

# BarrioSeguro

Barangay Monitoring and Incident Reporting System

Dela Cruz, Juan Gabriel  
Estonilo, Julius Evan  
BSCS 3-3



# INTRODUCTION

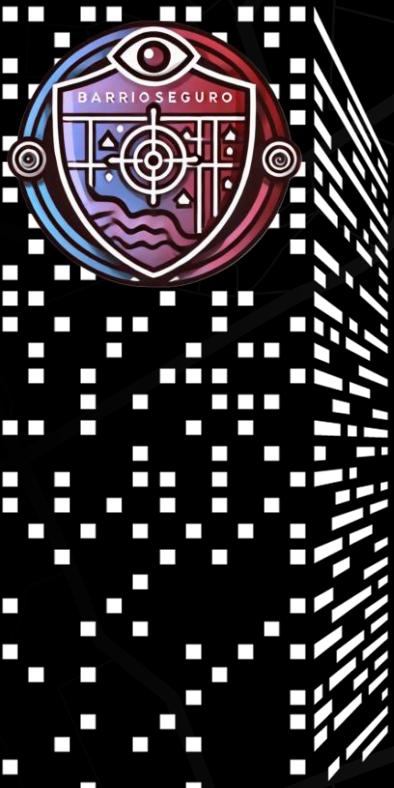
Barangays play a critical role in Philippine governance and community safety. However, outdated manual systems hinder their efficiency in handling incident reports and administrative processes.



## PROBLEM IDENTIFICATION

Reliance on manual methods for tracking data, incident reporting, and announcements results in inefficiencies, disorganized records, and delayed responses, weakening barangay-level governance. They rely on paper records and outdated devices, which makes it hard to quickly share important information.





## SOLUTION

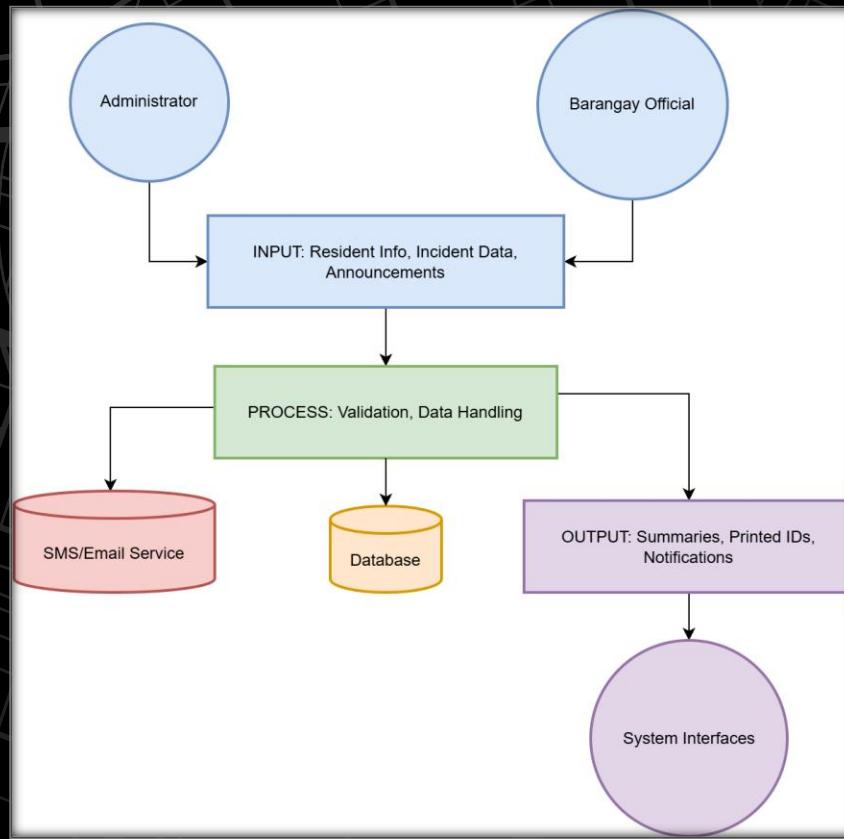
The development of a **desktop application** called “*BarrioSeguro*” – A proposed solution for the problems of inefficient barangay monitoring and incident reporting system.

# SOLUTION

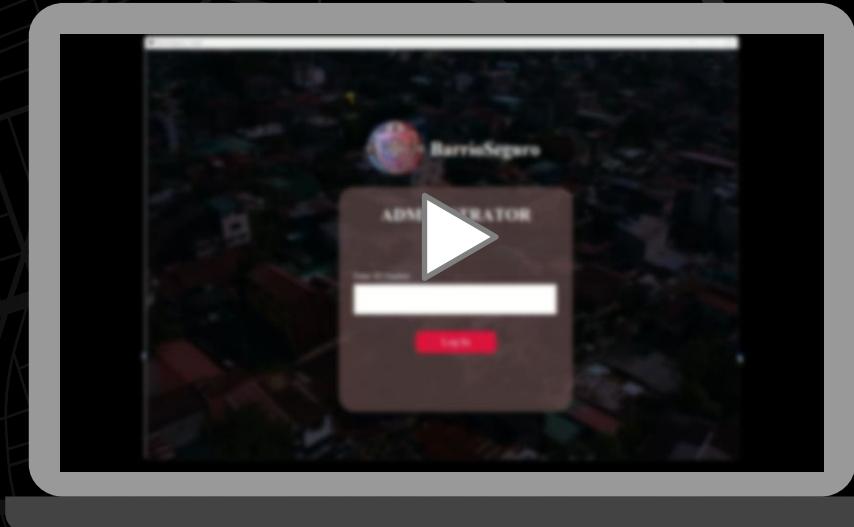
Key functionalities

- Incident Reporting Module
- Community Announcement Feature
- Resident Database Information
- Data Reports

# SYSTEM ARCHITECTURE



# RUNTIME SIMULATION VIDEO





BarrioSeguro

## ADMINISTRATOR

Enter ID Number

Log In

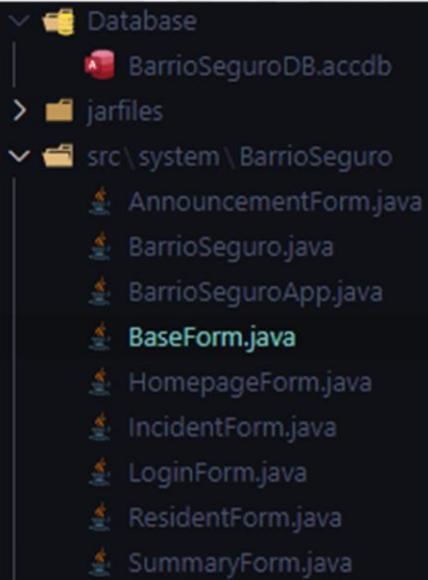
A black and white photograph of a modern skyscraper's glass facade, showing a grid of windows reflecting light.

