**Project Acceptance Tests**

Document Matcher for Cheating Detection

Course: CITS3200  
Supervisor: A/Prof Guy Curtis

Group: 11

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

The purpose of this document is to outline the testing strategy for *Documatch*, an RSID Code Detection and Analysis Web App. This program compares RSID codes between two or more Word Documents, calculates the percentage of common RSID codes, and visualises the results using a user-friendly interface and instructions. The detailed tests aim to ensure the project meets the specified requirements, functions as intended and provides an intuitive and reliable result to the EU (End User).

**Test A – RSID Extraction & Detection**

This test focuses on the extraction functionality of *Documatch*. The purpose of this test is to verify that the program can accurately detect the existence of and extract RSID codes from Word documents (*.docx* format) that reflect the accuracy of the RSID extraction process. Any subsequent percentage calculation and visualisation steps rely on this so it must remain accurate across a series of scenarios.

**Specification**

**Requirements**

* The program must identify and extract all RSID Codes from a given file.
* The program must handle file conversion and extract codes from the *document.xml*
* The program must not alter these Codes in any way during the extraction.

**Test Case**

1. Two *.docx* files with known RSID codes are uploaded.
   1. Document with simple content, basic text and formatting.
   2. Document with complex formatting, tables, and images.
2. RSID detection is run.
3. **Verify that the detected RSID codes match the expected codes**
4. RSID extraction is run.
5. **Verify that the extracted RSID codes match the expected codes**

**Criteria for Success**

* All RSID codes in both documents are accurately detected and listed.
* No RSID codes are missing, and no extraneous codes are included.

**Test B – Percentage Calculation of Common RSID Codes**

This test focuses comparison functionality of *Documatch*. When provided a set of RSID codes we must find the common elements between the documents.

**Specification**

**Requirements**

* The program must have at least two sets of RSID codes.
* The program must display a percentage summary.
* The program must be repeatable, generating the same result.

**Test Case**

1. Two *.docx* files uploaded.
   1. Document A and B from the previous test, with common RSID codes. *or*
   2. Document C which is completely different and shares no common codes.
2. RSID detection & extraction is run. [>2 sets]
3. RSID codes are compared arithmetically.
4. **Percentage Similarity is determined.**
5. Process is repeated for the same set of files.
6. Process is repeated for a different set of files.

**Criteria for Success**

* Document A and Document B should show a percentage reflecting the commonality of RSID codes (e.g., 50%).
* Document A and Document C should show 0% common RSID codes.
* The percentage result must be the same for multiple runs of the test.

**Test C – Non-Numerical Representation of Similarity**

This test focuses on the output functionality of *Documatch*. We must confirm that the program represents the percentage of common RSID codes in a meaningful, non-numerical way, making it easier for the EU to interpret the results. This can be in the form of highlighted overlap, colour coding and charts.

**Specification**

**Requirements**

* The program must clearly distinguish matching and non-matching RSID codes using colour coding, charts, or other visual indicators which should reflect the numerical representation.

**Test Case**

1. Two *.docx* files uploaded.
   1. Documents A & B with partial overlap in RSID codes, representing moderate similarity.
   2. Document D with high similarity to Document A shares a majority of RSID codes.
2. RSID detection & extraction is run. [>2 sets]
3. RSID codes are compared arithmetically.
4. Results are interpreted and a graphical representation is generated.
5. **Review the generated visual summary for Document A and Document B**
6. Repeat the process by uploading Document A and Document D and review the visual summary.
7. Evaluate the clarity and effectiveness of the visual summary, noting how easily the EU can distinguish between high and low-similarity cases.

**Criteria for Success**

* For Document A and Document B, the visual element should indicate moderate similarity
* For Document A and Document D, the visual element should indicate high similarity
* The visual representation should be easily interpretable, providing EU with a clear understanding of the similarity level without needing to read a number.

**Test D – Location Identification of RSID Codes**

This test focuses on the interactive elements of RSID comparison in *Documatch*. We need to confirm that the program accurately identifies and highlights the locations of common RSID codes and maps them to physical elements within the document shown to the EU.

**Specification**

**Requirements**

* The program must have one complete set of RSID codes.
* The program must be able to locate elements of the document and assign them to definable objects.
* Visually responsive for the EU with clear visual cues.

**Test Case**

1. One *.docx* file is uploaded.
   1. Document with complete formatting, multiple sections.
2. RSID detection & extraction is run.
3. RSID codes are displayed on the page.
4. **Corresponding sections of the Document are displayed.**
5. Each section of the file is mapped and highlighted with each RSID code detected.
6. **Cross-reference the highlighted sections with the known locations of RSID codes in the original documents.**
7. Each code is validated to check if it matches.

**Criteria for Success**

* The location of each common RSID code within the document should be clearly and accurately highlighted.
* All identified matches should correspond precisely to the known RSID code locations in the input documents.

**Test E – Batch Processing of Documents**

This test focuses on *Documatch* and its capability to process multiple documents simultaneously and efficiently. The purpose is to ensure that the program can handle multiple files in one operation within an expected time limit and produce correct and consistent outputs.

**Specification**

**Requirements**

* The program must support the upload of multiple .docx files simultaneously.
* The program must perform RSID operations for all uploaded files in a single batch.
* **The program must process the batch within an acceptable time limit, ensuring efficient performance.**

**Test Case**

1. Multiple *.docx* files are uploaded to the value of *x*
2. Run RSID Detection & Extraction as a Batch Operation on *x* files
3. **Measure the time taken to run the Batch Operation with *x* files**
4. Compare RSID Codes Across Documents
5. **Generate Summary Report**
6. **Repeat the Process**

**Criteria for Success**

* The program should process the entire batch of documents within an acceptable time limit, with no significant delays or performance issues.
* It must display a timing process within acceptable bounds.
* The batch processing should produce the same results for repeated tests with the same documents, demonstrating reliability and consistency.
* No RSID codes are missing, and no extraneous codes are included.

**Test F – Platform Orientated Access and Use**

This test focuses on the accessibility and complete functionality of *Documatch*. As a web app, the Program should be fully functional across different devices, browsers and environments to be suitable for a wide range of end-user cases.

**Specification**

**Requirements**

* No matter which device, the same content is displayed.
* No matter which screen size, the same content is displayed.
* Upload, Comparison and Representation must be consistent across platforms.

**Test Case**

1. The program is accessed on a device.
   1. A Desktop using Google Chrome.
   2. A Desktop using Mozilla Firefox.
   3. A Phone using Apple Safari.
   4. A Tablet using Google Chrome.
2. Navigate through the application, upload two documents, and perform an RSID comparison.
3. **Review the results, ensuring all features (e.g, RSID detection, visualisation, download) work as expected.**
   1. Test the responsiveness of the user interface, ensuring that it adapts well to different screen sizes and resolutions.
   2. **Check for any browser-specific issues, such as layout problems or missing functionality.**
4. Repeat the process on a different device until all devices are tested.

**Criteria for Success**

* The program should be fully functional across all tested devices, with a consistent user experience across each.
* The program's user interface will be responsive, adapting without losing content.
* All generated results should be the same for the same documents tested.