Chinese College Students’ Goodwill Activity Contest Website

Hao Liu
Abstract

**Chinese College Students' Goodwill Activity Contest Website**

*Hao Liu*

Since there has been a rapid growth and popularization of the internet, websites play a more and more important role. On the one hand, for an organizer of an activity, a website supplies a more efficient and economical solution to expanded publicity and to get technical support. On the other hand, it helps participants transcend geographical limitations and join the activity more easily.

In this thesis project we developed and maintained the 2008 Google Chinese college students goodwill activity website.

This work has four highlighted features. First, it use many secure techniques, such as https, remote database to ensure data storage safety and confidential. Secondly, in order to enhance efficiency, memory and file management, scientific server configuration and other techniques are used in the implementation. Thirdly, usability theory is assessed in the process of design system logic as well as the data input form. Fourthly, popular website management tools are used, like Google analysis and Google webmaster. The result of the thesis has been approved by the activity committee and the students. We developed and maintain the web server, web program and database stably for 8 months. 619,941 visitors came from 88 countries and territories.

30000 students from 700 universities submitted 5000 applications through the website. The website provided a stable and efficient technical support for the activity. At the same time, a lot of new techniques are researched and implemented in the developing process.
Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Yuke Zhao at Google Research center for providing me the opportunity to conduct and accomplish my master thesis at Google Research. I also appreciate all his guidance and suggestions for helping me fulfill this thesis work. I would like to thank Xu Wenzhi, Pan Zhuming for all their instructions and advices around this thesis. In the mean time, I want to thank my reviewer at Uppsala University, Justin Pearson, for his comments. Last but not least, I would like to thank all the other people who helped me during the whole thesis work.
# Contents

1 Introduction ........................................................................................................... 5  
1.1 Overview ........................................................................................................ 5  
1.2 Tasks in Detail .............................................................................................. 5  
1.3 Result of Thesis work .................................................................................. 6  
2 Related works .................................................................................................. 7  
2.1 Scenario ......................................................................................................... 7  
2.2 System setting up ......................................................................................... 8  
2.2.1 System requirement analysis ............................................................... 8  
2.2.2 Web server technique research ............................................................. 8  
2.3 System architecture ..................................................................................... 10  
2.4 Data Storage ................................................................................................. 10  
2.5 Relative developing tools ................................................................ .......... 12  
2.6 Web server hardware .................................................................................. 12  
2.7 Environment set up (Detailed instruction is in the Appendix 1) ............... 13  
2.8 System Logic ................................................................................................. 13  
3 Module design and Implementation ................................................................. 13  
3.1 Homepage overview .................................................................................... 13  
3.2 Services system ............................................................................................ 14  
3.2.1 Scenario and Goal .................................................................................. 14  
3.2.2 System logic workflow .......................................................................... 15  
3.2.3 Code analysis ......................................................................................... 15  
3.3 Application system: .................................................................................... 22  
3.3.1 Scenario ................................................................................................ 22  
3.3.2 Workflow ............................................................................................... 24  
3.3.3 Function and Code analysis .................................................................. 28  
3.4 Validation system ........................................................................................ 31  
3.4.1 Scenario ................................................................................................ 31  
3.4.2 Workflow ............................................................................................... 33  
3.4.3 Code Analysis ....................................................................................... 33  
3.5 Publishing system ........................................................................................ 36  
3.5.1 Scenario ................................................................................................ 36  
3.5.2 Workflow ............................................................................................... 37  
3.5.3 Code Analysis ....................................................................................... 37  
3.6 Evaluation system ........................................................................................ 41  
3.6.1 Scenario ................................................................................................ 41  
3.6.2 Workflow ............................................................................................... 42  
3.6.3 Code Analysis ....................................................................................... 42  
4 Conclusion ......................................................................................................... 45  
5 Discussion ......................................................................................................... 46  
5.1 Limitation ..................................................................................................... 46  
5.2 Future work .................................................................................................. 48  
Appendices .......................................................................................................... 51  
Appendix 1 Installation of Apache + JK + Tomcat on Windows XP ............... 51  
Appendix 2 Hypertext Transfer Protocol over Secure Socket Layer ............... 56  
Appendix 3 MD5 ................................................................................................. 57
1 Introduction

1.1 Overview

Under the sponsorship of Dr. Li Kaifu, vice president of Google, Google Inc. (China) advocates a Chinese college students’ goodwill activity contest. Google plans to use three years, and donate 150 million US dollars to support Chinese university students to propose and achieve their Goodwill originality.

This contest needs a website to extend influence and provide web service support, such as publishing latest news and following online functions.

Services system: Solve questions and problems the students submit.
Application system: Help students submit their originality, detailed plan and budget.
Validation system: Help reviewer evaluate no infraction info in the application.
Evaluation system: Help activity committee evaluate students’ application.
Publication system: Publish evaluation result and winner teams’ project homepage.
Storage system: Combine local database and remote google database to confirm sensitive data storage security.

In the first half of 2008, I joined the development of the contest website and write this thesis. The purpose of this thesis is to develop a functional and reliable website, which can supply adequate web service to the contest. At the same time, I do research on a series of latest tools and techniques concerning web development and management, for example, use google webmaster to help website management.

1.2 Tasks in Detail

- Implement reliable, efficient website functionality
  - Set reliable web server and internet connection, confirm students from any university which use different telecom network to browse this website.
  - Design scientific system architecture and data process logic.
  - For application system, design user-friendly and scientific data input form, in order to enhance the data quality.
  - For evaluation system, design scientific and fair evaluation logic.
  - For storage system, use memory and file operate techniques make efficient data process logic to manage mass data. Combine secure remote Google database and local database to separately store sensitive personal and common data.
  - For publication system, design user friendly logic to classify and sort the large amount of applications.

- Expanded contest publicity
  - Use Google webmaster Tools to help search engine find the webpage and expand contest publicity.
  - Use Google analysis Tool to analyze the user information of the website.
1.3 Result of Thesis work

- Setup and maintain the web server, web program and database stably for 8 months.
- 619,941 visitors from 88 countries/territories visited the website
- 30000 students from 700 universities submitted 5000 applications through the website

Figure 1.1: Survey of website user distribution From Google Analysis

The remainder of the paper is structured as follows.

Chapter 2 discusses the general scenario, related techniques, environment setting up, system architecture the module design.

Chapter 3 provides the implementation of each module as well as analysis of the website logic and code.

Chapter 4 states with conclusion and summary.

Chapter 5 discusses the limitations of this thesis work and future work is discussed.
2 Related works

2.1 Scenario

In order to select the project with best originality from thousands of applications, the activity is divided into three stages: primary, intermediary and final stages. Only the team which passes each stage’s competition has the right to enter next stage. In each stage, web services are used to publish latest news and provide application, validation, evaluation and publication functions.

The website logic and the workflow of the contest

- **Primary stage**
  - Common students enter application system; fill in and submit originality form following standard format.
  - System generates an applicant account (user name and password) and team homepage. Then sends account information to applicant mail box.
  - Common data is stored in local database. In order to confirm the other personal information data security, these sensitive data will be sent to Google Database in US.
  - Activity committee uses online evaluation system to judge each team’s idea.

- **Intermediary stage**
  - Publish the result of primary competition and the winner teams’ homepages.
  - Primary winners login to application system, fill in and save detailed plan form following standard format. Team homepage will be updated with new contents. Blog link can be added to help team propagate their plan.
  - Before submission, students can edit their detailed plan as they want. After submission, homepage can’t be changed any more and will be validated.
  - Activity committee validates the submitted plan to confirm no infraction info included, and then publishes teams’ homepages.
  - Common students can visit teams’ homepage and vote for them.
  - Activity committee uses evaluation system to judge each team’s detailed plan.

- **Final stage**
  - Publish the result of Intermediary stage and the winner teams’ homepages.
Winner team logins into application system, write, edit, save and submit their
detailed budget following standard format.
Submission, validation, evaluation and publication almost similar as the
intermediary stage.

2.2 System setting up

2.2.1 System requirement analysis

Analyze and predict the characteristic and requirement of this contest website,
including the source of the visitor, website traffic demand and hardware as well as
techniques requirement.

For this activity website, visitors are mainly from universities in around 43
provinces in China. Most of them use the network provided by CERNET (China
Education and Research Network). Now there are three telecom operators in china
distinguished by geography, and the reliable interconnection between different
telecom networks is always a problem. In order to confirm that the website has
stable access to all students from the whole of china, we decided to entrust our
web server hardware to one of most popular telecom operators and make mirror
server in another two telecom operators.

Before sever entrustment, we use “Ping” and “Tracert” command to do
connection test from different university to get server connection speed and
stability statistics. We predict that the maximum daily website traffic is 50 thousand
visitors; the peal value is 5000 visitor online at the same time.

