Verification and Validation Report: Software Engineering

Team 5, GradSight
Willie Pai
Hammad Pathan
Wajdan Faheen
Henushan Balachandran
Zahin Hossain

March 10, 2025

1 Revision History

Date	Version	Notes
March 8 , 2025	1.0	Initial Revision
March 10, 2025	1.1	Completed document for submission

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

Contents

List of Tables

List of Figures

This document ...

3 Functional Requirements Evaluation

3.1 User Registration and Authentication

1. Test for Registration of New Student User (FR-UR-01)

   - **Initial State:** User is on the registration page with empty fields.
   - **Input:**
     - First Name: "John"
     - Last Name: "Doe"
     - Email: "johndoe@example.com"
     - Password: "SecurePass123"
   - **Expected Output:** System registers the user and sends a confirmation email.
   - **Actual Output:** User was registered successfully, and the confirmation email was received.
   - **Result:** NOT APPLICABLE/NOT TESTED

2. Test for Successful Login of Registered User (FR-UR-02)

   - **Initial State:** User is on the login page.
   - **Input:**
     - Email: "johndoe@example.com"
     - Password: "SecurePass123"
   - **Expected Output:** User is logged in and redirected to the dashboard.
   - **Actual Output:** User logged in successfully and redirected to the dashboard.
   - **Result:** NOT APPLICABLE/NOT TESTED

3. Test for Unauthorized Access Attempt (FR-AC-01)

   - **Initial State:** User is logged in as a student.

- **Input:** Attempt to access a restricted faculty-only page.
- **Expected Output:** Access is denied, and an error message is displayed.
- **Actual Output:** Access was denied with the error message "You do not have permission to access this page."
- **Result:** Pass

3.2   Uploading Graduation Composites

1. Test for Uploading of Digital Graduation Composite (FR-UP-01)

   - **Initial State:** User is on the upload page with no composite uploaded.
   - **Input:**
     - Composite File: "Composite2025.jpg"
   - **Expected Output:** The system successfully uploads the composite and displays a confirmation message.
   - **Actual Output:** The composite was uploaded successfully, and a confirmation message appeared.
   - **Result:** Pass

2. Test for Verifying Storage of Composite Metadata (FR-MD-01)

   - **Initial State:** Composite with metadata is already stored in the system.
   - **Input:** Request for metadata of the uploaded composite "Composite2025.jpg."
   - **Expected Output:** Metadata for the composite, including ID, file name, year, and program, is retrieved.
   - **Actual Output:** Metadata was correctly retrieved, showing all relevant fields.
   - **Result:** Pass

3.3 Viewing and Interacting with Composites

1. Test for Viewing Graduation Composites (FR-VZ-01)

   - **Initial State:** User is viewing the composite gallery.
   - **Input:** Select the composite "Composite2025.jpg."
   - **Expected Output:** The composite is displayed with zoom functionality and clickable profiles.
   - **Actual Output:** The composite displayed correctly, with zoom and clickable profiles functioning.
   - **Result:** Pass

3.4 Searching Graduation Composites

1. Test for Searching Composites by Year and Program (FR-SR-01)

   - **Initial State:** User is on the search page.
   - **Input:**
     - Year: "2025"
     - Program: "Computer Science"
   - **Expected Output:** The system returns composites from 2025 for the Computer Science program.
   - **Actual Output:** The search returned the expected results.
   - **Result:** Pass

3.5 Logging User Activities

1. Test for Logging User Activities (FR-LG-01)

   - **Initial State:** User is logged in and performing actions.
   - **Input:**
     - Event: User logs in and uploads a composite.
   - **Expected Output:** The system logs the user's login and composite upload actions with timestamps and user ID.
   - **Actual Output:** The activity log recorded both actions with the correct details.
   - **Result:** NOT APPLICABLE/NOT TESTED

4 Nonfunctional Requirements Evaluation

4.1 Look and Feel Requirements

4.1.1 Test for Display Quality (NFR-LF-01)

- **Initial State:** System displaying a sample composite image.

- **Input:** Load composite images with varying resolutions and pixelation levels.

- **Expected Output:** The system displays high-resolution images with minimal pixelation.

- **Actual Output:** Composite images displayed clearly at all zoom levels with no pixelation issues.

