FINAL Lab Quiz 2:
Registration with Event

Requirements:

- **Java Code**
- **NetBeans IDE**
- **Documentation**

Groupmates:

(List here the names of your group and their contribution)

Note: This is an individual activity

Lab Activity Problem:

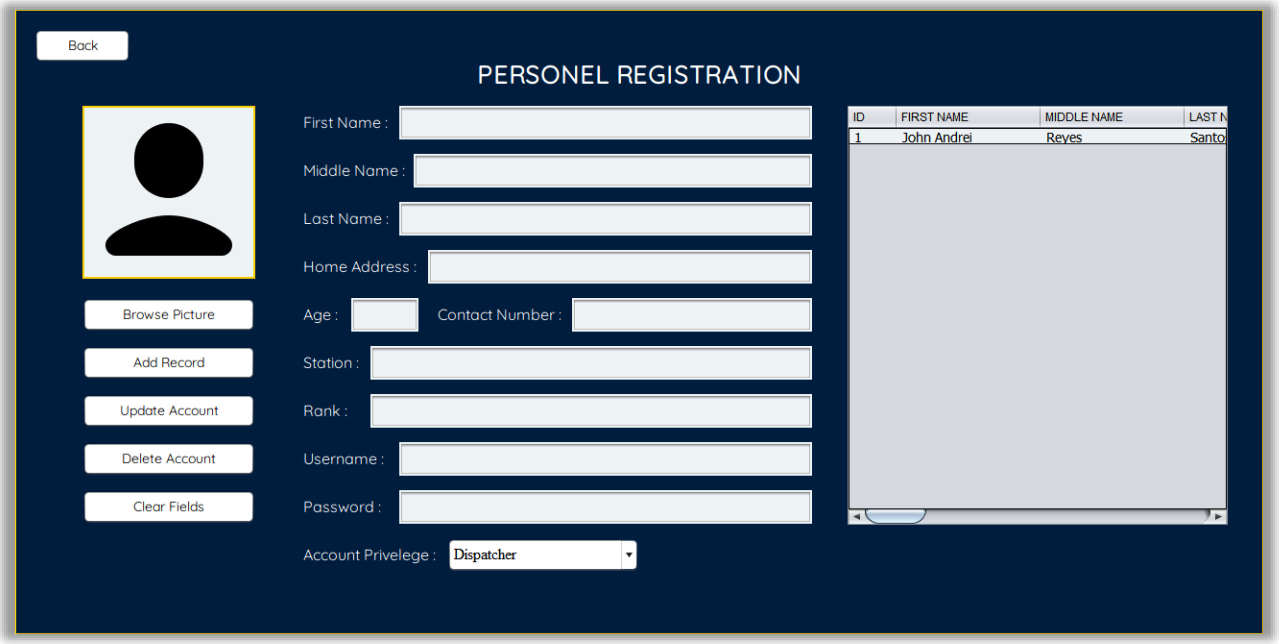
1. Create a program that register data, it gets the inputted data of user and post it in frame.

   Name: (TextField)
   Mobile: (TextField)
   Gender: (Radio Button)
   Date of Birth: (JTextField or JComboBox)
   Address: (TextArea)
   Button: Submit and Reset (JButtons)

Grade Matrix:

Functionality(Events)	:	40%
Design	:	30%
Documentation	:	30%
		100%

BTSRegistration.java



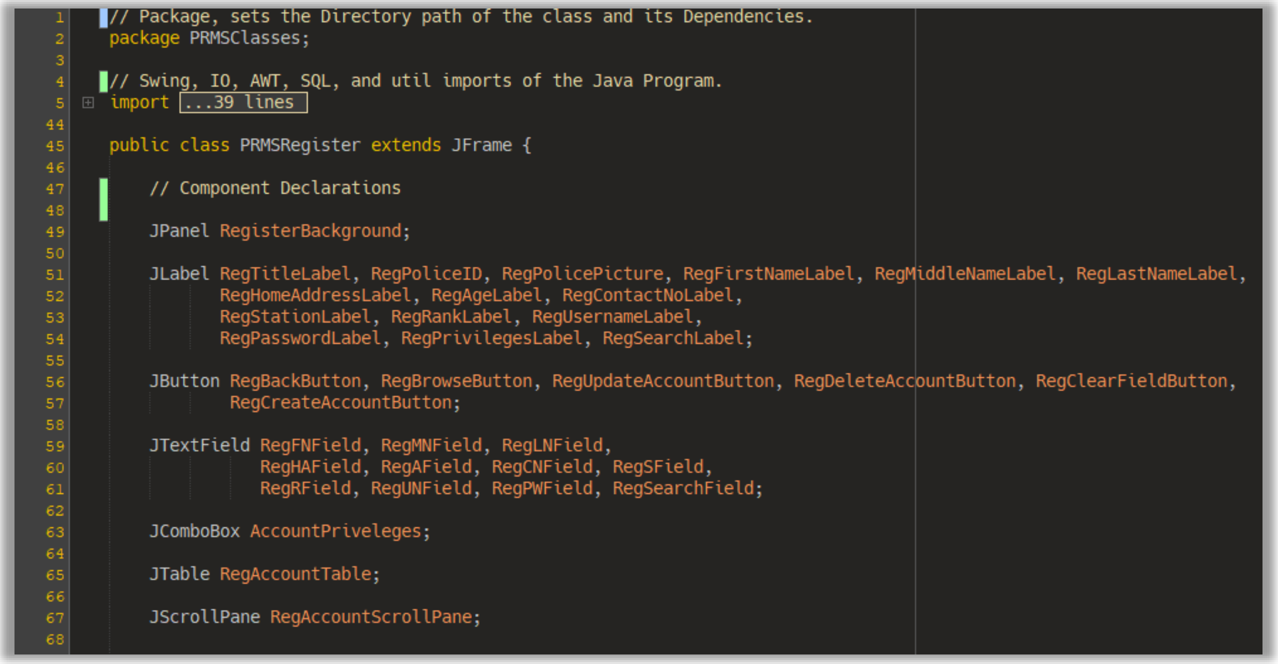
Account Privelege : Dispatcher

ID	FIRST NAME	MIDDLE NAME	LAST N
1	John Andrei	Reyes	Santo

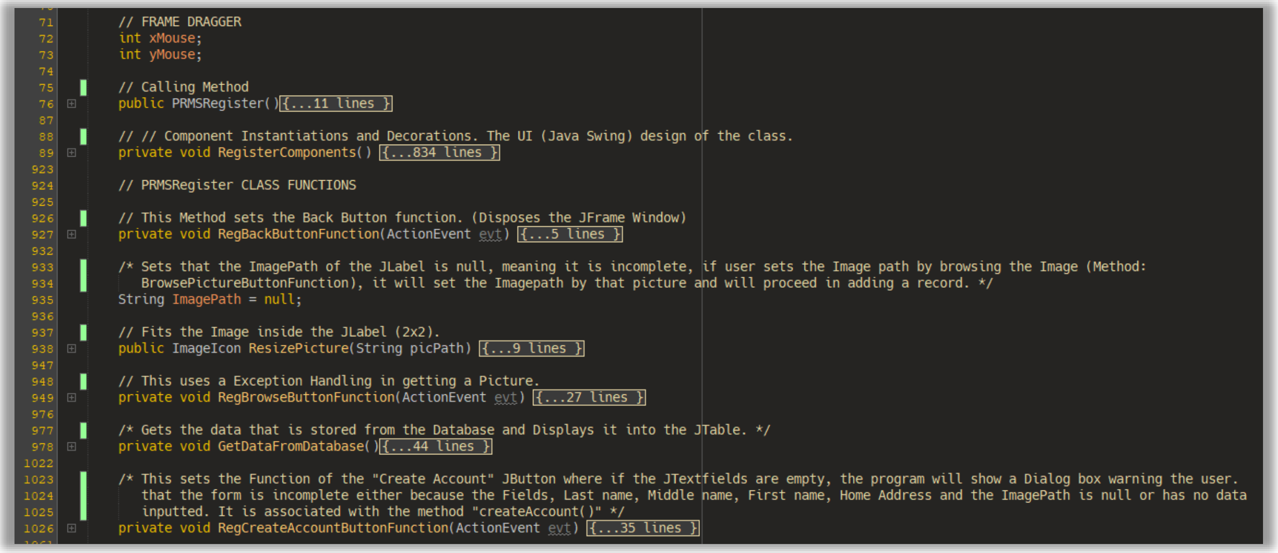
This JFrame window that Adds, Updates, and Registers users data in the Police Record Management System. This creates an account for the user to be able to log into the Login Window.

