Manual testing

Project screen

Test case 1. Project screen views.

Objective: Test project screen views changes.

Procedure:

1. Navigate to the project screen.
2. Press “Grid view” button.
3. Press “List view” button.
4. Press “Table view” button.

Checklist:

1. In the top of the screen “Projects” title is visible.
2. Buttons “Grid view”, “List view” and “Table view” are visible.
3. After selecting grid view projects are displayed in grid as squares with title and action buttons: “Open”, “Generate”, “Delete” and “Edit”
4. After selecting “list view” projects are visible inline with same actions button.
5. After selecting table view there is no actions button, but there is detailed information.
6. After selecting any of this button it becomes disabled.

Test case 2. Project Addition.

Objective: Confirm project addition works properly without issues.

Procedure:

1. Navigate to the projects screen.
2. Press “Add” button marked with plus sign icon.
3. In the add dialog window press “Cancel” button.
4. In the dialog window input “Project name” and “Project folder” information.
5. Press “Add button”

Checklist:

1. Add button is visible and clickable.
2. Add dialog contains “Project name” and “Project folder” inputs.
3. “Project name” is free-hand input.
4. “Project folder” is opening path dialog.

5. Canceling not adding new project.
6. Add button add new project and refreshes the view.

Test case 3. Project edition.

Objective: Testify project edition dialog and confirm that it worked without any issues.

Procedure:

1. Navigate to the projects screen.
2. Select any project and press “Edit” button on that project.
3. Press “Cancel” button in the dialog.
4. In the edit dialog there is “Project name” input.
5. Input any information.
6. Press “Save”.

Checklist:

1. Edit button is displayed for every project in the list.
2. Edit dialog appears after clicking on Edit button.
3. After canceling the operation, nothing is updated.
4. After saving changes, they are visible in the view.
5. “Project name” is free-hand input

Test case 4. Project parsing.

Objective: Confirm there is ability to parse the project and there are no issues during parsing.

Procedure:

1. Navigate to the projects screen.
2. Press “Parse” button near the “add” button in the bottom of the screen.
3. Select any project folder to parse.
4. Wait for the result.

Checklist:

1. “Parse” button is visible and clickable.
2. After pressing this button, path dialog is shown up.
3. Selected folder will be set for the new project record.
4. Name of the directory will be set as project name.
5. If anything went wrong, error message appears and new log record will be added under errors screen.

Test case 5. Project generation.

Objective: Confirm project generation is works without issues.

Procedure:

1. Navigate to the projects screen.
2. Select any project.
3. Press “Generate” button on this project from the view screen or from the details screen from the header section.
4. Wait until generation proceed.

Checklist:

1. Generation button is visible on the project record component.
2. Generation button is visible on the header section of the project details page.
3. Generation can be started.
4. After generation files are updated.
5. Generation information is added to Actions screen.

Test case 6. Project deletion.

Objective: Confirm deletion flow is working as expected without any errors.

Procedure:

1. Navigate to the projects screen.
2. Select any project.
3. Press delete button.
4. In the delete dialog press “cancel”.
5. In the delete dialog press “Delete”.

Checklist:

1. Delete button is visible in the projects view on each project record.
2. Delete button is visible on the project detail page.
3. Delete dialog appears after clicking on delete button.
4. Delete dialog contains two buttons “Cancel” and “Delete”.
5. After canceling delete dialog closes.
6. After deleting dialog closes and project removed from the list.

Test case 7. Project opening.

Objective: Confirm that project details page is visible and accessible.

Procedure:

1. Navigate to the projects screen.
2. Select any project and navigate to its detail page using “->” button.

Checklist:

1. Navigation button is on each project record in the view.
2. Project’s detail page is rendered without any issues.
3. Add, Edit, delete, generate buttons are visible on the header section.
4. Project title is visible on the header section and equals to the project’s title you selected.

Test case 8. Project details.

Objective: Confirm that project details is loaded properly from the database and displayed.

