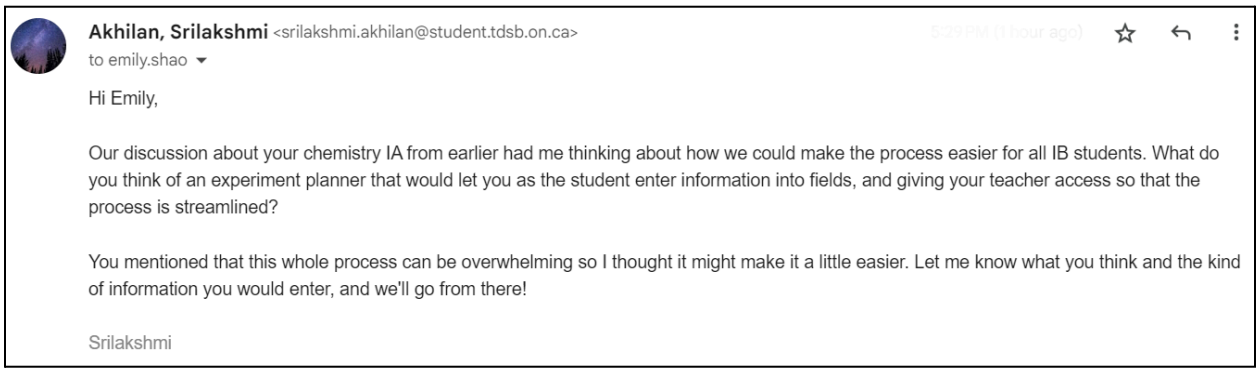
Appendix A

Email Screenshots between Client & Programmer

Date: September 19th 2023



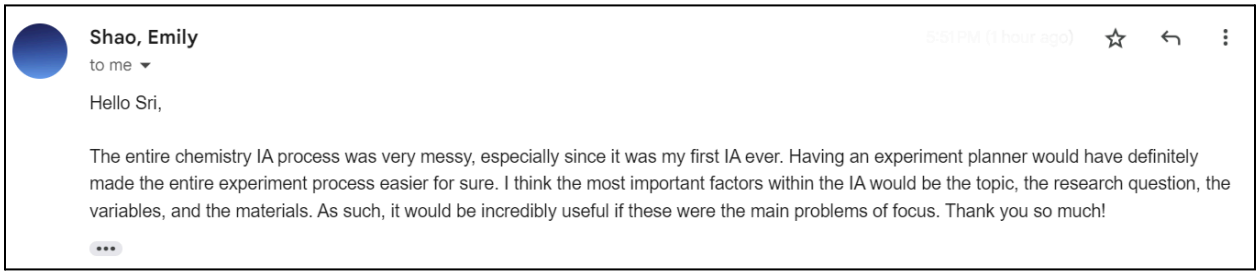
Akhilan, Srilakshmi <srilakshmi.akhilan@student.tdsb.on.ca>
5:29 PM (1 hour ago)
to emily.shao

Hi Emily,

Our discussion about your chemistry IA from earlier had me thinking about how we could make the process easier for all IB students. What do you think of an experiment planner that would let you as the student enter information into fields, and giving your teacher access so that the process is streamlined?

You mentioned that this whole process can be overwhelming so I thought it might make it a little easier. Let me know what you think and the kind of information you would enter, and we'll go from there!

Srilakshmi



Shao, Emily
5:51 PM (1 hour ago)
to me

Hello Sri,

The entire chemistry IA process was very messy, especially since it was my first IA ever. Having an experiment planner would have definitely made the entire experiment process easier for sure. I think the most important factors within the IA would be the topic, the research question, the variables, and the materials. As such, it would be incredibly useful if these were the main problems of focus. Thank you so much!

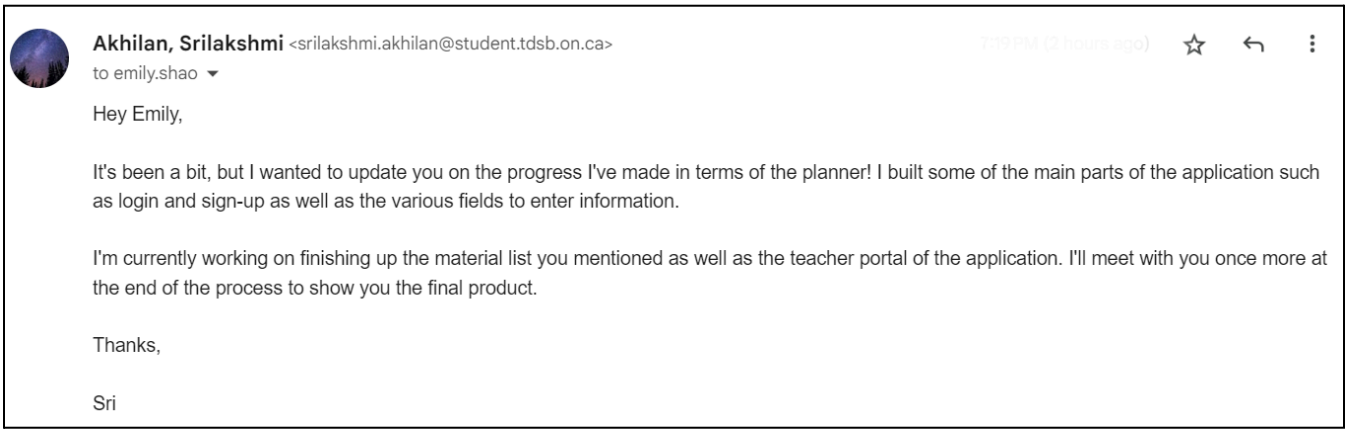
In-Person Discussion Transcript

Date: September 20th 2023

```
Discussion Transcript

Sri: Hey Emily, I got your ideas on the experiment features, but I was wondering if you would be able to provide me with an
inventory list to get a sense of the materials we're working with.

Emily: No problem, I have a list available and I'll be sure to send it to you! I will say, the labs usually have a fair
number of materials, so I want the list to be clear and simple so I can quickly find the materials I need.

Sri: That shouldn't be a problem, are there any other features you would like?

Emily: I've always found apps with different graphic features to be cool, images, and different components would be nice?

Sri: Sounds great, I'll let you know how the process goes!

Emily: Thanks!
```

Appendix B

Progress Update Email - (from programmer to client)
Date: Feb 26th 2024



Akhilan, Srilakshmi <srilakshmi.akhilan@student.tdsb.on.ca> 7:19 PM (2 hours ago)
to emily.shao

Hey Emily,

It's been a bit, but I wanted to update you on the progress I've made in terms of the planner! I built some of the main parts of the application such as login and sign-up as well as the various fields to enter information.

