System Test Plan

for

LScan

Version 1.0

Prepared by Sohan Tamang

PSU, Fall Capstone 2018

1-18-2019

Table of Contents

Revision History

Date	Reason For Changes	Version
1-18-2019	Creation	0.1
3-10-2019	Revision	0.2
	Version 1	1.0

1. Introduction

1.1. Purpose

This is a system testing plan document for LScan software. The purpose of this document is to provide plans and strategies for team members to follow to performing System Test on LScan. This document also serves as acceptance criteria for features during the development process.

1.2. Objectives

The objectives of this document are:

- Meet the client's requirements and specifications
- Produce a functional, user-friendly, and bug-free product

1.3 Minimum Viable Product

1. Product will execute on a standard commercial computer (basically, it will not require an exotic computing platform).
2. Conversion of STL scan into compliant[1] LDraw part definition that will use LDraw primitives and subparts definitions where appropriate.
3. The converted LDraw part definition can be seamlessly used by most (if not all) existing CAD programs that use the LDraw parts library. The initial selection of CAD programs to be used for testing will include Bricksmith and LeoCAD.
4. A minimal user interface with the following features
    a. GUI (not command line) that uses standard GUI paradigms (mouse control, menus, buttons, etc.).
    b. Allows file selection of an STL file to be converted. File selection process should highlight allowed file extensions.
    c. Allows for entry of new LDraw part metadata, as well as the ability to edit existing LDraw part metadata. Where possible, fields should be pre-populated. Allows user to edit pre-populated fields. At a minimum, the metadata required by the LDraw parts organization will be added to resulting output LDraw file. A link to this part of the specification is posted in the references section of this document.
    d. Where possible, automatically creates the converted LDraw part file name consistent with the part under conversion. Allows user edit of file name.
    e. "Go", "Pause" and "Cancel" buttons
    f. Aesthetics (like animated transitions or fancy logos), while they enhance the professional appearance, are not necessary. However, the UI should be perceived as "usable" (among other things, this implies a responsive and performant interface).
    g. Localization is not required. The user interface will be in en-US.

[1] In this context, compliant means fully adheres to the LDraw Parts Specification.

5. A mechanism for comparing official LDraw part definition with the converted LDraw part definition. *The exact measure of the "comparison" is to be determined*.

2. Testing Strategy

2.1. Testing Strategy

System testing activities will cover the testing of new functionalities, modified functionalities, workflows, internal and external interrupts.

2.2. Testing Types

2.2.1. Functional Testing

Functional testing ensures that each component meets the functional requirements mentioned in the requirement document.

2.2.2. Usability Testing

The objective of this test is to ensure that the user interface is responsive and easy to navigate.

2.3. Testing Suspension Criteria

System testing will be suspended if the LScan program fails to open at all. The system test will be suspended until the issue is fixed with the permission of the team lead.

2.4. Test Data

Appropriate test cases can be grouped into test scenarios. Data will be prepared to complete the testing activities. However, manual testing can be performed where test data are not required/feasible.

3. Execution Plan

The execution plan will detail the test cases to be executed. The Execution plan will be put together to ensure that all the requirements are covered. The execution plan will be designed to accommodate some changes if necessary. if testing is incomplete on any day. All the test cases of the projects under test in this release are arranged in a logical order depending upon their interdependency.

3.1. Test Cases

3.1.1. Test Case 1: Cross-Platform Test

Purpose:
Verify that the LScan software operates on popular platforms

Requirement Traceability:
MVP requirement : 1

Setup:

1. Obtain the most current version of the LScan software
2. Follow the installation instructions to install the software
3. Locate the LScan software

Test Data

Action	Expected Output
Run all the test cases in MacOS	LScan should open successfully
Run all the test cases in Windows OS	LScan should open successfully

3.1.2. Test Case 2: UI - Mouse Control Test

Purpose:
Verify that the LScan software responses to mouse controls.

Requirement Traceability:
MVP requirement : 4

Setup:
1. Open LScan software
2. Use a connected standard mouse to test the following.

Test Data

Action	Expected Output
Click Info button	Software information should be displayed
Drag the software to different locations	Opened software should move to accordingly
Resize the software panel using the mouse	The panel should size should change

3.1.3. Test Case 3: UI - File Browser Test

Purpose:
Verify that the *BROWSE FILE* button gets the path of the correct file.

Requirement Traceability:
MVP requirement : 4

Setup:
1. Launch LScan
2. Click *BROWSE FILE* on the panel

Test Data

Action	Expected Output
Do nothing (default)	A file explorer window should pop up
Browse to a folder with different file types	Only files with .stl extension should be displayed
Select a file to convert	The file path should be copied to the LScan display

3.1.4. Test Case 4: Error Handling Test

Purpose:
Verify that the software does not crash while an error occurs

Requirement Traceability:
MVP requirement : 4 f

Setup:

1. Launch LScan

Test Data

Action	Expected Output
Enter Invalid File Path, enter metadata and press CONVERT	Prompt *invalid file path* error message. Software should not crash Metadata information should persist
Convert non-STL file with valid parameters	Prompt *invalid file type* error message. Software should not crash Metadata information should persist
Convert a **.STL* file with invalid file contents	Prompt the error message with a reason. Software should not crash Metadata information should persist

3.1.5. Test Case 5: LDraw File Compliance Test

Purpose:
Verify that the converted file complies with the LDraw file specification 1.0.2

Requirement Traceability:
MVP requirement : 2

Setup:
1. Launch LScan
2. Click *BROWSE FILE* on the panel and select a valid STL file
3. Fill up the metadata information
4. Click CONVERT
5. Save the file

