International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

Automation of Assessment and Approximation of Software Reliability

Bonthu Kotaiah[1]& R.A. Khan[2]

[1] Assistant Professor, Maulana Azad National Urdu University, Hyderabad, India
[2] Professor, Babasaheb Bhimrao Ambedkar University, Lucknow, India
[1]kotaiah_bonthuklce@yahoo.com, [2]khanraees@yahoo.com

ABSTRACT
Software reliability is defined as the probability of software to deliver correct service over a period of time under a specified environment. This is becoming more and more important in various software organizations to discover the faults that occur commonly during development process. As the demand of the software application programs increases the quality becomes higher and higher and the reliability of these software becomes more essential. Hence Software reliability is mentioned to be as the one of the important factor during development. Many analytical models were being proposed over the years for assessing the reliability of a software system and for modeling the growth trends of software reliability with different capabilities of prediction at different testing phases. In our work, a JAVA based automation program has been developed for automatic calculation of MTTR,MTTF, MTBF and Availability, Reliability. Approximation of Software Reliability to test the accuracy level is also automated and presented in our work.

1. Introduction

1.1 Software Reliability Assessment

Software Reliability is expressed in execution time but hardware reliability is expressed in clock time. [102] [103]

Wasserman in [10] gives a modern definition of reliability including time, condition, and customer satisfaction. What happens when a product cannot perform its intended function is called a failure. The following may some of the reasons of failures

1. Design Deficiencies
   - ❖ Omitting an important design feature
   - ❖ Deficiencies in product design which lead to early failures
   - ❖ The design of the process can also have deficiencies resulting in a defective product
2. Quality Control
   - ❖ Due to quality control problems, inefficient or unsuccessful products which lead to performance problems while being used
   - ❖ Possible damage to products while handling or distribution
3. Possible misuse of the product by the end user or during service

1.2 Dataset

for the process of automation, we had taken 17 programs of Glace EMR Medical Billing Software (on which I had worked previously as a Software Engineer at L Cube Innovative Solutions Pvt. Ltd.,) and find out the MTTF (Mean Time to Failure), MTTR (Mean Time to Repair) and MTBR (Mean Time Between Repair) and Software Reliability Approximated value based on the program

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

execution observations. We input these 3 values as input to input layer of Neural Network and apply sigmoid fuzzy membership function at the hidden layer of neural network and try to find out the software reliability approximated value. The previous values assessed using conventional traditional software reliability growth models and our Neuro Fuzzy systems based model are compared and we found to be our model is the promising one.

The dataset was collected from failure of 17 projects. Table I shows all the 17 projects and the information recorded based on the failure observations. The data represents a variety of applications in Glace EMR Billing and was recorded in the 2013. The application types are Patient Registration, Service Entry, Reports, Online Patient Insurance Verification applications. The attributes recorded for each software are Software Code, Type of Application, Size of Software (in Lines of Code (LOC)), Number of Failures.

Table 1: Software Reliability data project information

Software	Type of Application	Size(LOC)	No. of Failures
GE01	Patient Registration	22,300	235
GE02	Patient Registration	10,500	129
GE03	Patient Registration	9,800	34
GE04	Patient Registration	31,870	59
GE05	Patient Registration	12,400	13
GE06	Service Entry	4,870	5
GE07	Service Entry	26,490	321
GE08	Service Entry	23,400	256
GE09	Service Entry	21,700	213
GE10	Reports	10,890	117
GE11	Reports	28,740	333
GE12	Reports	36,350	375
GE13	Online Patient Insurance	61,800	821
GE14	Online Patient Insurance	34,700	354
GE15	Online Patient Insurance	39,800	383
GE16	Online Patient Insurance	43,200	412
GE17	Online Patient Insurance	44,600	451

1.3 Parameters used for Validation

The MTTF can be given as

$MTTF = \frac{1}{\lambda}$ and

$MTBF = MTTF + MTTR$

Where MTTR is the mean time of recovery defined as the average time a component takes to recover from a failure. The measures MTBF, MTTF and MTTR are usually considered to apply in

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

the case of a system operating continuously; however for a system operating on demand as is the case here, equivalent definitions apply where time is treated in discrete units [7].
MTBF = Average time between consecutive software system failures =MTTF+MTTR
MTTR = Average time taken to repair the system after the occurrence of failure.
Software Reliability = MTBF / (1+MTBF)
Availability = MTBF/(MTBF+MTTR) , is the likelihood that a software system will work at a given time.

2. Experimental Results

Table 2: Production Time Analysis for the Program Dataset

S.No	Program #	Total Production time(Hrs.)	Uptime at x1(Hrs.)	Uptime at x2(Hrs.)	Downtime at x1(Hrs.)	Downtime at x2(Hrs.)	No. of breakdowns at x1(Hrs.)	No. of breakdowns at x2(Hrs.)
1	GE01	256	216	202	40	54	3	11
2	GE02	324	260	203	64	121	9	16
3	GE03	236	168	154	68	82	2	19
4	GE04	600	450	435	150	165	16	23
5	GE05	371	300	265	71	106	13	35
6	GE06	447	430	410	17	37	15	21
7	GE07	865	560	525	305	340	10	25
8	GE08	843	615	575	228	268	4	31
9	GE09	943	720	706	223	237	17	28
10	GE10	135	85	78	50	57	4	6
11	GE11	242	130	132	112	110	36	22
12	GE12	369	240	206	129	163	24	30
13	GE13	122	68	64	54	58	23	9
14	GE14	107	72	74	35	33	6	15
15	GE15	371	265	253	106	118	18	34
16	GE16	453	370	398	83	55	21	37
17	GE17	325	285	256	40	69	27	29

3. Calculations

Total Production time= Uptime+ down time

MTBF= $\frac{\text{Total uptime (total time- total downtime)}}{\text{Number of Breakdowns}}$

(or) MTTF+ MTTR

Where,

MTTF= Mean Time to Failure (in hours/minutes/seconds).

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

MTTR= Mean Time to Repair (in hours/minutes/seconds).
MTBF= Mean Time between Failures (in hours/minutes/seconds).

$$MTTR = \frac{\text{Total downtime}}{\text{Number of breakdowns}}$$

$$MTTF = \frac{(\text{Failure at obs.1} + \text{Failure at obs.2} + \ldots + \text{Failure at obs.N})}{\text{Number of software programs under test}}$$

$$\text{Availability (For Repairable software systems)} = \frac{MTBF}{(MTBF + MTTR)}$$

Automated java program for Software Reliability calculation, Approximation based on Euler's mathematical theorem was presented in appendix-A.

Table 3: calculation of MTBF & MTTR

S.No.	Program	MTTF	MTTR	MTBF
1	GE01	0	9.12	45.18
2	GE02	0	7.336	20.78
3	GE03	0	19.158	46.05
4	GE04	0	8.25	23.51
5	GE05	0	4.25	15.32
6	GE06	0	1.447	24.09
7	GE07	0	22.05	38.5
8	GE08	0	32.82	86.14
9	GE09	0	10.791	33.784
10	GE10	0	11	17.12
11	GE11	0	4.056	4.80
12	GE12	0	5.042	8.43
13	GE13	0	4.396	5.03
14	GE14	0	4.016	8.46
15	GE15	0	4.679	11.08
16	GE16	0	2.719	14.18
17	GE17	0	1.93	9.69

Table 4: Calculation of Availability

S.No.	Program	MTTR	MTBF	Availability
1	GE01	9.12	45.18	0.832
2	GE02	7.336	20.78	0.739
3	GE03	19.158	46.05	0.706
4	GE04	8.25	23.51	0.739
5	GE05	4.25	15.32	0.783
6	GE06	1.447	24.09	0.943
7	GE07	22.05	38.5	0.635
8	GE08	32.82	86.14	0.724

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

9	GE09	10.791	33.784	0.757
10	GE10	11	17.12	0.608
11	GE11	4.056	4.80	0.543
12	GE12	5.042	8.43	0.609
13	GE13	4.396	5.03	0.533
14	GE14	4.016	8.46	0.678
15	GE15	4.679	11.08	0.703
16	GE16	2.719	14.18	0.839
17	GE17	1.93	9.69	0.833

4. Theoretical Validation

From the above section 3.11 a theoretical valuation can be done with the formula mentioned in the context. For example at the 1st Iteration

Table 5: Calculation of Reliability & its Approximation at 1st iteration

x	y(Measured Value)	f(a)=MTBF/(1+MTBF)	Approximated value= y - h * f(a)
1	45.18	0.97835	45.123
2	20.78	0.95409	20.725
3	46.05	0.97875	45.993
4	23.51	0.9592	23.454
5	15.32	0.93873	15.266
6	24.09	0.96014	24.034
7	38.5	0.97468	38.443
8	86.14	0.98852	86.083
9	33.784	0.97125	33.728
10	17.12	0.94481	17.065
11	4.80	0.82759	4.752
12	8.43	0.89396	8.3782
13	5.03	0.83416	4.9816
14	8.46	0.89429	8.4081
15	11.08	0.91722	11.027
16	14.18	0.93412	14.126
17	9.69	0.90645	9.6374