For the domain name, we go to the website http://en.hichina.com, apply the
URL “www.gong1chuang1.com” for this Google cup college students’ goodwill
activity contest website. In Chinese, “gong1” has the same pronunciation as
“goodwill”, and “chuang1” has the same pronunciation as “creative idea”. That’s the
reason why we combine those two words to make the domain.

2.2.2 Web server technique research

2.2.2.1 Comparison of several current popular web servers

- Overview
- Operation system support
- Features
- Relative Web programming techniques

<table>
<thead>
<tr>
<th>Server</th>
<th>Developed by</th>
<th>Cost (USD)</th>
<th>Open source</th>
<th>Software license</th>
<th>Last stable version</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache HTTP Server</td>
<td>Apache Software Foundation</td>
<td>Free</td>
<td>Yes</td>
<td>Apache License</td>
<td>2.2.9</td>
<td>2008-6-13</td>
</tr>
<tr>
<td>Apache Tomcat</td>
<td>Apache Software Foundation</td>
<td>Free</td>
<td>Yes</td>
<td>Apache License</td>
<td>6.0.16</td>
<td>2008-2-7</td>
</tr>
<tr>
<td>Internet Information Services (IIS)</td>
<td>Microsoft</td>
<td>Included with newer Windows NT Family products</td>
<td>No</td>
<td>proprietary</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Weblogic</td>
<td>BEA Weblogic</td>
<td>$9000+</td>
<td>No</td>
<td>proprietary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yaws</td>
<td>Claes Wikström</td>
<td>Free</td>
<td>Yes</td>
<td>BSD variant</td>
<td>1.77</td>
<td>2008-6-16</td>
</tr>
</tbody>
</table>

Table 2.1: Web server Overview
Table 2.2: Features

<table>
<thead>
<tr>
<th>Server</th>
<th>Security</th>
<th>Dynamic content[3]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>basic access authentication</td>
<td>digest access authentication</td>
</tr>
<tr>
<td>Apache HTTP Server</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Apache Tomcat</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet Information Services</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>JBoss Web</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Yaws</td>
<td>Yes</td>
<td>?</td>
</tr>
</tbody>
</table>

Table 2.3: Operation system support

<table>
<thead>
<tr>
<th>Server</th>
<th>Windows</th>
<th>Mac OS</th>
<th>Linux</th>
<th>BSD</th>
<th>Solaris</th>
<th>eCS</th>
<th>OpenVMS</th>
<th>AIX</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache HTTP Server</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Apache Tomcat</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet Information Services</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yaws</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>?</td>
</tr>
</tbody>
</table>

Table 2.4: Relative Web programming techniques

- Web technique
  - Internet Information server 5.0: ASP, ASP.NET, CGI, Python, PHP
  - Apache: PHP, CGI/Perl, Python
  - Tomcat: CGI/Perl, JSP, Servlets, JavaBeans
  - iPlanet: JSP, Servlets, Enterprise JavaBeans
  - Internet Information Services: ASP
  - Go webserver: LotusScript

2.2.2.2 Conclusion

We compared the advantages and disadvantages of several popular web servers. At last we chose to use “apache + Tomcat” technique:

- This contest website is simple and doesn’t have high demand of stability and security.

- Predicted daily traffic is about 50,000. The supervisor from google prefers the free server and doesn’t want the expensive “weblogic”.

- J2EE is chose as the context web servers technique. J2EE is a popular, easy and powerful web programming technique. It is easy to be combined with JavaBeans to implement complex functions. Also, it’s easy to find J2EE instruction and experiences on the internet. Tomcat has a good support to J2EE technique.
• Windows server 2003 is chosen as the web server operation system. Windows server 2003 is stable enough as the server. It’s easy for the beginner to install, configure and manage Tomcat and Apache.

• Apache is good at process static WebPages, while Tomcat is good at process dynamic WebPages. The combination of those two servers is a powerful solution to analyze WebPages. Finally we chose “apache_2.0.54” and “apache-tomcat-5.5.17”.

2.3 System architecture

![System architecture diagram]

2.4 Data Storage.

In the activity, we store all application data for each project, including common data and some sensitive and important personal information. Activity committee and Google paid a lot of attention to the storage security of those data. The storage system contains two subsystems.

• Local storage subsystem
  All common data, such as project plan and budget info, will be stored in local MySQL database. Three parts are included:
  o local database
  o local database connection
  o SQLyog database management tool

• Remote storage subsystem
  Remote U.S. Google security database is used to store all sensitive and important personal information, such as applicants’ name, contact info, and so on. After sensitive data is submitted to the system, they will be temporarily stored
in the local database “person” table. Every night system administrator will manually export “person” table to csv file and send it to the administrator of U.S Google database. Afterwards google engineer will import this csv file to google database. Two parts are included:

- Remote database,
- Mail function for remote database.

The reason why we choose MySQL as database is MySQL has become the world's most popular open source database based on its consistent fast performance, high reliability and ease of use. We choose mysql-5.2.0 as our local database and use SQLyog Enterprise V6 as the database use interface tool.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Prime key</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMIN</td>
<td>Admin information, for validation and evaluation system</td>
<td>Admin_ID</td>
</tr>
<tr>
<td>AREA</td>
<td>Area information about project implement</td>
<td>A_ID</td>
</tr>
<tr>
<td>BUDGET</td>
<td>Budget information</td>
<td>P_ID</td>
</tr>
<tr>
<td>COOPERATION</td>
<td>Cooperation partner information</td>
<td>CO_ID</td>
</tr>
<tr>
<td>FEEDBACK</td>
<td>Contact info</td>
<td></td>
</tr>
<tr>
<td>GRADE</td>
<td>Students grade info</td>
<td>G_ID</td>
</tr>
<tr>
<td>EVALUATION</td>
<td>Result of the evaluation</td>
<td>T_ID</td>
</tr>
<tr>
<td>PERSON</td>
<td>Students’ personal information</td>
<td>I_ID</td>
</tr>
<tr>
<td></td>
<td>Sensitive data about private information</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Temporally store in local database</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Daily export and send to remote database</td>
<td></td>
</tr>
<tr>
<td>PROJECT</td>
<td>Project info in primary stage</td>
<td>T_ID</td>
</tr>
<tr>
<td>PROJECT_COPY</td>
<td>Project info in primary stage (For backup)</td>
<td>T_ID</td>
</tr>
<tr>
<td>PROJECT_2</td>
<td>Project info for Intermediary stage</td>
<td>T_ID</td>
</tr>
<tr>
<td>PROJECT_2_COPY</td>
<td>Project info for Intermediary stage (For backup)</td>
<td>T_ID</td>
</tr>
<tr>
<td>UNIVERSITY</td>
<td>University info</td>
<td>S_ID</td>
</tr>
<tr>
<td>VOLUNTEER</td>
<td>Volunteer info</td>
<td></td>
</tr>
<tr>
<td>VOTE</td>
<td>Vote info</td>
<td>T_ID</td>
</tr>
</tbody>
</table>

Table 2.5: Database Dictionary
2.5 Relative developing tools

- MyEclipse_5.5.1GA_E3.2.2 is used as the developing IDE. It's combined with many developing functions, such as J2EE code editing, webpage testing, webpage package and webpage publish.

- CuteFTP 5.0 is a FTP software used to transfer local developed website contents to the web server.

- Google Analysis tool is used to generate website traffic report to analyze the visitors' source and interests about the website.

- Google Webmaster Tools provides website administrator with detailed reports about their pages' visibility on Google.

- Dreamweaver is used to develop webpage, combined with “CSS + DIV” techniques.

2.6 Web server hardware.

- IBM x3250 is selected because the stability, quality and service.
- CPU 3040, Memory 2G is ordered to confirm that the web server has stable performance under pressure test and high traffic situation.
• Double 160G hard disks are chosen and RAID technique is used to confirm the data security.

2.7 Environment set up (Detailed instruction is in the Appendix 1)

• JDK installation
• Apache installation
• Apache + Tomcat (Installing the JK Connector)
• Mysql installation
• Installation Checklist
  o Checking Installation

2.8 System Logic

Primary stage ⟷ Intermediary stage ⟷ Final stage

Registe
Login
Fill in Idea Application
Online idea Judge System
Generate Project Homepage
Publish idea competition Result
Contact us (Feedback)
Volunteer Application
Corporate application
Blue for applicant
Green for common visitor
Red for Administrator

Login
Fill in Detail Application
Edit Detail Application
Submit Detail Application
Online detail Judge System
Update Project Homepage
Vote for project
Validate Project Homepage
Publish Project Homepage
Login
Fill in Budget Application
Edit Budget Application
Submit Budget Application
Online Budget Judge
Update Project Homepage
Publish Budget Competition Result
Validate Project Homepage
Publish Project Homepage

Figure 2.4: System logic

3 Module design and Implementation

3.1 Homepage overview
3.2 Services system

3.2.1 Scenario and Goal

Help common visitors contact activity committee, get help though website.

• Feedback page “Feedback.jsp” used to help visitors submit the questions to administrator
• Volunteer page “Volunteer.jsp” used to help volunteers apply online.

