

Spectrum Analyzer Analysis Tool

Test Document

Team 5

Masood Afzali

Ashly Altman

Brooke Ebetino

Tyler Haley

Joey Thompson

12/01/2023

Kennesaw State University Department of Software Engineering and Game Development

Course:

Dr. Yan Huang

1. Testing Strategy

1.1 Overall strategy

Unit testing will be performed on individual functions using pytest. Integration testing will ensure components work together properly. System testing will validate the entire workflow end-to-end. Regression testing will be done after code changes to detect new issues.

Testing will be led by the Testing Lead and performed by all team members. Testing will begin once the backend framework is complete.

1.2 Test Selection

Unit test cases will target key functions using statement coverage to exercise different branches. Integration test cases will focus on module interfaces using boundary value analysis. System test cases will follow main success scenarios and edge cases.

1.3 Adequacy Criterion

Unit tests will target 70% statement coverage. Integration and system tests will cover all boundary values for input parameters and error conditions.

1.4 Bug Tracking

Bugs will be tracked using Jira and tagged with priority, severity, and component tags.

1.5 Technology and Tools

- pytest for test automation
- Jira for bug tracking

2. Test Cases

Test Case	Purpose	Steps	Expected Result	Actual Result	Pass/Fail
TC1_ValidVideo	Validate analysis completes successfully on valid video	1. Place spectrum analyzer video of supported format in input folder 2. Run analysis code	Analysis completes successfully without errors Processed video CSV file is generated in output folder	CSV file generated successfully	Pass
TC2_InvalidVideoFormat	Validate error handling for unsupported video formats	1. Place video file of unsupported format (e.g. WMV) in input folder 2. Run analysis code	"Unsupported format" error message is displayed Analysis does not start	"Unsupported format" warning displayed Analysis did not start	Pass
TC3_CorruptedVideo	Validate error handling for	1. Place corrupted spectrum analyzer video file in input	"Video corrupted" error message is displayed	"Video corrupted" warning displayed	Pass

	corrupted video files	folder 2. Run analysis code	Analysis does not start	Analysis did not start	
TC4_MissingVideo	Validate error handling for missing video file	1. Delete video file from input folder 2. Run analysis code	"No video found" error message is displayed Analysis does not start	"No video found" warning displayed Analysis did not start	Pass
TC5_Frequency Accuracy	Validate accuracy of frequency value calculation	1. Configure generator with known frequency signal 2. Record expected frequency value 3. Run configured video 4. Load output CSV file 5. Compare frequency value to expected	Frequency value in CSV file matches configured expected value within $\pm 1\text{Hz}$ tolerance	Frequency value in CSV matched expected within $\pm 1\text{Hz}$	Pass
TC6_Amplitude Accuracy	Validate accuracy of amplitude value calculation	1. Configure generator with known amplitude signal 2. Record expected amplitude value 3. Run configured video 4. Load output CSV file 5. Compare amplitude value to expected	Amplitude value in CSV file matches configured expected value within $\pm 1\text{dB}$ tolerance	Amplitude value matched expected within $\pm 1\text{dB}$	Pass
TC7_FrameRateHandling	Validate correct processing of different frame rates	1. Configure 30fps and 60fps spectrum analyzer videos 2. Run videos 3. Check number of rows in CSV files	Number of rows in CSV proportional to frame rates 30fps video has half as many rows as 60fps video	30fps CSV had half the rows of 60fps CSV	Pass
TC8_GridDetection	Validate spectrum analyzer grid is detected	1. Use video without grid overlay 2. Process video	"No grid detected" warning message is displayed in console	"No grid detected" warning displayed	Pass

		3. Check for grid detection warning message			
--	--	---	--	--	--

3. Test Results

Testing was performed from November 1, 2023, to November 30, 2023, on v1.0 of the SpecVidAnalyzer application.

Environment

- OS: Windows 10
- CPU: Intel Core i7-9700K
- RAM: 32GB
- Python 3.7

Scope

A total of 102 test cases were executed covering:

- Unit Tests: 32 tests
- Integration Tests: 20 tests
- System Tests: 40 tests
- Regression Tests: 10 tests

Metrics

- Total test cases executed: 102
- Test cases passed: 94
- Test cases failed: 8
- Total defects discovered: 6
- Defects resolved: 5
- Outstanding known defects: 1

Overall Metrics

- Percentage of tests passed: 92.2%
- Functional test coverage: 85%
- Structural test coverage:
 - Unit test statement coverage: 73%
 - Integration test branch coverage: 90%

Discovered Defects

#	Description	Priority	Status
D1	Grid overlay not detected on some videos	High	Fixed in v1.0.1
D2	Amplitude accuracy drops for high frequency signals	Medium	Fixed in v1.0.2
D3	Application crash when video frame rate too low	High	Fixed in v1.0.2
D4	Incorrect timestamp if video > 4GB	Low	Fixed in v1.0.3
D5	UI freezes during large video processing	Medium	Fixed in v1.0.3
D6	Truncated CSV file if system runs out of disk space	Low	Open

Summary

Overall, the testing was successful according to target metrics. Most defects were minor and fixed promptly without significantly impacting delivery timelines. One known medium priority defect remains outstanding related to disk space constraints. Further optimization of video processing performance is recommended as the next priority.