.setText("Q.7 "+s1);

234
jb[0].setText(s2);jb[1].setText(s3);jb[2].setText(s4);jb[3].setText(s
5); 235 } 236 if(current==7)

237 {

238 String sql="select * from stuqao where qno=8";

239 ResultSet rs = stmt.executeQuery(sql);

240 rs.next();

241 String s1 =rs.getString("question");

242 String s2 =rs.getString("option1");

243 String s3 =rs.getString("option2");

244 String s4 =rs.getString("option3");

245 String s5 =rs.getString("option4");

246 l.setText("Q.8 "+s1);

247
jb[0].setText(s2);jb[1].setText(s3);jb[2].setText(s4);jb[3].setText(s5); 248 }

249 if(current==8)

250 {

251 String sql="select * from stuqao where qno=9";

252 ResultSet rs = stmt.executeQuery(sql); 253 rs.next();

254 String s1 =rs.getString("question");

255 String s2 =rs.getString("option1");

256 String s3 =rs.getString("option2");

257 String s4 =rs.getString("option3");

258 String s5 =rs.getString("option4");

259 l.setText("Q.9 "+s1);

260
jb[0].setText(s2);jb[1].setText(s3);jb[2].setText(s4);jb[3].setText(s5); 261 }

262 if(current==9)

263 {

264 String sql="select * from stuqao where qno=10";

265 ResultSet rs = stmt.executeQuery(sql);

266 rs.next();

267 String s1 =rs.getString("question");

268 String s2 =rs.getString("option1");

269 String s3 =rs.getString("option2");

270 String s4 =rs.getString("option3");

271 String s5 =rs.getString("option4");

272 l.setText("Q.10 "+s1);

273
jb[0].setText(s2);jb[1].setText(s3);jb[2].setText(s4);jb[3].setText(s5); 274 }

275 l.setBounds(30,40,450,20);

276 for(int i=0,j=0;i<=90;i+=30,j++)

277 jb[j].setBounds(50,80+i,200,20);

278 }

279 catch(Exception e)

280 {

281 System.out.println("setnext\n"+e);

282 }

283 }

284 void adduserans() //function to connect to qa database and insert user's answers into ua table

285 {

286 try

287 {

288 Class.forName("com.mysql.jdbc.Driver");

289 Connection con =
DriverManager.getConnection("jdbc:mysql:///qa","root","root");

290 Statement stmt = con.createStatement();

291 for(int i=0;i<=3;i++)

292 {

293 if(jb[i].isSelected()) //if answer has been selected in radio
button group 294 {

295 String sql1 = "insert into stuua(qno,userans) values("+
(current+1)+",'"+jb[i].getText()+"') on duplicate key update
userans='"+jb[i].getText()+"'";

296 //if user opts to change answer, she/he can do so because of
the above command.

297 //This allows the table to accept overwriting of existing
values

298 stmt.executeUpdate(sql1);

299 //con.close();

300 break;

301 }

302 }

303 }

304 catch(Exception e)

```
305 {
306 System.out.println("adduserans\n"+e);
307 }
308 }
309 void check() //function to check number of correct answers
310 {
311 try
312 {
313 Class.forName("com.mysql.jdbc.Driver");
314 Connection con =
DriverManager.getConnection("jdbc:mysql:///qa","root","root");
315 Statement stmt = con.createStatement();
316 for(int i=1;i<=10;i++)
317 {
318 String sql="select userans, correctans from stuua where
qno="+i+"";
319 ResultSet rs = stmt.executeQuery(sql);
320 rs.next();
321 String s1 =rs.getString("userans");
322 String s2 =rs.getString("correctans");
323 if(!(s1.equals(""))) //checks if the user has attempted the
question or not 324 attempted++;
325 if(s1.equals(s2)) //checks if user's answer matches correct
answer
```

```
326 count++;

327 }

328 }

329 catch(Exception e)

330 {

331 System.out.println("check\n"+e);

332 }

333 }

334 void showAnswerKey() //function to print answer key if
requested for 335 {

336 try

337 {

338 Class.forName("com.mysql.jdbc.Driver");

339 Connection con =
DriverManager.getConnection("jdbc:mysql:///qa","root","root");

340 Statement stmt = con.createStatement();

341 String answerkey="";

342 answerkey+="Answer Key:\nQ.No. Your answer Correct
Answer\n";

343 for(int i=1;i<=10;i++) 344 {

345 String sql="select userans, correctans from stuua where
qno="+i+"";

346 ResultSet rs = stmt.executeQuery(sql);

347 rs.next();
```

