

# Software Design Document (SDD)

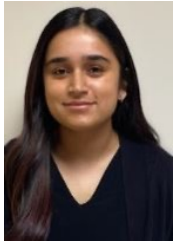
Course: CS 4850 – Section 04

Semester: Fall 2025

Professor: Sharon Perry

Date: September 7, 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

## Table of Contents

<b><i>Table of Contents</i></b> .....	<b>2</b>
<b><i>1.0 Introduction</i></b> .....	<b>4</b>
<b>1.1 Document Outline</b> .....	<b>4</b>
<b>1.2 Document Description</b> .....	<b>4</b>
1.2.1 Introduction .....	4
1.2.2 System Overview .....	4
<b><i>2.0 Design Considerations</i></b> .....	<b>5</b>
<b>2.1 Assumptions and Dependencies</b> .....	<b>5</b>
<b>2.2 General Constraints</b> .....	<b>5</b>
<b>2.3 Goals and Guidelines</b> .....	<b>6</b>
<b>2.4 Development Methods</b> .....	<b>6</b>
<b><i>3.0 Architectural Strategies</i></b> .....	<b>6</b>
<b><i>4.0 System Architecture</i></b> .....	<b>7</b>
<b>4.1 Subsystem Architecture</b> .....	<b>8</b>
<b><i>5.0 Policies and Tactics</i></b> .....	<b>9</b>
<b><i>6.0 Detailed System Design</i></b> .....	<b>11</b>
<b>6.1 Classification</b> .....	<b>12</b>
<b>6.2 Definition</b> .....	<b>12</b>
<b>6.3 Responsibilities</b> .....	<b>13</b>
<b>6.4 Constraints</b> .....	<b>13</b>
<b>6.5 Composition</b> .....	<b>13</b>
<b>6.6 Uses/Interactions</b> .....	<b>14</b>
<b>6.7 Resources</b> .....	<b>14</b>
<b>6.8 Processing</b> .....	<b>14</b>

<b>6.9 Interface/Exports .....</b>	<b>15</b>
<b>6.10 Detailed Subsystem Design .....</b>	<b>15</b>
<b>7.0 Glossary .....</b>	<b>16</b>
<b>8.0 Bibliography.....</b>	<b>16</b>

# 1.0 Introduction

## 1.1 Document Outline

This document describes the technical design for the *Automation of Digital Intake with MS 365 – Carelink of Georgia* project. It takes the requirements from the Software Requirements Specification (SRS) and turns them into:

- Concrete design decisions
- System components and subsystems
- Workflows and data flows
- Notes about tools and constraints

This is what we use as a guide while building and testing the system.

## 1.2 Document Description

### 1.2.1 Introduction

CareLink of Georgia is a nonprofit healthcare provider that works with patients who may not have insurance or steady access to care. Right now, new patient intake is done on paper. Patients fill out forms by hand, and then staff have to scan, save, and organize those forms manually. This takes time, creates extra work, and can slow things down at the clinic.

Our project is focused on replacing that paper intake process (for at least one form) with a digital workflow using Microsoft 365 and Adobe Sign. The goal is to make the process:

- Easier for staff to manage
- Easier for patients to complete and sign
- More organized and secure

We are not building a full EMR or billing system. We are building a digital intake flow that fits into what CareLink already does.

### 1.2.2 System Overview

At a high level, the system works like this:

- Staff use a Form Selection Tool (built in Microsoft Forms) to choose:
  - Staff name
  - Staff email

- Patient name
  - Which intake form to send
- A Power Automate workflow listens for that form submission.
- The workflow pulls the correct blank form PDF from a SharePoint “Blank Forms” library.
- Power Automate then sends that form to Adobe Sign as an agreement for the patient to complete and sign.
- The patient fills it out and signs electronically (on a tablet, clinic computer, or via email link).
- Once the form is signed, the workflow pulls the signed PDF back and saves it into a Form Submissions library in SharePoint.
- Along with the PDF, the system also creates a JSON file with the form data so it can be used later for reporting or integration.

Legacy paper forms will be handled later through a separate “legacy digitization” process. The idea is to scan old forms, extract the data using OCR/AI, map it into a digital version, and store those results in the same SharePoint structure.

