Blog.java

package Community;

import java.awt.Button;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Image;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.awt.image.BufferedImage;

import java.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.imageio.ImageIO;

import javax.swing.JButton;

import javax.swing.JComboBox;

import javax.swing.JFileChooser;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTextPane;

import javax.swing.ScrollPaneConstants;

import javax.swing.border.EmptyBorder;

import javax.swing.text.SimpleAttributeSet;

import javax.swing.text.StyleConstants;

import Client.connection;

import Info.Usr\_Info;

import MainPage.MainFace;

public class blog extends JPanel {

// 设置字体的格式

Usr\_Info user;

String[] str\_Name = { "宋体", "黑体", "Dialog", "Arial" };

String[] str\_Size = { "12", "14", "18", "22", "30", "40" };

String[] str\_Style = { "常规", "斜体", "粗体", "粗斜体" };

String[] str\_Color = { "黑色", "红色", "蓝色", "黄色", "绿色" };

File imgFile; // 保存图片

JPanel contentPane; // 整个界面，所有的部件都放在其上

JFrame fr\_main; // 保存主页面的引用

JTextPane textPane\_title = new JTextPane(); // 输入帖子题目的部分

JTextPane textPane = new JTextPane(); // 输入帖子的部分

// 方便写HTML文件

String head = ""; // 保存HTML内正文前的部分，如<p><b> 等

String tail = "</font>"; // 保存HTML内正文后的部分，如</p></b>等

String cont;

File html\_file; // 帖子存入的HTML文件

BufferedWriter write; // 写帖子的流

int pre\_pos = 0; // 帖子内容是一段话一段话保存的，pre\_pos存当前待保存段的起始位置

int cur\_pos = 0; // 当前待保存段的结束位子

int img\_no = 0; // 一共存了几张图

/\*\*

\* Create the frame.

\*

\* @throws IOException

\*/

public blog(JFrame main, Usr\_Info user\_, connection con) throws IOException {

user = user\_;

fr\_main = main;

((MainFace) fr\_main).newblog = this;

// 路径需要设置,初始化html文件

deleteDirectory("./Yopin/tmp");

File dir = new File("./Yopin/tmp");

dir.mkdirs();

html\_file = new File("./Yopin/tmp/blog.html");

if (!html\_file.exists()) {

html\_file.createNewFile();

}

write = new BufferedWriter(new FileWriter("./Yopin/tmp/blog.html"));

write.write(

"<script language=javascript> function hero() { var herowidth=560; var heroheight=350; window.resizeTo(herowidth,heroheight);}hero();</script><body onresize=hero();>");

write.write("<p><font size=\"4\" face=\"arial\" color=\"black\">");

head = "";

tail = "";

// 初始化html文件结束

// 设置整体布局、大小

setBounds(0, 0, 640, 500);

setLayout(null);

contentPane = new JPanel();

contentPane.setVisible(true);

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

this.add(contentPane);

contentPane.setBounds(0, 0, 640, 500);

contentPane.setLayout(null);

// 帖子的标题

JLabel lblTitle = new JLabel("新的博客");

lblTitle.setBounds(240, 10, 200, 40);

lblTitle.setFont(new Font("Xingkai SC", Font.ITALIC, 40));

lblTitle.setForeground(Color.DARK\_GRAY);

contentPane.add(lblTitle);

// 设置输入帖子标题的部件

JLabel lblAddTitle = new JLabel("标题");

lblAddTitle.setBounds(10, 47, 65, 50);

contentPane.add(lblAddTitle);

lblAddTitle.setFont(new Font("Xingkai SC", Font.ITALIC, 25));

textPane\_title.setBounds(85, 55, 300, 35);

contentPane.add(textPane\_title);

textPane\_title.setFont(new Font("Xingkai SC", Font.BOLD, 20));

// 设置插入图片的部件

JButton lblImg = new JButton();

lblImg.setBounds(520, 47, 100, 50);

lblImg.setOpaque(false);

lblImg.setText("添加图片");

contentPane.add(lblImg);

// 点击添加图片从计算机内选择图片

lblImg.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

JFileChooser select = new JFileChooser();

select.showDialog(select, "Select Photo");

File file = select.getSelectedFile();

if (file.exists()) {

try {

// 将当前段存入HTML文件内

String txt = textPane.getText();

cur\_pos = txt.length();