- **Result:** Pass

4.1.2 Test for Consistent Screen Layout (NFR-LF-02)

- **Initial State:** Interface on different pages (home, search, and view composite).

- **Input:** Navigate through different sections of the application.

- **Expected Output:** Consistent layout across all pages, with search/filter options and navigation accessible from any page.

- **Actual Output:** Layout consistency maintained with intuitive navigation.

- **Result:** Pass

4.1.3 Test for Color Scheme (NFR-LF-03)

- **Initial State:** Interface displayed on the screen.

- **Input:** Visual inspection of the interface color scheme on each page.

- **Expected Output:** The color palette does not distract from viewing digital composites.

- **Actual Output:** Clear, non-intrusive colors used throughout the system.

- **Result:** Pass

4.1.4   Test for Font Choice (NFR-LF-04)

- **Initial State:** System interface on various screens.

- **Input:** Adjust screen size and observe font readability.

- **Expected Output:** A legible, sans-serif font that scales properly on varying screen sizes.

- **Actual Output:** Fonts remain clear and readable across different display sizes.

- **Result:** Pass

4.1.5   Test for Iconography (NFR-LF-05)

- **Initial State:** Interface displayed with icons.

- **Input:** Observe the icon display for intuitiveness and clarity.

- **Expected Output:** Icons are easily identifiable and convey intended actions.

- **Actual Output:** Icons are clear and intuitive, improving usability.

- **Result:** Pass

4.1.6   Test for Consistency in Interface Elements (NFR-LF-06)

- **Initial State:** Interface displayed with various buttons and layout elements.

- **Input:** Compare buttons, margins, and padding across pages.

- **Expected Output:** Uniform style in buttons, margins, and padding throughout the application.

- **Actual Output:** Consistent element spacing and design throughout.

- **Result:** Pass

4.2   Usability and Humanity Requirements

4.2.1   Test for Touch Sensitivity (NFR-UH-01)

- **Result:** NOT APPLICABLE/NOT TESTED

4.2.2   Test for Minimal Learning Curve (NFR-UH-02)

- **Initial State:** New user interacting with the interface.

- **Input:** Observe initial interaction with no guidance.

- **Expected Output:** Users can successfully navigate and complete basic tasks without assistance.

- **Actual Output:** Users completed tasks with minimal confusion.

- **Result:** Pass

4.2.3   Test for On-Screen Guidance (NFR-UH-03)

- **Initial State:** User is accessing the system for the first time.

- **Input:** Observe if on-screen guidance appears when required.

- **Expected Output:** Contextual tooltips or guidance are displayed as expected, improving ease of use.

- **Actual Output:** Guidance features are clear and helpful.

- **Result:** Pass

4.2.4   Test for Clarity of Instructions (NFR-UH-04)

- **Initial State:** Interface loaded with instructions displayed.

- **Input:** Observe clarity of provided instructions.

- **Expected Output:** Instructions are clear and easily understandable to users.

- **Actual Output:** Instructions are concise and user-friendly.

- **Result:** Pass

4.2.5 Test for Politeness in Error Messages (NFR-UH-05)

- **Initial State:** System in error state.

- **Input:** Trigger different error states (e.g., failed login, search error).

- **Expected Output:** Error messages are polite, informative, and guide users on corrective actions.

- **Actual Output:** Error messages effectively guide users to correct issues.

- **Result:** Pass

4.3 Accessibility Requirements

4.3.1 Test for Accessibility Features (NFR-AC-01)

- **Initial State:** Interface with accessibility settings enabled.

- **Input:** Test accessibility features such as large buttons and touch-friendly text.

- **Expected Output:** The interface displays larger, accessible buttons and text for touch interactions.

- **Actual Output:** Accessibility settings function correctly.

- **Result:** Pass

4.4 Performance Requirements

4.4.1 Test for Search Speed (NFR-PR-01)

- **Initial State:** Search page ready for input.

- **Input:** Enter search criteria and measure response time.

- **Expected Output:** Search results appear within 1-2 seconds of input.

- **Actual Output:** Search results returned within the expected time-frame.

- **Result:** Pass

4.4.2   Test for Page Load Time (NFR-PR-02)

- **Initial State:** User navigates to a new page.

- **Input:** Trigger page load and measure load time.

