CHAPTER-4
REQUIREMENT ELICITATION

4.1 Introduction:

Current practices in software development are informal and ambiguous. In traditional software development lifecycle, most of the activities involve manual intervention, so there are chances of large error rate. Hence, there is a need of such an approach which will overcome these problems in implementation of software development at various phases that includes requirement gathering, design, and coding, etc. It can be reduced by automation of these activities.

Requirement gathering, which is the most significant phase and which can affect more on software development, is the most time-consuming phase. Various phases of software development require proper coordination among themselves. If gathering of requirements is not done accurately and incorrectly prioritized, may lead to poor design, poor development of software and a system that will not satisfy the customer. Applying new methodologies in each phase of software development may improve the results and can give maximum satisfaction to customer. The proposed system suggests new more robust and fault tolerant approach which minimizes manual intervention. Proposed system is the web based tool for requirement elicitation so that gathering the requirements from all stakeholders in large scale project become easy and time-saving task. Design phase would identify and resolve the inconsistencies in system design. During the coding phase system will apply iterative code review process to detect and correct the faults present in the code document for improving the accuracy of a code.

During requirement elicitation, most of the times requirement engineer collects the requirement regarding project by conducting face to face meeting, group discussion, voting and focus groups, etc. These practices are time-consuming, and the major problem with such techniques is biasing. If the project size is large and there are a large number of requirements then collecting requirements from a large number of stakeholders is a cumbersome task. Requirement engineer uses formal language for communication with clients, which most of the time unaware of the language used by requirement engineer. The result is ambiguous requirements that directly affect the business. So there is a need of scalable and automated
approach that will be designed as the online tool so that stakeholders can directly participate during requirement elicitation phase. The requirements, which will get maximum recommendation from stakeholders, will get the highest priority, and that will directly affect the business.

Developing scalable, automated recommender system that will help to work all stakeholders in a collaborative fashion.

The proposed system is based on fuzzy approach for collaborative filtering as recommender for requirements suggested by different stakeholders. It assigning priorities to requirements so that system will be helpful for decision makers to decide which requirements would be beneficial for business with minimum risk and maximum profit with customer’s satisfaction.

4.2 Problem definition:

Develop a subsystem that is automated recommender system that will help to work with all stakeholders in a collaborative fashion and provide simpler user interface to assign priorities for various requirements using binary search tree.

4.3 Objectives:

Goals of research work are as follows:

1. To develop an automated recommender system that will help to work with all stakeholders in the collaborative fashion.
2. To provide a simpler user interface that will give information about priorities assigned to various requirements. It will easily maintain requirements in different categories like functional, nonfunctional requirements, etc.

4.4 Development of proposed module:

Proposed module work is divided in different steps which includes collection of requirements with rating, periodization of requirements and representation of requirement in binary search tree format.

1. Developing the online tool to collect requirements from all stakeholders for the project.
2. Making prediction of non-rated requirements that will be rated high by user, using User-Item rating matrix generated using step 1 and Singular Value Decomposition algorithm.


4.4.1 Layout of proposed work:

Collection of requirements from different stakeholders of the project is done and given as input to the proposed system shown in figure 4.1.

Let Ms denotes the number of stakeholders. P denotes the set of requirements. X1…..XNs denotes the user session (requirements recommended by user). Let XA represents an active user session. Let MP ⊂ P be all the requirements not yet recommended for the active user, for which system is going to provide expectations. A collaborative filter is a function f that takes as input all past user sessions and creates its result recommendation values for requirements not yet recommended for the active user.

\[ X_{Aj} = f(X_1, X_2, \ldots, X_N), \quad \forall \ j \in MP \]

**Fig. No. 4.1 Proposed Framework for Requirement Elicitation**

Use collaborative filtering as recommender engine for recommendation of requirements
1. Collect requirement from the user using what, when, which and how, etc. questions.

2. Using singular value decomposition make predictions of requirements that will be rated by user most probably.

3. Classify requirements into functional, nonfunctional and business requirement.

Prioritizing requirements in simple, unambiguous manner which will help to make the decision regarding implementation of a requirement.

Use of Binary Search Tree for prioritization of requirements.

1. All requirements must have priority.

2. Select highest priority requirement as a root node.

3. Select another requirement and compare it to the root node and place in the tree using the binary search tree technique.

