Test Case Scenario

Setup Context: User has a preferred file for PPALMS to use, User has Python with required modules (Tkinter) installed on their machine,

1. User interface opening screen component is instantiated when the splitclass.py program is run from the command line on the user's machine.

2. User sees PPALMS welcome screen with greeting “"Welcome to PPALMS! To make a new Problem Set, click 'Start'!" and two options: “Start” and “Exit PPALMS”
3. User clicks ‘Start’
4. The PPALMS Screen has two boxes that the user sees for input; the first is for the file path, and the second is for the unique source code name.
5. User types an input for the file path
6. User types an input for the source code name
7. User clicks on the button prompt “Import!” when both boxes are filled
8. PPALMS retrieves the inputs of both boxes
9. PPALMS checks if the file in the source code already exists under its unique name
10. PPALMS opens the file and checks that it exists, that it contains all ASCII characters, that the file is found, and that it can be read
    a. **Related Requirement(s): 1**
11. The file is successfully imported into PPALMS and PPALMS creates a solution code file with the name that the User typed into PPALMS as input in step 6.
12. User clicks the “Include/Exclude” button to indicate that they wish to select specific lines from the file to make the solution code.
13. User is prompted to fill in input for two text boxes which represent the line numbers of lower and upper bounds for the lines they wish to include/exclude.

    b. **Related Requirement(s): 2**

14. User inputs values into the boxes and clicks either “Include Lines” or “Exclude Lines”
15. PPALMS performs a series of checks on the user inputs. PPALMS checks that the lower and upper bounds are positive integers, that the lower bound is less than or equal to the upper bound, and that both bounds do not exceed the number of lines in the file.

    b. **Related Requirement(s): 2**

16. When the user designates lines that they wish to include/exclude, PPALMS removes the lines specified for exclusion, or the lines not selected for inclusion.

    b. **Related Requirement(s): 2**

17. PPALMS displays the content of the source code file with updated lines. The user is prompted to make tuples.

    c. **Related Requirement(s): 3**

18. User is prompted to fill in input for two text boxes which represent the line numbers of lower and upper bounds for the lines they wish to tuple.

    c. **Related Requirement(s): 3**

19. User clicks on the “Create Tuples” button

    c. **Related Requirement(s): 3**

20. PPALMS creates a solution code folder for the source code
21. PPALMS performs checks that both of the input boxes contain integers, that the second value is less than the first value, both of the lines are positive values, and that the lines are adjacent.

    c. **Related Requirement(s): 3**

22. User continues to add lines for tupling and clicks “Next” to indicate that they are finished tupling lines.

    c. **Related Requirement(s): 3**

23. A list of the integer tuples is added to the solution code configuration file within the solution code folder
24. PPALMS presents the user with an input box where they have to type their LMS selection. User may choose between Canvas, Blackboard, or Moodle

    d. **Related Requirement(s): 4**

25. User selects “Enter Selection” When have typed their LMS input
26. PPALMS adds LMS selection to the solution code configuration file

    d. **Related Requirement(s): 4**

27. PPALMS presents the user with an input box where they have to type their chosen question type

    e. **Related Requirement(s): 6**

28. PPALMS checks the solution code configuration file for the LMS type

    e. **Related Requirement(s): 6**

29. The user is presented with a list of available question types that are supported by their chosen LMS

    e. **Related Requirement(s): 6**

30. User selects “Enter Selection” when they have typed their question type input

Requirement #: 1 **Requirements Type**: TBD **Use Case**: Import File

Date: 10/03/2022

Author: Katie Scheck   **Source**: 09/26 interview

Introduction: In order for PPALMS to generate problem sets, the user must provide, as input, a file containing source code from their local machine.

Rationale: As per the customer interview on 09/26, the customer stated that PPALMS should accept any text-based file. Since PPALMS does not provide source code, PPALMS must be able to receive a file from the user's local machine as input in order for PPALMS to generate problem sets.