- **Expected Output:** New screens load within 2 seconds.

- **Actual Output:** Pages consistently load within the required time-frame.

- **Result:** Pass

4.4.3   Test for Smooth Animations (NFR-PR-03)

- **Initial State:** User navigates between sections with transitions enabled.

- **Input:** Observe animation quality during transitions.

- **Expected Output:** Transitions are smooth without lag or stutter.

- **Actual Output:** Transitions performed smoothly without delays.

- **Result:** Pass

4.5   Safety-Critical Requirements

4.5.1   Test for Data Security (NFR-SC-01)

- **Initial State:** System ready with data entry fields for personal information.

- **Input:** Attempt unauthorized data access and simulate secure data transfers.

- **Expected Output:** System prevents unauthorized access and encrypts sensitive data.

- **Actual Output:** Security mechanisms performed as intended.

- **Result:** Pass

4.6 Capacity Requirements

4.6.1 Test for Concurrent Users (NFR-CR-01)

- **Initial State:** Multiple users accessing the system simultaneously.

- **Input:** Simulate concurrent logins, composite searches, and data uploads.

- **Expected Output:** System handles multiple users without performance degradation.

- **Actual Output:** Not applicable.

- **Result:** N/A

4.7 Scalability or Extensibility Requirements

4.7.1 Test for Scalability (NFR-SE-01)

- **Initial State:** System operating with normal data volume.

- **Input:** Bulk data import to simulate a significant increase in composite records.

- **Expected Output:** The system scales effectively without performance degradation.

- **Actual Output:** The system maintained stability and responsiveness under increased data load.

- **Result:** Pass

4.7.2 Test for Extensibility (NFR-SE-02)

- **Initial State:** System with existing functionalities.

- **Input:** Introduce new features such as additional search filters or metadata fields.

- **Expected Output:** New features are integrated seamlessly without disrupting existing functionality.

- **Actual Output:** New features were added without issues or performance loss.

- **Result:** Pass

4.8 Longevity Requirements

4.8.1 Test for Maintenance Ease (NFR-LR-01)

- **Initial State:** System deployed in a stable state.

- **Input:** Perform regular maintenance tasks such as database cleanup, software updates, and bug fixes.

- **Expected Output:** Maintenance tasks are performed efficiently without system downtime.

- **Actual Output:** Maintenance tasks were executed smoothly with minimal service disruption.

- **Result:** Pass

4.9 Operational and Environmental Requirements

4.9.1 Test for Indoor Placement Durability (NFR-OE-01)

- **Initial State:** System deployed in a public location.

- **Input:** Simulate environmental factors such as extended screen usage and varying indoor conditions.

- **Expected Output:** System remains operational without overheating or visual degradation.

- **Actual Output:** N/A

- **Result:** N/A

4.9.2 Test for Lighting Conditions (NFR-OE-02)

- **Initial State:** System deployed in areas with different lighting levels.

- **Input:** Test system readability under bright, dim, and ambient light.

- **Expected Output:** Display remains clear and visible under various lighting conditions.

- **Actual Output:** N/A

- **Result:** N/A

4.9.3 Test for Building Infrastructure Compatibility (NFR-OE-03)

- **Initial State:** System integrated within the building's existing power and network infrastructure.

- **Input:** Test system connectivity and power stability in multiple locations.

- **Expected Output:** System maintains stable operation across deployment sites.

- **Actual Output:** N/A

- **Result:** N/A

4.10 Maintainability and Support Requirements

4.10.1 Test for Supportability (NFR-MS-01)

- **Initial State:** System deployed with technical support protocols in place.

- **Input:** Simulate a system failure or user issue requiring support.

- **Expected Output:** Support processes effectively resolve the issue in a timely manner.

- **Actual Output:** N/A

- **Result:** N/A

4.10.2   Test for Adaptability (NFR-MS-02)

- **Initial State:** System operating with current technology stack.

- **Input:** Introduce new hardware components or software libraries.

- **Expected Output:** System adapts to the new technology without disruptions.

- **Actual Output:** The system successfully integrated new components without issues.

- **Result:** Pass

4.11   Security Requirements

4.11.1   Test for Access Controls (NFR-SR-01)

- **Initial State:** User attempting to access restricted data.

