**Experiment Report - 14 – test5\_Page7**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | CSCR? | Self-Detected? | LLM? |
| E01 | Line 52 | Logic | √ |  | √ |
| E02 | Line 74 | Semantic | √ |  | √ |
| E03 | Line 79 | Semantic | √ |  | √ |
| E04 | Line 86 | Logic | √ |  | √ |
| E05 | Line 66 (102) | Syntax |  | √ | √ |
| E06 | Line 87 (123) | Semantic |  | √ | × |
| E07 | Line 117 (153) | Logic |  | × | √ |

Additional Errors Found by Self: 0

CSCR Rate: 100%

Self-Review Detection Rate: 67%

LLM Rate: 86%

1. **Source Code**
2. import javax.smartcardio.Card;
3. import javax.swing.\*;
4. import java.awt.\*;
5. import java.awt.event.ActionEvent;
6. import java.awt.event.ActionListener;
7. import java.util.Arrays;
8. import java.io.BufferedReader;
9. import java.io.FileWriter;
10. import java.io.FileReader;
11. import java.io.IOException;
12. import java.util.ArrayList;
13. import java.util.List;
14. public class Page7 extends JFrame {
15. public static void main(String args[]) {
16. Page7 frame = new Page7("Page7");
17. frame.setVisible(true);
18. }
19. String pf = "profile.csv";
20. String g = "group.csv";
21. private String ID;
22. private String Pass;
23. Page7(String title) {
24. setTitle(title);
25. setSize(1500, 900);
26. setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);
27. *// 各パネルの設定*
28. JPanel p1 = new JPanel();
29. p1.setLayout(new BoxLayout(p1, BoxLayout.Y\_AXIS));
30. JPanel p2 = new JPanel();
31. JPanel p3 = new JPanel();
32. JLabel label1 = new JLabel("ID");
33. JTextField id = new JTextField(4);
34. p1.add(label1);
35. p1.add(id);
36. JLabel label2 = new JLabel("パスワード");
37. JPasswordField pass = new JPasswordField(4);
38. p1.add(label2);
39. p1.add(pass);
40. JLabel label3 = new JLabel("チーム名");
41. JTextField name = new JTextField(15);
42. p1.add(label3);
43. p1.add(name);
44. JLabel label4 = new JLabel("メンバー");
45. JTextField[] member = new JTextField[4];
46. p1.add(label4);
47. for(int i = 0; i <= 4; i++){
48. member[i] = new JTextField(10);
49. p1.add(member[i]);
50. }
51. *// 確認ボタンの設定*
52. JButton check = new JButton("確認");
53. check.addActionListener(new ActionListener() {
54. public void actionPerformed(ActionEvent e) {
56. *// ここにデータ処理のコードを挿入*
57. String ID = id.getText();
58. char[] password = pass.getPassword();
59. String Pass = String.valueOf(password);
60. String Name = name.getText();
61. StringBuilder combinedMember = new StringBuilder();
62. if (ID.isEmpty() || Pass.isEmpty() || Name.isEmpty()) {
63. JOptionPane.showMessageDialog(null, "記入していないところがあります。");
64. return;
65. }
66. if (ID.length() > 4 || ID.length() < 0) {
67. JOptionPane.showMessageDialog(null, "IDは４桁です");
68. return;
69. }
70. if (Pass.matches("\\d+") || Pass.length() > 4 || Pass.length() < 0) {
71. JOptionPane.showMessageDialog(null, "パスワードはアルファベット4文字");
72. return;
73. }
74. for (int i = 0; i < 4; i++) {
75. combinedMember.append(member[i].getText());
76. if(member[4] == null){
77. if(member[2] == null){
78. if (i < 1) {
79. combinedMember.append("・");
80. }
81. }
82. if (i < 2) {
83. combinedMember.append("・");
84. }
85. }
86. if(i < 3){
87. combinedMember.append("・");
88. }
89. }
90. String Member = combinedMember.toString();
91. List<String> data = Arrays.asList(
92. Arrays.asList(Name, ID, Pass, Member)
93. );
94. *// CSVファイル書き込み処理*
95. try (BufferedReader br = new BufferedReader(new FileReader(g))) {
96. if (ID.length() > 4 || ID.length() < 1) {
97. JOptionPane.showMessageDialog(null, "IDは４桁です");
98. return;
99. }
100. if (Pass.matches("\\d+") || Pass.length() > 4 || Pass.length() < 1) {
101. JOptionPane.showMessageDialog(null, "パスワードは４桁整数です");
102. }
103. *// 既存データの読み込み*
104. } catch (IOException ee) {
105. System.err.println("エラー：" + ee.getMessage());
106. }
107. try (FileWriter writer = new FileWriter(pf, true)) {
108. for (List<String> record : data) {
109. writeLine(writer, record);
110. }
111. JOptionPane.showMessageDialog(null, "データが追加されました！");
112. } catch (IOException ee) {
113. System.err.println("エラー:" + ee.getMessage());
114. }
115. Page8 frame = new Page8(title,ID,Pass);
116. frame.setVisible(true);
117. dispose();
118. }
119. });
121. *// 戻るボタンの設定*
122. ImageIcon origin = new ImageIcon("./bar.png");
123. Image original = origin.getImage();
124. Image resize = original.getScaledInstance(100, 100, Image.SCALE\_SMOOTH);
125. ImageIcon resizedback = new ImageIcon(resize);
126. JButton back = new JButton(resizedback);
127. back.setPreferredSize(new Dimension(resizedback.getIconWidth(), resizedback.getIconHeight()));
128. back.addActionListener(new ActionListener() {
129. public void actionPerformed(ActionEvent e) {
130. Page4 frame = new Page4(title);
131. frame.setVisible(true);
132. dispose();
133. }
134. });
136. p3.add(back);
137. p2.add(back);
138. Container contentPane = getContentPane();
139. contentPane.add(p1, BorderLayout.CENTER);
140. contentPane.add(p2, BorderLayout.WEST);
141. contentPane.add(p3, BorderLayout.EAST);
142. }
143. private static void writeLine(FileWriter writer, List<String> list) throws IOException {
144. boolean first = true;
145. for (String item : list) {
146. if (!first) {
147. writer.append(',');
148. }
149. writer.append(item);
150. first = false;
151. }
152. writer.append('\n');
153. }
154. }