String tmp = txt.substring(pre\_pos);

for (int i = 0; i < tmp.length(); ++i) {

if (tmp.charAt(i) == '\n') {

write.write("<br/>");

} else {

write.write(tmp.charAt(i));

}

}

write.write(tail);

write.write("</p><p><img src=" + img\_no + ".jpg\" width=\"550\" height=\"350\"></p><p>");

// 将选择的图片存入缓存文件夹内，方便上传

copyFile(file.getPath(), "./Yopin/tmp/" + img\_no + ".jpg");

txt += "\npicture " + img\_no + "\n";

img\_no++;

pre\_pos = txt.length();

write.write(head);

textPane.setText(txt);

// 将选择的图片显示在小框内，共用户查看

Image img = ImageIO.read(file);

BufferedImage image = new BufferedImage(30, 30, BufferedImage.TYPE\_INT\_RGB);

Graphics g = image.getGraphics();

g.drawImage(img, 10, 158, 30, 30, Color.LIGHT\_GRAY, null);

g.dispose();

imgFile = file;

} catch (IOException event) {

JOptionPane.showMessageDialog(null, "1", "1", JOptionPane.ERROR\_MESSAGE);

}

} else {

JOptionPane.showMessageDialog(null, "22", "22", JOptionPane.ERROR\_MESSAGE);

}

}

});

// 选择发帖类型的部件

JLabel lbl\_kind = new JLabel("发帖类型");

lbl\_kind.setBounds(430, 47, 70, 21);

lbl\_kind.setOpaque(true);

contentPane.add(lbl\_kind);

JComboBox Jkind = new JComboBox(Community.Kind);

Jkind.setOpaque(false);

Jkind.setBounds(425, 69, 90, 21);

Jkind.setOpaque(true);

contentPane.add(Jkind);

// 编辑文本的部件，可滚动

JPanel panel = new JPanel();

panel.setBounds(10, 110, 764, 300);

contentPane.add(panel);

panel.setLayout(null);

JScrollPane scrollPane = new JScrollPane();

scrollPane.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_NEVER);

scrollPane.setBounds(0, 0, 605, 371);

scrollPane.getVerticalScrollBar().setUnitIncrement(15);

panel.add(scrollPane);

textPane.setBounds(0, 0, 600, 2000);

scrollPane.setViewportView(textPane);

// 选择字体颜色的部件

JComboBox fontColor = new JComboBox(str\_Color);

fontColor.setBounds(10, 430, 80, 21);

contentPane.add(fontColor);

// 选择字体样式的部件

JComboBox fontName = new JComboBox(str\_Name);

fontName.setBounds(97, 430, 80, 21);

contentPane.add(fontName);

// 选择字体大小的部件

JComboBox fontSize = new JComboBox(str\_Size);

fontSize.setBounds(184, 430, 80, 21);

contentPane.add(fontSize);

// 选择是否加粗、斜的部件

JComboBox fontStyle = new JComboBox(str\_Style);

fontStyle.setBounds(261, 430, 80, 21);

contentPane.add(fontStyle);

// 保存字体设置，点此按钮后，接下来的输入就会按照改后的字体格式

Button change = new Button("保存设置");

change.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

setFontAttrib(fontName, fontSize, fontStyle, fontColor);

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

change.setBounds(363, 430, 116, 23);

contentPane.add(change);

// 取消发帖部件，返回上级菜单

Button cancel = new Button("取消");

cancel.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

((MainFace) fr\_main).newblog.setVisible(false);

((MainFace) fr\_main).community.setVisible(true);

}

});

cancel.setBounds(480, 430, 57, 23);

contentPane.add(cancel);

// 发表部件，发表帖子。同时将内容上传服务器

Button publish = new Button("发表");

publish.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

String txt = textPane.getText(); // 将当前段落存入HTML

cur\_pos = txt.length();

String tmp = txt.substring(pre\_pos);

for (int i = 0; i < tmp.length(); ++i) {

if (tmp.charAt(i) == '\n') {

write.write("<br/>");

} else {

write.write(tmp.charAt(i));

}

}

write.write(tail);

write.write("</p></body>");

write.flush();

write.close();

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

// 将帖子和摘要发送给服务器

blogAbstract abs = new blogAbstract();

abs.kind = Jkind.getSelectedIndex() + 1;

blogText t = new blogText();

Date date = new Date();

abs.authorName = user.getName();

abs.commentID = 0;

DateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm");

abs.time = format.format(date);

abs.title = textPane\_title.getText();

abs.id = abs.latestID + 1;

t.text = textPane.getText();

try {

if (img\_no == 0) {

copyFile("./YoPin/Pictures/YoPin0/user\_image.jpg", "./Yopin/tmp/0.jpg");

}

con.createblog(abs, "./Yopin/tmp", new File("./Yopin/tmp/0.jpg"));

} catch (Exception ep) {

ep.printStackTrace();

}

((MainFace) fr\_main).newblog.setVisible(false);

((MainFace) fr\_main).community = new Community(user\_, con, fr\_main);

((MainFace) fr\_main).lbl\_RightInfo.add(((MainFace) fr\_main).community);

JOptionPane.showMessageDialog(null, "发表成功");

}

});

publish.setBounds(555, 430, 57, 23);

contentPane.add(publish);

}

// 修改字体的函数

void setFontAttrib(JComboBox fontName, JComboBox fontSize, JComboBox fontStyle, JComboBox fontColor)

throws IOException {

String txt = textPane.getText();

cur\_pos = txt.length();

String tmp = txt.substring(pre\_pos);

for (int i = 0; i < tmp.length(); ++i) {

if (tmp.charAt(i) == '\n') {

write.write("<br/>");

} else {

write.write(tmp.charAt(i));

}

}

write.write(tail);

pre\_pos = cur\_pos;

System.out.println("change font");

System.out.println(txt);

SimpleAttributeSet attr = new SimpleAttributeSet();

StyleConstants.setBold(attr, true);

StyleConstants.setItalic(attr, true);

// change font style

String temp\_style = (String) fontStyle.getSelectedItem();

if (temp\_style.equals("常规")) {

StyleConstants.setBold(attr, false);

StyleConstants.setItalic(attr, false);

head = "";

tail = "";

} else if (temp\_style.equals("黑体")) {

StyleConstants.setBold(attr, true);

StyleConstants.setItalic(attr, false);

head = "<b>";

tail = "</b>";

} else if (temp\_style.equals("斜体")) {

StyleConstants.setBold(attr, false);

StyleConstants.setItalic(attr, true);

head = "<i>";

tail = "</i>";

} else if (temp\_style.equals("黑斜体")) {

StyleConstants.setBold(attr, true);

StyleConstants.setItalic(attr, true);

head = "<b><i>";

tail = "</i></b>";

}

// change font size

int size = Integer.parseInt((String) fontSize.getSelectedItem());

StyleConstants.setFontSize(attr, size);

size -= 8;

head += "<font size=\"" + size + "\" ";

// change font color

String temp\_color = (String) fontColor.getSelectedItem();

Color color = null;

if (temp\_color.equals("黑色")) {

color = new Color(0, 0, 0);

head += "color=\"black\">";

} else if (temp\_color.equals("红色")) {

color = new Color(255, 0, 0);

head += "color=\"red\">";

} else if (temp\_color.equals("蓝色")) {

color = new Color(0, 0, 255);

head += "color=\"blue\">";

} else if (temp\_color.equals("黄色")) {

color = new Color(255, 255, 0);

head += "color=\"yellow\">";

} else if (temp\_color.equals("绿色")) {

color = new Color(0, 255, 0);

head += "color=\"green\">";

}

tail += "</font>";

StyleConstants.setForeground(attr, color);

// change font name

String name = (String) fontName.getSelectedItem();

StyleConstants.setFontFamily(attr, name);

// save the change

write.write(head);

textPane.setCharacterAttributes(attr, true);

}

private void copyFile(String src, String dst) throws IOException {

System.out.println(src);

File f1 = new File(src);

File f2 = new File(dst);

BufferedInputStream read = null;

BufferedOutputStream w = null;

if (!f1.exists())

return;

if (!f2.exists())

f2.createNewFile();

try {

read = new BufferedInputStream(new FileInputStream(src));

w = new BufferedOutputStream(new FileOutputStream(dst));

byte buf[] = new byte[85];

int n = 0;

n = read.read(buf, 0, 80);

while (n != -1) {

w.write(buf, 0, n);

n = read.read(buf, 0, 80);

}

w.flush();

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

// 删除当前目录

public boolean deleteDirectory(String sPath) {

// 如果sPath不以文件分隔符结尾，自动添加文件分隔符

if (!sPath.endsWith(File.separator)) {

sPath = sPath + File.separator;