This JFrame Class includes, 1 JPanel (as Background), 6 JButtons, 13 JLabels, 10 JTextFields, and a JTable.

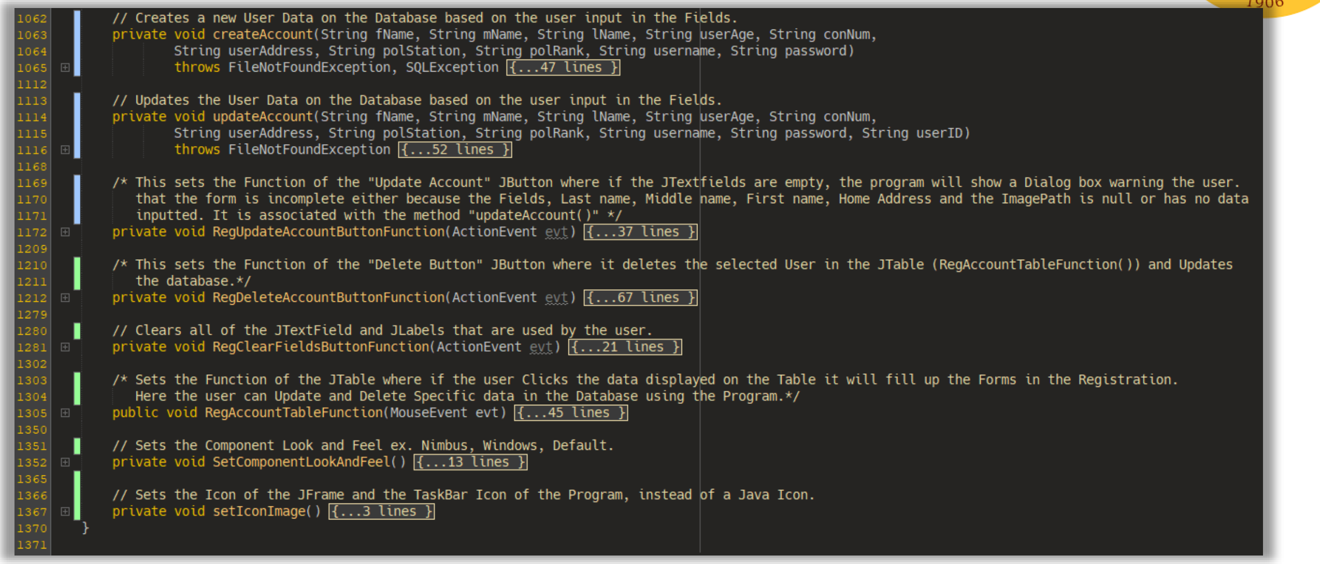
PRMS Register Source Code Summary



```
// Package, sets the Directory path of the class and its Dependencies.
package PRMSClasses;

// Swing, IO, AWT, SQL, and util imports of the Java Program.
import ...39 lines

public class PRMSRegister extends JFrame {

    // Component Declarations

    JPanel RegisterBackground;

    JLabel RegTitleLabel, RegPoliceID, RegPolicePicture, RegFirstNameLabel, RegMiddleNameLabel, RegLastNameLabel,
           RegHomeAddressLabel, RegAgeLabel, RegContactNoLabel,
           RegStationLabel, RegRankLabel, RegUsernameLabel,
           RegPasswordLabel, RegPrivilegesLabel, RegSearchLabel;

    JButton RegBackButton, RegBrowseButton, RegUpdateAccountButton, RegDeleteAccountButton, RegClearFieldButton,
            RegCreateAccountButton;

    JTextField RegFNField, RegMNField, RegLNField,
               RegHAField, RegAField, RegCNField, RegSField,
               RegRField, RegUNField, RegPWField, RegSearchField;

    JComboBox AccountPriveleges;

    JTable RegAccountTable;

    JScrollPane RegAccountScrollPane;