Procedure:

1. Navigate to the project detail screen.
2. Press on the Projects title.

Checklist:

1. Project title is clickable.
2. After clicking on the project title sidebar is displayed.
3. Sidebar appears with animation.
4. On the sidebar “date created”, “date edited”, “files”, “folders” information displayed and is read-only.
5. Sidebar has clickable close icon.
6. After clicking on this icon sidebar removed from the screen.
7. Sidebar removes with animation.

Test case 9. Project conversion to another version.

Objective: Confirm project can be converted to another programming language version.

Procedure:

1. Navigate to the project detail page.
2. Select “Convert to version” (next convert) button.
3. In the conversion dialog select any of suggested versions.
4. Discard converting, via pressing “Cancel” button.
5. Start converting using “Convert” button.

Checklist:

1. “Convert” button is visible and appears in the footer of the project detail screen.
2. Convert dialog contains information of the current project version.
3. Convert dialog contains dropdown menu with all possible versions.
4. Convert dialog has “Convert” and “Cancel” buttons.
5. “Cancel” button removes this dialog without any changes.
6. “Convert” button statrs conversion.
7. After conversion starts, message appears on the screen.
8. In case conversion failed, logs will be added under the Errors screen.
9. In case conversion succeed, action will be added under the Actions screen and can be undone.

Test case 10. Project conversion to another programming language.

Objective: Confirm project can be converted to another programming language.

Procedure:

1. Navigate to the project detail screen.
2. Press “Translate” button.
3. In the translation dialog select programming language from the dropdown menu.
4. Press “Translate” button to start conversion.
5. Press “Cancel” button to discard conversion process.

Checklist:

1. Convert to another programming language is clickable.
2. Conversion dialog contains title “Translate to”.
3. Conversion dialog has dropdown menu with all possible programming languages to convert.
4. “Cancel” button discards conversion to another programming language.
5. “Translate” button starts conversion to the programming language you selected.
6. After translation starts, message appears with text “Converting project ${projectName} to ${language} started”.

Project’s files screen

Test case 11. Project’s files screen views.

Objective: Confirm ability to change project files screen view.

Procedure:

1. Navigate to the projects screen.
2. Select any project and open it.
3. Press “Grid view” button.
4. Press “List view” button.
5. Press “Table view” button.

Checklist:

1. In the top of the screen project’s name is set as title and is visible.
2. Buttons “Grid view”, “List view” and “Table view” are visible.
3. After selecting grid view files are displayed in grid as squares with title and action buttons: “Open”, “Generate”, “Delete” and “Edit”
4. After selecting “list view” files are visible inline with same actions button.
5. After selecting table view there is no actions button, but there is detailed information about the files.
6. After selecting any of this button it becomes disabled.

Test case 12. Project file adition.

Objective: Confirm project file addition works properly without issues.

Procedure:

1. Navigate to the project files screen.
2. Press “Add” button marked with plus sign icon.
3. In the add dialog window press “Cancel” button.
4. In the dialog window input “File name”.
5. Press “Add button”

Checklist:

1. Add button is visible and clickable.
2. Add dialog contains file name free-hand input.
3. Canceling not adding new file under the project.
4. Add button adds new file under the project refreshes the view.

Test case 13. Project file edition.

Objective: Testify project file editiong dialog and confirm that it worked without any issues.

Procedure:

1. Navigate to the project’s files screen.
2. Select any file and press “Edit” button.
3. Press “Cancel” button in the dialog.
4. Change file name in the dialog.
5. Press “Save”.

Checklist:

1. Edit button is displayed for every project in the list.
2. Edit dialog appears after clicking on Edit button.
3. After canceling the operation, nothing is updated.
4. After saving changes, they are visible in the view.
5. “File name” is free-hand input.

Test case 14. Project file parsing.

Objective: Confirm there is ability to parse only specific files under the project and there are no issues during parsing.

Procedure:

1. Navigate to the project’s files screen.
2. Press “Parse” button near the “add” button in the bottom of the screen.
3. Select any file from your PC to parse.
4. Wait for the result.

Checklist:

1. “Parse” button is visible and clickable.
2. After pressing this button, path dialog is shown up.
3. Selected file will be added under the selected project record.
4. Name of the file will be set same.
5. If anything went wrong, error message appears and new log record will be added under errors screen.