I'm currently working on finishing up the material list you mentioned as well as the teacher portal of the application. I'll meet with you once more at the end of the process to show you the final product.

Thanks,

Sri

Appendix C

Final Consultation with Client

Date: Mar 6th 2024

Client Interview Transcript

Sri: Hi Emily, I see you've used the program, what are your thoughts?

Emily: Well, as someone who had to already experience the messy process of planning my chemistry IA and having it planned with my teacher, I actually think this program suits the needs of many students who were in my position.

Sri: That's great, do you have anything in specific you liked?

Emily: I personally found that the entering and saving of information for the student was fairly simple, which I definitely appreciate as someone who isn't as technologically advanced. The separate fields and clear instructions were also really helpful.

Sri: I'm glad it worked for you! Are there any thoughts on the teacher end?

Emily: Honestly, I found that feature to be super cool, because it makes the whole process so much simpler. I think the links between the student and teacher profiles eases up the time that it takes to review IA plans especially because I've been on tight deadlines before and experience major procrastination.

Sri: Yeah, I get that. Would you say the application was similar to your own idea/design?

Emily: I definitely think it had most of the elements I wanted, however, I wish you had more graphics, like you did on the menu page because it made the app more appealing especially since it's for school purposes.

Sri: I appreciate the feedback! Thank you for your time!

Appendix D

ChemPlanApp Class Code

```
/**
 * ICS4U7 IA: Chemistry IA Planner
 * Author: Srilakshmi Akhilan
 * Date: 04/03/2024
 */

// Import the following packages:
import java.awt.*;

import javax.swing.*;

/**
 * The following class helps run the main program
 * @author Sri
 *
 */
public class ChemPlanApp extends JFrame {

	static CardLayout cl;
	static Container c;

	WelcomePanel welcomeP;
	AboutPanel aboutP;
	AppMenuPanel appMenuP;
	SignUpPanel signUpP;
	LoginPanel loginP;
	StudentPlannerPanel studentPlanP;
	IAPanel iaP;
	MaterialsPanel matP;
	TeacherPlannerPanel teacherPlanP;
	LogoutPanel logoutP;
	ReviewIAPanel reviewIAP;

	public ChemPlanApp() {

		this.setTitle("don't bomb it");
		this.setVisible(true);
		this.setResizable(false);
//		this.setLocationRelativeTo(null);
```

```
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        c = getContentPane();
        cl = new CardLayout();
        c.setLayout(cl);

        this.setSize(700, 550);
        c.setBackground(new Color(119, 158, 186));

        welcomeP = new WelcomePanel();
        c.add("welcomePage", welcomeP);

        aboutP = new AboutPanel();
        c.add("aboutPage", aboutP);

        appMenuP = new AppMenuPanel();
        c.add("appMenuPage", appMenuP);

        signUpP = new SignUpPanel();
        c.add("signUpPage", signUpP);

        loginP = new LoginPanel();
        c.add("loginPage", loginP);

        logoutP = new LogoutPanel();
        c.add("logoutPage", logoutP);

    }

    public static void main(String[] args) {
        new ChemPlanApp();
    }
}
```

Appendix E

AboutPanel Class Code

```
// Import the following packages:
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/**
 * The following class is for the about frame of the program
 * @author Sri
 *
 */
public class AboutPanel extends JPanel {

	private JLabel abHeader, crLbl;
	private String file, line, copyright;
	private JTextArea aboutApp;
	private JButton back;

	public AboutPanel() {
		this.setLayout(null);
		this.setBackground(new Color(119, 158, 186));

		abHeader = GUIComp.setLabel("your one stop lab planner", 150, 23, 400, 40, Font.BOLD,
35, 255, 255, 255);
		this.add(abHeader);

		file = "./aboutDBI.txt";

		aboutApp = new JTextArea();
		aboutApp.setBounds(50, 80, 600, 380);
		aboutApp.setFont(new Font("Ink Free", Font.BOLD, 16));
		aboutApp.setForeground(new Color(19, 52, 92));
		aboutApp.setOpaque(false); // make JTextArea transparent

		aboutApp.setEditable(false);
		aboutApp.setLineWrap(true);
		aboutApp.setWrapStyleWord(true);

		this.add(aboutApp);
```

```
		// Try reading in the about file and add each line to the JTextArea
		try {
			BufferedReader readInLine = new BufferedReader(new FileReader(file));

			while((line = readInLine.readLine()) != null) {
				aboutApp.read(readInLine, "aboutApp");
			}
		}

		// If file doesn't exist, catch exception and print an error message to the console
		catch (IOException iox) {
			System.out.println("Problem reading " + file);
		}

		copyright = "Copyright don't bomb it 2023. All Rights Reserved. CompSciIA2024";
		crLbl = GUIComp.setLabel(copyright, 120, 480, 430, 20, Font.PLAIN, 14, 255, 255, 255);
		this.add(crLbl);

		back = GUIComp.setButton("back", 580, 445, 80, 35, 20);
		this.add(back);

		back.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub

				ChemPlanApp.cl.show(ChemPlanApp.c, "welcomePage");
			}

		});
	}
}
```

Appendix F

AppMenuPanel Class Code

```
// Import the following packages:
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/**
 * The following class is for the program's main menu
 * @author lakhs
 *
 */
public class AppMenuPanel extends JPanel {

	private JLabel signUpHeader, selection, studentLbl, teacherLbl, chkForAccnt;
	private JButton student_btn, teacher_btn, login_btn, back_btn;

	static boolean sBtnPressed = false;
	static boolean loginPressed = false;

	public AppMenuPanel() {

		this.setLayout(null);
		this.setBackground(new Color(119, 158, 186));

		signUpHeader = GUIComp.setLabel("sign-up", 280, 30, 120, 45, Font.BOLD, 35, 255, 255,
255);
		this.add(signUpHeader);

		selection = GUIComp.setLabel("select the appropriate profile", 210, 90, 350, 30,
Font.PLAIN, 20, 245, 240, 228);
		this.add(selection);

		student_btn = GUIComp.setButton(170, 160, 130, 150, 186, 210, 232, new
ImageIcon("student.jpg"));
		this.add(student_btn);

		student_btn.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
```

```
                        // TODO Auto-generated method stub

                        sBtnPressed = true;

                        ChemPlanApp.cl.show(ChemPlanApp.c, "signUpPage");
                        // create student object
                    }

                });

                teacher_btn = GUIComp.setButton(340, 160, 180, 150, 186, 210, 232, new
ImageIcon("teacher.jpg"));
                this.add(teacher_btn);

                teacher_btn.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {
                        // TODO Auto-generated method stub

                        sBtnPressed = false;
                        ChemPlanApp.cl.show(ChemPlanApp.c, "signUpPage");
                        // create teacher object
                    }

                });

                studentLbl = GUIComp.setLabel("STUDENT", 187, 330, 120, 25, Font.PLAIN, 20, 245, 240,
228);
                this.add(studentLbl);

                teacherLbl = GUIComp.setLabel("EDUCATOR", 375, 330, 120, 25, Font.PLAIN, 20, 245,
240, 228);
                this.add(teacherLbl);

                chkForAccnt = GUIComp.setLabel("already have an account?", 170, 400, 250, 30,
Font.PLAIN, 20, 245, 240, 228);
                this.add(chkForAccnt);

                login_btn = GUIComp.setButton("login", 410, 393, 110, 42, 25);
                this.add(login_btn);
```

```
		login_btn.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub

				loginPressed = true;
				ChemPlanApp.cl.show(ChemPlanApp.c, "loginPage");
			}

		});

		back_btn = GUIComp.setButton("back", 580, 455, 80, 35, 20);
		this.add(back_btn);

		back_btn.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub

				ChemPlanApp.cl.show(ChemPlanApp.c, "welcomePage");
			}

		});
	}
}
```

