

## **APPENDIX**

### **1. Consultation with the client at planning stage**

Me: Hi A! I heard that you are having trouble managing students' seating plan again.

Client A: Yeah, a bit unfortunate.

Me: Are you still using my code written in Grade 10? I was still a newbie back then.

Client A: Yep, but it seems that your code has no maintenance so the result was frustrating. It does not seem "random", I mean students just keep seating with each other and they always talk to each other. How can their grades improve if they keep on being distracted from the class?

Me: I know, indeed. So I decide to help you by designing a brand new seating assignment program. New features, user-friendly design, and good portability! I still remember that when running my previous code on a Windows computer, sometimes the program runs infinitely and displaying chaotic characters.

Client A: Wow, really? That's so helpful! Which programming language will you use then?

Me: Probably java. I am familiar with it, and I have just learned how to write the GUI. Clients without programming experience cannot stand the command-line interface, right?

Client A: Yeah, the moments using your old code suffers.

Me: So, what kind of features would you want?

Client A: Keep the random seating with the new version please. Also, I want to directly edit the seating. I just want to keep away some annoying kids :)

Me: OK. I will try to add a feature that prevents certain students from seating next to each other.

Client A: Cool. Also, it is a bit tricky but can you manage to design a program that stores students' exam scores on each subject, and try to assign students with good grades with those that have more space of improvement? It helps the students as a whole a lot.

Me: I will try on that. I can probably write that out. Any other suggestions?

Client A: I will tell you any additional demands when I think of them. Good luck then!

Me: By the way, thank you for your support. It is truly a great experience to help my teacher. I will let you know the basic design of my product in a few weeks.

Client A: Thank you so much! Bye bye!

## 2. My java codes

Account class

```
/*
 * 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.
 */
package seatingassignment;

/**
 *
 * @author Student-3
 */
public class Account {
    private String username;
    private String password;

    /**
     * Default constructor
     * Nothing happened
     */
    public Account() {
        //in 1989 June 4th
        //nothing happened
    }

    /**
     * Constructor
     * @param _username username of the account
     * @param _password password of the account
     */
    public Account(String _username, String _password) {
        username=_username;
        password=_password;
    }

    public Account(String _username, char[] _password) {
        username=_username;
        password=String.valueOf(_password);
    }

    /**
     * Accessor
```

```

    * Get the value of the username
    * @return username
    */
    public String getUsername() {
        return username;
    }

    /**
     * Mutator
     * Set the value of the username to _username
     * @param _username username value assigned
     */
    public void setUsername(String _username) {
        username=_username;
    }

    /**
     * Accessor
     * Get the value of the password
     * @return password
     */
    public String getPassword() {
        return password;
    }

    /**
     * Mutator
     * Set the value of the password to _password
     * @param _password password value assigned
     */
    public void setPassword(String _password) {
        password=_password;
    }

    /**
     * Overrides the toString method, return the String format of the Account
     * @return String format of the Account
     */
    @Override
    public String toString() {
        return username+","+password;
    }

    /**
     * Return the String of the Account in csv format
     * @return String of the Account in csv format

```

```

    */
    public String toCSV(){
        return String.format("%s,%s", username, password);
    }
}

```

AccountManager class

```

/*
 * 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.
 */
package seatingassignment;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Student-3
 */
public class AccountManager {
    public static final String SEPARATOR = ",";
    public static ArrayList<Account> accounts = new ArrayList<Account>();
    public static Account currentAccount;

    /**
     * Add a new account to accounts
     * @param _username username of the account
     * @param _password password of the account
     */
    public static void addAccount(String _username, String _password) {
        Account _acc=new Account(_username,_password);
        accounts.add(_acc);
    }
}

```

```

/**
 * Add a new account to accounts
 * @param _acc the account
 */
public static void addAccount(Account _acc) {
    accounts.add(_acc);
}

/**
 * The user logs in to the system
 * @return user's account
 * @throws FileNotFoundException
 * @throws Exception
 */
public static Account initAccount() throws FileNotFoundException, Exception {
    accounts=new ArrayList<Account>();

    //load every previous accounts
    Scanner terminalScanner=new Scanner(System.in);

    //file input
    Scanner csvScanner = new Scanner(new File("Accounts.csv"));
    while(csvScanner.hasNext()) {
        String[] nextUserInfo=csvScanner.nextLine().split(SEPARATOR);
        AccountManager.addAccount(nextUserInfo[0], nextUserInfo[1]);
    }
    csvScanner.close();

    int opt;
    String myUsername,myPassword;
    System.out.println("Input 1 to create a new account; input 2 to log in to the system; input 3
to delete an account.");
    opt=terminalScanner.nextInt();
    if(opt==1) { //create a new account

        while(true) {
            System.out.print("Create a new account:\nInput username: ");
            myUsername=terminalScanner.next();
            System.out.print("Input password: ");
            myPassword=terminalScanner.next();

            int ok=1;
            for(Account acc:accounts) {
                if(acc.getUsername().equals(myUsername)) {
                    System.out.println("Account already exist! Create your account again.");
                    ok=0;
                }
            }
            if(ok==0) continue;
        }
    }
}

```

```

        break;
    }
}
if(ok==0) continue;
System.out.println("Account successssfully created!");
Account myAcc=new Account(myUsername,myPassword);
AccountManager.addAccount(myAcc);

try {
    //add the new account to Accounts.csv
    BufferedWriter csvWriter = new BufferedWriter(new FileWriter(new
File("Accounts.csv")));
    for(Account _acc:accounts) {
        csvWriter.write(_acc.toCSV() + "\n");
        csvWriter.flush();
    }
    csvWriter.close();
    //add a new directory containing the Accounts information
    File f=new File("Accounts/"+myUsername);
    f.mkdir();

} catch (IOException ex) {
    Logger.getLogger(AccountManager.class.getName()).log(Level.SEVERE, null, ex);
}
currentAccount=myAcc;

try {
    //input student namelist.csv
    BufferedWriter csvWriter = new BufferedWriter(new FileWriter(new
File("Accounts/"+myUsername+"/namelist.csv")));
    csvWriter.write("<Input student namelist here. One student per line.>\n");
    csvWriter.flush();
    csvWriter.close();
} catch (IOException ex) {
    Logger.getLogger(AccountManager.class.getName()).log(Level.SEVERE, null, ex);
}
System.out.println("Import student list at "+"Accounts/"+myUsername+".csv");

return myAcc;
}
}
else if(opt==2) { //login
    if(accounts.isEmpty()) {
        throw new Exception("No registered users! How can you possibly login!");
    }
}

```

```

while(true) {
    System.out.println("Please log in to the system: ");
    System.out.print("Input username: ");
    myUsername=terminalScanner.next();
    System.out.print("Input password: ");
    myPassword=terminalScanner.next();

    for(Account acc:accounts) {
        if(acc.getUsername().equals(myUsername) &&
acc.getPassword().equals(myPassword)) {
            System.out.println("Login Successful!");
            currentAccount=acc;
            return acc;
        }
    }
    System.out.println("Username and password don't match. Input again.");
}
}
else if(opt==3) {//delete an existing account
    System.out.println("Displaying all the accounts stored: ");
    for(Account acc:accounts)
        System.out.println(acc);
    System.out.println("Which account do you want to delete? Input the account name:");
    while(true) {
        int ok=0;
        String _userName=terminalScanner.next();
        for(int i=0;i<accounts.size();i++) {
            if(accounts.get(i).getUsername().equals(_userName)) {
                System.out.println("Account "+accounts.get(i).getUsername()+" is canceled");
                accounts.remove(i);

                try {
                    BufferedWriter csvWriter = new BufferedWriter(new FileWriter(new
File("Accounts.csv")));
                    for(Account _acc:accounts) {
                        csvWriter.write(_acc.toCSV() + "\n");
                        csvWriter.flush();
                    }
                    csvWriter.close();
                } catch (IOException ex) {
                    Logger.getLogger(Account.class.getName()).log(Level.SEVERE, null, ex);
                }

                String pathName="Accounts/"+_userName+"/";
                System.out.println("The path is: "+pathName);
            }
        }
    }
}