```

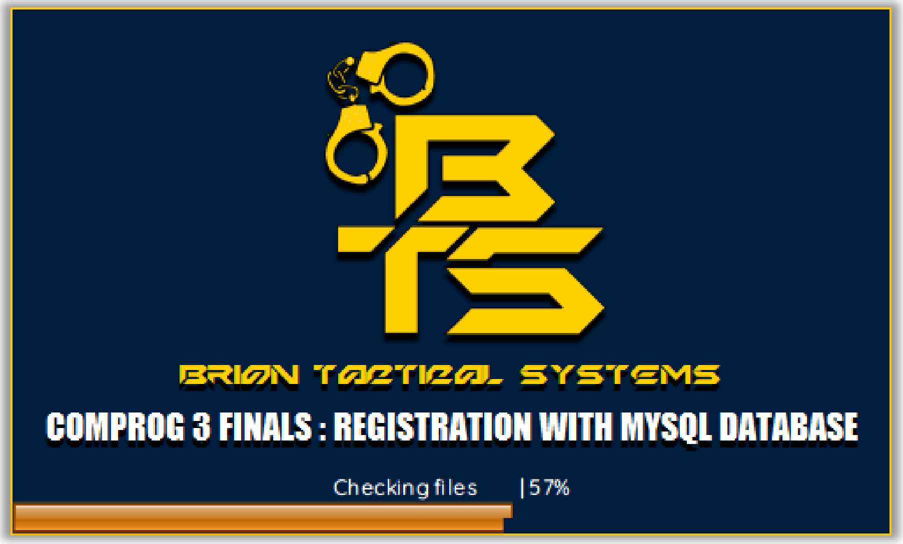


```
    // FRAME DRAGGER
    int xMouse;
    int yMouse;

    // Calling Method
    public PRMSRegister(){...11 lines }

    // // Component Instantiations and Decorations. The UI (Java Swing) design of the class.
    private void RegisterComponents() {...834 lines }

    // PRMSRegister CLASS FUNCTIONS

    // This Method sets the Back Button function. (Disposes the JFrame Window)
    private void RegBackButtonFunction(ActionEvent evt) {...5 lines }

    /* Sets that the ImagePath of the JLabel is null, meaning it is incomplete, if user sets the Image path by browsing the Image (Method:
       BrowsePictureButtonFunction), it will set the Imagepath by that picture and will proceed in adding a record. */
    String ImagePath = null;

    // Fits the Image inside the JLabel (2x2).
    public ImageIcon ResizePicture(String picPath) {...9 lines }

    // This uses a Exception Handling in getting a Picture.
    private void RegBrowseButtonFunction(ActionEvent evt) {...27 lines }

    /* Gets the data that is stored from the Database and Displays it into the JTable. */
    private void GetDataFromDatabase(){...44 lines }

    /* This sets the Function of the "Create Account" JButton where if the JTextfields are empty, the program will show a Dialog box warning the user.
       that the form is incomplete either because the Fields, Last name, Middle name, First name, Home Address and the ImagePath is null or has no data
       inputted. It is associated with the method "createAccount()" */
    private void RegCreateAccountButtonFunction(ActionEvent evt) {...35 lines }
```



```
    // Creates a new User Data on the Database based on the user input in the Fields.
    private void createAccount(String fName, String mName, String lName, String userAge, String conNum,
            String userAddress, String polStation, String polRank, String username, String password)
            throws FileNotFoundException, SQLException {...47 lines }

    // Updates the User Data on the Database based on the user input in the Fields.
    private void updateAccount(String fName, String mName, String lName, String userAge, String conNum,
            String userAddress, String polStation, String polRank, String username, String password, String userID)
            throws FileNotFoundException {...52 lines }

    /* This sets the Function of the "Update Account" JButton where if the JTextfields are empty, the program will show a Dialog box warning the user.
       that the form is incomplete either because the Fields, Last name, Middle name, First name, Home Address and the ImagePath is null or has no data
       inputted. It is associated with the method "updateAccount()" */
    private void RegUpdateAccountButtonFunction(ActionEvent evt) {...37 lines }

    /* This sets the Function of the "Delete Button" JButton where it deletes the selected User in the JTable (RegAccountTableFunction()) and Updates
       the database.*/
    private void RegDeleteAccountButtonFunction(ActionEvent evt) {...67 lines }

    // Clears all of the JTextField and JLabels that are used by the user.
    private void RegClearFieldsButtonFunction(ActionEvent evt) {...21 lines }

    /* Sets the Function of the JTable where if the user Clicks the data displayed on the Table it will fill up the Forms in the Registration.
       Here the user can Update and Delete Specific data in the Database using the Program.*/
    public void RegAccountTableFunction(MouseEvent evt) {...45 lines }

    // Sets the Component Look and Feel ex. Nimbus, Windows, Default.
    private void SetComponentLookAndFeel() {...13 lines }

    // Sets the Icon of the JFrame and the TaskBar Icon of the Program, instead of a Java Icon.
    private void setIconImage() {...3 lines }
}
```