At 2nd iteration

Table 6: Calculation of Reliability & its Approximation at 2nd iteration

x	y(Measured Value)	f(a)=MTBF/(1+MTBF)	Approximated value= y - h * f(a)
1	45.123	0.97835	45.066
2	20.725	0.95409	20.67
3	45.993	0.97875	45.936
4	23.454	0.9592	23.398
5	15.266	0.93873	15.212

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

6	24.034	0.96014	23.978
7	38.443	0.97468	38.386
8	86.083	0.98852	86.026
9	33.728	0.97125	33.672
10	17.065	0.94481	17.01
11	4.752	0.82759	4.704
12	8.3782	0.89396	8.3264
13	4.9816	0.83416	4.9332
14	8.4081	0.89429	8.3562
15	11.027	0.91722	10.974
16	14.126	0.93412	14.072
17	9.6374	0.90645	9.5848

At 3rd Iteration

Table 7: Calculation of Reliability & its Approximation at 3rd iteration

x	y(Measured Value)	f(a)=MTBF/(1+MTBF)	Approximated value= y - h * f(a)
1	45.066	0.97835	45.009
2	20.67	0.95409	20.615
3	45.936	0.97875	45.879
4	23.398	0.9592	23.342
5	15.212	0.93873	15.158
6	23.978	0.96014	23.922
7	38.386	0.97468	38.329
8	86.026	0.98852	85.969
9	33.672	0.97125	33.616
10	17.01	0.94481	16.955
11	4.704	0.82759	4.656
12	8.3264	0.89396	8.2746
13	4.9332	0.83416	4.8848
14	8.3562	0.89429	8.3043
15	10.974	0.91722	10.921
16	14.072	0.93412	14.018
17	9.5848	0.90645	9.5322

At 4th Iteration

Table 8: Calculation of Reliability & its Approximation at 4th Iteration

x	y(Measured Value)	f(a)=MTBF/(1+MTBF)	Approximated value= y - h * f(a)
1	45.009	0.97835	44.952
2	20.615	0.95409	20.56
3	45.879	0.97875	45.822
4	23.342	0.9592	23.286
5	15.158	0.93873	15.104

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

6	23.922	0.96014	23.866
7	38.329	0.97468	38.272
8	85.969	0.98852	85.912
9	33.616	0.97125	33.56
10	16.955	0.94481	16.9
11	4.656	0.82759	4.608
12	8.2746	0.89396	8.2228
13	4.8848	0.83416	4.8364
14	8.3043	0.89429	8.2524
15	10.921	0.91722	10.868
16	14.018	0.93412	13.964
17	9.5322	0.90645	9.4796

At 5th Iteration

Table 9: Calculation of Reliability & its Approximation at 5th iteration

x	y(Measured Value)	f(a)=MTBF/(1+MTBF)	Approximated value= y - h * f(a)
1	44.952	0.97835	44.895
2	20.56	0.95409	20.505
3	45.822	0.97875	45.765
4	23.286	0.9592	23.23
5	15.104	0.93873	15.05
6	23.866	0.96014	23.81
7	38.272	0.97468	38.215
8	85.912	0.98852	85.855
9	33.56	0.97125	33.504
10	16.9	0.94481	16.845
11	4.608	0.82759	4.56
12	8.2228	0.89396	8.171
13	4.8364	0.83416	4.788
14	8.2524	0.89429	8.2005
15	10.868	0.91722	10.815
16	13.964	0.93412	13.91
17	9.4796	0.90645	9.427

% Reliability= (Average of Approximated vales/ Average of observed Values) x 100

In 5th iteration, we got 99.70%, so we will stop iteration process because we got good approximated % of reliability.

Overall percentage of Reliability= (23.97/ 24.04)*100=99.70

5. Future Work and Suggestions

As a future direction, one can do the automation of the software reliability by using different soft computing techniques like Fuzzy Systems, SVM(Support Vector Machine), etc.,

6. Conclusion

In our work, we try to automate the process of assessment and approximation of software reliability, based on uptime, downtime, number of breakdowns and hybrid combination of MTBF and Availability.

References

[1] M. M. T. Thwin and T. S. Quah, Eds., Application of Neural Network for Predicting Software Development Faults using Object-Oriented Design Metrics, vol. 5. Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), November 2002.

[2] N. Karunanithi and D. Whitley, "Prediction of Software Reliability Using Feed forward and Recurrent Neural Nets," in Neural Networks, 1992. IJCNN, vol.1. Baltimore, MD: IEEE, pp. 800–805, June 1992.

[3] R. Sitte, "Comparison of software-reliability-growth predictions: neural networks vs. parametric-recalibration," Reliability, IEEE Transactions, vol. 48, no. 3, pp. 285–291, September 1999.

[4] N. RajKiran and V. Ravi, "Software Reliability Prediction using Wavelet Neural Networks," in International Conference on Computational Intelligence and Multimedia Applications, vol. 1. Sivakasi, Tamil Nadu: IEEE, pp. 195 – 199, December 2007.

[5] J. H. Lo, "The Implementation of Artificial Neural Networks Applying to Software Reliability Modeling," Control and Decision Conference, 2009. CCDC '09, Chinese, pp. 4349 – 4354, June 2009.

[6] L. Zhao, J. pei Zhang, J. Yang, and Y. Chu, "Software reliability growth model based on fuzzy wavelet neural network," in 2nd International Conference on Future Computer and Communication (ICFCC), vol. 1. Wuhan: IEEE, pp. 664–668, May 2010.

[7] Quah, T. and Thwin, M. (2002), "Application of Neural Networks for Predicting Software Development Faults Using Object-Oriented Metrics" Proceedings of Ninth International Conference on Neural Information Processing (ICONIP'02), vol. 5, pp. 2312-2316.

[8] R. Sitte, "Comparison of software reliability growth prediction: Neural Networks VS Parametric recalibration," IEEE Trans. on Reliability, vol. 48, no.3, pp. 285-291, 1999.

[9] Aljahdali, S., Sheta, A., and Rine, D., "Predicting Accumulated Faults in Software Using Radial Basis Function Network", Proceedings of the ISCA 17th International Conference on Computers and their Application, 4-6, April 2002, pp. 26-29.

[10] Kaufmann A and Gupta M. M., Fuzzy mathematical models in engineering and management science, North-Holland, Amstredam, (1988).

[11] Cai K.Y., System failure and fuzzy methodology: an introductory overview. Fuzzy Sels and Systems, 83 (1996) 113-133

[12] Specht, D.F., 1991. A general regression neural network. IEEE Transactions on Neural Networks 2 (6), 568-576.

[13] Kasabov, N.K., 2002. DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Transactions on fuzzy systems 10 (2).

[14] Makridakis, S., Anderson, A., Carbone, R., Fildes, R., Hibdon, M., Lewandowski, R., Newton, J., Parzen, E., Winkler, R., 1982. The accuracy of extrapolation (time series) methods: results of a forecasting competition. Journal of Forecasting 1, 111-153.

[15] Perrone, M.P., Cooper, L.N., 1993. When networks disagree: ensemble methods for hybrid neural netwoks. In: Mammone, R.J. (Ed.), Neural Networks for speech and Image processing. Chapman Hall, pp. 126142.

[16] Yu, L., Wang, S.Y., Lai, K.K., 2005. A novel non-linear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates. Computers and Operations Research 32 (10), 25232541.

Appendix- A
Automation of Calculation and Approximation of Software Reliability

Source Code Files

1. main.java

```
package org.pra.ui;
import com.jtattoo.plaf.bernstein.BernsteinLookAndFeel;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;

public class Main extends javax.swing.JFrame {
   /**
    * Creates new form Main
    */
   public Main() {
      try {
UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
UIManager.setLookAndFeel(new BernsteinLookAndFeel());
initComponents();
this.setLocationRelativeTo(null);
 } catch (ClassNotFoundException ex) {
         Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
      } catch (InstantiationException ex) {
         Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
      } catch (IllegalAccessException ex) {
         Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
      } catch (UnsupportedLookAndFeelException ex) {
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel2 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jLabel2.setFont(new java.awt.Font("Bookman Old Style", 0, 18)); // NOI18N
        jLabel2.setText("Welcome to the Software Reliability Calculator");
        jButton1.setText("Click Here");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
 layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addContainerGap(47, Short.MAX_VALUE)
                .addComponent(jLabel2)
                .addGap(37, 37, 37))
            .addGroup(layout.createSequentialGroup()
                .addGap(200, 200, 200)
                .addComponent(jButton1)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(87, 87, 87)
                .addComponent(jLabel2)
                .addGap(69, 69, 69)
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
            .addComponent(jButton1)
            .addContainerGap(104, Short.MAX_VALUE))
        );
        pack();
    }// </editor-fold>
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        dispose();
        login mm = new login();
        mm.setVisible(true);
        setSize(350, 250);
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex)
{ java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex)
{

java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
        //</editor-fold>
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Main().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel2;
    // End of variables declaration
}
```

