Software Test Plan (STP) & Software Test Report (STR)

Course: CS 4850 – Section 04

Semester: Fall 2025

Professor: Sharon Perry

Date: November 30, 2025

Project Title: Automation of Digital Intake with MS 365 – Carelink of Georgia

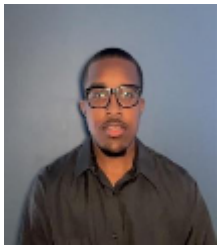
			
Aashna Suthar (Team Leader) Documentation	Franck Dipanda Documentation + Developer	Angel Gutierrez Developer	Ayush Kachhadiya Documentation + Tester

Table of Contents

1. Overview *4*

1.1 Purpose **4**

1.2 Scope **4**

2. Testing Summary *4*

2.1 Scope of Testing **4**

2.1.1 In Scope 4

2.1.2 Out of Scope 5

3. Analysis of Scope and Test Focus Areas *5*

3.1 Release Content **5**

3.2 Regression Testing **6**

3.3 Platform Testing **6**

4. Progression Test Objectives *6*

5. Regression Test Objectives *7*

6. Other Testing *7*

6.1 Security **7**

6.2 Stress & Volume Testing **7**

6.3 Connectivity Testing **8**

6.4 Disaster Recovery / Backup **8**

6.5 Unit Testing **8**

6.6 Integration Testing **8**

7. Test Strategy *9*

7.1 Test Level Responsibility **9**

7.2 Test Type & Approach **9**

7.3 Build Strategy **9**

7.4 Test Execution Schedule **9**

7.5 Facility, Data & Resource Plan **10**

7.6 Testing Tools **10**

7.7 Testing Handover Procedure **10**

7.8 Testing Metrics **10**

8. Test Environment Plan *10*

8.1 Test Environment Diagram **10**

8.2 Test Environment Details **11**

8.2.1 Testers 11

8.2.2 Hardware 12

8.2.3 Software 12

8.2.4 Interfaces 12

8.2.5 Other Materials 12

8.3 Establishing Environment **12**

8.4 Environment Control **13**

8.5 Environment Roles **13**

9. Assumptions and Dependencies *13*

9.1 Assumptions **13**

9.2 Dependencies **13**

10. Entry and Exit Criteria *14*

10.1 Entry Criteria **14**

10.2 Exit Criteria **14**

11. Administrative Plan *14*

11.1 Approvals **14**

11.2 Test Milestones **14**

11.3 Training **15**

11.4 Defect Management **15**

12. Definitions *15*

13. References *15*

14. Points of Contact *16*

15. Software Test Report (STR) *16*

1. Overview

1.1 Purpose

The purpose of this document is to explain how we tested our digital intake system for CareLink of Georgia and to record the results of that testing. This document includes both the:

- Software Test Plan (STP) - what we planned to test and how
- Software Test Report (STR) - what happened during the testing

The main goals of this testing were to:

- Make sure the digital intake form works correctly
- Confirm forms are stored properly in SharePoint
- Verify Adobe signatures work as expected
- Ensure staff receive notifications
- Check that the workflow is reliable and easy to use

Testing helped us to make sure that the system was reliable, secure, and ready to be used by the CareLink staff.

1.2 Scope

This test plan covers testing done by our project team for the CareLink Digital Intake Automation system. It explains:

- What parts of the system were tested – features, tools, and flows
- How testing was done – step-by-step tests, data and environment
- What tools and data were used – devices, accounts, and applications
- What results were observed – pass/fail results and final status

All testing was done using sample data only. No real patient information was used at any time.

2. Testing Summary

2.1 Scope of Testing

2.1.1 In Scope

The following features and components were tested as part of this project:

- Microsoft Form Selection Tool
    - Staff-facing form used to choose which intake form to send
- Digital intake form completion
    - Patient-facing form sent for signing through Adobe Sign
- Adobe Sign electronic signatures
    - Verifying that the patient can sign and submit the agreement
- Power Automate workflow execution
    - Triggering on form submission and running until completion
- SharePoint form storage
    - Making sure signed forms are saved in the correct document library
- Staff email notifications through Outlook
    - Email sent to the staff email that was entered in the selection tool

These features make up the full digital intake workflow, so they were all tested together and individually.

2.1.2 Out of Scope

The following items were not tested:

- Integration with the clinic's internal platform Methasoft
    - We were not given direct access due to HIPAA and employment restrictions
- Real patient data testing
    - Only test names and sample data was used
- Billing, insurance, or clinical charting
- Mobile app testing

These items were outside the project scope or restricted due to security and compliance rules.

