Testing Guide for IDAK Application

Testing is an integral part of maintaining the integrity and reliability of the IDAK application. This guide provides an overview of the testing procedures that should be followed to ensure the application functions correctly and is free of defects.

Overview of Testing Types

The IDAK application should undergo several types of testing:

1. **Unit Testing**: Tests individual units of code for functionality.
2. **Integration Testing**: Ensures that different parts of the application work together as expected.
3. **System Testing**: Verifies that the entire application meets the specified requirements.
4. **Regression Testing**: Confirms that recent code changes have not adversely affected existing features.

Unit Testing

Unit tests are written to validate that each function performs as intended in isolation.

1. **Writing Unit Tests**:
   - Unit tests should be written for every new function and method.
   - Each test should be independent and repeatable.
   - Mock external dependencies using modules like `unittest.mock` to simulate databases, file systems, and network services.
2. **Running Unit Tests**:

Run unit tests using the following command:
```
python -m unittest unit_tests
```

3. **Interpreting Test Results**:
   - Analyze the output of the test run for any failed tests.
   - Investigate and fix any failed tests before proceeding with further code changes.

Integration and System Testing

Integration and system testing require you to test the application as a whole.

1. **Test Environment**:
   - Set up a test environment that mirrors the production environment as closely as possible.
2. **Test Cases**:
   - Create test cases that cover all the application's features, including:
     - Reading data from the Excel file.
     - Converting RTF to HTML.
     - Attaching files to emails.

- Sending emails via SMTP.
3. **Automated Testing**:
   - Consider setting up automated testing with continuous integration tools to run tests for each deployment.

Regression Testing

Perform regression testing to ensure new code contributions have not introduced any new issues.

1. **Test Plans**:
   - Maintain a test plan that includes a checklist of all critical paths that need retesting after changes.
2. **Automated Regression Tests**:
   - Automated tests should be rerun after each code change to ensure no new issues have been introduced.

Security Testing

Conduct security testing to validate the security features of the application.

1. **Vulnerability Scanning**:
   - Use automated tools to scan the application for known vulnerabilities, particularly in third-party libraries.
2. **Penetration Testing**:
   - Perform penetration testing to simulate attacks on the application and identify potential security weaknesses.

Performance Testing

Ensure the application performs well under expected loads.

1. **Load Testing**:
   - Simulate the maximum number of expected users to test how the application behaves under load.
2. **Stress Testing**:
   - Go beyond the expected load to determine the application's breaking point and how it fails.

Documentation of Tests

Keep thorough documentation for all tests, including:

1. **Test Cases**:
   - Document the purpose, inputs, execution steps, and expected results for each test case.
2. **Test Results**:
   - Record the outcomes of test cases, including any discrepancies from expected results.

3. **Test Coverage**:
     ○ Maintain a coverage report to ensure that all parts of the application are tested.

Reporting Issues

If issues are found during testing:

1. **Bug Reporting**:
     ○ Report bugs or issues in a detailed and reproducible manner.
2. **Tracking**:
     ○ Use a tracking system to monitor the status of reported bugs and fixes.

Best Practices

- Prioritize testing based on the risk and impact of the application features.
- Regularly update test cases to reflect changes in the application.
- Encourage developers to write tests for new features or bug fixes.

Adhering to this testing guide will help ensure the quality and stability of the IDAK application. If you have any questions or require assistance with testing procedures, please contact the quality assurance team or designated testing personnel.