2. login.java

```
package org.pra.ui;
import com.sun.java.swing.plaf.windows.WindowsClassicLookAndFeel;
import com.sun.java.swing.plaf.windows.WindowsLookAndFeel;
import java.text.ParseException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;
import javax.swing.plaf.basic.BasicLookAndFeel;
import javax.swing.plaf.metal.MetalLookAndFeel;
import javax.swing.plaf.nimbus.NimbusLookAndFeel;

public class login extends javax.swing.JFrame {
    /**
     * Creates new form login
     */
    public login() {
            initComponents();
            this.setLocationRelativeTo(null)
    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents()
        jLabel1 = new javax.swing.JLabel();
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        txtu = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        txtp = new javax.swing.JPasswordField();
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
        jLabel1.setFont(new java.awt.Font("Bookman Old Style", 0, 18)); // NOI18N
        jLabel1.setText("Software Reliability Calculator");
        jLabel2.setText("User Name");
        jLabel3.setText("Password");
        jButton1.setText("Login");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    layout.createSequentialGroup()
                .addContainerGap(96, Short.MAX_VALUE)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
    .addGroup(layout.createSequentialGroup()
    .addComponent(jLabel3)
     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(txtp, javax.swing.GroupLayout.PREFERRED_SIZE, 97,
    javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()
    .addComponent(jLabel2)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(txtu, javax.swing.GroupLayout.PREFERRED_SIZE, 97,
    javax.swing.GroupLayout.PREFERRED_SIZE))
    .addComponent(jLabel1, javax.swing.GroupLayout.Alignment.LEADING))
    .addGap(78, 78, 78))
    .addGroup(layout.createSequentialGroup()
.addGap(181, 181, 181)
.addComponent(jButton1)
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
     layout.setVerticalGroup(
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
  layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addGap(21, 21, 21)
.addComponent(jLabel1)
.addGap(52, 52, 52)
  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel2)
    .addComponent(txtu, javax.swing.GroupLayout.PREFERRED_SIZE,
  javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(44, 44, 44)
  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel3)
    .addComponent(txtp, javax.swing.GroupLayout.PREFERRED_SIZE,
  javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 41,
  Short.MAX_VALUE)
    .addComponent(jButton1)
    .addGap(57, 57, 57))
        );
    pack();
     }// </editor-fold>
     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String id = txtu.getText();
        char ch[] = txtp.getPassword();
        String p = String.valueOf(ch);
        if (id.equals("software") && p.equals("123456")) {
            dispose();
            reliabilitytable tr = new reliabilitytable();
            tr.setVisible(true);
            tr.getdata();
            setSize(350, 250);
        } else {
            JOptionPane.showMessageDialog(rootPane, "user name does't match with password",
  "Error", JOptionPane.INFORMATION_MESSAGE);
            txtu.setText("");
            txtp.setText("");
            txtu.requestFocus();
        }
     }
     /**
      * @param args the command line arguments
      */
     // Variables declaration - do not modify
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPasswordField txtp;
    private javax.swing.JTextField txtu;
    // End of variables declaration
}
```

3. percentagereliability.java

```
package org.pra.ui;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JTable;
public class percentagereliability extends javax.swing.JFrame {
    /**
     * Creates new form percentagereliability
     */
    public percentagereliability() {
        initComponents();
    }
    public percentagereliability(double averageMean, double averageApprox, double
percentageReliability) {
        initComponents();
        this.setLocationRelativeTo(null);
        jLabel5.setText(Double.toString(Math.round(averageMean * 100.0) / 100.0));
        jLabel6.setText(Double.toString(Math.round(averageApprox * 100.0) / 100.0));
        jLabel7.setText(Double.toString(Math.round(percentageReliability * 100.0) / 100.0));
    }
    public void getApproximationData() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            com.mysql.jdbc.Connection con = (com.mysql.jdbc.Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/reliability", "root", "root");
            String sql = "select * from approximation_reliability";
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
        com.mysql.jdbc.PreparedStatement ps = (com.mysql.jdbc.PreparedStatement)
con.prepareStatement(sql);
        ResultSet rs = ps.executeQuery();
        int j = 0;
        while (rs.next()) {
           j++;
        }
        String[][] sh = new String[j][15];
        rs.beforeFirst();
        j = 0;
        while (rs.next()) {
           sh[j][0] = rs.getString("approx_id");
           sh[j][1] = rs.getString("mean_time_between_failure");
           sh[j][2] = rs.getString("reliability_function");
           sh[j][3] = rs.getString("approximated_value");
           j++;
        }
        String[] cols = {"ID", "MTBF(y)", "f(a)=y/(1+y)", "APPROXIMATED VALUE"};
        jTable1.setModel(new javax.swing.table.DefaultTableModel(sh, cols));
        /**
         *
         * @param args the command line arguments
         */
      } catch (ClassNotFoundException ex) {
         Logger.getLogger(reliabilitytable.class.getName()).log(Level.SEVERE, null, ex);
      } catch (SQLException ex) {
      Logger.getLogger(reliabilitytable.class.getName()).log(Level.SEVERE, null, ex);
      }
   }
   /**
    * This method is called from within the constructor to initialize the form.
    * WARNING: Do NOT modify this code. The content of this method is always
    * regenerated by the Form Editor.
    */
   @SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">
   private void initComponents() {
      jScrollPane1 = new javax.swing.JScrollPane();
      jTable1 = new javax.swing.JTable();
      jLabel1 = new javax.swing.JLabel();
      jLabel2 = new javax.swing.JLabel();
      jLabel3 = new javax.swing.JLabel();
      jLabel4 = new javax.swing.JLabel();
      jLabel5 = new javax.swing.JLabel();
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jTable1.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        jScrollPane1.setViewportView(jTable1);
        jLabel1.setFont(new java.awt.Font("Bookman Old Style", 0, 18)); // NOI18N
        jLabel1.setText("Software Reliability Approximation");
        jLabel2.setText("Average of Measured Values:");
        jLabel3.setText("Average of Approximated Values:");
        jLabel4.setText("Percentage Reliability:");
        jButton1.setText("Back");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addGap(26, 26, 26)
                .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 115,
Short.MAX_VALUE)
                .addComponent(jLabel1)
                .addGap(187, 187, 187))
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane1))
            .addGroup(layout.createSequentialGroup()
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
            .addGap(120, 120, 120)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel2)
.addComponent(jLabel3)
.addComponent(jLabel4))
.addGap(95, 95, 95)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel7, javax.swing.GroupLayout.DEFAULT_SIZE, 63,
Short.MAX_VALUE))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
 layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(jButton1))
            .addGap(26, 26, 26)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 216,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(51, 51, 51)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 14,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(30, 30, 30)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 14,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel3))
            .addGap(26, 26, 26)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel7)
                .addComponent(jLabel4))
            .addContainerGap(23, Short.MAX_VALUE))
        );
        pack();
    }// </editor-fold>
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            // TODO add your handling code here
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/reliability",
"root", "root");
            String sql = "TRUNCATE TABLE approximation_reliability";
            com.mysql.jdbc.PreparedStatement ds = (com.mysql.jdbc.PreparedStatement)
con.prepareStatement(sql);
            ds.executeUpdate();
            dispose();
            reliabilitytable pertr = new reliabilitytable();
            pertr.setVisible(true);
            pertr.getdata();
            setSize(350, 250);
        } catch (ClassNotFoundException ex) {
 Logger.getLogger(percentagereliability.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
Logger.getLogger(percentagereliability.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    // End of variables declaration
}
```

