Test

Before scan through the test cases. it is better to have a look at the "Entry Data" section at the bottom of the page, or the "User Guide" page or Readme in github. They will offer an intuitive view of the process to you.

User Stories related to parameters will not have test cases since there is no operation towards them and will be directly displayed in the output file after command input. Parameter columns in green in the user story page are those our team extracted succesful, shown well in the output csv file.

US01: I want parameter values to be output when running the program, as a csv

TC01: Output as CSV(Successful)

Test Type: Functional	Execution Type: Manual
Objective: Verify if the output of our program is demonstrated as a CSV file	
Pre-Conditions: N\A	
Notes: Enter "--format csv" followed correct command, or simply "-f" "--f" works but not a formal command.	
Post-Conditions: A .csv file is generated.	
Time Constraint: Minimum: 20 min, Maximum: 25 min	

TC02: Output as CSV(Unsuccessful)

Test Type: Functional	Execution Type: Manual
Objective: Verify application behavior when the csv file can not be generated	
Pre-Conditions: N\A	
Notes: Enter command with invalid format Uppercase letter "CSV" is not allowed Double dash should be used if the full form format is used: "-format csv" is not allowed. Any space is invalid except the one before csv: "-- format csv". "- -format csv" The program does not run when the csv is open	
Post-Conditions: A .csv file is not generated and there is an error	
Time Constraint: Minimum: 20 min, Maximum: 25 min	

US02: I also want each parameter value to be evaluated with a pass /fail result and recorded in the csv output

TC03: Pass/Fail result are recorded in csv(Successful)

Test Type: Functional	Execution Type: Manual
Objective: Verify if the output of the program contains pass/fail result	
Pre-Conditions: N\A	
Notes: Following the commands developers provide, the column named "Parameter Evaluation" in csv refers to pass/fail results. "NOT IMPLEMENTED" is shown in the columnto parameters that developers have not extracted. "VMAT unknown" only appears when the parameter is not known in VMAT dicom file. Commands will be shown in the "Entry Data" section below. You can find them in "User Guid page" and "Readme" in github	
Post-Conditions: The output contains a pass/fail result	
Time Constraint: Minimum: 20 min, Maximum: 25 min	

TC04: Pass/Fail result are recorded in csv(unsuccessful)

Test Type: Functional	Execution Type: Manual
Objective: Verify if the output of our program does not contain pass /fail result	
Pre-Conditions: N\A	
Notes: Following the correct instruction will make this test successful. If users do not follow the guidence carefully, they will meet errors and can not see the output csv. PASS/FAIL result is always there if user successfully run the program. Commands will be shown in the "Entry Data" section below. You can find them in "User Guid page" and "Readme" in github	
Post-Conditions: A .csv file is not generated and there is an error, so the pass/fail result can not be seen.	
Time Constraint: Minimum: 20 min, Maximum: 25 min	

US03: I want the value acceptance ranges to be output as well

TC05: The extracted value and the true value of the parameter are contained in the output file(Successful)

Test Type: Functional	Execution Type: Manual
Objective: Verify if the output of the program contains the extracted value and the true value of the parameter	
Pre-Conditions: N\A	

TC06: The extracted value and the true value of the parameter are contained in the output file(Unsuccessful)

Test Type: Functional	Execution Type: Manual
Objective: Verify if the output of the program does not contain the extracted value and the true value of the parameter	
Pre-Conditions: N\A	

Notes: Following the commands developers provide, the column named "Parameter Value" and "Parameter Solution" refer to the value extracted by developers and the true value in truth table respectively. Commands will be shown in the "Entry Data" section below. You can find them in "User Guid page" and "Readme" in github
Post-Conditions: The extracted value and the true value of the parameter can be clearly seen in the output file
Time Constraint: Minimum: 20 min, Maximum: 25 min

US04: It would be useful that the range of acceptable values will automated modified depending on the file

TC07: Different file will have different extracted value and the true value of the parameter in the output file(Successful)

Test Type: Functional	Execution Type: Manual
Objective: Verify if the range of acceptable values in output can automated changed based on different files	
Pre-Conditions: N\A	
Notes: Following the commands developers provide, the column named "Parameter Value" and "Parameter Solution" refer to the value extracted by developers and the true value in truth table respectively. With different file input, different range of acceptable values will dispalyed in the output. Commands will be shown in the "Entry Data" section below. You can find them in "User Guid page" and "Readme" in github	
Post-Conditions: The range of acceptable values in output can automated changed based on different files	
Time Constraint: Minimum: 20 min, Maximum: 25 min	

US05: I want to be able to specify the case of the files I'm processing (necessary if partial parameter extraction)

TC09: Cases can be choosed by user(Successful)

Test Type: Functional	Execution Type: Manual
Objective: Verify if users can select case number	
Pre-Conditions: N\A	
Notes: If no case number argument in the command: Question "What is the case number for docX" will be asked in the command line as same time as the number of files queried. If case number argument in the command: For example, "--case_number 6", then case 6 of all files will be checked and outputed. ("-c 6" also works) When user want to choose different case for differernt file, they can simply do not enter case number in the command then waiting for the prompt question as mentioned above. Otherwise, new command can be used if users want to get the output only by typing command: "Resources/Input/YellowLvlIII_7a.dcm,1 Resources/Input/YellowLvlIII_7b.dcm,2", as part of the command, will show case 1 for file X and case 2 for file Y.("Resources/Input,1 OTHER_DICOM_FOLDER,2" is feasible) Commands will be shown in the "Entry Data" section below. You can find them in "User Guid page" and "Readme" in github	
Post-Conditions: The case number is specified	