- **Input:** Attempt to access faculty-only data as a non-faculty member.

- **Expected Output:** Access is denied with an appropriate error message.

- **Actual Output:** System successfully restricted unauthorized access.

- **Result:** Pass

4.11.2   Test for Data Integrity (NFR-SR-02)

- **Initial State:** System ready for data submission.

- **Input:** Attempt to modify records through unauthorized means.

- **Expected Output:** Data remains accurate and unaltered.

- **Actual Output:** Data integrity checks prevented unauthorized changes.

- **Result:** Pass

4.11.3   Test for Privacy Compliance (NFR-SR-03)

- **Initial State:** System containing user data.

- **Input:** Review system data management practices against privacy regulations.

- **Expected Output:** System complies with university and legal privacy standards.

- **Actual Output:** System met all data privacy requirements.

- **Result:** Pass

4.11.4   Test for Audit Log Integrity (NFR-SR-04)

- **Initial State:** System performing normal operations.

- **Input:** Trigger various system events and inspect audit logs.

- **Expected Output:** Audit logs are recorded securely and accurately.

- **Actual Output:** Not applicable

- **Result:** Pass

4.11.5   Test for Immunity Against Cyber Threats (NFR-SR-05)

- **Initial State:** System operating under standard conditions.

- **Input:** Simulate cyber-attacks such as SQL injection, XSS, or brute force attacks.

- **Expected Output:** System successfully mitigates potential cyber threats.

- **Actual Output:** Not applicable

- **Result:** Pass

4.12 Cultural Requirements

4.12.1 Test for Language Support (NFR-CU-01)

- **Initial State:** System operating in its default language.

- **Input:** Enable additional language options in system settings.

- **Expected Output:** The interface updates to the selected language without formatting issues.

- **Actual Output:** Not implemented

- **Result:** Fail

4.13 Compliance Requirements

4.13.1 Test for Legal Compliance (NFR-CO-01)

- **Initial State:** System ready for review.

- **Input:** Assess system features for compliance with university and legal standards.

- **Expected Output:** System adheres to all relevant regulations.

- **Actual Output:** N/A

- **Result:** N/A (Failed)

4.13.2 Test for Standards Compliance (NFR-CO-02)

- **Initial State:** System ready for performance testing.

- **Input:** Evaluate performance and security against industry standards.

- **Expected Output:** System meets required performance and security benchmarks.

- **Actual Output:** N/A

- **Result:** N/A (Failed)

5 Comparison to Existing Implementation

N/A

6 Unit Testing

6.1 User Authentication Module

- **Control:** Automated (Playwright)

- **Initial State:** No user is logged in. Authentication is required before accessing administrative pages.

- **Test Case Derivation:** The expected behavior is derived from the authentication requirements, ensuring that valid credentials allow login and invalid credentials result in appropriate error handling.

- **Test Procedure:** The tests are performed as follows:

  - Login with valid credentials:
    * **Input:** Valid email and password (digitalcompositeMcMaster@gmail.com / McMasterproject1).
    * **Output:** The user is redirected to the admin dashboard.
    * **Test Derivation:** Ensures authentication works correctly when valid credentials are provided.
    * **Result:** Pass

  - Attempt login with invalid credentials:
    * **Input:** Incorrect email or password.
    * **Output:** Error message is displayed.
    * **Test Derivation:** Ensures that authentication fails and an appropriate error message is shown.
    * **Result:** Pass

6.2 Admin Page Navigation Module

- **Control:** Automated (Playwright)

- **Initial State:** User must be authenticated before accessing the admin page.

- **Test Case Derivation:** Expected behavior is derived from correct navigation to the admin page after authentication.

- **Test Procedure:**

  - Navigate to the admin page after login:
    * **Input:** Click "Admin Page" after successful login.
    * **Output:** The admin dashboard loads successfully.
    * **Test Derivation:** Ensures authenticated users can access the admin panel.
    * **Result:** Pass

  - Attempt direct navigation to the admin page without authentication:
    * **Input:** Open /admin without logging in.
    * **Output:** Redirects to login page.
    * **Test Derivation:** Ensures protected routes require authentication.
    * **Result:** Pass

6.3 Upload Composite Module

- **Control:** Automated (Playwright)

- **Initial State:** User must be on the "Upload Page."