PROGRAM  
CODE



# CONNECTION OF DATABASE

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
```



```
public abstract class BaseForm extends JFrame {
    // This is a "static final String" for the database path
    // It's a text Label that doesn't change, telling us where our database is
    protected static final String DATABASE_PATH = "jdbc:ucanaccess://Database/BarrioSeguroDB.accdb";

    // This function gives us a database connection
    // "throws SQLException" means it might show an error if we can't connect
    protected Connection getConnection() throws SQLException {
        // Uses "DriverManager" to get a connection to our Access database
        // We must have that .accdb file for it to work
        return DriverManager.getConnection(DATABASE_PATH);
    }
}
```

# MICROSOFT ACCESS DATABASE

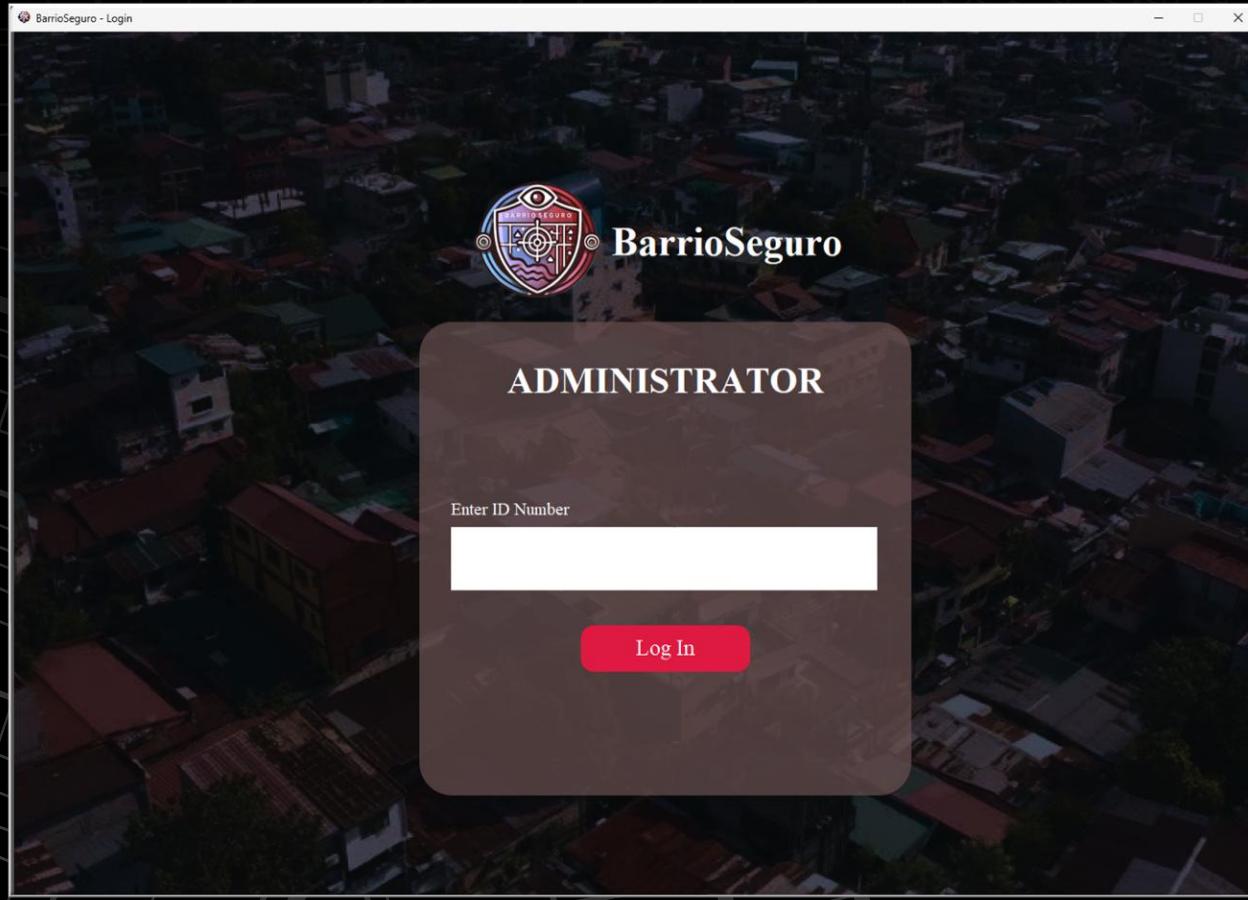
ID	idNumber	Click to Add
1	00000000	
2	12345678	
3	10203040	

resident_id	resident(firstName)	resident(midName)	resident(lastName)	resident(suffix)	resident_address	resident_DoB	resident_contact	resident_email
11 Juan Gabriel	Duque		Dela Cruz		421 Celeridad Pasay City	09/26/2003	09212653256	juangabriel.delacruz01@gmail.com
113 Linda			Wright	IV	759 Snow Land, Brgy. San Isidro, Manila	09/22/1987	0971 909 3909	empowerict.2021@gmail.com
114 Carla			Baldwin	Sr.	625 Pierce Forest, Brgy. San Isidro, Manila	01/07/2006	0939 547 3665	jejehcheng1@gmail.com
115 Diane			Davis	Jr.	60366 Nancy Ferry Suite 772, Brgy. San Isidro, Manila	03/29/1946	0935 359 5623	jejehcheng2@gmail.com
116 Mary	Tina		Howard		49024 Fields Roads, Brgy. San Isidro, Manila	03/07/1960	0979 442 8215	jejehcheng197@gmail.com
117 Chad	Shannon		Baker	III	28552 Shane Pike Apt. 388, Brgy. San Isidro, Manila	08/06/1982	0996 408 5324	j33394303@gmail.com
119 Julia	Stephanie		Terry	IV	945 Perez Streets Suite 896, Brgy. San Isidro, Manila	08/22/1954	0954 853 6306	jejehcheng8@gmail.com

incident_id	incident(firstName)	incident(midName)	incident(lastName)	incident(suffix)	incident_date	incident_type	incident_description	incident_progress	Click to Add
116 Gab			Dela Cruz		11/11/2011	Assault	Sample Description okay	Ongoing	
117 Steven			Dudley	III	06/11/2023	Physical Abuse	Board shoulder most rule four. Call coach line born	Under Investigation	
118 Robin			Newton		10/03/2023	Burglary	Source movement common. Will thank respond th	Resolved	
119 Ryan	Megan		Gonzales	Sr.	02/12/2023	Burglary	Tonight sit total throughout artist give friend go. Wo	Under Investigation	
120 William	Sarah		George	Sr.	09/11/2024	Domestic Violen	Life result to thus measure consider. Know partner	Resolved	
121 Jerry	Daniel		Hull		11/25/2023	Drug Possessio	Kid system unit common history soldier. Seven sin	Ongoing	
122 Eric			Cobb		02/11/2024	Vandalism	Commercial fill peace them fight skin. Who identify	Under Investigation	
123 Kevin			Sanders	IV	10/18/2023	Vandalism	Boy ok research public war. Budget whole age bas	Under Investigation	
124 Darius			Nelson		01/15/2024	Drug Possessio	Oil civil key indeed ten discover. Standard contain i	Under Investigation	
125 Ana			Gardner	III	03/20/2023	Vandalism	Or although once. Figure board improve firm. Many	Under Investigation	
126 Joshua	Lawrence		Diaz	III	11/04/2023	Traffic Violation	Increase type production agent much movement pi	Resolved	
127 Mitchell	Jennifer		Chavez		02/18/2023	Drug Possessio	Hear go personal short. Light her leg word. Commi	Resolved	
128 Daniel	Patricia		Torres		04/04/2023	Accident	Old patient needs to be moved to another hospital	Under Investigation	

# LOGIN FORM

BarrioSeguro - Login



The background of the login form is a dark, grainy aerial photograph of a dense urban neighborhood with many small houses and buildings.

**BarrioSeguro**

ADMINISTRATOR

Enter ID Number

Log In

The login form features a central input field for "Enter ID Number" with a placeholder text "Enter ID Number". Below the input field is a red "Log In" button. The entire form is set against a semi-transparent dark overlay that allows the background image to be seen through it. The title bar of the window shows "BarrioSeguro - Login".

# LOGIN FORM