}

File dirFile = new File(sPath);

// 如果dir对应的文件不存在，或者不是一个目录，则退出

if (!dirFile.exists() || !dirFile.isDirectory()) {

return false;

}

boolean flag = true;

// 删除文件夹下的所有文件(包括子目录)

File[] files = dirFile.listFiles();

for (int i = 0; i < files.length; i++) {

// 删除子文件

if (files[i].isFile()) {

flag = deleteFile(files[i].getAbsolutePath());

if (!flag)

break;

} // 删除子目录

else {

flag = deleteDirectory(files[i].getAbsolutePath());

if (!flag)

break;

}

}

if (!flag)

return false;

// 删除当前目录

if (dirFile.delete()) {

return true;

} else {

return false;

}

}

// 删除文件

public boolean deleteFile(String sPath) {

boolean flag = false;

File file = new File(sPath);

// 路径为文件且不为空则进行删除

if (file.isFile() && file.exists()) {

file.delete();

flag = true;

}

return flag;

}

}

blogabstract.java

package Community;

public class blogAbstract {

public static int latestID = -1;

// 评论数量

public int commentID = -1;

// 帖子的抽象摘要

public int id; // 帖子id

public String title; // 题目

public String authorName; // 名字

public String time; // 时间

public String filename; // icon图片路径

public int kind = 1;// 帖子类别

public blogAbstract() {

id = -1; //

filename = null;

}

}

Community.java

package Community;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.FontMetrics;

import java.awt.GridLayout;

import java.awt.Image;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.io.IOException;

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*

\* make changes by ZSZ

\* add listener to btnMypage

\* change address(not absolute)

\*/

/\*\*

\*

\* @author zqh

\*/

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.ScrollPaneConstants;

import Client.connection;

import Info.Usr\_Info;

import MainPage.MainFace;

public class Community extends JPanel {

// 部件

static final String Kind[] = { "娱乐", "科技", "汽车", "体育", "财经", "军事" };

private connection con;

int ID;

JPanel thisPanel;

private String usrdir = "./src/usr/";

private Usr\_Info user;

JLabel background;

JScrollPane s;

JPanel mains;

public JButton btnMyPage;

ShowBlog blPage;

JPanel title;

int textkind = 1;

JFrame frm\_main;

JLabel Ytitle;

JPanel kindBack;

JPanel kind;

JLabel[] Tkind = new JLabel[6];

// Label内文字自动换行，用HTML实现

void JlabelSetText(JLabel jLabel, String longString) {

StringBuilder builder = new StringBuilder("<html>");

char[] chars = longString.toCharArray();

FontMetrics fontMetrics = jLabel.getFontMetrics(jLabel.getFont());

int start = 0;

int len = 0;

while (start + len < longString.length()) {

while (true) {

len++;

if (start + len > longString.length())

break;

if (fontMetrics.charsWidth(chars, start, len) > jLabel.getWidth()) {

break;

}

}

builder.append(chars, start, len - 1).append("<br>");

start = start + len - 1;

len = 0;

}

builder.append(chars, start, longString.length() - start);

builder.append("</html>");

jLabel.setText(builder.toString());

}

// 初始化帖子摘要界面

private void initBlogPane(int blogID, int id) {

blogAbstract blog = con.getBlogText(textkind, blogID);

blog.id = blogID;

JPanel content = new JPanel();

content.setLayout(null);

// 设置帖子的图片

if (blog.filename != null) {

ImageIcon img = new ImageIcon(blog.filename);

img.setImage(img.getImage().getScaledInstance(100, 100, Image.SCALE\_DEFAULT));

JLabel lblIcon = new JLabel();

lblIcon.setIcon(img);

lblIcon.setBounds(0, 0, 100, 100);

content.add(lblIcon);

}

// 输出帖子的摘要，时间、作者、内容等

JLabel lblText = new JLabel();

lblText.setFont(new Font("华文仿宋", 0, 18));

lblText.setBounds(110, 0, 500, 60);

JLabel lblAuthor = new JLabel("作者：" + blog.authorName);

lblAuthor.setBounds(365, 60, 150, 20);

lblAuthor.setForeground(Color.GRAY);

lblAuthor.setFont(new Font("", 0, 13));

JLabel lblTime = new JLabel("发布时间：" + blog.time);

lblTime.setForeground(Color.GRAY);

lblTime.setFont(new Font("", 0, 13));

lblTime.setBounds(365, 80, 250, 20);