Inputs: A file of any type that the program accepts

Requirement Description: The system shall allow the user to import a file containing source code

The program will create problems from the file if and only if:

1. The imported source code is text-based[1]
2. The file is successfully imported into PPALMS
3. The imported source code does not share a name with any other source code files stored in PPALMS.

Outputs: Imported source code is available in PPALMS, a unique name is stored as an attribute of the imported source code in PPALMS and the source code is henceforth referred to as solution code.

Persistent Changes: None

Related Requirements: None

Conflicts: None

Support Materials: None

Test Cases: TBD

[1] ASCII-encoded data refers to a schema containing unique values for 128 alphabetic, numeric, or special additional characters and control codes. ASCII is the standard language of embedded systems.

Requirement #: 2	**Requirements Type**: TBD	**Use Case**: Select Inclusion/Exclusion Lines from File

Date: 10/03/2022

Introduction: After a user imports a file containing source code to PPALMS, they may want to include/exclude specific lines of the file. PPALMS will generate a problem set based off of the lines that were selected for inclusion/exclusion, respectively.

Rationale: A user may want to include/exclude specific lines of their file from the problem set. PPALMS shall be able to generate a problem set from the file that either excludes or includes the lines selected in the problem set.

Inputs: A file with a type accepted by PPALMS, a set of lines to be used from the file.

Requirement Description: The system shall allow the user to select specific lines from their file to include or exclude to create a problem set.

Outputs: An updated source code file within PPALMS derived from the lines not selected for exclusion.

Persistent Changes: The requested number of lines to exclude from the file.

Related Requirements: Requirement #3[2]

Conflicts: None

Support Materials: None

Test Cases: TBD

[2] Completion of requirements #2 and #3 make ‘solution code’ from the source code

Requirement #: 3

Requirements Type: TBD

Use Case: Form and Order Lines as Tuples

Date: 10/03/2022

Introduction: Coupling adjacent lines of code together as a tuple is a useful feature when creating Parson's Problems. PPALMS must support this functionality.

Rationale: A customer interview conducted on 09/26 indicates that some lines need to be linked together in a fixed order for cases where line order is necessary to the context of the problem and cases where line order does not change the solution, so a successful product will allow the customer to create tuples of connected lines with defined ordering that will be treated as "one line" of code.

Author: Christine Brennan **Source**: 09/26 Interview

Inputs: Imported Source Code, Line Numbers for Coupling

Requirement Description: The system shall allow the user to form and order line tuples

1. The system will form lines into a tuple if and only if:
    a. The professor requests that the lines become part of a tuple
    b. The lines requested are adjacent
        i. If the lines are not adjacent, the system assumes they are separate tuples
2. After tuples ares formed, adjust line order accordingly, meaning:
    a. assigning the 'line number' of the tuple to the lowest 'line number/index' of the elements of the tuple
    b. increasing the 'line number/index' of the following lines until the 'line number/index' of every line is one more than the line that precedes it and one less than the line that follows it.

Outputs: Requested LMS[3], Requested Question Type

Persistent Changes: Number of lines (decreases by the sum of the numbers of lines in tuples and the number of tuples), line order (specific 'line

[3] Different LMS are only compatible with specific file extensions and file types.

numbers/indices')

Related Requirements: Requirements #1 & #2 (satisfactory source code as input)[4], Requirement #4 (result of requirement necessary to continue)

Conflicts: None, TBD

Support Materials: None, TBD

Test Cases: TBD

[4] Completion of requirements #2 and #3 make 'solution code' from the source code

Requirement #: 4	**Requirements Type**: TBD	**Use Case**: Select Compatible Question Type

Date: 10/03/2022

Introduction: The most common action performed by the system is to allow instructors to choose from various question types compatible with the LMS for Canvas, Moodle, or Blackboard. PPALMS must support this functionality.

Rationale: A customer interview conducted on 09/26 indicates that functionality for the user to reorder problems is highly desirable, but not compatible with all LMS. The success of the product hinges on the professor choosing the best compatible question type for their source code.

Author: Christine Brennan **Source**: 09/26 Interview

Inputs: LMS[5], Question type, Solution File[6]

Requirement Description: The system selects the question type they want to be produced from their solution file.