```
// A "JPasswordField" to keep the ID secret (like a password)
// This is where the user types their ID
JPasswordField idNumberTextBox = new JPasswordField();
idNumberTextBox.setFont(new Font(name:"SansSerif", Font.PLAIN, size:30));
idNumberTextBox.setBounds(x:36, y:234, width:486, height:72);
// Add an empty border inside the text box for spacing
idNumberTextBox.setBorder(new EmptyBorder(top:10, left:20, bottom:10, right:20));
adminPanel.add(idNumberTextBox);

// We create a "Log In" button and give it rounded style
JButton loginButton = new JButton(text:"Log In");
styleRoundedButton(loginButton); // Custom styling for button
loginButton.setFont(new Font(name:"Times New Roman", Font.PLAIN, size:25));
loginButton.setBounds(x:184, y:346, width:193, height:53);

adminPanel.add(loginButton);
```

# FOREGROUND EVENT (ActionEvent)

```
// When the user clicks "Log In," check if the ID is correct
You, 2 days ago | 2 authors (You and one other)
loginButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent eventForLogIn) {
        // If "isValidID" is true, we close this form and open the homepage
        if (isValidID(new String(idNumberTextBox.getPassword()))) {
            dispose(); // Close Login form
            appController.openHomepageForm();
        } else {
            // If it's not a valid ID, show an error box
            JOptionPane.showMessageDialog(LoginForm.this, message:"Invalid ID Number",
                title:"Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});
```

# BACKGROUND METHOD

```
// This method checks if the typed ID exists in the database table named "Barangay_Official"
private boolean isValidID(String idNumber) {
    // A text command that counts how many times this ID number appears
    String query = "SELECT COUNT(*) FROM Barangay_Official WHERE idnumber = ?";

    // "try" ensures we close our connection each time
    // "Connection" is how we talk to the database
    try (Connection connectIDnumber = getConnection()) {
        // Prepare the SQL statement so we can safely set the ID
        try (PreparedStatement statementIDnumber = connectIDnumber.prepareStatement(query)) {
            // Replace the question mark with the actual ID
            statementIDnumber.setString(parameterIndex:1, idNumber);

            // Execute the query and get results
            ResultSet resultIDnumber = statementIDnumber.executeQuery();
            resultIDnumber.next();
            // Get the count from the first column. If more than 0, ID is valid
            int count = resultIDnumber.getInt(columnIndex:1);
            return count > 0; // Returns true if ID exists
        }
    } catch (SQLException handleDatabaseException) {
        // If something is wrong, we print the error and show a message box
        handleDatabaseException.printStackTrace();
        JOptionPane.showMessageDialog(LoginForm.this, "Database error: " + handleDatabaseException.getMessage(), title:"Error", JOptionPane.ERROR_MESSAGE);
    }
}

// If all else fails, we return false (invalid ID)
return false;
}
```

# Homepage

BarrioSeguro - Homepage



BarrioSeguro

Announcement

Resident Database

Incident Reports

Summary Reports

Log Out

Welcome!



BARRIOSEGURO

Barangay Monitoring and Incident Reporting System

# FOREGROUND EVENT (ActionEvent)

```
// First button is called "announceBtn" with text "Announcement"
// We use "styleButton" to make it look nice
JButton announceBtn = new JButton(text:"Announcement");
styleButton(announceBtn);
announceBtn.setBounds(x:0, y:250, width:430, height:78);

// This is what happens when we click "Announcement"
You, 2 days ago | 1 author (You)
announceBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent eventForAnnounceBtn) {
        // Move to a new screen called the Announcement Form
        // We do it with EventQueue.invokeLater to handle it safely
        You, 2 days ago | 1 author (You)
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    appController.openAnnouncementForm();
                } catch (Exception handleAnnounceException) {
                    handleAnnounceException.printStackTrace();
                }
                // Close this window after opening the next
                dispose();
            }
        });
    }
});
```

```
// "Log Out" button for leaving the program
JButton logoutBtn = new JButton(text:"Log Out");
styleRoundedButton(logoutBtn);
logoutBtn.setBounds(x:135, y:805, width:150, height:55);

// When clicked, close this form and go to the Login screen
You, 3 days ago | 1 author (You)
logoutBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent eventForSummaryBtn) {
        dispose();

        EventQueue.invokeLater(() -> {
            logoutAndGoToLogin();
        });
    }
});
```

same for “Resident Database,” “Incident Reports,” and “Summary Reports” buttons

# ANNOUNCEMENT

The image shows a screenshot of a software application window titled "BarrioSeguro - Announcements". The window has a dark background featuring a grayscale aerial photograph of a dense urban residential area. On the left side of the window, there is a vertical sidebar with four red buttons labeled "Announcement", "Resident Database", "Incident Reports", and "Summary Reports". At the bottom of this sidebar is a red "Log Out" button. The main content area contains a large, semi-transparent white box. Inside this box, at the top, is a "Subject" input field. Below it is a text input field with the placeholder "Enter a message ...". At the bottom of the white box are two red buttons: "Send to SMS" on the left and "Send to Email" on the right.

# FOREGROUND EVENT (FocusEvent and ActionEvent)

```
// We add "FocusListener" so that if the text says "Subject," it goes away when clicked
You, 2 days ago | 1 author (You)
subjectTextField.addFocusListener(new FocusListener() {
    @Override
    public void focusGained(FocusEvent eventForSubjectField) {
        // If it's still the default text, empty it and switch the color to black
        if (subjectTextField.getText().equals(anObject:"Subject")) {
            subjectTextField.setText(t:"");
            subjectTextField.setForeground(Color.BLACK);
        }
    }

    @Override
    public void focusLost(FocusEvent eventForSubjectField) {
        // If the user left the field empty, return the placeholder text
        if (subjectTextField.getText().isEmpty()) {
            subjectTextField.setText(t:"Subject");
            subjectTextField.setForeground(Color.LIGHT_GRAY);
        }
    }
});
```

same for message text area

```
// Make a button that says "Send to SMS" (not fully implemented, just a placeholder)
JButton smsBtn = new JButton(text:"Send to SMS");
styleRoundedButton(smsBtn);
You, 2 days ago | 2 authors (You and one other)
smsBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent eventForSMSbtn) {
        // If our subject and message are okay, attempt an SMS send
        if (validateFields()) {
            String subject = subjectTextField.getText();
            String messageContent = messageTextField.getText();
            sendSMS(subject, messageContent);
        }
    }
});
smsBtn.setBounds(x:167, y:497, width:177, height:55);
```

same for “Send to Email” button

# BACKGROUND METHOD