Appendix G

SignUpPanel Class Code

```
// Import the following packages:
import java.util.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.io.*;

/**
 * The following class is for the sign-up frame of the application
 * @author Sri
 *
 */
public class SignUpPanel extends JPanel {

	private JLabel signUpHeader, createAccntLbl, userLbl, pwdLbl, minCharLbl, infoStatus;
	private JTextField userInput, pwdInput;
	private JButton signUpBtn, backToSignUpBtn, reLoginBtn;
	private String userField, pwdField;
	private HashMap<String, GeneralUser> info = new HashMap<String, GeneralUser>();
	private String accntFile = "./accounts.txt";

	public SignUpPanel() {

		this.setLayout(null);
		this.setBackground(new Color(119, 158, 186));

		signUpHeader = GUIComp.setLabel("sign-up", 280, 30, 120, 45, Font.BOLD, 35, 255, 255,
255);
		this.add(signUpHeader);

		createAccntLbl = GUIComp.setLabel("create your account here!", 235, 90, 330, 30,
Font.PLAIN, 20, 245, 240, 228);
		this.add(createAccntLbl);

		userLbl = GUIComp.setLabel("username", 180, 160, 110, 30, Font.PLAIN, 20, 51, 55,
138);
		pwdLbl = GUIComp.setLabel("password", 180, 210, 110, 30, Font.PLAIN, 20, 51, 55, 138);
		minCharLbl = GUIComp.setLabel("min 8 characters", 180, 240, 180, 20, Font.PLAIN, 14,
51, 55, 138);
```

```
this.add(userLbl);
this.add(pwdLbl);
this.add(minCharLbl);

userInput = new JTextField();
userInput.setBounds(290, 165, 200, 20);
userInput.setFont(new Font("Ink Free", Font.PLAIN, 15));

pwdInput = new JTextField();
pwdInput.setBounds(290, 215, 200, 20);
pwdInput.setFont(new Font("Ink Free", Font.PLAIN, 15));

this.add(userInput);
this.add(pwdInput);

signUpBtn = new JButton("complete sign-up");
signUpBtn.setBounds(225, 290, 230, 40);
signUpBtn.setFont(new Font("Ink Free", Font.BOLD, 25));
signUpBtn.setForeground(new Color(44, 79, 110));
signUpBtn.setBackground(new Color(186, 210, 232));
this.add(signUpBtn);

infoStatus = new JLabel();
infoStatus.setText("");
infoStatus.setBounds(160, 350, 420, 40);
infoStatus.setFont(new Font("Ink Free", Font.PLAIN, 20));
infoStatus.setForeground(new Color(245, 240, 228));
this.add(infoStatus);

signUpBtn.addActionListener(new ActionListener() {

	@Override
	public void actionPerformed(ActionEvent e) {
		// TODO Auto-generated method stub

		userField = userInput.getText();
		pwdField = pwdInput.getText();

		ChemPlanApp.c.repaint();

		if (userField.length() < 1 || pwdField.length() < 8) {
```

```
                    infoStatus.setText("seems like your info is incorrect, try again!");
                    infoStatus.setBounds(160, 350, 420, 40);
                }
                else {

                    if (AppMenuPanel.sBtnPressed == true) {

                        infoStatus.setText("");
                        info.put(userField, new StudentUser(userField,
pwdField));

                        // Add info to a text file
                        try {

                            // Try writing user's info into the text file
                            BufferedWriter bw = new BufferedWriter(new
FileWriter(accntFile, true));
                            bw.write(userField + " " + pwdField + " student "
+ "\n");

                            bw.close(); // close the writer after info is added

                            // Inform the user that they can now login
                            infoStatus.setText("you're ready to sign-up!");
                            infoStatus.setBounds(160, 370, 200, 40);

                            // Direct user to login
                            reLoginBtn = new JButton("login here");
                            reLoginBtn.setBounds(380, 370, 150, 40);
                            reLoginBtn.setFont(new Font("Ink Free",
Font.BOLD, 25));
                            reLoginBtn.setForeground(new Color(44, 79,
110));
                            reLoginBtn.setBackground(new Color(213, 218,
237));

                            reLoginBtn.addActionListener(new
ActionListener() {

                                @Override
```

```
                                                public void
actionPerformed(ActionEvent e) {
                                                        // TODO Auto-generated
method stub

                                                        // Link to the login page

ChemPlanApp.cl.show(ChemPlanApp.c, "loginPage");
                                                }

                                        });

                                        // Add re-login button to frame
                                        add(reLoginBtn);
                                }

                                // Else, catch the exception
                                catch(IOException iox) {
                                        System.out.println("Problem writing " +
accntFile);
                                }
                        }

                        else {

                                infoStatus.setText("");
                                info.put(userField, new TeacherUser(userField,
pwdField));

                                // Add info to a text file
                                try {

                                        // Try writing user's info into the text file
                                        BufferedWriter bw = new BufferedWriter(new
FileWriter(accntFile, true));

                                        bw.write(userField + " " + pwdField + " teacher "
+ "\n");


                                        bw.close(); // close the writer after info is added

                                        // Inform the user that they can now login
                                        infoStatus.setText("you're ready to sign-up!");
```

```
                                                infoStatus.setBounds(160, 370, 200, 40);

                                                // Direct user to login
                                                reLoginBtn = new JButton("login here");
                                                reLoginBtn.setBounds(380, 370, 150, 40);
                                                reLoginBtn.setFont(new Font("Ink Free",
Font.BOLD, 25));
                                                reLoginBtn.setForeground(new Color(44, 79,
110));
                                                reLoginBtn.setBackground(new Color(213, 218,
237));

                                                reLoginBtn.addActionListener(new
ActionListener() {

                                                        @Override
                                                        public void
actionPerformed(ActionEvent e) {
                                                                // TODO Auto-generated
method stub

                                                                // Link to the login page

ChemPlanApp.cl.show(ChemPlanApp.c, "loginPage");
                                                        }

                                                });

                                                // Add re-login button to frame
                                                add(reLoginBtn);
                                        }

                                        // Else, catch the exception
                                        catch(IOException iox) {
                                                System.out.println("Problem writing " +
accntFile);
                                        }
                                }
                        }
                }

        });
```

```
		this.add(signUpBtn);

		backToSignUpBtn = GUIComp.setButton("back to sign-up menu", 440, 455, 230, 35, 20);
		this.add(backToSignUpBtn);

		backToSignUpBtn.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub

				infoStatus.setText("");
				userInput.setText("");
				pwdInput.setText("");
				ChemPlanApp.cl.show(ChemPlanApp.c, "appMenuPage");
			}

		});
	}
}
```