4. Perform steps three until all requirements have been compared and inserted into the binary search tree.

5. For arrangement purposes, traverse through the entire BST in order and put most necessary requirements at the start of the list. Finally, put the least important requirements, at the end of the list.

4.5 Algorithms used for Requirement Elicitation:

Singular value decomposition and Binary Search Tree are used for Requirement Elicitation

4.5.1 Singular Value Decomposition (SVD):

Practically, to extract the resolution is computationally hard, as the size of E is proportional to the number of training signals. Usually careful solver is not required. The entire SVD algorithm converges to a local minimum and not a global one, and respectively, the goal of SVD is actually to increase a given primary dictionary, not detect an optimal one. The considerably faster method is to use an approximate solution

Our application uses a single iteration of alternate-optimization over the atom $d$ and the coefficients row $g^T$, which is given by

$$ d := \frac{E_g}{\|E_g\|_2} $$

$$ g := E^T d $$
This procedure finally meets to the optimum, and when reduced, supplies an estimate that still decreases the punishment term. The single iteration of this process is normally enough to offer actual close results to the full calculation.

An important benefit of this technique is that it removes the necessity to explicitly compute the matrix E. This calculation is time and memory consuming. The complete Approximate SVD implementation is given as algorithm:

1. Input: Signal set X, initial dictionary D₀, target sparsity K, number of iterations k.
2. Output: Dictionary D and sparse matrix Γ such that \( X \approx D\Gamma \)
3. Init: Set \( D \leftarrow D₀ \)
4. for \( n = 1 \ldots k \) do
5. \( \forall i : \Gamma_i \leftarrow \text{Argmin} \| x_i - D \|_2^2 \) subject To \( \| \Gamma_i \|_0 \leq K \)
6. for \( j = 1 \ldots L \) do
7. \( D_j \leftarrow 0 \)
8. \( I \leftarrow \{ \text{indices of the signals in } X \text{ whose representations use } d_j \} \)
9. \( g \leftarrow \Gamma_{j,1} \)
10. \( d \leftarrow X_{Ig} - D\Gamma_{Ig} \)
11. \( d \leftarrow d / \| d \|_2 \)
12. \( g \leftarrow X_{Ig}^T d - (D\Gamma_i)^T d \)
13. \( D_j \leftarrow d \)
14. \( \Gamma_{j,i} \leftarrow g^T \)
15. end for
16. end for

4.5.2 Algorithm Binary search tree:

A Binary Search Tree is a binary tree with a search property where the elements in the left sub-tree are less than the root and elements in the right sub-tree are greater than the root.

For example:

Inserting an element in a BST (Binary Search Tree):

To insert an element in the Binary Search Tree, we first need to find where to insert it. This can be done by traversing left or right as we did for searching for an element.
The following is the algorithm to do that.
Check if the root is present or not, if not then it’s the first element.
If the root is present then we need to find where to insert it.
Move left or right by comparing until the current node becomes null.
Once the current node becomes null, make it the child of the parent’s node.
The implementation of BST is as follows.

public void InsertNode (object data)
{
    TNode newNode = new TNode(data);
    if (root.Data == null) //First node insertion
        root = newNode;
    else
    {
        current = root;
        while (true)
        {
            tempParent = current;
            if (Convert.ToInt32(newNode.Data) < Convert.ToInt32(current.Data))
            {
                current = current.Left;
                if(current== null)
                {
                    tempParent.Left =newNode;
                    newNode.Parent =tempParent;
                    return;
                }
            }
            else
            {
                current = current.Right;
                if(current == null)
                {
                    tempParent.Right= newNode;
                    newNode.Parent =tempParent;
                    return;
                }
            }
        }
    }
}

4.6 UML Design Modeling:

4.6.1 Requirement Elicitation Module:

a) Class Diagram:
Figure 4.2 Class Diagram of Requirement Elicitation

Figure 4.2 shows various classes namely input, classification, rating, mechanism, prioritization and binary search. It also shows different attributes and methods used for different classes.

b) Sequence Diagram:
Fig. No. 4.3 Sequence Diagram of Requirement Elicitation

We have used different objects user, classify requirement, requirement prioritization and binary search tree which are shown in figure 4.3. These objects interact with each other and also shows lifeline of each object. The sequence diagram is used primarily to show the interactions between objects in a sequential order.
c) **Use Case Diagram:**

In this module use cases are used to collect requirements, classify requirements, collaborative filtering for rating mechanism, prioritize requirements, place requirements into binary search tree and make decision of implementing requirements which are shown in figure 4.4. Use case diagrams explain the interaction of any person or external device with the system which is under design process. Use case diagram displays the relationship between actors and use cases. This use case diagram shows interaction between user and different use cases of the module.