1. The system will produce the problem set of the selected question type after selection if and only if:
    a. The question type is compatible with their chosen LMS (Canvas, Moodle, or Blackboard) (See Requirement #6).
    b. A solution file has to be uploaded and any necessary tuples/insertions/exclusions have been annotated.[7]
2. If a question type is compatible, the system will proceed towards producing the problem set for the user from the following (Continue to Req #5).
    a. Reordering (most highly desired by customer), in which the order that the lines and tuples are correctly organized is changed.
    b. Multiple choice questions that provide a number of incorrect ‘mutant’ answers for each line or tuple.
    c. For Python only, indentation problems will be implemented as 2-D

[5] Design choice: the user will be prompted to select their desired LMS before they are able to select a question type for their problem set to be generated off of. See Requirement #6 for details.

[6] Design choice: will source code be uploaded and annotated before the user picks a question type? If it is uploaded beforehand, the system could determine which question types are compatible because it would already know the restraints/features of the language used. On the contrary, a user could reasonably be expected to know which type of question they want and know it is compatible with their file type before they decide to run the software

[7] Also dependent upon design choice as mentioned in previous footnotes.

Parson’s Problems.

d. “Find the bug” problems, where the upload lines are replaced with mutants.

e. Fill-in-the-blank problems, where one term (variable, operator, character) is removed from the line.

Outputs: Requested Question Type

Persistent Changes: TBD

Related Requirements: Requirements #1-#3 (Satisfactory source code), Requirement #5 (Output is satisfactory for continuing software), Requirement #6 (LMS)

Conflicts: None, TBD

Support Materials: None, TBD

Test Cases: TBD

Requirement #: 5	**Requirements Type**: TBD	**Use Case**: Generate Variations

Date: 10/03/2022

Introduction: Generating variations of questions from solution code is the heart of Parson's Problems and in turn, PPALMS.

Rationale: Customer interview on 09/26 indicates that the highest priority requirement of PPALMS is that variations of the solution code will need to be created and accessible by clicking on 'Recently Added'. The customer also stated the system need not grade or check accuracy of problems, but it should generate all possible variations based on the selected problem type.

Author: Christine Brennan **Source**: 09/26 Interview

Inputs: Imported Solution Code, Requested problem type

Requirement Description: The system shall allow the user to generate variations of problems from the source code

1. The system will have methods of generation for each problem type.
    a. Reordering (most highly desired by customer), in which the order that the lines and tuples are correctly organized is changed.
    b. Multiple choice questions that provide a number of incorrect 'mutant' answers for each line or tuple.
    c. For Python only, indentation problems will be implemented as 2-D Parson's Problems.
    d. "Find the bug" problems, where the upload lines are replaced with mutants.
    e. Fill-in-the-blank problems, where one term (variable, operator, character) is removed from the line.
2. The problem set generated will contain every possible solution where each element is a distinct problem.

Outputs: Requested Learning Management System (LMS), Requested Question Type

Persistent Changes: The generated problem set is added to 'Recently Added'

Related Requirements: Requirements #1-#3 as satisfactory imported source code, Requirement #4 as requested problem type, For use in Requirement #6

Conflicts: None, TBD

Support Materials: None, TBD
Test Cases: TBD

Requirement #: 6	**Requirements Type**: TBD	**Use Case**: LMS Selection

Date: 10/03/2022

I**ntroduction**: The Parson's Problems need to be usable in the user's desired Learning Management System (LMS). LMS selections may be Canvas, Moodle, or Blackboard. Therefore, the user must be able to request an LMS target.

Rationale: Customer interview on 09/26 indicates that different LMS will require different file types and only be compatible with certain file types. For this reason, it is useful for the software to track which LMS the user wants their problem set to be compatible with.

Author: Christine Brennan **Source**: 09/26 Interview

Inputs: Solution Code, LMS Selection (from user)

Requirement Description: The system shall allow the user to select the target LMS.