```

```

        File folder = new File(pathName);
        File[] userFiles = folder.listFiles();
        for(File file : userFiles) {
            file.delete();
        }

        folder.delete();
        System.out.println(folder.exists());

        ok=1;
        break;
    }
}
if(ok==1) break;
else System.out.println("Account not found! Input again!");
}
return null;
}
else {
    throw new Exception("Invalid option!");
}
}
}

```

## Layout class

```

/*
 * 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.
 */
package seatingassignment;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

/**

```

```

*
* @author Student-3
*/
public class Layout {

    private String name;
    public static int ROWS;
    public static int COLUMNS;
    public static ArrayList<Pair> posCanceled;
    public static int numCanceled;
    public static int numSeats;
    public static boolean[][] map;
    // public static ArrayList<Integer> emptyColumns; // not implemented

    /**
     * Default constructor
     * Evokes the initi method
     * @throws Exception
     */
    public Layout() throws Exception {
        initi();
    }

    /**
     * Constructor
     * Construct a Layout object
     * @param _name name of the layout
     * @param _rows number of rows
     * @param _columns number of columns
     * @param _numCanceled number of positions to be canceled
     * @param _posCanceled positions canceled in the layout
     */
    public Layout(String _name, int _rows, int _columns, int _numCanceled, ArrayList<Pair>
_posCanceled) {
        name=_name;
        ROWS=_rows;
        COLUMNS=_columns;
        posCanceled=_posCanceled;
        numCanceled=_posCanceled.size();
        numSeats=ROWS*COLUMNS-numCanceled;
        map=new boolean[ROWS+1][COLUMNS+1];

        for(int i=1;i<=ROWS;i++)
            for(int j=1;j<=COLUMNS;j++) {
                map[i][j]=true;
            }
    }

```

```

        for(Pair pr:_posCanceled) {
            map[pr.getX()][pr.getY()]=false;
        }
    }

/**
 * Constructor
 * Construct a Layout object by reading from a file
 * @param _file the Layout file
 * @throws FileNotFoundException
 */
public Layout(File _file) throws FileNotFoundException {
    System.out.println("The file name in \"Layout\" is "+_file.getName());
    Scanner scn=new Scanner(System.in);
    Scanner fileReader=new Scanner(_file);
    String fileName=_file.getName();
    name=fileName.substring(0,fileName.length()-4);
    //System.out.println("In Layout: name = "+name);
    ROWS=Integer.parseInt(fileReader.next());
    //System.out.println("In Layout: ROWS = "+ROWS);
    COLUMNS=Integer.parseInt(fileReader.next());
    //System.out.println("In Layout: COLUMNS = "+COLUMNS);
    numCanceled=Integer.parseInt(fileReader.next());
    //System.out.println("In Layout: numCanceled = "+numCanceled);
    numSeats=ROWS*COLUMNS-numCanceled;
    posCanceled=new ArrayList<Pair>();
    for(int i=0;i<numCanceled;i++) {
        String str=fileReader.next();
        int pos=str.indexOf(",");
        int _x=Integer.parseInt(str.substring(1,pos));
        int _y=Integer.parseInt(str.substring(pos+1,str.length()-1));
        Pair newPair=new Pair(_x,_y);
        posCanceled.add(newPair);
    }
    map=new boolean[ROWS+1][COLUMNS+1];
    for(int i=1;i<=ROWS;i++)
        for(int j=1;j<=COLUMNS;j++) {
            map[i][j]=true;
        }
    for(Pair pr:posCanceled) {
        map[pr.getX()][pr.getY()]=false;
    }

    scn.close();
    fileReader.close();
}

```

```
/**
 * Accessor
 * Get the name of the layout
 * @return the name of the layout
 */
```

```
public String getName() {
    return name;
}
```

```
/**
 * Mutator
 * Set the name to _name
 * @param _name name value assigned
 */
```

```
public void setName(String _name) {
    name=_name;
}
```

```
/**
 * Accessor
 * Get the row number of the layout
 * @return row number of the layout
 */
```

```
public int getRows() {
    return ROWS;
}
```

```
/**
 * Mutator
 * Set the row number of the layout to _rows
 * @param _rows row number assigned
 */
```

```
public void setRows(int _rows) {
    ROWS=_rows;
}
```

```
/**
 * Accessor
 * Get the column number of the layout
 * @return column number of the layout
 */
```

```
public int getColumns() {
    return COLUMNS;
}
```

```

/**
 * Mutator
 * Set the column number of the layout to _columns
 * @param _columns column number assigned
 */
public void setColumns(int _columns) {
    COLUMNS=_columns;
}

/**
 * Accessor
 * Get the number of seats in the layout
 * @return number of seats
 */
public int getNumberOfSeats() {
    return numSeats;
}

/**
 * Mutator
 * Set the number of seats in the layout to _numSeats
 * @param _numSeats number of seats assigned
 */
public void setNumberOfSeats(int _numSeats) {
    numSeats=_numSeats;
}

/**
 * Accessor
 * Get the positions of those canceled blocks
 * @return ArrayList of canceled positions
 */
public ArrayList<Pair> getPosCanceled() {
    return posCanceled;
}

/**
 * Mutator
 * Set the canceled positions to _posCanceled
 * @param _posCanceled canceled positions assigned
 */
public void setPosCanceled(ArrayList<Pair> _posCanceled) {
    posCanceled=_posCanceled;
}

/**

```

```

    * Accessor
    * Get the map display of the current layout
    * @return map display of the layout
    */
    public boolean[][] getMap() {
        return map;
    }

    /**
     * Initialize a layout by creating an Layout object from user input
     * @throws Exception
     */
    public void initi() throws Exception {
        Scanner scn=new Scanner(System.in);
        System.out.println("init");
        System.out.println("Input the name of this new layout:");
        String _name=scn.next();
        System.out.println("Input rows:");
        int row=scn.nextInt();
        System.out.println("Input columns:");
        int col=scn.nextInt();
        System.out.println("Input number of blocks to cancel:");
        int cancelNum=scn.nextInt();
        System.out.println("You have selected "+cancelNum+" blocks to cancel. Input their
coordinates in (<row>,<column>) format respectively.");
        ArrayList<Pair> _posCanceled=new ArrayList<Pair>();

        for(int i=1;i<=cancelNum;i++) {
            System.out.println("Number "+i+": ");
            int _x=scn.nextInt(), _y=scn.nextInt();
            _posCanceled.add(new Pair(_x,_y));
        }

        Layout myLayout=new Layout(_name,row,col,cancelNum,_posCanceled);
        System.out.println("Display the current map:");
        myLayout.displayMap();

        System.out.println("Are you satisfied with this layout? Press 1 if you like and the layout
will be saved; press 0 to clear this entry to start a new layout. ");
        int opt=scn.nextInt();
        if(opt==1) {
            try {
                //write the configuration into a txt file
                String pathName = "Accounts/" + AccountManager.currentAccount.getUsername() +
"/" + myLayout.name + ".txt";
                File file1 = new File(pathName);