## 2.0 Design Considerations

### 2.1 Assumptions and Dependencies

- For this design, we are assuming:
  - CareLink already has working Microsoft 365 licenses (SharePoint, Power Automate, Forms, Outlook).
  - CareLink has access to Adobe Sign (through Adobe Acrobat / Acrobat Sign) for digital signatures.
  - Staff will have access to tablets, desktops, or laptops to run the workflow and help patients sign.
  - Clinic internet is stable enough to support Microsoft 365 and Adobe Sign during intake.
  - CareLink may still use existing systems like Methasoft in the background, but this project does not integrate directly with Methasoft.
- Staff are willing to adjust to a digital workflow as long as it is simple, and they are trained.

### 2.2 General Constraints

- We must follow HIPAA / PHI security expectations (no real patient data in testing, proper storage once live).

- We must stay inside the Microsoft 365 environment and Adobe Sign - no custom backend servers or complex custom apps.
- Nonprofit budget: we focus on low-code / no-code tools they already own.
- Staff have mixed technical experience, so the system must be simple:
  - Clear screens
  - Minimal clicks
  - No “hidden” technical steps for everyday use
- The project must fit within the Fall 2025 timeline and milestones.

## 2.3 Goals and Guidelines

- Overall design goals:
  - Make the system reusable so CareLink can add more forms later.
  - Keep everything simple and staff-friendly, no extra complexity.
  - Keep forms and signed documents secure, auditable, and organized.
  - Use mostly built-in features of Microsoft 365 and Adobe so it’s easier to maintain.
- Keep the design flexible so later on they can connect the JSON data to a database or another system if they want.

## 2.4 Development Methods

The team is following an iterative / MVP approach:

- Start with one intake form and one main end-to-end workflow.
- Get that working from start to finish:
  - Form Selection Tool → Adobe → SharePoint → JSON
- Test it, do small demos, get feedback from our sponsor.
- Adjust the design and documentation based on what we learn.
- Use that as a template for any additional forms or future changes.

## 3.0 Architectural Strategies

- To keep things maintainable for CareLink, our main strategies are:
  - Use low-code tools (Microsoft Forms, Power Automate, SharePoint, Adobe Sign) instead of building a custom web app from scratch.
  - Design with security first:
    - SharePoint permissions and private site
    - M365 authentication
    - No public links to signed intake forms
  - Make the system repeatable for other forms, not just one hard-coded case.

- Use JSON output from the workflow so data can be connected later to reporting dashboards or a database without redesigning everything.
- Keep the architecture as close as possible to what CareLink staff can own and maintain after the semester ends.

## 4.0 System Architecture

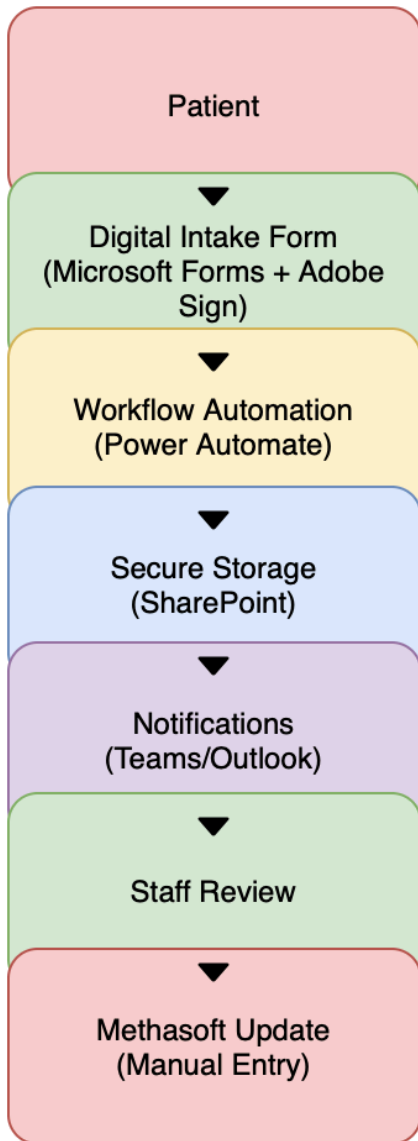
*Figure 1: High-Level Digital Intake Workflow*

At a high level, the digital intake process looks like this:

