Code and Technical Document

PAT Last Phase – Grade 12



Prepared by Inge Senekal

Curro Roodeplaat

Contents

[Code: DB 3](#_Toc80980792)

[Code: User 4](#_Toc80980793)

[Login Screen 5](#_Toc80980794)

[Register Screen 11](#_Toc80980795)

[User Class 19](#_Toc80980796)

[UserManager Class 22](#_Toc80980797)

[Code :Application 25](#_Toc80980798)

[Application Screen 25](#_Toc80980799)

[Code: Task 30](#_Toc80980800)

[Task List Screen 30](#_Toc80980801)

[Task Detail Screen 41](#_Toc80980802)

[Planner Screen 53](#_Toc80980803)

[Planner Detail Screen 65](#_Toc80980804)

[Task 72](#_Toc80980805)

[Subject 77](#_Toc80980806)

[TaskManager 78](#_Toc80980807)

[JTableButtonRender 83](#_Toc80980808)

[Code Help 83](#_Toc80980809)

[Help Screen 83](#_Toc80980810)

[Externally Sourced Code 89](#_Toc80980811)

[Removing the boarder of a button: 89](#_Toc80980812)

[DB: 89](#_Toc80980813)

[Explanation of Critical Algorithms 91](#_Toc80980814)

[Getting Task Algorithm 91](#_Toc80980815)

[Why code is critical 91](#_Toc80980816)

[Insert or Updating a Task Algorithm 91](#_Toc80980817)

[Why code is critical 92](#_Toc80980818)

[Advanced Techniques 92](#_Toc80980819)

[Testing 97](#_Toc80980820)

[Evaluation of Solution 97](#_Toc80980821)

[Functional Testing: 97](#_Toc80980822)

[Test Plan 97](#_Toc80980823)

[Test Results 104](#_Toc80980824)

# Code: DB

This code was provided by the teacher it connects the database with the program.

package backend;

/\*

\* 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.

\*/

/\*

To get your Java progrm to connect to MS Access you have to download UCanAccess at

https://sourceforge.net/projects/ucanaccess/files/

and add the following JAR files to the classpath:

ucanaccess-4.0.4.jar

hsqldb-2.3.1.jar

jackcess-2.1.11.jar

commons-lang-2.6.jar

commons-logging-1.1.3.jar

The version numbers here may differ than the latest versions you downloaded.

https://www.codejava.net/java-se/jdbc/java-jdbc-example-connect-to-microsoft-access-database

Add these libraries to the NetBeans Project

\*/

import java.sql.\*;

/\*\*

\*

\* @author Mr Christophers

\*/

public class DB {

private static final String url = "jdbc:mysql://localhost:3306/optimise\_my\_time";

private Connection conn;

private PreparedStatement statement;

private ResultSet resultSet;

/\*\*

\* Constructor for the DB class, this contains data from the optimise\_my\_time database.

\*/

public DB() {

try {

conn = DriverManager.getConnection(url, "root", "nbuser");

} catch (SQLException ex) {

System.out.println("Cannot connect");

}

}

/\*\*

\* This method updates data in the optimise\_my\_time database

\* @param update (Contains the SQL database)

\* @throws SQLException

\*/

public void update(String update) throws SQLException {

statement = conn.prepareStatement(update);

statement.executeUpdate();

statement.close();

}

/\*\*

\* The method is used for any other SQL statement with no separate methods made in other classes

\* @param stmt (Contains the SQL statement

\* @return the SQL statement

\* @throws SQLException to bypass any error

\*/

public ResultSet query(String stmt) throws SQLException {

statement = conn.prepareStatement(stmt);

resultSet = statement.executeQuery();

return resultSet;

}

/\*\*

\*

\* @param rs

\* @return the meta data

\*/

public String processResultSet(ResultSet rs) {

String temp = "";

try {

ResultSetMetaData meta = rs.getMetaData();

int size = meta.getColumnCount();

while (rs.next()) {

for (int i = 1; i <= size; i++) {

Object value = rs.getObject(i);

temp += "#" + value;

}

temp += "\n";

}

} catch (SQLException ex) {

ex.printStackTrace();

}

return temp;

}

}

# Code: User

The following code is based on the **user**. This will include the screens for the user to Login and Register from the **frontend** and the User and UserManager class from the **backend** of the program.

## Login Screen

//Inge Senekal

package frontend;

import backend.TaskManager;

import backend.UserManager;

import java.awt.Color;

import javax.swing.BorderFactory;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Inges

\*/

public class LoginScreen extends javax.swing.JFrame {

/\*\*

\* Creates new form LoginScreen

\*/

public LoginScreen() {

initComponents();

//Sets the background colour for the JFrame to white

getContentPane().setBackground(Color.white);

//Removes the border around the buttons

btnRegister.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnRegister.setContentAreaFilled(false);

btnLogin.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnLogin.setContentAreaFilled(false);

//Calls the method populateSubjects from the backend class TaskManager

//Subjects are populated before user has logged in.

TaskManager.populateSubjects();

}

/\*\*

\* 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() {

lblLogo = new javax.swing.JLabel();

lblUserName = new javax.swing.JLabel();

lblPassword = new javax.swing.JLabel();

btnLogin = new javax.swing.JButton();

btnRegister = new javax.swing.JButton();

txtUserName = new javax.swing.JTextField();

psfPassword = new javax.swing.JPasswordField();

lblUserNameError = new javax.swing.JLabel();

lblPasswordError = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Optimise My Time");

setBackground(new java.awt.Color(255, 255, 255));

setMinimumSize(new java.awt.Dimension(570, 430));

setName("LoginFrame"); // NOI18N

setResizable(false);

lblLogo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/OMT.PNG"))); // NOI18N

lblLogo.setMaximumSize(new java.awt.Dimension(351, 226));

lblLogo.setMinimumSize(new java.awt.Dimension(351, 226));

lblLogo.setPreferredSize(new java.awt.Dimension(351, 226));

lblUserName.setFont(new java.awt.Font("Calibri", 1, 14)); // NOI18N

lblUserName.setForeground(new java.awt.Color(18, 125, 244));

lblUserName.setText("User Name");

lblPassword.setFont(new java.awt.Font("Calibri", 1, 14)); // NOI18N

lblPassword.setForeground(new java.awt.Color(18, 125, 244));

lblPassword.setText("Password");

btnLogin.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

btnLogin.setForeground(new java.awt.Color(18, 125, 244));

btnLogin.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Login.png"))); // NOI18N

btnLogin.setText("Log In");

btnLogin.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnLoginActionPerformed(evt);

}

});

btnRegister.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

btnRegister.setForeground(new java.awt.Color(18, 125, 244));

btnRegister.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Register.png"))); // NOI18N

btnRegister.setText("Register");

btnRegister.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnRegisterActionPerformed(evt);

}

});

txtUserName.setSelectionColor(new java.awt.Color(255, 255, 255));

lblUserNameError.setFont(new java.awt.Font("Calibri", 0, 10)); // NOI18N

lblUserNameError.setForeground(new java.awt.Color(255, 0, 0));

lblPasswordError.setFont(new java.awt.Font("Calibri", 0, 10)); // NOI18N

lblPasswordError.setForeground(new java.awt.Color(255, 0, 0));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(170, 170, 170)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addComponent(btnLogin)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 48, Short.MAX\_VALUE)

.addComponent(btnRegister))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(lblPassword, javax.swing.GroupLayout.PREFERRED\_SIZE, 65, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblUserName))

.addGap(55, 55, 55)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(txtUserName, javax.swing.GroupLayout.DEFAULT\_SIZE, 120, Short.MAX\_VALUE)

.addComponent(psfPassword))))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(lblUserNameError, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblPasswordError, javax.swing.GroupLayout.DEFAULT\_SIZE, 90, Short.MAX\_VALUE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(105, Short.MAX\_VALUE)

.addComponent(lblLogo, javax.swing.GroupLayout.PREFERRED\_SIZE, 289, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(134, 134, 134))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(lblLogo, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(24, 24, 24)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(txtUserName, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblUserName)

.addComponent(lblUserNameError, javax.swing.GroupLayout.PREFERRED\_SIZE, 21, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED, 32, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lblPassword)

.addComponent(psfPassword, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblPasswordError, javax.swing.GroupLayout.PREFERRED\_SIZE, 21, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(30, 30, 30)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(btnLogin)

.addComponent(btnRegister))

.addContainerGap(47, Short.MAX\_VALUE))

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {

//If Statement: If the username and password is not the length of 0

//If the username and password text fields are not empty

if (txtUserName.getText().length() > 0 && psfPassword.getPassword().length > 0) {

//Declaires the password as a String

String password = "";

//?

for (char ch : psfPassword.getPassword()) {

password += ch;

}

//Declaires loggedIn as a boolean

//Calls the the method login from the UserManager class

boolean loggedIn = UserManager.login(txtUserName.getText(), password);

//If loggedIn is true

if (loggedIn) {

//Than display the Application screen

new Application().setVisible(true);

this.dispose();

}else{

//Else Display a popup error message

JOptionPane.showMessageDialog(null, "The username or password is incorrect.");

}

} else {

//Else if username and password is empty

//Display a error message(in labels)informing the user that it has to be filled in.

lblUserNameError.setText("The username has to filled in");

lblPasswordError.setText("The password has to be filled in");

}

}

private void btnRegisterActionPerformed(java.awt.event.ActionEvent evt) {

//Opens Register screen

new Register().setVisible(true);

this.dispose();

}

/\*\*

\* @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(LoginScreen.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(LoginScreen.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(LoginScreen.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(LoginScreen.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 LoginScreen().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btnLogin;

private javax.swing.JButton btnRegister;

private javax.swing.JLabel lblLogo;

private javax.swing.JLabel lblPassword;

private javax.swing.JLabel lblPasswordError;

private javax.swing.JLabel lblUserName;

private javax.swing.JLabel lblUserNameError;

private javax.swing.JPasswordField psfPassword;

private javax.swing.JTextField txtUserName;

// End of variables declaration

} private javax.swing.JTextField txtEmailAddress;

private javax.swing.JTextField txtTelephone;

private javax.swing.JTextField txtUserName;

// End of variables declaration

}

## Register Screen

//Inge Senekal

package frontend;

import backend.UserManager;

import javax.swing.BorderFactory;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Inges

\*/

public class Register extends javax.swing.JFrame {

/\*\*

\* Creates new form Register

\*/

public Register() {

initComponents();

//Sets the background colour for the JFrame to white

getContentPane().setBackground(new java.awt.Color(255, 255, 255));

//Removes the border around the button

btnRegister.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnRegister.setContentAreaFilled(false);

}

/\*\*

\* 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() {

lblBackground = new javax.swing.JLabel();

lblPassword = new javax.swing.JLabel();

lblTelephone = new javax.swing.JLabel();

lblEmailAddress = new javax.swing.JLabel();

lblUserName = new javax.swing.JLabel();

txtTelephone = new javax.swing.JTextField();

txtUserName = new javax.swing.JTextField();

txtEmailAddress = new javax.swing.JTextField();

lblCommentName = new javax.swing.JLabel();

lblCommentTelephone = new javax.swing.JLabel();

lblCommentPassword = new javax.swing.JLabel();

btnRegister = new javax.swing.JButton();

psfPassword = new javax.swing.JPasswordField();

lblUserNameError = new javax.swing.JLabel();

lblEmailError = new javax.swing.JLabel();

lblPasswordError = new javax.swing.JLabel();

lblTelephoneError = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Optimise My Time");

setBackground(new java.awt.Color(255, 255, 255));

setMinimumSize(new java.awt.Dimension(405, 420));

setName("RegisterFrame"); // NOI18N

lblBackground.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/OMT.PNG"))); // NOI18N

lblPassword.setFont(new java.awt.Font("Calibri", 1, 14)); // NOI18N

lblPassword.setForeground(new java.awt.Color(18, 125, 244));

lblPassword.setText("Password:");

lblTelephone.setFont(new java.awt.Font("Calibri", 1, 14)); // NOI18N

lblTelephone.setForeground(new java.awt.Color(18, 125, 244));

lblTelephone.setText("Telephone:");

lblEmailAddress.setFont(new java.awt.Font("Calibri", 1, 14)); // NOI18N

lblEmailAddress.setForeground(new java.awt.Color(18, 125, 244));

lblEmailAddress.setText("Email Address:");

lblUserName.setFont(new java.awt.Font("Calibri", 1, 14)); // NOI18N

lblUserName.setForeground(new java.awt.Color(18, 125, 244));

lblUserName.setText("User Name: ");

txtTelephone.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

txtTelephoneActionPerformed(evt);

}

});

lblCommentName.setFont(new java.awt.Font("Calibri", 0, 8)); // NOI18N

lblCommentName.setText("First Name and Surname");

lblCommentTelephone.setFont(new java.awt.Font("Calibri", 0, 8)); // NOI18N

lblCommentTelephone.setText("(\*\*\* \*\*\* \*\*\*\*)");

lblCommentPassword.setFont(new java.awt.Font("Calibri", 0, 8)); // NOI18N

lblCommentPassword.setText("Minimum of 8 characters");

btnRegister.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

btnRegister.setForeground(new java.awt.Color(18, 125, 244));

btnRegister.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Register.png"))); // NOI18N

btnRegister.setText("Register");

btnRegister.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnRegisterActionPerformed(evt);

}

});

psfPassword.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

psfPasswordActionPerformed(evt);

}

});

lblUserNameError.setFont(new java.awt.Font("Calibri", 0, 10)); // NOI18N

lblUserNameError.setForeground(new java.awt.Color(255, 0, 0));

lblEmailError.setFont(new java.awt.Font("Calibri", 0, 10)); // NOI18N

lblEmailError.setForeground(new java.awt.Color(255, 0, 0));

lblPasswordError.setFont(new java.awt.Font("Calibri", 0, 10)); // NOI18N

lblPasswordError.setForeground(new java.awt.Color(255, 0, 0));

lblTelephoneError.setFont(new java.awt.Font("Calibri", 0, 10)); // NOI18N

lblTelephoneError.setForeground(new java.awt.Color(255, 0, 0));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(66, 66, 66)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(lblEmailAddress, javax.swing.GroupLayout.PREFERRED\_SIZE, 111, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblTelephone, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 101, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblPassword, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 99, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblUserName, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(30, 30, 30)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(lblCommentTelephone, javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(btnRegister, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(txtEmailAddress, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(txtTelephone, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(psfPassword, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 142, Short.MAX\_VALUE)

.addComponent(lblCommentPassword, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 99, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblCommentName, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(txtUserName, javax.swing.GroupLayout.Alignment.LEADING))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblUserNameError, javax.swing.GroupLayout.DEFAULT\_SIZE, 74, Short.MAX\_VALUE)

.addComponent(lblPasswordError, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblTelephoneError, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblEmailError, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))))

.addGap(89, 89, 89))))

.addGroup(layout.createSequentialGroup()

.addGap(110, 110, 110)

.addComponent(lblBackground)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(lblBackground, javax.swing.GroupLayout.PREFERRED\_SIZE, 177, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(3, 3, 3)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblUserNameError, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 13, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(txtUserName, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblUserName)))

.addGap(1, 1, 1)

.addComponent(lblCommentName)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lblPassword)

.addComponent(psfPassword, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblPasswordError, javax.swing.GroupLayout.PREFERRED\_SIZE, 21, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lblCommentPassword)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(txtTelephone, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblTelephone)

.addComponent(lblTelephoneError, javax.swing.GroupLayout.PREFERRED\_SIZE, 21, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lblCommentTelephone, javax.swing.GroupLayout.PREFERRED\_SIZE, 13, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblEmailError, javax.swing.GroupLayout.PREFERRED\_SIZE, 21, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lblEmailAddress)

.addComponent(txtEmailAddress, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(18, 18, 18)

.addComponent(btnRegister)

.addContainerGap(30, Short.MAX\_VALUE))

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void btnRegisterActionPerformed(java.awt.event.ActionEvent evt) {

//If the username, password, telephone, email text fields are not empty

boolean errors = false;

if(txtUserName.getText().length() > 45){

lblUserNameError.setText("Username is too long.");

errors = true;

}

if(txtEmailAddress.getText().length() > 150){

lblEmailError.setText("Email is too long.");

errors = true;

}

if(txtTelephone.getText().length() > 13){

lblTelephoneError.setText("Phone is too long.");

errors = true;

}

if(psfPassword.getPassword().length > 45){

lblPassword.setText("Password is too long.");

errors = true;

}else if(psfPassword.getPassword().length < 8){

lblPassword.setText("Password is too short.");

errors = true;

}

try{

Double.parseDouble(txtTelephone.getText());

}catch(Exception e){

lblTelephoneError.setText("Not a vald telephone number");

errors = true;

}

if(!txtEmailAddress.getText().contains("@") && !txtEmailAddress.getText().contains(".")){

lblEmailError.setText("Not a valid email address");

errors = true;

}

if (errors == false && txtUserName.getText().length() > 0 && psfPassword.getPassword().length > 0 && txtTelephone.getText().length() > 0 && txtEmailAddress.getText().length() > 0) {

//Declaires the password as a String

String password = "";

for (char ch : psfPassword.getPassword()) {

password += ch;

}

//Declaires registerIn as a boolean

//Calls the the method register from the UserManager class

boolean registerIn = UserManager.register(txtUserName.getText(), password, txtEmailAddress.getText(), txtTelephone.getText());