**Fig. No. 4.4 Use case Diagram of Requirement Elicitation**
d) **Activity Diagram:**

![Activity Diagram](image)

Fig. No. 4.5 Activity Diagram of Requirement Elicitation

We have used different processes viz. Collect requirements from stakeholders, classify requirements, collaborative filtering rating mechanism, assign priority to requirements and binary search tree which are shown in figure 4.5. Activity Diagram can be used to describe the dynamic aspects of the system.
e) State Chart Diagram:

![State Chart Diagram of Requirement Elicitation](image)

Fig. No. 4.6 State chart diagram of Requirement Elicitation

The various states namely collect requirements from stakeholders, classify requirements, collaborative filtering rating mechanism, assign priority to requirement and binary search tree are shown in figure 4.6. These processes interact with each other. State chart diagram describes a state machine. State chart diagrams are used to model the states and events operating on the system.
f) Component Diagram:

![Component Diagram](image)

Fig. No. 4.7 Component diagram of Requirement Elicitation

In Requirement Elicitation module different components used as input requirements from stakeholders, requirement classification, collaborative filtering for rating mechanism etc. which are shown in figure 4.7. The component diagram provides a graphical view of the dependencies and generalizations among software components. Purpose of component diagram is to describe the components used to make the functionalities.

f) Deployment Diagram: 
In deployment of the module input requirement, classify requirement, collaborative filter for rating mechanism, prioritizing requirements and binary search tree are used in shown in figure 4.8. Component diagrams are used to describe the components in the system and Deployment diagrams shows how components are deployed in the hardware.

4.7 Snapshots:

4.7.1 Requirement Elicitation Module:
1) Admin Login:

![Admin Login](image1)

**Fig. No. 4.9 Admin login**

The figure 4.9 shows login page for the administrator in requirement elicitation module.

2) Shows Login of Exiting User:

![Existing User Login](image2)

**Fig. No. 4.10 Existing user login**

The figure 4.10 shows login page for the existing user in requirement elicitation module.

3) User submits Title of Project and Its requirement:
Fig. No. 4.11 The information of the project and its submission

The figure 4.11 stakeholder submits its requirement with rating in requirement elicitation.

4) User rating for the different types of questions:

Fig. No. 4.12 Displays the questions selected by the user and its answer

The figure 4.12 stakeholder submits its answers to selected questions.
5) Shows Collaborative filtering the Information Filled by Stakeholders:

![Collaborative filtering the Information Filled by Stakeholders](image1)

The figure 4.13 shows the process of collaborative filtering in requirement elicitation.

6) Shows Questions to be asked for stakeholders:

![Questions to be asked for stakeholders](image2)

The figure 4.14 shows the list of questions in requirement elicitation.
7) Shows the information filled different stakeholders:

The figure 4.15 shows information filled by stakeholders in requirement elicitation.

8) List of titles of project filled by the different stakeholder:

The figure 4.16 shows list of title of project submitted by the stakeholders.
9) Shows the priority of requirement and reply to the question given by the stakeholder:

![Image 1](image1)

**Fig. No. 4.17** Shows the priority of requirement and reply to the question given by the stakeholder

10) Shows the priority of requirement and reply for the question given by the stakeholder:

![Image 2](image2)

**Fig. No. 4.18** Priority of requirement
11) Shows priority of the requirements in Binary Search Tree:

Fig. No. 4.19 Priority of the requirements in Binary Search Tree

Figure 4.19 shows the requirements are placed in the binary tree according to the priority.
### 4.8 Test Case:

**Table No 4.1 Test Case**

<table>
<thead>
<tr>
<th>Test Cases ID</th>
<th>Test Case Name</th>
<th>Description</th>
<th>Step Carried</th>
<th>Expected Results</th>
<th>Actual Results</th>
<th>Test Case Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC1</td>
<td>Requirement Elicitation</td>
<td>Module for collecting different requirements from stakeholders</td>
<td>User can actually enter into requirement elicitation process</td>
<td>User gets successfully entered into elicitation module</td>
<td>As expected</td>
<td>Pass</td>
</tr>
</tbody>
</table>
| TC2           | Admin Login          | Check the functionality of Admin Login                                       | 1. Start Application  
2. Enter Username  
3. Enter password.  
4. Click on Submit | 1 Admin should able to Start application and able to see Login page.  
2. Admin should able to enter Username  
3. Admin should able to enter Password  
4. For correct login, Index page should be displayed.  
5. For incorrect login, “Invalid user” message should be displayed. | As expected       | Pass             |
| TC3           | User Login           | Check the functionality of User Login                                         | 1. Start Application  
2. Enter Username  
3. Enter password.  
4. Click on Submit | 1 User should able to Start application and able to see Login page.  
2. User should able to enter Username  
3. User should able to enter Password  
4. For correct login, Index page should be displayed.  
5. For incorrect login, | As expected       | Pass             |
<table>
<thead>
<tr>
<th>Test Cases ID</th>
<th>Test Case Name</th>
<th>Description</th>
<th>Step Carried</th>
<th>Expected Results</th>
<th>Actual Results</th>
<th>Test Case Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC4</td>
<td>Requirements with its rating</td>
<td>Submission of requirements with its rating</td>
<td>1. Subject 2. Requirement 3. Rating 4. Next</td>
<td>Requirements with rating should get stored at backend</td>
<td>As expected</td>
<td>Pass</td>
</tr>
<tr>
<td>TC5</td>
<td>Questionnaires from stakeholders</td>
<td>Different questions and answers are taken from stakeholders</td>
<td>1. Select question 2. Answer of respective question 3. User selected question 4. User not selected question 5. All user data 6. Collaborative filtering</td>
<td>Answers from different stakeholders are submitted</td>
<td>As expected</td>
<td>Pass</td>
</tr>
<tr>
<td>TC6</td>
<td>BST</td>
<td>Binary search tree is constructed and displayed</td>
<td>1. Select project 2. Its requirement 3. BST</td>
<td>BST as per the priority of requirement</td>
<td>As expected</td>
<td>Pass</td>
</tr>
<tr>
<td>TC7</td>
<td>Result</td>
<td>Result Generation</td>
<td>Result as per selection of project</td>
<td>Result is generated</td>
<td>As expected</td>
<td>Pass</td>
</tr>
</tbody>
</table>

### 4.9 Result Discussion:

The proposed system is tested on a project that assigned to different groups having same technical skills. Some groups have asked to complete the project using the current methodologies of SDLC and others are have asked to complete the same project using proposed system. The group who has used current methods in SDLC has gone through the traditional approaches to requirement elicitation. It is found that, the group which has used the proposed system for the collection and prioritization of the requirements required in less time as compared to others groups those who have used the traditional approaches for the collection and prioritization of the requirements.

From the result discussion, it is clear that project completion success rate of proposed system approach is more than that of the traditional system. Also, the accuracy obtained by the proposed system is more than that of the traditional system.
Table No. 4.2 Table Success rate of proposed system and traditional system

<table>
<thead>
<tr>
<th>Project No. Tested</th>
<th>%Project Success Rate Ratio(proposed)</th>
<th>%Project Success Rate Ratio(Traditional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 1</td>
<td>90</td>
<td>75</td>
</tr>
<tr>
<td>Project 2</td>
<td>95</td>
<td>80</td>
</tr>
<tr>
<td>Project 3</td>
<td>92</td>
<td>50</td>
</tr>
<tr>
<td>Project 4</td>
<td>95</td>
<td>15</td>
</tr>
<tr>
<td>Project 5</td>
<td>90</td>
<td>35</td>
</tr>
</tbody>
</table>

Fig. No. 4.20 Success rates of proposed system and traditional system

4.10 Pseudo Code:

```csharp
using System.Collections;
using System.Text;
using System.Diagnostics;

class BST : System.Web.UI.Page
{
    string str = @"Data Source=IT-PC\SQLEXPRESS; Initial Catalog=Requirement; Integrated Security=True";

    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection(str);
        conn.Open();
        string com = "SELECT DISTINCT Subject FROM Subject_Requ";
        SqlCommand cmdm = new SqlCommand(com, conn);
        SqlDataReader drr;
        drr = cmdm.ExecuteReader();
    }
}
```
while (drr.Read())
{
    DropDownList1.Items.Add(drr[0].ToString());
}
conn.Close();