348 String s1 =rs.getString("userans"); //stores user's answer

349 if(s1.equals("")) //if user hasn't attempted this question, we assign NA to variable

350 s1="NA";

351 String s2 =rs.getString("correctans"); //stores correct answer

352 if(i<=9)

353 answerkey+=" "+(char)(i+48)+" "+s1+" "+s2+"\n";

354 else //0-9 is 48-57 in ascii

355 answerkey+=" 10"+" "+s1+" "+s2+"\n";

356 }

357 JOptionPane.showMessageDialog(this,answerkey); //prints answer key 358 stmt.executeUpdate("delete from ua");

359 stmt.executeUpdate("delete from stuua");

360 stmt.executeUpdate("delete from stuqao");

361 stmt.executeUpdate("delete from qao");

362 System.exit(0);

363 con.close();

364 }

365 catch(Exception e)

366 {

367 System.out.println("showAnswerKey\n"+e);

368 }

369 }

```
370 static void qaoDBcon() //function to connect to qa database
and insert question and options into qao table
371 {
372 try
373 {
374 Class.forName("com.mysql.jdbc.Driver");
375 Connection con =
DriverManager.getConnection("jdbc:mysql:///qa","root","root");
376 Statement stmt = con.createStatement();
377 stmt.executeUpdate("insert into qao values(1,'What
language are we using ?','C1','C1++','J1ava','P1ython')");
378 stmt.executeUpdate("insert into qao values(2,'What
language are we using ?','C2','C2++','J2ava','P2ython')");
379 stmt.executeUpdate("insert into qao values(3,'What
language are we using ?','C3','C3++','J3ava','P3ython')");
380 stmt.executeUpdate("insert into qao values(4,'What
language are we using ?','C4','C4++','J4ava','P4ython')");
381 stmt.executeUpdate("insert into qao values(5,'What
language are we using ?','C5','C5++','J5ava','P5ython')");
382 stmt.executeUpdate("insert into qao values(6,'What
language are we using ?','C6','C6++','J6ava','P6ython')");
383 stmt.executeUpdate("insert into qao values(7,'What
language are we using ?','C7','C7++','J7ava','P7ython')");
384 stmt.executeUpdate("insert into qao values(8,'What
language are we using ?','C8','C8++','J8ava','P8ython')");
```

```
385 stmt.executeUpdate("insert into qao values(9,'What
language are we using ?','C9','C9++','J9ava','P9ython')");

386 stmt.executeUpdate("insert into qao values(10,'What
language are we using ?','C10','C10++','J10ava','P10ython')");

387 stmt.executeUpdate("insert into qao values(11,'What
language are we using ?','C11','C11++','J11ava','P11ython')");

388 stmt.executeUpdate("insert into qao values(12,'What
language are we using ?','C12','C12++','J12ava','P12ython')");

389 stmt.executeUpdate("insert into qao values(13,'What
language are we using ?','C13','C13++','J13ava','P13ython')");

390 stmt.executeUpdate("insert into qao values(14,'What
language are we using ?','C14','C14++','J14ava','P14ython')");

391 stmt.executeUpdate("insert into qao values(15,'What
language are we using ?','C15','C15++','J15ava','P15ython')");

392 stmt.executeUpdate("insert into qao values(16,'What
language are we using ?','C16','C16++','J16ava','P16ython')");

393 stmt.executeUpdate("insert into qao values(17,'What
language are we using ?','C17','C17++','J17ava','P17ython')");

394 stmt.executeUpdate("insert into qao values(18,'What
language are we using ?','C18','C18++','J18ava','P18ython')");

395 stmt.executeUpdate("insert into qao values(19,'What
language are we using ?','C19','C19++','J19ava','P19ython')");

396 stmt.executeUpdate("insert into qao values(20,'What
language are we using ?','C20','C20++','J20ava','P20ython')");

397 con.close();

398 }

399 catch(Exception e)
```

```
400 {
401 System.out.println("qaoDBcon\n"+e);
402 }
403 }
404 static void uaDBcon() //function to connect to qa database
and insert correct answers of all questions into ua table
405 {