1. Staff Form Selection
  - a. Staff fills out the Microsoft Form “Form Selection Tool” with:
    - i. Staff Name
    - ii. Staff Email
    - iii. Patient Name
    - iv. Selected Form
2. Power Automate Trigger
  - a. A Power Automate flow is triggered whenever a new response is submitted.
3. Form + Signature Workflow
  - a. The flow:
    - i. Reads the form response
    - ii. Looks up the correct blank PDF from the “Blank Forms” SharePoint library
    - iii. Creates an Adobe Sign agreement from that PDF
    - iv. Sends it out for the patient to sign
4. Patient Signing (Adobe Sign)
  - a. Patient opens the Adobe Sign link on a clinic device (or via email)
  - b. Fills in the fields and signs electronically
5. Return and Storage
  - a. Once signed, Power Automate:
    - i. Uses a “Do Until” loop with delays to wait until Adobe confirms the agreement is completed
    - ii. Downloads the signed PDF
    - iii. Saves it into the “Form Submissions” library in SharePoint
    - iv. Creates a JSON file with the form data and stores it alongside the PDF
6. Staff Review
  - a. Staff can open the Form Submissions library to:
    - i. Search by patient name, form name, or date

- ii. Open the signed PDF
- iii. Download or print if needed

The optional legacy digitization flow will feed data into the same SharePoint structures (PDF + JSON) so everything stays consistent.



## 4.1 Subsystem Architecture

We can break the system into four main subsystems:

1. Form Selection & Intake
  - a. Microsoft Forms “Form Selection Tool” (staff-facing)



- b. Connects to Power Automate as the main trigger
- 2. Workflow Automation
  - a. Power Automate intake flow:
    - i. Reads form responses
    - ii. Grabs the correct blank PDF from SharePoint
    - iii. Sends to Adobe Sign
    - iv. Waits for signing (Do Until + Delay)
    - v. Gets signed PDF + data
    - vi. Stores PDF + JSON in SharePoint
- 3. Storage & Organization
  - a. SharePoint site with:
    - i. Blank Forms library
    - ii. Form Submissions library (signed PDFs + JSON)
    - iii. Project Manager list for internal links and documentation
- 4. Legacy Form Digitization (Planned)
  - a. Process for:
    - i. Scanning old paper forms
    - ii. Using OCR / AI (e.g., ChatGPT) to pull out field values
    - iii. Filling a digital version of the form via a Python script
    - iv. Storing the result in the same SharePoint structure

High-Level Flow (text form):

- Staff --> Microsoft Form Selection Tool
- Microsoft Form --> Power Automate Flow
- Power Automate --> SharePoint (Blank Forms) + Adobe Sign
- Adobe Sign --> Power Automate (signed PDF + data)
- Power Automate --> SharePoint (Form Submissions + JSON)
- Staff --> SharePoint (review and retrieval)

## 5.0 Policies and Tactics

To keep the system stable and usable, we are following these design policies:

- Error Handling
  - Use Power Automate's built-in error handling and run history to check failures.
  - If the signed document is not returned correctly, staff can still access it from the Adobe Sign account as a backup.
- Security
  - Use a private SharePoint site with limited access.

- All libraries live under that site with proper permissions.
  - Only specific staff and administrators can see signed forms.
- Backup / Recovery
  - Rely on SharePoint version history and recycle bin for backups.
  - Adobe Sign also keeps a copy of signed forms.
- Training
  - Provide a user guide with screenshots.
  - Show staff how to:
    - Use the Form Selection Tool
    - Track Adobe Sign agreements
    - Find signed forms in SharePoint