SqlConnection con = new SqlConnection(str);
con.Open();
string ss = "Delete from Bst";
SqlCommand cmd = new SqlCommand(ss, con);
cmd.ExecuteNonQuery();

protected void btnbst_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection(str);
    con.Open();
    string ss1 = "SELECT Uid, Rate FROM Subject_Requ WHERE (Subject = " + DropDownList1.SelectedItem + ") ORDER BY Rate DESC ";
    SqlCommand cmd1 = new SqlCommand(ss1, con);
    SqlDataReader dr = cmd1.ExecuteReader();
    while (dr.Read())
    {
        string no = dr[0].ToString();
        string rate = dr[1].ToString();
        SqlConnection con2 = new SqlConnection(str);
        con2.Open();
        string ss2 = "insert into Bst (Uid,Rating) values ('" + no + ":' + rate + ")";
        SqlCommand cmd2 = new SqlCommand(ss2, con2);
        cmd2.ExecuteNonQuery();
    }
    Process.Start(@"F:\Requirement\Source\BinaryTree\obj\Release\BinaryTree");
}
}

using System.Windows.Forms;

public partial class Result : System.Web.UI.Page
{
    string str = @"Data Source=IT-PC\SQLEXPRESS;Initial Catalog=Requirement;Integrated Security=True";
protected void Page_Load(object sender, EventArgs e)
{
    Panel4.Visible = false;
}

protected void LinkButton2_Click(object sender, EventArgs e)
{
    Panel4.Visible = true;
    SqlConnection con = new SqlConnection(str);
    DataTable dt = new DataTable();
    string st = "select Question from Userrating where username=" + Session["username"].ToString() + "";
    con.Open();
    SqlDataAdapter da = new SqlDataAdapter(st, con);
    da.Fill(dt);
    GridView1.DataSource = dt;
    GridView1.DataBind();
}

protected void LinkButton3_Click(object sender, EventArgs e)
{
    Panel4.Visible = true;
    SqlConnection con = new SqlConnection(str);
    DataTable dt1 = new DataTable();
    string st1 = "Select * from Userrating ";
    SqlDataAdapter da1 = new SqlDataAdapter(st1, con);
    da1.Fill(dt1);
    GridView1.DataSource = dt1;
    GridView1.DataBind();
}

protected void LinkButton4_Click1(object sender, EventArgs e)
{
    Panel4.Visible = true;
    SqlConnection con = new SqlConnection(str);
    DataTable dt2 = new DataTable();
    string st2 = "SELECT Username, Question FROM Userrating WHERE (Username != " + Session["username"].ToString() + "")";
    SqlDataAdapter da2 = new SqlDataAdapter(st2, con);
    da2.Fill(dt2);
    GridView1.DataSource = dt2;
    GridView1.DataBind();
    con.Close();
}
protected void LinkButton5_Click(object sender, EventArgs e)
{
    Panel4.Visible = true;

    // Colaborative filtering
    stringsd = "", sd1 = "", sd2 = "", sd3 = "", sd4 = "", sd5 = "", sd6 = "", sd7 = "", sd8 = "", sd9 = "", sd10 = "";
    string gg2 = "", gg1 = "";
    int cn2 = 0, cn1 = 0;