JlabelSetText(lblText, blog.title);

content.add("Center", lblText);

content.add("South", lblTime);

content.add(lblAuthor);

// 点帖子的话，进入帖子的详细内容

content.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

blog.kind = textkind;

blPage = new ShowBlog(blog, user, con);

blPage.setBounds(0, 0, 640, 500);

setVisible(false);

blPage.setVisible(true);

blPage.btnBack.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

blPage.setVisible(false);

setVisible(true);

}

});

((MainFace) frm\_main).lbl\_RightInfo.add(blPage);

}

});

// 将新的帖子摘要加入主界面

content.setBackground(Color.white);

mains.add(content);

}

// 显示帖子摘要的主界面

public Community(Usr\_Info user\_, connection con\_, JFrame frm\_main\_) {

this.setOpaque(true);

this.setForeground(Color.black);

this.setBackground(Color.WHITE);

thisPanel = this;

user = user\_;

con = con\_;

frm\_main = frm\_main\_;

((MainFace) frm\_main\_).community = thisPanel;

this.setFont(new Font("Xingkai SC", Font.BOLD, 18));

// 放多个帖子摘要的部件

s = new JScrollPane();

s.getVerticalScrollBar().setUnitIncrement(20);

s.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_NEVER);

this.setLayout(null);

this.setBounds(0, 0, 640, 500);

this.add(s);

s.setBounds(20, 80, 620, 420);

s.getViewport().setLayout(null);

s.setBorder(null);

// 界面上的题目、种类

title = new JPanel();

title.setLayout(null);

Ytitle = new JLabel("悠品");

Ytitle.setForeground(Color.WHITE);

Ytitle.setFont(new Font("华文仿宋", 1, 30));

Ytitle.setBounds(20, 15, 100, 25);

title.add(Ytitle);

title.setBackground(new Color(183, 39, 18));

title.setBounds(0, 0, 640, 50);

// 提供选择不同种类帖子的部件

kindBack = new JPanel();

kindBack.setBounds(0, 50, 640, 30);

kindBack.setLayout(null);

kind = new JPanel();

kind.setLayout(new GridLayout(1, 6));

kind.setBounds(20, 0, 600, 30);

for (int i = 0; i < 6; ++i) {

Tkind[i] = new JLabel();

Tkind[i].setFont(new Font("", 0, 20));

Tkind[i].setText(Kind[i]);

Tkind[i].setHorizontalAlignment(JLabel.CENTER);

Tkind[i].addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

for (int j = 0; j < 6; ++j) {

if (e.getSource() == Tkind[j]) {

Tkind[j].setForeground(Color.red);

textkind = j + 1;

} else

Tkind[j].setForeground(Color.BLACK);

}

}

});

kind.add(Tkind[i]);

}

Tkind[0].setForeground(Color.red);

kindBack.add(kind);

this.add(kindBack);

this.add(title);

// 放多个帖子摘要的部件

mains = new JPanel();

mains.setPreferredSize(new Dimension(760, 2000));

mains.setBounds(0, 0, 660, 2000);

mains.setLayout(new GridLayout(20, 1, 0, 2));

// 从服务器读取若干帖子摘要，将其显示

blogAbstract.latestID = con.blognumber(textkind);

ID = blogAbstract.latestID;

int maxnum = ID;

if (ID > 19)

maxnum = 19;

// 每次创建一个新的帖子摘要，加入主界面

for (int i = 0; i < maxnum; ++i) {

initBlogPane(ID - i, i);

}

JPanel content = new JPanel();

String str = "<html>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;More <html/>";

content.setLayout(new BorderLayout());

// More按钮，点击可以刷新显示的帖子

JLabel btnMore = new JLabel(str);

btnMore.setHorizontalAlignment(JLabel.CENTER);

btnMore.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

if (ID < 19)

return;

ID -= 19;

mains = new JPanel();

mains.setPreferredSize(new Dimension(760, 2000));

mains.setBounds(0, 0, 760, 2000);

mains.setLayout(new GridLayout(20, 1, 2, 0));

int max\_num = ID;

if (ID >= 19)

max\_num = 19;

for (int i = 0; i < max\_num; ++i) {

initBlogPane(ID - i, i);

}

mains.add(content);

s.setViewportView(mains);

}

});

content.add("Center", btnMore);

mains.add(content);

s.setViewportView(mains);

this.add(s);