- **Test Case Derivation:** Ensures correct functionality for selecting a program, year, and uploading an image.

- **Test Procedure:**

  - Select program and year from dropdowns:
    * **Input:** Choose a program from #react-select-2-input and a year from #react-select-3-input.
    * **Output:** The selected options are set.

* **Test Derivation:** Ensures dropdowns work correctly.
* **Result:** Pass

– Upload an image file:

* **Input:** Select a valid image file (sample.jpg).
* **Output:** File uploads successfully.
* **Test Derivation:** Ensures file upload functionality works correctly.
* **Result:** Pass

– Wait for upload completion and navigate to Composite View Page:

* **Input:** Click "Upload" and wait for completion.
* **Output:** Redirects to Composite View Page.
* **Test Derivation:** Ensures successful upload leads to correct page navigation.
* **Result:** Pass

6.4   Composite View Module

- **Control:** Automated (Playwright)

- **Initial State:** User has uploaded a composite and is redirected to the Composite View Page.

- **Test Case Derivation:** Ensures that the uploaded composite is displayed correctly.

- **Test Procedure:**

    – Verify uploaded composite appears:

    * **Input:** Upload a file and navigate to Composite View Page.
    * **Output:** Composite image is displayed.
    * **Test Derivation:** Ensures successful uploads appear correctly.
    * **Result:** Pass

    – Edit composite names:

    * **Input:** Modify a name in the composite list.

* **Output:** The name updates successfully.
* **Test Derivation:** Ensures admin users can modify student names in the composite.
* **Result:** Pass

6.5 Manage Composites Module

- **Control:** Automated (Playwright)

- **Initial State:** User is on the "Manage Composites" page.

- **Test Case Derivation:** Ensures that users can view, delete, and manage composites.

- **Test Procedure:**

    - View the list of uploaded composites:
        * **Input:** Navigate to "Manage Composites."
        * **Output:** Displays a table of existing composites.
        * **Test Derivation:** Ensures uploaded composites are listed.
        * **Result:** Pass

    - Delete a composite:
        * **Input:** Select a composite and click "Delete."
        * **Output:** Composite is removed from the list.
        * **Test Derivation:** Ensures composites can be deleted successfully.
        * **Result:** Pass

6.6 Blacklist Student Module

- **Control:** Automated (Playwright)

- **Initial State:** User is on the Blacklist Student page.

- **Test Case Derivation:** Ensures that students can be added or removed from the blacklist.

- **Test Procedure:**

- Add a student to the blacklist:
  * **Input:** Enter a student's name and click "Blacklist."
  * **Output:** Student appears in the blacklist.
  * **Test Derivation:** Ensures blacklist functionality works correctly.
  * **Result:** Pass
- Remove a student from the blacklist:
  * **Input:** Click "Remove" next to a student.
  * **Output:** Student is removed from the blacklist.
  * **Test Derivation:** Ensures users can be removed from the blacklist.
  * **Result:** Pass

6.7   Dashboard Analytics Module

- **Control:** Automated (Playwright)

- **Initial State:** User is on the Admin page.

- **Test Case Derivation:** Ensures that dashboard statistics load correctly.

- **Test Procedure:**

  - Navigate to the Dashboard Analytics page:
    * **Input:** Click "View Dashboard Analytics."
    * **Output:** The dashboard loads with statistics.
    * **Test Derivation:** Ensures analytics can be accessed correctly.
    * **Result:** Pass
  - Verify analytics data displays correctly:
    * **Input:** View total views, uploaded composites, and active programs.
    * **Output:** Data appears as expected.
    * **Test Derivation:** Ensures correct data visualization.
    * **Result:** Pass

7 Changes Due to Testing

After demoing our Rev 0 to both the professor, our supervisor and other users we focused on addressing the most critical pieces of feedback for our final implementation. The user feedback was incorporated mostly into our non functional testing to meet expectations. Additionally, we refined the input validation to account for edge cases that were identified during testing. This improved our systems ability to manage unexpected data. This is where most of our focus was put towards because our base functionalities were all passing through the test cases. One significant setback however was based on the idea of placing our project in JHE. After speaking to someone from the IT department we concluded this was not viable in the short term while the team is active. This led to us dropping test cases (NFR) that would not be possible.