doublesss = 0.0, sss1 = 0.0, ss3 = 0.0, ss4 = 0.0, ss6 = 0.0;
SqlConnectioncono = new SqlConnection(str);
cono.Open();
//string sp4 = "Delete from Avgcound";
string sp5 = "Delete from Favg";
string sp6 = "Delete from Favg1";
string sp8 = "Delete from Average";
string sp9 = "Delete from Aques";
// SqlCommand ccc4 = new SqlCommand(sp4, cono);
SqlCommand ccc5 = new SqlCommand(sp5, cono);
SqlCommand ccc6 = new SqlCommand(sp6, cono);
SqlCommand ccc9 = new SqlCommand(sp8, cono);
SqlCommand ccc10 = new SqlCommand(sp9, cono);
// ccc4.ExecuteNonQuery();
ccc5.ExecuteNonQuery();
ccc6.ExecuteNonQuery();
ccc9.ExecuteNonQuery();
ccc10.ExecuteNonQuery();
cono.Close();
cono.Open();
stringstrs = "SELECT distinct Uid FROM Subject_ Requ ";
SqlCommandcmdsd = new SqlCommand(strs, cono);
SqlDataReaderrd = cmdsd.ExecuteReader();
while (rd.Read())
{
    sd += rd[0].ToString() + " ";
}
string[] cal = sd.Split(' '); rd.Close();
cono.Close();
cono.Open();
string strs1 = "SELECT Requirmen FROM Subject_Requ";
SqlCommand cmds1 = new SqlCommand(strs1, cono);
SqlDataReader rd1 = cmds1.ExecuteReader();
while (rd1.Read())
{
    sd1 += rd1[0].ToString() + ",";
}
string[] cal1 = sd1.Split(',');
cono.Close();
for (int i = 0; i < cal1.Length - 1; i++)
{
    SqlConnection con2 = new SqlConnection(str);
    con2.Open();
    string str4 = "SELECT Rate FROM Subject_Requ where Requriment= " + cal1[i].ToString() + ";";
    SqlCommand cmd4 = new SqlCommand(str4, con2);
    SqlDataReader rd4 = cmd4.ExecuteReader();
    try
    {
        rd4.Read();
        {
            sd2 = rd4[0].ToString();
            cn2++;
            sd8 += rd4[0].ToString() + " ";
        }
    }
    rd4.Close();
    }catch (Exception ex)
    {
        sd2 = "0";
    }
    con2.Close();
} cono.Open();
string strr = "SELECT AVG(CAST(Rate AS float)) AS Expr1 FROM Subject_Requ WHERE (Requriment= " + cal1[i].ToString() + ") ";
SqlCommand cmdr = new SqlCommand(strr, cono);
SqlDataReader rd5 = cmdr.ExecuteReader();
try
{
    rd5.Read();
    {
gg2 = rd5[0].ToString();
if (gg2 == "")
{
    gg2 = "0";
}
rd5.Close();
}
catch (Exception ex3)
{
    cono.Close();
}
double mm = 0.0, mm1 = 0.0, mm2 = 0.0;
    mm1 = Convert.ToDouble(sd2);
    mm2 = Convert.ToDouble(gg2);
    mm = Convert.ToDouble(mm1 - mm2);
    sss1 += mm * mm;
    ss4 = Math.Sqrt(sss1);
cono.Open();
string str5 = "insert into Favg (Ques,output) values ('" + cal1[i] + ":" + mm + ") ";
SqlCommand cmd5 = new SqlCommand(str5, cono);
cmd5.ExecuteNonQuery();
cono.Close();

for (int j = 1; j < cal1.Length - 1; j++)
{
    con2.Open();
    string str6 = "SELECT Rate FROM Subject_Requ where Requirement=" + cal1[j].ToString() + ":";
    SqlCommand cmd6 = new SqlCommand(str6, con2);
    SqlDataReader rd6 = cmd6.ExecuteReader();
    try
    {
        rd6.Read();
        {
            sd6 = rd6[0].ToString();
        }
        rd6.Close();
    }
catch (Exception ex1)
    {
        sd6 = "0";
    }
cono.Close();
cono.Open();
string str7 = "SELECT AVG(CAST(Rate AS float)) AS Expr1 FROM Subject_Requ WHERE (Requriment= "" + cal1[j].ToString() + ") ";
SqlCommand cmd7 = new SqlCommand(str7, cono);
SqlDataReader rd7 = cmd7.ExecuteReader();
try
    {
        rd7.Read();
        {
            gg1 = rd7[0].ToString();
            if (gg1 == ")
            {
                gg1 = "0";
            }
        }
    }
rd7.Close();
}
catch (Exception ex2)
    {
        gg1 = "0";
    }
cono.Close();