3. Analysis of Scope and Test Focus Areas

3.1 Release Content

The system features tested in this phase included:

- Form submission from Microsoft Forms
- Adobe Sign agreement creation and signing
- Automatic file return to SharePoint
- Metadata storage (patient name, staff name, date)
- Email notification to staff after completion

Each of these items was verified through multiple test runs.

3.2 Regression Testing

Basic regression testing was completed each time the workflow was updated to make sure nothing broke from earlier versions. The following items were checked after each change:

- Forms still submitted correctly
- Storage in SharePoint still worked
- Email notifications were still triggered

This helped ensure that new updates did not affect existing features.

3.3 Platform Testing

Testing was completed using:

- Windows laptops
- Web browsers (Edge, Chrome)
- Outlook online

All testing was done using the school's Microsoft 365 environment.

4. Progression Test Objectives

These are the main features we wanted to verify at least once from end-to-end.

Reference	Function	Test Objective	Evaluation Criteria	Priority
T1	Form Submission	Verify patient can submit form	Form submits successfully	High
T2	Adobe Signing	Verify patient can sign digitally	Signature captured	High
T3	Automation Flow	Verify workflow runs after submission	Flow completes without error	High
T4	SharePoint Storage	Verify signed form is saved	Form appears in library	High

T5	Email Notification	Verify staff receives email	Email received	Medium

Each of these objectives represents a critical step in the overall intake process.

5. Regression Test Objectives

These tests were repeated after changes to make sure nothing broke.

Reference	Function	Objective	Evaluation Criteria	Priority
R1	Form Reuse	Ensure previous forms still submit correctly	No submission errors	Medium
R2	Storage Access	Ensure permissions still apply	Unauthorized access blocked	High

Examples of regression checks:

- Resubmitting the Form Selection Tool with different patients
- Confirming existing signed files in SharePoint were still accessible
- Logging in with a different user account to make sure permissions were still enforced

6. Other Testing

6.1 Security

Security testing confirmed that only authorized staff could view submitted forms. This was verified by:

- Logging in with different accounts
- Testing restricted access levels
- Confirming that unauthorized users could not access files

We did not perform penetration testing. Our goal was to confirm that SharePoint permissions and MS 365 authentication were working as expected for this prototype.

6.2 Stress & Volume Testing

Formal stress testing was not performed due to the limited prototype environment. However:

- Multiple form submissions were completed in short time periods
- The workflow successfully handled repeated use

- No system crashes or failures were observed
- Formal stress testing was not performed due to the limited prototype environment.

6.3 Connectivity Testing

Testing was performed on different networks (home Wi-Fi and campus Wi-Fi) to confirm that the system works under normal connectivity conditions.

If the internet connection is lost during signing, Adobe will show an error message. In this case, the form can be reopened at a later time via the email link.

6.4 Disaster Recovery / Backup

We did not build a separate disaster recovery plan. Instead, we are relying on built-in MS 365 features:

- SharePoint version history
- Recycle Bin for deleted files
- MS' own cloud backup and retention

We did not simulate a full outage or full restore, but we confirmed:

- Older file versions could be restored
- Deleted test files could be recovered from the Recycle Bin

6.5 Unit Testing

Each part of the system was tested separately before full workflow testing, including:

- Microsoft Form submission
- Adobe Sign agreement creation
- SharePoint upload actions

This helped us catch small issues early on, such as missing required fields or incorrect library names.

6.6 Integration Testing

Full end-to-end testing was performed to confirm that:

- Microsoft Forms
- Adobe Sign
- Power Automate

- Outlook
- SharePoint

all work together as one complete system.

7. Test Strategy

7.1 Test Level Responsibility

All testing was performed by the project team. No external testers were used for this phase.

7.2 Test Type & Approach

Testing was primarily done manually. Each test case was executed step-by-step, and results were recorded as pass or fail. If a test failed, the project team would check the flow in Microsoft Power Automate to see which specific action caused the failure. The developers would then correct the bug and move on to other test cases.

7.3 Build Strategy

The system was built in stages:

1. First, the digital form was created
2. Then Adobe signing was added
3. After that, storage in SharePoint was configured
4. Finally, full workflow automation was tested

This layered approach made it easier to debug when something went wrong.

7.4 Test Execution Schedule

Testing was performed throughout Milestone 2 during the development phase and after each major workflow update.

- Early testing:
    - From behavior and basic submission
- Mid-phase testing:
    - Adobe Sign integration and signing
- Late-phase testing:
    - Full workflow, notifications, and regression tests

We also re-ran key tests right before writing this STP/STR to confirm everything was still working.