```

```

        file1.createNewFile();
        BufferedWriter writer = new BufferedWriter(new FileWriter(pathName));
        //writer.write(myLayout.name + "\n");
        writer.write(myLayout.ROWS + "\n");
        writer.write(myLayout.COLUMNS + "\n");
        writer.write(myLayout.numCanceled + "\n");
        for(Pair pr:myLayout.posCanceled)
            writer.write(pr+"\n");
        writer.close();
    } catch (IOException ex) {
        Logger.getLogger(Layout.class.getName()).log(Level.SEVERE, null, ex);
    }
    System.out.println("Your layout named "+myLayout.name+" is successfully saved.");
}
else if(opt==0) {
    //restart the initialization process
    initi();
}
else {
    throw new Exception("Invalid option!");
}
}

/**
 * Display the current map layout
 */
public void displayMap() {
    for(int i=1;i<=ROWS;i++) {
        for(int j=1;j<=COLUMNS;j++) {
            if(map[i][j]==true) {
                System.out.print("_____ ");
            }
            else {
                System.out.print("xxxxxxx ");
            }
        }
        System.out.println();
    }
}
}
}

```

GUI class

package seatingassignment;

```

import java.awt.Component;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.text.JTextComponent;
import static seatingassignment.AccountManager.SEPARATOR;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author Student-3
 */
public class GUI extends javax.swing.JFrame {

    private ArrayList<Account> accounts;
    private Account currentAccount;

    /**
     * Creates new form GUI
     */
    public GUI() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")

```

```
// <editor-fold defaultstate="collapsed" desc="Generated Code">  
private void initComponents() {
```

```
    createNewAccountFrame = new javax.swing.JFrame();  
    createNewAccountLabel = new javax.swing.JLabel();  
    usernameNewAccountLabel = new javax.swing.JLabel();  
    passwordNewAccountLabel = new javax.swing.JLabel();  
    usernameNewAccountTextField = new javax.swing.JTextField();  
    passwordNewAccountPasswordField = new javax.swing.JPasswordField();  
    createNewAccountButton = new javax.swing.JButton();  
    loginFrame = new javax.swing.JFrame();  
    loginLabel = new javax.swing.JLabel();  
    usernameLoginLabel = new javax.swing.JLabel();  
    passwordLoginLabel = new javax.swing.JLabel();  
    usernameLoginTextField = new javax.swing.JTextField();  
    passwordLoginPasswordField = new javax.swing.JPasswordField();  
    confirmLoginButton = new javax.swing.JButton();  
    deleteAccountFrame = new javax.swing.JFrame();  
    deleteAccountLabel = new javax.swing.JLabel();  
    deleteAccountScrollPane = new javax.swing.JScrollPane();  
    deleteAccountTextArea = new javax.swing.JTextArea();  
    usernameDeleteAccountTextField = new javax.swing.JTextField();  
    usernameDeleteAccountLabel = new javax.swing.JLabel();  
    passwordDeleteAccountLabel = new javax.swing.JLabel();  
    passwordDeleteAccountPasswordField = new javax.swing.JPasswordField();  
    confirmDeleteButton = new javax.swing.JButton();  
    createNewLayoutFrame = new javax.swing.JFrame();  
    jLabel1 = new javax.swing.JLabel();  
    jLabel2 = new javax.swing.JLabel();  
    jLabel3 = new javax.swing.JLabel();  
    jLabel4 = new javax.swing.JLabel();  
    jLabel5 = new javax.swing.JLabel();  
    jLabel6 = new javax.swing.JLabel();  
    jTextField1 = new javax.swing.JTextField();  
    jTextField2 = new javax.swing.JTextField();  
    jTextField3 = new javax.swing.JTextField();  
    jTextField4 = new javax.swing.JTextField();  
    jTextField5 = new javax.swing.JTextField();  
    jButton1 = new javax.swing.JButton();  
    startAssigningSeatsFrame = new javax.swing.JFrame();  
    jLabel8 = new javax.swing.JLabel();  
    jScrollPane2 = new javax.swing.JScrollPane();  
    jTextArea2 = new javax.swing.JTextArea();  
    jFrame3 = new javax.swing.JFrame();  
    displayFrame = new javax.swing.JFrame();  
    jLabel7 = new javax.swing.JLabel();
```

```

jButton2 = new javax.swing.JButton();
jButton3 = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
jTextArea1 = new javax.swing.JTextArea();
seatingAssignmentLabel = new javax.swing.JLabel();
scrollPane = new javax.swing.JScrollPane();
textArea = new javax.swing.JTextArea();
mainMenuBar = new javax.swing.JMenuBar();
loginMenu = new javax.swing.JMenu();
createNewAccountMenuItem = new javax.swing.JMenuItem();
loginMenuItem = new javax.swing.JMenuItem();
deleteAccountMenuItem = new javax.swing.JMenuItem();
optionsMenu = new javax.swing.JMenu();
createNewLayoutMenuItem = new javax.swing.JMenuItem();
startAssigningSeatsMenuItem = new javax.swing.JMenuItem();
checkNewFeaturesMenuItem = new javax.swing.JMenuItem();

createNewLabel.setFont(createNewLabel.getFont().deriveFont(createNewAccountLabel.getFont().getStyle() | java.awt.Font.BOLD, 18));
createNewLabel.setText("Create New Account");

usernameNewLabel.setText("Username");

passwordNewLabel.setText("Password");

usernameNewAccountTextField.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        usernameNewAccountTextFieldActionPerformed(evt);
    }
});

passwordNewAccountPasswordField.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        passwordNewAccountPasswordFieldActionPerformed(evt);
    }
});

createNewAccountButton.setText("Create");
createNewAccountButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        createNewAccountButtonActionPerformed(evt);
    }
});

```

```

        javax.swing.GroupLayout createNewAccountFrameLayout = new
javax.swing.GroupLayout(createNewAccountFrame.getContentPane());
        createNewAccountFrame.getContentPane().setLayout(createNewAccountFrameLayout);
        createNewAccountFrameLayout.setHorizontalGroup(

createNewAccountFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addGroup(createNewAccountFrameLayout.createSequentialGroup())
        .addGroup(createNewAccountFrameLayout.createParallelGroup(javax.swing.GroupLa
yout.Alignment.LEADING)
        .addGroup(createNewAccountFrameLayout.createSequentialGroup())
        .addGroup(100, 100, 100)
        .addGroup(createNewAccountFrameLayout.createParallelGroup(javax.swing.Gro
upLayout.Alignment.LEADING)
        .addComponent(usernameNewAccountLabel)
        .addComponent(passwordNewAccountLabel))
        .addGroup(18, 18, 18)
        .addGroup(createNewAccountFrameLayout.createParallelGroup(javax.swing.Gro
upLayout.Alignment.LEADING, false)
        .addComponent(usernameNewAccountTextField)
        .addComponent(passwordNewAccountPasswordField,
javax.swing.GroupLayout.DEFAULT_SIZE, 167, Short.MAX_VALUE)
        .addGroup(createNewAccountFrameLayout.createSequentialGroup())
        .addGroup(32, 32, 32)
        .addComponent(createNewAccountButton))))
        .addGroup(createNewAccountFrameLayout.createSequentialGroup())
        .addGroup(151, 151, 151)
        .addComponent(createNewAccountLabel,
javax.swing.GroupLayout.PREFERRED_SIZE, 214,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(207, Short.MAX_VALUE))
    );
    createNewAccountFrameLayout.setVerticalGroup(

createNewAccountFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addGroup(createNewAccountFrameLayout.createSequentialGroup())
        .addGroup(44, 44, 44)
        .addComponent(createNewAccountLabel,
javax.swing.GroupLayout.PREFERRED_SIZE, 24,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(91, 91, 91)
        .addGroup(createNewAccountFrameLayout.createParallelGroup(javax.swing.GroupLa
yout.Alignment.BASELINE)
        .addComponent(usernameNewAccountLabel)