//If registerIn is true

if (registerIn) {

//Than display the Application screen

new Application().setVisible(true);

this.dispose();

} else {

//Else Display a popup error message

JOptionPane.showMessageDialog(null, "The user exists.");

}

} else {

//Else if username, password, telephone and email is empty

//Display a error message(in labels)informing the user that it has to be filled in.

lblUserNameError.setText("The username has to filled in");

lblPasswordError.setText("The password has to be filled in");

lblTelephoneError.setText("The telephone has to be filled in");

lblEmailError.setText("The email has to be filled in");

}

}

// Variables declaration - do not modify

private javax.swing.JButton btnRegister;

private javax.swing.JLabel lblBackground;

private javax.swing.JLabel lblCommentName;

private javax.swing.JLabel lblCommentPassword;

private javax.swing.JLabel lblCommentTelephone;

private javax.swing.JLabel lblEmailAddress;

private javax.swing.JLabel lblEmailError;

private javax.swing.JLabel lblPassword;

private javax.swing.JLabel lblPasswordError;

private javax.swing.JLabel lblTelephone;

private javax.swing.JLabel lblTelephoneError;

private javax.swing.JLabel lblUserName;

private javax.swing.JLabel lblUserNameError;

private javax.swing.JPasswordField psfPassword;

private javax.swing.JTextField txtEmailAddress;

private javax.swing.JTextField txtTelephone;

private javax.swing.JTextField txtUserName;

// End of variables declaration

}

## User Class

//Inge Senekal

package backend;

/\*\*

\*

\* @author Inges

\*/

public class User {

//Declair userID as an integer

//Declair userName, password, userEmail, userPhone as a String

private int userID;

private String userName;

private String password;

private String userEmail;

private String userPhone;

/\*\*

\*Parameters found inside the constructor method

\* @param userID stores the unique identifier for the user

\* @param userName stores the name of the user

\* @param password stores the password of the user

\* @param userEmail stores the email address of the user

\* @param userPhone stores the phone number of the user

\*

\*/

//Constructor method instantiates the user objects

public User(int userID, String userName, String password, String userEmail, String userPhone) {

this.userID = userID;

this.userName = userName;

this.password = password;

this.userEmail = userEmail;

this.userPhone = userPhone;

}

/\*\*

\*Accessor method gets the user’s user ID

\* returns the integer value of the user ID

\*/

public int getUserID() {

return userID;

}

/\*\*

\*Mutator method updates the user’s user ID

\* Used to change the userID

\*/

public void setUserID(int userID) {

this.userID = userID;

}

/\*\*

\*Accessor method gets the username

\* returns the string value of the username

\*/

public String getUserName() {

return userName;

}

/\*\*

\*Accessor method gets the password

\* returns the string value of the password

\*/

public String getPassword() {

return password;

}

/\*\*

\*Accessor method gets the users email address

\* returns the string value of the user Email address

\*/

public String getUserEmail() {

return userEmail;

}

/\*\*

\*Accessor method gets the users phone number

\* returns the string value of the user phone numbers

\*/

public String getUserPhone() {

return userPhone;

}

/\*\*

\*Mutator method updates the username

\* Used to change the username

\*/

public void setUserName(String name) {

userName = name;

}

/\*\*

\*Mutator method updates the password

\* Used to change the password

\*/

public void setPassword(String p) {

password = p;

}

/\*\*

\*Mutator method update the users email address

\* Used to change the email

\*/

public void setUserEmail(String email) {

userEmail = email;

}

/\*\*

\*Mutator method updates the users phone number

\* Used to change the phone

\*/

public void setUserPhone(String phone) {

userPhone = phone;

}

}

## UserManager Class

//Inge Senekal

package backend;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author Inges

\*/

public class UserManager {

/\*\*

\*Creates an object of the user class

\*/

public static User user;

/\*\*

\*Create an object of the DB class which can not be changed

\*/

public static final DB db = new DB();

/\*\*

\* The login in methods allows the user to enter its user name and password.

\* The method will go through the database at the table user and search for the record related to the variable.

\* @param username It is the users username

\* @param password It the users password

\* @return true or false based on whether the user can log in.

\*/

public static boolean login(String username, String password) {

try {

ResultSet rs = db.query("Select \* From tblUser Where username = '"

+ username + "' and password = '" + password + "';");

if (rs.next()) {

user = new User(rs.getInt("UserID"), rs.getString("username"), rs.getString("password"), rs.getString("email"), rs.getString("telephone"));

return true;

} else {

return false;

}

} catch (SQLException ex) {

Logger.getLogger(UserManager.class.getName()).log(Level.SEVERE, null, ex);

}

return false;

}

/\*\*

\* The register method allows a new user to enter their username, password, phone number and email address

\* The method will go through the database at the table user and search for the record related to the variable username.

\* If it is not related to a existing user it will update the User table by adding a new record.

\* @param username It is the new users username

\* @param password It is the new users password

\* @param email It is the new users email address

\* @param phone It is the new users phone number

\* @return It returns true or false based on wither the user is already in the database or not.

\* If the user is not in the database it adds a new record to the database with the above parameters

\*/

public static boolean register(String username, String password, String email, String phone) {

try {

ResultSet rs = db.query("Select \* From tblUser Where username = '"

+ username + "';");

if (rs.next()) {

return false;

} else {

db.update("Insert Into tblUser(username, password, telephone, email) VALUES('" + username + "','" + password + "','" + phone + "','" + email + "');");

}

} catch (SQLException ex) {

Logger.getLogger(UserManager.class.getName()).log(Level.SEVERE, null, ex);

}

return login(username, password);

}

}

# Code :Application

## Application Screen

The following code is based of only the Application screen. This screen is found in the **frontend** and contains not links the any **backend** classes.

//Inge Senekal

package frontend;

import backend.UserManager;

import java.awt.Color;

import javax.swing.BorderFactory;

/\*\*

\*

\* @author Inges

\*/

public class Application extends javax.swing.JFrame {

/\*\*

\* Creates new form Application

\*/

public Application() {

initComponents();

//Sets the background colour for the JFrame to white

getContentPane().setBackground(Color.white);

//Removes the border around the buttons

btnCalendar.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnCalendar.setContentAreaFilled(false);

btnTaskList.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnTaskList.setContentAreaFilled(false);

btnLogOut.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnLogOut.setContentAreaFilled(false);

btnHelp.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnHelp.setContentAreaFilled(false);

//Gives the label the text of the user name that currently uses the program.

lblShowUserName.setText(UserManager.user.getUserName());

revalidate();

}

/\*\*

\* 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() {

lblIcon = new javax.swing.JLabel();

lblUserName = new javax.swing.JLabel();

lblShowUserName = new javax.swing.JLabel();

lblWelcome = new javax.swing.JLabel();

btnCalendar = new javax.swing.JButton();

btnTaskList = new javax.swing.JButton();

lblCalendar = new javax.swing.JLabel();

btnLogOut = new javax.swing.JButton();

lblTaskList = new javax.swing.JLabel();

lblLogOut = new javax.swing.JLabel();

btnHelp = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Optimise My Time");

setMaximumSize(new java.awt.Dimension(500, 400));

setMinimumSize(new java.awt.Dimension(500, 400));

setPreferredSize(new java.awt.Dimension(500, 400));

setResizable(false);

getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

lblIcon.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/OMT\_1.PNG"))); // NOI18N

getContentPane().add(lblIcon, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 11, 102, 72));

lblUserName.setFont(new java.awt.Font("Calibri", 1, 11)); // NOI18N

lblUserName.setForeground(new java.awt.Color(18, 125, 244));

lblUserName.setText("User Name");

getContentPane().add(lblUserName, new org.netbeans.lib.awtextra.AbsoluteConstraints(330, 33, 56, -1));

lblShowUserName.setFont(new java.awt.Font("Calibri", 0, 11)); // NOI18N

getContentPane().add(lblShowUserName, new org.netbeans.lib.awtextra.AbsoluteConstraints(390, 30, 96, 15));

lblWelcome.setFont(new java.awt.Font("Calibri", 0, 36)); // NOI18N

lblWelcome.setForeground(new java.awt.Color(18, 125, 244));

lblWelcome.setText("Welcome");

getContentPane().add(lblWelcome, new org.netbeans.lib.awtextra.AbsoluteConstraints(165, 100, 153, -1));

btnCalendar.setBackground(new java.awt.Color(255, 255, 255));

btnCalendar.setForeground(new java.awt.Color(255, 255, 255));

btnCalendar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Calendar.png"))); // NOI18N

btnCalendar.setBorder(null);

btnCalendar.setMargin(new java.awt.Insets(2, 2, 2, 2));

btnCalendar.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnCalendarActionPerformed(evt);

}

});

getContentPane().add(btnCalendar, new org.netbeans.lib.awtextra.AbsoluteConstraints(193, 208, 106, -1));

btnTaskList.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/TaskList.png"))); // NOI18N

btnTaskList.setHideActionText(true);

btnTaskList.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnTaskListActionPerformed(evt);

}

});

getContentPane().add(btnTaskList, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 210, 116, 97));

lblCalendar.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

lblCalendar.setForeground(new java.awt.Color(18, 125, 244));

lblCalendar.setText("<html>Click here to<br/>view your calendar</html>");

getContentPane().add(lblCalendar, new org.netbeans.lib.awtextra.AbsoluteConstraints(206, 162, 106, 34));

btnLogOut.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/LogOut.png"))); // NOI18N

btnLogOut.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnLogOutActionPerformed(evt);

}

});

getContentPane().add(btnLogOut, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 210, 100, -1));

lblTaskList.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

lblTaskList.setForeground(new java.awt.Color(18, 125, 244));

lblTaskList.setText("<html>Click here to<br/>view your task list</html>");

getContentPane().add(lblTaskList, new org.netbeans.lib.awtextra.AbsoluteConstraints(67, 164, -1, -1));

lblLogOut.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

lblLogOut.setForeground(new java.awt.Color(18, 125, 244));

lblLogOut.setText("<html>Click here to<br/>log out</html>");

getContentPane().add(lblLogOut, new org.netbeans.lib.awtextra.AbsoluteConstraints(330, 164, 65, -1));

btnHelp.setFont(new java.awt.Font("Calibri", 0, 11)); // NOI18N

btnHelp.setForeground(new java.awt.Color(18, 125, 244));

btnHelp.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Help.png"))); // NOI18N

btnHelp.setText("Get Help");

btnHelp.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnHelpActionPerformed(evt);

}

});

getContentPane().add(btnHelp, new org.netbeans.lib.awtextra.AbsoluteConstraints(391, 63, -1, 24));

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void btnLogOutActionPerformed(java.awt.event.ActionEvent evt) {

//Opens the Login Screen

new LoginScreen().setVisible(true);

this.dispose();

}

private void btnCalendarActionPerformed(java.awt.event.ActionEvent evt) {

//Opens the Planner Screen

new Planner().setVisible(true);

this.dispose();

}

private void btnTaskListActionPerformed(java.awt.event.ActionEvent evt) {

//Opens the Task List Screen

new TaskList().setVisible(true);

this.dispose();

}

private void btnHelpActionPerformed(java.awt.event.ActionEvent evt) {

//Opens the Help Screen

new Help(this).setVisible(true);

this.dispose();

}

/\*\*

\* @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(Application.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Application.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Application.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Application.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 Application().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btnCalendar;

private javax.swing.JButton btnHelp;

private javax.swing.JButton btnLogOut;

private javax.swing.JButton btnTaskList;

private javax.swing.JLabel lblCalendar;

private javax.swing.JLabel lblIcon;

private javax.swing.JLabel lblLogOut;

private javax.swing.JLabel lblShowUserName;

private javax.swing.JLabel lblTaskList;

private javax.swing.JLabel lblUserName;

private javax.swing.JLabel lblWelcome;

// End of variables declaration

}

# Code: Task

The following code is based on the **task of the user**. This will include the screens for the task such as the Task List, Task Detail and Planner, Planner Detail from the **frontend** and the Task, Subject and TaskManager class from the **backend** of the program

## Task List Screen

//Inge Senekal

package frontend;

//Import classes used in this screen.

import backend.JTableButtonRenderer;

import backend.Task;

import backend.TaskManager;

import backend.UserManager;

import java.awt.Color;

import java.awt.Component;

import java.awt.event.ActionEvent;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import javax.swing.BorderFactory;

import javax.swing.JButton;

import javax.swing.JTable;

import javax.swing.table.\*;

/\*\*

\*

\* @author Inges

\*/

public class TaskList extends javax.swing.JFrame {

/\*\*

\* Creates new form TaskList

\*/

public TaskList() {

initComponents();

//Sets the background colour for the JFrame to white.

getContentPane().setBackground(Color.white);

//Removes the border around the buttons.

btnNewTask.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnNewTask.setContentAreaFilled(false);

btnReturnHome.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnReturnHome.setContentAreaFilled(false);

btnGetHelp.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnGetHelp.setContentAreaFilled(false);

//Gives the label the text of the user name that currently uses the program.

lblShowUserName.setText(UserManager.user.getUserName());

revalidate();

populateTaskList();

}

private void populateTaskList() {

//Creates an array of objects from the TaskManager class.

Task[] taskList = TaskManager.getTasks();

//Creates a array of the headings for each columb in the table below.

String[] headers = {

"Task Name", "Task Status", "Subject", "Task Priority", "Task Due Date", "Task Duration", "Edit", "Delete"

};

//Creates and object array in the form of a table.

//It has 8 columbs and the total amount of rows as tasks found in the database.

Object[][] tasks = new Object[taskList.length][8];

//Start loop until it is equal to the amount of task in database.

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

//Creates a button to edit everytime a task is found.

JButton btnEdit = new JButton();

//The size of the button has a width of 20 and a height of 20.

btnEdit.setSize(20, 20);

//The button contains an icon which is found in the resources package

btnEdit.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Edit.png"))); // NOI18N

btnEdit.setActionCommand("" + taskList[i].getTaskID());

//An action listener, when the button is selected it will perform an action.

//This action is to open the Task Detial Screen. To edit a task.

btnEdit.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnEditActionPerformed(evt);

}

});

//Creates a button to delete.

JButton btnDelete = new JButton();

//The height and width of the button is 20.

btnDelete.setSize(20, 20);

//The button contains a icon which is found in the resources package

btnDelete.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/delete.png"))); // NOI18N

btnDelete.setActionCommand("" + taskList[i].getTaskID());

//An action listener, when the button is selected it will perform an action.

//This action is to delete the task

btnDelete.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnDeleteActionPerformed(evt);

}

});

//Sets the positions of the table to according to the

//task name,status, subject name, subject priority, enddate and time allocated.

//Sets the last two positions equal to the edit and delete buttons

tasks[i][0] = taskList[i].getTaskName();

tasks[i][1] = taskList[i].getTaskStatus();

tasks[i][2] = taskList[i].getSubjName();

tasks[i][3] = taskList[i].getSubjPriority();

tasks[i][4] = taskList[i].getTaskEndDate();

tasks[i][5] = taskList[i].getTimeAllocation();

tasks[i][6] = btnEdit;

tasks[i][7] = btnDelete;

}

//Sets up the table according to the tasks and the headers.

tblTask.setModel(new DefaultTableModel(tasks, headers));

//Sets the height of each row equal to 30.

tblTask.setRowHeight(30);

//Sets the width of the column for delete and edit equal to 20.

tblTask.getColumn("Edit").setWidth(20);

tblTask.getColumn("Delete").setWidth(20);

//Adds the buttons to the columb where row is Edit and Delete.