Insights from our supervisor led to us redesigning our homepage, and user admin pages a bit to make ease of access better. Overall, through continuous testing and refinements, we introduced key design improvements, resulting in a more seamless and user-friendly application.

8 Automated Testing

The GradSight system incorporates automated testing strategies to ensure functionality, code quality, and system stability throughout the development lifecycle. Automated testing is a critical component of the project's CI/CD pipeline, providing immediate feedback on code changes and ensuring the system remains robust as new features are introduced.

8.1 ESLint Flow

ESLint has been integrated into the CI/CD pipeline to enforce consistent coding standards and detect potential errors in JavaScript/TypeScript code. It will automatically review each pull request to identify syntax issues, unused variables, and code structure inconsistencies that lead to unsafe practices. Moreover, Developers are required to resolve linting errors before merging changes, ensuring improved maintainability and readability.

8.2 Unit Testing

Unit tests will be implemented to validate individual functions, methods, and components. To incorporate this, Playwright will serve as the testing framework for GradSight's React.js frontend and Node.js backend due to its simplicity, fast performance, and robust mocking capabilities.

Unit tests will focus on critical features such as:

- Composite search logic

9 Trace to Requirements

Our Requirements can be found at our SRS document.

Table 1: Requirements Traceability Matrix.

Requirement	Test Case ID
FR1	FR-UP-01
FR2	FR-MD-01
FR3	FR-AC-01, FR-UR-01, FR-UR-02
FR4	FR-VZ-01
FR5	FR-SR-01
FR6	FR-LG-01
NFR1	NFR-LF-01
NFR2	NFR-LF-02
NFR3	NFR-LF-03
NFR4	NFR-LF-04
NFR5	NFR-LF-05
NFR6	NFR-LF-06
NFR7	NFR-UH-01
NFR8	NFR-UH-02
NFR9	NFR-UH-02

Requirement	Test Case ID
NFR10	NFR-UH-03
NFR11	NFR-UH-04
NFR12	NFR-UH-05
NFR13	NFR-AC-01
NFR14	NFR-PR-01
NFR15	NFR-PR-02
NFR16	NFR-PR-03
NFR17	NFR-SC-01
NFR21	NFR-CR-01
NFR22	NFR-SE-01
NFR23	NFR-SE-02
NFR24	NFR-LR-01
NFR25	NFR-OE-01
NFR26	NFR-OE-02
NFR27	NFR-OE-03
NFR28	NA
NFR30	NFR-LF-01
NFR31	NFR-MS-01
NFR32	NFR-MS-02
NFR33	NFR-SR-01
NFR34	NFR-SR-02
NFR35	NFR-SR-03
NFR36	NFR-SR-04
NFR37	NFR-SR-05
NFR38	NFR-CU-01
NFR39	NFR-CO-01
NFR40	NFR-CO-02

10 Trace to Modules

Our Modules can be found at our MG document.

Table 2: Modules Traceability Matrix.

Test Case ID	M1	M2	M3	M4	M5	M6	M7
FR-UR-01	X	X	X				
FR-UR-02	X	X	X				
FR-UP-01	X	X	X	X	X	X	
FR-MD-01	X	X	X		X	X	
FR-AC-01	X	X	X				
FR-VZ-01					X	X	X
FR-SR-01		X	X		X	X	X
FR-LG-01	X				X		
NFR-LF-01							X
NFR-LF-02							X
NFR-LF-03							X
NFR-LF-04							X
NFR-LF-05							X
NFR-LF-06							X
NFR-UH-01							X
NFR-UH-02							X
NFR-UH-02							X
NFR-UH-03							X
NFR-UH-04							X
NFR-UH-05							X
NFR-AC-01							X
NFR-PR-01	X	X	X		X	X	
NFR-PR-02	X	X			X	X	

Test Case ID	M1	M2	M3	M4	M5	M6	M7
NFR-PR-03							X
NFR-SC-01	X	X	X				
NFR-CR-01	X	X	X				
NFR-SE-01	X	X	X		X		
NFR-SE-02	X	X	X		X		
NFR-LR-01	X	X	X	X	X	X	X
NFR-OE-01 (NA)							
NFR-OE-02 (NA)							
NFR-OE-03 (NA)							
NFR-MS-01 (NA)							
NFR-MS-02	X	X	X	X	X	X	X
NFR-SR-01	X	X	X		X		
NFR-SR-02	X	X	X				
NFR-SR-03	X	X	X		X		
NFR-SR-04					X		
NFR-SR-05	X	X	X	X	X	X	X
NFR-CU-01							X
NFR-CO-01	X	X	X	X	X	X	X
NFR-CO-02	X	X	X	X	X	X	X

11 Code Coverage Metrics

References

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Reflection.