```

```

        .addComponent(usernameNewAccountTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(16, 16, 16)
        .addGroup(createNewAccountFrameLayout.createParallelGroup(javax.swing.GroupLa
yout.Alignment.BASELINE)
        .addComponent(passwordNewAccountLabel)
        .addComponent(passwordNewAccountPasswordField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(53, 53, 53)
        .addComponent(createNewAccountButton)
        .addContainerGap(149, Short.MAX_VALUE))
    );

    loginLabel.setFont(new java.awt.Font("Lucida Grande", 1, 18)); // NOI18N
    loginLabel.setText("Login");

    usernameLoginLabel.setText("Username");

    passwordLoginLabel.setText("Password");

    passwordLoginPasswordField.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            passwordLoginPasswordFieldActionPerformed(evt);
        }
    });

    confirmLoginButton.setText("Confirm");
    confirmLoginButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            confirmLoginButtonActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout loginFrameLayout = new
javax.swing.GroupLayout(loginFrame.getContentPane());
    loginFrame.getContentPane().setLayout(loginFrameLayout);
    loginFrameLayout.setHorizontalGroup(

loginFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(loginFrameLayout.createSequentialGroup()
            .addGap(104, 104, 104)
            .addGroup(loginFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
                .addComponent(usernameLoginLabel)

```

```

        .addComponent(passwordLoginLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 34,
Short.MAX_VALUE)
        .addGroup(loginFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
        .addComponent(confirmLoginButton)
        .addGroup(loginFrameLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)
        .addComponent(usernameLoginTextField)
        .addComponent(passwordLoginPasswordField,
javax.swing.GroupLayout.DEFAULT_SIZE, 163, Short.MAX_VALUE)))
        .addGap(225, 225, 225))
        .addGroup(loginFrameLayout.createSequentialGroup())
        .addGap(241, 241, 241)
        .addComponent(loginLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    loginFrameLayout.setVerticalGroup(

loginFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(loginFrameLayout.createSequentialGroup())
        .addGap(66, 66, 66)
        .addComponent(loginLabel)
        .addGap(65, 65, 65)
        .addGroup(loginFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.BASELINE)
        .addComponent(usernameLoginLabel)
        .addComponent(usernameLoginTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(17, 17, 17)
        .addGroup(loginFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.BASELINE)
        .addComponent(passwordLoginLabel)
        .addComponent(passwordLoginPasswordField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(71, 71, 71)
        .addComponent(confirmLoginButton)
        .addContainerGap(139, Short.MAX_VALUE))
    );

deleteAccountLabel.setFont(new java.awt.Font("Lucida Grande", 1, 18)); // NOI18N
deleteAccountLabel.setText("Delete an Account");

```

```

deleteAccountTextArea.setColumns(20);
deleteAccountTextArea.setLineWrap(true);
deleteAccountTextArea.setRows(5);
deleteAccountTextArea.setText("Displaying all the accounts stored: \n\ntest, tester,
Wennie, frog, clientA\n\nWhich account do you want to delete? Input the account name and
password below.");
deleteAccountTextArea.setWrapStyleWord(true);
deleteAccountScrollPane.setViewportView(deleteAccountTextArea);

usernameDeleteAccountLabel.setText("Account");

passwordDeleteAccountLabel.setText("Password");

confirmDeleteButton.setText("Confirm");

javax.swing.GroupLayout deleteAccountFrameLayout = new
javax.swing.GroupLayout(deleteAccountFrame.getContentPane());
deleteAccountFrame.getContentPane().setLayout(deleteAccountFrameLayout);
deleteAccountFrameLayout.setHorizontalGroup(
deleteAccountFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(deleteAccountFrameLayout.createSequentialGroup()
        .addGroup(deleteAccountFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(deleteAccountFrameLayout.createSequentialGroup()
                .add(deleteAccountLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE)
                .add(deleteAccountScrollPane, javax.swing.GroupLayout.PREFERRED_SIZE, 200, javax.swing.GroupLayout.PREFERRED_SIZE)
                .add(confirmDeleteButton, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE)
            )
            .addGroup(deleteAccountFrameLayout.createSequentialGroup()
                .add(usernameDeleteAccountLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE)
                .add(passwordDeleteAccountLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 100, javax.swing.GroupLayout.PREFERRED_SIZE)
            )
        )
        .addContainerGap(100, false)
    )
);

```

```

        .addGroup(deleteAccountFrameLayout.createParallelGroup(javax.swing.
 GroupLayout.Alignment.LEADING, false)
        .addComponent(usernameDeleteAccountTextField)
        .addComponent(passwordDeleteAccountPasswordField,
 javax.swing.GroupLayout.DEFAULT_SIZE, 183, Short.MAX_VALUE))))))
        .addContainerGap(182, Short.MAX_VALUE))
    );
    deleteAccountFrameLayout.setVerticalGroup(

deleteAccountFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
G)
        .addGroup(deleteAccountFrameLayout.createSequentialGroup())
        .addGap(61, 61, 61)
        .addComponent(deleteAccountLabel)
        .addGap(52, 52, 52)
        .addComponent(deleteAccountScrollPane,
 javax.swing.GroupLayout.PREFERRED_SIZE, 141,
 javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(33, 33, 33)
        .addGroup(deleteAccountFrameLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
        .addComponent(usernameDeleteAccountTextField,
 javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
 javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(usernameDeleteAccountLabel))
        .addGap(18, 18, 18)
        .addGroup(deleteAccountFrameLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
        .addComponent(passwordDeleteAccountLabel)
        .addComponent(passwordDeleteAccountPasswordField,
 javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
 javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addComponent(confirmDeleteButton)
        .addContainerGap(57, Short.MAX_VALUE))
    );

jLabel1.setFont(new java.awt.Font("Lucida Grande", 1, 18)); // NOI18N
jLabel1.setText("Create a New Layout");

jLabel2.setText("Name");

jLabel3.setText("Number of rows");

jLabel4.setText("Number of columns");

```

```

jLabel5.setText("Number of blocks canceled");

jLabel6.setText("Coordinates of blocks canceled");

jTextField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1ActionPerformed(evt);
    }
});

jButton1.setText("Create");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

javax.swing.GroupLayout createNewLayoutFrameLayout = new
javax.swing.GroupLayout(createNewLayoutFrame.getContentPane());
createNewLayoutFrame.getContentPane().setLayout(createNewLayoutFrameLayout);
createNewLayoutFrameLayout.setHorizontalGroup(

createNewLayoutFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(createNewLayoutFrameLayout.createSequentialGroup()
        .addGroup(createNewLayoutFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(createNewLayoutFrameLayout.createSequentialGroup()
                .addGap(154, 154, 154)
                .addComponent(jLabel1))
            .addGroup(createNewLayoutFrameLayout.createSequentialGroup()
                .addGap(75, 75, 75)
                .addGroup(createNewLayoutFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(createNewLayoutFrameLayout.createSequentialGroup()
                        .addComponent(jLabel6)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jTextField5))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
createNewLayoutFrameLayout.createSequentialGroup()
                        .addGroup(createNewLayoutFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel3)
                            .addComponent(jLabel2)
                            .addComponent(jLabel4)