```
// "sendEmail" sends an email to the addresses in our database of residents
// It uses the JavaMail Library for sending email
private void sendEmail(String subject, String messageContent) {
    // Our Gmail address and password for sending emails
    // Must be valid, or the email won't go through
    final String username = "gabdelacruz926@gmail.com";
    final String password = "wnvmmmbowuvxtbbvr";

    // A set of properties that tell JavaMail how to talk to Gmail's server
    Properties propsGmailServer = new Properties();
    propsGmailServer.put(key:"mail.smtp.auth", value:"true");
    propsGmailServer.put(key:"mail.smtp.starttls.enable", value:"true");
    propsGmailServer.put(key:"mail.smtp.host", value:"smtp.gmail.com");
    propsGmailServer.put(key:"mail.smtp.port", value:"587");

    // An "Authenticator" helps verify we have the correct username and password
    You, 3 days ago | 1 author (You)
    Authenticator emailAuthenticator = new Authenticator() {
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(username, password);
        }
    };

    // A session is the connection to the mail server
    Session sessionForEmail = Session.getInstance(propsGmailServer, emailAuthenticator);
```

```
try {
    // Open our database and find every resident's email
    Connection connectSendEmail = getConnection();
    String query = "SELECT resident_email FROM ResidentDB";
    Statement statementSendEmail = connectSendEmail.createStatement();
    ResultSet resultSendEmail = statementSendEmail.executeQuery(query);

    // We gather all emails into a string separated by commas
    StringBuilder listEmailAddress = new StringBuilder();
    while (resultSendEmail.next()) {
        String email = resultSendEmail.getString(columnLabel:"resident_email");
        listEmailAddress.append(email).append(str:",");
    }
    // Remove the last comma
    if (listEmailAddress.length() > 0) {
        listEmailAddress.setLength(listEmailAddress.length() - 1);
    }

    // "MimeMessage" is the actual email we're sending
    Message createNewMessage = new MimeMessage(sessionForEmail);
    createNewMessage.setFrom(new InternetAddress(username));
    // "Message.RecipientType.TO" passes our list of emails
    createNewMessage.setRecipients(Message.RecipientType.TO, InternetAddress.parse(listEmailAddress.toString()));
    createNewMessage.setSubject(subject);
    createNewMessage.setText(messageContent);

    // Send the email
    Transport.send(createNewMessage);
```

# BACKGROUND METHOD (cont'd)

```
// Clear out the fields after sending
subjectTextField.setText(t:"");
messageTextField.setText(t:"");

// Close everything: result set, statement, connection
resultSendEmail.close();
statementSendEmail.close();
connectSendEmail.close();

// Show a success message
JOptionPane.showMessageDialog(AnnouncementForm.this, message:"Emails sent successfully!");

} catch (Exception handleEmailException) {
    // If something goes wrong, show an error
    handleEmailException.printStackTrace();
    JOptionPane.showMessageDialog(AnnouncementForm.this, "Failed to send emails. " + handleEmailException.getMessage());
}
```

# RESIDENT DATABASE

BarrioSeguro - Resident Database



## BarrioSeguro

Announcement

Resident Database

Incident Reports

Summary Reports

Log Out

Add

Update

Delete

Print ID

First Name	Middle Name	Last Name	Suffix	Address	Date of Birth	Contact Num.	Email
Juan Gabriel	Duque	Dela Cruz		421 Celeridad P.	26/09/2003	09212653256	juangabriel.dela...
Seth		Rojas		898 Anderson Gl.	06/09/1942	0949 7812370	seth.rojas@gm...
Linda		Wright	IV	759 Snow Land.	22/09/1987	0971 9093909	linda.wright@q...
Carla		Baldwin	Sr.	622 Pierce Fore.	07/01/2006	0939 5473665	carla.baldwin@y...
Diane		Davis	Jr.	60365 Nancy Fe.	29/03/1946	0935 3505623	diane.davis@t...
Tina		Howard		49024 Fields Ro.	07/03/1960	0979 4423213	mari.howard@h...
Chad	Shannon	Baker	III	28662 Shone Pl.	06/08/1982	0906 4085234	chad.baker@gm...
Rachel	Brittany	Jackson	III	1639 Joseph Ro.	19/09/1998	0929 6486161	rachel.jackson...
Julia	Stephanie	Terry	IV	945 Perez Street.	22/08/1954	0954 8516306	julia.terry@hot...
Rebecca		Winters	IV	9279 Santa Dale	20/10/1972	0960 1292018	rebecca.winters...
John		Reyes	III	6236 Chambers.	13/04/1995	0957 2778912	john.reyes@gm...
Sheri		Hernandez	Sr.	6390 Michael Vill...	16/04/1988	0937 3846597	sherihernandez...
Jessica	David	Mercado		844 Reed Keys...	16/11/1973	0976 9629231	jessica.mercad...
James	Antel	Miller		39329 Reed Est...	13/09/1997	0970 4381218	james.miller@q...
Steven		Bell		11726 Leslie La...	17/06/2004	0954 3303762	steven.bell@q...
Garrett		Peterson		634 Rodriguez R.	10/07/1950	0926 1837161	garrett.peterson...
Michele	Beth	Becker	IV	014 Acosta Spur.	11/08/1952	0994 9654741	michele.becker...
Daryl		Richards	Sr.	23183 Katrina W...	20/03/1990	0957 3961362	daryl.richards@...
Anna	Peter	Hunt	III	459 Moore Islan...	16/06/1963	0988 2559093	anna.hunt@gm...
Andrew		Lopez		35487 Stanley F...	23/01/1992	0998 6324607	andrew.lope@y...
Yolanda		Murray		283 Pratt Trace...	29/10/1941	0922 6315708	yolanda.murray...
Cheyenne		Howard		0369 Patrick Sho...	27/12/1990	0965 7031673	cheyenne.howar...
Dawn	Jeffrey	Vance		20656 Angela I...	05/01/1997	0945 3146206	dawn.vance@ho...
Zachary	Joseph	Gonzalez		4662 John Knoll	08/06/1957	0914 8282105	zachary.gonzale...
Ethan	Alyssa	Oconnor	IV	1671 Palmer Ro.	23/06/1990	0992 9615716	ethan.oconnor@...
Daryl		Montgomery	Sr.	82817 Turner R...	13/12/1979	0938 4604740	daryl.montgom...
Kimberly		Harvey	IV	35096 Nicholas...	25/09/1972	0929 7595585	kimberly.harvey...
Vanessa		Valdez		451 Sparks Junc...	15/09/1968	0962 9208829	vanessa.valdez...
Juan	Melissa	White	IV	6045 Henry Inlet	19/02/1982	0980 3135178	luan.white@hot...

# RESIDENT DATABASE (Search Query)

## FOREGROUND EVENTS (FocusEvent and KeyEvent)

```
// When the user clicks inside the box, if the placeholder is "Search," we remove it
You, 2 days ago | 1 author (You)
searchTextField.addFocusListener(new FocusListener() {
    @Override
    public void focusGained(FocusEvent eventForSearchField) {
        if (searchTextField.getText().equals("Search")) {
            searchTextField.setText("");
            searchTextField.setForeground(Color.BLACK);
        }
    }

    @Override
    public void focusLost(FocusEvent eventForSearchField) {
        // If the user leaves it empty, put the placeholder back
        if (searchTextField.getText().isEmpty()) {
            searchTextField.setText("Search");
            searchTextField.setForeground(Color.LIGHT_GRAY);
        }
    }
});
```

