PPALMS Execution Scenarios

Set-up: Unzip the tar.gz containing the PPALMS files. A file named *provided.txt* is provided, which contains lines of basic C++ programming code. The file contains line numbers and a basic main function that adds two integers. This is a simple example file, but the system can function with any provided file that can be read as text. There are then two ways to run the system: one method is through a single line on the terminal, which takes arguments and generates questions if able (End-to-end Scenario A); and another in which PPALMS is run continuously and may generate multiple questions given some input (End-to-end Scenario B).

To run unit tests, the CMake and GoogleTest libraries must be installed. Unit tests can be built by running the following lines in the terminal:

```
make UnitTests
./UnitTests     OR    ./UnitTests <class_to_test>
```

The first ./UnitTests will run all available tests, while the second will run the set of tests for the given class (settings, linegrabber, questionbank, generator).

End-to-end Scenario A:

Step 1: Open terminal window and navigate to the directory containing the PPALMS files. If no PPALMS executable exists, run the following lines in the terminal to build the project:

```
make PPALMS
```

Step 2: Run the executable by typing the following command into terminal:

```
./PPALMS ./provided.txt canvas rearrange 1 6-7 8 10
```

Result: A file named "qti.txt" will be created containing the generated "rearrange" question, which is based on the given lines (6-7, 8, and 10). The lines will be randomly mixed.

Verification: The file "qti.txt" contains a question consisting of the lines of code, randomly rearranged. The correct answer is listed after the *a*. The system currently does not record all possible correct answers and only records the original order of lines.

Traceability: Requirements 1, 3-7, 13-14

End-to-end Scenario B:

Step 1: Open terminal window and navigate to the directory containing the PPALMS files. If no PPALMS executable exists, run the following lines in the terminal to build the project:

```
make PPALMS
```

Step 2: Run the executable by typing the following command into terminal:

```
./PPALMS ./provided.txt canvas rearrange 1 6-7 8 10
```

Step 3: Load the source code file by typing the following command into terminal:

```
path ./provided.txt
```

Step 4: Select a target LMS by typing the following command into terminal:

```
lms canvas
```

* Note: Currently has no effect on output.

Step 5 : Set as reordering by typing the following command into terminal:

```
type rearrange
```

Step 6: Select mutation quantity by typing the following command into terminal:

```
mutations 1
```

* Note: Currently has no effect on output.

Step 7: Select desired lines for question generation by typing the following the

```
lines 6-7 8 10 end
```

Step 8: Generate reordering questions by typing the following command into terminal:

```
generate
```

Step 9: To obtain questions for export to selected LMS target type the following command into terminal:

```
export
```

Step 10: After exporting questions close the program by typing the following command into terminal:

```
exit
```

Result: A file named “qti.txt” will be created containing the generated “rearrange” question, which is based on the given lines (6-7, 8, and 10). The lines will be randomly mixed.

Verification: The file “qti.txt” contains a question consisting of the lines of code, randomly rearranged. The correct answer is listed after the *a*. The system currently does not record all possible correct answers and only records the original order of lines.

Traceability: Requirements 1, 3-7, 13-14

PPALMS General-Purpose Execution Procedures

Execution Procedure A:

Step 1: Open terminal window and navigate to the directory containing the PPALMS executable files.

Step 2: Run executable by typing the following command into terminal:

./PPALMS <file_path> <lms> <question_type> <mutations> <lines_nums> …

Traceability: Requirements 1, 3-7, 13-14

Result: A file named "qti.txt" will be created containing the generated "rearrange" question, which is based on the given lines. The lines will be randomly mixed.

Verification: The file "qti.txt" contains a question consisting of the lines of code, randomly rearranged. The correct answer is listed after the *a*. The system currently does not record all possible correct answers and only records the original order of lines.

Traceability: Requirements 1, 3-7, 13-14

Execution Procedure B:

Step 1: Open terminal window and navigate to the directory containing the PPALMS executable files.

Step 2: Run the executable by typing the following command into terminal:

./PPALMS

Traceability: Requirement 2 - Stand-Alone System

Step 3: Load the source code file by typing the following command into terminal:

path <path to file>

* Note: Replace "<path to file>" with the name of the source code file you want to load.

Verification: path verified during generation in Step 8.

Traceability: Requirement 1 - Importing Source Code

Step 4: Select a target LMS by typing the following command into terminal:

lms <lms type>

* Note: Replace “<lms type>” with Canvas, Moodle, or Blackboard.

Verification: “success” is output to the terminal.

Traceability: Requirement 5 - Selecting the Intended LMS.

Step 5 : Set as reordering by typing the following command into terminal:

type <question type>

*Note: Replace “<question type>” with *rearrange*.

Verification: “success” is output to the terminal.

Traceability: Requirement 7 - Reordering Problems.

Step 6: Select mutation quantity by typing the following command into terminal:

mutations <# of mutations>

* Note: Replace “<# of mutations>” with a number.

Verification: “success” is output to the terminal.

Traceability: Requirement 12 - Maximum Number of Generated Questions.

Step 7: Select desired lines for question generation by typing the following the

lines <lines> <lines> …

* Note: Replace “<lines>” with either a single number or a range of numbers (e.g., 1-10 11 13-15). Each chunk of lines needs to be separated by a space.

Verification: “-- success --” is outputted to the terminal.

Traceability: Requirement 4 - Selecting Lines from Source Code

Step 8: Generate reordering questions by typing the following command into terminal:

generate

Verification: “-- success --” is output to terminal

Traceability: Requirement 3 - Original Source Code File Will Not Be Modified by the System
Requirement 6 - Problem Generation
Requirement 13 - Generated Problems Stored in System

Step 9: [Optional] Erase any generated questions by typing the following command terminal:

erase

Verification: “--success--" is output to the terminal.

Step 10: To obtain questions for export to selected LMS target type the following command into terminal:

```
export
```

Verification: “--success--" is output to the terminal.
Traceability: Requirement 14 - Generated Problems Output to a File

Step 11: After exporting questions close the program by typing the following command into terminal:

```
exit
```

Result: A file named “qti.txt” will be created containing the generated “rearrange” question, which is based on the given lines. The lines will be randomly mixed.

Verification: The file “qti.txt” contains a question consisting of the lines of code, randomly rearranged. The correct answer is listed after the *a*. The system currently does not record all possible correct answers and only records the original order of lines.

Traceability: Requirements 1, 3-7, 13-14