## 6.0 Detailed System Design

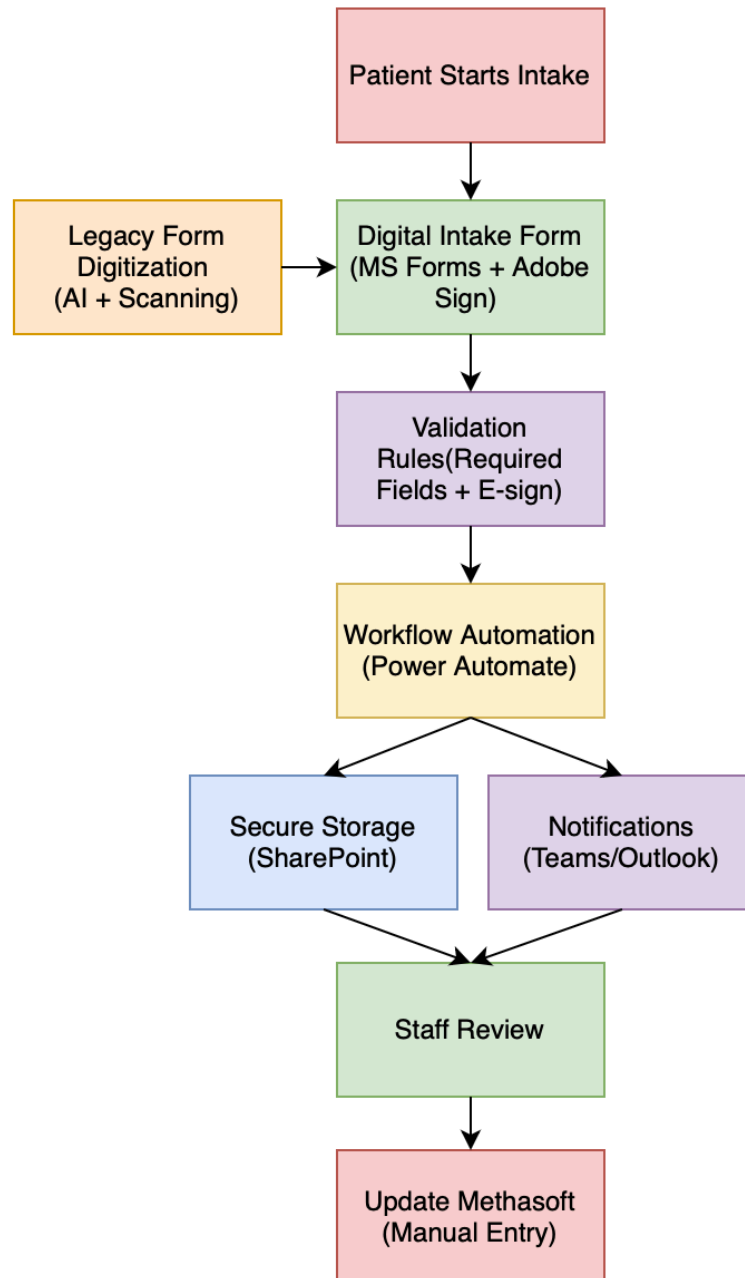


Figure 2: Detailed System Design Flowchart

This diagram shows the step-by-step flow of the system. The patient fills out the intake form, and Power Automate takes over after submission. One path sends the form to SharePoint for storage, and another sends staff a notification through Teams or Outlook. Both paths come together at the staff review step, where the form is checked and then updated into Methasoft. The legacy form

digitization is also shown as feeding into the intake process so paper forms can go through the same flow.

## 6.1 Classification

This system consists of four parts:

1. Digital Intake Form: This component of the system contains the intake form that will be filled out by the patients and allows for electronic signatures.
2. Workflow Automation: The automation will be done using Microsoft Power Automate for integrating the e-signature functionality, sending notifications to staff after form completion, and to transfer the information to Methasoft.
3. Storage: The system will use Microsoft SharePoint for storing the forms, which also allows for enhanced security using permissions, which manages who can view/edit the information.
4. Legacy Form Digitization: This part of the system will allow legacy paper forms to be scanned and converted into a digital format using AI tools then processed into SharePoint storage.

## 6.2 Definition

To define each part a bit more:

- Form Selection Tool
  - Simple Microsoft Form with fields for staff name, staff email, patient name, and form name.
  - This is the “front door” into the workflow.
- Workflow Automation (Power Automate)
  - Trigger: When a new response is submitted in the Form Selection Tool.
  - Actions:
    - Get response details
    - Initialize variables (patient name, staff name, form title, date/time)
    - Get file content for the selected PDF from SharePoint
    - Create an Adobe Sign agreement
    - Wait until the agreement is signed (Do Until loop + Delay)
    - Retrieve the signed PDF as a file
    - Store it in the Form Submissions library
    - Create a JSON object with key fields and store that as a separate file
- Storage
  - Blank Forms library holds the original, unsent intake forms.
  - Form Submissions library holds signed PDFs + their JSON data.
- Legacy Digitization

- Scan a completed paper form → PDF
- Use OCR / AI to get the structured data
- Run a Python script to fill in a digital version
- Store it back into SharePoint, same libraries as everything else

## 6.3 Responsibilities

Each subsystem has its own role:

- Form Selection Tool
  - Collects the basic info needed to start the workflow.
  - Prevents staff from having to manually attach forms each time.
- Workflow Automation
  - Handles all the “behind the scenes” work after the form is submitted.
  - Ensures the right form goes to Adobe.
  - Waits for signing and stores everything in the correct place.
- Storage Subsystem
  - Keeps all documents organized and secure.
  - Makes it easy for staff to find what they need later.
- Legacy Digitization Subsystem
  - Gives CareLink a path to clean up and digitize older paper records in the future.