```
// This "KeyAdapter" triggers a filter function whenever the user types
You, 2 days ago | 1 author (You)
searchTextField.addKeyAdapter(new KeyAdapter() { // search function
    @Override
    public void keyReleased(KeyEvent eventKeyForSearchField) {
        // We convert everything to lowercase so searching is easier
        String searchQuery = searchTextField.getText().trim().toLowerCase();
        // Filter the table results based on this text
        filterTable(searchQuery);
    }
});
```

# RESIDENT DATABASE (Add)

BarrioSeguro - Resident Database



## BarrioSeguro

Announcement

Resident Database

Incident Reports

Summary Reports

Log Out

Add

Update

Delete

Print ID

Search

First Name	Middle Name	Last Name	Suffix	Address	Date of Birth	Contact Num..	Email
Juan Gabriel	Duque	Dela Cruz		421 Celeridad P.	26/09/2003	09212653256	juangabriel dela...
Seth	Add Resident			898 Anderson Gl	06/09/1942	0949 781 2370	seth.rojas@gm...
Linda				759 Snow Land	22/09/1987	0971 909 3909	linda.wright@o...
Carla				625 Pierce Ford	07/10/2006	0939 547 3665	carla.baldwin@y...
Diane				60365 Nancy Fe	29/03/1946	0935 359 5623	diane.davis@o...
Mary				49024 Fields Ro	07/03/1960	0979 442 8215	mary.howard@h...
Chad				28552 Shane Pl.	06/08/1982	0966 408 5324	chad.baker@pm...
Rachel				1639 Joseph Ro	19/09/1998	0929 648 6161	rachel.jackson...
Julia				945 Perez Street	22/08/1954	0954 853 6306	julia.terry@hot...
Rebecca				9279 Sara Dale	20/10/1972	0960 129 2016	rebecca.winters...
John				6236 Chambers	03/04/1995	0957 277 8912	john.reyes@gm...
Sheri				6390 Michael Vill	16/04/1988	0937 384 6597	sheril.hernandez...
Jessica				844 Reed Keys	16/11/1973	0976 962 9231	jessica.mercad...
James				39329 Reed Est.	13/09/1997	0970 438 1218	james.miller@q...
Steven				11728 Leslie La...	17/06/2004	0954 330 3762	steven.bell@gm...
Garrett				634 Rodriguez R.	10/07/1950	0926 183 7761	garrett.peterson...
Michelle				014 Acosta Spur	11/08/1952	0994 965 4741	michelle.becker...
Darryl				23183 Kahina W.	20/03/1990	0957 398 1362	darryl.richards@...
Anna				459 Moore Islan...	16/06/1963	0986 256 9093	anna.hunt@gm...
Andrew				35487 Stanley F...	23/01/1992	0998 632 4607	andrew.lopez@y...
Yolanda				283 Pratt Trace ...	29/10/1941	0922 831 5706	yolanda.martinez...
Cheyenne				039 Patrick Str...	27/12/1990	0965 703 1673	cheyenne.howard...
Down				20565 Angela T...	05/11/1997	0948 314 6208	dawn.dance@ho...
Zachary				4682 John Knoll	08/06/1957	0914 823 2105	zachary.gonzale...
Ethan				1671 Palmer Ro.	23/06/1980	0982 961 5716	ethan.connor@...
Daryl				62817 Turner R...	13/12/1979	0938 460 4740	daryl.montgomery...
Kimberly				35096 Nicholas...	25/09/1972	0929 759 5585	kimberly.harvey...
Vanessa				451 Sparks Junc...	15/09/1968	0962 920 8629	vanessa.valdez...
Juan				6045 Henn Inlet	19/02/1982	0980 313 5178	luan.white@hot...

# RESIDENT DATABASE (Add)

## FOREGROUND EVENT (ActionEvent)

```
// Make an "Add" button for adding a new resident
JButton btnAdd = new JButton(text:"Add");
styleRoundedButton(btnAdd);
// If they press "Add," we open a small form that collects the new resident's info
You, 2 days ago | I author (You)
btnAdd.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent eventForAddBtn) {
        // "addResidentInfoPanel" is a small grid layout to hold text fields
        JPanel addResidentInfoPanel = new JPanel(new GridLayout(rows:3, cols:2, hgap:10, vgap:10));
        // We Label and then create a text field for each piece of info
        addResidentInfoPanel.add(new JLabel(text:"First Name:"));
        JTextField firstNameField = new JTextField();
        addResidentInfoPanel.add(firstNameField);

        addResidentInfoPanel.add(new JLabel(text:"Middle Name (Optional):"));
        JTextField middleNameField = new JTextField();
        addResidentInfoPanel.add(middleNameField);

        addResidentInfoPanel.add(new JLabel(text:"Last Name:"));
        JTextField lastNameField = new JTextField();
        addResidentInfoPanel.add(lastNameField);

        addResidentInfoPanel.add(new JLabel(text:"Suffix (Optional):"));
        JTextField suffixField = new JTextField();
        addResidentInfoPanel.add(suffixField);

        // If the user clicks OK, we collect the text field values
        if (result == JOptionPane.OK_OPTION) {
            String firstName = firstNameField.getText().trim();
            String middleName = middleNameField.getText().trim();
            String lastName = lastNameField.getText().trim();
            String suffix = suffixField.getText().trim();
            String address = addressField.getText().trim();
            String dateOfBirth = dobField.getText().trim();
            String contact = contactField.getText().trim();
            String email = emailField.getText().trim();

            // Check if some required fields are empty
            if (firstName.isEmpty() || lastName.isEmpty() || address.isEmpty() || dateOfBirth.isEmpty() || contact.isEmpty() || email.isEmpty()) {
                // Show a warning if needed
                JOptionPane.showMessageDialog(parentComponent:null, message:"Please fill in all required fields.", title:"Validation Error", JOptionPane.ERROR_MESSAGE);
                return;
            }

            // Contact must be exactly 11 digits, or it's invalid
            if (!contact.matches(regex:"\\d{11}")) {
                JOptionPane.showMessageDialog(parentComponent:null, message:"Contact number must be 11 digits.", title:"Validation Error", JOptionPane.ERROR_MESSAGE);
                return;
            }

            // Date must follow dd/mm/yyyy
            if (!dateOfBirth.matches(regex:"\\d{2}/\\d{2}/\\d{4}")) {
                JOptionPane.showMessageDialog(parentComponent:null, message:"Date of Birth must be in the format dd/mm/yyyy.", title:"Validation Error", JOptionPane.ERROR_MESSAGE);
                return;
            }

            // Everything looks okay, so attempt to store it in the database
            addResidentToDatabase(firstName, middleName, lastName, suffix, address, dateOfBirth, contact, email);
        }
    }
});
```

# RESIDENT DATABASE (Add)

## BACKGROUND METHOD (similar to Update, but we UPDATE the query)

