**Experiment Report - 35 - test10\_templateselector**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | line 12 | Syntax | √ | √ | √ |
| E02 | line 19 | Semantic | √ | × | × |
| E03 | line 22 | Syntax | √ | √ | √ |

Additional Errors Found by Self: 0

Self-Review Detection Rate: 100%

Peer 1 Detection Rate: 67%

Peer 2 Detection Rate: 67%

1. **Source Code**
2. #include "templateselector.h"
3. #include "templateeditor.h" // or the correct path to TemplateEditor's header file
4. TemplateSelector::~TemplateSelector() {}
5. void TemplateSelector::addTemplateBlock(const QString &title, const QString &templateText)
6. {
7. QFrame \*frame = new QFrame(this);
8. frame->setFrameShape(QFrame::StyledPanel);
9. QVBoxLayout \*frameLayout = new QVBoxLayout(frame);
10. QLabel \*titleLabel = new QLabel(title, this);
11. frameLayout->addWidget(titleLabel);
12. QLabel \*templateLabel = new QLabel("<pre>" + templateText + "</pre>", this);
13. templateLabel->setWordWrap(true);
14. frameLayout->addWidget(templateLabel);
15. QPushButton \*editButton = new QPushButton("Edit Templat", this);
16. frameLayout->addWidget(editButton);
17. connect(editButton, &QPushButton::clicked, this, []() {
18. // Close the TemplateSelector dialog before opening the TemplateEditor
19. this->close(); // Close TemplateSelector dialog immediately
20. TemplateEditor \*editor = new TemplateEditor(templateText, this);
21. if (editor->exec() == QDialog::Accepted) {
22. QString result = editor->getEditedTemplate();
23. emit templateSelected(result); // Signal to return the edited template
24. }
25. delete editor;
26. });
27. mainLayout->addWidget(frame);
28. }