```

```

        .addComponent(jLabel5))
        .addGap(39, 39, 39)
        .addGroup(createNewLayoutFrameLayout.createParallelGroup(javax.swing.
GroupLayout.Alignment.LEADING)
        .addComponent(jTextField1, javax.swing.GroupLayout.DEFAULT_SIZE,
155, Short.MAX_VALUE)
        .addComponent(jTextField2)
        .addComponent(jTextField4)
        .addComponent(jTextField3))))))
        .addGap(88, 88, 88))
        .addGroup(createNewLayoutFrameLayout.createSequentialGroup())
        .addGap(200, 200, 200)
        .addComponent(jButton1)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    createNewLayoutFrameLayout.setVerticalGroup(

createNewLayoutFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
        .addGroup(createNewLayoutFrameLayout.createSequentialGroup())
        .addGap(40, 40, 40)
        .addComponent(jLabel1)
        .addGap(61, 61, 61)
        .addGroup(createNewLayoutFrameLayout.createParallelGroup(javax.swing.GroupLay
out.Alignment.BASELINE)
        .addComponent(jLabel2)
        .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(createNewLayoutFrameLayout.createParallelGroup(javax.swing.GroupLay
out.Alignment.BASELINE)
        .addComponent(jLabel3)
        .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(createNewLayoutFrameLayout.createParallelGroup(javax.swing.GroupLay
out.Alignment.BASELINE)
        .addComponent(jLabel4)
        .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(createNewLayoutFrameLayout.createParallelGroup(javax.swing.GroupLay
out.Alignment.BASELINE)
        .addComponent(jLabel5)

```

```

        .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(createNewLayoutFrameLayout.createParallelGroup(javax.swing.GroupLay
out.Alignment.BASELINE)
        .addComponent(jLabel6)
        .addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(28, 28, 28)
        .addComponent(jButton1)
        .addContainerGap(36, Short.MAX_VALUE))
    );

    jLabel8.setFont(new java.awt.Font("Lucida Grande", 1, 18)); // NOI18N
    jLabel8.setText("Start Assigning Seats");

    jTextArea2.setColumns(20);
    jTextArea2.setRows(5);
    jScrollPane2.setViewportView(jTextArea2);

    javax.swing.GroupLayout startAssigningSeatsFrameLayout = new
javax.swing.GroupLayout(startAssigningSeatsFrame.getContentPane());
    startAssigningSeatsFrame.getContentPane().setLayout(startAssigningSeatsFrameLayout);
    startAssigningSeatsFrameLayout.setHorizontalGroup(

startAssigningSeatsFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
        .addGroup(startAssigningSeatsFrameLayout.createSequentialGroup())
        .addGroup(startAssigningSeatsFrameLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.LEADING)
        .addGroup(startAssigningSeatsFrameLayout.createSequentialGroup())
        .addGap(204, 204, 204)
        .addComponent(jLabel8))
        .addGroup(startAssigningSeatsFrameLayout.createSequentialGroup())
        .addGap(119, 119, 119)
        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
372, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(168, Short.MAX_VALUE))
    );
    startAssigningSeatsFrameLayout.setVerticalGroup(

startAssigningSeatsFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
        .addGroup(startAssigningSeatsFrameLayout.createSequentialGroup())
        .addGap(55, 55, 55)
        .addComponent(jLabel8)

```

```

        .addGap(34, 34, 34)
        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 245,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(136, Short.MAX_VALUE))
    );

    javax.swing.GroupLayout jFrame3Layout = new
javax.swing.GroupLayout(jFrame3.getContentPane());
    jFrame3.getContentPane().setLayout(jFrame3Layout);
    jFrame3Layout.setHorizontalGroup(
        jFrame3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 400, Short.MAX_VALUE)
    );
    jFrame3Layout.setVerticalGroup(
        jFrame3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 300, Short.MAX_VALUE)
    );

    jLabel7.setFont(new java.awt.Font("Lucida Grande", 1, 18)); // NOI18N
    jLabel7.setText("Map Display");

    jButton2.setText("Confirm");

    jButton3.setText("Cancel");

    jTextArea1.setColumns(20);
    jTextArea1.setRows(5);
    jTextArea1.setText("_____
_____
xxxxxxx \n
_____
\n
_____
\n
_____
xxxxxxxx xxxxxxxx ");
    jScrollPane1.setViewportView(jTextArea1);

    javax.swing.GroupLayout displayFrameLayout = new
javax.swing.GroupLayout(displayFrame.getContentPane());
    displayFrame.getContentPane().setLayout(displayFrameLayout);
    displayFrameLayout.setHorizontalGroup(

displayFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(displayFrameLayout.createSequentialGroup()
        .addGroup(displayFrameLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
            .addGroup(displayFrameLayout.createSequentialGroup()
                .addGroup(displayFrameLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gment.LEADING)
                    .addGroup(displayFrameLayout.createSequentialGroup()
                        .addGroup(displayFrameLayout.createSequentialGroup()
                            .addGap(142, 142, 142)
                            .addComponent(jButton2)
                            .addGap(115, 115, 115)

```

```

        .addComponent(jButton3))
    .addGroup(displayFrameLayout.createSequentialGroup()
        .addGap(268, 268, 268)
        .addComponent(jLabel7))
    .addGroup(displayFrameLayout.createSequentialGroup()
        .addGap(132, 132, 132)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
515, javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap(145, Short.MAX_VALUE))
);
displayFrameLayout.setVerticalGroup(

displayFrameLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(displayFrameLayout.createSequentialGroup()
        .addGap(48, 48, 48)
        .addComponent(jLabel7)
        .addGap(90, 90, 90)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 215,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 91,
Short.MAX_VALUE)
        .addGroup(displayFrameLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
            .addComponent(jButton2)
            .addComponent(jButton3))
        .addGap(61, 61, 61))
    );

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

seatingAssignmentLabel.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
seatingAssignmentLabel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
seatingAssignmentLabel.setText("Seating Assignment");

textArea.setColumns(20);
textArea.setLineWrap(true);
textArea.setRows(5);
textArea.setText("Hi, clientA!\n\n");
textArea.setWrapStyleWord(true);
scrollPane.setViewportView(textArea);

loginMenu.setText("Login");

createNewAccountMenuItem.setText("Create a new account");
createNewAccountMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        createNewAccountMenuItemActionPerformed(evt);
    }
});
loginMenu.add(createNewAccountMenuItem);

loginMenuItem.setText("Login to the system");
loginMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        loginMenuItemActionPerformed(evt);
    }
});
loginMenu.add(loginMenuItem);

deleteAccountMenuItem.setText("Delete an account");
deleteAccountMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        deleteAccountMenuItemActionPerformed(evt);
    }
});
loginMenu.add(deleteAccountMenuItem);

mainMenuBar.add(loginMenu);

optionsMenu.setText("Options");

createNewLayoutMenuItem.setText("Create a new layout");
createNewLayoutMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        createNewLayoutMenuItemActionPerformed(evt);
    }
});
optionsMenu.add(createNewLayoutMenuItem);

startAssigningSeatsMenuItem.setText("Start assigning seats");
startAssigningSeatsMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        startAssigningSeatsMenuItemActionPerformed(evt);
    }
});
optionsMenu.add(startAssigningSeatsMenuItem);

checkNewFeaturesMenuItem.setText("Check new features");
checkNewFeaturesMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        checkNewFeaturesMenuItemActionPerformed(evt);
    }
}