```
// "addResidentToDatabase" inserts a new resident into our SQL database
// It uses eight pieces of info: names, address, birth date, contact info, etc.
private void addResidentToDatabase(String firstName, String middleName, String lastName, String suffix, String address, String dateOfBirth, String contact, String email) {
    // Here is the SQL command that inserts new rows into "ResidentDB"
    // We provide values for each column: firstName, mIdName, LastName, suffix, address, DoB, contactNo, email
    String query = "INSERT INTO ResidentDB (resident(firstName, resident(middleName, resident(lastName, resident(suffix, resident(address, resident(DoB, resident(contactNo, resident(email)))))))) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

    try {
        // "getConnection" is from our "BaseForm" to open the database
        Connection connectAddResident = getConnection();
        // "PreparedStatement" helps us safely place variables into the query
        PreparedStatement prepareAddResident = connectAddResident.prepareStatement(query)
    } {
        // Convert the String date into a proper SQL date using our helper function
        java.sql.Date convertSQLdate = convertToDate(dateOfBirth); // convert date so that format is unison mm/dd/yyyy

        // Insert data into the 1st question mark for firstName, 2nd for middleName, etc.
        prepareAddResident.setString(1, firstName);
        // If "middleName" is empty, we store NULL in the database
        prepareAddResident.setString(2, middleName.isEmpty() ? null : middleName);
        prepareAddResident.setString(3, lastName);
        prepareAddResident.setString(4, suffix.isEmpty() ? null : suffix);
        prepareAddResident.setString(5, address);
        prepareAddResident.setDate(6, convertSQLdate);
        prepareAddResident.setString(7, contact);
        prepareAddResident.setString(8, email);

        // Execute the query to insert a row, and store how many rows were added
        int rowsInserted = prepareAddResident.executeUpdate();
        // If at least one row is added, we show a success message
        if (rowsInserted > 0) {
            JOptionPane.showMessageDialog(ResidentForm.this, "Resident added successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);
            // Refresh the table to show new entries
            loadResidentData();
        }
    } catch (ParseException handleParseAddResident) {
        // If the date is in the wrong format, show an error
        JOptionPane.showMessageDialog(ResidentForm.this, "Invalid Date of Birth format. Please use dd/MM/yyyy.", "Date Format Error", JOptionPane.ERROR_MESSAGE);
    } catch (SQLException handleDatabaseAddResident) {
        // If the database is unreachable or another SQL error occurs
        handleDatabaseAddResident.printStackTrace();
        JOptionPane.showMessageDialog(ResidentForm.this, "Error adding resident: " + handleDatabaseAddResident.getMessage(), "Database Error", JOptionPane.ERROR_MESSAGE);
    }
}
```

# RESIDENT DATABASE (Update)

BarrioSeguro - Resident Database



## BarrioSeguro

Announcement

Resident Database

Incident Reports

Summary Reports

Log Out

Add

Update

Delete

Print ID

Search

First Name	Middle Name	Last Name	Suffix	Address	Date of Birth	Contact Num.	Email
Mark	James	Gonzalez	Sr.	1002 Charles Vill...	17/07/1940	0955 267 6054	mark.gonzalez...
Nathan		Ivan		10048 Melinda La...	26/06/1948	0945 963 8751	nathan.jensen@...
Amanda	Rachel	Allen		4 Brandon C...	17/01/1943	0986 717 9130	amanda.allen@...
Jessica	Cynthia	Erickson		2941 Lindsey Dr...	22/09/1966	0977 231 7001	jessica.erickson...
Erin		Kline		294 Angela Path...	29/05/1949	0937 247 6148	erin.kline@yahoo...
John		Hoffman		1835 Bell Vista...	28/03/1936	0930 480 4824	john.hoffman@g...
Sherri		Callahan		88328 Kayla For...	06/03/1996	0967 511 9077	sherri.callahan...
Destiny		Kramer		4059 Hayes Vist...	12/03/1982	0928 864 2846	destiny.kramer...
Faramis	Luo Yi	Mathilda	Jr.	123 Dawn Brgy. Pin...	01/01/1999	09663593155	faramis2004@gmail.com

Update Resident

First Name: Faramis

Middle Name (Optional): Luo Yi

Last Name: Mathilda

Suffix (Optional): Jr.

Address: 123 Dawn Brgy. Pinagbuhatan, Manila

Date of Birth (dd/MM/yyyy): 01/01/1999

Contact Number (11 digits): 09663593155

Email: faramis2004@gmail.com

OK Cancel

# RESIDENT DATABASE (Update)

## FOREGROUND EVENT (ActionEvent)

```
// A button to update an existing record
JButton btnUpdate = new JButton(text:"Update");
styleRoundedButton(btnUpdate);
You, 2 days ago | 1 author (You)
btnUpdate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent eventForUpdateBtn) {
        int selectedRow = residentTable.getSelectedRow();

        // We check if a row is selected; if not, alert the user
        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(parentComponent:null, message:"Please select a resident to update.", title:"No Selection", JOptionPane.WARNING_MESSAGE);
            return;
        }

        // We gather data from the selected row
        int residentId = (Integer) residentTable.getValueAt(selectedRow, column:0);
        String firstName = (String) residentTable.getValueAt(selectedRow, column:1);
        String middleName = (String) residentTable.getValueAt(selectedRow, column:2);
        String lastName = (String) residentTable.getValueAt(selectedRow, column:3);
        String suffix = (String) residentTable.getValueAt(selectedRow, column:4);
        String address = (String) residentTable.getValueAt(selectedRow, column:5);
        String dateOfBirth = residentTable.getValueAt(selectedRow, column:6).toString();
        String contact = (String) residentTable.getValueAt(selectedRow, column:7);
        String email = (String) residentTable.getValueAt(selectedRow, column:8);
```