Notes: Following the correct instruction will make this test successful. If users do not follow the guidence carefully, they will meet errors and can not see the output csv. Therefore, the accpetance ranges can not be seen. Commands will be shown in the "Entry Data" section below. You can find them in "User Guid page" and "Readme" in github
Post-Conditions: A .csv file is not generated and there is an error, so the accpetance ranges can not be seen.
Time Constraint: Minimum: 20 min, Maximum: 25 min

TC08: Different file will have different extracted value and the true value of the parameter in the output file(Unsuccessful)

Test Type: Functional	Execution Type: Manual
Objective: Verify if the output of the range of acceptable values can not automated changed based on different files	
Pre-Conditions: N\A	
Notes: Following the correct instruction will make this test successful. If users do not follow the guidence carefully, they will meet errors and can not see the output csv. Therefore, the changed range of acceptable values can not be seen. Commands will be shown in the "Entry Data" section below. You can find them in "User Guid page" and "Readme" in github	
Post-Conditions: The range of acceptable values in output can not automated changed based on different files	
Time Constraint: Minimum: 20 min, Maximum: 25 min	

TC10: Cases can be choosed by user(Unsuccessful)

Test Type: Functional	Execution Type: Manual
Objective: Verify if users can not select case number	
Pre-Conditions: N\A	
Notes: Enter command with invalid format Uppercase letter "CASE_NUMBER" is not allowed, as well as CASE_number, case_number Double dash should be used if the full form format is used: "-case_number 6" is not allowed. case-num is not allowed. Any space is invalid: "-- case_number 6". "- -case_number 6" Commands will be shown in the "Entry Data" section below. You can find them in "User Guid page" and "Readme" in github	
Post-Conditions: The case number is not specified and the program can not run	
Time Constraint: Minimum: 20 min, Maximum: 25 min	

Time Constraint: Minimum: 20 min, Maximum: 25 min

US06: I want to be able to process batches of dicoms at a time

TC11: Multiple input(Successful)

Test Type: Functional	Execution Type: Manual
Objective: Verify if mutiple files can be entered in one command	
Pre-Conditions: N\A	
Notes: Enter correct file batches argument of the command: "--inputs Resources/Input", processing all files in the folder; "--inputs Resources/Input OTHER_DICOM_FOLDER", processing all files in both folders; "--inputs Resources/Input/YellowLvlIII_7a.dcm Resources/Input /YellowLvlIII_7b.dcm", processing any number of separate dicoms Abbreviation “-i” is feasible Commands will be shown in the "Entry Data" section below. You can find them in "User Guid page" and "Readme" in github	
Post-Conditions: It is able to process batches of files at a time	
Time Constraint: Minimum: 20 min, Maximum: 25 min	

TC12: Multiple input(Unsuccessful)

Test Type: Functional	Execution Type: Manual
Objective: Verify if mutiple files can not be entered in one command	
Pre-Conditions: N\A	
Notes: Space is not allowed for OTHER_DICOM_FOLDER path, for example, "Dicom file" is not allowed, need to be replaced by "Dicom_file" Enter command with invalid format: Uppercase letter "INPUT" is not allowed, as well as "-I" Double dash should be used if the full form format is used: "-input" is not allowed. Any space is invalid: "-- input". "- -input" Commands will be shown in the "Entry Data" section below. You can find them in "User Guid page" and "Readme" in github	
Post-Conditions: It is not able to process batches of files at a time	
Time Constraint: Minimum: 20 min, Maximum: 25 min	

Entry Data

Field	Value	Full Command
US01, TC01, TC02	"--format csv", "-f csv"	**python app.py --inputs Resources/Input --format csv**;(Process all dicoms in folder, with case number prompt for each dicom found) **python app.py --inputs Resources/Input --format csv --case_number 6**;(Process all dicoms in folder, treating each as a case 6) **python app.py --inputs Resources/Input OTHER_DICOM_FOLDER --format csv --case_number 6**;(Process all dicoms in both folders, treating all dicoms found as case 6) **python app.py --inputs Resources/Input/YellowLvlIII_7a.dcm Resources/Input/YellowLvlIII_7b.dcm --format csv**;(Process any number of separate dicoms, with case number prompts for each) **python app.py --inputs Resources/Input/YellowLvlIII_7a.dcm,1 Resources/Input/YellowLvlIII_7b.dcm,2 --format csv**;(Process any number of separate dicoms, giving case numbers for each (format: dicom,case, dicom2,case ... etc) **python app.py --inputs Resources/Input,1 OTHER_DICOM_FOLDER,2 --format csv**(Process any number of folders, giving case numbers for each (format: folder,case, folder2,case ... etc)
US05, TC09, TC10	"--case_number 6", "-c 6", "Resources/Input/YellowLvlIII_7a.dcm,1 Resources/Input /YellowLvlIII_7b.dcm,2" "Resources/Input,1 OTHER_DICOM_FOLDER,2"	
US06, TC11, TC12	"--inputs Resources/Input", "--inputs Resources/Input OTHER_DICOM_FOLDER", "--inputs Resources/Input/file1 Resources/Input/file2"	

Commands for different scenarios are list above.