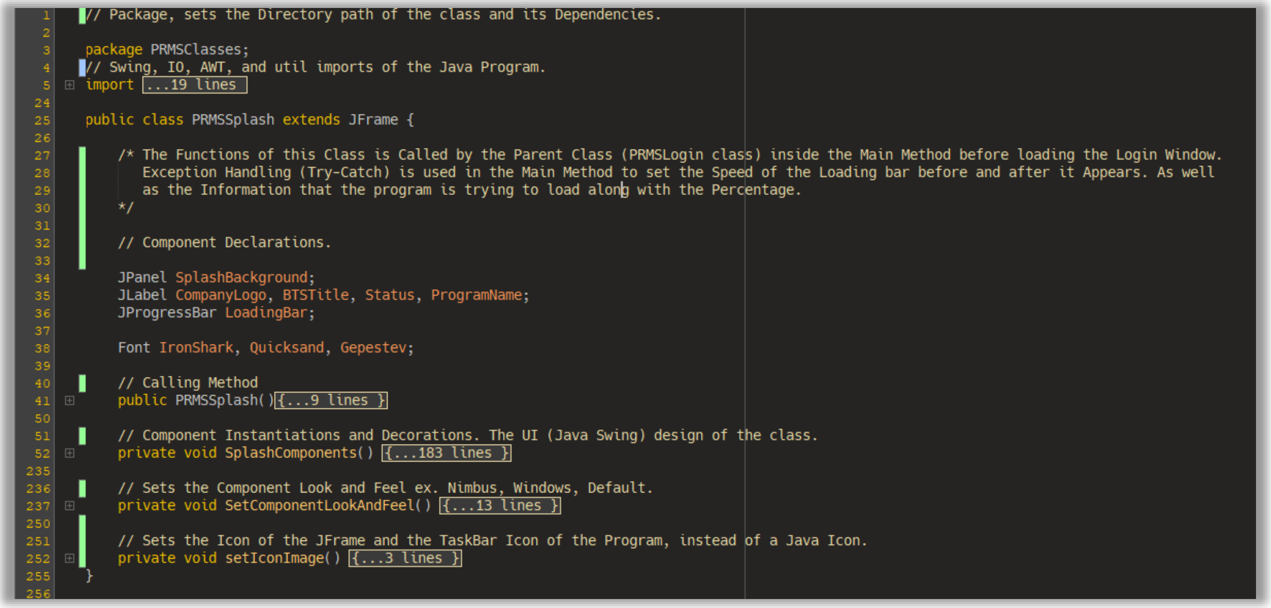
BTSSplash.java



This JFrame window displays the Splash Screen before proceeding into the PRMSLogin Window. This