4. reliabilitytable.java

```
package org.pra.ui;
import com.mysql.jdbc.Connection;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.event.ListSelectionEvent;

public class reliabilitytable extends javax.swing.JFrame {
    /**
     * Creates new form reliabilitytable
     */
    public reliabilitytable() {
        initComponents();
        this.setLocationRelativeTo(null);
    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jLabel1 = new javax.swing.JLabel();
        search = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        jLabel12 = new javax.swing.JLabel();
        jLabel13 = new javax.swing.JLabel();
        jLabel14 = new javax.swing.JLabel();
        jLabel15 = new javax.swing.JLabel();
        jButton2 = new javax.swing.JButton();
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
        jButton4 = new javax.swing.JButton();
        jButton5 = new javax.swing.JButton();
        jButton6 = new javax.swing.JButton();
        jButton7 = new javax.swing.JButton();
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jTable1.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        jScrollPane1.setViewportView(jTable1);
        jLabel1.setFont(new java.awt.Font("Bookman Old Style", 0, 18)); // NOI18N
        jLabel1.setText("DATA SET OF SOFTWARE SYSTEM");
        jButton1.setText("Search Program");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        jButton3.setText("Calculate New");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });
        jPanel1.setBackground(new java.awt.Color(51, 255, 255));
        jLabel3.setForeground(new java.awt.Color(255, 51, 51));
        jLabel3.setText("PROG#=>PROGRAM#");
        jLabel4.setForeground(new java.awt.Color(255, 51, 51));
        jLabel4.setText("UAX1=>UPTIME AT X1(HRS.)");
        jLabel5.setForeground(new java.awt.Color(255, 51, 51));
        jLabel5.setText("DAX1=>DOWNTIME AT X1(HRS.)");
        jLabel6.setForeground(new java.awt.Color(255, 51, 51));
        jLabel6.setText("UAX2=>UPTIME AT X2(HRS.)");
        jLabel7.setForeground(new java.awt.Color(255, 51, 51));
        jLabel7.setText("DAX2=>DOWNTIME AT X2(HRS.)");
        jLabel8.setForeground(new java.awt.Color(255, 51, 51));
        jLabel8.setText("TPT=>TOTAL PRODUCTION TIME");
```

```
        jLabel2.setForeground(new java.awt.Color(255, 51, 51));
        jLabel2.setText("NOBX1=>NO. OF BREAKDOWNS AT X1(HRS.)");
        jLabel9.setForeground(new java.awt.Color(255, 51, 51));
        jLabel9.setText("NOBX2=>NO. OF BREAKDOWNS AT X2(HRS.)");
        jLabel10.setForeground(new java.awt.Color(255, 51, 51));
        jLabel10.setText("MTTR=>MEAN TIME TO REPAIR");
        jLabel11.setForeground(new java.awt.Color(255, 51, 51));
        jLabel11.setText("MTTF=>MEAN TIME TO FAILURE");
        jLabel12.setForeground(new java.awt.Color(255, 51, 51));
        jLabel12.setText("MTBF=>MEAN TIME BETWEEN FAILURES");
        jLabel13.setForeground(new java.awt.Color(255, 51, 51));
        jLabel13.setText("AFRSS=>AVAILABILITY FOR REPAIRABLE SOFTWARE
SYSTEM");
        jLabel14.setForeground(new java.awt.Color(255, 51, 51));
        jLabel14.setText("AFNRHS=>AVAILABILITY FOR NON- REPAIRABLE HARDWARE
SYSTEM");
        jLabel15.setFont(new java.awt.Font("Bookman Old Style", 1, 18)); // NOI18N
        jLabel15.setText("ABBREVATIONS");
        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel5)
                    .addComponent(jLabel7)
                    .addComponent(jLabel4)
                    .addComponent(jLabel6))
                .addGap(28, 28, 28)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel2)
                    .addComponent(jLabel3)
                    .addComponent(jLabel9)
                    .addComponent(jLabel8))
                .addGap(62, 62, 62)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel10)
                    .addComponent(jLabel13)
                    .addComponent(jLabel11)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel12)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel14)))
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
            .addContainerGap())
         .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(65, 65, 65)
            .addComponent(jLabel15)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
      );
      jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jLabel15)
 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 17,
Short.MAX_VALUE
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
               .addComponent(jLabel10)
               .addComponent(jLabel4)
               .addComponent(jLabel3))
            .addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
               .addGroup(jPanel1Layout.createSequentialGroup()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                     .addComponent(jLabel12)
                     .addComponent(jLabel2)
                     .addComponent(jLabel6))
                  .addGap(18, 18, 18))
               .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                  .addComponent(jLabel14)
                  .addGap(7, 7, 7)))
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                  .addComponent(jLabel9)
                  .addComponent(jLabel11))
               .addComponent(jLabel5))
            .addGap(15, 15, 15)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
               .addComponent(jLabel7)
               .addComponent(jLabel8)
               .addComponent(jLabel13))
            .addContainerGap())
      );
      jButton2.setText("Exit");
      jButton2.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
```

```
        }
    });
    jButton4.setText("Refresh");
    jButton4.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton4ActionPerformed(evt);
        }
    });
    jButton5.setText("Calculate Approximated Value");
    jButton5.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton5ActionPerformed(evt);
        }
    });
    jButton6.setText("Delete program");
    jButton6.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton6ActionPerformed(evt);
        }
    });
    jButton7.setText("Delete All Records");
    jButton7.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton7ActionPerformed(evt);
        }
    });
    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
 layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
 javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addContainerGap())
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jScrollPane1)
                    .addContainerGap())
                .addGroup(layout.createSequentialGroup()
 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jButton3)
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jButton5)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)
 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jButton7)
                        .addGap(14, 14, 14)
                        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 95,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(120, 120, 120))
                    .addComponent(jLabel1))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jButton1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jButton6)
                .addGap(35, 35, 35)
                .addComponent(search, javax.swing.GroupLayout.PREFERRED_SIZE, 89,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(55, 55, 55))))
        );
        layout.setVerticalGroup(
 layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
             .addContainerGap()
             .addComponent(jLabel1)
             .addGap(25, 25, 25)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(search, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton1)
                .addComponent(jButton6))
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton3)
                .addComponent(jButton4)
                .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton5)
                .addComponent(jButton7)))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 206,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(7, 7, 7))
      );
      pack();
   }// </editor-fold>
   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      try {
         // TODO add your handling code here
         Class.forName("com.mysql.jdbc.Driver");
         java.sql.Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/reliability", "root", "root");
         String name = search.getText();
         if (name.equals("")) {
            JOptionPane.showMessageDialog(rootPane, "Please enter program name for searching",
"Info", JOptionPane.ERROR_MESSAGE);
         } else {

            String sql = "select * from reliabiltity_availbility where program_name='" + name + "'";
            PreparedStatement ps = (PreparedStatement) con.prepareStatement(sql);
            ResultSet rs = ps.executeQuery();
            int j = 0;
            while (rs.next()) {
               j++;
            }
            String[][] sh = new String[j][15];
            rs.beforeFirst();
            j = 0;
            while (rs.next()) {
               sh[j][0] = rs.getString("id");
               sh[j][1] = rs.getString("program_name");
               sh[j][2] = rs.getString("uptime_at_x1");
               sh[j][3] = rs.getString("uptime_at_x2");
               sh[j][4] = rs.getString("downtime_at_x1");
               sh[j][5] = rs.getString("downtime_at_x2");
               sh[j][6] = rs.getString("total_production_time");
               sh[j][7] = rs.getString("no_of_breakdowns_at_x1");
               sh[j][8] = rs.getString("no_of_breakdowns_at_x2");
               sh[j][9] = rs.getString("mean_time_to_failure");
               sh[j][10] = rs.getString("mean_time_to_repair");
               sh[j][11] = rs.getString("mean_time_between_failures");
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
                sh[j][12] = rs.getString("availability1");
                sh[j][13] = rs.getString("availability2");
                sh[j][14] = rs.getString("reliability");
                j++;
            }
            jTable1.addMouseListener(new MouseAdapter() {
                public void mouseClicked(MouseEvent e) {
                    if (e.getClickCount() == 1) {
                        JTable target = (JTable) e.getSource();
                        int row = target.getSelectedRow();
                        String programId = sh[row][1];
                        dispose();
                        tableentry tr = new tableentry(programId);
                        tr.setVisible(true);
                        setSize(350, 250);
                    }
                }
            });
            String[] cols = {"ID", "PROG#", "UAX1", "UAX2", "DAX1", "DAX2", "TPT",
"NOBX1", "NOBX2", "MTTF", "MTTR", "MTBF", "AFRSS", "AFNRHS", "RELIABILITY"};
            if (sh.length == 0) {
                jTable1.setModel(new javax.swing.table.DefaultTableModel(sh, cols));
                JOptionPane.showMessageDialog(rootPane, "No record found", "Info",
JOptionPane.ERROR_MESSAGE);
            } else {
                jTable1.setModel(new javax.swing.table.DefaultTableModel(sh, cols));
            }
        }
    } catch (ClassNotFoundException ex) {
    Logger.getLogger(reliabilitytable.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {

   Logger.getLogger(reliabilitytable.class.getName()).log(Level.SEVERE, null, ex);
    }
}
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    dispose();
    tableentry tr = new tableentry();
    tr.setVisible(true);
    setSize(350, 250);
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
        shutdownSoftware();
    }
    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.getdata();
    }
    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            // TODO add your handling code here:.
            Class.forName("com.mysql.jdbc.Driver");
            com.mysql.jdbc.Connection con = (com.mysql.jdbc.Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/reliability", "root", "root");
            String sql = "select * from reliabiltity_availbility";
            com.mysql.jdbc.PreparedStatement ps = (com.mysql.jdbc.PreparedStatement)
con.prepareStatement(sql);
            ResultSet rs = ps.executeQuery();
            int j = 0;
            while (rs.next()) {
                j++;
            }
            double values = j;
            String[][] sh = new String[j][15];
            rs.beforeFirst();
            int k = 0;
            while (rs.next()) {
                sh[k][11] = rs.getString("mean_time_between_failures");
                double meanTimeBetweenFailure = Double.parseDouble(sh[k][11]);
                double h = 1 / values;
                double reliabilityFunction = meanTimeBetweenFailure / (1 + meanTimeBetweenFailure);
                double approximatedValue = meanTimeBetweenFailure - (h * reliabilityFunction);
                sumMeanTime = sumMeanTime + meanTimeBetweenFailure;
                sumApproximated = sumApproximated + approximatedValue;
                String mean_time_between_failure =
Double.toString(Math.round(meanTimeBetweenFailure * 100.0) / 100.0);
                String reliability_function = Double.toString(Math.round(reliabilityFunction * 100.0) /
100.0);
                String approximated_value = Double.toString(Math.round(approximatedValue * 100.0) /
100.0);
                String sql1 = "insert into approximation_reliability(mean_time_between_failure,
reliability_function, approximated_value) values (?,?,?)";
                com.mysql.jdbc.PreparedStatement ps1 = (com.mysql.jdbc.PreparedStatement)
con.prepareStatement(sql1);
                ps1.setString(1, mean_time_between_failure);
                ps1.setString(2, reliability_function);
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
            ps1.setString(3, approximated_value);
            int r = ps1.executeUpdate();
            k++;
          }
          double AverageMean = sumMeanTime / Double.parseDouble(Integer.toString(k));
          double AverageApprox = sumApproximated / Double.parseDouble(Integer.toString(k));
          double percentageReliability = (AverageApprox / AverageMean) * 100;
          dispose();
          percentagereliability pr = new percentagereliability(AverageMean, AverageApprox,
percentageReliability);
          pr.setVisible(true);
          pr.getApproximationData();
        } catch (ClassNotFoundException ex) {
        Logger.getLogger(reliabilitytable.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
         Logger.getLogger(reliabilitytable.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if (JOptionPane.showConfirmDialog(this, "Do you realy want to delete?", "Confirmation!",
JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION) {
                try {
          // TODO add your handling code here
          Class.forName("com.mysql.jdbc.Driver");
          java.sql.Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/reliability", "root", "root");
          String name = search.getText();
          String sql = "delete from  reliabiltity_availbility where program_name='" + name + "'";
          com.mysql.jdbc.PreparedStatement ds = (com.mysql.jdbc.PreparedStatement)
con.prepareStatement(sql);
          ds.executeUpdate();
          JOptionPane.showMessageDialog(null, "deleted successfully.....", "DELETE",
JOptionPane.INFORMATION_MESSAGE);
          this.getdata();
          search.setText(null);
        } catch (ClassNotFoundException ex) {
         Logger.getLogger(reliabilitytable.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
        Logger.getLogger(reliabilitytable.class.getName()).log(Level.SEVERE, null, ex);
        }
        }
    }
    private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
        // TODO add your handling code here:
         if (JOptionPane.showConfirmDialog(this, "Do you realy want to delete all the records?",
"Confirmation!", JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION) {
                    try {
              // TODO add your handling code here
              Class.forName("com.mysql.jdbc.Driver");
              java.sql.Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/reliability", "root", "root");
              String name = search.getText();
              String sql = "TRUNCATE TABLE reliabiltity_availbility";
              com.mysql.jdbc.PreparedStatement ds = (com.mysql.jdbc.PreparedStatement)
con.prepareStatement(sql);
              ds.executeUpdate();
              this.getdata();
              JOptionPane.showMessageDialog(null, "deleted successfully.....", "DELETE",
JOptionPane.INFORMATION_MESSAGE);
          } catch (ClassNotFoundException ex) {
           Logger.getLogger(reliabilitytable.class.getName()).log(Level.SEVERE, null, ex);
          } catch (SQLException ex) {
           Logger.getLogger(reliabilitytable.class.getName()).log(Level.SEVERE, null, ex);
          }
          }
    }
    private void shutdownSoftware() {
        if (JOptionPane.showConfirmDialog(this, "Do you realy want to quit?", "Confirmation!",
JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION) {
            setVisible(false);
            dispose();
            System.exit(0);
        }
    }
    public void getdata() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/reliability", "root", "root");
            String sql = "select * from reliabiltity_availbility";
            PreparedStatement ps = (PreparedStatement) con.prepareStatement(sql);
            ResultSet rs = ps.executeQuery();
            int j = 0;
            while (rs.next()) {
                j++;
            }
            String[][] sh = new String[j][15];
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
            rs.beforeFirst();
            j = 0;
            int i = j;
            while (rs.next()) {
                sh[j][0] = Integer.toString(++i);
                sh[j][1] = rs.getString("program_name");
                sh[j][2] = rs.getString("uptime_at_x1");
                sh[j][3] = rs.getString("uptime_at_x2");
                sh[j][4] = rs.getString("downtime_at_x1");
                sh[j][5] = rs.getString("downtime_at_x2");
                sh[j][6] = rs.getString("total_production_time");
                sh[j][7] = rs.getString("no_of_breakdowns_at_x1");
                sh[j][8] = rs.getString("no_of_breakdowns_at_x2");
                sh[j][9] = rs.getString("mean_time_to_failure");
                sh[j][10] = rs.getString("mean_time_to_repair");
                sh[j][11] = rs.getString("mean_time_between_failures");
                sh[j][12] = rs.getString("availability1");
                sh[j][13] = rs.getString("availability2");
                sh[j][14] = rs.getString("reliability");
                j++;
            }
            String[] cols = {"ID", "PROG#", "UAX1", "UAX2", "DAX1", "DAX2", "TPT", "NOBX1",
"NOBX2", "MTTF", "MTTR", "MTBF", "AFRSS", "AFNRHS", "RELIABILITY"};
            jTable1.setModel(
                    new javax.swing.table.DefaultTableModel(sh, cols));
            /**
             *
             * @param args the command line arguments
             */
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(reliabilitytable.class
                    .getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(reliabilitytable.class
                    .getName()).log(Level.SEVERE, null, ex);
        }
    }
    double sumMeanTime = 0;
    double sumApproximated = 0;
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
```

```
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JButton jButton7;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    private javax.swing.JTextField search;
    // End of variables declaration
}
```