Test case 15. Project file generation.

Objective: Confirm there is ability to generate some files from the project.

Procedure:

1. Navigate to the projects screen.
2. Select any project file.
3. Press “Generate” button on this file from the view screen or from the file’s details screen in the header section.
4. Wait until generation proceed.

Checklist:

1. Generation button is visible on the project record component.
2. Generation button is visible on the header section of the project details page.
3. Generation can be started.
4. After generation files are updated.
5. Generation information is added to Actions screen.

Test case 16. Project file deletion.

Objective: Confirm deletion flow is working as expected without any errors.

Procedure:

1. Navigate to the project’s files screen.
2. Select any files under the project.
3. Press delete button.
4. In the delete dialog press “cancel”.
5. In the delete dialog press “Delete”.

Checklist:

1. Delete button is visible in the project’s files view on each file record.
2. Delete button is visible on the project’s file detail page.
3. Delete dialog appears after clicking on delete button.
4. Delete dialog contains two buttons “Cancel” and “Delete”.
5. After canceling delete dialog closes.
6. After deleting dialog closes and files are removed from the list.

Test case 17. Project’s file details page opening.

Objective: Confirm that project details page is visible and accessible.

Procedure:

1. Navigate to the project’s files screen.
2. Select any file and navigate to its detail page using “->” button.

Checklist:

1. Navigation button is on each file record in the view.
2. Project’s file’s detail page is rendered without any issues.
3. Add, Edit, delete, generate buttons are visible on the header section.
4. Title is visible on the header section and equals to the file’s name you selected.

Test case 18. Project’s file’s details.

Objective: Confirm that project details is loaded properly from the database and displayed.

Procedure:

1. Navigate to the project’s file detail screen.
2. Press on the file detail title (File name).

Checklist:

1. File’s detail title is clickable.
2. After clicking on the file name, sidebar is displayed.
3. Sidebar appears with animation.
4. On the sidebar “date created”, “date edited”, “files”, “folders” information displayed and is read-only.
5. Sidebar has clickable close icon.
6. After clicking on this icon sidebar removed from the screen.
7. Sidebar removes with animation.

Test case 19. Project’s file conversion to another version.

Objective: Confirm project’s file can be converted to another programming language version.

Procedure:

1. Navigate to the project’s file’s detail screen.
6. Select “Convert to version” (next convert) button.
7. In the conversion dialog select any of suggested versions.
8. Discard converting, via pressing “Cancel” button.
9. Start converting using “Convert” button.

Checklist:

1. “Convert” button is visible and appears in the footer of the project detail screen.
2. Convert dialog contains information of the current project’s file version.
3. Convert dialog contains dropdown menu with all possible versions.
4. Convert dialog has “Convert” and “Cancel” buttons.

5. “Cancel” button removes this dialog without any changes.
6. “Convert” button starts conversion.
7. After conversion starts, message appears on the screen.
8. In case conversion failed, logs will be added under the Errors screen.
9. In case conversion succeed, action will be added under the Actions screen and can be undone.

Test case 20. Project’s file conversion to another programming language.

Objective: Confirm project’s file can be converted to another programming language.

Procedure:

1. Navigate to the project’s file detail screen.
2. Press “Translate” button.
3. In the translation dialog select programming language from the dropdown menu.
4. Press “Translate” button to start conversion.
5. Press “Cancel” button to discard conversion process.

Checklist:

1. Convert to another programming language is clickable.
2. Conversion dialog contains title “Translate to”.
3. Conversion dialog has dropdown menu with all possible programming languages to convert.
4. “Cancel” button discards conversion to another programming language.
5. “Translate” button starts conversion to the programming language you selected.
6. After translation starts, message appears with text “Converting file ${fileName} under the project ${projectName} to ${language} started”.

Documentation’s screen

Test case 21. Documentation’s screen views.

Objective: Test documentation’s screen views changes.

Procedure:

1. Navigate to the documentation’s screen.
2. Press “Grid view” button.
3. Press “List view” button.
4. Press “Table view” button.