// 创建博客按钮，点击可以创建新的博客

JButton btnBlog = new JButton("创建博客");

btnBlog.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

blog b;

try {

b = new blog(frm\_main\_, user\_, con);

((MainFace) frm\_main\_).lbl\_RightInfo.add(b);

((MainFace) frm\_main\_).community.setVisible(false);

b.setVisible(true);

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

});

btnBlog.setBounds(540, 10, 100, 34);

title.add(btnBlog);

// 标题

btnMyPage = new JButton("\u6211\u7684\u4E3B\u9875");

btnMyPage.setBounds(447, 417, 113, 34);

this.add(btnMyPage);

this.setVisible(true);

}

}

Showblog.java

package Community;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.FontMetrics;

import java.awt.GridLayout;

import java.awt.Image;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.io.IOException;

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*

\* make changes by ZSZ

\* add listener to btnMypage

\* change address(not absolute)

\*/

/\*\*

\*

\* @author zqh

\*/

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.ScrollPaneConstants;

import Client.connection;

import Info.Usr\_Info;

import MainPage.MainFace;

public class Community extends JPanel {

// 部件

static final String Kind[] = { "娱乐", "科技", "汽车", "体育", "财经", "军事" };

private connection con;

int ID;

JPanel thisPanel;

private String usrdir = "./src/usr/";

private Usr\_Info user;

JLabel background;

JScrollPane s;

JPanel mains;

public JButton btnMyPage;

ShowBlog blPage;

JPanel title;

int textkind = 1;

JFrame frm\_main;

JLabel Ytitle;

JPanel kindBack;

JPanel kind;

JLabel[] Tkind = new JLabel[6];

// Label内文字自动换行，用HTML实现

void JlabelSetText(JLabel jLabel, String longString) {

StringBuilder builder = new StringBuilder("<html>");

char[] chars = longString.toCharArray();

FontMetrics fontMetrics = jLabel.getFontMetrics(jLabel.getFont());

int start = 0;

int len = 0;

while (start + len < longString.length()) {

while (true) {

len++;

if (start + len > longString.length())

break;

if (fontMetrics.charsWidth(chars, start, len) > jLabel.getWidth()) {

break;

}

}

builder.append(chars, start, len - 1).append("<br>");

start = start + len - 1;

len = 0;

}

builder.append(chars, start, longString.length() - start);

builder.append("</html>");

jLabel.setText(builder.toString());

}

// 初始化帖子摘要界面

private void initBlogPane(int blogID, int id) {

blogAbstract blog = con.getBlogText(textkind, blogID);

blog.id = blogID;

JPanel content = new JPanel();

content.setLayout(null);

// 设置帖子的图片

if (blog.filename != null) {

ImageIcon img = new ImageIcon(blog.filename);

img.setImage(img.getImage().getScaledInstance(100, 100, Image.SCALE\_DEFAULT));

JLabel lblIcon = new JLabel();

lblIcon.setIcon(img);

lblIcon.setBounds(0, 0, 100, 100);

content.add(lblIcon);

}

// 输出帖子的摘要，时间、作者、内容等

JLabel lblText = new JLabel();

lblText.setFont(new Font("华文仿宋", 0, 18));

lblText.setBounds(110, 0, 500, 60);

JLabel lblAuthor = new JLabel("作者：" + blog.authorName);

lblAuthor.setBounds(365, 60, 150, 20);

lblAuthor.setForeground(Color.GRAY);

lblAuthor.setFont(new Font("", 0, 13));

JLabel lblTime = new JLabel("发布时间：" + blog.time);

lblTime.setForeground(Color.GRAY);

lblTime.setFont(new Font("", 0, 13));

lblTime.setBounds(365, 80, 250, 20);

JlabelSetText(lblText, blog.title);

content.add("Center", lblText);

content.add("South", lblTime);

content.add(lblAuthor);

// 点帖子的话，进入帖子的详细内容

content.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

blog.kind = textkind;

blPage = new ShowBlog(blog, user, con);

blPage.setBounds(0, 0, 640, 500);

setVisible(false);

blPage.setVisible(true);

blPage.btnBack.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

blPage.setVisible(false);

setVisible(true);

}

});

((MainFace) frm\_main).lbl\_RightInfo.add(blPage);

}

});

// 将新的帖子摘要加入主界面

content.setBackground(Color.white);

mains.add(content);