5. tableentry.java

```
package org.pra.ui;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
public class tableentry extends javax.swing.JFrame {
    /**
     * Creates new form tableentry
     */
    public tableentry() {
        initComponents();
        this.setLocationRelativeTo(null);
    }
    public tableentry(String programId) {
```

```
        this.programId = programId;
        try {
            initComponents();
            Class.forName("com.mysql.jdbc.Driver");
            java.sql.Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/reliability", "root", "root");
            String value = programId;
            String sql = "select * from reliabiltity_availbility where program_name=?";
            com.mysql.jdbc.PreparedStatement ps = (com.mysql.jdbc.PreparedStatement)
con.prepareStatement(sql);
            ps.setString(1, value);
            ResultSet rs = ps.executeQuery();
            if (rs.next()) {
                String add1 = rs.getString("program_name");
                prog.setText(add1);
                String add2 = rs.getString("uptime_at_x1");
                uptimeAtx1.setText(add2);
                String add3 = rs.getString("uptime_at_x2");
                uptimeAtx2.setText(add3);
                String add4 = rs.getString("downtime_at_x1");
                downtimeAtx1.setText(add4);
                String add5 = rs.getString("downtime_at_x2");
                downtimeAtx2.setText(add5);
                String add6 = rs.getString("no_of_breakdowns_at_x1");
                noOfBreakdownsAtx1.setText(add6);
                String add7 = rs.getString("no_of_breakdowns_at_x2");
                noOfBreakdownsAtx2.setText(add7);
            }
            this.setLocationRelativeTo(null);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(tableentry.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
        Logger.getLogger(tableentry.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
        jLabel1 = new javax.swing.JLabel();
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
        jLabel2 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        prog = new javax.swing.JTextField();
        uptimeAtx1 = new javax.swing.JTextField();
        uptimeAtx2 = new javax.swing.JTextField();
        downtimeAtx1 = new javax.swing.JTextField();
        downtimeAtx2 = new javax.swing.JTextField();
        noOfBreakdownsAtx1 = new javax.swing.JTextField();
        noOfBreakdownsAtx2 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
        jLabel1.setFont(new java.awt.Font("Bookman Old Style", 0, 18)); // NOI18N
        jLabel1.setText("Enter Values for Reliability Calculation");
        jLabel2.setText("Program #");
        jLabel4.setText("Uptime at x1 (Hrs.)");
        jLabel5.setText("Uptime at x2 (Hrs.)");
        jLabel6.setText("Downtime at x1 (Hrs.)");
        jLabel7.setText("Downtime at x2 (Hrs.)");
        jLabel8.setText("No. of Breakdowns at x1 (Hrs.)");
        jLabel9.setText("No. of Breakdowns at x2 (Hrs.)");
        jButton1.setText("Sumit");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        jButton2.setText("Back");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });
        jButton3.setText("Reset");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
        });
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap(33, Short.MAX_VALUE)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(jLabel1, javax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(layout.createSequentialGroup()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel2)
                            .addComponent(jLabel4)
                            .addComponent(jLabel5)
                            .addComponent(jLabel6)
                            .addComponent(jLabel7)
                            .addComponent(jLabel8)
                            .addComponent(jLabel9))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                            .addComponent(noOfBreakdownsAtx2,
javax.swing.GroupLayout.DEFAULT_SIZE, 105, Short.MAX_VALUE)
                            .addComponent(prog)
                            .addComponent(uptimeAtx1)
                            .addComponent(uptimeAtx2)
                            .addComponent(downtimeAtx1)
                            .addComponent(downtimeAtx2)
                            .addComponent(noOfBreakdownsAtx1))))
                .addGap(26, 26, 26))
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jButton2))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(104, 104, 104)
                        .addComponent(jButton1)
                        .addGap(60, 60, 60)
                        .addComponent(jButton3)))
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
 layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jButton2)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 12,
Short.MAX_VALUE)
            .addComponent(jLabel1)
            .addGap(26, 26, 26)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel2, javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(prog, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(29, 29, 29)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel4)
                .addComponent(uptimeAtx1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(23, 23, 23)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel5)
                .addComponent(uptimeAtx2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(23, 23, 23)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel6)
                .addComponent(downtimeAtx1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(downtimeAtx2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel7))
            .addGap(23, 23, 23)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel8)
                .addComponent(noOfBreakdownsAtx1,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(noOfBreakdownsAtx2,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel9))
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
            .addGap(28, 28, 28)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton1)
                .addComponent(jButton3))
            .addGap(28, 28, 28))
        );
        pack();
    }// </editor-fold>
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

        try {
            // TODO add your handling code here:
            String program_name = prog.getText();
            String uptime_at_x1 = uptimeAtx1.getText();
            String uptime_at_x2 = uptimeAtx2.getText();
            String downtime_at_x1 = downtimeAtx1.getText();
            String downtime_at_x2 = downtimeAtx2.getText();
            String no_of_breakdowns_at_x1 = noOfBreakdownsAtx1.getText();
            String no_of_breakdowns_at_x2 = noOfBreakdownsAtx2.getText();
            if (program_name.isEmpty() && uptime_at_x1.isEmpty() && uptime_at_x2.isEmpty() &&
downtime_at_x1.isEmpty() && downtime_at_x2.isEmpty() &&
no_of_breakdowns_at_x1.isEmpty() && no_of_breakdowns_at_x2.isEmpty()) {
                JOptionPane.showMessageDialog(this, "Please fill all the fields", "Info",
JOptionPane.ERROR_MESSAGE);
            } else {
                if (failureMeanTime.equals("")) {
                    failureMeanTime = JOptionPane.showInputDialog("Enter number of software
programs under test for MTTF calculation");
                }
                double uptime1 = Double.parseDouble(uptime_at_x1);
                double uptime2 = Double.parseDouble(uptime_at_x2);
                double downtime1 = Double.parseDouble(downtime_at_x1);
                double downtime2 = Double.parseDouble(downtime_at_x2);
                double totalProductionTime1 = uptime1 + downtime1;
                double totalProductionTime2 = uptime2 + downtime2;
                if (totalProductionTime1 != totalProductionTime2) {
                    JOptionPane.showMessageDialog(rootPane, "Uptime at x1(Hrs.) and Downtime at
x1(Hrs.) must be equal to Uptime at x2(Hrs.) and Downtime at x2(Hrs.)", "Info",
JOptionPane.ERROR_MESSAGE);
                }else{
                String total_production_time = Double.toString(totalProductionTime1);
                double numberOfSoftwareProgramsUnderTest = Double.parseDouble(failureMeanTime);
                double meanTimeToFailure = (downtime1 + downtime2) /
numberOfSoftwareProgramsUnderTest;
```

```
String mean_time_to_failure = Double.toString(Math.round(meanTimeToFailure *
100.0) / 100.0);
double breakDownAtx1 = Double.parseDouble(no_of_breakdowns_at_x1);
double breakDownAtx2 = Double.parseDouble(no_of_breakdowns_at_x2);
double meanTimeToRepair = (downtime1 + downtime2) / (breakDownAtx1 +
breakDownAtx2);
double meanTimeBetweenFailure = meanTimeToFailure + meanTimeToRepair;
String mean_time_to_repair = Double.toString(Math.round(meanTimeToRepair * 100.0)
/ 100.0);
String mean_time_between_failures =
Double.toString(Math.round(meanTimeBetweenFailure * 100.0) / 100.0);
double availb1 = meanTimeBetweenFailure / (meanTimeBetweenFailure +
meanTimeToRepair);
double availb2 = meanTimeToFailure / (meanTimeToFailure + meanTimeToRepair);
double relia = meanTimeBetweenFailure / (1 + meanTimeBetweenFailure);
String availability1 = Double.toString(Math.round(availb1 * 100.0) / 100.0);
String availability2 = Double.toString(Math.round(availb2 * 100.0) / 100.0);
String reliability = Double.toString(Math.round(relia * 100.0) / 100.0);
Class.forName("com.mysql.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/reliability", "root", "root");
if (programId.equals("")) {
   String sql = "insert into reliabiltity_availbility (program_name, uptime_at_x1,
uptime_at_x2,
downtime_at_x1,downtime_at_x2,no_of_breakdowns_at_x1,no_of_breakdowns_at_x2,mean_time
_to_failure,mean_time_to_repair,mean_time_between_failures,total_production_time,availability1,
availability2,reliability) values (?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
    PreparedStatement ps = (PreparedStatement) con.prepareStatement(sql);
    ps.setString(1, program_name);
    ps.setString(2, uptime_at_x1);
    ps.setString(3, uptime_at_x2);
    ps.setString(4, downtime_at_x1);
    ps.setString(5, downtime_at_x2);
    ps.setString(6, no_of_breakdowns_at_x1);
    ps.setString(7, no_of_breakdowns_at_x2);
    ps.setString(8, mean_time_to_failure);
    ps.setString(9, mean_time_to_repair);
    ps.setString(10, mean_time_between_failures);
    ps.setString(11, total_production_time);
    ps.setString(12, availability1);
    ps.setString(13, availability2);
    ps.setString(14, reliability);
    int r = ps.executeUpdate();
    if (r > 0) {
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