```

```
});  
optionsMenu.add(checkNewFeaturesMenuItem);  
  
mainMenuBar.add(optionsMenu);  
  
setJMenuBar(mainMenuBar);  
  
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());  
getContentPane().setLayout(layout);  
layout.setHorizontalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(layout.createSequentialGroup()  
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)                .addGroup(layout.createSequentialGroup()  
                    .addComponent(seatingAssignmentLabel,  
                        javax.swing.GroupLayout.PREFERRED_SIZE, 193,  
                        javax.swing.GroupLayout.PREFERRED_SIZE))  
                    .addGroup(layout.createSequentialGroup()  
                        .addGap(103, 103, 103)  
                        .addComponent(scrollPane, javax.swing.GroupLayout.PREFERRED_SIZE, 382,  
                            javax.swing.GroupLayout.PREFERRED_SIZE)))  
                .addContainerGap(143, Short.MAX_VALUE))  
        );  
layout.setVerticalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(layout.createSequentialGroup()  
            .addGap(41, 41, 41)  
            .addComponent(seatingAssignmentLabel,  
                javax.swing.GroupLayout.PREFERRED_SIZE, 58,  
                javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
            .addComponent(scrollPane, javax.swing.GroupLayout.PREFERRED_SIZE, 194,  
                javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addGap(152, 152, 152))  
        );  
  
pack();  
} // </editor-fold>
```

```
private void createNewAccountMenuItemActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
clearTextComponents(createNewAccountFrame);  
createNewAccountFrame.setVisible(true);  
createNewAccountFrame.pack();
```

```

accounts=new ArrayList<Account>();
//Scanner terminalScanner=new Scanner(System.in);
Scanner csvScanner = null;
try {
    csvScanner = new Scanner(new File("Accounts.csv"));
} catch (FileNotFoundException ex) {
    Logger.getLogger(GUI.class.getName()).log(Level.SEVERE, null, ex);
}
while(csvScanner.hasNext()) {
    String[] nextUserInfo=csvScanner.nextLine().split(SEPARATOR);
    AccountManager.addAccount(nextUserInfo[0], nextUserInfo[1]);
}
csvScanner.close();

}

private void usernameNewAccountTextFieldActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
}

private void
passwordNewAccountPasswordFieldActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void passwordLoginPasswordFieldActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void loginMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    clearTextComponents(loginFrame);
    loginFrame.setVisible(true);
    loginFrame.pack();
}

private void deleteAccountMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    clearTextComponents(deleteAccountFrame);
    deleteAccountFrame.setVisible(true);
    deleteAccountFrame.pack();
}

```

```

private void createNewAccountButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        // TODO add your handling code here:
        // String myAccount = usernameNewAccountTextField.getText().trim();
        // char[] myPassword = passwordNewAccountPasswordField.getPassword();

        /*
        User foundCustomer = library.getUser(email);
        if (foundCustomer == null || foundCustomer.getPassword().length() != password.length)
        {
            JOptionPane.showMessageDialog(loginFrame, "Login failed, incorrect email or
password");
        } else {
            boolean flag = true;
            for (int i = 0; i < password.length; i++) {
                if (password[i] != foundCustomer.getPassword().charAt(i)) {
                    JOptionPane.showMessageDialog(loginFrame, "Login failed, incorrect email or
password");
                    flag = false;
                }
            }
            if (flag) {
                currentUser = foundCustomer;
                JOptionPane.showMessageDialog(loginFrame, "Successfully logged in.");
                checkoutMenu.setEnabled(true);
                adminMenu.setEnabled(true);
                loginFrame.setVisible(false);
            }
        }
        */

        throw new Exception("Not implemented yet");
    } catch (Exception ex) {
        Logger.getLogger(GUI.class.getName()).log(Level.SEVERE, null, ex);
    }
    while(true) {
        String myUsername = usernameNewAccountTextField.getText().trim();
        char[] myPassword = passwordNewAccountPasswordField.getPassword();

        int ok=1;
        for(Account acc:accounts) {
            if(acc.getUsername().equals(myUsername)) {
                //System.out.println("Account already exist! Create your account again.");
                JOptionPane.showMessageDialog(createNewAccountFrame, "Account already
exist! Create your account again.");
                ok=0;
            }
        }
    }
}

```

```

        break;
    }
}
if(ok==0) {
    usernameNewAccountTextField.setText("");
    passwordNewAccountPasswordField.setText("");
    continue;
}
//System.out.println("Account successssfully created!");
JOptionPane.showMessageDialog(createNewAccountFrame, "Account successssfully
created!");
Account myAcc=new Account(myUsername,myPassword);
AccountManager.addAccount(myAcc);

try {
    //add the new account to Accounts.csv
    BufferedWriter csvWriter = new BufferedWriter(new FileWriter(new
File("Accounts.csv")));
    for(Account _acc:accounts) {
        csvWriter.write(_acc.toCSV() + "\n");
        csvWriter.flush();
    }
    csvWriter.close();
    //add a new directory containing the Accounts information
    File f=new File("Accounts/"+myUsername);
    f.mkdir();

} catch (IOException ex) {
    Logger.getLogger(AccountManager.class.getName()).log(Level.SEVERE, null, ex);
}
currentAccount=myAcc;

try {
    //input student namelist.csv
    BufferedWriter csvWriter = new BufferedWriter(new FileWriter(new
File("Accounts/"+myUsername+"/namelist.csv")));
    csvWriter.write("<Input student namelist here. One student per line.>\n");
    csvWriter.flush();
    csvWriter.close();
} catch (IOException ex) {
    Logger.getLogger(AccountManager.class.getName()).log(Level.SEVERE, null, ex);
}
System.out.println("Import student list at "+"Accounts/"+myUsername+".csv");
}
}

```

```

private void confirmLoginButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JOptionPane.showMessageDialog(loginFrame, "Login successful!");
}

private void createNewLayoutMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    clearTextComponents(createNewLayoutFrame);
    createNewLayoutFrame.setVisible(true);
    createNewLayoutFrame.pack();
}

private void startAssigningSeatsMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void checkNewFeaturesMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    clearTextComponents(displayFrame);
    displayFrame.setVisible(true);
    displayFrame.pack();
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

```

```

        javax.swing.UIManager.setLookAndFeel(info.getClassName());
        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(GUI.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(GUI.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {

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

java.util.logging.Logger.getLogger(GUI.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new GUI().setVisible(true);
    }
});
}

private void clearTextComponents(JFrame jf) {
    for (Component c : jf.getContentPane().getComponents()) {
        if (c instanceof JTextField) {
            ((JTextComponent) c).setText("");
        }

    }
    for (Component c : jf.getComponents()) {
        if (c instanceof JTextArea) {
            ((JTextArea) c).setText("");
        }
    }
}
}