```
// Try to update the database, or show an error if something goes wrong
try {
    updateResidentInDatabase(residentId, updatedFirstName, updatedMiddleName, updatedLastName, updatedSuffix, updatedAddress, updatedDob, updatedContact, updatedEmail);
} catch (Exception ex) {
    JOptionPane.showMessageDialog(parentComponent:null, "Error updating resident: " + ex.getMessage(), title:"Update Error", JOptionPane.ERROR_MESSAGE);
}
```

# RESIDENT DATABASE (Delete)

BarrioSeguro - Resident Database



## BarrioSeguro

Announcement

Resident Database

Incident Reports

Summary Reports

Log Out

Add

Update

Delete

Print ID

Search

First Name	Middle Name	Last Name	Suffix	Address	Date of Birth	Contact Num.	Email
Juan Gabriel	Duque	Dela Cruz		421 Celeridad P.	26/09/2003	09212653256	juangabrieldela...
Seth		Rojas		898 Anderson Gl.	06/09/1942	0949 781 2370	seth.rojas@gm...
Linda		Wright	IV	759 Snow Land.	22/09/1987	0971 909 3909	linda.wright@o...
Carla		Baldwin	Sr.	625 Pierce Fore.	07/10/2006	0939 547 3665	carla.baldwin@y...
Diane		Davis	Jr.	60368 Nancy Fe.	29/03/1946	0935 369 5623	diane.davis@q...
Tina		Howard		49024 Fields Ro.	07/03/1960	0979 442 8215	mari.howard@h...
Chad		Shannon	Baker	III	08/08/1982	0906 408 6234	chad.baker@gm...
Rachel		Brittany	Jackson	III			
Julia							
Rebecca							
John							
Sheri							
Jessica							
James							
Steven							
Garrett							
Michele	Beth	Becker	IV	945 Perez Street.	22/08/1954	0954 852 6306	julia.terry@hot...
Daryl		Richards	Sr.	9279 Sara Dale	20/10/1972	0960 129 2018	rebecca.winters...
Anna		Peter	III	6236 Chambers	03/04/1995	0957 277 8912	john.reyes@gm...
Andrew		Hunt		6390 Michael Vill	16/04/1988	0937 384 6597	sheri.hernandez...
Yolanda		Lopez		844 Reed Keys	16/11/1973	0976 982 9231	jessica.mercad...
Cheyenne		Murray		39329 Reed Est.	13/09/1997	0970 436 1218	jesus.miller@p...
Dawn		Howard		11728 Leslie La.	17/06/2004	0954 330 3762	steven.bell@gn...
Zachary	Jeffrey	Vance		634 Rodriguez R.	10/07/1950	0928 182 7161	garrett.peterson...
Joseph		Becker		0114 Acosta Spur.	11/08/1952	0994 865 4741	michele.becker...
Ethan	Daryl	Gonzalez		23183 Katrina W.	20/03/1990	0957 398 1382	daryl.richards@o...
Alyssa		Oconnor	IV	450 Moon Island	16/06/1963	0988 255 9093	anna.hunt@gm...
				23047 Stanley F.	23/01/1992	0998 632 4987	andrew.lopez@y...
				283 Pratt Trace	29/10/1941	0922 921 5706	yolanda.murray...
				030 Patrick Sho.	27/12/1990	0990 703 6573	cheyenne.howar...
				20556 Angela T.	05/01/1997	0948 314 6208	dawn.vance@ho...
				4662 John Knoll	08/06/1957	0914 828 2105	zachary.gonzale...
				1671 Palmer Ro.	23/06/1990	0992 961 5716	ethan.oconnor@...
				82817 Turner R.	13/12/1979	0938 460 4740	daryl.montgom...
				35096 Nicholas	25/09/1972	0929 750 5585	kimberly.harvey...
				451 Sparks Junc.	15/09/1968	0962 920 8829	vanesa.valdez...
				6045 Henry Inlet	19/02/1982	0980 313 5178	luan.white@hot...

Confirm Deletion

Are you sure you want to delete this resident?

Yes No

# RESIDENT DATABASE (Delete)

## FOREGROUND EVENT (ActionEvent)

```
// A button to remove a resident record
JButton btnDel = new JButton(text:"Delete");
styleRoundedButton(btnDel);
You, 2 days ago | 1 author (You)
btnDel.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent eventForDeleteBtn) {
        int selectedRow = residentTable.getSelectedRow();

        // Check if one row is selected
        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(parentComponent:null, message:"Please select a resident to delete.", title:"No Selection", JOptionPane.WARNING_MESSAGE);
            return;
        }

        // Get the ID for that row
        int residentId = (int) residentTable.getValueAt(selectedRow, column:0);

        // Ask the user if they want to delete it for sure
        int confirm = JOptionPane.showConfirmDialog(parentComponent:null, message:"Are you sure you want to delete this resident?", title:"Confirm Deletion", JOptionPane.YES_NO_OPTION);

        if (confirm == JOptionPane.YES_OPTION) {
            // Actually delete the selected row from the database
            deleteResidentFromDatabase(residentId);
        }
    }
});
```

similar for Print ID

# INCIDENT REPORTS

The image shows a screenshot of a Windows application window titled "BarrioSeguro - Incident Reports". The window has a dark theme with a city map background. On the left, there's a sidebar with four red buttons: "Announcement", "Resident Database", "Incident Reports" (which is highlighted in white), and "Summary Reports". At the bottom of the sidebar is a red "Log Out" button. The main area contains an "Incident Reports" form. It includes fields for "Enter First Name", "Enter Middle Name (empty if none)", "Enter Last Name", "Enter Suffix (empty if none)", "Enter Date: DD/MM/YYYY", "Choose a type of Incident" (with "Assault" selected), "Incident Progress" (with "Under Investigation" selected), and a large text area labeled "Enter a message...". A red "Submit" button is at the bottom right of the form.

BarrioSeguro

Announcement

Resident Database

Incident Reports

Summary Reports

Log Out

Enter First Name

Enter Middle Name (empty if none)

Enter Last Name

Enter Suffix (empty if none)

Enter Date: DD/MM/YYYY

Choose a type of Incident

Assault

Incident Progress

Under Investigation

Enter a message...

Submit

# FOREGROUND EVENTS (FocusEvent and ActionListener)

```
incidentDescription.addFocusListener(new FocusListener() {  
    @Override  
    public void focusGained(FocusEvent eventForIncidentDesc) { // When the text area is clicked  
        if (incidentDescription.getText().equals(anObject:"Enter a message...")) {  
            incidentDescription.setText(t:""); // Clear the default text  
            incidentDescription.setForeground(Color.BLACK); // Set text color to black  
        }  
    }  
  
    @Override  
    public void focusLost(FocusEvent eventForIncidentDesc) { // When the text area Loses focus  
        if (incidentDescription.getText().isEmpty()) {  
            incidentDescription.setText(t:"Enter a message..."); // Reset to default text  
            incidentDescription.setForeground(Color.LIGHT_GRAY); // Set text color to gray  
        }  
    }  
});
```

```
submitbtn.addActionListener(eventForSubmitBtn -> { // When the button is clicked  
    if (validateIncidentDescription()) { // Check if description is valid  
        submitIncident(); // Submit the incident  
    }  
});
```

# BACKGROUND METHOD

```
// Method for submitting the incident form to the database
private void submitIncident() {
    if (!validateFormFields()) { // Check if required fields are filled
        return;
    }

    try {
        String dateString = incidentDate.getText(); // Get date from text field
        SimpleDateFormat formatDateString = new SimpleDateFormat(pattern:"dd/MM/yyyy"); // Date format we want
        formatDateString.setLenient(lenient:false); // Exact format is expected

        Date parsedDate = null;
        try {
            parsedDate = formatDateString.parse(dateString); // Try to convert text to date
        } catch (Exception handleParsedDateException) {
            JOptionPane.showMessageDialog(this, message:"Invalid date format. Please use dd/MM/yyyy.");
            return;
        }

        java.sql.Date convertedSQLdate = new java.sql.Date(parsedDate.getTime()); // Convert to SQL date