```
                JOptionPane.showMessageDialog(rootPane, "submitted successfully", "INSERT",
JOptionPane.INFORMATION_MESSAGE);
                prog.setText(null);
                uptimeAtx1.setText(null);
                uptimeAtx2.setText(null);
                downtimeAtx1.setText(null);
                downtimeAtx2.setText(null);
                noOfBreakdownsAtx1.setText(null);
                noOfBreakdownsAtx2.setText(null);
                prog.requestFocus();
              } else {
                JOptionPane.showMessageDialog(rootPane, "error occurred", "INSERT",
JOptionPane.INFORMATION_MESSAGE);
              }
            } else {
              String sql1 = "update reliabiltity_availbility set program_name='" + program_name +
"', uptime_at_x1='" + uptime_at_x1 + "', uptime_at_x2='" + uptime_at_x2 + "', downtime_at_x1='"
+ downtime_at_x1 + "',downtime_at_x2='" + downtime_at_x2 + "',no_of_breakdowns_at_x1='" +
no_of_breakdowns_at_x1 + "',no_of_breakdowns_at_x2='" + no_of_breakdowns_at_x2 +
"',mean_time_to_failure='" + mean_time_to_failure + "',mean_time_to_repair='" +
mean_time_to_repair + "',mean_time_between_failures='" + mean_time_between_failures +
"',total_production_time='" + total_production_time + "',availability1='" + availability1 +
"',availability2='" + availability2 + "',reliability='" + reliability + "' where id='" + programId + "'";
              com.mysql.jdbc.PreparedStatement ds = (com.mysql.jdbc.PreparedStatement)
con.prepareStatement(sql1);
              int r = ds.executeUpdate();
              if (r > 0) {
                JOptionPane.showMessageDialog(rootPane, "Updated successfully", "INSERT",
JOptionPane.INFORMATION_MESSAGE);
                prog.setText(null);
                uptimeAtx1.setText(null);
                uptimeAtx2.setText(null);
                downtimeAtx1.setText(null);
                downtimeAtx2.setText(null);
                noOfBreakdownsAtx1.setText(null);
                noOfBreakdownsAtx2.setText(null);
                prog.requestFocus();
              } else {
                JOptionPane.showMessageDialog(rootPane, "error occurred", "INSERT",
JOptionPane.INFORMATION_MESSAGE);
              }
            }
          }
          }
```

```
    } catch (ClassNotFoundException | SQLException ex) {
   Logger.getLogger(tableentry.class.getName()).log(Level.SEVERE, null, ex);
    }
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    dispose();
    reliabilitytable rt = new reliabilitytable();
    rt.setVisible(true);
    rt.getdata();
    setSize(1000, 250);
}
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    prog.setText(null);
        uptimeAtx1.setText(null);
        uptimeAtx2.setText(null);
        downtimeAtx1.setText(null);
        downtimeAtx2.setText(null);
        noOfBreakdownsAtx1.setText(null);
        noOfBreakdownsAtx2.setText(null);
        prog.requestFocus();
}
public String failureMeanTime = "";
public String programId = "";
// Variables declaration - do not modify
private javax.swing.JTextField downtimeAtx1;
private javax.swing.JTextField downtimeAtx2;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JTextField noOfBreakdownsAtx1;
private javax.swing.JTextField noOfBreakdownsAtx2;
private javax.swing.JTextField prog;
private javax.swing.JTextField uptimeAtx1;
private javax.swing.JTextField uptimeAtx2;
// End of variables declaration
```

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

}

DATABASE STRUCTURE:

Database: `reliability`

TABLES

1. Table structure for table `reliabiltity_availbility`