Appendix H

LoginPanel Class Code

```
// Import the following packages:
import java.util.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.io.*;

/**
 * The following class is for the login page of the application
 * @author Sri
 *
 */
public class LoginPanel extends JPanel {

	private JLabel loginHeader, instrInfo, userLbl, pwdLbl, infoStatus;
	private JButton enterBtn, backToSignUpBtn;
	private String readLine, user, pwd;
	private HashMap<String, GeneralUser> info = new HashMap<String, GeneralUser>();
	private String accntFile = "./accounts.txt";
	private JTextField pwdInfo;

	static JTextField userInfo = new JTextField();

	public LoginPanel() {

		this.setLayout(null);
		this.setBackground(new Color(119, 158, 186));

		loginHeader = GUIComp.setLabel("login", 300, 30, 120, 45, Font.BOLD, 35, 255, 255,
255);
		this.add(loginHeader);

		instrInfo = GUIComp.setLabel("enter the appropriate info", 220, 90, 330, 30, Font.PLAIN,
20, 245, 240, 228);
		this.add(instrInfo);

		userLbl = GUIComp.setLabel("username", 180, 160, 110, 30, Font.PLAIN, 20, 51, 55,
138);
		this.add(userLbl);
```

```
		pwdLbl= GUIComp.setLabel("password", 180, 210, 110, 30, Font.PLAIN, 20, 51, 55, 138);
		this.add(pwdLbl);

		infoStatus = new JLabel();
		infoStatus.setBounds(180, 320, 400, 40);
		infoStatus.setFont(new Font("Ink Free", Font.PLAIN, 20));
		infoStatus.setForeground(new Color(245, 240, 228));
		this.add(infoStatus);

//		JTextField userInfo = new JTextField();
		userInfo.setBounds(290, 165, 200, 20);
		userInfo.setFont(new Font("Ink Free", Font.PLAIN, 15));

		pwdInfo = new JTextField();
		pwdInfo.setBounds(290, 215, 200, 20);
		pwdInfo.setFont(new Font("Ink Free", Font.PLAIN, 15));

		this.add(userInfo);
		this.add(pwdInfo);

		// Design enter button
		enterBtn = new JButton("enter");
		enterBtn.setBounds(280, 270, 110, 40);
		enterBtn.setFont(new Font("Ink Free", Font.BOLD, 25));
		enterBtn.setForeground(new Color(44, 79, 110));
		enterBtn.setBackground(new Color(186, 210, 232));
		this.add(enterBtn);

		enterBtn.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub

				// Try reading the accounts file and adding the info to the hash map
				try {
					BufferedReader read = new BufferedReader(new
FileReader(accntFile));
					String username = "";
					String password = "";
					String userType = "";
```

```
                                        String[] accountInfo;

                                        while ((readLine = read.readLine()) != null) {
                                                accountInfo = readLine.split(" "); // split the info in the
array w/ a space

                                                // Store username and password info in the array
                                                username = accountInfo[0];
                                                password = accountInfo[1];
                                                userType = accountInfo[2];

                                                // Add account information to hash map
                                                if (userType.equals("student") && info.get(username)
== null) {
                                                        info.put(username, new
StudentUser(username, password));
                                                } else if (info.get(username) == null) {
                                                        info.put(username, new
TeacherUser(username, password));
                                                }

                                        }

                                        read.close();

                                }

                                // If the file doesn't exist, catch the exception and print an error
message
                                catch (IOException iox) {
                                        System.out.println("Problem reading " + accntFile);
                                }

                                        user = userInfo.getText();
                                        pwd = pwdInfo.getText();

                                        if (info.containsKey(user)) {
                                                if (info.get(user).getPassword().equals(pwd)) {

                                                        // Link to user's profile page
                                                        if (info.get(user).GetIsStudent()) {
```

```
                                                    StudentPlannerPanel studentPlanP =
new StudentPlannerPanel(user);

ChemPlanApp.c.add("studentPlannerPage", studentPlanP);
                                                    ChemPlanApp.cl.show(ChemPlanApp.c,
"studentPlannerPage");
                                                }
                                                else {
                                                    TeacherPlannerPanel teacherPlanP =
new TeacherPlannerPanel(user);

ChemPlanApp.c.add("teacherPlannerPage", teacherPlanP);
                                                    ChemPlanApp.cl.show(ChemPlanApp.c,
"teacherPlannerPage");
                                                }
                                            }

                                            else {
                                                infoStatus.setBounds(220, 320, 400, 40);
                                                infoStatus.setText("wrong password, try
again!");

                                                userInfo.setText("");
                                                pwdInfo.setText("");

                                            }
                                        }

                                        else {
                                            infoStatus.setBounds(180, 320, 400, 40);
                                            infoStatus.setText("this account doesn't exist, try
again!");

                                            userInfo.setText("");
                                            pwdInfo.setText("");

                                        }

                                        userInfo.setText("");
                                        pwdInfo.setText("");

                                    }
```

```
		});

		backToSignUpBtn = GUIComp.setButton("back to sign-up menu", 440, 455, 230, 35, 20);
		this.add(backToSignUpBtn);

		backToSignUpBtn.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub

				infoStatus.setText("");
				userInfo.setText("");
				pwdInfo.setText("");
				ChemPlanApp.cl.show(ChemPlanApp.c, "appMenuPage");
			}

		});
	}
}
```