TableCellRenderer buttonRenderer = new JTableButtonRenderer();

tblTask.getColumn("Edit").setCellRenderer(buttonRenderer);

tblTask.getColumn("Delete").setCellRenderer(buttonRenderer);

//Add a listener to that the table to determine the row and columb based on the click of the mouse

//This performs a action once the button is clicked.

tblTask.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent evt) {

int column = tblTask.getColumnModel().getColumnIndexAtX(evt.getX());

int row = evt.getY() / tblTask.getRowHeight();

if (row < tblTask.getRowCount() && row >= 0 && column < tblTask.getColumnCount() && column >= 0) {

Object value = tblTask.getValueAt(row, column);

if (value instanceof JButton) {

((JButton) value).doClick();

}

}

}

});

}

private void btnEditActionPerformed(ActionEvent evt) {

//Open Task Detail Screen

new TaskDetail(evt.getActionCommand()).setVisible(true);

this.dispose();

}

private void btnDeleteActionPerformed(ActionEvent evt) {

//Delete the task

//After task is deleted a updated Task List screen will be displayed

TaskManager.deleteTask(evt.getActionCommand());

new TaskList().setVisible(true);

this.dispose();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

lblOMT = new javax.swing.JLabel();

lblUserName = new javax.swing.JLabel();

lblShowUserName = new javax.swing.JLabel();

btnReturnHome = new javax.swing.JButton();

lblHomePage = new javax.swing.JLabel();

pnlTaskViewHeading = new javax.swing.JPanel();

lblTaskView = new javax.swing.JLabel();

pnlNewTask = new javax.swing.JPanel();

btnNewTask = new javax.swing.JButton();

btnGetHelp = new javax.swing.JButton();

lblSortBy = new javax.swing.JLabel();

splTableScroll = new javax.swing.JScrollPane();

tblTask = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Optimise My Time");

setResizable(false);

lblOMT.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/OMT\_1.PNG"))); // NOI18N

lblOMT.setText("jLabel1");

lblUserName.setFont(new java.awt.Font("Calibri", 1, 12)); // NOI18N

lblUserName.setForeground(new java.awt.Color(18, 125, 244));

lblUserName.setText("User Name");

lblShowUserName.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

btnReturnHome.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/HomePage.png"))); // NOI18N

btnReturnHome.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnReturnHomeActionPerformed(evt);

}

});

lblHomePage.setText("Return to home page ");

pnlTaskViewHeading.setBackground(new java.awt.Color(0, 204, 255));

lblTaskView.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblTaskView.setForeground(new java.awt.Color(255, 255, 255));

lblTaskView.setText("Task View");

pnlNewTask.setForeground(new java.awt.Color(153, 153, 153));

btnNewTask.setBackground(new java.awt.Color(255, 255, 255));

btnNewTask.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

btnNewTask.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/NewTask.png"))); // NOI18N

btnNewTask.setText("New Task");

btnNewTask.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnNewTaskActionPerformed(evt);

}

});

btnGetHelp.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

btnGetHelp.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Help.png"))); // NOI18N

btnGetHelp.setText("Get Help");

btnGetHelp.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnGetHelpActionPerformed(evt);

}

});

lblSortBy.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Sort.png"))); // NOI18N

lblSortBy.setText("Sort By");

javax.swing.GroupLayout pnlNewTaskLayout = new javax.swing.GroupLayout(pnlNewTask);

pnlNewTask.setLayout(pnlNewTaskLayout);

pnlNewTaskLayout.setHorizontalGroup(

pnlNewTaskLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlNewTaskLayout.createSequentialGroup()

.addContainerGap()

.addComponent(btnNewTask)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(btnGetHelp)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblSortBy)

.addContainerGap())

);

pnlNewTaskLayout.setVerticalGroup(

pnlNewTaskLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlNewTaskLayout.createSequentialGroup()

.addContainerGap(12, Short.MAX\_VALUE)

.addGroup(pnlNewTaskLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(btnNewTask, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(btnGetHelp)

.addComponent(lblSortBy)))

);

javax.swing.GroupLayout pnlTaskViewHeadingLayout = new javax.swing.GroupLayout(pnlTaskViewHeading);

pnlTaskViewHeading.setLayout(pnlTaskViewHeadingLayout);

pnlTaskViewHeadingLayout.setHorizontalGroup(

pnlTaskViewHeadingLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlTaskViewHeadingLayout.createSequentialGroup()

.addGap(25, 25, 25)

.addComponent(lblTaskView)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(pnlNewTask, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pnlTaskViewHeadingLayout.setVerticalGroup(

pnlTaskViewHeadingLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlTaskViewHeadingLayout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblTaskView)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(pnlNewTask, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

tblTask.setAutoCreateRowSorter(true);

tblTask.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

}

));

tblTask.setAutoResizeMode(javax.swing.JTable.AUTO\_RESIZE\_NEXT\_COLUMN);

splTableScroll.setViewportView(tblTask);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(10, 10, 10)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(lblOMT, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(490, 490, 490)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(btnReturnHome, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(4, 4, 4)

.addComponent(lblHomePage, javax.swing.GroupLayout.PREFERRED\_SIZE, 116, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(lblUserName, javax.swing.GroupLayout.PREFERRED\_SIZE, 62, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(8, 8, 8)

.addComponent(lblShowUserName, javax.swing.GroupLayout.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addComponent(splTableScroll, javax.swing.GroupLayout.PREFERRED\_SIZE, 770, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(pnlTaskViewHeading, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(11, 11, 11)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(9, 9, 9)

.addComponent(lblOMT))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(btnReturnHome)

.addComponent(lblHomePage))

.addGap(23, 23, 23)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblUserName)

.addGroup(layout.createSequentialGroup()

.addGap(1, 1, 1)

.addComponent(lblShowUserName, javax.swing.GroupLayout.PREFERRED\_SIZE, 13, javax.swing.GroupLayout.PREFERRED\_SIZE)))))

.addGap(6, 6, 6)

.addComponent(pnlTaskViewHeading, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(7, 7, 7)

.addComponent(splTableScroll, javax.swing.GroupLayout.PREFERRED\_SIZE, 350, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void btnNewTaskActionPerformed(java.awt.event.ActionEvent evt) {

//Display Task Detail Screen

new TaskDetail().setVisible(true);

this.dispose();

}

private void btnReturnHomeActionPerformed(java.awt.event.ActionEvent evt) {

//Display application screen

new Application().setVisible(true);

this.dispose();

}

private void btnGetHelpActionPerformed(java.awt.event.ActionEvent evt) {

//Display Help screen if user closes help screen it returns to the Task List screen

new Help(this).setVisible(true);

this.dispose();

}

/\*\*

\* @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(TaskList.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(TaskList.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(TaskList.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(TaskList.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 TaskList().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btnGetHelp;

private javax.swing.JButton btnNewTask;

private javax.swing.JButton btnReturnHome;

private javax.swing.JLabel lblHomePage;

private javax.swing.JLabel lblOMT;

private javax.swing.JLabel lblShowUserName;

private javax.swing.JLabel lblSortBy;

private javax.swing.JLabel lblTaskView;

private javax.swing.JLabel lblUserName;

private javax.swing.JPanel pnlNewTask;

private javax.swing.JPanel pnlTaskViewHeading;

private javax.swing.JScrollPane splTableScroll;

private javax.swing.JTable tblTask;

// End of variables declaration

}

}

## Task Detail Screen

//Inge Senekal

package frontend;

import backend.\*;

import java.awt.Color;

import java.sql.\*;

import java.time.LocalDate;

import java.time.ZoneId;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.BorderFactory;

import javax.swing.ComboBoxModel;

import javax.swing.DefaultComboBoxModel;

/\*\*

\*

\* @author Inges

\*/

public class TaskDetail extends javax.swing.JFrame {

private boolean update;

private String id;

private String[] subjects;

private ComboBoxModel model;

private boolean errors;

/\*\*

\* Creates new form TaskDetail

\*/

public TaskDetail() {

subjects = new String[TaskManager.subjects.length];

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

subjects[i] = TaskManager.subjects[i].getSubjectName();

}

initComponents();

//Sets the background colour for the JFrame to white

getContentPane().setBackground(Color.white);

update = false;

//Removes the border around the button

btnHelp.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnHelp.setContentAreaFilled(false);

btnSubmit.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnSubmit.setContentAreaFilled(false);

}

/\*\*

\* Sets the variables to the labels on the screen by calling the methods

\* created in the Task and TaskManager class

\*

\* @param id (Displays the names of the task based of the Id

\*/

public TaskDetail(String id) {

this();

update = true;

this.id = id;

Task t = TaskManager.getTask(id);

txtTaskName.setText(t.getTaskName());

dprStartDate.setDate(java.sql.Date.valueOf(t.getTaskStartDate()));

dprEndDate.setDate(java.sql.Date.valueOf(t.getTaskEndDate()));

cmbRelatedSubject.setSelectedItem(t.getSubjName());

txtTimeAllocation.setText("" + t.getTaskTimeAllocation());

txaTaskDescription.setText(t.getTaskDescription());

}

/\*\*

\* 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() {

lblTaskName = new javax.swing.JLabel();

lblTimeAllocation = new javax.swing.JLabel();

lblStartDate = new javax.swing.JLabel();

lblRelatedSubject = new javax.swing.JLabel();

lblEndDate = new javax.swing.JLabel();

lblTaskDescription = new javax.swing.JLabel();

lblTDComment = new javax.swing.JLabel();

lblTAComment = new javax.swing.JLabel();

btnSubmit = new javax.swing.JButton();

dprStartDate = new org.jdesktop.swingx.JXDatePicker();

dprEndDate = new org.jdesktop.swingx.JXDatePicker();

pnlOMT = new javax.swing.JPanel();

lblTask = new javax.swing.JLabel();

btnHelp = new javax.swing.JButton();

txtTimeAllocation = new javax.swing.JTextField();

txtTaskName = new javax.swing.JTextField();

cmbRelatedSubject = new javax.swing.JComboBox<>();

splTaskDescription = new javax.swing.JScrollPane();

txaTaskDescription = new javax.swing.JTextArea();

lblErrorStartDate = new javax.swing.JLabel();

lblErrorEndDate = new javax.swing.JLabel();

lblErrorTaskName = new javax.swing.JLabel();

lblErrorRelatedSubject = new javax.swing.JLabel();

lblErrorTaskDescription = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Optimise My Time");

setMaximumSize(new java.awt.Dimension(676, 446));

setMinimumSize(new java.awt.Dimension(676, 446));

setResizable(false);

lblTaskName.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblTaskName.setText("Task Name:");

lblTimeAllocation.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblTimeAllocation.setText("Time Allocation:");

lblStartDate.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblStartDate.setText("Start Date:");

lblRelatedSubject.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblRelatedSubject.setText("Related Subject:");

lblEndDate.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblEndDate.setText("End Date:");

lblTaskDescription.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblTaskDescription.setText("Task Description:");

lblTDComment.setFont(new java.awt.Font("Calibri", 0, 8)); // NOI18N

lblTDComment.setText("Please enter minutes only ");

lblTAComment.setFont(new java.awt.Font("Calibri", 0, 11)); // NOI18N

lblTAComment.setText("Limited to 500 characters");

btnSubmit.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

btnSubmit.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Submit.png"))); // NOI18N

btnSubmit.setText("Submit");

btnSubmit.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnSubmitActionPerformed(evt);

}

});

pnlOMT.setBackground(new java.awt.Color(0, 204, 255));

lblTask.setBackground(new java.awt.Color(18, 125, 244));

lblTask.setFont(new java.awt.Font("Cambria", 1, 14)); // NOI18N

lblTask.setForeground(new java.awt.Color(255, 255, 255));

lblTask.setText("Task");

btnHelp.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

btnHelp.setForeground(new java.awt.Color(18, 125, 244));

btnHelp.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Help.png"))); // NOI18N

btnHelp.setText("Get Help");

btnHelp.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnHelpActionPerformed(evt);

}

});

javax.swing.GroupLayout pnlOMTLayout = new javax.swing.GroupLayout(pnlOMT);

pnlOMT.setLayout(pnlOMTLayout);

pnlOMTLayout.setHorizontalGroup(

pnlOMTLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlOMTLayout.createSequentialGroup()

.addContainerGap()

.addComponent(lblTask, javax.swing.GroupLayout.PREFERRED\_SIZE, 47, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(43, 43, 43)

.addComponent(btnHelp)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pnlOMTLayout.setVerticalGroup(

pnlOMTLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlOMTLayout.createSequentialGroup()

.addGroup(pnlOMTLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lblTask)

.addComponent(btnHelp, javax.swing.GroupLayout.PREFERRED\_SIZE, 18, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

cmbRelatedSubject.setModel(new DefaultComboBoxModel(subjects));

txaTaskDescription.setColumns(20);

txaTaskDescription.setLineWrap(true);

txaTaskDescription.setRows(5);

txaTaskDescription.setWrapStyleWord(true);

splTaskDescription.setViewportView(txaTaskDescription);

lblErrorStartDate.setFont(new java.awt.Font("Calibri", 0, 8)); // NOI18N

lblErrorEndDate.setFont(new java.awt.Font("Calibri", 0, 8)); // NOI18N

lblErrorTaskName.setFont(new java.awt.Font("Calibri", 0, 8)); // NOI18N

lblErrorRelatedSubject.setFont(new java.awt.Font("Calibri", 0, 8)); // NOI18N

lblErrorTaskDescription.setFont(new java.awt.Font("Calibri", 0, 8)); // NOI18N

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(pnlOMT, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(139, 139, 139)

.addComponent(lblTimeAllocation, javax.swing.GroupLayout.PREFERRED\_SIZE, 136, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(30, 30, 30)

.addComponent(txtTimeAllocation, javax.swing.GroupLayout.PREFERRED\_SIZE, 182, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(4, 4, 4)

.addComponent(lblTDComment))

.addGroup(layout.createSequentialGroup()

.addGap(384, 384, 384)

.addComponent(btnSubmit, javax.swing.GroupLayout.PREFERRED\_SIZE, 103, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(139, 139, 139)

.addComponent(lblTaskDescription, javax.swing.GroupLayout.PREFERRED\_SIZE, 136, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(30, 30, 30)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblTAComment, javax.swing.GroupLayout.PREFERRED\_SIZE, 136, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addComponent(splTaskDescription, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(lblErrorTaskDescription, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(139, 139, 139)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(lblRelatedSubject, javax.swing.GroupLayout.PREFERRED\_SIZE, 136, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(30, 30, 30)

.addComponent(cmbRelatedSubject, javax.swing.GroupLayout.PREFERRED\_SIZE, 182, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(lblErrorRelatedSubject, javax.swing.GroupLayout.PREFERRED\_SIZE, 109, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(lblTaskName, javax.swing.GroupLayout.PREFERRED\_SIZE, 136, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(30, 30, 30)

.addComponent(txtTaskName, javax.swing.GroupLayout.PREFERRED\_SIZE, 182, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(lblErrorTaskName, javax.swing.GroupLayout.PREFERRED\_SIZE, 109, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(lblStartDate, javax.swing.GroupLayout.PREFERRED\_SIZE, 136, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(30, 30, 30)

.addComponent(dprStartDate, javax.swing.GroupLayout.PREFERRED\_SIZE, 182, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(lblErrorStartDate, javax.swing.GroupLayout.PREFERRED\_SIZE, 109, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(lblEndDate, javax.swing.GroupLayout.PREFERRED\_SIZE, 136, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(30, 30, 30)

.addComponent(dprEndDate, javax.swing.GroupLayout.PREFERRED\_SIZE, 182, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(lblErrorEndDate, javax.swing.GroupLayout.PREFERRED\_SIZE, 109, javax.swing.GroupLayout.PREFERRED\_SIZE)))))

.addGap(0, 62, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(pnlOMT, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(22, 22, 22)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblErrorTaskName, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lblTaskName, javax.swing.GroupLayout.PREFERRED\_SIZE, 22, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(txtTaskName, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(3, 3, 3)

.addComponent(lblRelatedSubject))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(cmbRelatedSubject, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblErrorRelatedSubject, javax.swing.GroupLayout.PREFERRED\_SIZE, 22, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(20, 20, 20)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblStartDate)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(dprStartDate, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblErrorStartDate, javax.swing.GroupLayout.PREFERRED\_SIZE, 22, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblEndDate)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(dprEndDate, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblErrorEndDate, javax.swing.GroupLayout.PREFERRED\_SIZE, 22, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblTimeAllocation)

.addComponent(txtTimeAllocation, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addGap(2, 2, 2)

.addComponent(lblTDComment, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(14, 14, 14)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblTaskDescription)

.addGroup(layout.createSequentialGroup()

.addGap(2, 2, 2)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblErrorTaskDescription, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(splTaskDescription, javax.swing.GroupLayout.PREFERRED\_SIZE, 127, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lblTAComment)

.addGap(18, 18, 18)

.addComponent(btnSubmit))

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

//Buttons performs an update or a insert

//It calls the method created in the DB class and the created SQL statement allows for the records to be updated

//If the user want to enter a completely new task the SQL statement Insert Into will be used.

private void btnSubmitActionPerformed(java.awt.event.ActionEvent evt) {

errors = false;

if (txtTimeAllocation.getText().length() > 0) {

//Data validation

try {

double time = Double.parseDouble(txtTimeAllocation.getText());

} catch (NumberFormatException ex) {

errors = true;

lblTDComment.setText("The time entered is not a number.");

}

}

if (txtTaskName.getText().length() <= 0) {

errors = true;

lblErrorTaskName.setText("The Task Name must be filled");

}

if (txaTaskDescription.getText().length() <= 0) {

errors = true;

lblErrorTaskDescription.setText("The Task Description must be filled");

}

if (cmbRelatedSubject.getSelectedIndex() < 0) {

errors = true;

lblErrorRelatedSubject.setText("Subject must be choosen");

}

if (dprStartDate.getDate() == null) {

errors = true;

lblErrorStartDate.setText("A date must be selected ");

}

if (dprEndDate.getDate() == null) {

errors = true;

lblErrorEndDate.setText("A date must be selected ");

}

if (txaTaskDescription.getText().length() <= 0) {

errors = true;

lblErrorTaskDescription.setText("There must be a task description.");

}

if (txaTaskDescription.getText().length() > 500) {

errors = true;

lblErrorTaskDescription.setText("The task description should not be more than 500 words.");

}

if (errors == false && dprEndDate.getDate() != null && dprStartDate.getDate() != null && txtTaskName.getText() != null && txtTimeAllocation.getText() != null && txaTaskDescription.getText() != null && cmbRelatedSubject.getSelectedIndex() > -1) {

try {

if (update == true) {

LocalDate start = dprStartDate.getDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate();

LocalDate end = dprEndDate.getDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate();

TaskManager.db.update("UPDATE tblTask SET TaskName = '" + txtTaskName.getText() + "' WHERE TaskID = " + id + ";");