3.2.2 System logic workflow

```
Figure 3.2: System logic workflow
```

3.2.3 Code analysis

➢ Feedback content input

The main input screen is page “Feedback.jsp”. Visitor inputs his/her question information into feedback form. HTML “form” tag is used to get data.

*Example jsp code in Feedback.jsp:*

```html
<form id="form1" name="form1" method="post" action = "Feedback_session.jsp" onsubmit="return validate();">
  
  <td width = 160 height=40 valign="center">
    <label>姓名</label>
    
    <input type="submit" name="Submit" value="提交" />
  </td>

</form>
```

➢ Content validation
After visitor clicks the “submit” button, JavaScript function “validate ()” will be called to validate all the input fields in accordance with format required.

Example jsp code in Feedback.jsp:

```javascript
<SCRIPT language="JavaScript">
function validate()
{
    var mail = 'document.getElementById("core_idea").value';
    if (!emailValidator(mail)) {
        document.getElementById("mail_warning ").innerHTML = "*邮件格式错误";
        return false;
    }
}
</SCRIPT>
```

Validate () function calls “emailValidator()” function in JavaScript file “Validation.js”, which implements 8 format validation functions:

- **isEmpty()**: Judge whether the fields is empty for required fields
- **isNumeric()**: Judge whether the fields is Number like mobile fields
- **isChecked()**: Judge whether the fields is blank for required fields
- **isRadioChecked()**: Judge whether the fields is checked in checkbox
- **isAlphabet()**: Judge whether the fields is alphabet
- **isAlphanumeric()**: Judge whether the fields is alphabet or Number
- **emailValidator()**: Judge whether the fields is mach email format
- **lengthRestriction()**: Judge whether the length of fields is in the restriction

Figure 3.3: Contact us webpage
**Example jsp code in Validation.js:**

```javascript
function isEmpty(field) {
    if (field.length == 0) {
        field.focus();
        return true;
    }
    return false;
}

function isChecked(field) {
    if (field == '') {
        field.focus();
        return false;
    }
    else
        return true;
}

function isRadioChecked(field1, field2) {
    if (false == field1.checked && false == field2.checked) {
        field.focus();
        return false;
    }
    else
        return true;
}

function isAlphabet(field) {
    var alphaExp = /^[a-zA-Z]$/;
    if (field.match(alphaExp)) {
        return true;
    }
    else {
        field.focus();
        return false;
    }
}

function lengthRestriction(field, min, max) {
    var uInput = field;
    if (uInput.length >= min && uInput.length <= max) {
        return true;
    }
    else {
        field.focus();
        return false;
    }
}

function emailValidator(field) {
    var emailExp = /^[a-zA-Z0-9]+@[a-zA-Z0-9]+@[a-zA-Z0-9]{2,4}$/;
    if (field.match(emailExp)) {
        return true;
    }
    else {
        field.focus();
    }
}
```
After all the fields are validated successfully, the page “Feedback.jsp” sends the data input and jump to session page “Feedback_session.jsp”.

- **Session operation**

“Feedback_session.jsp” first store the fields value in the corresponding session, which will be used to write back to the input fields when meet exception.

**Example jsp code in Feedback_session.jsp:**

```jsp
session.setAttribute("name", request.getParameter("name"));
session.setAttribute("school", request.getParameter("school"));
session.setAttribute("pho", request.getParameter("pho"));
session.setAttribute("mail", request.getParameter("mail"));
session.setAttribute("addr", request.getParameter("addr"));
session.setAttribute("type", request.getParameter("select"));
session.setAttribute("caption", request.getParameter("caption"));
session.setAttribute("content", request.getParameter("content"));
```

**Validation of “Identifying code”.**

```jsp
String rand = (String)session.getAttribute("rand");
String input = request.getParameter("vali");
if (rand.equals(input)) {
} else {
    out.println("<script>alert('验证码错啦');</script>");
    out.println("<script> self.location = 'feedback.jsp';</script>");
}
```

If the Identifying code is wrong, the page will jump back to “Feedback.jsp” and print a warning of “Identifying code is wrong”.

- **Database operation**

After content validation, the page will jump to database operation page “Feedback_operation.jsp”. This page has two main functions

  o Store feedback information into database
    - Database connection java class “data.LoginData.java”

  o Send feedback information to the administrator

**Example jsp code in “data.LoginData.java”**

```java
package data;
import java.sql.*;

public class LoginData{
    //Define connection
    public Connection conn=null;

    public LoginData(){
```
```java
public Connection getConn()
{
    return this.conn;
}

public boolean connect()
{
    try{
        //Use JDBC to create database connection
        Class.forName("org.gjt.mm.mysql.Driver").newInstance();
        //Use getConnection() function in the DriverManager class to build connection
        //First parameter is the username, second is the password
        this.conn=java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/googlecup?useUnicode=true&characterEncoding=GBK","root","1234");
    }catch(Exception ex){
        ex.printStackTrace();
        return false;
    }
    return true;
}

➢ Store data into database “feedback_operation.jsp”

Example jsp code in “feedback_operation.jsp”

```
response.sendRedirect("feedback.jsp");

Send mail java class in “SendMail.java”

Example jsp code in “SendMail.java”

```java
//Include java mail package
package mail;
import java.security.Security;
import java.util.Date;
import java.util.Properties;
import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

//Class définition
public class SendMail {
private String toMail=null;
private String content=null;
private String title="Your Password";

public String getContent() {
    return content;
}
public void setContent(String content) {
    this.content = content;
}
public String getToMail() {
    return toMail;
}
public void setToMail(String toMail) {
    this.toMail = toMail;
}
public String getTitle() {
    return title;
}
public void setTitle(String title) {
    this.title = title;
}

public SendMail(){
}
//Send mail function
public void sendmail() throws AddressException, MessagingException {
    Security.addProvider(new com.sun.net.ssl.internal.ssl.SSLProvider());
    final String SSL_FACTORY = "javax.net.ssl.SSLSocketFactory";
    // Get a Properties object
    Properties props = System.getProperties();
    props.setProperty("mail.smtp.host", "smtp.gmail.com");
```
```java
props.setProperty("mail.smtp.socketFactory.class", SSL_FACTORY);
props.setProperty("mail.smtp.socketFactory.fallback", "false");
props.setProperty("mail.smtp.port", "465");
props.setProperty("mail.smtp.socketFactory.port", "465");
props.put("mail.smtp.auth", "true");
final String username = "gong1chuang1";
final String password = "xuwenzhi1104";
try{
    Session session = Session.getDefaultInstance(props, new
    Authenticator()
    {
        protected PasswordAuthentication getPasswordAuthentication()
        {
            return new PasswordAuthentication(username, password);
        }
    });
    // -- Create a new message --
    Message msg = new MimeMessage(session);
    // -- Set the FROM and TO fields --
    msg.setFrom(new InternetAddress(username + "@163.com"));
    msg.setRecipients(Message.RecipientType.TO,
                       InternetAddress.parse(this.toMail,false));
    msg.setSubject(title);
    msg.setText(this.content);
    msg.setSentDate(new Date());
    Transport.send(msg);
    System.out.println("Message sent.");
} catch (Exception ex) {
    System.err.println("The reason of sending failed: " +
    ex.getMessage());
    System.err.println("Detailed reason:");
    ex.printStackTrace(System.err);
    
}
```
Email example

Subject: Forget my password

Name: Erik Liu
School: Uppsala University
Mobile: +352 621 402 757
Email: liuhao191@yahoo.com.cn
Address: Råckbagsgatan 34-211 Uppsala Sweden 07516
Category: 报名表格
Title: Forget my password
Content:
Hello,
I forget my project homepage password. Can you help me?
Thanks

Figure 3.4 Example mail

3.3 Application system:

3.3.1 Scenario

The activity plans to select a project with best originality through three stages. In each stage, students log in to the application system on the website, fill in and submit the application form online. The system will validate the application, store data in the database, generate team account and update team homepage. Then the application will be evaluated and published by the committee of this activity.

In the first stage of the selection, students enter the homepage of the website, follow "Application" link to general original webpage, fill in necessary information and submit the data. Afterwards, the system will generate a team account and a project homepage which includes all application data. Meanwhile, the system sends account information (username and password) to the applicant's mail box. Only the applicant can use his/her account to login and see his/her own homepage, but he/she can't edit the homepage. This design can keep each team's originality confidential.
Later on the committee will review all original homepages, and select outstanding ones into next stage. First stage selection result will be published in the website. Applicants can check whether they passed the first stage. Common users can browse each team’s homepage after the publication.