406 try
407 {
408 Class.forName("com.mysql.jdbc.Driver");
409 Connection con =
DriverManager.getConnection("jdbc:mysql:///qa","root","root");
410 Statement stmt = con.createStatement();
411 stmt.executeUpdate("insert into ua values(1,'','J1ava')");
412 stmt.executeUpdate("insert into ua values(2,'','J2ava')");
413 stmt.executeUpdate("insert into ua values(3,'','J3ava')");
414 stmt.executeUpdate("insert into ua values(4,'','J4ava')");
415 stmt.executeUpdate("insert into ua values(5,'','J5ava')");
416 stmt.executeUpdate("insert into ua values(6,'','J6ava')");
417 stmt.executeUpdate("insert into ua values(7,'','J7ava')");
418 stmt.executeUpdate("insert into ua values(8,'','J8ava')");
419 stmt.executeUpdate("insert into ua values(9,'','J9ava')");
420 stmt.executeUpdate("insert into ua values(10,'','J10ava')");
```

```
421 stmt.executeUpdate("insert into ua values(11,'','J11ava')");
422 stmt.executeUpdate("insert into ua values(12,'','J12ava')");
423 stmt.executeUpdate("insert into ua values(13,'','J13ava')");
424 stmt.executeUpdate("insert into ua values(14,'','J14ava')");
425 stmt.executeUpdate("insert into ua values(15,'','J15ava')");
426 stmt.executeUpdate("insert into ua values(16,'','J16ava')");
427 stmt.executeUpdate("insert into ua values(17,'','J17ava')");
428 stmt.executeUpdate("insert into ua values(18,'','J18ava')");
429 stmt.executeUpdate("insert into ua values(19,'','J19ava')");
430 stmt.executeUpdate("insert into ua values(20,'','J20ava')");
431 con.close();
432 }
433 catch(Exception e)
434 {
435 System.out.println("uaDBcon\n"+e);
436 }
437 }
438 static void pickrandom()
439 /*function to pick 10 random questions from the qao table,
store them in stuqao table and display to the user
440 this function also stores the correct answers of these 10
picked questions into the table stuua
441 */
```

```
442 {
443 try
444 {
445 Class.forName("com.mysql.jdbc.Driver");
446 Connection con =
DriverManager.getConnection("jdbc:mysql:///qa","root","root");
447 Statement stmt = con.createStatement();
448 int a[]=new int[21];
449 int c=0;
450 int p;
451 for(int i=0;i<=20;i++)
452 a[i]=0;
453 while(c!=10)
454 {
455 p=1+(int)(Math.random()*20); //generating random integers
in range [1,20]
456 if(a[p]==0)
457 {
458 a[p]=1; //marking the 10 randomly selected integers
459 c++; //counting number of random indexes marked
460 }
461 }
462 c=0;
```

```
463 for(int i=1;i<=20;i++)
464 {
465 if(a[i]==1) //checks if integer 'i' has been picked by code as a random integer
466 {
467 c++;
468 String sql="select * from qao where qno="+i+"";
469 ResultSet randomrs = stmt.executeQuery(sql);
470 randomrs.next();
471 String s1 =randomrs.getString("question");
472 String s2 =randomrs.getString("option1");
473 String s3 =randomrs.getString("option2");
474 String s4 =randomrs.getString("option3");
475 String s5 =randomrs.getString("option4");
476 stmt.executeUpdate("insert into stuqao values("+c+",'"+s1+"','"+s2+"','"+s3+"','"+s4+"','"+s5+"')");
477 randomrs.close();
478 sql="select * from ua where qno="+i+"";
479 randomrs = stmt.executeQuery(sql);
480 randomrs.next();
481 s1 =randomrs.getString("correctans");
482 stmt.executeUpdate("insert into stuua values("+c+",'','"+s1+"')");
483 randomrs.close();
```

```
485 }
486 }
487 con.close();
488 }
489 catch(Exception e)
490 {
491 System.out.println("pickrandom\n"+e);
492 }
493 }
494 public static void main(String s[])
495 {
496 qaoDBcon(); //creating question-option database
497 uaDBcon(); //creating user answer-correct answer database
498 pickrandom(); //creating question-option database that will be asked to student
499 new OnlineTest("Online Exam System"); // creating object
500 }
```

