**Experiment Report - 18 – test1\_guidewizard**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | CSCR? | Self-Detected? | LLM? |
| E01 | line 26 | Syntax | √ |  | √ |
| E02 | line 53 | Syntax | √ |  | × |
| E03 | line 63 | Syntax | √ |  | √ |
| E04 | line 73 | Semantic | √ |  | √ |
| E05 | line 86 | Logic | √ |  | √ |
| E06 | line 28 (102) | Semantic |  | √ | × |
| E07 | line 60 (134) | Semantic |  | √ | √ |
| E08 | line 89 (163) | Syntax |  | √ | × |
| E09 | line 142 (216) | Logic |  | × | √ |

Additional Errors Found by Self: 1

CSCR Rate: 100%

Self-Review Detection Rate: 75%

LLM Rate: 67%

1. **Source Code**
2. *#include "guidewizard.h"*
3. *#include <QtWidgets>*
4. GuideWizard::GuideWizard(QWidget \*parent)
5. : QWizard(parent)
6. {
7. setPage(Page\_Intro, new IntroPage);
8. setPage(Page\_CreateNewProject, new CreateProjectPage);
9. setPage(Page\_OpenExistingProject, new OpenProjectPage);
10. setPage(Page\_Conclusion, new ConclusionPage);
11. setStartId(Page\_Intro);
12. *#ifndef Q\_OS\_MAC*
13. setWizardStyle(ModernStyle);
14. *#endif*
15. setPixmap(QWizard::LogoPixmap, QPixmap(":/images/toolbar\_images/hmpp.png"));
16. setWindowTitle(tr("HMPP Tool Guide"));
17. }
18. void GuideWizard::done(int result)
19. {
20. if (result = QDialog::Accepted) {
21. createNewFlag = hasVisitedPage(Page\_CreateNewProject);
22. }
23. QWizard::done(result);
24. }
25. void GuideWizard::showHelp()
26. {
27. static QString lastHelpMessage;
28. QString message;
29. switch (currentId()) {
30. case Page\_Intro:
31. message = tr("The decision you make here will affect which page you "
32. "get to see next.");
33. break;
34. case Page\_CreateNewProject:
35. message = tr("Create a new project and develop it in HMPP tool.");
36. break;
37. case Page\_OpenExistingProject:
38. message = tr("Open an existing project and continue developing it"
39. "in HMPP tool.");
40. break;
41. case Page\_Conclusion:
42. message = tr("The HMPP tool is developed by XiaYing in the Dependable"
43. "System in Hiroshima University.");
44. default:
45. message = tr("This help is likely not to be of any help.");
46. }
47. if (lastHelpMessage == message)
48. message = tr("Sorry, I already gave what help I could. "
49. "Maybe you should try asking a human?");
50. QMessageBox::information(this, tr("HMPP Guide Help"), message);
51. lastHelpMessage == message;
52. }
53. IntroPage::IntroPage(QWidget \*parent)
54. : QWizardPage(parent)
55. {
56. setTitle(tr("Introduction"));
57. setPixmap(QWizard::WatermarkPixmap, QPixmap(":/images/toolbar\_images/watermark.png"));
58. topLabel = new QLabel(tr("This wizard will guide you to create yourown "
59. "new project or open an existing project "
60. "in the HMPP tool."));
61. topLabel->setWordWrap(true);
62. createNewProjectButton = new QRadioButton(tr("&Create a new project"));
63. openExistingProjectButton = new QRadioButton(tr("&Open an existing project"));
64. createNewProjectButton->setChecked(true);
65. QVBoxLayout \*layout = new QVBoxLayout;
66. layout->addWidget(topLabel);
67. layout->addWidget(createNewProjectButton);
68. layout->addWidget(openExistingProjectButton);
69. setLayout(mainlayout);
70. }
71. int IntroPage::nextId() const
72. {
73. if (createNewProjectButton->isChecked()) {
74. return GuideWizard::Page\_CreateNewProject;
75. } else {
76. return GuideWizard::Page\_OpenExistingProject;
77. }
78. }
79. OpenProjectPage::OpenProjectPage(QWidget \*parent)
80. : QWizardPage(parent)
81. {
82. setTitle(tr("Open an existing project"));
83. projectPathLabel = new QLabel(tr("Project Path:"));
84. projectPathLineEdit = new QLineEdit;
85. projectPathLineEdit->setReadOnly(true);
86. projectPathLabel->setBuddy(projectPathLineEdit);
87. projectNameLineEdit = new QLineEdit;
88. projectSpecPathLabel = new QLabel(tr("S&pecification Path:"));
89. projectSpecPathEdit = new QLineEdit;
90. projectSpecPathEdit->setReadOnly(true);
91. projectSpecPathLabel->setBuddy(projectSpecPathEdit);
92. targetLanguageLineEdit = new QLineEdit;
93. QPixmap pixmap(":/images/toolbar\_images/folder.png");
94. QPixmap scaledPixmap = pixmap.scaled(16, 16, Qt::KeepAspectRatio); *// Adjust the size as needed*