TaskManager.db.update("UPDATE tblTask SET SubjectID = " + (cmbRelatedSubject.getSelectedIndex() + 1) + " WHERE TaskID = " + id + ";");

TaskManager.db.update("UPDATE tblTask SET StartDate = '" + start.toString() + "' WHERE TaskID = " + id + ";");

TaskManager.db.update("UPDATE tblTask SET EndDate = '" + end.toString() + "' WHERE TaskID = " + id + ";");

TaskManager.db.update("UPDATE tblTask SET TaskName = '" + txtTaskName.getText() + "' WHERE TaskID = " + id + ";");

TaskManager.db.update("UPDATE tblTask SET TimeAllocation = " + txtTimeAllocation.getText() + " WHERE TaskID = " + id + ";");

TaskManager.db.update("UPDATE tblTask SET Description = '" + txaTaskDescription.getText() + "' WHERE TaskID = " + id + ";");

} else {

LocalDate start = dprStartDate.getDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate();

LocalDate end = dprEndDate.getDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate();

System.out.println("Successfully added task");

TaskManager.db.update("INSERT INTO tblTask(TaskName, UserID, StatusID, SubjectID, SubjPriorityID, CreateDate, StartDate, EndDate, Complete, TimeAllocation, Description) "

+ "VALUES('" + txtTaskName.getText() + "'," + UserManager.user.getUserID() + ",2," + (cmbRelatedSubject.getSelectedIndex() + 1) + ",1,'" + LocalDate.now().toString() + "','" + start.toString() + "','" + end

+ "','No','" + txtTimeAllocation.getText() + "','" + txaTaskDescription.getText() + "');");

}

} catch (SQLException ex) {

Logger.getLogger(TaskDetail.class.getName()).log(Level.SEVERE, null, ex);

}

}

if (errors == false) {

new TaskList().setVisible(true);

this.dispose();

}

}

//Opens the Help Screen returns to this screen ones closed

private void btnHelpActionPerformed(java.awt.event.ActionEvent evt) {

new Help(this).setVisible(true);

this.dispose();

}

// Variables declaration - do not modify

private javax.swing.JButton btnHelp;

private javax.swing.JButton btnSubmit;

private javax.swing.JComboBox<String> cmbRelatedSubject;

private org.jdesktop.swingx.JXDatePicker dprEndDate;

private org.jdesktop.swingx.JXDatePicker dprStartDate;

private javax.swing.JLabel lblEndDate;

private javax.swing.JLabel lblErrorEndDate;

private javax.swing.JLabel lblErrorRelatedSubject;

private javax.swing.JLabel lblErrorStartDate;

private javax.swing.JLabel lblErrorTaskDescription;

private javax.swing.JLabel lblErrorTaskName;

private javax.swing.JLabel lblRelatedSubject;

private javax.swing.JLabel lblStartDate;

private javax.swing.JLabel lblTAComment;

private javax.swing.JLabel lblTDComment;

private javax.swing.JLabel lblTask;

private javax.swing.JLabel lblTaskDescription;

private javax.swing.JLabel lblTaskName;

private javax.swing.JLabel lblTimeAllocation;

private javax.swing.JPanel pnlOMT;

private javax.swing.JScrollPane splTaskDescription;

private javax.swing.JTextArea txaTaskDescription;

private javax.swing.JTextField txtTaskName;

private javax.swing.JTextField txtTimeAllocation;

// End of variables declaration

}

## Planner Screen

//Inge Senekal

package frontend;

import backend.UserManager;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.time.LocalDate;

import javax.swing.BorderFactory;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableModel;

/\*\*

\*

\* @author Inges

\*/

public class Planner extends javax.swing.JFrame {

private TableModel tblCurrentMonth;

private LocalDate curr;

private final String[] DAYS\_OF\_THE\_WEEK = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};

private Object[][] days;

private boolean clicked = false;

/\*\*

\* Creates new form Planner

\*/

public Planner() {

initComponents();

//Sets the background colour for the JFrame to white.

getContentPane().setBackground(Color.white);

//Removes the border around the buttons.

btnForwardMonth.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnForwardMonth.setContentAreaFilled(false);

btnBackMonth.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnBackMonth.setContentAreaFilled(false);

btnHomeScreen.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnHomeScreen.setContentAreaFilled(false);

btnHelp.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnHelp.setContentAreaFilled(false);

//Gives the label the text of the user name that currently uses the program.

lblDisplayUserName.setText(UserManager.user.getUserName());

revalidate();

curr = LocalDate.now();

updateMonth();

}

//This updates the Month of the calendar

//The tables is set up as a calendar with the 5 rows and 7 columbs

//Each cell in the tables is given a number based of the day in the month

//Leap years are accomidated for

//COntains a mouse clicker, if the user clicks on a cell it will move the user to the Planner Detail Screen.

private void updateMonth() {

LocalDate currMonth = LocalDate.of(curr.getYear(), curr.getMonth(), 1);

days = new Object[5][7];

lblMonth.setText(currMonth.getMonth().name() + " " + currMonth.getYear());

int start = -1;

int counter = 1;

int max = currMonth.getMonth().minLength();

if (currMonth.isLeapYear()) {

max = currMonth.getMonth().maxLength();

}

switch (currMonth.getDayOfWeek()) {

case SUNDAY:

start = 0;

break;

case MONDAY:

start = 1;

break;

case TUESDAY:

start = 2;

break;

case WEDNESDAY:

start = 3;

break;

case THURSDAY:

start = 4;

break;

case FRIDAY:

start = 5;

break;

case SATURDAY:

start = 6;

break;

}

for (int i = 0;

i < 5; i++) {

if (i > 0) {

start = 0;

}

for (int j = start; j < 7; j++) {

if (counter <= max) {

days[i][j] = counter;

counter++;

} else {

days[i][j] = null;

}

}

}

tblCurrentMonth = new DefaultTableModel(days, DAYS\_OF\_THE\_WEEK);

tblCalendar.setModel(tblCurrentMonth);

tblCalendar.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent evt) {

int column = tblCalendar.getColumnModel().getColumnIndexAtX(evt.getX());

int row = evt.getY() / tblCalendar.getRowHeight();

goToPlannerDetails(row, column);

}

});

}

//When any cell is click on on the Calendar it will move the user to the Planner Detail Screen

private void goToPlannerDetails(int x, int y) {

if (days[x][y] == null) {

//System.out.println("Nothing");

} else {

if (clicked == false) {

clicked = true;

LocalDate ld = LocalDate.of(curr.getYear(), curr.getMonthValue(), (int) days[x][y]);

new PlannerDetail(ld).setVisible(true);

this.dispose();

}

}

}

private void btnDaysActionPerformed(ActionEvent evt) {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

/\*\*

\* 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() {

srpCalendar = new javax.swing.JScrollPane();

tblCalendar = new javax.swing.JTable();

pnlCalendarHelp = new javax.swing.JPanel();

btnBackMonth = new javax.swing.JButton();

lblCalendarIcon = new javax.swing.JLabel();

btnForwardMonth = new javax.swing.JButton();

lblMonth = new javax.swing.JLabel();

btnHelp = new javax.swing.JButton();

lblHelp = new javax.swing.JLabel();

pnlCalendarMonthView = new javax.swing.JPanel();

lblCalendarMonthView = new javax.swing.JLabel();

lblIcon = new javax.swing.JLabel();

lblReturnHome = new javax.swing.JLabel();

btnHomeScreen = new javax.swing.JButton();

lblUserName = new javax.swing.JLabel();

lblDisplayUserName = new javax.swing.JLabel();

pnlBlue = new javax.swing.JPanel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Optimise My Time");

setMaximumSize(new java.awt.Dimension(700, 580));

setMinimumSize(new java.awt.Dimension(700, 580));

setPreferredSize(new java.awt.Dimension(700, 580));

setResizable(false);

tblCalendar.setFont(new java.awt.Font("Calibri", 1, 12)); // NOI18N

tblCalendar.setAutoResizeMode(javax.swing.JTable.AUTO\_RESIZE\_ALL\_COLUMNS);

tblCalendar.setRowHeight(80);

srpCalendar.setViewportView(tblCalendar);

pnlCalendarHelp.setBackground(new java.awt.Color(153, 153, 153));

btnBackMonth.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Left.png"))); // NOI18N

btnBackMonth.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnBackMonthActionPerformed(evt);

}

});

lblCalendarIcon.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Calen.png"))); // NOI18N

btnForwardMonth.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Right.png"))); // NOI18N

btnForwardMonth.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnForwardMonthActionPerformed(evt);

}

});

lblMonth.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

lblMonth.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

btnHelp.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Help.png"))); // NOI18N

btnHelp.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnHelpActionPerformed(evt);

}

});

lblHelp.setText("Get Help");

javax.swing.GroupLayout pnlCalendarHelpLayout = new javax.swing.GroupLayout(pnlCalendarHelp);

pnlCalendarHelp.setLayout(pnlCalendarHelpLayout);

pnlCalendarHelpLayout.setHorizontalGroup(

pnlCalendarHelpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlCalendarHelpLayout.createSequentialGroup()

.addContainerGap()

.addComponent(btnBackMonth, javax.swing.GroupLayout.PREFERRED\_SIZE, 22, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lblCalendarIcon, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(btnForwardMonth, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lblMonth, javax.swing.GroupLayout.PREFERRED\_SIZE, 103, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(64, 64, 64)

.addComponent(btnHelp, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lblHelp)

.addContainerGap(411, Short.MAX\_VALUE))

);

pnlCalendarHelpLayout.setVerticalGroup(

pnlCalendarHelpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, pnlCalendarHelpLayout.createSequentialGroup()

.addGroup(pnlCalendarHelpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, pnlCalendarHelpLayout.createSequentialGroup()

.addGroup(pnlCalendarHelpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlCalendarHelpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(btnBackMonth, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblCalendarIcon, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(btnForwardMonth, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(lblMonth, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(0, 0, Short.MAX\_VALUE))

.addComponent(lblHelp, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(btnHelp, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap())

);

pnlCalendarMonthView.setBackground(new java.awt.Color(0, 204, 255));

lblCalendarMonthView.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblCalendarMonthView.setForeground(new java.awt.Color(255, 255, 255));

lblCalendarMonthView.setText("Calendar Month View ");

javax.swing.GroupLayout pnlCalendarMonthViewLayout = new javax.swing.GroupLayout(pnlCalendarMonthView);

pnlCalendarMonthView.setLayout(pnlCalendarMonthViewLayout);

pnlCalendarMonthViewLayout.setHorizontalGroup(

pnlCalendarMonthViewLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlCalendarMonthViewLayout.createSequentialGroup()

.addContainerGap()

.addComponent(lblCalendarMonthView)

.addContainerGap(603, Short.MAX\_VALUE))

);

pnlCalendarMonthViewLayout.setVerticalGroup(

pnlCalendarMonthViewLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlCalendarMonthViewLayout.createSequentialGroup()

.addComponent(lblCalendarMonthView)

.addGap(0, 12, Short.MAX\_VALUE))

);

lblIcon.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/OMT\_1.PNG"))); // NOI18N

lblReturnHome.setText("Return to Home Screen");

btnHomeScreen.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/HomePage.png"))); // NOI18N

btnHomeScreen.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnHomeScreenActionPerformed(evt);

}

});

lblUserName.setFont(new java.awt.Font("Calibri", 1, 12)); // NOI18N

lblUserName.setForeground(new java.awt.Color(18, 125, 244));

lblUserName.setText("User Name");

pnlBlue.setBackground(new java.awt.Color(0, 204, 255));

javax.swing.GroupLayout pnlBlueLayout = new javax.swing.GroupLayout(pnlBlue);

pnlBlue.setLayout(pnlBlueLayout);

pnlBlueLayout.setHorizontalGroup(

pnlBlueLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 700, Short.MAX\_VALUE)

);

pnlBlueLayout.setVerticalGroup(

pnlBlueLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 30, Short.MAX\_VALUE)

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(440, 440, 440)

.addComponent(lblUserName, javax.swing.GroupLayout.PREFERRED\_SIZE, 60, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(440, 440, 440)

.addComponent(btnHomeScreen, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(460, 460, 460)

.addComponent(lblReturnHome, javax.swing.GroupLayout.PREFERRED\_SIZE, 130, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(10, 10, 10)

.addComponent(lblIcon))

.addComponent(pnlCalendarMonthView, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addGap(510, 510, 510)

.addComponent(lblDisplayUserName, javax.swing.GroupLayout.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(pnlCalendarHelp, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(srpCalendar, javax.swing.GroupLayout.PREFERRED\_SIZE, 700, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(pnlBlue, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(30, 30, 30)

.addComponent(lblUserName))

.addComponent(btnHomeScreen)

.addComponent(lblReturnHome, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblIcon)

.addGroup(layout.createSequentialGroup()

.addGap(50, 50, 50)

.addComponent(pnlCalendarMonthView, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(30, 30, 30)

.addComponent(lblDisplayUserName, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addComponent(pnlCalendarHelp, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(srpCalendar, javax.swing.GroupLayout.PREFERRED\_SIZE, 430, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(pnlBlue, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

layout.linkSize(javax.swing.SwingConstants.VERTICAL, new java.awt.Component[] {lblDisplayUserName, lblUserName});

getAccessibleContext().setAccessibleName("Optimise My time");

pack();

setLocationRelativeTo(null);

}// </editor-fold>

//Methods used for a button

//This allows for the user to move forward in the year based of the months

private void btnForwardMonthActionPerformed(java.awt.event.ActionEvent evt) {

int year = curr.getYear();

int month = curr.getMonthValue() + 1;

if (month == 13) {

month = 1;

year++;

}

curr = LocalDate.of(year, month, 1);

updateMonth();

}

//Methods used for a button

//This allows for the user to move back in the year based of the months

private void btnBackMonthActionPerformed(java.awt.event.ActionEvent evt) {

int year = curr.getYear();

int month = curr.getMonthValue() - 1;

if (month == 0) {

month = 12;

year--;

}

curr = LocalDate.of(year, month, 1);

updateMonth();

}

//Opens the Appplication Screen

private void btnHomeScreenActionPerformed(java.awt.event.ActionEvent evt) {

new Application().setVisible(true);

this.dispose();

}

//Opens the Help Screen returns to this screen ones closed

private void btnHelpActionPerformed(java.awt.event.ActionEvent evt) {

new Help(this).setVisible(true);

this.dispose();

}

/\*\*

\* @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(Planner.class

.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Planner.class

.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Planner.class

.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Planner.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 Planner().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btnBackMonth;

private javax.swing.JButton btnForwardMonth;

private javax.swing.JButton btnHelp;

private javax.swing.JButton btnHomeScreen;

private javax.swing.JLabel lblCalendarIcon;

private javax.swing.JLabel lblCalendarMonthView;

private javax.swing.JLabel lblDisplayUserName;

private javax.swing.JLabel lblHelp;

private javax.swing.JLabel lblIcon;

private javax.swing.JLabel lblMonth;

private javax.swing.JLabel lblReturnHome;

private javax.swing.JLabel lblUserName;

private javax.swing.JPanel pnlBlue;

private javax.swing.JPanel pnlCalendarHelp;

private javax.swing.JPanel pnlCalendarMonthView;

private javax.swing.JScrollPane srpCalendar;

private javax.swing.JTable tblCalendar;

// End of variables declaration

}

## Planner Detail Screen

//Inge Senekal

package frontend;

import backend.Task;

import backend.TaskManager;

import java.awt.Color;

import java.time.LocalDate;

import javax.swing.BorderFactory;

/\*\*

\*

\* @author Inges

\*/

public class PlannerDetail extends javax.swing.JFrame {

private boolean update;

/\*\*

\* Creates new form PlannerDetail

\*

\* @param ld

\*/

public PlannerDetail(LocalDate ld) {

//Creates an object of the Task class and assigns the values of the detailTask method of the TaskManager class

Task task = TaskManager.detailTask(ld);

initComponents();

//If there is no task it must display a blank screen

if (task != null) {

lblDetailEndDate.setText(task.getTaskEndDate().toString());

lblDetailRelatedSubject.setText(task.getSubjName());

lblDetailTaskDescription.setText(task.getTaskDescription());

lblDetailTaskName.setText(task.getTaskName());

lblDetailTimeAllocation.setText("" + task.getTaskTimeAllocation());

lblDetailStartDate.setText(task.getTaskStartDate().toString());

}

update = false;

//Sets the background colour for the JFrame to white

getContentPane().setBackground(Color.white);

//Removes the border around the buttons.

btnHelp.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnHelp.setContentAreaFilled(false);

btnClose.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnClose.setContentAreaFilled(false);

}

/\*\*

\* Creates an object of the task class and assigns values from the getTask method in the Task Manager Class.

\* @param id (Gets the task based on the taskID)

\* When the user opens the screen the task and its information will be displayed in the labels of the screen.