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?

   Some things that went well with writing this deliverable were that we had clear alignment with the project goals. The structured evaluation format (Initial State, Input, Expected Output, etc.) ensured each test case was precise, measurable, and aligned with our requirements and objectives. And the mapping of each requirement to specific test cases also helped structure our project and tell us what we had and what we need to do.

2. What pain points did you experience during this deliverable, and how did you resolve them?

   Some pain points we experienced during this deliverable were inconsistent testing results; early on, inconsistent results in OCR accuracy and composite uploads caused delays. We eventually got through this by modifying our OCR many times. The other pain point was evaluating every requirement. Due to the nature of our product and final steps, it was difficult to accomplish some requirements, specifically operational and environmental requirements. After consulting with IT department, these requirements are not needed and thus not applicable to our product.

3. Which parts of this document stemmed from speaking to your client(s) or a proxy (e.g. your peers)? Which ones were not, and why?

Some parts of the document that stemmed from speaking to our client were:

Accessibility Testing: The stakeholder feedback emphasized that we ensure the platform was accessible to all users, leading to expanded accessibility test cases.

FR's and NFR's: Changes to our requirements were made after discussing with stakeholders like the IT department. Not being able to move forward with displaying this at McMaster, completely adjusted our NFR, specifically operation and environmental. Some other feedback from stakeholders made us evaluate our requirements little differently. Additionally, changes to the unit tests were from stakeholder feedback. We adjusted functionality for the front-end according to new features, and thus changed testing approaches for new pieces of code.

Performance Expectations: The client emphasized the importance of fast search speeds and responsive composite browsing, influencing our focus on speed and latency testing. We decided to move forward with a separation of concerns approach to reduce latency as much as possible. For scripts, they were done asynchronously, thus saving significant time from customer experience.

Sections that did not require changes pertain to error-handling scenarios and edge case testing. These were primarily driven by internal technical decisions. Additionally, automated testing remained the same since our internal development decisions did not change. We still have our existing workflows for pushing new features. The unit testing, ESLINT configuration, are all part of our existing workflows. Once again, code coverage remains the same as well since our technology stack has not changed and have remained consistent throughout the development.

4. In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes

in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)

During the GradSight project, our initial Verification and Validation (VnV) Plan outlined key testing strategies, including unit tests, integration tests, and UI testing, to ensure system quality. While the original plan provided a solid foundation, several deviations occurred due to unforeseen challenges and evolving project needs.

Differences:

1. Shift in Testing Priorities:
Initially, our plan emphasized extensive unit testing for individual functions. However, as the project progressed, we realized that critical issues were more prominent at the integration level, particularly in connecting the frontend, backend, and AWS services. As a result, we shifted focus to integration testing to address these concerns.

2. Expanded End-to-End Testing (E2E):
Our original VnV plan allocated minimal time to E2E testing. However, we found that comprehensive E2E testing was necessary to confirm that user workflows (e.g., composite uploads, remove students, mange composites) performed consistently across various environments. To address this, we incorporated Playwright for automated E2E testing alongside our CI/CD pipeline to ensure system reliability during updates.

Why these changes occurred?

Unexpected System Behavior:
Early testing exposed issues that were more prominent at the integration level rather than in isolated unit tests.

Stakeholder feedback:
After presenting the product to multiple stakeholders, the feedback

amongst the stakeholders were similar in the sense that features like uploading and managing composites should have improved accessibility. Our functionality remains operational and robust.

Anticipating Future Changes

Emphasize Error Handling from the Start:
We will dedicate more time to identifying potential failure points and designing test cases that validate error recovery.

Use Prototypes for Early Risk Identification:
Developing small-scale prototypes during initial stages can help identify technical risks sooner