```
CREATE TABLE IF NOT EXISTS `reliabiltity_availbility` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `program_name` varchar(255) NOT NULL,
  `uptime_at_x1` varchar(255) NOT NULL,
  `uptime_at_x2` varchar(255) NOT NULL,
  `downtime_at_x1` varchar(255) NOT NULL,
  `downtime_at_x2` varchar(255) NOT NULL,
  `no_of_breakdowns_at_x1` varchar(255) NOT NULL,
  `no_of_breakdowns_at_x2` varchar(255) NOT NULL,
  `mean_time_to_failure` varchar(255) NOT NULL,
  `mean_time_to_repair` varchar(255) NOT NULL,
  `mean_time_between_failures` varchar(255) NOT NULL,
  `total_production_time` varchar(255) NOT NULL,
  `availability1` varchar(255) NOT NULL,
  `availability2` varchar(255) NOT NULL,
  `reliability` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 AUTO_INCREMENT=18 ;
```

2. Table structure for table `approximation_reliability`

```
CREATE TABLE IF NOT EXISTS `approximation_reliability` (
 `approx_id` int(11) unsigned NOT NULL AUTO_INCREMENT,
 `mean_time_between_failure` varchar(255) NOT NULL,
  `reliability_function` varchar(255) NOT NULL,
   `approximated_value` varchar(255) NOT NULL,
 PRIMARY KEY (`approx_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

Hardware/Software Requirements:

1. Netbeans 8.
2. Java JDK 1.8.
3. MySQL 5.5 / MySQL Workbench 5.2 CE(Community Edition)
4. RAM : 2 GB
5. Processor : Intel Dual Core or above
6. Size on Disk : 248 KB(44 Files, 14 folders)

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

EXECUTION FLOW

Database Import

Goto start menu -> Choose MySQL Workbench 5.2 CE -> Click on local instance MySQL -> click on Manage Import/Export -> select server as Local MySQL -> click ok -> give the password as 'root' -> Workbench will be loaded -> click on data import/restore -> click on 'import from self contained file' and choose path of .sql file of our program ->click on start import -> your database will be imported into MySQL workbench with name as 'reliability'.

PROJECT CONFIGURATION IN NETBEANS

Go to start menu -> from the programs choose NetBeansIDE 8.0 -> IDE will be loaded -> click on File -> click on open project -> browse for the project folder -> open project -> make sure that the libraries like 'mysql-connector-java-5.1.6-bin.jar' and 'JTattoo-1.6.11.jar' are properly exists or not. ->Right click on project folder -> choose either build or clan and build option for creating project '.jar' file in project 'dist' folder -> if you want to run the project, right click on project and choose run or press 'F5' from the keyboard -> our program will be opened.

PROJECT EXECUTION SEQUENCE

When you run the program, first screen you will get is main screen as shown in the below figure 1.

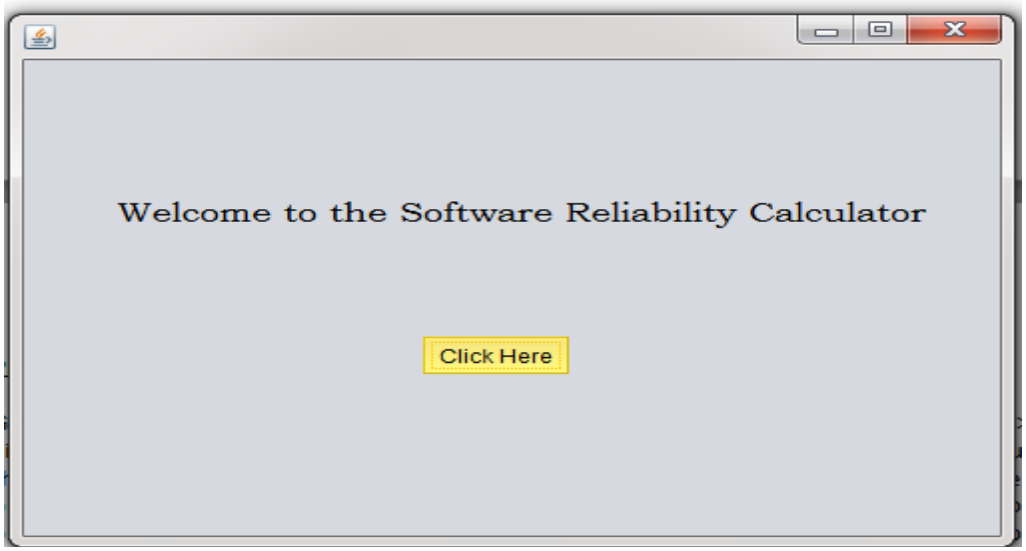


Figure 1: Welcome Screen

Click on 'click here' option, then you will get login screen as shown in the figure 2 below.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

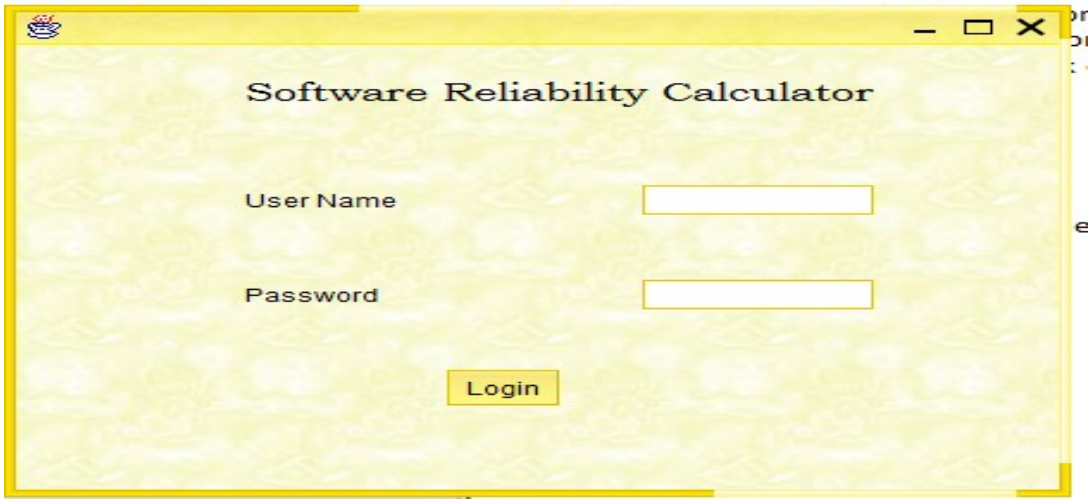
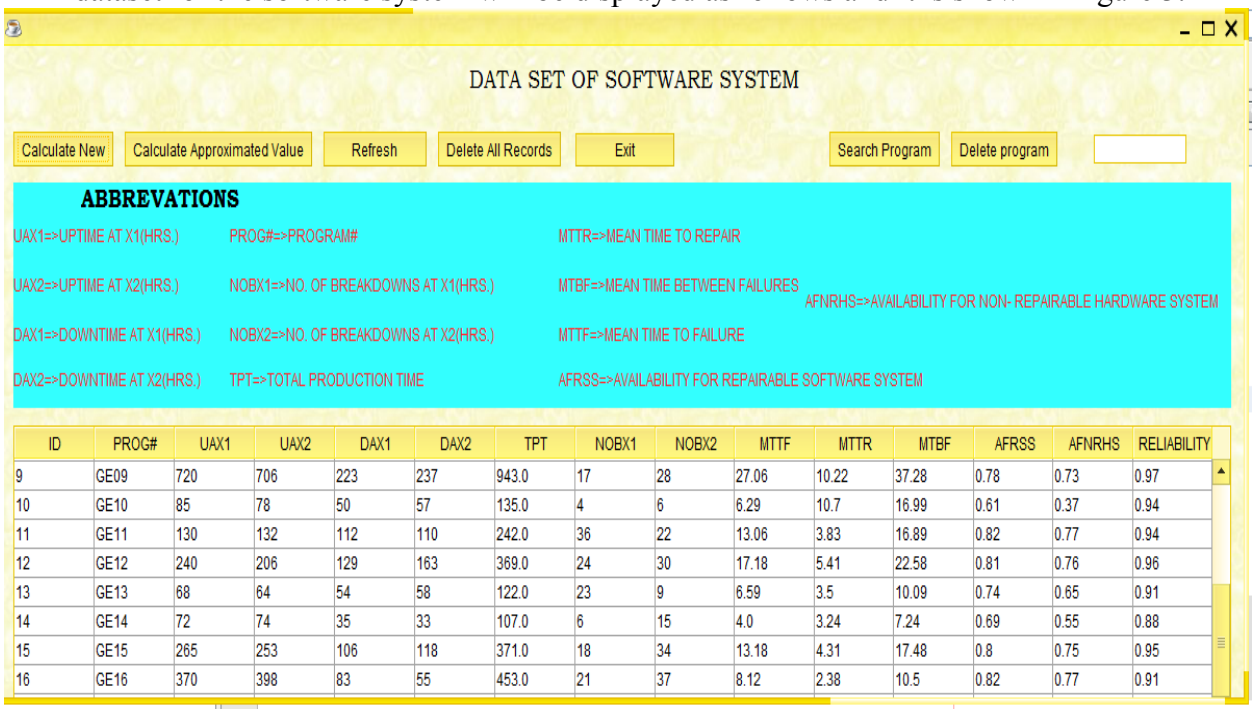


Figure 2: Login Screen

Supply the user name as 'software' and password as '123456'. Then you will get dataset for the software system will be displayed as follows and it is shown in figure 3.