Checklist:

1. In the top of the screen “Documentations” title is visible.
2. Buttons “Grid view”, “List view” and “Table view” are visible.
3. After selecting grid view documentations are displayed in grid as squares with title and action buttons: “Open”, “Generate”, “Delete” and “Edit”
4. After selecting “list view” documentations are visible inline with same actions button.
5. After selecting table view there is no actions button, but there is detailed information.
6. After selecting any of this button it becomes disabled.

Test case 22. Documentation Addition.

Objective: Confirm documentation addition works properly without issues.

Procedure:

1. Navigate to the Documentation’s screen.
2. Press “Add” button marked with plus sign icon.
3. In the add dialog window press “Cancel” button.
4. In the dialog window input “Project name” and “Project folder” information.
5. Press “Add button”

Checklist:

1. Add button is visible and clickable.
2. Add dialog contains “Documentation name” and “Code structure” inputs.
3. “Documentation name” is free-hand input.

4. “Code structure” is dropdown with all available code structure.
5. Canceling not adding new documentation.
6. Add button add new documentation and refreshes the view.

Test case 23. Documentation edition.

Objective: Testify project edition dialog and confirm that it worked without any issues.

Procedure:

1. Navigate to the documentation screen.
2. Select any documentation and press “Edit” button on that project.
3. Press “Cancel” button in the dialog.
4. In the edit dialog there is “Documentation name” input.
5. In the edit dialog there is “Code structure” dropdown
6. Input any information.
7. Press “Save”.

Checklist:

1. Edit button is displayed for every documentation in the list.
2. Edit dialog appears after clicking on Edit button.
3. After canceling the operation, nothing is updated.
4. After saving changes, they are visible in the view.
5. “Documentation name” is free-hand input
6. “Code structure” is a dropdown with available options.

Test case 24. Documentation parsing.

Objective: Confirm there is ability to parse the documentation and there are no issues during parsing.

Procedure:

1. Navigate to the documentation screen.
2. Press “Parse” button near the “add” button in the bottom of the screen.
3. Select any method from the dropdown menu to parse.
4. Wait for the result.

Checklist:

1. “Parse” button is visible and clickable.
2. After pressing this button, path dialog is shown up.
3. Selected metho documentation will be set for the new project record.
4. If documentation is not found – operation is aborted.
5. Name of the method will be set as documentation name.
6. If anything went wrong, error message appears and new log record will be added under errors screen.

Test case 25. Documentation generation.

Objective: Confirm project generation is works without issues.

Procedure:

1. Navigate to the documentation screen.
2. Select any documentation.
3. Press “Generate” button on this documentation from the view screen or from the details screen from the header section.
4. Wait until generation proceed.

Checklist:

1. Generation button is visible on the documentation record component.
2. Generation button is visible on the header section of the documentation details page.
3. Generation can be started.
4. After generation files that contains current documentation will be updated.
5. Generation information is added to Actions screen.

Test case 26. Documentation deletion.

Objective: Confirm deletion flow is working as expected without any errors.

Procedure:

1. Navigate to the documentation screen.
2. Select any documentation.
3. Press delete button.
4. In the delete dialog press “cancel”.
5. In the delete dialog press “Delete”.

Checklist:

1. Delete button is visible in the documentation view on each project record.
2. Delete button is visible on the documentation detail page.
3. Delete dialog appears after clicking on delete button.
4. Delete dialog contains two buttons “Cancel” and “Delete”.
5. After canceling delete dialog closes.
6. After deleting dialog closes and documentation removed from the list.

Test case 27. Documentation opening.

Objective: Confirm that documentation details page is visible and accessible.

Procedure:

1. Navigate to the documentation screen.
2. Select any documentation and navigate to its detail page using “->” button.

Checklist:

1. Navigation button is on each documentation record in the view.
2. Documentation’s detail page is rendered without any issues.
3. Add, Edit, delete, generate buttons are visible on the header section.
4. Documentation title is visible on the header section and equals to the Documentation’s title you selected.