7.5 Facility, Data & Resource Plan

- Testing used school laptops and personal devices
- Sample data was created for testing purposes
- No real patient data was used

7.6 Testing Tools

- Microsoft Forms
- Adobe Acrobat Sign
- Power Automate
- SharePoint
- Outlook
- Web browsers

7.7 Testing Handover Procedure

Once testing is complete, the final workflow will be handed over to CareLink staff along with training documentation.

7.8 Testing Metrics

We tracked simple, practical metrics, including:

- Number of successful submissions
- Number of failed submissions
- Workflow execution success rate

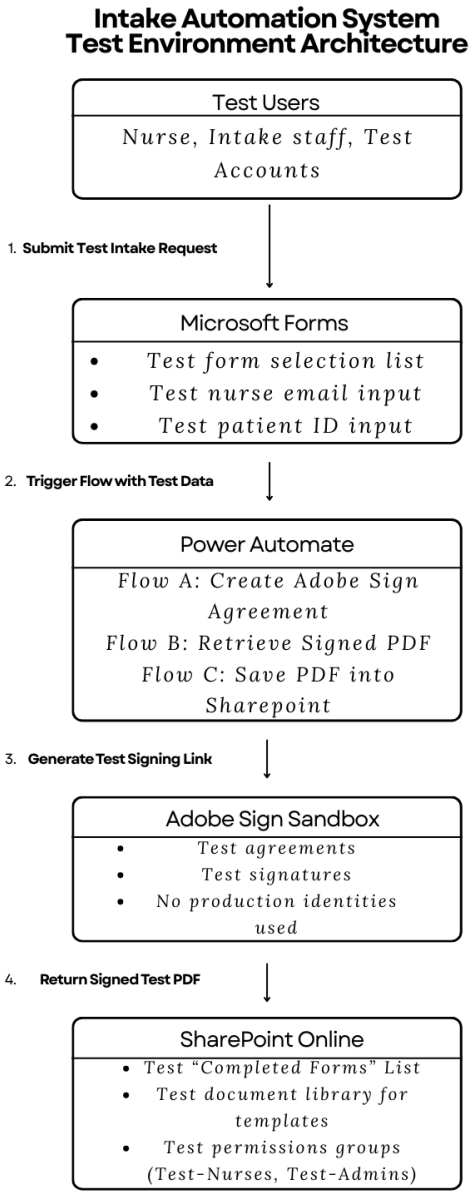
These metrics helped us decide whether the prototype was stable enough to hand off.

8. Test Environment Plan

This section explains the setup that was used to test the digital intake system. All testing was done in a controlled school environment using Microsoft 365 tools. No real clinic systems or patient data were used.

8.1 Test Environment Diagram

A test environment diagram (shown below) is included in the report showing how Microsoft Forms connects to Power Automate, which then connects to Adobe Sign, SharePoint, and Outlook. This diagram represents the full testing environment and data flow for the project.



8.2 Test Environment Details

8.2.1 Testers

Testing was completed by the project team:

- Project team members
- No external QA testers were involved
- Each tester ran multiple test cases using sample data

8.2.2 Hardware

The following hardware was used during the testing:

- Personal Laptops
- Personal iPad for additional testing
- Personal smartphones
- Standard home Wi-Fi and campus Wi-Fi networks

A variety of device and connection types ensured that the system was accessible for both CareLink staff and patients.

8.2.3 Software

The software used for testing was included:

- Microsoft Forms
- Adobe Acrobat Sign
- Microsoft Power Automate
- SharePoint Online
- Outlook App/Online
- Web browsers: Edge, Chrome, and Safari

8.2.4 Interfaces

The system connections tested were:

- Microsoft Forms --> Power Automate
- Power Automate --> Adobe Sign
- Adobe Sign --> Power Automate --> SharePoint
- Power Automate --> Outlook email notification

Each interface was tested individually and then tested again as part of the full workflow.

8.2.5 Other Materials

Additional materials used for testing:

- Sample test patient names and ID
- Sample forms used for repeated submissions
- Access credentials for school Microsoft 365 accounts

8.3 Establishing Environment

The test environment was created within the school's Microsoft 365 tenant using:

- A dedicated Microsoft Form

- A new SharePoint document library
- A Power Automate workflow built for this project
- Adobe Sign integration configured through student accounts

Everything was built specifically for testing and was separated from any real CareLink systems.

