

## Documentation Report

This document provides an overview of the development process for the Medication Tracker project. It includes essential components such as prototype screenshots, database schema, authentication and session management code snippets, and challenges we encountered with solutions applied.

### 1. Screenshot of the Login and Dashboard

#### Login Page:



A screenshot of a login dialog box. The dialog has a title bar labeled "Login". Inside, there are two input fields: "Username:" and "Password:". Below the password field is a "Login" button.



A screenshot of a dialog box titled "Input". It contains a label "Enter Patient Name:" followed by a single-line text input field. At the bottom, there are two buttons: "OK" and "Cancel".

## Dashboard:

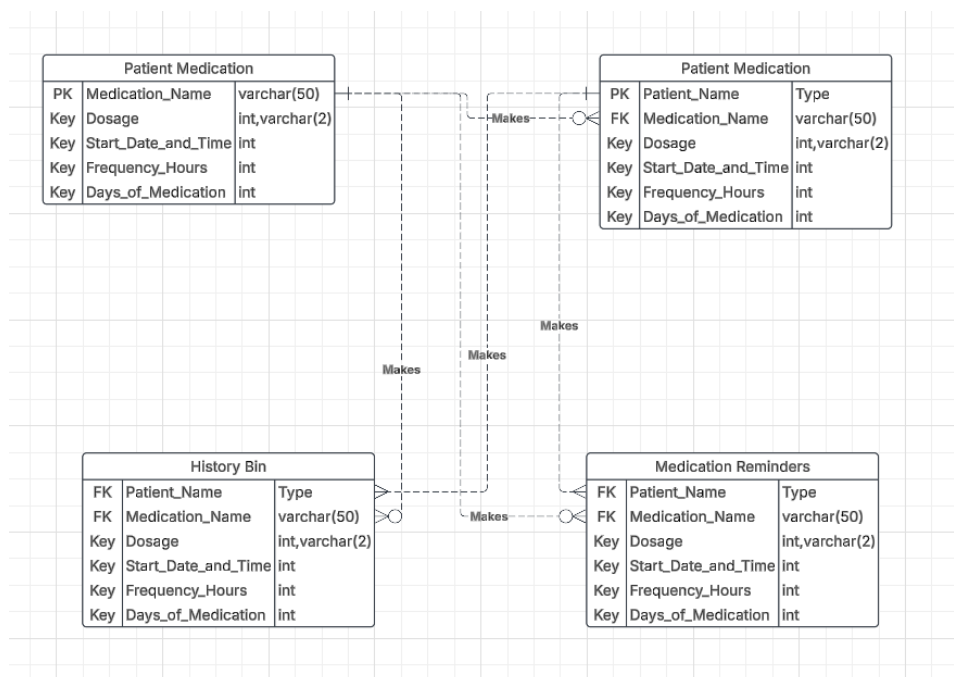
Patient: Ron

History Bin

Add Medication

## 2. Database Schema Diagram

Below is the database schema diagram which illustrates the structure of tables and their relationships.



### 3. Code Snippets for Authentication and Session Management

#### Authentication Code:

```
private void authenticateUser() {

    String username = usernameField.getText();

    String password = new String(passwordField.getPassword());

    Authentication auth = new Authentication();

    if (auth.login(username, password)) {

        openMedicationTracker(username);

    } else {

        JOptionPane.showMessageDialog(this, "Invalid username or password.", "Login
Failed", JOptionPane.ERROR_MESSAGE);

    }

}

public class Authentication {

    private final String storedDoctorPassword = BCrypt.hashpw("1234",
BCrypt.gensalt());

    private final String storedSecretaryPassword = BCrypt.hashpw("5678",
BCrypt.gensalt());

    public boolean login(String username, String password) {

        if ("Doctor".equals(username) && BCrypt.checkpw(password,
storedDoctorPassword)) {
```

```

        return true;

    } else if ("Secretary".equals(username) && BCrypt.checkpw(password,
storedSecretaryPassword)) {

        return true;

    }

    return false;

}
}

```

### **Session Management Code:**

```

package medicationtracker;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class LoginForm extends JFrame {

    private JTextField usernameField;

    private JPasswordField passwordField;

    private JButton loginButton;

    public LoginForm() {

        setTitle("Login");

        setSize(400, 200);
    }
}

```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
setLayout(new GridBagLayout());
```

```
GridBagConstraints gbc = new GridBagConstraints();  
gbc.insets = new Insets(5, 5, 5, 5);
```

```
JLabel userLabel = new JLabel("Username:");  
gbc.gridx = 0;  
gbc.gridy = 0;  
add(userLabel, gbc);
```

```
usernameField = new JTextField(15);  
gbc.gridx = 1;  
add(usernameField, gbc);
```

```
JLabel passLabel = new JLabel("Password:");  
gbc.gridx = 0;  
gbc.gridy = 1;  
add(passLabel, gbc);
```

```
passwordField = new JPasswordField(15);  
gbc.gridx = 1;  
add(passwordField, gbc);
```

```
loginButton = new JButton("Login");
```

```
gbc.gridx = 1;
```

```
gbc.gridy = 2;
```

```
add(loginButton, gbc);
```

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

```
    @Override
```

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

```
        authenticateUser();
```

```
    }
```

```
});
```

```
setLocationRelativeTo(null);
```

```
setVisible(true);
```

```
}
```

```
private void authenticateUser() {
```

```
    String username = usernameField.getText();
```

```
    String password = new String(passwordField.getPassword());
```

```
    if ("Doctor".equals(username) && "1234".equals(password)) {
```

```
        openMedicationTracker("Doctor");
```

```

    } else if ("Secretary".equals(username) && "5678".equals(password)) {
        openMedicationTracker("Secretary");
    } else {
        JOptionPane.showMessageDialog(this, "Invalid username or password.", "Login
Failed", JOptionPane.ERROR_MESSAGE);
    }
}

private void openMedicationTracker(String role) {
    dispose();

    String patientName = JOptionPane.showInputDialog(this, "Enter Patient Name:");
    if (patientName != null && !patientName.trim().isEmpty()) {
        new MedicationTracker(patientName);
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new LoginForm());
}
}

```

#### **4. Challenges encountered and solutions applied**

First challenge we have encountered was the login system displayed an "Invalid username or password" message even when entering the correct credential. This is because the password was stored in a hashed format using BCrypt, but the stored

password was being compared incorrectly. So, we provided a solution by ensuring that BCrypt's checkpw method was used properly to compare hashed passwords instead of direct string comparison. Next challenge was, the system threw an IllegalArgumentException: Invalid salt version when attempting to authenticate users. Since the stored password hash was either incompatible with the current BCrypt version or not properly generated. So, we regenerated password hashes using a consistent BCrypt version and verified that stored hashes were correctly formatted. Lastly, creating an executable JAR file that included all dependencies and resources. Since some dependencies were not included in the JAR, causing runtime errors. We used NetBeans' "Clean and Build" function to package all necessary dependencies into the JAR file.

## **Conclusion**

To conclude, this report summarizes the development process of the Medication Tracker project, providing details on authentication, session management, database design, and challenges faced. The completed system is now successfully documented and stored in the GitHub repository for version control and future improvements.