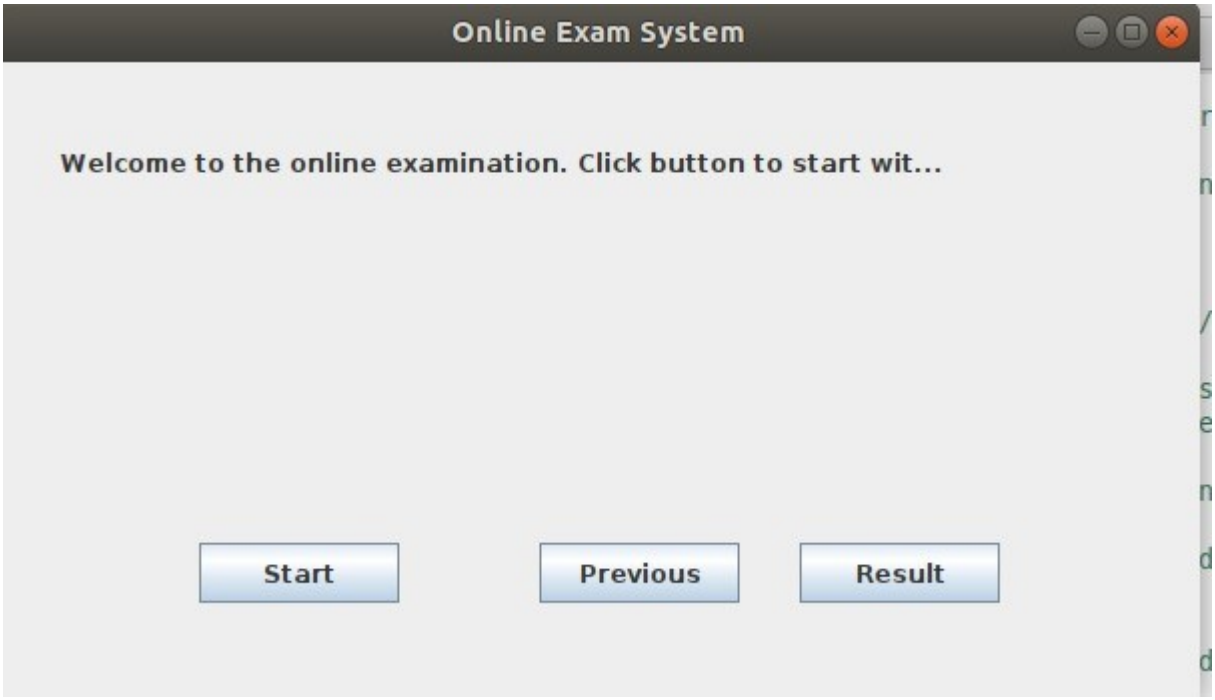
TESTING

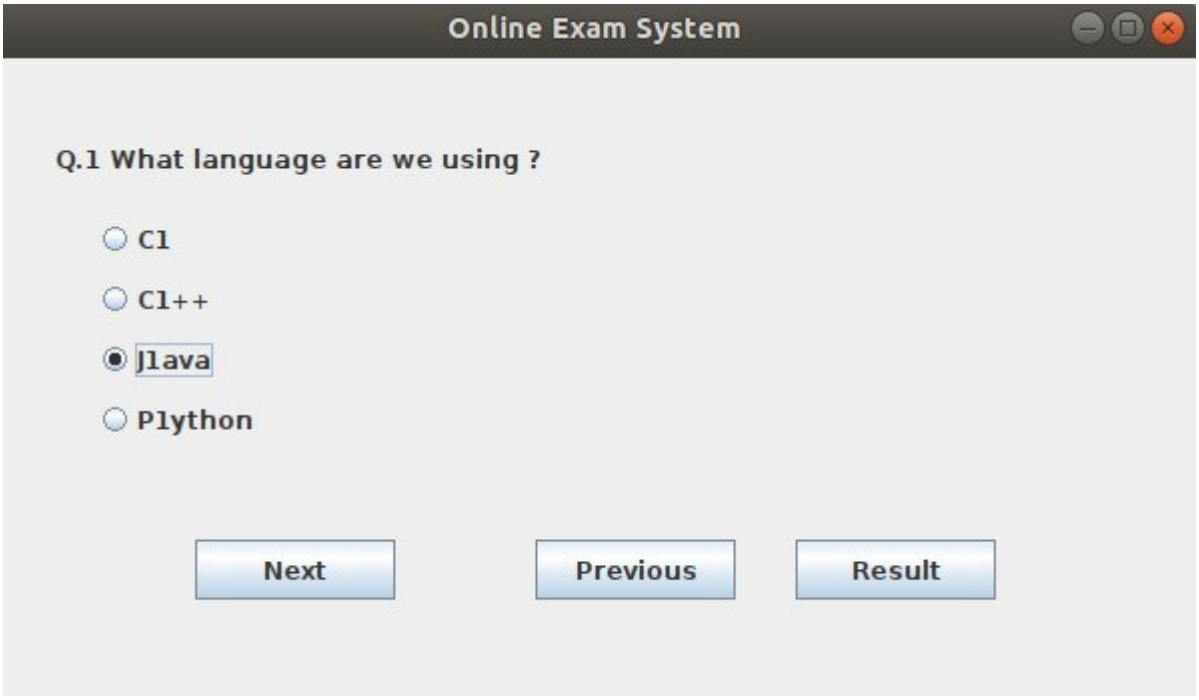
System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. Testing is the process of executing a program with the explicit intention of finding errors that is making the program fail. The tester may