\*/

public PlannerDetail(String id) {

update = true;

Task t = TaskManager.getTask(id);

lblDetailTaskName.setText(t.getTaskName());

lblRelatedSubject.setText(t.getSubjName());

lblDetailStartDate.setText(t.getTaskStartDate().toString());

lblDetailEndDate.setText(t.getTaskEndDate().toString());

lblDetailTimeAllocation.setText("" + t.getTaskTimeAllocation());

lblDetailTaskDescription.setText("" + t.getTaskDescription());

}

/\*\*

\* 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() {

pnlTaskDetials = new javax.swing.JPanel();

lblTask = new javax.swing.JLabel();

btnHelp = new javax.swing.JButton();

lblOMT = new javax.swing.JLabel();

lblTaskName = new javax.swing.JLabel();

lblTimeAllocation = new javax.swing.JLabel();

lblStartDate = new javax.swing.JLabel();

lblRelatedSubject = new javax.swing.JLabel();

lblEndDate = new javax.swing.JLabel();

btnClose = new javax.swing.JButton();

lblTaskDescription1 = new javax.swing.JLabel();

lblDetailTaskName = new javax.swing.JLabel();

lblDetailRelatedSubject = new javax.swing.JLabel();

lblDetailStartDate = new javax.swing.JLabel();

lblDetailEndDate = new javax.swing.JLabel();

lblDetailTimeAllocation = new javax.swing.JLabel();

lblDetailTaskDescription = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

pnlTaskDetials.setBackground(new java.awt.Color(0, 204, 255));

lblTask.setBackground(new java.awt.Color(18, 125, 244));

lblTask.setFont(new java.awt.Font("Cambria", 1, 14)); // NOI18N

lblTask.setForeground(new java.awt.Color(255, 255, 255));

lblTask.setText("Task Details");

btnHelp.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

btnHelp.setForeground(new java.awt.Color(18, 125, 244));

btnHelp.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Help.png"))); // NOI18N

btnHelp.setText("Get Help");

btnHelp.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnHelpActionPerformed(evt);

}

});

javax.swing.GroupLayout pnlTaskDetialsLayout = new javax.swing.GroupLayout(pnlTaskDetials);

pnlTaskDetials.setLayout(pnlTaskDetialsLayout);

pnlTaskDetialsLayout.setHorizontalGroup(

pnlTaskDetialsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlTaskDetialsLayout.createSequentialGroup()

.addContainerGap()

.addComponent(lblTask)

.addGap(18, 18, 18)

.addComponent(btnHelp)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pnlTaskDetialsLayout.setVerticalGroup(

pnlTaskDetialsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlTaskDetialsLayout.createSequentialGroup()

.addGroup(pnlTaskDetialsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lblTask)

.addComponent(btnHelp, javax.swing.GroupLayout.PREFERRED\_SIZE, 18, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

lblOMT.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/OMT\_1.PNG"))); // NOI18N

lblTaskName.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblTaskName.setText("Task Name:");

lblTimeAllocation.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblTimeAllocation.setText("Time Allocation:");

lblStartDate.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblStartDate.setText("Start Date:");

lblRelatedSubject.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblRelatedSubject.setText("Related Subject:");

lblEndDate.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblEndDate.setText("End Date:");

btnClose.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

btnClose.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/Close.png"))); // NOI18N

btnClose.setText("Close");

btnClose.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnCloseActionPerformed(evt);

}

});

lblTaskDescription1.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblTaskDescription1.setText("Task Description:");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(pnlTaskDetials, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(btnClose, javax.swing.GroupLayout.PREFERRED\_SIZE, 103, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGap(135, 135, 135)

.addComponent(lblOMT, javax.swing.GroupLayout.PREFERRED\_SIZE, 102, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 157, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGap(43, 43, 43)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(lblRelatedSubject, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblTaskName, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblStartDate, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblEndDate, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblTimeAllocation, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblTaskDescription1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(88, 88, 88)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblDetailTimeAllocation, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblDetailStartDate, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblDetailRelatedSubject, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblDetailTaskName, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblDetailEndDate, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblDetailTaskDescription, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))))

.addGap(19, 19, 19))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(lblOMT)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(pnlTaskDetials, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(26, 26, 26)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblTaskName)

.addComponent(lblDetailTaskName, javax.swing.GroupLayout.PREFERRED\_SIZE, 18, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblRelatedSubject)

.addComponent(lblDetailRelatedSubject, javax.swing.GroupLayout.PREFERRED\_SIZE, 18, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblStartDate)

.addComponent(lblDetailStartDate, javax.swing.GroupLayout.PREFERRED\_SIZE, 18, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(24, 24, 24)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(lblEndDate, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblDetailEndDate, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(lblTimeAllocation, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblDetailTimeAllocation, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblTaskDescription1)

.addComponent(lblDetailTaskDescription, javax.swing.GroupLayout.PREFERRED\_SIZE, 95, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(48, 48, 48)

.addComponent(btnClose, javax.swing.GroupLayout.PREFERRED\_SIZE, 29, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(16, Short.MAX\_VALUE))

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void btnHelpActionPerformed(java.awt.event.ActionEvent evt) {

//Display Help screen if user closes help screen it returns to the Planner Detail screen

new Help(this).setVisible(true);

this.dispose();

}

private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {

//Once the user closes the Planner detail screen by clicking on this button it will return to the planner screen

new Planner().setVisible(true);

this.setVisible(false);

this.dispose();

}

// Variables declaration - do not modify

private javax.swing.JButton btnClose;

private javax.swing.JButton btnHelp;

private javax.swing.JLabel lblDetailEndDate;

private javax.swing.JLabel lblDetailRelatedSubject;

private javax.swing.JLabel lblDetailStartDate;

private javax.swing.JLabel lblDetailTaskDescription;

private javax.swing.JLabel lblDetailTaskName;

private javax.swing.JLabel lblDetailTimeAllocation;

private javax.swing.JLabel lblEndDate;

private javax.swing.JLabel lblOMT;

private javax.swing.JLabel lblRelatedSubject;

private javax.swing.JLabel lblStartDate;

private javax.swing.JLabel lblTask;

private javax.swing.JLabel lblTaskDescription1;

private javax.swing.JLabel lblTaskName;

private javax.swing.JLabel lblTimeAllocation;

private javax.swing.JPanel pnlTaskDetials;

// End of variables declaration

}

## Task

//Inge Senekal

package backend;

import java.time.LocalDate;

/\*\*

\*

\* @author Inges

\*/

public class Task {

private int taskID;

private String taskName;

private LocalDate taskCreateDate;

private LocalDate taskStartDate;

private LocalDate taskEndDate;

private String taskComplete;

private int taskTimeAllocation;

/\*\*

\*Accessor method gets the unique identifier for the task

\* returns the integer value of the Task ID

\*/

public int getTaskID() {

return taskID;

}

/\*\*

\* Mutator method updates the unique identifier of the task

\* Used to change the taskID

\*/

public void setTaskID(int taskID) {

this.taskID = taskID;

}

private String taskDescription;

private int subjPriority;

private String subjName;

private String taskStatus;

/\*\*

\* Constructor method instantiates the task object

\* Parameters found inside the constructor method

\* @param TaskID unique identifier of the task

\* @param taskName the name of the task

\* @param taskCreateDate the date the task was created

\* @param taskStartDate the date the task should start

\* @param taskEndDate the date the task must be done

\* @param taskComplete whether the task is complete or not

\* @param taskTimeAllocation the time allocation for task

\* @param taskDescription the description of the task

\* @param subjPriority the priority of the subject

\* @param subjName the name of the subject

\* @param taskStatus the status of the task

\*/

public Task(int taskID, String taskName, String taskCreateDate, String taskStartDate, String taskEndDate, String taskComplete, int taskTimeAllocation, String taskDescription, int subjPriority, String subjName, String taskStatus) {

this.taskID = taskID;

this.taskName = taskName;

this.taskCreateDate = LocalDate.parse(taskCreateDate);

this.taskStartDate = LocalDate.parse(taskStartDate);

this.taskEndDate = LocalDate.parse(taskEndDate);

this.taskComplete = taskComplete;

this.taskTimeAllocation = taskTimeAllocation;

this.taskDescription = taskDescription;

this.subjPriority = subjPriority;

this.subjName = subjName;

this.taskStatus = taskStatus;

}

/\*\*

\*Accessor method gets the name of the task

\* returns the string value of the task name

\*/

public String getTaskName() {

return taskName;

}

/\*\*

\*Accessor method gets whether the task is completed

\* returns whether or not the task is completed

\*/

public String getTaskComplete() {

return taskComplete;

}

/\*\*

\*Accessor method get the time allocation of the task

\* returns a string value of the time allocation

\*/

public int getTimeAllocation() {

return taskTimeAllocation;

}

/\*\*

\*Accessor method gets the description of the task

\* returns a string value of the task description

\*/

public String getTaskDescription() {

return taskDescription;

}

/\*\*

\*Accessor method gets the priority of the subject

\* returns a string value of the subject priority

\*/

public int getSubjPriority() {

return subjPriority;

}

/\*\*

\*Accessor method get the name of the subject

\* returns a string value of the subject name

\*/

public String getSubjName() {

return subjName;

}

/\*\*

\*Accessor method gets the status of the task

\* returns a string value of the status

\*/

public String getTaskStatus() {

return taskStatus;

}

/\*\*

\*Mutator method updates the tasks name

\* Used to change the task name

\*/

public void setTaskName(String taskname) {

taskName = taskname;

}

/\*\*

\*Accessor method gets the create date of the task

\* returns a localdate value as the date task was created

\*/

public LocalDate getTaskCreateDate() {

return taskCreateDate;

}

/\*\*

\*Mutator method update the tasks create date

\* Used to change the date the task was created

\*/

public void setTaskCreateDate(LocalDate taskCreateDate) {

this.taskCreateDate = taskCreateDate;

}

/\*\*

\*Accessor method gets the start date of the task

\* returns a string value as the date the task started

\*/

public LocalDate getTaskStartDate() {

return taskStartDate;

}

/\*\*

\*Mutator method update the tasks start date

\* Used to change the date of when the task started

\*/

public void setTaskStartDate(LocalDate taskStartDate) {

this.taskStartDate = taskStartDate;

}

/\*\*

\*Accessor method gets the end date of the task

\* returns a string value as the date the task ends

\*/

public LocalDate getTaskEndDate() {

return taskEndDate;

}

/\*\*

\*Mutator method updates the tasks end date

\* Used to change the date the task has to end

\*/

public void setTaskEndDate(LocalDate taskEndDate) {

this.taskEndDate = taskEndDate;

}

/\*\*

\*Accessor method gets the time allocation of the task

\* returns an integer value of time allocation of the task

\*/

public int getTaskTimeAllocation() {

return taskTimeAllocation;

}

/\*\*

\*Mutator method update the time allocation for the task

\* Used to change the time allocation of the task

\*/

public void setTaskTimeAllocation(int taskTimeAllocation) {

this.taskTimeAllocation = taskTimeAllocation;

}

/\*\*

\*Mutator method updates time allocation field

\* Used to change whether the task is completed or not

\*/

public void setTaskComplete(String complete) {

taskComplete = complete;

}

/\*\*

\*Mutator method update the time allocation of the task

\* Used to change the time allocation of the task

\*/

public void setTimeAllocation(int timeallocation) {

taskTimeAllocation = timeallocation;

}

/\*\*

\*Mutator method update the description of the task

\* Used to change the description of the task

\*/

public void setTaskDescription(String description) {

taskDescription = description;

}

/\*\*

\*Mutator method update the subjects priority

\* Used to change the subject priority

\*/

public void setSubjPriority(int subjectpriority) {

subjPriority = subjectpriority;

}

/\*\*

\*Mutator method updates the name of the subject

\* Used to change the the subject name

\*/

public void setSubjName(String subjectname) {

subjName = subjectname;

}

/\*\*

\*Mutator method update the task status

\* Used to change the status of the task

\*/

public void setTaskStatus(String status) {

taskStatus = status;

}

}

## Subject

//Inge Senekal

package backend;

/\*\*

\*

\* @author Inges

\*/

public class Subject {

//Declair subjID as an integer

//Declair subjectName as a String

private String subjectName;

private int subjID;

/\*\*

\*Constructor method instantiates the subject’s object

\* @param subjectName the name of the subject

\* @param subjID unique identifier for the subject

\*/

public Subject(String subjectName, int subjID) {

this.subjectName = subjectName;

this.subjID = subjID;

}

/\*\*

\*Accessor method gets the name of the subject

\* @param subjectName

\* @return the string value of the subjects Name

\*/

public String getSubjectName() {

return subjectName;

}

/\*\*

\*Mutator method updates the name of the subject

\* Used to change the name of the subject

\* @param subjectName

\*/

public void setSubjectName(String subjectName) {

this.subjectName = subjectName;

}

/\*\*

\*Accessor method gets the unique identifier of the subject

\* @param subjID

\* returns a integer value for the subjects ID

\*/

public int getSubjID() {

return subjID;

}

/\*\*

\*Mutator method updates the subject ID

\* Used to change the ID of the subject

\* @param subjId

\*/

public void setSubjID(int subjID) {

this.subjID = subjID;

}

}

## TaskManager

//Inge Senekal

package backend;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author Inges

\*/

public class TaskManager {

/\*\*

\* Creates a object of the Task class called task

\*/

public static Task task;

/\*\*

\* Create an object of the DB class which can not be changed

\*/

public static final DB db = new DB();

/\*\*

\* Creates a array of the Subject class called subjects

\*/

public static Subject[] subjects;

/\*\*

\* Searches for a task based on taskID.

\* @param id (the id of the task being searched for)

\* @return the task and its information

\*/

public static Task getTask(String id) {

try {

ResultSet rs = db.query("SELECT \* FROM tblTask WHERE TaskID = " + id + ";");

if (rs.next()) {

ResultSet subjects = db.query("SELECT Subject FROM tblSubject WHERE SubjectID =" + rs.getInt("SubjectID") + ";");

ResultSet ts = db.query("SELECT Status FROM tblTaskStatus WHERE StatusID = " + rs.getInt("StatusID") + ";");

subjects.next();

ts.next();

Task t = new Task(rs.getInt("TaskID"), rs.getString("TaskName"), rs.getString("CreateDate"), rs.getString("StartDate"), rs.getString("EndDate"),

rs.getString("Complete"), rs.getInt("TimeAllocation"), rs.getString("Description"), rs.getInt("SubjPriorityID"),

subjects.getString("Subject"), ts.getString("Status"));

return t;

}

} catch (SQLException ex) {

Logger.getLogger(TaskManager.class.getName()).log(Level.SEVERE, null, ex);

}

return null;

}

/\*\*

\* Searches and compiles and array of tasks associated with the user.

\* @return the array of tasks associated with the user ID

\*/

public static Task[] getTasks() {

ArrayList<Task> list = new ArrayList<>();

try {

ResultSet rs = db.query("Select \* From tblTask WHERE UserID = " + UserManager.user.getUserID() + ";");

while (rs.next()) {

ResultSet subjects = db.query("SELECT Subject FROM tblSubject WHERE SubjectID =" + rs.getInt("SubjectID") + ";");

ResultSet ts = db.query("SELECT Status FROM tblTaskStatus WHERE StatusID = " + rs.getInt("StatusID") + ";");

subjects.next();

ts.next();

list.add(new Task(rs.getInt("TaskID"), rs.getString("TaskName"), rs.getString("CreateDate"), rs.getString("StartDate"), rs.getString("EndDate"),

rs.getString("Complete"), rs.getInt("TimeAllocation"), rs.getString("Description"), rs.getInt("SubjPriorityID"),

subjects.getString("Subject"), ts.getString("Status")));

}

} catch (SQLException ex) {

Logger.getLogger(TaskManager.class.getName()).log(Level.SEVERE, null, ex);

}

Task[] tasks = new Task[list.size()];

for (int i = 0; i < list.size(); i++) {

tasks[i] = list.get(i);

}

return tasks;

}

/\*\*

\* Searches for the subject in the database and creates an array containing all the subjects

\*/

public static void populateSubjects() {

try {

ResultSet rs = db.query("SELECT \* FROM tblSubject;");

ArrayList<Subject> al = new ArrayList<>();

while (rs.next()) {

al.add(new Subject(rs.getString("Subject"), rs.getInt("SubjectID")));

}

subjects = new Subject[al.size()];

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

subjects[i] = al.get(i);

}

} catch (SQLException ex) {

Logger.getLogger(TaskManager.class.getName()).log(Level.SEVERE, null, ex);

}

}

/\*\*

\* Searches for a task based on taskID

\* Delete the task based on the searched for taskID

\* @param id (the id of the task being deleted)

\*/

public static void deleteTask(String id) {

try {

db.update("DELETE FROM tblTask WHERE TaskID = " + id + ";");

} catch (SQLException ex) {

Logger.getLogger(TaskManager.class.getName()).log(Level.SEVERE, null, ex);

}

}

/\*\*

\* Searches for a task based on the end date associated with the task.

\* @param lb (contains a date that is being searched for)

\* @return the task and its information based of the searched for date.

\*/

public static Task detailTask(LocalDate lb) {

try {

ResultSet rs = db.query("Select \* From tblTask WHERE UserID = "

+ UserManager.user.getUserID() + " AND EndDate = '"

+ lb.toString() + "' LIMIT 1;");

if (rs.next()) {

ResultSet subjects = db.query("SELECT Subject FROM tblSubject WHERE SubjectID =" + rs.getInt("SubjectID") + ";");

ResultSet ts = db.query("SELECT Status FROM tblTaskStatus WHERE StatusID = " + rs.getInt("StatusID") + ";");

subjects.next();

ts.next();

return new Task(rs.getInt("TaskID"), rs.getString("TaskName"),

rs.getString("CreateDate"), rs.getString("StartDate"),

rs.getString("EndDate"), rs.getString("Complete"),

rs.getInt("TimeAllocation"), rs.getString("Description"),

rs.getInt("SubjPriorityID"), subjects.getString("Subject"),

ts.getString("Status"));

}

} catch (SQLException ex) {

Logger.getLogger(TaskManager.class.getName()).log(Level.SEVERE, null, ex);

}

return null;

}

/\*\*

\* Creates an array of all the topics associated with the help table in the database

\* @return the help topic.