doublnnn = 0.0, nnl = 0.0, nn2 = 0.0;
nn1 = Convert.ToDouble(sd6);
nn2 = Convert.ToDouble(gg1);
nnn = Convert.ToDouble(nn1 - nn2);
    ss3 += nnn * nnn;
    ss6 = Math.Sqrt(ss3);
sss += mm * nnn;
cono.Open();
string str9 = "insert into Favg1 (Ques,output) values (" + cal1[j] + "," + nnn + ") ";
SqlCommand cmd9 = new SqlCommand(str9, cono);
    cmd9.ExecuteNonQuery();
cono.Close();

double nem = sss;

double den = ss4 * ss6;
doublesim = nem / den;
if (nem == 0)
    
else
    
cono.Open();
string str11 = "insert into Average (iQues,jQues,Avg) values ('" + cal1[i] + "," + cal1[j] + "," + sim + ")";
SqlCommand cmd11 = newSqlCommand(str11, cono);
cmd11.ExecuteNonQuery();
cono.Close();
}

SqlCommand con = newSqlConnection(str);
DataTable dt5 = newDataTable();
string stt3 = "SELECT DISTINCT TOP (5) jQues FROM Average";
SqlDataAdapter da5 = newSqlDataAdapter(stt3, con);
da5.Fill(dt5);
GridView1.DataSource = dt5;
GridView1.DataBind();
con.Close();
}
protectedvoid LinkButton6_Click(object sender, EventArgs e)
{
Response.Redirect("BST.aspx");
}
}
usingSystem.Data;
publicpartialclassAdminResult :System.Web.UI.Page
{
stringstr = @"Data Source=IT-PC\SQLEXPRESS;Initial Catalog=Requirement;Integrated Security=True";
protectedvoid Page_Load(object sender, EventArgs e)
{
    Panel3.Visible = false;
    Panel4.Visible = false;
SqlConnection con = newSqlConnection(str);
    con.Open();
    string com = "SELECT DISTINCT Subject FROM Subject_Requ";
    SqlCommandcmd = newSqlCommand(com, con);
    SqlDataReaderdr;
    dr = cmd.ExecuteReader();
    while (dr.Read())
        
    DropDownList1.Items.Add(dr[0].ToString());
}
con.Close();
}
protectedvoid LinkButton1_Click(object sender, EventArgs e)
{
    Panel3.Visible = true;
    Panel4.Visible = true;
SqlConnection co = newSqlConnection(str);
co.Open();
string ss5 = "DELETE FROM Admmax";
SqlCommand cmd2 = newSqlCommand(ss5, co);

cmd2.ExecuteNonQuery();

string ss2 = "", ss3 = "", rt = "";
string ss4 = "";
SqlConnection con = newSqlConnection(str);
con.Open();
string ss1 = "SELECT Requiment,Rating,Rate FROM Subject_Requ WHERE (Subject = "+DropDownList1.SelectedItem+");"
SqlCommand cmd = newSqlCommand(ss1, con);
SqlDataReader dr = cmd.ExecuteReader();
while (dr.Read())
{
    ss2 = dr[0].ToString();
    ss3 = dr[1].ToString();
    rt = dr[2].ToString();
    ss4 += "insert into Admmax (Requiment,Rating,Rate) values ('" + ss2 + "," + ss3 + "," + rt + ");";
}

dr.Close();

SqlCommand cmd1 = newSqlCommand(ss4, con);

cmd1.ExecuteNonQuery();

DataTable dt = newDataTable();
string grd1 = "SELECT distinct Requiment,Rating,Rate FROM Admmax ORDER BY Rate DESC";
SqlDataAdapter ad = newSqlDataAdapter(grd1, con);
ad.Fill(dt);
GridView1.DataSource = dt;
GridView1.DataBind();

string nn = "Valid";
Session["no"] = nn;
con.Close();

DataTable dt1 = newDataTable();
string grd2 = "SELECT Username, Question, Comments FROM Userrating";
SqlDataAdapter da = newSqlDataAdapter(grd2, con);
da.Fill(dt1);
GridView2.DataSource = dt1;
GridView2.DataBind();
protected void LinkButton2_Click(object sender, EventArgs e)
{
    Panel3.Visible = true;
    Panel4.Visible = true;
    SqlConnection co = new SqlConnection(str);
    co.Open();
    string ss5 = "DELETE FROM Admmax";
    SqlCommand cmd2 = new SqlCommand(ss5, co);
    cmd2.ExecuteNonQuery();
    string ss2 = ", ss3 = ", rt=";
    string ss4 = ";
    SqlConnection con = new SqlConnection(str);
    con.Open();
    //string ss1 = "SELECT Ques, Rating FROM Userrating WHERE (Rating < 3)"
    string ss1 = "SELECT Requirement,Rating,Rate FROM Subject_Requ WHERE (Subject = "+DropDownList1.SelectedItem++")"
    SqlCommand cmd = new SqlCommand(ss1, con);
    SqlDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        ss2 = dr[0].ToString();
        ss3 = dr[1].ToString();
        rt = dr[2].ToString();