```

```
// Variables declaration - do not modify
private javax.swing.JMenuItem checkNewFeaturesMenuItem;
private javax.swing.JButton confirmDeleteButton;
private javax.swing.JButton confirmLoginButton;
private javax.swing.JButton createNewAccountButton;
private javax.swing.JFrame createNewAccountFrame;
private javax.swing.JLabel createNewAccountLabel;
private javax.swing.JMenuItem createNewAccountMenuItem;
private javax.swing.JFrame createNewLayoutFrame;
private javax.swing.JMenuItem createNewLayoutMenuItem;
private javax.swing.JFrame deleteAccountFrame;
private javax.swing.JLabel deleteAccountLabel;
private javax.swing.JMenuItem deleteAccountMenuItem;
private javax.swing.JScrollPane deleteAccountScrollPane;
private javax.swing.JTextArea deleteAccountTextArea;
private javax.swing.JFrame displayFrame;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JFrame jFrame3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextArea jTextArea2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JFrame loginFrame;
private javax.swing.JLabel loginLabel;
private javax.swing.JMenu loginMenu;
private javax.swing.JMenuItem loginMenuItem;
private javax.swing.JMenuBar mainMenuBar;
private javax.swing.JMenu optionsMenu;
private javax.swing.JPasswordField passwordDeleeAccountPasswordField;
private javax.swing.JLabel passwordDeleteAccountLabel;
private javax.swing.JLabel passwordLoginLabel;
```

```

private javax.swing.JPasswordField passwordLoginPasswordField;
private javax.swing.JLabel passwordNewAccountLabel;
private javax.swing.JPasswordField passwordNewAccountPasswordField;
private javax.swing.JScrollPane scrollPane;
private javax.swing.JLabel seatingAssignmentLabel;
private javax.swing.JFrame startAssigningSeatsFrame;
private javax.swing.JMenuItem startAssigningSeatsMenuItem;
private javax.swing.JTextArea textArea;
private javax.swing.JLabel usernameDeleteAccountLabel;
private javax.swing.JTextField usernameDeleteAccountTextField;
private javax.swing.JLabel usernameLoginLabel;
private javax.swing.JTextField usernameLoginTextField;
private javax.swing.JLabel usernameNewAccountLabel;
private javax.swing.JTextField usernameNewAccountTextField;
// End of variables declaration
}

```

Student class

```

/*
 * 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.
 */
package seatingassignment;

/**
 *
 * @author Student-3
 */
public class Student {

    private String name;
    private Pair position;
    private int notSeatWithNumber;

    /**
     * Constructor
     * Create a Student object with a name
     * @param _name name of the student
     */
    public Student(String _name) {
        name=_name;
    }
}

```

```

/**
 * Constructor
 * Create a Student object with the name and the position
 * @param _name name of the student
 * @param _position position of the student
 */
public Student(String _name, Pair _position) {
    name=_name;
    position=_position;
    notSeatWithNumber=-1;//default
}

/**
 * Constructor
 * Create a Student object with the name, the position, and its advanced property
 * Not actually implemented
 * @param _name name of the student
 * @param _position position of the student
 * @param _notSeatWithNumber advanced feature of the student
 */
public Student(String _name, Pair _position, int _notSeatWithNumber) {
    name=_name;
    position=_position;
    notSeatWithNumber=_notSeatWithNumber;
}

/**
 * Accessor
 * Get the name of the student
 * @return name of the student
 */
public String getName() {
    return name;
}

/**
 * Mutator
 * Set the name of the student to _name
 * @param _name name assigned
 */
public void setName(String _name) {
    name=_name;
}

/**

```

```

    * Accessor
    * Get the position of the student
    * @return position of the student
    */
    public Pair getPos() {
        return position;
    }

    /**
     * Mutator
     * Set the position of the student to _pos
     * @param _pos position assigned
     */
    public void setPos(Pair _pos) {
        position=_pos;
    }

    /**
     * Accessor
     * Get the advanced feature of the student
     * @return advanced feature of the student
     */
    public int getNotSeatWithNumber() {
        return notSeatWithNumber;
    }

    /**
     * Mutator
     * Set the advanced feature of the student
     * @param _notSeatWithNumber advanced feature assigned
     */
    public void setNotSeatWithNumber(int _notSeatWithNumber) {
        notSeatWithNumber=_notSeatWithNumber;
    }
}

```

SeatingAssignment class

```

/*
 * 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.
 */
package seatingassignment;

```

```

import java.io.FileNotFoundException;
import java.util.Scanner;

/**
 *
 * @author Student-3
 */
public class SeatingAssignment {

    /**
     * Main class of the program
     * @param args the command line arguments
     * @throws FileNotFoundException
     * @throws Exception
     */

    public static void main(String[] args) throws FileNotFoundException, Exception {

        Scanner scn=new Scanner(System.in);
        System.out.println("Welcome to the Seating Assignment Program, developed by Stanley!");

        Account myAcc=AccountManager.initAccount();
        if(myAcc==null) throw new Exception("Please login again.");
        System.out.println("Hi, "+AccountManager.currentAccount.getUsername());

        System.out.println("Options:\n1. create a new layout\n2. start assigning seats\n3. check
features");

        int opt=scn.nextInt();
        if(opt==1) {
            Layout newLayout=new Layout();
            //a new layout is added to the account
        }
        else if(opt==2) {
            Scheme newScheme = new Scheme();
            newScheme.RandomSeating();//start assigning seats
        }
        else if(opt==3) {
            System.out.println("Current feature: random seating.\nOther features not implemented.");
        }
        else {
            throw new Exception("Invalid option!");
        }
    }
}

```

Scheme class

```
/*
 * 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.
 */
package seatingassignment;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Student-3
 */
public class Scheme {

    private String name;
    private Layout layout;
    private ArrayList<Student> students;
    private int numberOfStudents;
    private boolean[][] map;
    private String[][] mapNames;

    /**
     * Default constructor
     * Create a default scheme
     */
    public Scheme() {
        name="";
        layout=null;
        students=new ArrayList<Student>();
        numberOfStudents=-1;
        map=new boolean[0][0];
    }
}
```

```

        mapNames=new String[0][0];
    }

/**
 * Constructor
 * Create a normal scheme
 * @param _name name of the scheme
 * @param _layout layout of the scheme
 * @param _students List of students
 */
public Scheme(String _name, Layout _layout, ArrayList<Student> _students) {
    name=_name;
    layout=_layout;
    students=_students;
    numberOfStudents=students.size();
    map=_layout.getMap();
    mapNames=new String[0][0];
}

/**
 * Accessor
 * Get the name of the scheme
 * @return name of the scheme
 */
public String getName() {
    return name;
}

/**
 * Mutator
 * Set the name of the scheme to _name
 * @param _name name assigned
 */
public void setName(String _name) {
    name=_name;
}

/**
 * Accessor
 * Get the number of students in the scheme
 * @return number of students in the scheme
 */
public int getNumberOfStudent() {
    return numberOfStudents;
}

```

```

/**
 * Accessor
 * Get the layout of the scheme
 * @return layout of the scheme
 */
public Layout getLayout() {
    return layout;
}

/**
 * Mutator
 * Set the layout of the scheme to _layout
 * @param _layout layout value assigned
 */
public void setLayout(Layout _layout) {
    layout=_layout;
}

/**
 * Accessor
 * Get the student list in the scheme
 * @return student list in the scheme
 */
public ArrayList<Student> getStudentList() {
    return students;
}

/**
 * Mutator
 * Set the student list in the scheme to _students
 * @param _students student list assigned
 */
public void setStudentList(ArrayList<Student> _students) {
    students=_students;
}

/**
 * Perform random seating on the scheme
 * @throws Exception
 */
public void RandomSeating() throws Exception {
    //initialize
    String username=AccountManager.currentAccount.getUsername();
    String thePath="Accounts/"+username+"/namelist.csv";
    Scanner fileReader = null;
    Scanner scn = new Scanner(System.in);