\*/

public static String[] getHelpTopics() {

try {

ResultSet rs = db.query("SELECT Topic FROM tblHelp;");

ArrayList<String> al = new ArrayList<>();

while (rs.next()) {

al.add(rs.getString("Topic"));

}

String[] temp = new String[1];

return al.toArray(temp);

} catch (SQLException ex) {

Logger.getLogger(TaskManager.class.getName()).log(Level.SEVERE, null, ex);

}

return null;

}

/\*\*

\* Creates an array of all the descriptions associated with the help table in the database

\* @return the help description

\*/

public static String[] getHelpDesc() {

try {

ResultSet rs = db.query("SELECT Description FROM tblHelp;");

ArrayList<String> al = new ArrayList<>();

while (rs.next()) {

al.add(rs.getString("Description"));

}

String[] temp = new String[1];

return al.toArray(temp);

} catch (SQLException ex) {

Logger.getLogger(TaskManager.class.getName()).log(Level.SEVERE, null, ex);

}

return null;

}

}

## JTableButtonRender

//Inge Senekal

package backend;

import java.awt.Component;

import javax.swing.JButton;

import javax.swing.JTable;

import javax.swing.table.TableCellRenderer;

/\*\*

\*

\* @author Inges

\*/

//Mehtods helps with adding a button to a table

//It allows a button to be added to a table.

public class JTableButtonRenderer implements TableCellRenderer {

@Override

public Component getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column) {

JButton button = (JButton) value;

return button;

}

}

# Code Help

## Help Screen

package frontend;

import backend.TaskManager;

import java.awt.Color;

import javax.swing.BorderFactory;

import javax.swing.DefaultComboBoxModel;

import javax.swing.JFrame;

/\*\*

\*

\* @author Inges

\*/

public class Help extends javax.swing.JFrame {

private String[] helpTopics;

private String[] helpDesc;

private JFrame frame;

/\*\*

\* Creates new form Help

\* Set the labels to the variables in the database

\* @param jframe get the frame that the help need to be displayed on. Depending on what screen the user is on when they require help

\*/

public Help(JFrame frame) {

this.frame = frame;

helpTopics = TaskManager.getHelpTopics();

helpDesc = TaskManager.getHelpDesc();

DefaultComboBoxModel model = new DefaultComboBoxModel(helpTopics);

initComponents();

cmbHelpTopics.setModel(model);

//Sets the background colour for the JFrame to white

getContentPane().setBackground(Color.white);

//Removes the border around the buttons

btnHomeScreen.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnHomeScreen.setContentAreaFilled(false);

}

/\*\*

\* 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() {

lblIcon = new javax.swing.JLabel();

btnHomeScreen = new javax.swing.JButton();

lblHomeScreen = new javax.swing.JLabel();

pnlHelp = new javax.swing.JPanel();

lblPanelHelp = new javax.swing.JLabel();

pnlOption = new javax.swing.JPanel();

cmbHelpTopics = new javax.swing.JComboBox<>();

pnlClean = new javax.swing.JPanel();

pnlHelpTopic = new javax.swing.JPanel();

lblHelpTopic = new javax.swing.JLabel();

lblHelp = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setResizable(false);

getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

lblIcon.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/OMT\_1.PNG"))); // NOI18N

lblIcon.setText("jLabel1");

getContentPane().add(lblIcon, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 0, 100, -1));

btnHomeScreen.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Resources/HomePage.png"))); // NOI18N

btnHomeScreen.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnHomeScreenActionPerformed(evt);

}

});

getContentPane().add(btnHomeScreen, new org.netbeans.lib.awtextra.AbsoluteConstraints(272, 11, 20, -1));

lblHomeScreen.setText("Return back to screen");

getContentPane().add(lblHomeScreen, new org.netbeans.lib.awtextra.AbsoluteConstraints(302, 15, 120, -1));

pnlHelp.setBackground(new java.awt.Color(0, 204, 255));

pnlHelp.setForeground(new java.awt.Color(255, 255, 255));

lblPanelHelp.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

lblPanelHelp.setForeground(new java.awt.Color(255, 255, 255));

lblPanelHelp.setText("Help");

javax.swing.GroupLayout pnlHelpLayout = new javax.swing.GroupLayout(pnlHelp);

pnlHelp.setLayout(pnlHelpLayout);

pnlHelpLayout.setHorizontalGroup(

pnlHelpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlHelpLayout.createSequentialGroup()

.addContainerGap()

.addComponent(lblPanelHelp, javax.swing.GroupLayout.PREFERRED\_SIZE, 41, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(389, Short.MAX\_VALUE))

);

pnlHelpLayout.setVerticalGroup(

pnlHelpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblPanelHelp, javax.swing.GroupLayout.DEFAULT\_SIZE, 30, Short.MAX\_VALUE)

);

getContentPane().add(pnlHelp, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 58, 440, 30));

pnlOption.setBackground(new java.awt.Color(204, 204, 204));

cmbHelpTopics.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

cmbHelpTopicsActionPerformed(evt);

}

});

javax.swing.GroupLayout pnlOptionLayout = new javax.swing.GroupLayout(pnlOption);

pnlOption.setLayout(pnlOptionLayout);

pnlOptionLayout.setHorizontalGroup(

pnlOptionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlOptionLayout.createSequentialGroup()

.addContainerGap()

.addComponent(cmbHelpTopics, javax.swing.GroupLayout.PREFERRED\_SIZE, 140, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(290, Short.MAX\_VALUE))

);

pnlOptionLayout.setVerticalGroup(

pnlOptionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, pnlOptionLayout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(cmbHelpTopics, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

getContentPane().add(pnlOption, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 80, 440, 30));

pnlClean.setBackground(new java.awt.Color(153, 153, 153));

javax.swing.GroupLayout pnlCleanLayout = new javax.swing.GroupLayout(pnlClean);

pnlClean.setLayout(pnlCleanLayout);

pnlCleanLayout.setHorizontalGroup(

pnlCleanLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 20, Short.MAX\_VALUE)

);

pnlCleanLayout.setVerticalGroup(

pnlCleanLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 20, Short.MAX\_VALUE)

);

getContentPane().add(pnlClean, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 110, 20, 20));

pnlHelpTopic.setBackground(new java.awt.Color(153, 153, 153));

lblHelpTopic.setText("Help Topic");

javax.swing.GroupLayout pnlHelpTopicLayout = new javax.swing.GroupLayout(pnlHelpTopic);

pnlHelpTopic.setLayout(pnlHelpTopicLayout);

pnlHelpTopicLayout.setHorizontalGroup(

pnlHelpTopicLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlHelpTopicLayout.createSequentialGroup()

.addGap(39, 39, 39)

.addComponent(lblHelpTopic, javax.swing.GroupLayout.PREFERRED\_SIZE, 67, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(54, Short.MAX\_VALUE))

);

pnlHelpTopicLayout.setVerticalGroup(

pnlHelpTopicLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, pnlHelpTopicLayout.createSequentialGroup()

.addGap(0, 6, Short.MAX\_VALUE)

.addComponent(lblHelpTopic))

);

getContentPane().add(pnlHelpTopic, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 110, 160, 20));

getContentPane().add(lblHelp, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 140, 430, 180));

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void cmbHelpTopicsActionPerformed(java.awt.event.ActionEvent evt) {

lblHelp.setText(helpDesc[cmbHelpTopics.getSelectedIndex()]);

}

private void btnHomeScreenActionPerformed(java.awt.event.ActionEvent evt) {

//Returns to the screen the user was on before getting to the Help Screen

frame.setVisible(true);

this.dispose();

}

// Variables declaration - do not modify

private javax.swing.JButton btnHomeScreen;

private javax.swing.JComboBox<String> cmbHelpTopics;

private javax.swing.JLabel lblHelp;

private javax.swing.JLabel lblHelpTopic;

private javax.swing.JLabel lblHomeScreen;

private javax.swing.JLabel lblIcon;

private javax.swing.JLabel lblPanelHelp;

private javax.swing.JPanel pnlClean;

private javax.swing.JPanel pnlHelp;

private javax.swing.JPanel pnlHelpTopic;

private javax.swing.JPanel pnlOption;

// End of variables declaration

}

# Externally Sourced Code

## Removing the boarder of a button:

This code is sourced from Stack Overflow

btnNewTask.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

btnNewTask.setContentAreaFilled(false);

## DB:

This code was provided by the teacher it connects the database to the program.

package backend;

/\*

\* 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.

\*/

/\*

To get your Java progrm to connect to MS Access you have to download UCanAccess at

https://sourceforge.net/projects/ucanaccess/files/

and add the following JAR files to the classpath:

ucanaccess-4.0.4.jar

hsqldb-2.3.1.jar

jackcess-2.1.11.jar

commons-lang-2.6.jar

commons-logging-1.1.3.jar

The version numbers here may differ than the latest versions you downloaded.

https://www.codejava.net/java-se/jdbc/java-jdbc-example-connect-to-microsoft-access-database

Add these libraries to the NetBeans Project

\*/

import java.sql.\*;

/\*\*

\*

\* @author Mr Christophers

\*/

public class DB {

private static final String url = "jdbc:mysql://localhost:3306/optimise\_my\_time";

private Connection conn;

private PreparedStatement statement;

private ResultSet resultSet;

/\*\*

\* Constructor for the DB class, this contains data from the optimise\_my\_time database.

\*/

public DB() {

try {

conn = DriverManager.getConnection(url, "root", "nbuser");

} catch (SQLException ex) {

System.out.println("Cannot connect");

}

}

/\*\*

\* This method updates data in the optimise\_my\_time database

\* @param update (Contains the SQL database)

\* @throws SQLException

\*/

public void update(String update) throws SQLException {

statement = conn.prepareStatement(update);

statement.executeUpdate();

statement.close();

}

/\*\*

\* The method is used for any other SQL statement with no separate methods made in other classes

\* @param stmt (Contains the SQL statement

\* @return the SQL statement

\* @throws SQLException to bypass any error

\*/

public ResultSet query(String stmt) throws SQLException {

statement = conn.prepareStatement(stmt);

resultSet = statement.executeQuery();

return resultSet;

}

/\*\*

\*

\* @param rs

\* @return the meta data

\*/

public String processResultSet(ResultSet rs) {

String temp = "";

try {

ResultSetMetaData meta = rs.getMetaData();

int size = meta.getColumnCount();

while (rs.next()) {

for (int i = 1; i <= size; i++) {

Object value = rs.getObject(i);

temp += "#" + value;

}

temp += "\n";

}

} catch (SQLException ex) {

ex.printStackTrace();

}

return temp;

}

}

# Explanation of Critical Algorithms

## Getting Task Algorithm

Begin

method getTask String id

try statement

rs <- query statement(get all task from tblTask where the Task ID <- ID)

if rs has next

ResultSet <- subject

subject <- query statement(get subject from tblSubject where the SubjectID <- the SubjectID in rs)

ResultSet <- ts

ts <- query statement(get Status from tblTaskStatus where the StatusID <- the StatusID in rs)

get the next subject

get the next ts

Create a new Task object called t

Set the values for the equal to( TaskID, TaskName, CreateDate, StartDate, EndDate, Complete, TimeAllocation, Description, SubjPriorityID, Subject, Status)

end if statement

Return t

Cache statement

Return null(nothing)

End

### Why code is critical

This code is important for the Task List screen and the Task Detail screen in the frontend of the table. This code is found in the back end in the TaskManager class.

The following code allows task to be accessed from the database and stores the information in a object of the task class

This is used by the Task Detail screen to display the tasks information and for the same purpose in the Task List Screen

This is critical since without it none of the task will be fetched from the database

## Insert or Updating a Task Algorithm

Begin

Method btnSubmitActionPerformed

try statement

if update equals true

LocalDate <- start

start <- getDate in the drop down at the current time user selected

LocalDate <- end

end <- getDate in the drop down at the current time user selected

Call the TaskManager class and DB to create an update statement

Update the TaskName in the tblTasks table where the TaskID equals id

Update the SubjectID in the tblTasks table where the TaskID equals id

Update the StartDate in the tblTasks table where the TaskID equals id

Update the EndDate in the tblTasks table where the TaskID equals id

Update the TimeAllocation in the tblTasks table where the TaskID equals id

Update the Description in the tblTasks table where the TaskID equals id

else

LocalDate <- start

start <- getDate in the drop down at the current time user selected

LocalDate <- end

end <- getDate in the drop down at the current time user selected

Call the TaskManager class and DB to create an update statement

Insert Into the tblTasks table in the columns TaskName, UserID, StatusID, SubjectID, SubjPriorityID, CreateDate, StartDate, EndDate, Complete, TimeAllocation, Description.

Add the following values(found in the textfields, combo box or text area in the JPanel)

end if statement

catch statement

Set the TaskList screen visible

Dispose of this screen

End

### Why code is critical

This code is found in the Task Detail screen, a frontend screen that uses classes in the backend such as Task and TaskManager

This code allows fields in a record to be changed or it allows the user to enter a new task based on whether the user clicked on the edit button or the new task button.

This code is critical since without it the user can not change information on a task or add a new task in the program. Therefor there will be no point of the program.

## Advanced Techniques

**Task Table**

I created a method called populateTaskList, This creates a table in the Task List Screen which allows the user to view there task in a table format. This means that the each new task are below on another under their specific topic.

The reason why I would view this as an advanced technique and me going the extra mile is because not only did I create a output for the user to view their task but each task has a edit or delete button generated with the task.

This means that every time a new task is added to the database a two new buttons are also added to the screen.

for the heading of each topic and two extra for the edit and delete button.

**Calendar**

Another method I created was updateMonths()

This creates a calendar in the planner screen and allows the user to move between months to view their tasks dues on certain days

The reason why I would view this as an advanced technique and me going the extra mile is because I created a hardcoded calendar which can move between different months accurately and each cell can be clicked on to view their task.

This causes me to add a mouseclicker which is also not something we learn in school

# Testing

## Evaluation of Solution

The program meets the original problem since the program Optimize my Time allows users to add task and view the task on a calendar. This allows the user to plan and aid the user in time management. Therefor the goal of the program is met since this program does resolve the problem of time management.

However not all goals were met while creating this program. The complexity of the planner screen would have taken me too long, I do not have the necessary skills yet to build such an extensive calendar, but I do believe however that I made a new goal while completing this task, and that is by improving my skills in programming.

## Functional Testing:

I will ask two candidates to test my program based on test plan therefor my functional testing and test plan can be conducted at the same time

When the program runs the users will start of on the Login Screen here the user can either enter their details if they have an account or they can register.

If the user Logs in on the Login Screen the next screen will be the Application Screen. The same screen will be displayed after the user has registered.

On the Application screen the user can either as for Help (The user will be moved to the Help Screen, from now on the help screen will be accessible from any screen), or click on the Task List Icon which moves to the Task List Screen, or they can move to the Planner Screen

PS if the user is new to the program task would have to be added in first before it will be found on the Planner Screen

On the Task List Screen, the user has a few options:

* The user can get help by clicking on the get help button
* The user can decide to add a new task by clicking on “Add New Task” button
* Depending on whether a task is already in the database the task can either be edited or deleted.

If the user edits or adds a new task the user will be moved to the Task Detail Screen

* They user must ensure that all fields have values and after the task must be submitted
* There will be a button to take the user to the help screen

Once the user is done with the Task List Screen, they move back to the Application screen where they can view the task in a calendar format on the Planner Screen

The planner screen is in the format of a calendar each cell of the calendar can be clicked on which moves the user to a Planner Detail Screen. If the task is due on the date selected the user will be able to view information on the task that is due.