Appendix I

LogoutPanel Class Code

```
// Import the following packages:
import java.awt.*;
import javax.swing.*;

/**
 * The following class is for the logout page of the application
 * @author Sri
 *
 */
public class LogoutPanel extends JPanel {

	private JLabel logoutLbl;

	public LogoutPanel() {

		this.setLayout(null);
		this.setBackground(new Color(119, 158, 186));

		logoutLbl = GUIComp.setLabel("you've signed out successfully, see you later!", 105, 230,
470, 30, Font.PLAIN, 25, 51, 55, 138);
		this.add(logoutLbl);
	}
}
```

Appendix J

StudentPlannerPanel Class Code

```
// Import the following packages:
import java.io.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

/**
 * The following class is for the student's planning page
 * @author Sri
 *
 */
public class StudentPlannerPanel extends JPanel {

	private JLabel planHeader, usernameLbl, stdNumLbl, stdEmailLbl, stdLevelLbl, iaCreationH;
	private String userNm, studentNum, studentEmail;
	private JTextField stdNum, stdEmail;
	private JRadioButton sl, hl;
	private JButton createIABtn, logoutBtn, saveBtn;

	static String IAFileInfo;

	public StudentPlannerPanel(String username) {

		this.setLayout(null);
		this.setBackground(new Color(119, 158, 186));

		planHeader = GUIComp.setLabel("your planner", 240, 50, 240, 45, Font.BOLD, 35, 255,
255, 255);
		this.add(planHeader);

		userNm = username;

		usernameLbl = GUIComp.setLabel("hello, " + userNm, 20, 20, 200, 30, Font.PLAIN, 20,
255, 255, 255);
		this.add(usernameLbl);

		stdNumLbl = GUIComp.setLabel("student #: ", 20, 130, 200, 30, Font.PLAIN, 22, 51, 55,
138);
		this.add(stdNumLbl);
```

```
		stdEmailLbl = GUIComp.setLabel("student email: ", 20, 180, 200, 30, Font.PLAIN, 22, 51,
55, 138);
		this.add(stdEmailLbl);

		stdLevelLbl = GUIComp.setLabel("select level: ", 20, 230, 200, 30, Font.PLAIN, 22, 51, 55,
138);
		this.add(stdLevelLbl);

		iaCreationH = GUIComp.setLabel("create your IA here:", 20, 320, 200, 30, Font.PLAIN,
22, 245, 240, 228);
		this.add(iaCreationH);

		// JTextFields/JRadioButtons for the above info
		stdNum = new JTextField();
		stdNum.setBounds(130, 135, 200, 20);
		stdNum.setFont(new Font("Ink Free", Font.PLAIN, 15));
		this.add(stdNum);

		stdEmail = new JTextField();
		stdEmail.setBounds(165, 185, 200, 20);
		stdEmail.setFont(new Font("Ink Free", Font.PLAIN, 15));
		this.add(stdEmail);

		// Select only one radio button at a time
		ButtonGroup level = new ButtonGroup();

		// Customize radio buttons
		sl = GUIComp.setRadioButton("SL", 145, 235, 55, 25, level);
		this.add(sl);

		hl = GUIComp.setRadioButton("HL", 205, 235, 55, 25, level);
		this.add(hl);

		logoutBtn = GUIComp.setButton("logout", 510, 450, 140, 40, 22);
		this.add(logoutBtn);

		logoutBtn.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub
```

```
                // Link to logout page
                ChemPlanApp.cl.show(ChemPlanApp.c, "logoutPage");
        }

});

saveBtn = new JButton("save");
saveBtn.setBounds(35, 450, 80, 40);
saveBtn.setFont(new Font("Ink Free", Font.BOLD, 20));
saveBtn.setForeground(new Color(44, 79, 110));
saveBtn.setBackground(new Color(225, 218, 237));
this.add(saveBtn);

saveBtn.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub

                studentNum = stdNum.getText();
                studentEmail = stdEmail.getText();

                stdNum.setEditable(false);
                stdEmail.setEditable(false);
                sl.setEnabled(false);
                hl.setEnabled(false);

                IAFileInfo = "./" + "IAinfo.txt";

                try {

                        // Add student's basic info to their text file
                        FileWriter fw = new FileWriter(IAFileInfo, true);
                        fw.write("Student: " + userNm + "\n");
                        fw.write("Student #: " + studentNum + "\n");
                        fw.write("Email: " + studentEmail + "\n");

                        if (sl.isSelected()) {
                                fw.write("Level: SL" + "\n");
                        }
                        else if (hl.isSelected()) {
```

```
                        fw.write("Level: HL" + "\n");
                    }
                    else {
                        fw.write("Level: N/A" + "\n");
                    }

                    fw.close();
                }
                catch(IOException iox) {
                    System.out.println("Problem writing " + IAFileInfo);
                }

            }

});

createIABtn = new JButton("create IA");
createIABtn.setBounds(250, 390, 150, 50);
createIABtn.setFont(new Font("Ink Free", Font.BOLD, 23));
createIABtn.setForeground(new Color(44, 79, 110));
createIABtn.setBackground(new Color(218, 235, 209));
this.add(createIABtn);

createIABtn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub

                studentNum = stdNum.getText();
                studentEmail = stdEmail.getText();

                IAFileInfo = "./IAInfo/" + userNm + "IAinfo.txt";

                try {

                    // Add student's basic info to their text file
                    FileWriter fw = new FileWriter(IAFileInfo, true);
                    fw.write("Student: " + userNm + "\n");
                    fw.write("Student #: " + studentNum + "\n");
                    fw.write("Email: " + studentEmail + "\n");
```

```
                    if (sl.isSelected()) {
                        fw.write("Level: SL" + "\n");
                    }
                    else if (hl.isSelected()) {
                        fw.write("Level: HL" + "\n");
                    }
                    else {
                        fw.write("Level: N/A" + "\n");
                    }

                    fw.close();
                }
                catch(IOException iox) {
                    System.out.println("Problem writing " + IAFileInfo);
                }

                // Link to IA page
                IAPanel iaP = new IAPanel(IAFileInfo);
                ChemPlanApp.c.add("IAPage", iaP);
                ChemPlanApp.cl.show(ChemPlanApp.c, "IAPage");
            }

        });

    }

}
```