Test Data

Action	Expected Output
Check if the converted file is UTF-8 format	It should pass the test
Check the first character in each line	It should be a number between 0 and 5

Read the lines that start with (1 - 5)	It should only contain numbers
Read the metadata in the header	It should follow LDraw header specification and all the metadata entered by the user

3.1.6. Test Case 6: Metadata Edit Test

Purpose:
Verify that the user can update existing metadata

Requirement Traceability:
MVP requirement : 4 c

Setup:
1. Launch LScan
2. Click *BROWSE FILE* on the panel and select a valid STL file
3. Fill up the metadata information
4. Click CONVERT
5. Complete the Conversion

Test Data

Action	**Expected Output**
Update the metadata	Should save the metadata
Click SAVE Conversion	Should save the file
Open the saved file	The header should contain updated metadata

3.1.7. Test Case 7: CAD Software Test

Purpose:
Verify that the converted ldraw file can be opened using Bricksmith and LeoCAD software

Requirement Traceability:
MVP requirement : 4

Bricksmith

Setup:

1. Convert a STL file using LScan
2. Locate the converted *.dat file
3. Open Bricksmith software

Test Data

Action	Expected Output
Form Bricksmith go to File -> Open and open the converted LDraw file	The converted model should be rendered
Rotate the model into 360 degrees	Should be rotatable

LeoCAD

Setup:

1. Convert a STL file using LScan
2. Copy the converted DAT file to LDraw parts folder
3. Open LeoCAD

Test Data

Action	Expected Output
Search for the converted LDraw file from parts and drag and drop it to the project canvas	The converted model should be rendered
Rotate the model into 360 degrees	Should be rotatable

3.1.8. Test Case 8: Activity Logs Module Test

Purpose:

Verify that the periodic activity logs are displayed to the user.
Verify SAVE log works

Requirement Traceability:

MVP requirement : 4 f

Setup:

1. Open LScan
2. Browse file to convert and enter metadata

Test Data

Action	Expected Output
Click CONVERT	Log panel should display the current process
Wait for 5 seconds	Logs should be updated
Click SAVE LOG button	System file save dialog box should appear

3.1.9. Test Case 9: LScan Termination Test

Purpose:
Verify that clicking X closes the LScan regardless of the current state

Requirement Traceability:
MVP requirement : 4 f

Setup:
1. Open LScan

Test Data

Action	Expected Output
Click X in the GUI	Should close the LScan program immediately
Browse File, fill metadata and Click X	Should close the LScan program immediately
Start the conversion and press X	Should close the LScan program immediately

3.1.10. Test Case 10: Metadata Settings Test

Purpose:
Verify that the metadata panel remembers last used metadata settings.

Requirement Traceability:
MVP requirement : 4 c

Setup:
1. Launch LScan

2. Browse input file
3. Browse output file
4. Enter a new author **Anthony Namba**
5. Change the **License**
6. Close LScan
7. Launch LScan

Test Data

Action	**Expected Output**
Click Browse output	Takes to the last directory
Click Output	Takes to the last directory
Inspect Author	Anthony Namba appears
Inspect License	Remembers last license info

3.1.11. Test Case 12: Pause and Stop Button Test

Purpose:
Verify the PAUSE and STOP buttons behaves as expected

Requirement Traceability:
MVP requirement : 4 e

Setup:
1. Launch LScan
2. Fill Metadata
3. Click CONVERT

Test Data

Action	**Expected Output**
Click PAUSE	Conversion should be paused with log message displayed PAUSE should change to RESUME
Click RESUME	Should resume the conversion with log message displayed

	RESUME should change to PAUSE
Click STOP	It should Stop the conversion

4. Defect Reporting

4.1. Defect Types

A defect can be classified as a Bug or an Issue. A bug is a defect in the development process whereas an issue can be an issue with design or infrastructure.

4.2. Defect Severity

Critical	● Catastrophic or severe impact on the system functionality ● The manual process is costly to implement to remedy the defect
Medium	● Does not hinder the system functionality. ● Little to medium manual process can remedy the defect
Low	● Little to no impact on the system functionality ● Examples: Wrong color, fonts, typos etc

4.3. Defect Management

Defects will be tracked in weekly System Test Reports. An issue will be created in GitHub repo. An issue will be in the following format:

Defect Description

Describe the issue here

Reproduction Steps

List the steps to reproduce the defect. This section is very important for developers to correctly reproduce the defect.

Acceptance Criteria

Write the acceptance criteria that should be met to resolve the defect.

5. Test Environment

The purpose of this section is to instruct software testers to set up environments to perform all the system tests. This section lists all the necessary hardware and software setup to prepare for testing.

5.1. Hardware Setup

- A standard computer
- A standard keyboard
- A standard mouse

5.2. Software Setup

Hardware

System tests will be performed on three major operating systems. Testers should perform all the system tests in following operating systems:

- Windows OS
- Mac OS

Software

The moonshot of this project is to create an installer with users can download and install easily. However, before the software release, all the tests will be performed with manual setup. Software required to perform system testing are:

- Python 3.7 with standard packages
- Other packages mentioned in requirements.txt
- PyCharm
- Bricksmith and LeoCAD

6. Test Schedule

System test will be performed by the end of the sprint on Sundays. This allows the team some time to report and groom the critical defects that need immediate work. The tickets that were completed in a sprint will be grouped together into three modules User Interface module, Threading module, and a Conversion module for the purpose of system testing.

7. Disclaimers

This project is not endorsed, sponsored or associated with The LEGO™ Group. LEGO® is a trademark of the LEGO Group of companies.

This project is not endorsed or sponsored by the LDraw organization.