presents the startup of the program.

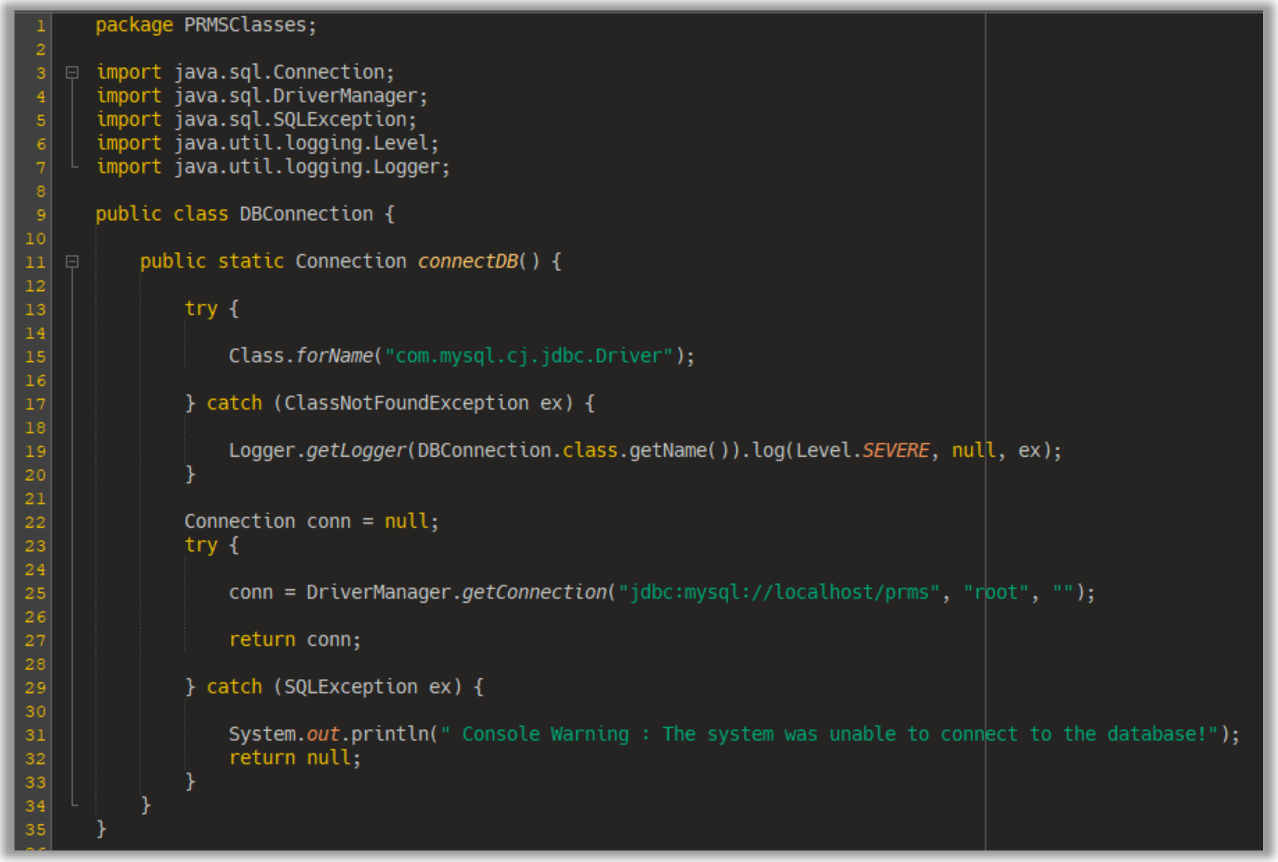
This JFrame Class includes, 1 JPanel, 4 JLabels, and a JProgressBar.