In the second stage, unsuccessful applicants can only read their general original homepage, can’t see detail originality form after they login with their accounts. Only winners can check their homepage and fill the original form in detail. After the form is filled in, they can decide to save or submit the form. If they choose only to save the form, their homepage will be updated. Before the deadline of the second stage, winners can login to their accounts, and edit their homepages before submission. After they submit, their homepage will be freeze and can’t be edited.
The submitted homepage will be validated by experts to check whether improper contents are contained in the homepage. After validation, homepages will be published on the website. Common users can browse them and vote for favorite team. After the second stage is completed, team vote function will be closed. Activity committee starts to review detailed originality, selects outstanding ones and publishes the result. First stage winners can check whether they pass second stage. Common users can read each team’s homepage.

The mechanism of the third stage is almost the same as the second stage. Winners of second stage login to their accounts, fill in budget form, “save”, “edit” and “submit” it. Functions of committee “validate” and “publish” are also the same.

After the third stage, final winners will be selected and published on the website. They will get the fund from Google to make their originality come true.

3.3.2 Workflow
Figure 3.6 Application system workflow
Figure 3.7 Final stage application form
图团队主页（草案）

- **项目名称（草案）**
  大学及企业大学生的管理方案

- **项目信息**
  学校：
  专业：
  学术单位大小：
  预备团队大小：3人
  需要的工具：
  工具使用范围：

- **内容**
  目前大学生团队及大学生的管理方案亟需改进，该方案从大学和企业两个方面进行改进。通过改进，使其更好地服务于大学生的管理需要。

- **项目亮点**
  （草案）

- **详细计划（草案）**
  1. **开始日期**：2008-07-10
  2. **结束日期**：2008-08-20
  3. **项目名称/目的**
  4. **项目计划**

- **评估**
  - **http://blog.sina.com.cn/schooloflibrary2008**
  - **评估**

- **项目预算（草案）**

<table>
<thead>
<tr>
<th>项目</th>
<th>金额</th>
</tr>
</thead>
<tbody>
<tr>
<td>电费</td>
<td>200元</td>
</tr>
<tr>
<td>电话费</td>
<td>100元</td>
</tr>
<tr>
<td>交通费</td>
<td>200元</td>
</tr>
<tr>
<td>工作餐</td>
<td>400元</td>
</tr>
<tr>
<td>纸张</td>
<td>90元</td>
</tr>
<tr>
<td>书籍</td>
<td>120元</td>
</tr>
<tr>
<td>其他杂费</td>
<td>0元</td>
</tr>
</tbody>
</table>

*Figure 3.8 Project homepage*
3.3.3 Function and Code analysis

- **Identifying code**

In order to prevent automatic spiteful system register, we use the mechanism of identifying code. Students need to enter Identifying code in the Input page “application.jsp”. This code will be compared with the rand code in the check page. If they are the same, students can do the following operation, or go back to input page and have the warning “Identifying code is not right”.

**Example jsp code in “application.jsp”**

```html
<input name="vali" type="text" size="4" maxlength="4" />

<img border=0 src="image.jsp"/>

<span id="vA"></span>
```

Identifying code checking in the check page “Data_operation.jsp”,

```java
String rand = (String)session.getAttribute("rand");
String input = (String)request.getParameter("vali");
if (rand.equals(input)) {
    session.setAttribute("vali","right");
    ..................
} else{
    session.setAttribute("vali","wrong");
    response.sendRedirect("application.jsp");
}
```

- **Personal information confidential**

Some input contents are about personal information, such as name and phone number. Google places great emphasis on those sensitive data. In order to keep them confidential, we design an automatic data transit solution. It can separate the common data from sensitive data. For sensitive data, after temporarily stored in the local database, we regularly export them to the excel format and send them to the administrator’s email box in the google database. Then the sensitive data will be deleted in the local database.

- **Generate password**

After database storage operation, the system will generate a project account and a homepage for each team. Only the members of a project with right login name and password can see their homepage. System will automatically generate a password and send it to applicant’s email address.

**Example jsp code in “data_operation.jsp”**

```java
package mail;

import javax.mail.MessagingException;
import javax.mail.internet.AddressException;

public class GenPswd {
    public String generatepassword(){
```
String retStr;  //Password string
String strTable =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
 // Symbol and character used in the password

int len = strTable.length();
retStr = ";
int count = 0;  //number of digital
int count1 = 0;  //number of character
int count2 = 0;  //number of Symbol

try{
    //Get the symbol in the list
    for (int i = 0; i < 8; i++) {
        int intR = (int) Math.floor( Math.random() * len );
        char c = strTable.charAt( intR );
        //Decide the type and number of password: digital, character, Symbol
        if (( '0' <= c ) && ( c <= '9' )) {
            count++;
        } else if (( 'A' <= c ) && ( c <= 'z' )) {
            count1++;
        } else {
            count2++;
        }
        retStr += strTable.charAt( intR );
    }
    return retStr;
} catch (Exception e){
    e.printStackTrace();
    return null;
}

String password=genpswd.generatepassword();
String mdpassword=StringUtil.MD5Encode(password);
//After generates the password, store it in the database and then send password mail
try{
    Connection conn=loginData1.getConn();
    Statement stmt=conn.createStatement();
    ResultSet rs;
    String sqlcount_0="select count(*) from person where i_mail = "+email+"";
    rs=stmt.executeQuery(sqlcount_0);
    rs.next();
    int rowCount_0=rs.getInt(1);
    if(rowCount_0>0){
        String sql="update person set i_pswd = ? where i_mail =?";
        PreparedStatement pstmt=conn.prepareStatement(sql) ;
        pstmt.setString(1,mdpassword);
        pstmt.setString(2,email);
        pstmt.execute();
        pstmt.close();
    }
    String insertsql="insert into mail(email,per_id,pswd)";
For the homepage generation, we designed a homepage template. It includes the general format, but the team specific info is blank. In the generation process, system loads the template first, then selects team’s data and replaces the blank in the template to generate the team homepage. Later system generates file name and path with the project ID, and store the homepage in the right path.

**Example jsp code in “data_operation.jsp”**

```java
try {
    // Get template file
    String filePath = "";
    filePath = request.getRealPath("/") + "generate/sampleindetail.jsp";
    String templateContent = "";
    FileInputStream fileinputstream = new FileInputStream(filePath);

    // New project homepage based on the template
    int length = fileinputstream.available();
    byte[] bytes = new byte[length];
    fileinputstream.read(bytes);
    fileinputstream.close();
    templateContent = new String(bytes);

    // Replace project content in the template
    templateContent = templateContent.replaceAll("####p_id#####", per_id);

    // Generate homepage file name based on calendar
    Calendar calendar = Calendar.getInstance();
    filename = String.valueOf(calendar.getTimeInMillis()) + ".jsp";

    // Generate homepage file path
    filepath0 = "generate/" + filename;
}
```

Generate homepage
3.4 Validation system

3.4.1 Scenario

After students’ submission, their applications need to be validated by the committee before publication. It is used to prevent improper content from being published and prevent the affect to this activity and Google.

Admin login page

Only the committee experts with administration access can login to validation system. In the whole validation process, “Https” secure protocol is used instead of “http”. Meanwhile, admin password is encoded by md5 cryptographic hash function. All these secure mechanisms can keep the validation process from attacking of malice.
Tag filter page

In the application, students need to tag their project information, such as university name, aspect (Education, environment, poverty...), implement area and so on. After experts login to validation system, they enter into the project filter page (commit_action.jsp). In this page, experts select tags to filter project they need to validate. Then the system displays a list of all proper project pages. System also support multiple tag filter, for example, select tag university “Peking University”, aspect “Education” and area “Tibet”. All “Education” projects implement in “Tibet” by “Peking University” students will be displayed in the list. General info (Project name, aspect, area, homepage link) of each project will also be displayed in the list page.
In the filtered list, expert can follow project link, and enter homepage. Afterwards they can check and edit the content. If any improper content such as advertisement or business content against goodwill found, expert will delete them. After the entire content is validated, expert clicks “publish” button. Then this project will be published and can be browsed by all common users.

In the database, each projects’ info stored in the “project” table. “Project” table has a “Status” field which stores project status. Status info is used to store “Saved”, “submitted”, “Validated” info and judge whether the project passes each stage.

3.4.2 Workflow

3.4.3 Code Analysis

- **Https configuration**

  Https secure protocol is used in the whole validation process instead of http.

  - Use JDK tool “keytool” to generate security certificate keystore.
    - In command line enter:
      ```
      JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA -keystore C:\tomcat.key
      ```
    - Input necessary configure value, password use tomcat default value: “changeit”
    - Then security certificate in PCKS1.2 format will be stored “C:\tomcat.key”
  - If JDK version 1.3 need to download Java Secure Socket Extensions (JSSE) package
    - Copy jcert.jar, jnet.jar, jsse.jar to path $JAVA_HOME/jre/lib/ext
Uncomment the Connector element for the SSL port of file \{tomcat\}/conf/server.xml. Now there are two connectors: one is port 8080, and the other is 8443 for SSL.

```xml
<Connector port="8443" maxHttpHeaderSize="8192"
           maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
           enableLookups="false" disableUploadTimeout="true"
           acceptCount="100" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS"
           keyStoreFile="yourKeystoreFileName"
           keyStoreType="JKS"
           keyStorePass="yourKeystorePassword"/>
```

Modify the file \{tomcat\}/webapps/ROOT/WEB-INF/web.xml. When the server opens the page “login.jsp”, it will automatically use Https instead of http.

```xml
<web-app>
  ...
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>login.jsp</web-resource-name>
      <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
      <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
  </security-constraint>
  ...
</web-app>
```


Dynamic multiply tag filter:

With the help of multifunction sql sentence, system supports multiple tag filters. It helps experts easily find specific project homepage to validate.

