**Experiment Report - 11 – test2\_Page6**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | CSCR? | Self-Detected? | LLM? |
| E01 | Line 62 | Semantic | √ |  | √ |
| E02 | Line 74 | Syntax | √ |  | √ |
| E03 | Line 41 (93) | Syntax |  | √ | √ |
| E04 | Line 71 (123) | Semantic |  | √ | √ |

Additional Errors Found by Self: 0

CSCR Rate: 100%

Self-Review Detection Rate: 100%

LLM Rate: 100%

1. **Source Code**
2. import javax.swing.\*;
3. import java.awt.\*;
4. import java.awt.event.ActionEvent;
5. import java.awt.event.ActionListener;
6. import java.io.BufferedReader;
7. import java.io.FileWriter;
8. import java.io.FileReader;
9. import java.io.IOException;
10. import java.util.ArrayList;
11. import java.util.Arrays;
12. import java.util.List;
13. public class Page6 extends JFrame {
14. private String Name;
15. private String Pass;
16. String user = "udata.csv";
17. public Page6(String title, String Name, String Pass) {
18. this.Name = Name;
19. this.Pass = Pass;
20. setTitle(title);
21. setSize(1500, 900);
22. setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);
23. JPanel p1 = new JPanel();
24. JPanel p2 = new JPanel();
25. JPanel p3 = new JPanel();
26. JTextField nameField = new JTextField(15);
27. JLabel label1 = new JLabel("名前");
28. JTextField mail = new JTextField(30);
29. JLabel label2 = new JLabel("メールアドレス");
30. JPasswordField passField = new JPasswordField(15);
31. JLabel label3 = new JLabel("パスワード");
32. JTextField address = new JTextField(30);
33. JLabel label4 = new JLabel("住所");
34. JCheckBox[] gender = new JCheckBox[2];
35. gender[0] = new JCheckBox("男");
36. gender[1] = new JCheckBox("女");
37. JTextField birth = new JTextField(30);
38. JLabel label5 = new JLabel("生年月日");
39. JTextField job = new JTextField(30);
40. JLabel label6 = new JLabel("職業");
41. JButton check = new JButton("確認");
42. check.addActionListener(new ActionListener() {
43. public void actionPerformed(ActionEvent e) {
44. String newName = nameField.getText();
45. String newMail = mail.getText();
46. char[] password = passField.getPassword();
47. String newPass = String.valueOf(password);
48. String newAddress = address.getText();
49. String newGender = gender[0].isSelected() ? gender[1].getText() : gender[0].getText();
50. String newBirth = birth.getText();
51. String newJob = job.getText();
52. List<String> newData = Arrays.asList(newName, newPass, newMail, newAddress, newBirth, newGender, newJob);
53. List<String[]> existingData = new ArrayList<>();
54. try (BufferedReader br = new BufferedReader(new FileReader(user))) {
55. String line;
56. while ((line = br.readLine()) != null) {
57. String[] fields = line.split(",");
58. *// メールアドレスが既存のデータと重複していないか確認*
59. if (fields[1].equals(newMail)) {
60. MailCheck = true;
61. break;
62. }
64. *// 名前とパスワードの両方が一致するかを確認*
65. if (fields[0].equals(newName) && fields[1].equals(newPass)) {
66. NamePassCheck = true;
67. break;
68. }
69. if (fields[0].equals(Name) && fields[1].equals(Pass)) {
70. *// 一致する行を新しいデータで置き換え*
71. existingData.add(newData.toArray(new String[0]));
72. } else {
73. existingData.add(fields);
74. }
75. }
76. } catch (IOException ee) {
77. System.err.println("エラー：" ee.getMessage());
78. }
79. *// データをファイルに上書き保存*
80. try (FileWriter writer = new FileWriter(user)) {
81. for (String[] record : existingData) {
82. writeLine(writer, Arrays.asList(record));
83. }
84. } catch (IOException ee) {
85. System.err.println("エラー:" + ee.getMessage());
86. }
87. Page4 frame = new Page4(newName, newPass); *// 新しいNameとPassをPage4に渡す*
88. frame.setVisible(true);
89. dispose();
90. }
91. });
92. p1.add(label1);
93. p1.add(nameField);
94. p1.add(label2);
95. p1.add(mail);
96. p1.add(label3);
97. p1.add(passField);
98. p1.add(label4);
99. p1.add(address);
100. p1.add(gender[0]);
101. p1.add(gender[1]);
102. p1.add(label5);
103. p1.add(birth);
104. p1.add(label5);
105. p1.add(job);
106. p2.add(back);
107. p3.add(check);
108. Container contentPane = getContentPane();
109. contentPane.add(p1, BorderLayout.CENTER);
110. contentPane.add(p2, BorderLayout.WEST);
111. contentPane.add(p3, BorderLayout.EAST);
112. }
113. private static void writeLine(FileWriter writer, List<String> list) throws IOException {
114. boolean first = true;
115. for (String item : list) {
116. if (!first) {
117. writer.append(',');
118. }
119. writer.append(item);
120. first = false;
121. }
122. writer.append('\n');
123. }
124. public static void main(String args[]) {
125. Page6 page6 = new Page6("defaultName", "defaultPass", "Page6");
126. page6.setVisible(true);
127. }
128. }