Test case 28. Documentation details.

Objective: Confirm that documentation details is loaded properly from the database and displayed.

Procedure:

1. Navigate to the documentation detail screen.
2. Press on the documentation’s title.

Checklist:

1. Documentation’s title is clickable.
2. After clicking on the documentation title sidebar is displayed.
3. Sidebar appears with animation.
4. On the sidebar “date created”, “date edited”, “files”, “folders” information displayed and is read-only.
5. Sidebar has clickable close icon.
6. After clicking on this icon sidebar removed from the screen.
7. Sidebar removes with animation.

Test case 29. Documentation conversion to another version.

Objective: Confirm documentation can be converted to another programming language version.

Procedure:

1. Navigate to the documentation detail page.
2. Select “Convert to version” (next convert) button.
3. In the conversion dialog select any of suggested versions.
4. Discard converting, via pressing “Cancel” button.
5. Start converting using “Convert” button.

Checklist:

1. “Convert” button is visible and appears in the footer of the documentation detail screen.
2. Convert dialog contains information of the current project version.
3. Convert dialog contains dropdown menu with all possible versions.
4. Convert dialog has “Convert” and “Cancel” buttons.

5. “Cancel” button removes this dialog without any changes.
6. “Convert” button stats conversion.
7. After conversion starts, message appears on the screen.
8. In case conversion failed, logs will be added under the Errors screen.
9. In case conversion succeed, action will be added under the Actions screen and can be undone.

Test case 30. Documentation conversion to another programming language.

Objective: Confirm documentation can be converted to another programming language.

Procedure:

1. Navigate to the documentation detail screen.
2. Press “Translate” button.
3. In the translation dialog select programming language from the dropdown menu.
4. Press “Translate” button to start conversion.
5. Press “Cancel” button to discard conversion process.

Checklist:

1. Convert to another programming language is clickable.
2. Conversion dialog contains title “Translate to”.
3. Conversion dialog has dropdown menu with all possible programming languages to convert.
4. “Cancel” button discards conversion to another programming language.
5. “Translate” button starts conversion to the programming language you selected.
6. After translation starts, message appears with text “Converting documentation ${documentationTitle} to ${language} started”.

Tests screen

Test case 31. Test’s screen views.

Objective: Conform test’s screen views changes.

Procedure:

1. Navigate to the tests screen.
2. Press “Grid view” button.
3. Press “List view” button.
4. Press “Table view” button.

Checklist:

1. In the top of the screen “Tests” title is visible.
2. Buttons “Grid view”, “List view” and “Table view” are visible.
3. After selecting grid view tests are displayed in grid as squares with title and action buttons: “Open”, “Generate”, “Delete” and “Edit”
4. After selecting “list view” tests are visible inline with same actions button.
5. After selecting table view there is no actions button, but there is detailed information.
6. After selecting any of this button it becomes disabled.

Test case 32. Test Addition.

Objective: Confirm test can be added properly without issues.

Procedure:

1. Navigate to the tests screen.
2. Press “Add” button marked with plus sign icon.
3. In the add dialog window press “Cancel” button.
4. In the dialog window input “Test name” and “Target method” information.
5. Press “Add button”

Checklist:

1. Add button is visible and clickable.
2. Add dialog contains “Test name” and “Target method” inputs.
3. “Test name” is free-hand input.
4. “Target method” is dropdown with all possible methods.

5. Canceling not adding new test.
6. Add button add new test and refreshes the view.

Test case 33. Test’s edition.

Objective: Confirm test edit dialog works without issues.

Procedure:

1. Navigate to the tests screen.
2. Select any test and press “Edit” button on that test.
3. Press “Cancel” button in the dialog.
4. Input another information.
5. Press “Save”.

Checklist:

1. Edit button is displayed for every test in the list.
2. Edit dialog appears after clicking on Edit button.
3. After canceling the operation, nothing is updated.
4. After saving changes, they are visible in the view.
5. “Test name” is free-hand input
6. “Target Method” is dropdown menu with all available methods.

Test case 34. Test parsing.