Example jsp code in “List_grade_operation.jsp”

```java
//Get tag value from form
String pro_class = request.getParameter("class"); //Category
String pro_area = request.getParameter("area"); //area
String search = request.getParameter("search");
String sub = request.getParameter("sub"); //university
String main = request.getParameter("main");
String main_area = request.getParameter("main_area");
String status = request.getParameter("status");
int int_status = Integer.parseInt(status);
int int_main_area = Integer.parseInt(main_area);
......
```

//Filter project depends on whether have been validated
```
if(int_status > -1)
{
    if(int_status == 3)
Display items in separate pages:

Committee experts use tag filtered projects. Committee experts use tag filtered projects. Several pages will be needed to display a large number of projects.

Example jsp code in "List_grade_operation.jsp"

```java
int intPageSize; //Number of items display in one page
int intRowCount = 0; //Number of total items
int intPageCount = 0; //Number of total pages
int intPage = 0; // Number of total pages will display
intPageSize = 30; //Set number of items display in one page
String strPage;
strPage = request.getParameter("page"); // Set number of total pages will display
if (strPage==null) { //If strpage value is null from QueryString
    intPage = 1; //Set Number of total pages to 1
} else { //Change format form string to int
    intPage = java.lang.Integer.parseInt(strPage);
    if (intPage<1) {
        intPage = 1;
    }
}
```

//Display page number link in the bottom of the webpage
3.5  Publishing system

3.5.1  Scenario

After applications are validated and evaluated by the committee experts, the project homepage will be published on the website. Applicants can check whether they pass the competition of each stage. Meanwhile, common users can use tags to filter projects like the validation system. The filtered projects are displayed on several pages with general information and voting information. Students can also sort the project list by vote ascending or descending. Afterwards, common users can follow each project’s link, enter their homepage, and vote for the team. The count of votes and popularity will be considered by the committee in the evaluation process.
3.5.2 Workflow

![Diagram showing the workflow of project list and vote count.]

3.5.3 Code Analysis

Vote function

In order to prevent automatic spiteful voting, publishing system uses cookie to control common user voting process. When a user vote for one project, a cookie will be written to user's PC. The cookie has two values: project number and validity for 1000*60*60*10 ms (10 minutes). When the same user votes again, the system will check the cookie in user’s browser. If the cookie with the same project number already exists and hasn’t out of validity, System will give the user a notice “Vote for the same project needs to wait for more then 10 minutes”. If vote for another project or vote for the same project but after cookie out of validity, system will write a new cookie to users PC.
This mechanism can prevent malice software from automatically voting for one project in a short time and keep fair competition of the activity.

**Exemple jsp code in “vote.jsp”**
```jsp
<%@ page contentType="text/html; charset=gb2312" language="java" errorPage="" %>
<%@ page import="java.util.*"%>
<%@ page import="java.lang.*"%>
<%@ page import="java.io.*"%>
<jsp:useBean id="vote" scope="request" class="com.gong1chuang1.vote.vote"/>

String vote1=request.getParameter("lang");
vote.n=4;
vote.filePath="vote.txt";
vote.readFile();
if(vote1.compareTo("0")==0)
    vote.voteNum[0]++;
if(vote1.compareTo("1")==0)
    vote.voteNum[1]++;
if(vote1.compareTo("2")==0)
    vote.voteNum[2]++;
if(vote1.compareTo("3")==0)
    vote.voteNum[3]++;
vote.writeFile();

<script language="javascript">alert("Thanks for your voting");
    self.location="index.jsp";</script>
```

In order to enhance the efficiency of the voting process, system tries to decrease the number of database operation. System doesn’t do it for each vote, but in batch processing. When a user votes for one project, the vote content will be temporarily stored in memory. Every ten minutes system will atomically clear the temporary file and transfer data to database. This will decrease the number of database operation in the voting peak time, save the server resource and enhance the efficiency.

**Exemple java code in “vote.java”**
```java
package com.gong1chuang1.vote;
import java.io.*;
import java.util.*;
public class vote extends Object {
    public String filePath="";
    public int n;
    private File voteFile;
    private BufferedReader fileRead;
    private PrintWriter fileWrite;
    public String systemMessage="";
    private String voteStr[] = new String[10];
    public int voteNum[] = new int[10];
    public void createFile() //Create vote file
        throws FileNotFoundException
```
Everyday a large number of users browse publishing webpage and use the tag filter function. If the filter function implemented in dynamic jsp technique, every users' filter needs call data operation. It will cost a lot of server and web connection resource. The performance will also be bad. So we use static html page instead of dynamic jsp page to display the project tag filter list. We wrote a java program to generate static tag filter project list, in advance, based on each tag and tag combine. For example, we generate list for all "education" category project, and save it to html file 
"../area_0_class_1_sch_00.html". This URL presents the tag filter information:
“area_0” means without “area” tag
“class_1” means with “category” tag, value “1” means “Education”
“Sch_00” means without “University” tag.
More complete url like “area_25_class_2_sch_00.jsp” means tag combine.
So when user clicks the tag in the publishing page, he/she just links to the right static tag
list html page, and doesn’t need dynamic data operation.

Exemple java code in “pub_generate.java”. To generate project list based on area