Appendix K

IAPanel Class Code

```
// Import the following packages:
import java.io.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

/**
 * The following class is for the frame where student's can enter their IA info
 * @author Sri
 *
 */
public class IAPanel extends JPanel {

	private JLabel iaHeader, topicLbl, rsQstnLbl, hypLbl, matLbl, varLbl;
	private JTextField topicF, rsQstnF, hypF, varF;
	private JButton saveInfoBtn, materialsBtn, backToPlannerBtn;
	private String topicInfo, rsQstnInfo, hypInfo, varInfo;

	static String IAFileInfo;

	public IAPanel(String IAInfo) {

		this.setLayout(null);
		this.setBackground(new Color(245, 240, 228));

		iaHeader = GUIComp.setLabel("the IA", 270, 50, 200, 45, Font.BOLD, 35, 44, 79, 110);
		this.add(iaHeader);

		// Add appropriate JLabel and JTextField for each piece of info
		topicLbl = GUIComp.setLabel("topic:", 20, 130, 150, 30, Font.PLAIN, 22, 51, 55, 138);
		this.add(topicLbl);

		topicF = GUIComp.setTextField(80, 125, 360, 40);
		this.add(topicF);

		// ***************

		rsQstnLbl = GUIComp.setLabel("research question:", 20, 180, 270, 30, Font.PLAIN, 22,
51, 55, 138);
```

```
this.add(rsQstnLbl);

rsQstnF = GUIComp.setTextField(200, 175, 360, 40);
this.add(rsQstnF);

// ***************

hypLbl = GUIComp.setLabel("hypothesis:", 20, 230, 220, 30, Font.PLAIN, 22, 51, 55, 138);
this.add(hypLbl);

hypF = GUIComp.setTextField(140, 225, 360, 40);
this.add(hypF);

// ***************

varLbl = GUIComp.setLabel("variables:", 20, 330, 220, 30, Font.PLAIN, 22, 51, 55, 138);
this.add(varLbl);

varF = GUIComp.setTextField(130, 325, 360, 40);
this.add(varF);

// ***************

matLbl = GUIComp.setLabel("materials:", 20, 280, 220, 30, Font.PLAIN, 22, 51, 55, 138);
this.add(matLbl);

materialsBtn = new JButton("add here");
materialsBtn.setBounds(130, 275, 120, 40);
materialsBtn.setFont(new Font("Ink Free", Font.BOLD, 20));
materialsBtn.setForeground(new Color(255, 255, 255));
materialsBtn.setBackground(new Color(167, 178, 194));
this.add(materialsBtn);

materialsBtn.addActionListener(new ActionListener() {

	@Override
	public void actionPerformed(ActionEvent e) {
		// TODO Auto-generated method stub

		// Link to the materials page
		MaterialsPanel matP = new MaterialsPanel(IAFileInfo);
		ChemPlanApp.c.add("materialsPage", matP);
```

```
                ChemPlanApp.cl.show(ChemPlanApp.c, "materialsPage");
            }

        });

        IAFileInfo = IAInfo; // change input source

        // Create a save info button for the IA content
        saveInfoBtn = new JButton("save info");
        saveInfoBtn.setBounds(30, 450, 120, 40);
        saveInfoBtn.setFont(new Font("Ink Free", Font.BOLD, 20));
        saveInfoBtn.setForeground(new Color(44, 79, 110));
        saveInfoBtn.setBackground(new Color(225, 218, 237));
        this.add(saveInfoBtn);

        saveInfoBtn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub

                topicInfo = topicF.getText();
                rsQstnInfo = rsQstnF.getText();
                hypInfo = hypF.getText();
                varInfo = varF.getText();

                try {
                    FileWriter fw = new FileWriter(IAFileInfo, true);
                    fw.write("Topic: " + topicInfo + "\n");
                    fw.write("Research Question: " + rsQstnInfo + "\n");
                    fw.write("Hypothesis: " + hypInfo + "\n");
                    fw.write("Variables: " + varInfo + "\n");
                    fw.write("********************" + "\n"); // break b/w the
different versions

                    fw.close();
                }
                catch(IOException iox) {
                    System.out.println("Problem writing " + IAFileInfo);
                }
            }
```

```
		});

		// Create a back button to the planner page
		backToPlannerBtn = GUIComp.setButton("back to planner", 452, 450, 200, 40, 22);
		this.add(backToPlannerBtn);

		backToPlannerBtn.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub

				ChemPlanApp.cl.show(ChemPlanApp.c, "studentPlannerPage");
			}

		});
	}
}
```

Appendix L

MaterialsPanel Class Code

```
// Import the following packages:
import java.io.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/**
 * The following class is for the inventory frame that appears for students' planning their IA
 * @author Sri
 *
 */
public class MaterialsPanel extends JPanel {

	private JPanel jp;
	private JLabel inventoryH;
	private String inventoryFile = "./inventory.txt", line;
	private ArrayList<String> tools = new ArrayList<String>();
	private ArrayList<JCheckBox> cb = new ArrayList<JCheckBox>();
	private JButton backToIABtn, saveInvBtn;

	public MaterialsPanel(String IAInfo) {

		this.setLayout(null);
		this.setBackground(new Color(119, 158, 186));

		inventoryH = GUIComp.setLabel("inventory", 270, 30, 240, 45, Font.BOLD, 35, 255, 255,
255);
		this.add(inventoryH);

		jp = new JPanel();
		jp.setBackground(new Color(119, 158, 186));
		jp.setLayout(new BoxLayout(jp, BoxLayout.PAGE_AXIS));

		JScrollPane matList = new JScrollPane(jp,
JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED, JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
		matList.setBounds(20, 80, 650, 350);
		matList.setBorder(null);
		this.add(matList);
```

```
String IAFileInfo = IAInfo; // change input source

// Read in the lines from the text file and add it to the array list
try {
	BufferedReader r = new BufferedReader(new FileReader(inventoryFile));
	while ((line = r.readLine()) != null) {
		tools.add(line);
	}

	r.close();
}
catch(IOException iox) {
	System.out.println("Problem reading " + inventoryFile);
}


// Loop through the array list and display the inventory checkboxes on the frame
for (int i = 0; i < tools.size(); i++) {
	JCheckBox materials = new JCheckBox();
	materials = GUIComp.setCheckBox(materials, tools.get(i), 0, 0, 0, 0);

	jp.add(materials);
	cb.add(materials);
}

// Create save inventory button
saveInvBtn = new JButton("save inventory");
saveInvBtn.setBounds(40, 450, 170, 40);
saveInvBtn.setFont(new Font("Ink Free", Font.BOLD, 20));
saveInvBtn.setForeground(new Color(44, 79, 110));
saveInvBtn.setBackground(new Color(225, 218, 237));
this.add(saveInvBtn);

saveInvBtn.addActionListener(new ActionListener() {

	@Override
	public void actionPerformed(ActionEvent e) {
		// TODO Auto-generated method stub

		try {
			FileWriter fw = new FileWriter(IAFileInfo, true);
```

```
                                        fw.write("Materials: " + "\n");

                                        for (int j = 0; j < tools.size(); j++) {
                                                if (cb.get(j).isSelected()) {
                                                        fw.write("- " + cb.get(j).getText() + "\n");
                                                }
                                        }

                                        fw.close();
                                }
                                catch(IOException iox) {
                                        System.out.println("Problem writing " + IAFileInfo);
                                }

                        }

                });


                // Create back to IA button
                backToIABtn = GUIComp.setButton("back to IA", 475, 450, 170, 40, 22);
                this.add(backToIABtn);

                backToIABtn.addActionListener(new ActionListener() {

                        @Override
                        public void actionPerformed(ActionEvent e) {
                                // TODO Auto-generated method stub

                                ChemPlanApp.cl.show(ChemPlanApp.c, "IAPage");
                        }

                });

        }
}
```