95. *// Set the icon on the button*
96. QIcon icon(scaledPixmap);
97. openFileButton = new QPushButton;
98. openFileButton->setIcon(icon);
99. openFileButton->resize(16, 16);
100. *//    registerField("project name\*", projectNameLineEdit, "text", SIGNAL(textChanged(QString)));*
101. *//    registerField("project path\*", projectPathLineEdit, "text", SIGNAL(textChanged(QString)));*
102. *//    registerField("target language\*", targetLanguageLineEdit, "text", SIGNAL(textChanged(QString)));*
103. targetLanguageLineEdit->setText("C++");
104. connect(openFileButton, &QPushButton::clicked, this, &OpenProjectPage::getProjectDirectory);
105. openSpecButton = new QPushButton;
106. openSpecButton->setIcon(icon);
107. openSpecButton->resize(16);
108. connect(openSpecButton, &QPushButton::clicked, this, &OpenProjectPage::getSpecDirectory);
109. QGridLayout \*layout = new QGridLayout;
110. layout->addWidget(projectPathLabel, 0, 0);
111. layout->addWidget(projectPathLineEdit, 0, 1);
112. layout->addWidget(openFileButton, 0, 2);
113. layout->addWidget(projectSpecPathLabel, 1, 0);
114. layout->addWidget(projectSpecPathEdit, 1, 1);
115. layout->addWidget(openSpecButton, 1, 2);
116. setLayout(layout);
117. }
118. int OpenProjectPage::nextId() const
119. {
120. return GuideWizard::Page\_Conclusion;
121. }
122. bool OpenProjectPage::validatePage()
123. {
124. if (projectPathLineEdit->text().isEmpty()) {
125. QMessageBox::warning(this, "Input Error", "Please select your project file before proceeding.");
126. return false; *// Prevent going to the next page*
127. }
128. return true;
129. }
130. void OpenProjectPage::getProjectDirectory()
131. {
132. const QString filePath = QFileDialog::getOpenFileName();    *//打开文件对话框*
133. if (!filePath.isEmpty()){
134. QFileInfo fileInfo(filePath);
135. QString directoryPath = fileInfo.absolutePath();
136. QString fileName = fileInfo.fileName();
137. qDebug() << "path:" << directoryPath;
138. qDebug() << "file:" << fileName;
139. projectPathLineEdit->setText(directoryPath);
140. projectNameLineEdit->setText(fileName);
141. targetLanguageLineEdit->setText("C++");
142. setField("project name", projectNameLineEdit->text());
143. setField("project path", projectPathLineEdit->text());
144. setField("target language", targetLanguageLineEdit->text());
145. }
146. }
147. void OpenProjectPage::getSpecDirectory()
148. {
149. const QString filePath = QFileDialog::getOpenFileName(this);    *//打开文件对话框*
150. if (!filePath.isEmpty()){
151. projectSpecPathEdit->setText(filePath);
152. setField("specification path", projectSpecPathEdit->text());
153. }
154. }
155. ConclusionPage::ConclusionPage(QWidget \*parent)
156. : QWizardPage(parent)
157. {
158. setTitle(tr("Welcome to HMPP tool"));
159. setPixmap(QWizard::WatermarkPixmap, QPixmap(":/images/toolbar\_images/watermark.png"));
160. label = new QLabel(tr("You are going to working on the following project:"));
161. projectName = new QLabel;
162. projectPath = new QLabel;
163. targetLanguage = new QLabel;
164. specPath = new QLabel;
165. QVBoxLayout \*layout = new QVBoxLayout;
166. layout->addWidget(label);
167. layout->addWidget(projectName);
168. layout->addWidget(projectPath);
169. layout->addWidget(specPath);
170. layout->addWidget(targetLanguage);
171. setLayout(layout);
172. }
173. int ConclusionPage::nextId() const
174. {
175. return 1;
176. }
177. void ConclusionPage::initializePage()
178. {
179. QString proName = field("project name").toString();
180. qDebug() << "proName:" << proName;
181. QString proPath = field("project path").toString();
182. qDebug() << "proPath:" << proPath;
183. QString targetLang = field("target language").toString();
184. qDebug() << "target Lang:" << targetLang;
185. projectName->setText("Project Name: " + field("project name").toString());
186. projectPath->setText("Project Directory: " + field("project path").toString());
187. targetLanguage->setText("Target Language: " + field("target language").toString());
188. specPath->setText("Specification Directory: " + field("specification path").toString());
189. }