Objective: Confirm there is ability to parse the test and there are no issues during parsing.

Procedure:

1. Navigate to the test screen.
2. Press “Parse” button near the “add” button in the bottom of the screen.
3. Select any method from the file to parse.
4. Wait for the result.

Checklist:

1. “Parse” button is visible and clickable.
2. After pressing this button, path dialog is shown up.
3. Selected file will be parsed into methods with tests and set for the new test record.
4. Name of the method will be set as test name.
5. If anything went wrong, error message appears and new log record will be added under errors screen.

Test case 35. Test generation.

Objective: Confirm test generation is works without issues.

Procedure:

1. Navigate to the tests screen.
2. Select any test.
3. Press “Generate” button on this test from the view screen or from the details screen from the header section.
4. Wait until generation proceed.

Checklist:

1. Generation button is visible on the test record component.
2. Generation button is visible on the header section of the test details page.
3. Generation can be started.
4. After generation files are updated.
5. Generation information is added to Actions screen.

Test case 36. Test deletion.

Objective: Confirm test’s deletion flow is working as expected without any errors.

Procedure:

1. Navigate to the test screen.
2. Select any test.
3. Press delete button.
4. In the delete dialog press “cancel”.
5. In the delete dialog press “Delete”.

Checklist:

1. Delete button is visible in the test view on each project record.
2. Delete button is visible on the test detail page.
3. Delete dialog appears after clicking on delete button.
4. Delete dialog contains two buttons “Cancel” and “Delete”.
5. After canceling delete dialog closes.
6. After deleting dialog closes and test removed from the list.

Test case 37. Test opening.

Objective: Confirm that test details page is visible and accessible.

Procedure:

1. Navigate to the test screen.
2. Select any project and navigate to its detail page using “->” button.

Checklist:

1. Navigation button is on each test record in the view.
2. Test’s detail page is rendered without any issues.
3. Add, Edit, delete, generate buttons are visible on the header section.
4. Test title is visible on the header section and equals to the Test’s name you selected.

Test case 38. Test details.

Objective: Confirm that test details is loaded properly from the database and displayed.

Procedure:

1. Navigate to the test detail screen.
2. Press on the test’s title.

Checklist:

1. Test’s title is clickable.
2. After clicking on the test’s title sidebar is displayed.
3. Sidebar appears with animation.
4. On the sidebar “date created”, “date edited”, “files”, “folders” information displayed and is read-only.
5. Sidebar has clickable close icon.
6. After clicking on this icon sidebar removed from the screen.
7. Sidebar removes with animation.

Test case 39. Test conversion to another version.

Objective: Confirm test can be converted to another programming language version.

Procedure:

1. Navigate to the test detail page.
2. Select “Convert to version” (next convert) button.
3. In the conversion dialog select any of suggested versions.
4. Discard converting, via pressing “Cancel” button.
5. Start converting using “Convert” button.

Checklist:

1. “Convert” button is visible and appears in the footer of the test detail screen.
2. Convert dialog contains information of the current test version.
3. Convert dialog contains dropdown menu with all possible versions.
4. Convert dialog has “Convert” and “Cancel” buttons.
5. “Cancel” button removes this dialog without any changes.
6. “Convert” button starts conversion.
7. After conversion starts, message appears on the screen.
8. In case conversion failed, logs will be added under the Errors screen.
9. In case conversion succeed, action will be added under the Actions screen and can be undone.

Test case 40. Test conversion to another programming language.

Objective: Confirm test can be converted to another programming language.

Procedure:

1. Navigate to the test detail screen.
2. Press “Translate” button.
3. In the translation dialog select programming language from the dropdown menu.
4. Press “Translate” button to start conversion.
5. Press “Cancel” button to discard conversion process.

Checklist:

1. Convert to another programming language is clickable.
2. Conversion dialog contains title “Translate to”.
3. Conversion dialog has dropdown menu with all possible programming languages to convert.
4. “Cancel” button discards conversion to another programming language.
5. “Translate” button starts conversion to the programming language you selected.
6. After translation starts, message appears with text “Converting test ${testName} to ${language} started”.