analysts, programmer or a specialist trained for software testing, is actually trying to make the program fail.

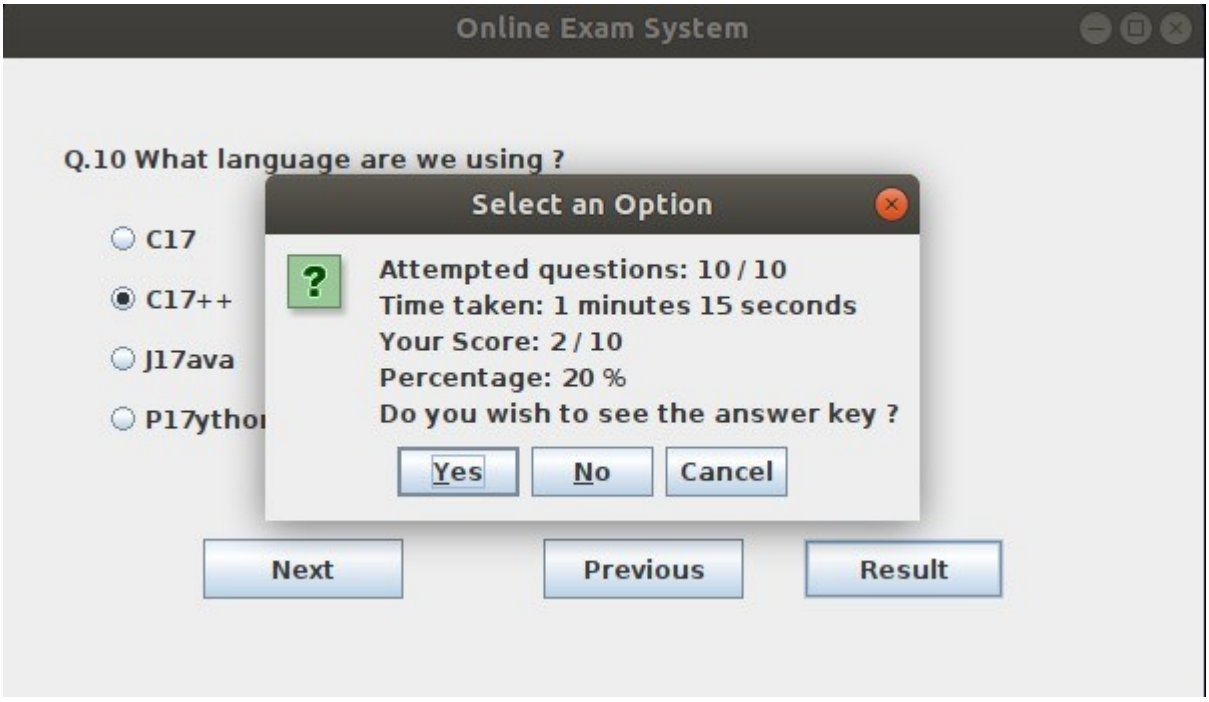
Analysts know that an effective testing program does not guarantee system reliability. Therefore reliability must be designed into the system.