        ss4 += "insert into Admmax (Requirement,Rating,Rate) values (" + ss2 + "," + ss3 + "," + rt + ");"
    }
    dr.Close();
    cmd.ExecuteNonQuery();
    SqlCommand cmd1 = new SqlCommand(ss4, con);
    cmd1.ExecuteNonQuery();
    DataTable dt = new DataTable();
    string grd1 = "SELECT distinct Requirement,Rating,Rate FROM Admmax ORDER BY Rate";
    SqlDataAdapter ad = new SqlDataAdapter(grd1, con);
    ad.Fill(dt);
    GridView1.DataSource = dt;
    GridView1.DataBind();
    string nn = "Invalid";
    Session["no"] = nn;
    con.Close();
    DataTable dt1 = new DataTable();
    string grd2 = "SELECT Username, Question,Comments FROM Userrating ";
SqlDataAdapter da = new SqlDataAdapter(grd2, con);
da.Fill(dt1);
GridView2.DataSource = dt1;
GridView2.DataBind();
con.Close();
}

protected void Button1_Click(object sender, EventArgs e)
{
SqlConnection co = new SqlConnection(str);
co.Open();
string ss4 = "DELETE FROM AdminQ";
SqlCommand cmd4 = new SqlCommand(ss4, co);
cmd4.ExecuteNonQuery();
string data = "";
string dd = Session["no"].ToString();
if (dd == "Valid")
{
foreach (GridViewRow row in GridView1.Rows)
{
if (row.RowType == DataControlRowType.DataRow)
{
CheckBox chkRow = (row.Cells[0].FindControl("Select") as CheckBox);
if (chkRow.Checked)
{
string Ques = row.Cells[1].Text;
data = data + Ques;

var textbox = row.FindControl("TextBox1") as TextBox;
var a = textbox.Text;
SqlConnection con = new SqlConnection(str);
con.Open();
string ff = "", ff1 = "";
//string ss1 = "select Uid,Rating from Userrating where Ques=" + Ques + " AND Rating>3";
string ss1 = "select Rate from Subject_Requ where Requriment=" + Ques + " AND Rate>3";
SqlCommand cmd1 = new SqlCommand(ss1, con);
SqlDataReader dr = cmd1.ExecuteReader();
dr.Read();
ff = dr[0].ToString();
// ff1 = dr[1].ToString();
dr.Close();
string ss = "insert into AdminQ (Question,Comment,Rating) values ('" + Ques + "," + textbox.Text + "," + ff + ") " ";
SqlCommand cmd = new SqlCommand(ss, con);
cmd.ExecuteNonQuery();
cmd1.ExecuteNonQuery();
con.Close();
}
}
else if (dd == "Invalid")
{
foreach (GridViewRow row in GridView1.Rows)
{
if (row.RowType == DataControlRowType.DataRow)
{
CheckBox chkRow = (row.Cells[0].FindControl("Select") as CheckBox);
if (chkRow.Checked)
4.11 Conclusion:

By automation of requirement gathering process, it is possible to collect the requirement from all stakeholders 24 x 7. By implementing collaborative filtering approach, it is possible to work with a group. System generates recommendations for requirements that are most likely for the user. It is also possible to maintain secrecy of the user that helps to remove biasing happening many times in requirement gathering process. A simple rating
mechanism help to assign priorities to requirements that makes requirement engineers life comfortable. Binary search tree implementation for priority assignment has helped to take decisions regarding which requirement should be considered first for implementation. Hence, the proposed subsystem has designed as an online tool so that stakeholders can directly participate during requirement elicitation phase, and the system has worked as recommender for participating users. Proposed system help to improve the quality of software product development by reducing product development time and with optimum resources.