Appendix M

TeacherPlannerPanel Class Code

```
// Import the following packages:
import java.io.*;
import java.util.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

/**
 * The following class is for the teacher's planning page
 * @author Sri
 *
 */
public class TeacherPlannerPanel extends JPanel {

	private JLabel planHeader, usernameLbl, welcomeH, instructionH, stdNmLbl, stdNumberLbl,
andLbl, errorLbl;
	private String chkUser, chkNum, line;
	private JTextField stdNameField, stdNumField;
	private JButton viewIABtn, logoutBtn;
	private BufferedReader br;
	private boolean found;

	static String userNm;
	static String IAFileInfo;

	public TeacherPlannerPanel(String username) {

		this.setLayout(null);
		this.setBackground(new Color(119, 158, 186));

		planHeader = GUIComp.setLabel("your planner", 240, 50, 240, 45, Font.BOLD, 35, 255,
255, 255);
		this.add(planHeader);

		userNm = LoginPanel.userInfo.getText();
		usernameLbl = GUIComp.setLabel("hello, " + userNm, 20, 20, 200, 30, Font.PLAIN, 20,
255, 255, 255);
		this.add(usernameLbl);
```

```
		welcomeH = GUIComp.setLabel("welcome, this is your page to review your students' IAs", 70, 130, 520, 30, Font.PLAIN, 22, 51, 55, 138);
		this.add(welcomeH);

		instructionH = GUIComp.setLabel("enter student username and email to view", 165, 180, 330, 20, Font.PLAIN, 18, 245, 240, 228);
		this.add(instructionH);

		// Create JLabels & JTextFields for the following info
		stdNmLbl = GUIComp.setLabel("student name:", 160, 230, 200, 30, Font.PLAIN, 20, 245, 241, 240);
		this.add(stdNmLbl);

		stdNameField = new JTextField();
		stdNameField.setBounds(295, 235, 200, 20);
		stdNameField.setFont(new Font("Ink Free", Font.PLAIN, 15));
		this.add(stdNameField);

		andLbl = GUIComp.setLabel("AND", 320, 275, 50, 30, Font.PLAIN, 21, 51, 55, 138);
		this.add(andLbl);

		stdNumberLbl = GUIComp.setLabel("student number:", 143, 320, 200, 30, Font.PLAIN, 20, 245, 241, 240);
		this.add(stdNumberLbl);

		stdNumField = new JTextField();
		stdNumField.setBounds(295, 325, 200, 20);
		stdNumField.setFont(new Font("Ink Free", Font.PLAIN, 15));
		this.add(stdNumField);

		// Logout JButton
		logoutBtn = GUIComp.setButton("logout", 510, 450, 140, 40, 22);
		this.add(logoutBtn);

		logoutBtn.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub

				// Link to logout page
				ChemPlanApp.cl.show(ChemPlanApp.c, "logoutPage");
```

```
		}

	});

	// Leads to the student's IA page
	viewIABtn = new JButton("view IA");
	viewIABtn.setBounds(260, 380, 140, 45);
	viewIABtn.setFont(new Font("Ink Free", Font.BOLD, 23));
	viewIABtn.setForeground(new Color(51, 55, 138));
	viewIABtn.setBackground(new Color(200, 195, 212));
	this.add(viewIABtn);

	viewIABtn.addActionListener(new ActionListener() {

		@Override
		public void actionPerformed(ActionEvent e) {
			// TODO Auto-generated method stub

			chkUser = stdNameField.getText();
			chkNum = stdNumField.getText();

			IAFileInfo = "./IAInfo/" + chkUser + "IAinfo.txt";

			try {
				br = new BufferedReader(new FileReader(IAFileInfo));

				while ((line = br.readLine()) != null) {
					String check = chkNum;
					if (line.contains(check)) {
						found = true;
					}
				}
			}
			catch(IOException iox) {
				System.out.println("Problem reading " + IAFileInfo);
			}

			if (found == true) {

				// Link to student's IA page
				ReviewIAPanel reviewIAP = new ReviewIAPanel(IAFileInfo,
chkUser);
```

```
                                        ChemPlanApp.c.add("reviewIAPage", reviewIAP);
                                        ChemPlanApp.cl.show(ChemPlanApp.c, "reviewIAPage");
                                    }

                                    else {
                                        errorLbl = GUIComp.setLabel("sorry, wrong user or email, please
try again", 155, 400, 400, 20, Font.PLAIN, 20, 245, 240, 228);
                                        add(errorLbl);
                                    }
                                }

                            });

                    }
}
```

Appendix N

ReviewIAPanel Class Code

```
// Import the following packages:
import java.io.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

/**
 * The following class is for the frame that allows the teacher user to review students' IAs
 * @author Sri
 *
 */
public class ReviewIAPanel extends JPanel {

	private JLabel studentIAH;
	private String expLine;
	private JTextArea expInfoTA;
	private JScrollPane sp;
	private BufferedReader readExp;
	private JButton backToPlanner;

	public ReviewIAPanel(String IAFileInfo, String studentUserName) {

		this.setLayout(null);
		this.setSize(700, 580);
		this.setBackground(new Color(119, 158, 186));

		studentIAH = GUIComp.setLabel(studentUserName + "'s " + "IA", 270, 20, 220, 45,
Font.BOLD, 35, 255, 255, 255);
		this.add(studentIAH);

		String expStr = ""; // initialize variable to add student's IA info

		// Design JTextArea and JScrollPane to have student's IA info
		expInfoTA = new JTextArea(expStr);
		expInfoTA.setBounds(50, 80, 600, 380);
		expInfoTA.setFont(new Font("Ink Free", Font.PLAIN, 18));
		expInfoTA.setForeground(new Color(51, 55, 138));
		expInfoTA.setEditable(false);
		expInfoTA.setLineWrap(true);
```

```
		expInfoTA.setWrapStyleWord(true);

		sp = new JScrollPane(expInfoTA, JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED,
JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
		sp.setBounds(50, 80, 600, 370);
		sp.setBackground(new Color(247, 246, 235));

		// Read in the contents of the student's experiment file to the JTextArea
		try {
			readExp = new BufferedReader(new FileReader(IAFileInfo));
			while ((expLine = readExp.readLine()) != null) {
				expStr += expLine + "\n";
			}

			readExp.close();
		}
		catch(IOException iox) {
			System.out.println("Problem reading " + IAFileInfo);
		}

		expInfoTA.setText(expStr); // add string with appended info to the text area
		this.add(sp);

		// Add back to planner button
		backToPlanner = GUIComp.setButton("back to planner", 452, 460, 200, 40, 22);
		this.add(backToPlanner);

		backToPlanner.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub

				ChemPlanApp.cl.show(ChemPlanApp.c, "teacherPlannerPage");
			}

		});
	}
}
```