ID	PROG#	UAX1	UAX2	DAX1	DAX2	TPT	NOBX1	NOBX2	MTTF	MTTR	MTBF	AFRSS	AFNRHS	RELIABILITY
9	GE09	720	706	223	237	943.0	17	28	27.06	10.22	37.28	0.78	0.73	0.97
10	GE10	85	78	50	57	135.0	4	6	6.29	10.7	16.99	0.61	0.37	0.94
11	GE11	130	132	112	110	242.0	36	22	13.06	3.83	16.89	0.82	0.77	0.94
12	GE12	240	206	129	163	369.0	24	30	17.18	5.41	22.58	0.81	0.76	0.96
13	GE13	68	64	54	58	122.0	23	9	6.59	3.5	10.09	0.74	0.65	0.91
14	GE14	72	74	35	33	107.0	6	15	4.0	3.24	7.24	0.69	0.55	0.88
15	GE15	265	253	106	118	371.0	18	34	13.18	4.31	17.48	0.8	0.75	0.95
16	GE16	370	398	83	55	453.0	21	37	8.12	2.38	10.5	0.82	0.77	0.91

Figure 3: Dataset for software system, showing Program no., uptime and down time, no. of breakdowns at time x1 and x2, total production time and calculated MTTF,MTTR, MTBF, Availability for both software and hardware systems, reliability.

You can search a specific program by using search option.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

Next choose calculate new option to add new dataset , then you will get a screen for adding new dataset as follows and it is shown in figure 4.

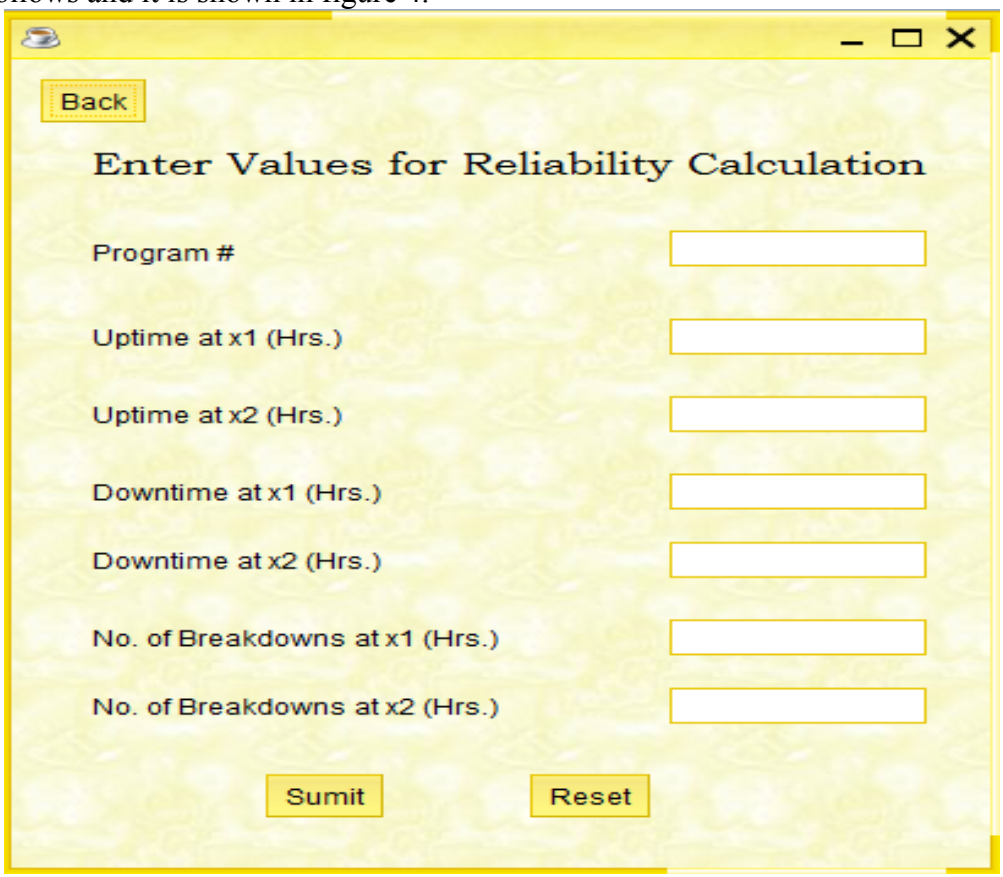


Figure 4: Dataset add, navigation, reset screen

After adding the data, click 'submit' to load the dataset into database.
click 'reset' to clear all the data from dataset entry form.
click 'back' button anytime to go and see the list of datasets and assessed values.
click on 'calculate approximated value' to get the approximated value of software reliability as per mathematical Euler's theorem based formula.
click 'exit' to come out of the program
click 'refresh' to get back to datasets when no records matched with search criteria.
click 'delete' to delete the record after searching. option 'delete all records' can be used to delete all the records of dataset from the database.

The results are shown in figure 5.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015
Available at http://internationaljournalofresearch.org

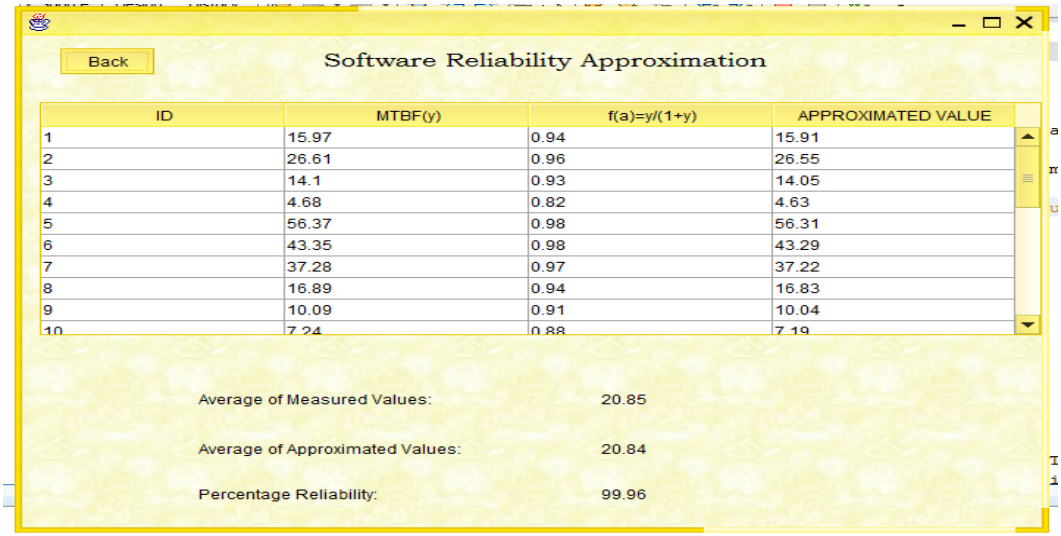


Figure 5: Approximated value of Software Reliability

Mr. Bonthu Kotaiah obtained his Bachelor's degree in Computer Applications from Nagarjuna University in 2001 and M.C.A from Nagarjuna University in 2008. During the period from September, 2001 to 2011, he has been involved in various aspects of Information Technology - an engineer(L-Cube Innovative Solutions), a Corporate Trainer (SyncSoft & Datapro(Vijayawada), COSS(Hyd.)), a Computer Programmer(Acharya Nagarjuna University). Currently he wishes to conduct research in the area of Software Engineering and Data Mining and Artifiicial Neural Networks, Fuzzy Logic & Genetic Alogorithms. His research interests include software Engineering, Neural networks. Presently, he is working as a Full-Time Research Scholar in Babasaheb Bhimrao Ambedkar University (A Central University) Lucknow, UP in the Department of Information Technology.

Dr. R.A. Khan, Presently Working with Babasaheb Bhimrao Ambedkar University (A Central University) Lucknow, UP as a **Professor & Head** in the Department of Information Technology, School for Information Science & Technology since December 2006. He is having More than **Ten Years** of teaching experience at PG and UG Level. **He obtained Master of Computer Application (M.C.A.)** from PTU, Jalandhar securing **73%** marks (**2000**) and **Ph.D.** from Jamia Millia Islamia (A Central University) New Delhi (**2004**).