PRMS Splash Screen Code Summary



```
// Package, sets the Directory path of the class and its Dependencies.

package PRMSClasses;
// Swing, IO, AWT, and util imports of the Java Program.
import ...19 lines

public class PRMSSplash extends JFrame {

    /* The Functions of this Class is Called by the Parent Class (PRMSLogin class) inside the Main Method before loading the Login Window.
       Exception Handling (Try-Catch) is used in the Main Method to set the Speed of the Loading bar before and after it Appears. As well
       as the Information that the program is trying to load along with the Percentage.
    */

    // Component Declarations.

    JPanel SplashBackground;
    JLabel CompanyLogo, BTSTitle, Status, ProgramName;
    JProgressBar LoadingBar;

    Font IronShark, Quicksand, Gepestev;

    // Calling Method
    public PRMSSplash(){...9 lines }

    // Component Instantiations and Decorations. The UI (Java Swing) design of the class.
    private void SplashComponents() {...183 lines }

    // Sets the Component Look and Feel ex. Nimbus, Windows, Default.
    private void SetComponentLookAndFeel() {...13 lines }

    // Sets the Icon of the JFrame and the TaskBar Icon of the Program, instead of a Java Icon.
    private void setIconImage() {...3 lines }
}
```

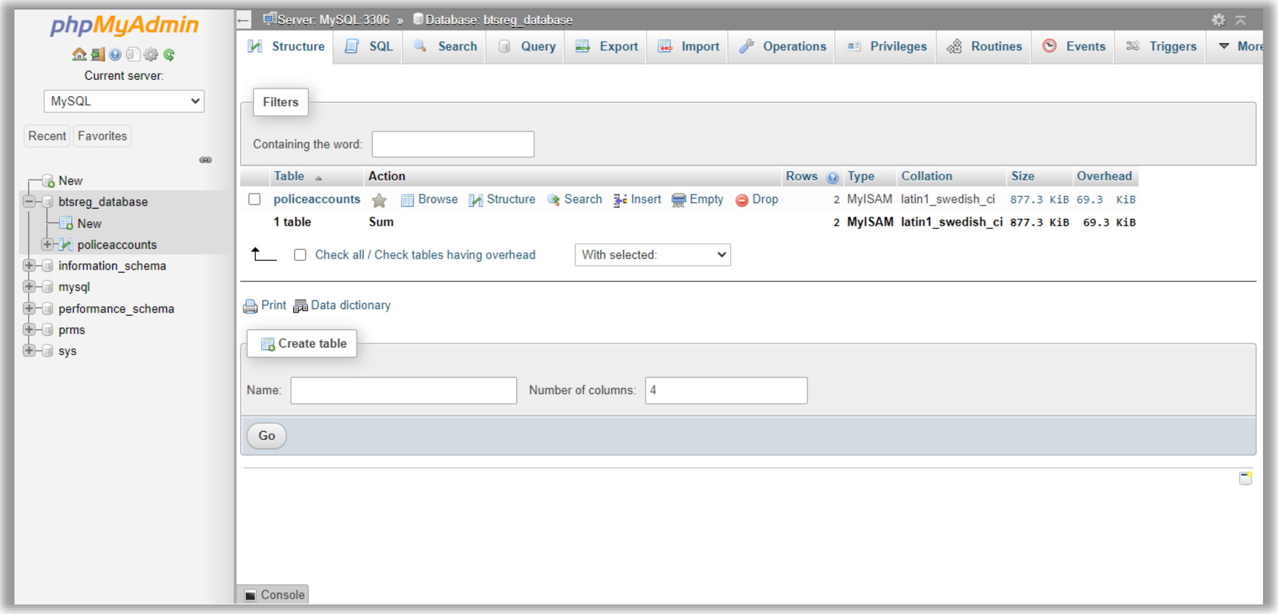
DBConnection.java



```java
package PRMSClasses;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class DBConnection {

    public static Connection connectDB( ) {

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

        } catch (ClassNotFoundException ex) {

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

        Connection conn = null;
        try {

            conn = DriverManager.getConnection("jdbc:mysql://localhost/prms", "root", "");

            return conn;

        } catch (SQLException ex) {

            System.out.println(" Console Warning : The system was unable to connect to the database!");
            return null;
        }
    }
}
```