Appendix O

WelcomePanel Class

```
// Import the following packages:
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/**
 * The following class is for the welcome frame of the application
 * @author Sri
 *
 */
public class WelcomePanel extends JPanel {

	private JLabel title, subTitle;
	private JButton about, mainMenu;

	public WelcomePanel() {

		this.setLayout(null);
		this.setBackground(new Color(119, 158, 186));

		// Initialize label titles for the welcome page
		String header = "don't bomb it";
		String subHeader = "the chem IA planner you didn't know you needed";

		title = GUIComp.setLabel(header, 135, 100, 600, 80, Font.BOLD, 67, 255, 255, 255);
		this.add(title);

		subTitle = GUIComp.setLabel(subHeader, 50, 200, 630, 40, Font.PLAIN, 28, 51, 55, 138);
		this.add(subTitle);

		about = GUIComp.setButton("about", 150, 270, 145, 45, 30);
		this.add(about);

		about.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub
```

```
                        ChemPlanApp.cl.show(ChemPlanApp.c, "aboutPage");
                }

        });

        mainMenu = GUIComp.setButton("main menu", 325, 270, 190, 45, 30);
        this.add(mainMenu);

        mainMenu.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent e) {
                        // TODO Auto-generated method stub

                        ChemPlanApp.cl.show(ChemPlanApp.c, "appMenuPage");
                }

        });
    }
}
```

Appendix P

GeneralUser Class Code

```
/**
 * The following super class is for a general program user
 * @author Sri
 *
 */
public class GeneralUser {

	private String username;
	private String password;

	public GeneralUser(String username, String password) {
		this.username = username;
		this.password = password;
	}

	public String getUsername() {
		return username;
	}

	public String getPassword() {
		return password;
	}

	public void setUsername(String username) {
		this.username = username;
	}

	public void setPassword(String password) {
		this.password = password;
	}

	public boolean GetIsStudent() {
		return false;
	}
}
```

Appendix Q

StudentUser Class Code

```
/**
 * The following sub-class for a student user inherits the super class GeneralUser
 * @author Sri
 *
 */
public class StudentUser extends GeneralUser {

	public StudentUser(String username, String password) {
		super(username, password);
		// TODO Auto-generated constructor stub
	}

	public boolean GetIsStudent() {
		return true;
	}
}
```

Appendix R

TeacherUser Class Code

```
/**
 * The following sub-class for a teacher user inherits the super class GeneralUser
 * @author Sri
 *
 */
public class TeacherUser extends GeneralUser {

	public TeacherUser(String username, String password) {
		super(username, password);
		// TODO Auto-generated constructor stub
	}

}
```

Appendix S

GUIComp Class Code

```
// Import the following packages:
import java.awt.*;
import javax.swing.*;

/**
 * The following class contains methods for all the main GUI components of the program
 * @author lakhs
 *
 */
public class GUIComp {

	/**
	 * The following method creates JLabels
	 * @param n
	 * @param x
	 * @param y
	 * @param w
	 * @param h
	 * @param fontType
	 * @param fontSize
	 * @param r
	 * @param g
	 * @param b
	 * @return JLabel
	 */
	public static JLabel setLabel(String n, int x, int y, int w, int h, int fontType, int fontSize, int r, int g,
int b) {

		// Set design of JLabels
		JLabel l = new JLabel(n);
		l.setBounds(x, y, w, h);
		l.setFont(new Font("Ink Free", fontType, fontSize));
		l.setForeground(new Color(r, g, b));

		return l; // add JLabel to the frame
	}

	/**
	 * The following method creates JButtons
```

```
 * @param n
 * @param x
 * @param y
 * @param w
 * @param h
 * @param fontSize
 * @param frame
 * @return JButton
 */
public static JButton setButton(String n, int x, int y, int w, int h, int fontSize) {

	// Set design of JButtons
	JButton b = new JButton(n);
	b.setBounds(x, y, w, h);
	b.setFont(new Font("Ink Free", Font.BOLD, fontSize));
	b.setForeground(new Color(44, 79, 110));
	b.setBackground(new Color(186, 210, 232));

	return b;
}

/**
 * The following method creates JButtons with images
 * @param x
 * @param y
 * @param w
 * @param h
 * @param r
 * @param g
 * @param bl
 * @param ic
 * @return JButton
 */
public static JButton setButton(int x, int y, int w, int h, int r, int g, int bl, ImageIcon ic) {

	JButton b = new JButton();
	b.setBounds(x, y, w, h);
	b.setBackground(new Color(r, g, bl));
	b.setIcon(ic);

	return b;
}
```

```
	/**
	 * The following method creates JRadioButtons
	 * @param n
	 * @param x
	 * @param y
	 * @param w
	 * @param h
	 * @param bg
	 * @return JRadioButton
	 */
	public static JRadioButton setRadioButton(String n, int x, int y, int w, int h, ButtonGroup bg) {

		JRadioButton rb = new JRadioButton(n);
		rb.setBounds(x, y, w, h);
		rb.setFont(new Font("Ink Free", Font.PLAIN, 18));
		rb.setBackground(null);
		rb.setForeground(new Color(245, 240, 228));

		bg.add(rb); // Add the radio buttons to the button group
		return rb;
	}

	/**
	 * The following method creates JTextFields for the IA page
	 * @param sp
	 * @param x
	 * @param y
	 * @param w
	 * @param h
	 * @return JTextField
	 */
	public static JTextField setTextField(int x, int y, int w, int h) {

		JTextField tf = new JTextField();
		tf.setFont(new Font("Ink Free", Font.PLAIN, 15));

//		sp = new JScrollPane(tf, JScrollPane.VERTICAL_SCROLLBAR_NEVER,
JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
		tf.setBounds(x, y, w, h);

		return tf;
```

```
	}

	public static JCheckBox setCheckBox(JCheckBox cb, String n, int x, int y, int w, int h) {
		cb = new JCheckBox(n);
		cb.setBounds(x, y, w, h);
		cb.setFont(new Font("Ink Free", Font.PLAIN, 19));
		cb.setBackground(null);

//		for (int i = 0; i < s.size(); i++) {
//			s.add(i, s.get(i));
//		}

		return cb;
	}
}
```