```java
public void generate_area(){
    String tedu=null;//Education
    String tpoor=null;//Poverty
    String content="
    for (int aid_i=1;aid_i<35;aid_i++)
    {
        aid=aid_i;
        try{
            LoginData logindata=new LoginData();
            Connection conn=logindata.getConn();
            Statement stmt=conn.createStatement();
            ResultSet rs=null;
            //Get project content from database
            String countsql="select count(*) from project_copy,mark
            where project_copy.t_id=mark.t_id and mark.stage_2_status='1' and
            a_id="+aid;
            rs=stmt.executeQuery(countsql);
            rs.next();
            count=rs.getInt(1);
            System.out.println(count);
            if(count>0){
                if(pagesize==0)pagesize=20;
                page = count/pagesize;
                r = count % pagesize;
                if(r!=0)page=page+1;
            }else {
                String fileame=null;
                try{
                    String filePath = "C://sever//Tomcat
                    5.5//webapps//googlecup//project//templete_secondary.jsp";
                    String templateContents="
                    FileInputstream fileinputstream = new
                    FileInputStream(filePath);//Get content from template page
                    int lenght = fileinputstream.available();
                    byte bytes[] = new byte[lenght];
                    fileinputstream.read(bytes);
                    fileinputstream.close();
                    templateContents = new String(bytes);
                    templateContents=templateContents.replaceAll("####content####","\"\"\"\"");
                    templateContents=templateContents.replaceAll("####sappage####","\"\"\"\"");
                    templateContents=templateContents.replaceAll("####pageto####","1\"");
                    templateContents=templateContents.replaceAll("####pagecount####","1\"");
                    seppage="";
                    content="";
                    filename="pub_2_area_"+aid+"_class_"+0+"_sch_"+0+0+".jsp";
```
In each competition stage, committee experts evaluate each application to decide whether the application can pass and enter next stage. Like validation system, admin user control, MD5 and https score mechanism are used in evaluation system. The evaluation process includes 3 rounds: first review, second review by two different common experts and final check by senior expert. Experts login to admin account, filter the tag and get project list. The list displays general information of the project and status of three rounds evaluation.
For the first and second review, common experts follow the each round review status link, enter project homepage and decide whether an application passed or not. Before evaluation, status shows “Wait for Review”. After evaluation, status shows expert’s name. It will prevent one expert from reviewing the same project twice. For each project, after the second review, system compares the result between first and second review.

Both “Pass” -> Final check automatically “Pass”
Both “Not Pass” -> Final check automatically “Not Pass”
A “Pass”, a “Not Pass” -> Final check automatically “Dispute”, need senior expert reviews again and makes decision.

For “Dispute” project final check, only senior experts have access to enter and make final decision directly.

3.6.2 Workflow

3.6.3 Code Analysis

//Judge authentication
String admin_online = (String)session.getAttribute("admin_online");
if (admin_online == null)
{
    out.println("<script>alert('error');</script>");
    response.sendRedirect("login1.jsp");
}

String comment_stage = "";
String stage = (String)session.getAttribute("stage");
out.println("stage :");
out.println(stage);
out.println(<br/>);
String p_id = (String)session.getAttribute("mark_id");
String grade_stage = "";
String comment = request.getParameter("comment");
    String pass_3_1 = "0";
    String pass_3_2 = "0";
    pass_3_1 = (String)request.getParameter("pass_3_1");
    pass_3_2 = (String)request.getParameter("pass_3_2");
if (pass_3_1 == null)
    pass_3_1 = "0";
if (pass_3_2 == null)
    pass_3_2 = "0";
out.println("comment :");
out.println(comment);
out.println(<br/>);
//Judge evaluation stage
if (stage.equals("3"))
{
    if (admin_online.equals("liuhao") || admin_online.equals("yuke"))
    {
        //response.sendRedirect("unsuccess.jsp");
    }
    else
    {
        response.sendRedirect("unsuccess.jsp");
        comment_stage = "2";
    }
else if (stage.equals("2"))
{
    comment_stage = "3_2";
}
else if (stage.equals("1"))
{
    comment_stage = "3_1";
}

String mark_tsql = "update mark set stage_3_" + stage + "_id = '" + admin_online;
mark_tsql = mark_tsql + ",stage_3_" + stage + "_status = '" + pass_3_1

String select_tsql = ";
String stage_2_1_status = ""
String stage_2_2_status= ""
String stage_2_status=" ";

try{
    Connection conn=loginData.getConn();
    if(!stage.equals("3")
    {
        out.println("mark_tsql");
        out.println(mark_tsql);
        out.println("<br/>");
        Statement smt_1=conn.createStatement();
        smt_1.executeUpdate(mark_tsql);
        smt_1.close();

        //Compare the result of two rounds
        Statement stmt_select=conn.createStatement();
        ResultSet rs;
        select_tsql = "select * from mark where p_id = '" + p_id + '"" ;
        out.println("select_tsql : ");
        out.println(select_tsql);
        out.println("<br/>");
        rs=stmt_select.executeQuery(select_tsql);
        rs.next();
        stage_2_1_status = rs.getString("stage_3_1_status");
        stage_2_2_status = rs.getString("stage_3_2_status");
        out.println(stage_2_1_status);
        out.println(stage_2_2_status);
        select_tsql = ";
        out.println(select_tsql);
        out.println("<br/>");

        if((stage_2_1_status != null) && (stage_2_2_status != null))
        {
            if((stage_2_1_status.equals("1")) && (stage_2_2_status.equals("1"))
            {
                out.println("pass");
                stage_2_status= "stage_3_status = '1';"
            }
            else if((stage_2_1_status.equals("0") &&
                    stage_2_2_status.equals("0"))
                    {
                out.println("not pass");
                stage_2_status= "stage_3_status = '0';"
            }
            else if((stage_2_1_status.equals("1") &&
                    stage_2_2_status.equals("0")) || (stage_2_1_status.equals("0") &&
                    stage_2_2_status.equals("1")) || (stage_2_1_status.equals("0") &&
                    stage_2_2_status.equals("0"))"}
stage_2_2_status.equals("1")
{
    out.println("not know");
    stage_2_status = "stage_3_status = '2'";
}
stmt_select.close();
out.println("stage_2_status");
out.println(stage_2_status);

mark_tsql = "update mark set " + stage_2_status + " where p_id = '" + p_id + "'";

out.println(mark_tsql);
Statement smt_update=conn.createStatement();
smt_update.executeUpdate(mark_tsql);
smt_update.close();
} else
    out.println("Need round two evaluation");

else if(stage.equals("3"))
{
    out.println("comment :");
    out.println(comment);
    out.println("<br/>");

    if(admin_online.equals("liuhao") | admin_online.equals("yuke"))
    {
        String sql_1_2 = "update mark set stage_3_status = '" + pass_3_2 + "', comment_3='" + comment + "' where p_id = '" + p_id + "'";
        out.println(sql_1_2);
        Statement smt_1_2=conn.createStatement();
smt_1_2.executeUpdate(sql_1_2);
smt_1_2.close();
    }
}
response.sendRedirect("grade_success.jsp");
conn.close();

4 Conclusion

In this thesis work, besides we developed and maintain the website common function for 8 month. We also implement some highlight features.

Firstly, in order to ensure the data storage safety and confidentially, we use many secure techniques, like https and remote database. We configure the https property of the web server and some WebPages. With it help, all webpage about personal sensitive data or confidential info will be encrypted before transmission and decrypting a message upon arrival. All sensitive personal data will only be temporarily store in local database, and then send to Google security remote database through automatically mail system.
Secondly, we tried to find the best solution to enhance the system efficiency. For example we decreased the expensive database operation to save system resource and reaction time. Instead of it, for large voting storage, we temporally store data in memory and file. Every 10 minutes, transfer these data to database use batch processing.

Thirdly, user friendly theory is valued in the system design. Different web browser and different version, like IE6, IE7, Firefox, has different support to javascript and css +div display. In the developing process, we met a lot of browser compatibility problem. Like the “alert()” don’t work well in fire fox. So we tested many functions to ensure it can work fine in every browser environment and supply outstanding end-user experience.

Fourthly, some website management tools are used, like google analysis and Google webmaster. Webmaster Tools helps us see how our site is performing in google search results, troubleshoot potential problems, and build Google-friendly sites. From Google Analytics support, we learned more about where our visitors come from and how they interact with our site. We’ll get the information we need to make better design, strengthen our content initiatives, and create higher-converting websites.

5 Discussion
5.1 Limitation

The website performance and efficiency is not only related to the web program, but also depending on the web server, database and user’s browser support.

Tomcat server

Because the website users flow at the peak time is vary large volume. The web server need run a lot of jsp and servlet pages, also with the database operation. At the same time operation system need to run so many threads. At this time we found the tomcat server will have “OutOfMemoryError” errors and shut down automatically. In order to ensure user browser website normally, at the peak time, administrator needs to check and restart the server manually. We found tomcat don’t have a good support to the large memory management.

In default, the Java Virtual Machine and Tomcat (WebServices) only use a small amount of memory. We can adjust the RAM they will use with the -Xmx option as in -Xmx1000M. (Meaning use up to 1000MB if available.)

Assign more RAM to the Java command line examples

1. Open, for example, ResizeImage.bat or ResizeImage.sh at:
   \{ install location\}\Samples\java\commandline\ResizeImage.bat
2. Add -Xmx1000M to the line:
   set COMMAND=java -cp "%THECLASSPATH%"
   com.adobe.altercast.commandline.ResizeImage %*
   resulting in:
   set COMMAND=java -Xmx1000M -cp "%THECLASSPATH%"
   com.adobe.altercast.commandline.ResizeImage %*

Assign more RAM to Tomcat (WebServices) on Windows

1. Open wrapper.properties at:
   \{your install location\}\server\conf\wrapper.properties
2. Scroll down to the bottom and you will find:
   # This is the command line that is used to start Tomcat. You can *add* extra
   # parameters to it but you can not remove anything.
# wrapper.cmd_line="$(wrapper.javabin)" -Xrs -Xmx512M -classpath "$(wrapper.class_path)" $(wrapper.startup_class) -config "$(wrapper.server_xml)" -home "$(AlterCastTomcatHome)"

3. Modify the value of: -Xmx512M

Browser Compatibility
Web browser can translate HTML code documents into a formatted Web page. The basic rules for translating are established by the World Wide Web consortium, which publishes the official HTML standards. But there's considerable room for interpretation within those ground rules. So there are still some format and functional translate difference between different web browser. In addition, the HTML standards usually run ahead of what the browsers support. No browser as yet supports 100% of the HTML Version 5 standard, but some browsers come closer than others. This problem is further complicated by browser-specific “HTML extensions.”

After investigation, we found most of our website user use Firefox, IE6, IE7 three versions' browse. But in the developing process, some css format displayed differently between them. And Javascript function doesn’t work well in Firefox.

At this time, we found some fresh and simple tools for cross-browser compatibility testing, tools that actually make this stuff pretty easy

#1 — Xenocode Browser Sandbox

The Xenocode Browser Sandbox is a game-changer for browser testing on Windows-based machines. With a single click of your mouse you can have an open and working browser without any installation. You can test in various IE versions, Firefox, Google Chrome, and even Safari. And really test, too, not just screenshots. To top everything off, the entire service is provide free of charge. Zip, nada, nothing.

Besides testing, we tried different css syntax to improve the compatibility between different browsers. Like use “!important” to increase the css priority