## Test Plan

**Deborah Senekal**

Login Screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1*** | ***User Name*** | *'Deborah Senekal'* | DSenekalSueSullivanAndre…. | *@!$#@@* |
| ***1.2*** | ***Password*** | *'Karma2010'* | Karma2010… | *[empty]* |

Register Screen

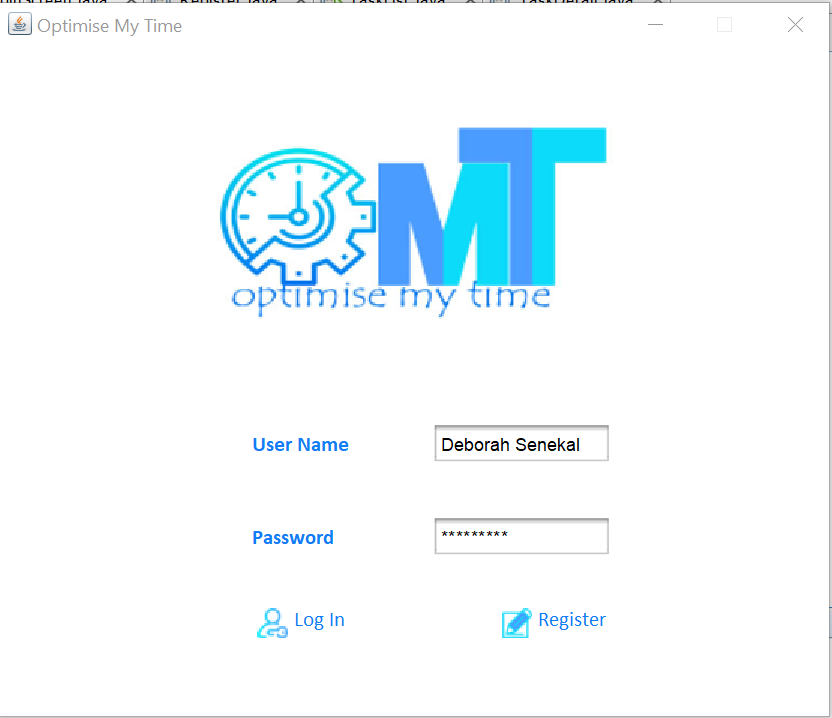
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1*** | ***User Name*** | *'Deborah Senekal'* | DSenekalSueSullivanAndre…. | *@!$#@@* |
| ***1.2*** | ***Password*** | *'Karma2010'* | Karma2010… | *[empty]* |
| ***1.3*** | ***Telephone*** | *'079 495 7281'* | *(more 13 ch)* | *[empty]* |
| ***1.4*** | ***Email Address*** | *'deborahsenekal@gmail.com'* | *[empty]* | *deborah* |

Task Detail Screen

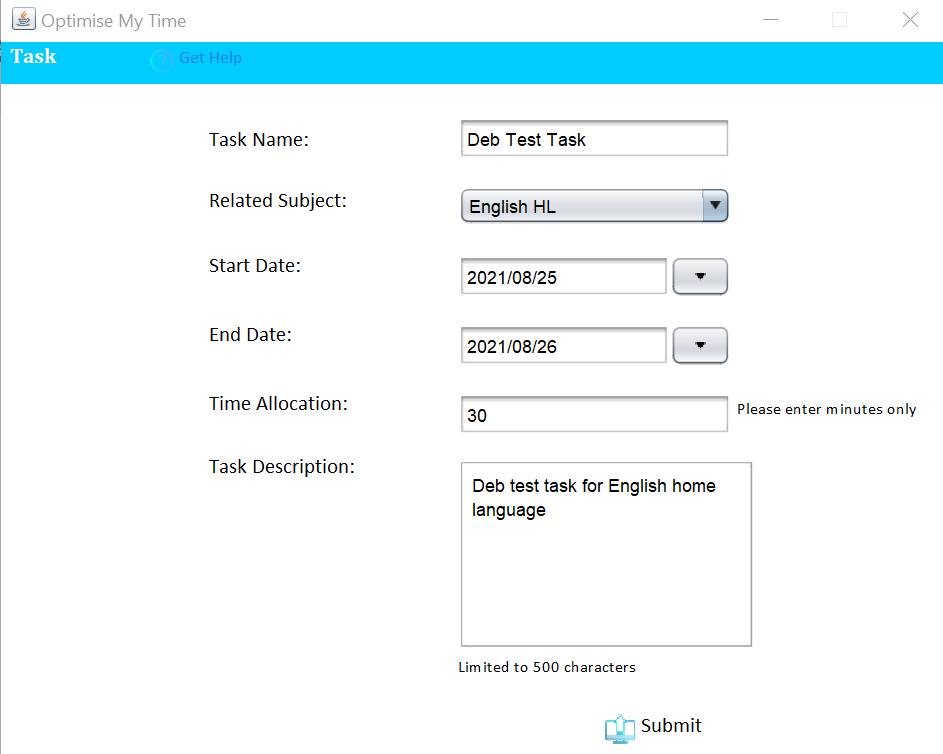
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1 Task Name*** | ***Task Name*** | *Deb Test Task* | *[ALL CAPS]Task* | *@($\*%&#@^#* |
| ***1.2*** | ***Related Subject*** | *English HL* | *English HL* | *English HL* |
| ***1.3*** | ***Start Date*** | *2021/08/25* | *2020/12/22* | *-* |
| ***1.4*** | ***End Date*** | *2021/08/26* | *2022/03/02* | *-* |
| ***1.5*** | ***Time Allocation*** | *30* | *3a* | *\*#$%)(#\*%&#* |
| ***1.6*** | ***Task Description*** | *Deb test task for English Home Language* | *[empty]* | *#$431845$\*#&2* |

Normal Testing

Login Register

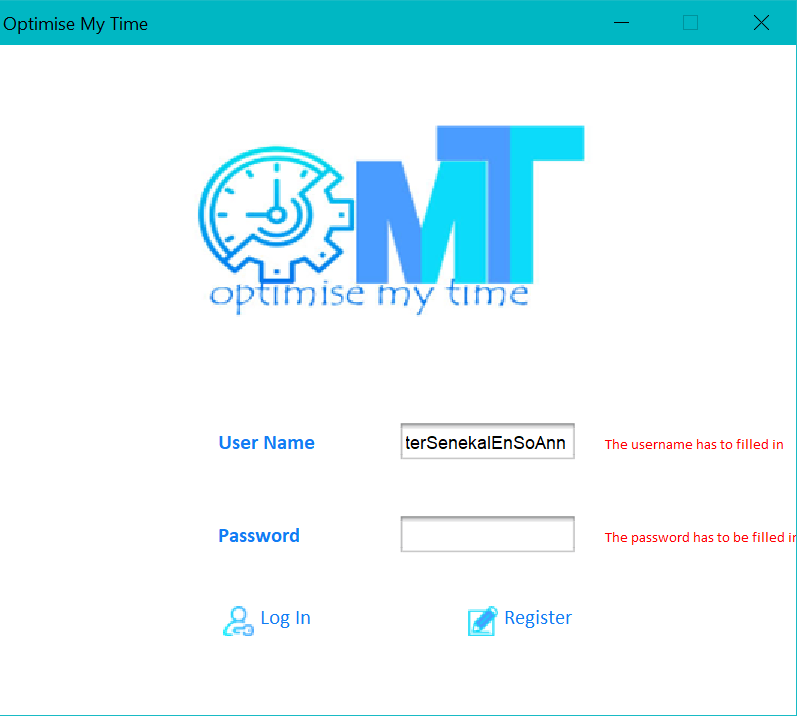


Task Detail

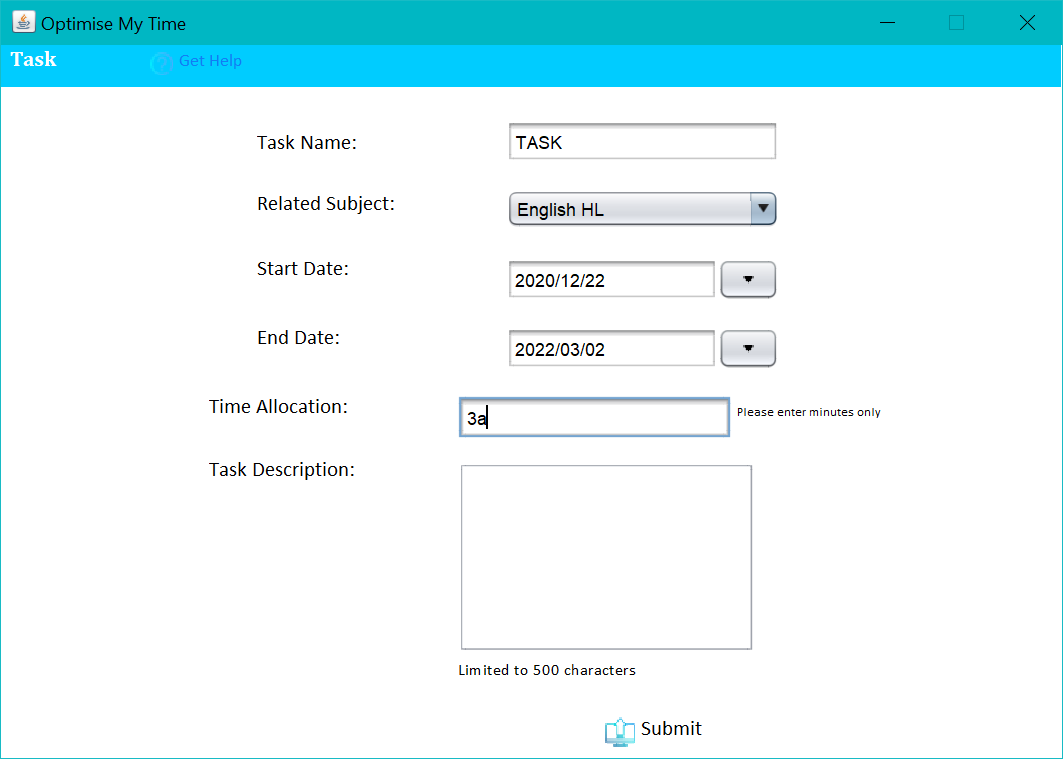


Extreme Testing

Login Register

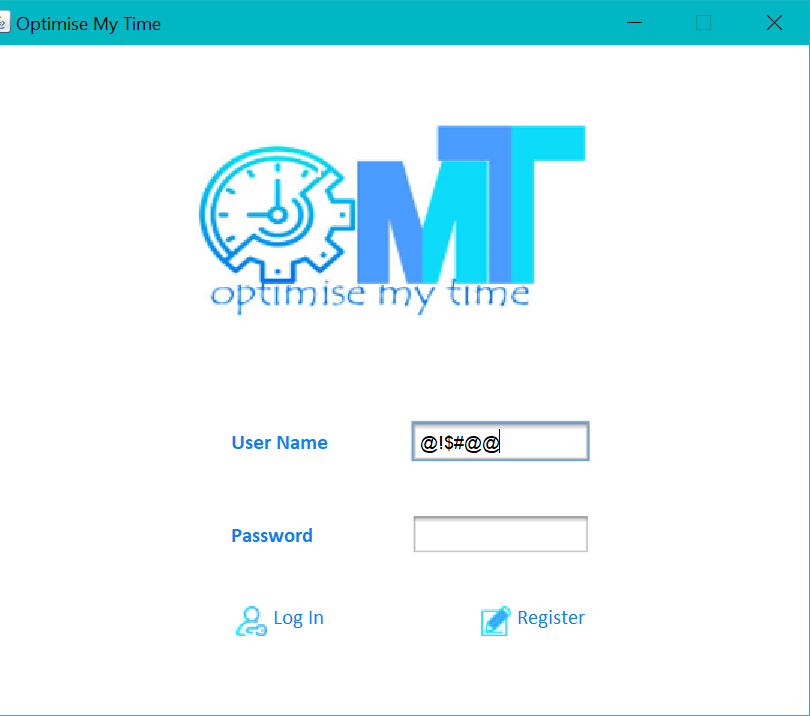
 

Task Detail

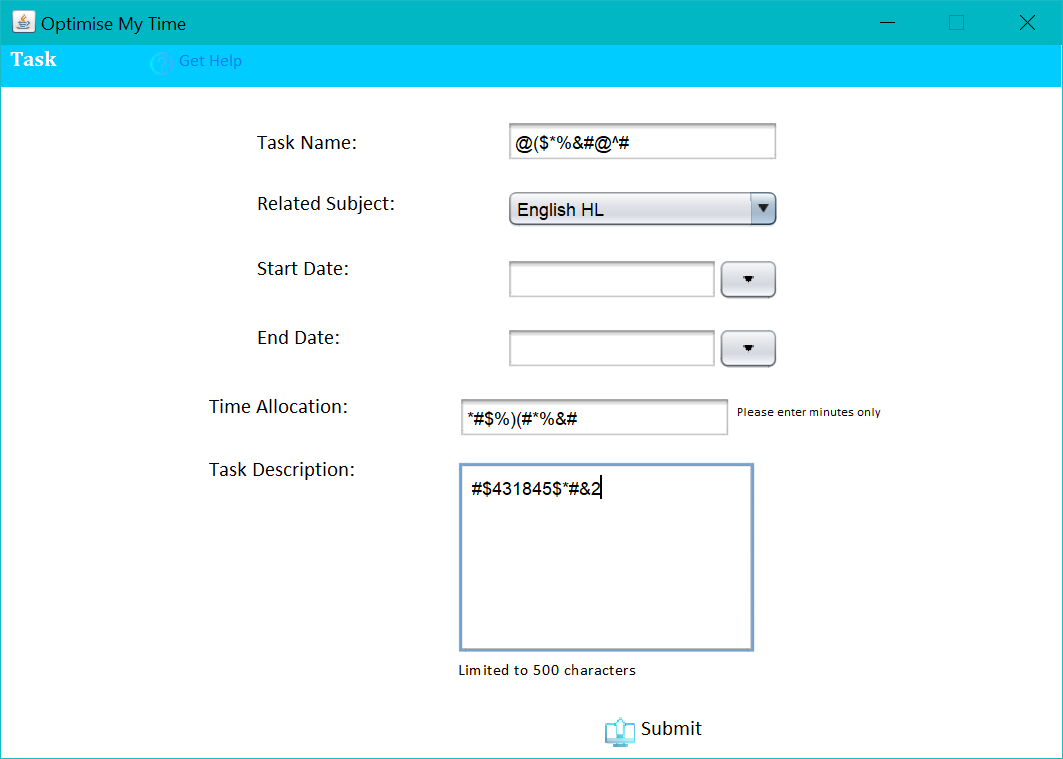


Erroneous Testing

Login Register



Task Detail



**Zachary Christophers**

Login Screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1*** | ***User Name*** | *Zach* | *(50 character name)* | *!@#$%^&\*(* |
| ***1.2*** | ***Password*** | *Test1234* | *(50 character password)* | *[empty]* |

Register Screen

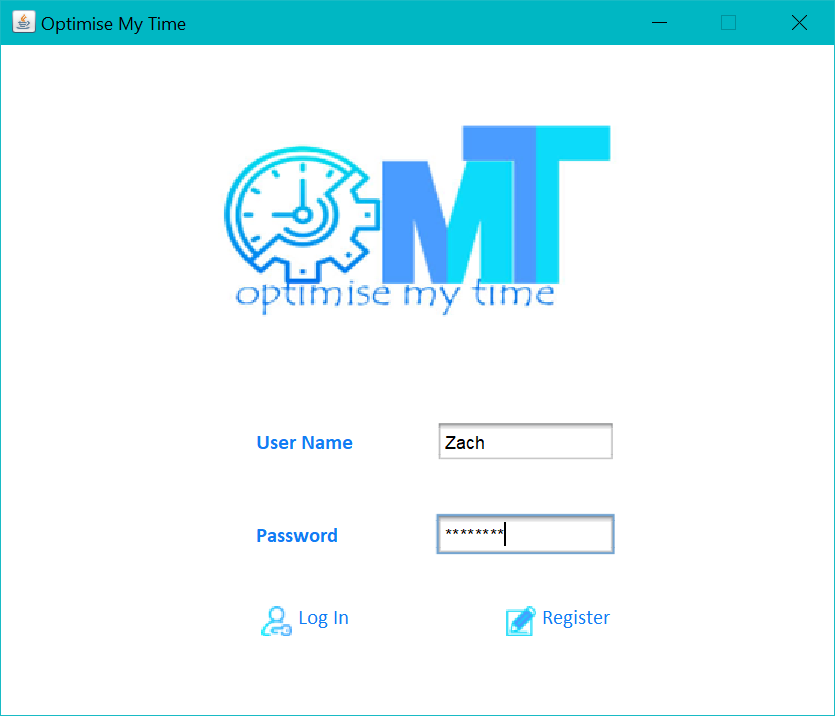
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1*** | ***User Name*** | *Zach* | *(50 character name)* | *!@#$%^&\*(* |
| ***1.2*** | ***Password*** | *Test1234* | *(50 character password)* | *[empty]* |
| ***1.3*** | ***Telephone*** | *0793179036* | *07912624848516* | *!#$#$%^&\*(* |
| ***1.4*** | ***Email Address*** | *Zachary.christophers@gmail.com* | *[empty]* | *zachemail* |