1. PPALMS will present the user with their LMS target options including Canvas, Moodle, and Blackboard.
2. The user will be able to indicate which of the listed LMS targets they would like their problem set to be compatible with.

Outputs: Requested LMS (Stored as an attribute of the problem set generation process)

Persistent Changes: LMS selection changes the compatible file type options for exporting the problem set

Related Requirements: Requirement #4[8], Requirement #7

Conflicts: Requirement #4[9]

Support Materials: None, TBD

Test Cases: TBD

[8] Because some question types are not compatible with all LMS, access to the desired LMS before creating a problem set of a certain type may be desired.

[9] Also dependent upon design choice as mentioned in previous footnotes.

Requirement #: 7 **Requirements Type**: TBD **Use Case**: Create compatible output file

Date: 10/03/2022

Introduction: In order to implement the created Parson's Problems into an LMS, the user must be able to export the set of Parson's Problems to their local machine. PPALMS should support this functionality.

Rationale: A customer interview conducted on 09/26 indicates that the outputs of PPALMS must be compatible with LMS Canvas, Blackboard, and Moodle. Upon export, the output file must be local to the user's machine and importable to an LMS.

Author: Audrey Gasser **Source**: 09/26 Interview

Inputs: Export file type, local machine destination for output file

Requirement Description: The system shall create a problem set output file of a compatible file extension type for inclusion in the previously selected LMS.

1. Export file types that are not compatible with the previously selected LMS will not be displayed to the user when selecting the export type.
2. If the LMS that the user selected is "Blackboard Learn", the user may export the problem set output file in the following formats:
    a. IMS QTI 2.1 Package (.zip)
    b. Microsoft Word (.doc, .docx)
    c. HTML5 (.htm, .html)
    d. PDF (.pdf)
    e. Rich Text Format (.rtf)
3. If the LMS that the user selected is "Canvas", the user may export the problem set output file in the following formats:
    a. IMS QTI Package (.zip)
    b. IMS QTI 1.1+ (.xml)
    c. Microsoft Word (.doc, .docx)
    d. HTML5 (.htm, .html)
    e. PDF (.pdf)
    f. Rich Text Format (.rtf)
4. If the LMS that the user selected is "Moodle", the user may export the problem set output file in the following formats:
    a. Blackboard POOL (.dat, .zip)
    b. Blackboard QTI (.dat, .zip)
    c. Microsoft Word (.doc, .docx)

d. HTML5 (.htm, .html)
e. Text File (.txt)
f. GIFT (.zip)
g. Moodle XML Format (.xml)

Outputs: Requested Output File for LMS

Persistent Changes: None / Output file downloaded to user's local file system

Related Requirements: Requirement #6[10]

Conflicts: None, TBD as requirements evolve

Support Materials: None, TBD

Test Cases: TBD

[10] Because some question types are not compatible with all LMS, access to the desired LMS before creating a problem set of a certain type may be desired.

Requirement #: 8	**Requirements Type**: TBD	**Use Case**: Reuse generated problem set

Date: 10/03/2022

Introduction: In order to allow for duplicate downloads of a previously created problem set, PPALMS must store recent Parson's Problems sets the user can select from. PPALMS should support this functionality.

Rationale: A customer interview conducted on 09/26 indicates that users request a 'Recently Added' to be available with PPALMS. With this functionality, the user does not need to reupload and configure source code each time the system is used. The user wants to be able to view and export problem sets that they created previously.

Author: Audrey Gasser

Source: 09/26 Interview

Inputs: Problem set

Requirement Description: The system shall allow the user to reuse generated problem sets.

1. If the problem set has been successfully generated prior, the user may select a generated problem set from 'Recently Added' for duplicated export.
2. The user shall not be able to change the question type of the generated problem set. (Req. 4)

Outputs: A window displaying the selected problem set from 'Recently Added'.

Persistent Changes: Reuse of a problem set brings it to the most recent position in 'Recently Added'

Related Requirements: Requirement #5[11]

Conflicts: None, TBD as requirements evolve

Support Materials: None, TBD

Test Cases: TBD

[11] Problem sets are available in 'Recently Added' after problem generation