8.4 Environment Control

To keep the environment secure and organized:

- Only the project team had edit permissions
- Testing data was monitored and reset as needed
- SharePoint versioning ensured files were preserved and recoverable

8.5 Environment Roles

Each team member had a primary responsibility:

- Form Administrator - Designed and updated the Microsoft Form
- Workflow Developer - Created and updated Power Automate flows
- SharePoint Manager - Monitored document storage
- Testers - Executed test cases and logged results

9. Assumptions and Dependencies

This section lists the main assumptions made during testing and the systems the project depends on.

9.1 Assumptions

- All team members have access to Microsoft 365 tools
- Provided Adobe Sign accounts remain active
- Internet connectivity is stable for testing
- SharePoint permissions remain unchanged

9.2 Dependencies

The project depends on the following services working properly:

- Adobe Sign availability
- Power Automate service uptime
- SharePoint Online storage
- Outlook notification service

The project also depended on the patient intake forms provided by CareLink that were in a fillable PDF format.

10. Entry and Exit Criteria

This section explains when testing was allowed to begin and when it was considered complete.

10.1 Entry Criteria

Before testing could begin, the following had to be ready:

- Digital intake form is created
- Adobe Sign connection configured
- SharePoint library set up
- Power Automate flow deployed
- Test data prepared

10.2 Exit Criteria

Testing was considered complete when:

- All high-priority tests pass
- No workflow failures remain
- Documents successfully route to SharePoint
- Notifications consistently trigger
- Final workflow ready for handoff

11. Administrative Plan

This section explains how approvals, milestones, training, and defect handling were managed.

11.1 Approvals

Final approval of the testing results requires:

- Project Team Lead approval
- Instructor/Sponsor approval
- CareLink representative approval

11.2 Test Milestones

The main testing milestones included:

- Form creation completed
- Workflow integration completed

- Initial testing completed
- Regression testing completed
- Final STR completed

11.3 Training

Training is planned to help CareLink staff understand the system. The training materials given to CareLink staff includes information on:

- Accessing SharePoint submissions
- Understanding digital signatures
- Responding to email notifications

11.4 Defect Management

Defects were handled in a simple way:

- Issues were logged manually by the team
- Fixes were applied by the developer
- Retesting was done after each fix
- No automated defect tracking tool was used

12. Definitions

This section lists some key terms used throughout the entire document:

- STP - Software Test Plan
- STR - Software Test Report
- Workflow - Automated sequence in Power Automate
- Digital Intake - Electronic submission process
- Adobe Sign - Digital signature platform
- SharePoint - File storage for completed forms

13. References

The following resources were used during testing and documentation:

- Project requirements document
- Microsoft 365 documentation
- Adobe Sign documentation
- Power Automate logs

14. Points of Contact

Primary Contact – Industry Sponsor	
Name	Ricnesha Priester
Organization	CareLink of Georgia
Phone	(678)-903-5103
Email	rpriester@carelinkofgeorgia.org

P26-T1 Team Member Contacts	
Aashna Suthar (Team Lead, Documentation)	asuthar@students.kennesaw.edu
Franck Dipanda (Developer, Documentation)	fdipanda@students.kennesaw.edu
Angel Gutierrez (Developer)	agutie34@students.kennesaw.edu
Ayush Kachhadiya (Tester, Documentation)	akachhad@students.kennesaw.edu

15. Software Test Report (STR)

15.1 Summary of Results

- Forms submitted successfully
- Adobe Sign agreements generated and signed
- Documents returned to SharePoint correctly
- Notifications triggered reliably
- Workflow stable under repeated use

15.2 Overall Status

Requirement	Pass	Fail	Severity
Form picker	X		Critical
Automation	X		Critical
Adobe Sign	X		Critical
SharePoint	X		Critical

Final Status: PASS

- The prototype meets the testing goals we defined for this semester and behaves the way we expect for the main intake use case.

15.3 Issues Identified

A few minor issues were found during testing:

- Occasional Outlook delays

    - Sometimes it took a few minutes to arrive
    - This did not break the workflow but is something to be just wary of
- Metadata formatting inconsistencies
    - Some file names and fields were not as clean or consistent as we wanted
    - We adjusted naming rules and variable usage in Power Automate

All of these issues were corrected during development.

15.4 Final Recommendation

Based on all of the testing results, the digital intake workflow is ready for handoff to the CareLink of Georgia staff. The system is stable, easy to use, and performs as expected. It is recommended for approval and future deployment with proper staff training. We do recommend that the staff members do a test trial of the system in small pilot testing. We also recommend gathering their feedback and suggestions on naming, views, and also usability. In addition, making minor adjustments necessary based on real-world experience is also appreciated and highly recommended.