}

// 显示帖子摘要的主界面

public Community(Usr\_Info user\_, connection con\_, JFrame frm\_main\_) {

this.setOpaque(true);

this.setForeground(Color.black);

this.setBackground(Color.WHITE);

thisPanel = this;

user = user\_;

con = con\_;

frm\_main = frm\_main\_;

((MainFace) frm\_main\_).community = thisPanel;

this.setFont(new Font("Xingkai SC", Font.BOLD, 18));

// 放多个帖子摘要的部件

s = new JScrollPane();

s.getVerticalScrollBar().setUnitIncrement(20);

s.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_NEVER);

this.setLayout(null);

this.setBounds(0, 0, 640, 500);

this.add(s);

s.setBounds(20, 80, 620, 420);

s.getViewport().setLayout(null);

s.setBorder(null);

// 界面上的题目、种类

title = new JPanel();

title.setLayout(null);

Ytitle = new JLabel("悠品");

Ytitle.setForeground(Color.WHITE);

Ytitle.setFont(new Font("华文仿宋", 1, 30));

Ytitle.setBounds(20, 15, 100, 25);

title.add(Ytitle);

title.setBackground(new Color(183, 39, 18));

title.setBounds(0, 0, 640, 50);

// 提供选择不同种类帖子的部件

kindBack = new JPanel();

kindBack.setBounds(0, 50, 640, 30);

kindBack.setLayout(null);

kind = new JPanel();

kind.setLayout(new GridLayout(1, 6));

kind.setBounds(20, 0, 600, 30);

for (int i = 0; i < 6; ++i) {

Tkind[i] = new JLabel();

Tkind[i].setFont(new Font("", 0, 20));

Tkind[i].setText(Kind[i]);

Tkind[i].setHorizontalAlignment(JLabel.CENTER);

Tkind[i].addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

for (int j = 0; j < 6; ++j) {

if (e.getSource() == Tkind[j]) {

Tkind[j].setForeground(Color.red);

textkind = j + 1;

} else

Tkind[j].setForeground(Color.BLACK);

}

}

});

kind.add(Tkind[i]);

}

Tkind[0].setForeground(Color.red);

kindBack.add(kind);

this.add(kindBack);

this.add(title);

// 放多个帖子摘要的部件

mains = new JPanel();

mains.setPreferredSize(new Dimension(760, 2000));

mains.setBounds(0, 0, 660, 2000);

mains.setLayout(new GridLayout(20, 1, 0, 2));

// 从服务器读取若干帖子摘要，将其显示

blogAbstract.latestID = con.blognumber(textkind);

ID = blogAbstract.latestID;

int maxnum = ID;

if (ID > 19)

maxnum = 19;

// 每次创建一个新的帖子摘要，加入主界面

for (int i = 0; i < maxnum; ++i) {

initBlogPane(ID - i, i);

}

JPanel content = new JPanel();

String str = "<html>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;More <html/>";

content.setLayout(new BorderLayout());

// More按钮，点击可以刷新显示的帖子

JLabel btnMore = new JLabel(str);

btnMore.setHorizontalAlignment(JLabel.CENTER);

btnMore.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

if (ID < 19)

return;

ID -= 19;

mains = new JPanel();

mains.setPreferredSize(new Dimension(760, 2000));

mains.setBounds(0, 0, 760, 2000);

mains.setLayout(new GridLayout(20, 1, 2, 0));

int max\_num = ID;

if (ID >= 19)

max\_num = 19;

for (int i = 0; i < max\_num; ++i) {

initBlogPane(ID - i, i);

}

mains.add(content);

s.setViewportView(mains);

}

});

content.add("Center", btnMore);

mains.add(content);

s.setViewportView(mains);

this.add(s);

// 创建博客按钮，点击可以创建新的博客

JButton btnBlog = new JButton("创建博客");

btnBlog.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

blog b;

try {

b = new blog(frm\_main\_, user\_, con);

((MainFace) frm\_main\_).lbl\_RightInfo.add(b);

((MainFace) frm\_main\_).community.setVisible(false);

b.setVisible(true);

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

});

btnBlog.setBounds(540, 10, 100, 34);

title.add(btnBlog);

// 标题

btnMyPage = new JButton("\u6211\u7684\u4E3B\u9875");

btnMyPage.setBounds(447, 417, 113, 34);

this.add(btnMyPage);

this.setVisible(true);

}

}