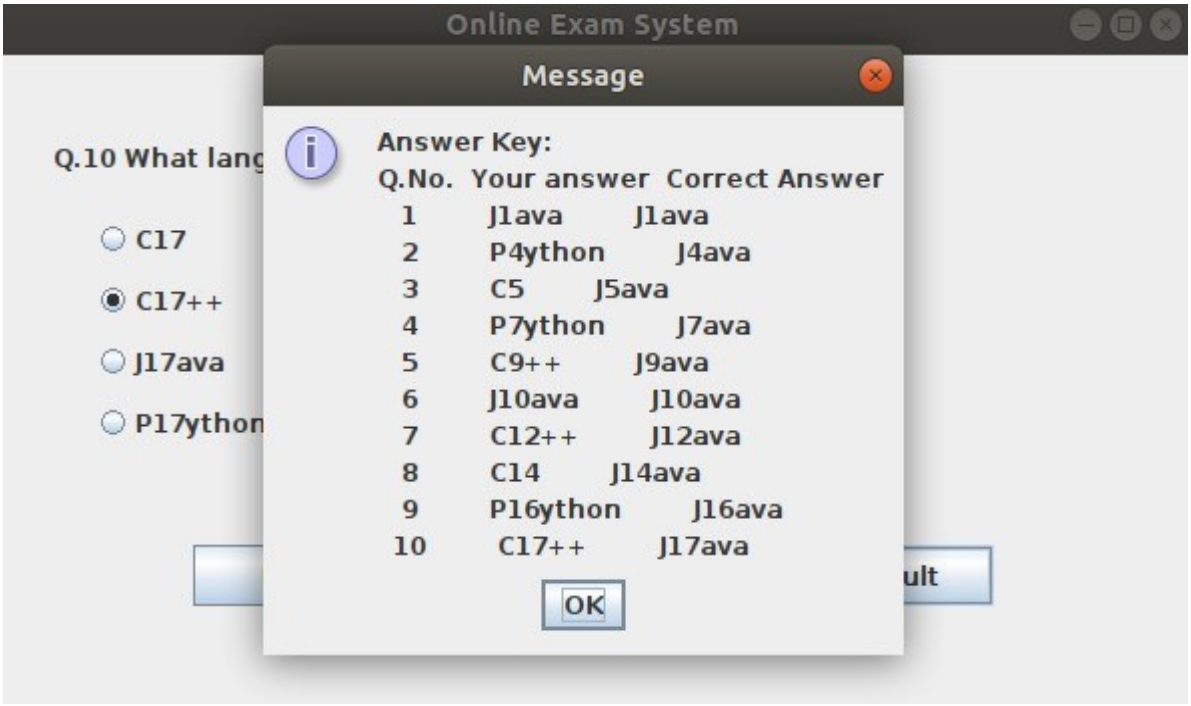
JDBC connection:

```
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql:///qa","root","root"); //connecting to database 'qa'
Statement stmt = con.createStatement();
if(e.getSource()==b1 && current ==9 ) //if all 10 questions have been displayed and user clicks on next, i.e., no mor
{
    adduserans(); //adding user's response to the 10th question
```




Online Exam System
Q.1 What language are we using ?
C1
C1++
J1ava
P1ython
Next
Previous
Result


Online Exam System
Q.10 What language are we using ?
C17
C17++
J17ava
P17thon
Select an Option
Attempted questions: 10 / 10
Time taken: 1 minutes 15 seconds
Your Score: 2 / 10
Percentage: 20 %
Do you wish to see the answer key ?
Yes
No
Cancel
Next
Previous
Result



Thus ,we have successfully implemented Online exam application.

CONCLUSION

Online Examination has been developed and the system was tested with proper data. The system results in regular timing preparation of the required output .In comparison with the manual system, the benefit under a computer system considerable in to saving of manpower, working hour and efforts.It can observe that the information required can be obtained with ease and accuracy in the computerized system. The user with minimum knowledge about computer can be able operate the system easily. Online massage has been provided to help the user to take necessary, correct action while using the

system. Various validation techniques have been used to implement accuracy of data in all formats of input.