```

```

int ok=1;

//check if namelist.csv exists
try {
    fileReader = new Scanner(new File(thePath));
} catch(FileNotFoundException e) {
    ok=0;
    System.out.println("Input the namelist at "+thePath);
}
if(ok==0) {
    System.out.println("Please input your namelist now, otherwise the program may crash!
Input anything to continue assigning your seats.");
    File file1 = new File(thePath);
    file1.createNewFile();
    String emptyStr=scn.next();
}

//read in the student namelist
fileReader = new Scanner(new File(thePath));
while (fileReader.hasNext()) {
    String aName=fileReader.nextLine();
    //System.out.println("The student name is "+aName);
    students.add(new Student(aName));
}
fileReader.close();

//read the choice of layout
System.out.println("Which layout do you want to use? Input the name of layout you want to
use. The list of layouts is shown below:");
String theName = null;
String yourChoice = null;
ArrayList<String> theNames = new ArrayList<String>();
File folder = new File("Accounts/"+username+"/");
File[] userFiles = folder.listFiles();
for(File file : userFiles) {
    if(file.getName().contains(".txt")) {
        theName=file.getName().substring(0,file.getName().length()-4);
        theNames.add(theName);
        System.out.print(theName+", ");
    }
}
System.out.println("");
scn=new Scanner(System.in);
ok=0;
while(ok==0) {
    yourChoice = scn.next();

```

```

System.out.println("Your choice is: "+yourChoice);
for(String aName:theNames)
    if(yourChoice.equals(aName)) {
        ok=1;
        break;
    }
if(ok==0)
    System.out.println("This name not found! Input again!");
}

String pathName="Accounts/"+username+"/"+yourChoice+".txt";
//System.out.println("The path is: "+pathName);
layout=new Layout(new File(pathName));
this.map=layout.map;
this.numberofStudents=layout.numSeats;

if(students.size()!=numberOfStudents) {
    System.out.println("Student number = "+students.size());
    System.out.println("Number of seats = "+ layout.numSeats);
    throw new Exception("Number of student does not match!");
}
else {
    mapNames=new String[layout.ROWS+1][layout.COLUMNS+1];
    int OK=0,filled=0;
    //ArrayList<String> book=new ArrayList<String>();
    ArrayList<String> seats=new ArrayList<String>();
    HashMap<String,Pair> pos=new HashMap<String,Pair>();
    HashMap<String,Integer> book=new HashMap<String,Integer>();
    while(OK==0) {
        filled=0;
        book.clear();
        seats.clear();
        pos.clear();
        while(filled<numberOfStudents) {
            int randm=(int)(Math.random()*numberOfStudents);
            //System.out.print("randm="+randm+"\n");
            if(book.get(students.get(randm).getName())==null) {
                book.put(students.get(randm).getName(),1);
                seats.add(students.get(randm).getName());
                filled++;
            }
        }
    }
    System.out.println("Attempting:");
    int tempnum=0;
    for(int i=1;i<=layout.ROWS;i++) {
        for(int j=1;j<=layout.COLUMNS;j++) {

```



\* To change this template file, choose Tools | Templates  
\* and open the template in the editor.  
\*/

```
package seatingassignment;
```

```
/**
```

```
 *
```

```
 * @author Student-3
```

```
 */
```

```
public class Pair {
```

```
    private int x;
```

```
    private int y;
```

```
    private int num;
```

```
    /**
```

```
     * Constructor
```

```
     * Create a Pair object using x and y coordinates
```

```
     * @param _x x coordinate of the position
```

```
     * @param _y y coordinate of the position
```

```
     */
```

```
    public Pair(int _x, int _y) {
```

```
        x=_x;
```

```
        y=_y;
```

```
        num=pairToNum(_x,_y);
```

```
    }
```

```
    /**
```

```
     * Constructor
```

```
     * Create a Pair object using a single number as the position
```

```
     * @param _num single number position
```

```
     */
```

```
    public Pair(int _num) {
```

```
        num=_num;
```

```
        y=_num/Layout.COLUMNS+1;
```

```
        x=_num-(y-1)*Layout.COLUMNS;
```

```
//        Pair _Pair=numToPair(_num);
```

```
//        x=_Pair.getX();
```

```
//        y=_Pair.getY();
```

```
    }
```

```
    /**
```

```
     * Convert an (x,y) pair to a single number
```

```
     * @param _x x coordinate of the position
```

```
     * @param _y y coordinate of the position
```

```
     * @return single number position
```

```

*/
public int pairToNum(int _x, int _y) {
    return (_y-1)*Layout.COLUMNS+_x;
}

/*
public Pair numToPair(int _num) {
    Pair _Pair=new Pair(_num);
    int _y=_num/Layout.COLUMNS+1;
    int _x=_num-(_y-1)*Layout.COLUMNS;
    _Pair.setX(_x);
    _Pair.setY(_y);
    return _Pair;
}
*/

/**
 * Accessor
 * Get the x coordinate of the pair
 * @return x coordinate of the pair
 */
public int getX() {
    return x;
}

/**
 * Mutator
 * Set the x coordinate of the pair to _x
 * @param _x x coordinate assigned
 */
public void setX(int _x) {
    x=_x;
}

/**
 * Accessor
 * Get the y coordinate of the pair
 * @return y coordinate of the pair
 */
public int getY() {
    return y;
}

/**
 * Mutator
 * Set the y coordinate of the pair to _y

```

```

    * @param _y y coordinate assigned
    */
    public void setY(int _y) {
        y=_y;
    }

    /**
     * Accessor
     * Get the single number coordinate of the pair
     * @return single number coordinate of the pair
     */
    public int getNum() {
        return num;
    }

    /**
     * Mutator
     * Set the single number coordinate of the pair to _num
     * @param _num single number coordinate assigned
     */
    public void setNum(int _num) {
        num=_num;
    }

    /**
     * Overrides the toString method, return the String format of the pair
     * @return String format of the pair
     */
    @Override
    public String toString() {
        return "("+x+", "+y+")";
    }
}

```

### 3. Consultation with the client at testing stage

Me: Hi A! How are you recently? Have you seen my current product?

Client A: I was fantastic when receiving your product! Those functions are so fascinating!

Me: Great. Are there any bugs? Did the program ever crash or “not responding”?

Client A: No, I think your program works well.

Me: I'm happy when hearing that. So, you've created your account and start assigning seats for our class?

Client A: One small thing is that, I don't know how to start assigning seats. It says “Requires students namelist.csv file”. I have no idea where to put the file and what kind of format there is for this file.

Me: Oh, then I will add a display of the correct path in my program. Thank you for the feedback! You should add it under Accounts/clientA/ directory.

Client A: Oh I see! A blank namelist.csv file is already created, but I didn't notice that! Sorry for being careless.

Me: It's okay. It is me being careless when not adding sufficient prompts.

Client A: Alright. So currently the program only supports random seating?

Me: That's right. Newer features will be added soon. Now I just want to make sure that the account management system is working properly, and seating plan produced is correct. Have you tried to complete a seating plan?

Client A: Wait for me to complete the namelist. Alright. Then this is the produced seating plan? Can I save it or edit it?

Me: You can choose whether to save it or discard it. By discarding it the system will produce another seating plan, until you are satisfied. A printable .csv file is produced under your directory.

Client A: Oh, I see! Wow, using file input and output is so fabulous! I can then edit the seating file and print it for the class. That's so wonderful.

Me: That's sure. Seems no other problems then. I will add other features as soon as possible. See you!

Client A: Thank you so much for helping me!