This Java Class establishes the connection between the Program and the Database, in this Case MySQL database is used in this Management System. It first gets the JDBC Driver (Also called ConnectorJ) that is included in the Libraries. Using exception handling, assuming that the connection is null, if the database is not running, the program will warn the user that the database is not running at startup or at the PRMS Login Class. But if the Connection is established, it will proceed in logging into the database using the URL link of the JDBC, Root username and password.

MySQL Database via WAMP Server.



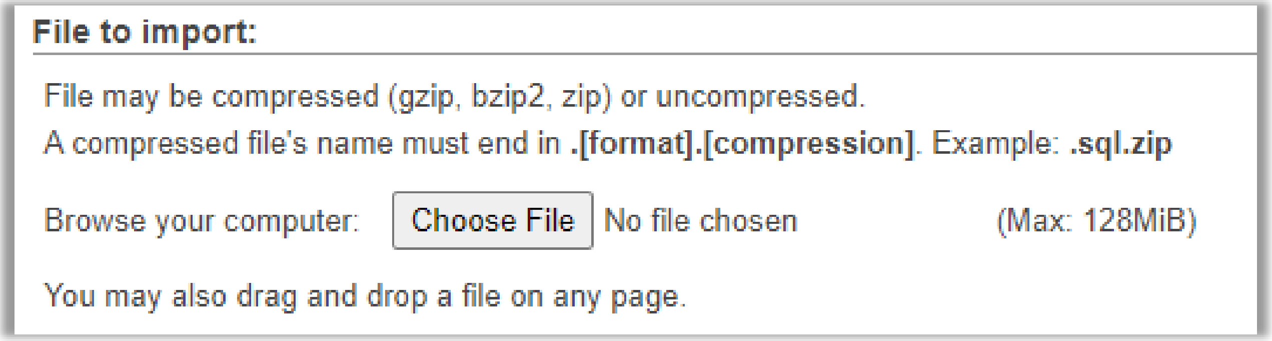
Containing the word:

	Table	Action	Rows	Type	Collation	Size	Overhead
□	policeaccounts	Browse Structure Search Insert Empty Drop	2	MyISAM	latin1_swedish_ci	877.3 KiB	69.3 KiB
	1 table	Sum	2	MyISAM	latin1_swedish_ci	877.3 KiB	69.3 KiB

Name:

Number of columns: 4

This is MySQL Database via the Browser. This is where all of the User Accounts and Personal Records are stored. Recommended for Beginners. The “btsreg_database.sql” file is in the folder named “Import Database File Before Use” that you can import into SQL.



File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: Choose File No file chosen (Max: 128MiB)

You may also drag and drop a file on any page.

Here are the Instruction from the Database in Importing the prms.sql file.

A Video Demo is Included in making the program work.
File Name : Project Demo.mkv