Task Detail Screen

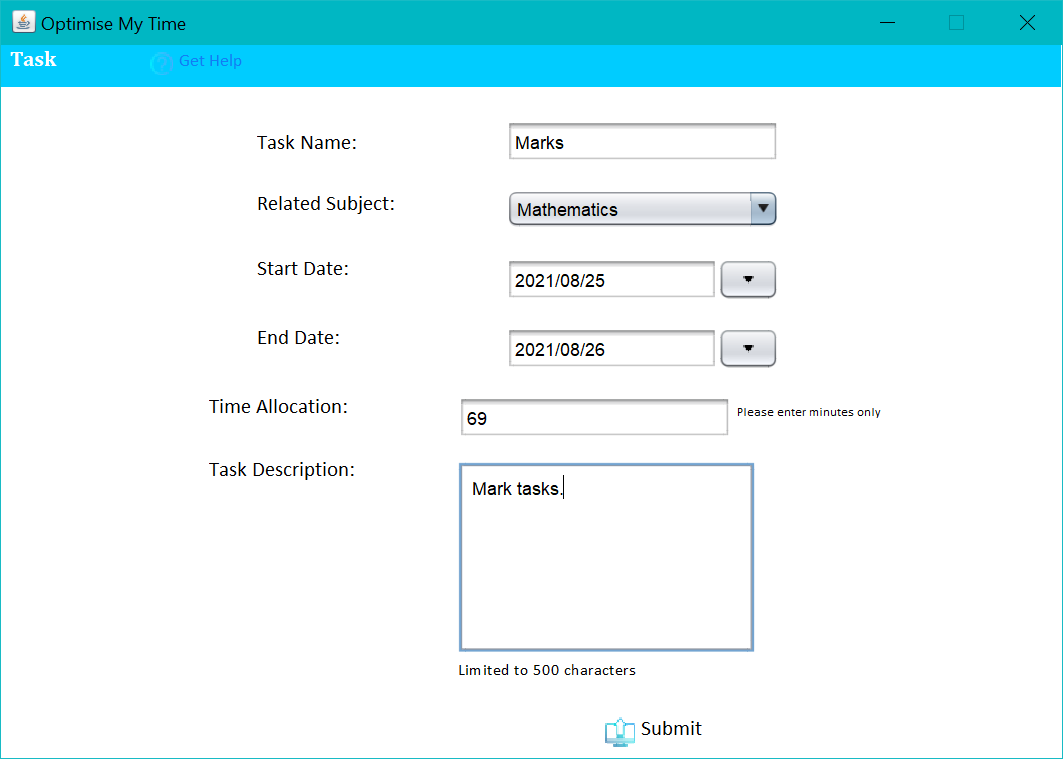
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1 Task Name*** | ***Task Name*** | *Mark tasks* | *MARK ALL THE TASKS GO GO GOGO GO GOG O* | *#$%^&\** |
| ***1.2*** | ***Related Subject*** | *Information Technology* | *Information Technology* | *Information Technology* |
| ***1.3*** | ***Start Date*** | *2021/08/24* | *2020/01/01* | *-* |
| ***1.4*** | ***End Date*** | *2021/08/26* | *2022/12/31* | *-* |
| ***1.5*** | ***Time Allocation*** | *69* | *69999999999420* | *#$%^* |
| ***1.6*** | ***Task Description*** | *Mark the tasks* | *Mark all the tasks in a timeous manner and complete them with minimal hassle or interruption* | *!@#$%^&\*()7* |

Normal Testing

Login Register

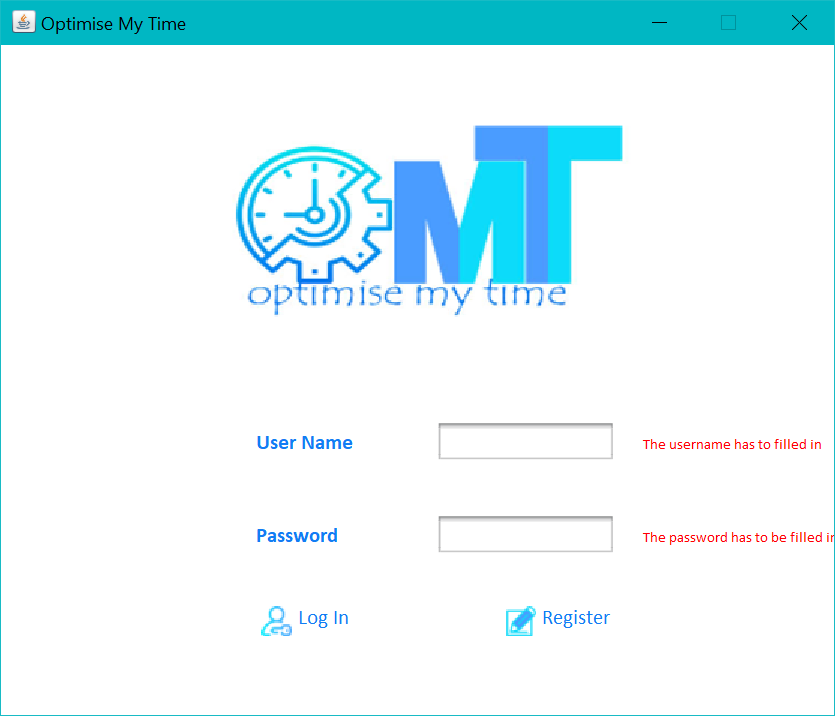
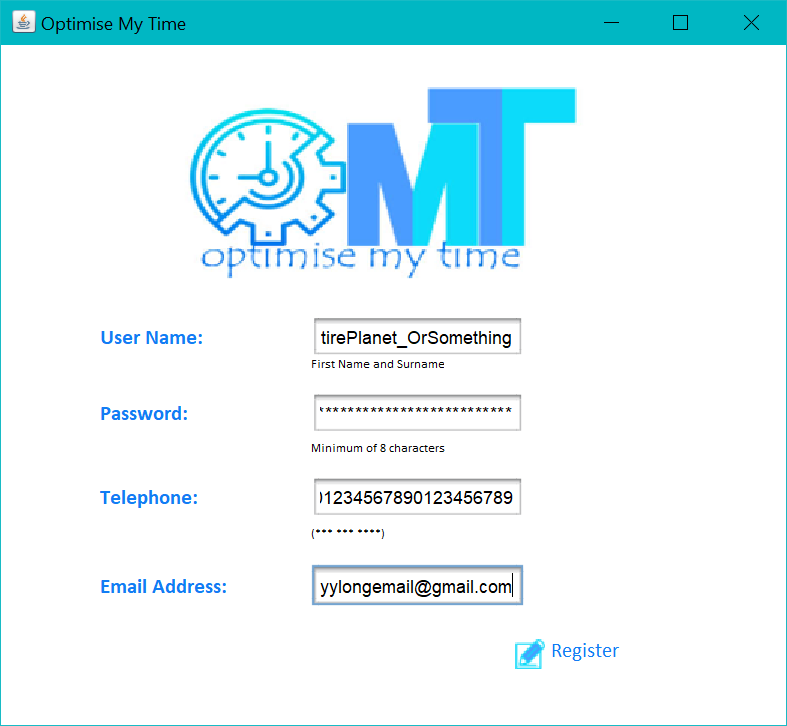


Task Detail

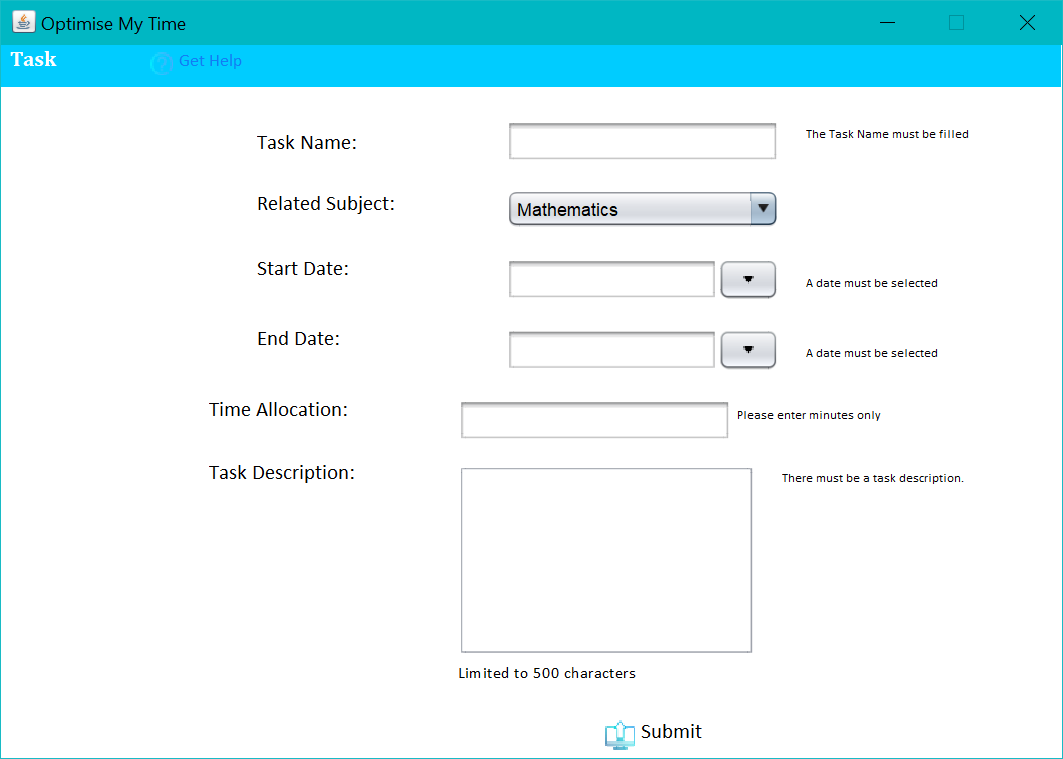


Extreme Testing

Login Register

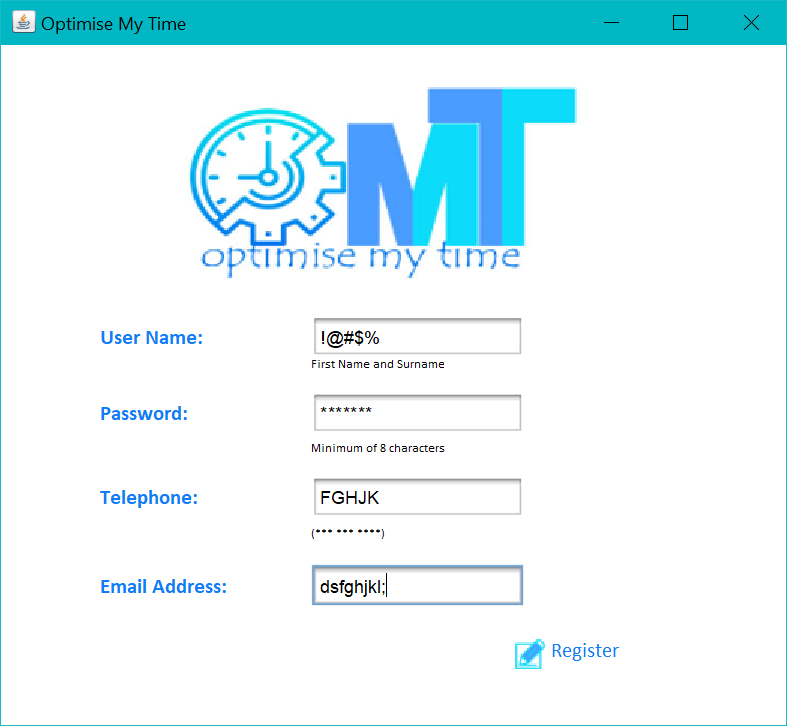
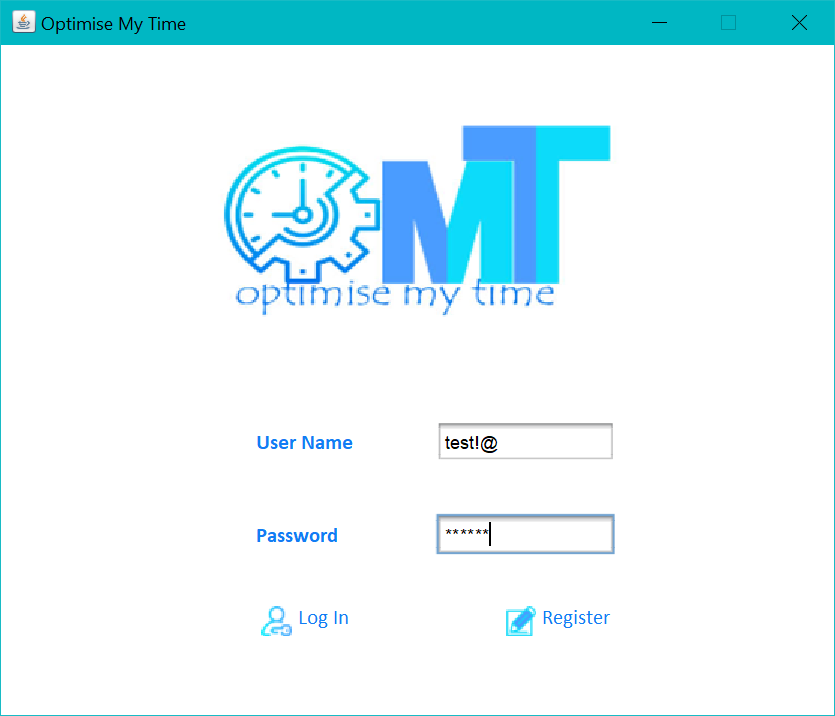
 

Task Detail

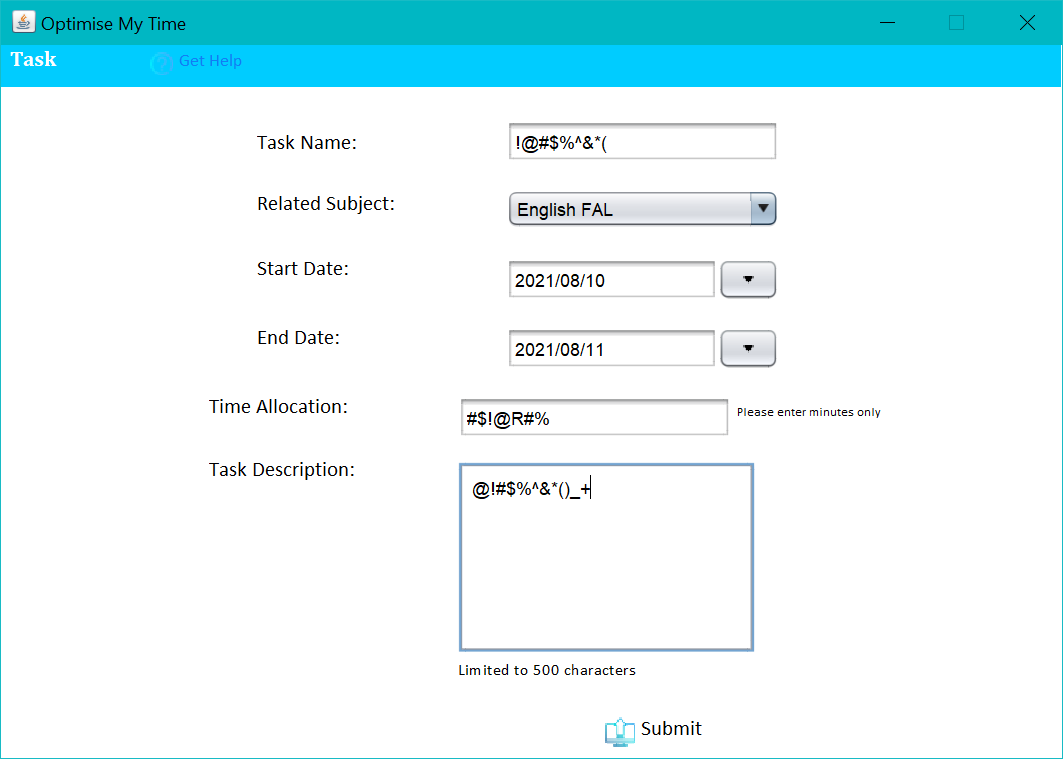


Erroneous Testing

Login Register



Task Detail



# Test Results

**Deborah Senekal**

Login Screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1*** | ***User Name*** | *Success* | *Success* | *Success* |
| ***1.2*** | ***Password*** | *Success* | *Success* | *Success* |

Register Screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1*** | ***User Name*** | *Success* | *Success* | *Success* |
| ***1.2*** | ***Password*** | *Success* | *Success* | *Success* |
| ***1.3*** | ***Telephone*** | *Success* | *Success* | *Success* |
| ***1.4*** | ***Email Address*** | *Success* | *Success* | *Success* |

Task Detail Screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1*** | ***Task Name*** | *Success* | *Success* | *Fail* |
| ***1.2*** | ***Related Subject*** | *Success* | *Success* | *Fail* |
| ***1.3*** | ***Start Date*** | *Success* | *Fail* | *Success* |
| ***1.4*** | ***End Date*** | *Success* | *Fail* | *Success* |
| ***1.5*** | ***Time Allocation*** | *Success* | *Success* | *Success* |
| ***1.6 Description*** | ***Task Description*** | *Success* | *Success* | *Success* |

**Zachary Christophers**

Login Screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1*** | ***User Name*** | *Success* | *Success* | *Success* |
| ***1.2*** | ***Password*** | *Success* | *Success* | *Success* |

Register Screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1*** | ***User Name*** | *Success* | *Success* | *Success* |
| ***1.2*** | ***Password*** | *Success* | *Success* | *Success* |
| ***1.3*** | ***Telephone*** | *Success* | *Success* | *Success* |
| ***1.4*** | ***Email Address*** | *Success* | *Success* | *Success* |

Task Detail Screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test #*** | ***Input*** | ***Normal*** | ***Extreme*** | ***Erroneous*** |
| ***1.1*** | ***Task Name*** | *Success* | *Success* | *Fail* |
| ***1.2*** | ***Related Subject*** | *Success* | *Success* | *Fail* |
| ***1.3*** | ***Start Date*** | *Success* | *Fail* | *Success* |
| ***1.4*** | ***End Date*** | *Success* | *Fail* | *Success* |
| ***1.5*** | ***Time Allocation*** | *Success* | *Success* | *Success* |
| ***1.6 Description*** | ***Task Description*** | *Success* | *Success* | *Success* |