        try (Connection connectSubmitIncident = getConnection()) { // Establish database connection
            You,
            String checkQuery = "SELECT COUNT(*) FROM IncidentDB WHERE incident(firstName = ? " +
                "AND incident.lastName = ? AND incident.date = ?"; // SQL query
            try (PreparedStatement checkStmt = connectSubmitIncident.prepareStatement(checkQuery)) {
                // Fill the SQL query with form data
                checkStmt.setString(parameterIndex:1, incidentFirstName.getText());
                checkStmt.setString(parameterIndex:2, incidentLastName.getText());
                checkStmt.setDate(parameterIndex:3, convertedSQLdate);
            }
        }
    }
}
```

# SUMMARY REPORTS

BarrioSeguro - Summary Reports



## BarrioSeguro

Announcement

Resident Database

Incident Reports

Summary Reports

Log Out

Search

First Name	Last Name	Date	Progress
Gab	Dela Cruz	11/11/2011	Ongoing
Steven	Dudley	11/06/2023	Under Investigation
Robin	Newton	03/10/2023	Resolved
Ryan	Gonzales	12/02/2023	Under Investigation
William	George	11/09/2024	Resolved
Jeny	Hull	25/11/2023	Ongoing
Eric	Cobb	11/02/2024	Under Investigation
Kevin	Sanders	18/10/2023	Under Investigation
Darius	Nelson	15/01/2024	Under Investigation
Ana	Gardner	20/03/2023	Under Investigation
Joshua	Diaz	04/11/2023	Resolved
Mitchell	Chavez	18/02/2023	Resolved
Becky	Torres	21/01/2023	Under Investigation
Kaitlyn	Dunn	06/05/2024	Resolved
Samuel	Gilbert	23/04/2023	Resolved
Stephanie	Luna	31/03/2024	Ongoing
Jennifer	Hartman	01/04/2023	Closed
Kathleen	Wheeler	23/04/2023	Closed
Fernando	Chambers	27/10/2024	Closed
Michael	Vasquez	11/12/2023	Resolved
Tony	Anderson	30/04/2023	Closed
Brad	Sexton	22/08/2024	Resolved
Marcus	Pena	08/10/2023	Resolved
Laura	Moody	12/10/2024	Closed
Jennifer	Carter	15/08/2024	Closed
Jason	Garcia	26/05/2024	Resolved
Karl	Goodwin	20/09/2023	Resolved
Mark	Chaney	20/11/2023	Closed
Vanessa	Beltran	21/11/2023	Resolved
Nicholas	Hicks	03/04/2024	Closed
Eric	Drake	13/04/2023	Under Investigation
Angela	Boyle	17/11/2024	Resolved
Michael	Combs	19/12/2024	Closed
Michael	Schneider	16/03/2024	Under Investigation

VIEW

# FOREGROUND EVENT (ActionEvent)

```
// Setting up the button to view selected row details
JButton viewBtn = new JButton("VIEW");
styleRoundedButton(viewBtn); // Custom method to style the button
viewBtn.addActionListener(new ActionListener() { // Adding an action Listener for the button
    public void actionPerformed(ActionEvent eventForViewBtn) { // When the button is clicked
        int selectedRow = incidentTable.getSelectedRow(); // Get the selected row in the table

        if (selectedRow == -1) { // If no row is selected
            JOptionPane.showMessageDialog(null, "Please select an incident row to view.", "No Selection", JOptionPane.WARNING_MESSAGE);
            return; // Show a message and return
        }

        if (selectedRow != -1) { // If a row is selected
            // Get details from the selected row
            String firstName = (String) incidentTable.getValueAt(selectedRow, 0);
            String lastName = (String) incidentTable.getValueAt(selectedRow, 1);
            String dateString = (String) incidentTable.getValueAt(selectedRow, 2);
            String progress = (String) incidentTable.getValueAt(selectedRow, 3);

            SimpleDateFormat formatDateConverter = new SimpleDateFormat("dd/MM/yyyy"); // Date format for conversion
            java.sql.Date giveDate = null;
            try {
                java.util.Date convertedDate = formatDateConverter.parse(dateString);
                giveDate = new java.sql.Date(convertedDate.getTime()); // Convert text to date
            } catch (Exception handleDateException) {
                JOptionPane.showMessageDialog(null, "Invalid date format."); // Show error if date is wrong
                return;
            }
        }
    }
});
```

# FOREGROUND EVENT (ActionEvent) [cont'd]

```
try (Connection connectSummary = getConnection() { // Establish database connection
    // SQL query to get the incident description and type
    String query = "SELECT incident_description, incident_type FROM IncidentDB " +
                   "WHERE incident(firstName = ? AND incident.lastName = ? AND incident_date = ?";

    // Fill the SQL query with row details
    try (PreparedStatement statementSummary = connectSummary.prepareStatement(query)) {
        statementSummary.setString(1, firstName);
        statementSummary.setString(2, lastName);
        statementSummary.setDate(3, giveDate);
        ResultSet resultSummary = statementSummary.executeQuery(); // Execute the query

        if (resultSummary.next()) { // If result found
            description = resultSummary.getString("incident_description"); // Get the description
            typeOfIncident = resultSummary.getString("incident_type"); // Get the type of incident
        } else {
            JOptionPane.showMessageDialog(null, "No matching record found."); // Show error if no result
        }
    }
} catch (SQLException handleViewDatabaseException) {
    handleViewDatabaseException.printStackTrace(); // Print any database errors
}

IncidentForm createIncidentForm = new IncidentForm(appController); // Create the incident form

// Fill the incident form with data from the selected row
createIncidentForm.fillData(firstName, "", lastName, "", dateString, progress, description, typeOfIncident);

createIncidentForm.setVisible(true); // Show the incident form

dispose(); // Close the summary form
}
});
```

# REFERENCES

- [1] C. O. Carpio, "Barangay management system," *Int. J. Multidisciplinary Res. Publ. (IJMRAP)*, vol. 3, no. 2, pp. 26–32, 2020. [Online]. Available: <http://ijmrapp.com/wp-content/uploads/2020/07/IJMRAP-V3N1P78Y20.pdf>
- [2] Department of the Interior and Local Government, "Engaging the Barangays: Guide for Mayors," DILG Regional Office No. 5, 2022. [Online]. Available: <https://region5.dilg.gov.ph/lgrrc/wp-content/uploads/2022/07/Engaging-the-Barangays-Guide-for-Mayors-rev2.pdf>
- [3] G. A. Empinado *et al.*, "Strengthening community safety awareness of barangay public safety officers on their duties and responsibilities," *Int. J. Recent Adv. Multidisciplinary Res.*, vol. 11, no. 1, pp. 9436–9442, 2024. [Online]. Available: <https://ijramr.com/sites/default/files/issues-pdf/5015.pdf>
- [4] L. Antonio, "Enhancing barangay justice system through the development of a web-based crime monitoring module," *SSRN*, 2020. [Online]. Available: <https://doi.org/10.2139/ssrn.3642023>
- [5] E. Bangad *et al.*, "Environmental factors affecting incidence of crime in a high-risk barangay of a commercial town," *EasyChair Preprint*, Oct. 2024. [Online]. Available: <https://login.easychair.org/publications/preprint/dbbH>
- [6] J. Sigales, "Index crime rate in PH down by 61.87% from 2022 to 2024 – PNP," *Inquirer.net*, Oct. 29, 2024. [Online]. Available: <https://newsinfo.inquirer.net/1998778/>
- [7] L. H. Lubomski, P. J. Pronovost, D. A. Thompson, C. G. Holzmueller, T. Dorman, L. L. Morlock, F. Dickman, M. Fahey, and A. W. Wu, "Building a better incident reporting system: Perspectives from a multisite project," *Journal of Science Communication*, vol. 11, no. 5, May 2004. Available: <https://www.researchgate.net/publication/265103821>.
- [8] C. Macrae, "The problem with incident reporting," *BMJ Quality & Safety*, vol. 25, no. 2, pp. 71–75, 2016, doi: 10.1136/bmjqqs-2015-004732. Available: <http://qualitysafety.bmjjournals.com/content/25/2/71>.



THANK YOU ALL FOR  
LISTENING!  
God bless us all.



# BarrioSeguro

Barangay Monitoring and Incident Reporting System

Dela Cruz, Juan Gabriel  
Estonilo, Julius Evan  
BSCS 3-3