Service testing

Test case 41. Error handling.

Objective: All errors are handled by the application.

Procedure:

1. Navigate to the errors screen.

Checklist:

1. Errors screen contains general information about the error.
2. Error detail screen contains detailed information about the error.

Test case 42. Language translation.

Objective: Verify is all application labels are translatable.

Procedure:

1. For every screen find label translation.

Checklist:

1. Every key label element is translatable.

Test case 43. Suggest translation.

Objective: User can suggest translation.

Procedure:

1. Navigate to the translation tab.
2. Edit translation keywords.
3. Press “Suggest” button.

Checklist:

1. Translation screen is displayed.
2. There are loaded all keywords.
3. Suggest button is visible.
4. After pressing suggest button email is sent.
5. Message “Thanks for the contribution” is displayed.

Test case 44. Application appearance changing.

Objective: Confirm that application can be modified in runtime.

Procedure:

1. Navigate to the settings screen.
2. Find Appearance section.
3. Edit any color setting.
4. Apply changes.

Checklist:

1. Settings screen is visible and accessible.
2. Appearance section is on the settings screen.
3. Values under this section are modifiable.
4. After changing settings, appearance is changed.
5. After saving changes appearance confirmed and loaded on application startup.
6. After discarding changes, appearance left as it was before editing.
7. If some settings need to restart application, confirmation dialog appears.

Test case 45. Generation process cancelation.

Objective: Generation process can be canceled.

Procedure:

1. For every generation process: project, file, documentation, test start generation.
2. On the progress bar of the generation press “X” button.

Checklist:

1. Generation cancelation button is visible.
2. After cancelation progress section contain message “Process was aborted. Files Restored”.
3. All generated files – deleted.
4. All files left to generation – canceled.
5. If files regenerated – modification not applied.

Test case 46. Generation process progress bar functionality.

Objective: Confirm generation process is covered with progress bar.

Procedure:

1. For every generation process: project, file, documentation, test start generation.

Checklist:

1. Generation progress section is visible.
2. Cancel button is displayed on the top of the progress section.
3. On the progress bar displayed last generated item name, items left.
4. After generation completed, progress is hidden.

Test case 47. Discarding actions functionality.

Objective: Confirm that all actions, happened within application can be undone.

Procedure:

1. After application testing navigate to the actions screen.
2. Select any action and press “Undo”.
3. Navigate to the “Undone” section and press “Redo”.

Checklist:

1. Action screen is reachable.
2. Action screen has two tabs: “Performed” and “Undone”.
3. Preformed section contains every action happened within application.
4. Undone action contains performed actions that was undo.
5. All performed action contains “Undo” button.
6. All Undone actions contains “Redo” button.
7. After clicking on “Undo” button, action moved to undone and user redirected to the affected region.
8. After clicking on “Redo” button, action moved to performed section and user redirected to the affected region.

Test case 48. Application startup failed logs.

Objective: Confirm application generate logs file if startup is failed.

Procedure:

1. Test handled during local testing.

Checklist:

1. If something cannot be loaded, application generated logs file

Test case 49. Open in file explorer functionality.

Objective: Verify projects or file can be reviled in the file explorer.

Procedure:

1. Navigate to the projects or files screen.
2. Press “Open in file explorer”

Checklist:

1. “Open in the file explorer” button is visible on the project and file record.
2. After clicking file explorer opened with selected file or project folder.

Test case 50. Restore file history screen.

Objective: Confirm that file history can be tracked.

Procedure:

1. Navigate to the projects or files screen.
2. Press “Open history”

Checklist:

1. “Open history” button is visible on the project and file record.
2. After clicking, user is redirected to the history screen of file or project.
3. Every history contains file saved information, file generated or regenerated.
4. File history record contains “Restore” button.

Test case 51. Restore file history functionality.

Objective: Confirm the file is restored for the selected moment of time.

Procedure:

1. Navigate to the file history scree (from test case 50).
2. Press “Restore” button

Checklist:

1. User redirected to the file/project restored.
2. File’s content is changed to the selected moment of time.