```java
E1{
    background-color: red !important; /*提升优先权*/
    background-color: blue;
}
```
At the meantime, we use javascript or jsp to test different implementation for the same function. In this way, we can find the solution has the best support to different browser. Like use AJAX and "innerHTML" to make user notice instead of "Alert()" function.

5.2 Future work

With the work has been done so far, we have tried to optimize the system architecture and performance. The result shows there is room for further improvement and becomes an indicator for future work.

Use Struts Build MVC Model

Because the time limitation, we develop the all the WebPages use JSP/Servlet technique.

The implement of presentation, business and controller mixed together. Then the function efficiency is affected without using JavaBean. It is not comply with the modeling development theory, make maintenance difficult. So we need to work on reorganize the architecture of the webpage, separate presentation, logic and functions. Use Struts build MVC Model is the best solution.

Struts can cleanly separate the model (application logic that interacts with a database) from the view (HTML pages presented to the client) and the controller (instance that passes information between view and model). Struts provides the controller (a servlet known as ActionServlet) and facillitates the writing of templates for the view or presentation layer (typically in JSP, but XML/XSLT and Velocity are also supported). We are only responsible for writing the model code, and for creating a central configuration file struts-config.xml which binds together model, view and controller.

Requests from the client are sent to the controller in the form of "Actions" defined in the configuration file; if the controller receives such a request it calls the corresponding Action class which interacts with the application-specific model code. The model code returns an "ActionForward", a string telling the controller which output page to send to the client. Information is passed between model and view in the form of special JavaBeans. A powerful custom tag library allows it to read and write the content of these beans from the presentation layer without the need for any embedded Java code.

Struts also supports internationalization, provides facilities for the validation of data submitted by web forms, and includes a template mechanism called "Tiles" which (for instance) allows the presentation layer to be composed from independent header, footer, and content components.
In the developing process, we have developed a large number of testing Webpage and updated page. Only depending on manually naming standard to manage the mess of these file is very hard. More difficult thing is to find the file of right version. It affects the efficiency of the development. For future development, we plan to use Subversion to help Version control.

A version control system maintains an organized set of all the versions of files that are made over time. Version control systems allow people to go back to previous revisions of individual files, and to compare any two revisions to view the changes between them. In this way, version control keeps a historically accurate and retrievable log of a file's revisions. Key terms in version control:

Checking in a file or directory
This copies your working directory back into the repository as a new version.

Checking out files or directories
This copies the latest revision of a file from the repository to your workspace. When you check out a directory, you check out all files and subdirectories under it.

Committing a file or directory
This is the same as checking in a file or directory. Often version control users will say that they have "committed a change"; this means that they made changes to their working copies of files and committed these back to the repository.

Conflict
When two developers make changes to their working copies of the same file and commit them to the repository, their work may conflict. When this happens, CVS will detect the conflict and require someone to resolve it before committing their changes.

Merging
Combining multiple changes made to different working copies of the same files in the source repository. Merging is a strategy for managing conflicts by letting multiple developers work at the same time (with no locks on files), and then incorporating their work into one combined version. Merging works well when two sets of changes are made to different lines in a file and can be easily combined. When changes to a file are made on the same line or lines, conflicts occur, requiring someone to edit the file manually before the changes can be committed to the source repository successfully.

Repository
A shared database with the complete revision history of all files under version control.
Resolving
Conflicts within a file created by two developers attempting to commit conflicting changes must be addressed by manually editing the file. Someone must go through the file line by line to accept one set of changes and delete the other set. Files with conflicts cannot be committed into the source repository successfully until they are resolved.

Revision
A numbered draft of specific updates to individual files. Each time you edit a file and commit it back to the repository, the file's revision number increases.

Version
The numbering scheme used to identify sets of files that are tagged and named at a certain point in time.

Workspace
Your copies of the files you want to edit on your local hard disk or Unix user account. When you edit files in your workspace, they will become out of sync with the repository. That's progress! Then you need to get your changes back into the repository so that everyone else can see them.

Subversion's architecture
6 References

Appendices

Appendix 1 Installation of Apache + JK + Tomcat on Windows XP
Installing the JDK

1. Go to http://java.sun.com/j2se/1.4.1/download.html and get the JDK version of J2SE 1.4.1. The JRE version works for Tomcat in some circumstances, but unless you know exactly what you are doing, the full JDK is the better choice. Its a big download (38Mb), so plan accordingly.

2. Run the installer. Take the defaults. This will install the JDK in c:\j2sdk1.4.1_01.

3. Setup the JAVA_HOME environment variable. Click Start->Control Panel->System. Then click Advanced, then click "Environment Variables".

4. Under System variables, click "New". For Variable Name, use JAVA_HOME (exactly like that, case and all). For Variable Value, use c:\j2sdk1.4.1_01. Click OK until the Control Panel is closed.

5. Verify the setting. Click Start->Run, and in the command box type "cmd" and click OK. At the command prompt, type echo %JAVA_HOME% and verify that the value returned matches the Variable Value from step 4.

Installing Apache

1. First, make sure there is no web server running on port 80 on your machine. It's possible that you have IIS installed and running, and you don't even know it. Go to Start->Administrative Tools->Services, and look for a service named "World Wide Web Publishing". If that service is listed, click on it's name, then click the "Stop" button in the toolbar. The stop button is a square. Then, choose Properties in the toolbar, and set "Startup Type" to "Manual" instead of "Automatic". This will prevent IIS from starting on your machine when it is rebooted. Only one service can run on port 80 at a time, and in this HOWTO that service will be Apache, not IIS.

2. Download the Apache binary for Windows. Choose a mirror site, and get the MSI installer. This will make things easier. The current version of Apache is 2.0.44. You can get the MSI installer from a mirror: http://httpd.apache.org/download.cgi

3. Save the MSI installer to a location on your hard drive. Wherever you put it, remember that location.

4. Click Start->Run->Browse, and browse to the location where you put the MSI installer. Run the MSI installer. You will see a couple of screens dealing with licenses, etc. Click through those, then you will see a screen that says "Server Information" and has some values in the fields already. Delete the Network Domain and Server Name values, and type "localhost" into both fields. Put your email address in the field entitled "Administrator's Email Address". Choose the option to install Apache for all users on port 80. Click Next.

5. On the next screen, the default is "Typical". Select "Custom" and click Next.

6. The next screen says "Custom Setup". Click the "Change..." button to change the installation directory.

7. The next screen says "Change Current Destination Folder". Click UP to the C drive, that is, C:\. Click the "new folder" icon (the folder with a "***" on it) and call the new folder "apache". Click "OK", then click Next. The goal here is to install Apache 2.0.43 in a folder called C:\apache, NOT in the default installation folder that the installer wants to use. This is because the default installation is a Windows-friendly location with spaces in the pathname. Spaces in pathnames can cause
problems with software developed for Linux, which is where most open source software is developed.

8. Click "Install". The installer will do its thing, and you will see various windows popping up and disappearing. This is normal. When the installer is finished, you should see a "successful" message and there should be a new icon in your system tray.

9. The installer starts Apache by default, so once the installer completes, you can open up a browser window and browse to http://localhost. You should see the default Apache home page. If you don't see this, go back to Step 3 and try again. The MSI installer is about as painless as you can get. This will leave you with c:\apache\Apache2 as your Apache home directory.

10. Once you have verified that Apache is running on port 80, right click on the Apache System Tray icon and choose "Open Apache Monitor". Then click the "Stop" button to stop Apache. Apache will be started again after Tomcat and JK are installed.

Installing Tomcat

1. Create a folder on your hard drive. Call it tomcat, and put it in the C drive root, so that you end up with C:\tomcat.


3. Execute the EXE file you downloaded in Step 2. This will start the installer. Read the license, and if you agree, click "I Agree" and go to the next screen.

4. On the next screen, select the box that says "NT Service". If you want to install the Tomcat source code (entirely optional, you don't need it to run Tomcat) then select that option as well. Leave the default selections as they are. Click Next.

5. On the next screen, DON'T take the installation directory default. Click the Browse button, and choose the folder you created in Step 1. This will leave you with something like C:\tomcat\Tomcat 4.1. This isn't ideal...you don't want spaces in the name. Change the last part of the path to something like "tomcat-4-1-18" so that you end up with c:\tomcat\tomcat-4-1-18 in the directory field.

6. Click Next. The installer will start copying files. Eventually you will be presented with a screen asking for a password for the admin user. Type in a password, making sure that you can remember it later. Leave the port number at 8080. If you change it, you will have to manually edit Tomcat's config files. Click Next.

7. When the installer is finished, you will have a directory structure for Tomcat at c:\tomcat\tomcat-4-1-18, and the service should be running. You can verify the service is running by accessing http://localhost:8080/. Note that since Tomcat is installed as a service, you can manage it with the Services tool by clicking Start->Administrative Tools->Services and looking for service called Apache Tomcat 4.1.


Installing the JK Connector
The JK connector is what Apache and Tomcat use to "speak" with each other. It's a separate piece of software, distinct from both Apache and Tomcat. There is also a JK2 connector, but in this HOWTO we will be working with the JK connector. The goal here is to get JSP and servlets to run on port 80, without having to set Tomcat to run on port 80. It is possible to run Tomcat on port 80 as a stand-alone web server, but in many situations, there is a need to use Apache on port 80. The connector acts as the conduit between Apache and Tomcat in that scenario.

1. Download the JK connector here: [http://jakarta.apache.org/builds/jakarta-tomcat-connectors/jk/release/v1.2.1/bin/win32/](http://jakarta.apache.org/builds/jakarta-tomcat-connectors/jk/release/v1.2.1/bin/win32/). In our case, we want the file called mod_jk-2.0.43.dll because we are using Apache 2.0.43.

2. The file is an Apache module, it is NOT a Tomcat JAR file or WAR file. It doesn't belong in Tomcat's directory structure, it belongs in a place where Apache can find it. For Apache 2.0.43, that is APACHE_HOME\modules, where APACHE_HOME is equal to your Apache install location. In our case, that is c:\apache\Apache2, so we want to put the JK DLL file into c:\apache\Apache2\modules. You should end up with a file called c:\apache\Apache2\modules\mod_jk-2.0.43.dll on your hard drive. If you notice that the other files in c:\apache\Apache2\modules end in ".so", that's OK. Both ".so" and "*.dll" are valid Apache module extensions on Windows systems.

NOTE: from now on, APACHE_HOME = c:\apache\Apache2 or the name of your Apache home directory.

3. Edit Apache's configuration file. It is located in APACHE_HOME\conf and is called httpd.conf. Don't be nervous, there is a copy in the same directory called httpd.default.conf. You can always revert back to the default settings if you need to by copying that file.

4. Edit httpd.conf in your favorite text editor. DO NOT edit it in Microsoft Word!! Use Notepad or another editor. Scroll down to where you see a bunch of lines that say "LoadModule". At the end of this list, add a line that says:

```
LoadModule jk_module modules/mod_jk-2.0.43.dll
```

Save the file you just edited to the APACHE_HOME\conf directory, and before going any further, verify that you have things setup correctly. Open a command prompt window by clicking Start->Run and typing `cmd` and clicking OK. At the prompt, type `c:\apache\Apache2\bin\Apache.exe -t` and hit return. You should see a message that says "Syntax OK". If you don't see this message, go back to Step 1 and review any changes you have made to determine the error. Note that using "-t" on the command line WILL NOT start the Apache service, it just checks the changes that you made to httpd.conf for any errors.

5. Edit Tomcat's configuration. First, make a copy of c:\tomcat\tomcat-4-1-18\conf\server.xml. Now edit c:\tomcat\tomcat-4-1-18\conf\server.xml. Look for a line that says

```
<Server port="8005" shutdown="SHUTDOWN" debug="0">
```

Just below that line, add the following:

```
<Listener className="org.apache.ajp.tomcat4.config.ApacheConfig"
    modJk="c:/apache/Apache2/modules/mod_jk-2.0.43.dll" />
```

Now look for a line that says
<Host name="localhost" debug="0" appBase="webapps" unpackWARs="true" autoDeploy="true">

Just below that line, add the following:

<Listener className="org.apache.ajp.tomcat4.config.ApacheConfig" append="true" forwardAll="false" modJk="c:/apache/Apache2/modules/mod_jk-2.0.43.dll" />

(note that this line, while similar to the first one, is different)

Save the changes you made to server.xml. Restart the Tomcat service. Wait a few seconds, and then check to see if there is a file called mod_jk.conf in c:\tomcat\tomcat-4-1-18\conf\auto (c:\tomcat\tomcat-4-1-18\conf\auto\mod_jk.conf). If there is, all is well. If there isn't, go back to the beginning of Step 5 and verify the changes you made to server.xml.

By adding the two Listener elements to server.xml, you are causing Tomcat to generate the necessary Apache configuration directives for mod_jk automatically. You don't need to type them by hand. This is a very convenient feature, but it is optional. If you want to configure Apache by hand, you are welcome to do so, but that is not covered here in this HOWTO.

6. Almost done on the Tomcat side. JK, in the Apache module, uses the concept of a "worker" to send and receive info to Tomcat. We need to tell the worker where Tomcat is, and what port we want it to use. This is done with a workers.properties file. We want to put the workers.properties file into c:\tomcat\Tomcat-4-1-18\conf\jk, though it can be anywhere. So, open up Notepad or your favorite text editor with a blank file, and add the following lines to it:

```
# BEGIN workers.properties
# Definition for Ajp13 worker
worker.list=ajp13
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
# END workers.properties
```

Save the file as c:\tomcat\Tomcat-4.1.18\conf\jk\workers.properties. Note that the default name for a JK worker is "ajp13". You can call it what you like, but my advice is don't change it until you have a completely working installation and can see how all the pieces fit together.

7. Now go back to Apache's httpd.conf file, and edit it (c:\apache\Apache2\conf\httpd.conf). At the very end of the file, add the following line:

```
Include c:/tomcat/tomcat-4-1-18/conf/auto/mod_jk.conf
```

Save the file, and verify your syntax by running c:\apache\apache2\bin\apache.exe -t at a command prompt. You should see a message that says "Syntax OK". If you don't see this message, review the edits you just made and make sure that c:/tomcat/tomcat-4-1-18/conf/auto/mod_jk.conf and workers.properties actually exist where they should.

**Installation Checklist**

OK, that was a lot of work. By now, you should have the following environment:

- JDK installed in c:\j2sdk1.4.1_01
• JAVA_HOME environment variable set for all users (System level) to c:\j2sdk1.4.1_01
• The JK DLL downloaded from the Jakarta site and placed in c:\apache\Apache2\modules
• Apache installed in c:\apache\apache2, with "Syntax OK" messages after both the
  LoadModule edit and the Jk/Include edit, and able to view the Apache welcome page at
  http://localhost/
• Tomcat installed in c:\tomcat\tomcat-4-1-18, successfully restarted with both Listener
  element edits to server.xml, and able to view the Tomcat welcome page and execute the
  Tomcat examples at http://localhost:8080/
• A file called workers.properties located in c:\tomcat\Tomcat-4-1-18\conf\jk

Checking Your Installation

1. Start Tomcat as a service using the service control panel. Start->Application Tools->Services.
   If Tomcat is running, stop it and then start it. Wait a few seconds before continuing.

2. Start Apache using the Apache Monitor in your System Tray. Right click on the System Tray
   icon, and choose "Open Apache Monitor", then click "Start". Wait a few seconds before
   continuing.

3. Verify the Apache welcome page at http://localhost/


5. Verify Tomcat examples available on port 80 at http://localhost/examples/jsp/index.html

Appendix 2 Hypertext Transfer Protocol over Secure Socket Layer

Hypertext Transfer Protocol Secure (HTTPS) is a combination of the Hypertext Transfer
Protocol and a network security protocol.

Both HTTP and the security protocol operate at the highest layer of the TCP/IP Internet
reference model, the Application layer; but the security protocol operates at lower sublayer,
encrypting an HTTP message prior to transmission and decrypting a message upon arrival.

HTTPS has also been known as "Hypertext Transfer Protocol over Secure Socket Layer", but
now HTTPS may be secured by the Transport Layer Security (TLS) instead of Secure Sockets
Layer (SSL) protocol.

To invoke HTTPS, one replaces "http://" with "https://" in the URI, or Web address. HTTPS
connections are often used for payment transactions on the Web and for sensitive transactions in
corporate information systems.

Strictly speaking, HTTPS is a separate protocol, but refers to the combination of a normal HTTP
interaction over an encrypted Secure Sockets Layer (SSL) or Transport Layer Security (TLS)
connection. This ensures reasonable protection from eavesdroppers and man-in-the-middle
attacks, provided that adequate cipher suites are used and that the server certificate is verified
and trusted. An https: URL may specify a TCP port. If it does not, the connection uses port 443
(unsecured HTTP typically uses port 80).

To prepare a Web server to accept HTTPS connections, the administrator must create a public-
key certificate for the Web server. These certificates can be created for Unix-based servers with
tools such as OpenSSL’s ssl-ca [1] or SuSE’s gensslcert. This certificate must be signed by a
certificate authority of one kind or another. The authority certifies that the certificate holder is
indeed the entity it claims to be. Web browsers are generally distributed with the signing
certificates of major certificate authorities so that they can verify certificates signed by them.
Appendix 3 MD5

In cryptography, MD5 (Message-Digest algorithm 5) is a widely used cryptographic hash function with a 128-bit hash value. As an Internet standard (RFC 1321), MD5 has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files. However, it has been shown that MD5 is not collision resistant[1] hence MD5 is not suitable for applications that rely on this property. An MD5 hash is typically expressed as a 32 digit hexadecimal number.

MD5 was designed by Ron Rivest in 1991 to replace an earlier hash function, MD4. In 1996, a flaw was found with the design of MD5. While it was not a clearly fatal weakness, cryptographers began recommending the use of other algorithms, such as SHA-1 (which has since been found vulnerable itself). In 2004, more serious flaws were discovered making further use of the algorithm for security purposes questionable.[2][3] In 2007 a group of researchers including Arjen Lenstra described how to create a pair of files that share the same MD5 checksum.[4] In an attack on MD5 published in December 2008, a group of researchers used this technique to fake SSL certificate validity.[5]