## 6.4 Constraints

Some design constraints include:

- We must stay inside Microsoft 365 + Adobe Sign (no custom servers or big custom apps).
- We do not have API access to systems like Methasoft and are not integrating directly.
- We must follow HIPAA expectations (no real PHI in development, secure storage in production).
- No extra budget for third-party tools beyond what CareLink already has.
- The system should work on low-cost hardware and clinic devices.

## 6.5 Composition

Technologies we are using together:

- Microsoft Forms – Form Selection Tool (staff-facing).
- Power Automate – Orchestrates the entire intake workflow.
- Adobe Sign – Handles filling and signing of the actual intake document.

- SharePoint – Stores blank forms, signed forms, and JSON data; also holds some project documentation.
- ChatGPT / OCR + Python (for legacy forms) – Planned for converting scanned paper forms into structured data and filling digital templates.

## 6.6 Uses/Interactions

How people and systems interact:

- Staff
  - Use the Form Selection Tool to kick off the workflow.
  - Open Outlook to see the Adobe Sign email (if needed).
  - Use SharePoint to review completed forms.
- Patients
  - See and complete the intake form through Adobe Sign (on a clinic device or via email).
- System
  - Power Automate connects Microsoft Forms, Adobe Sign, and SharePoint.
  - JSON data and PDFs are stored so future integration is possible.

## 6.7 Resources

We expect the following resources:

- Hardware
  - Clinic desktops or laptops for staff.
  - Tablets or kiosk-style devices for patient signing (if available).
- Software
  - Microsoft 365 (Forms, SharePoint, Power Automate, Outlook).
  - Adobe Sign / Adobe Acrobat.
  - Optional: Python + OCR/AI tools for legacy forms.
- Documentation & Training
  - User guide and screenshots.
  - Short demos for staff.

## 6.8 Processing

High-level processing steps:

1. Staff fills out the Form Selection Tool.
2. Power Automate flow runs and sets up the Adobe Sign agreement.
3. Patient completes and signs the form in Adobe Sign.
4. Flow waits until the agreement is completed.

5. Flow pulls the signed PDF + data and stores everything in SharePoint.
6. Staff can then search and access the form as needed.

## 6.9 Interface/Exports

### User Interfaces

- Microsoft Forms interface for staff.
- Adobe Sign interface for patients.
- SharePoint document libraries and lists for staff to browse and open files.

### Exports

- Signed forms exported/stored as PDFs.
- Structured data exported/stored as JSON (and could later be exported as CSV if needed).

## 6.10 Detailed Subsystem Design

### Digital Intake / Form Selection

- Required fields on the Microsoft Form help make sure the flow always has the data it needs (no blank patient name, etc.).

### Signature Flow

- Adobe Sign is configured through Power Automate to:
- Attach the correct blank PDF
- Use a meaningful naming pattern (e.g., PatientName\_FormName\_Date)
- Allow multiple signature options (type, draw, upload, mobile).

### Storage Subsystem

- Libraries in SharePoint:
- Blank Forms – Original copies of intake forms
- Form Submissions – Signed PDFs + JSON
- Metadata: patient name, staff name, form name, date submitted, etc.

### Automation Subsystem

- Uses variables, loops, and delay actions so it can wait for Adobe without blocking.
- Handles success and failure paths (e.g., what happens if Adobe doesn't return a signed form).

## 7.0 Glossary

- SharePoint: Microsoft platform for storing and organizing files and lists.
- Power Automate: Microsoft tool for building automated workflows.
- Adobe Sign: Digital signature and document workflow tool from Adobe.
- M365: Microsoft 365 (SharePoint, Forms, Outlook, Power Automate, etc.).
- HIPAA: Health Insurance Portability and Accountability Act (U.S. healthcare privacy law).
- PHI: Protected Health Information.
- JSON: JavaScript Object Notation, used here for structured form data.
- MVP: Minimum Viable Product (first working version with core features).

## 8.0 Bibliography

- CareLink of Georgia – <https://carelinkofgeorgia.org>
- Microsoft 365 Documentation – <https://learn.microsoft.com/microsoft-365>
- Microsoft Power Automate – <https://learn.microsoft.com/power-automate/>
- Adobe Acrobat / Adobe Sign – <https://www.adobe.com/acrobat/>
- HIPAA Regulations – <https://www.hhs.gov/hipaa>