Software Design Document (SDD)

Project: **Lab Activity 14**

Team Name: **Tommy W., Nadia H., Damarion M.P, Samantha V.M, Stephany C.P**

Group Number: **4**

Version: **4.0**

Date: **December 5, 2025**

Contents

Version Description

Version	Description	Date
1.0	Snapshot 1. Created system overview and basic architecture.	December 5, 2025
2.0	Snapshot 2. Expanded architecture; added AWS + Box components.	December 5, 2025
3.0	Snapshot 3. Added workflow details and error-handling descriptions.	December 8, 2025
4.0	Snapshot 4. Final revisions; added UI section, glossary updates, formatting fixes.	December 10, 2025

1 Introduction

1.1 Purpose of the Document

The purpose of this document is to provide a detail description of the Design details that will be used for Box Discovery Bates Namer. This sofware is a transition from a desktop-based system to a cloud based one. This Sofware Design Document(SSD) covers the system architecture, user interface and intended audience for the application.

1.2 Intended Audience

This document is intended for:

- The development team implementing the system.
- The course instructor and teaching assistants reviewing the design.
- Future developers who may extend or maintain the system.
- Various Stakeholders

1.3 Overview of the System

The **Box Discovery Bates Namer** is a cloud based solution designed to automate the prossessing the legal discovery PDF files. The applications extrats Bates numbers, Validates their sequence and renames the files accordingly and are stored in Box.com.

The System leverages a **serverless architecture** using **AWS Lambda functions** for scalability and efficiency. Upon a file upload event to **Box.com** , the application processes the files in memory to preserve data confidentiality and complies with data privacy requirements.

1.3.1 Key Features

Integration with Box.com

- Triggered by an HTTPS API Gateway call when files are uploaded to Box.com.
- Retrieves file details, including file ID, folder ID, and file name, from the event payload.

Custom Payload Creation

- An initial AWS Lambda function captures the file metadata and creates a custom payload, which is passed to the next Lambda function for processing.

This system provides a robust and efficient solution for legal professionals by automating discovery file management, enhancing accuracy, and streamlining the workflow.

2 System Architecture

2.1 Overall Workflow of the System

The Box Discovery Bates Namer follows an automated, event-driven workflow that integrates Box.com with AWS Lambda to process, validate, and rename legal discovery PDF files. The workflow begins when the user uploads a discovery folder into Box.com and ends when files are renamed, validated, organized, or flagged for errors. The following summarizes the complete workflow as described.

1. **User Uploads Discovery Files to Box.com**
   A legal professional uploads a folder using the naming format `PDCase#_Disc#` (e.g., `PD251234_02`). Each PDF uploaded triggers the Box Custom Skill, which sends an event to the system.

2. **Box Custom Skill Triggers AWS Processing Pipeline**
   Box issues a webhook request to an HTTPS API Gateway endpoint, which invokes the first Lambda function.

3. **BoxInputFunction Captures File Metadata**
   The initial Lambda function validates the webhook and extracts file metadata including:

   - `file_id`
   - `file_name`
   - `folder_id`
   - `access_token`
   - `user_id`

   A custom payload is constructed and forwarded to the next Lambda function.

This automated workflow eliminates manual processing, ensures file consistency, and provides reliable error reporting for legal professionals.

2.2 System Components

The Box Discovery Bates Namer system is built on a serverless architecture using AWS Lambda and Box.com. The major components are:

2.2.1 AWS Lambda Functions

Event-driven functions that perform the core processing:

- **BoxInputFunction** – Validates the webhook and extracts file metadata.
- **BoxFolderGetter** – Retrieves the folder name to identify the PD Case and Disc Number.

- **DiscoveryBatesNamer** – Downloads the PDF, extracts and validates Bates numbers, and generates the new filename.
- **BoxFileUpdater** – Renames and moves the processed file into the correct Box folder.
- **BoxErrorNotification** – Sends user error notifications via email.

2.2.2 Box API

Handles file storage operations, including:

- Downloading files for processing,
- Renaming files,
- Moving files to the appropriate destination folder.

2.2.3 AWS API Gateway

Provides the HTTPS endpoint that receives webhook events from Box and triggers the Lambda pipeline.

2.2.4 AWS SES

Sends email notifications to users when a file cannot be processed.

2.2.5 AWS CloudWatch

Logs all Lambda activity for debugging and monitoring.

2.3 Deployment Architecture

The Box Discovery Bates Namer is deployed entirely using a serverless cloud architecture. No traditional servers, containers, or databases are required. All processing occurs through AWS-managed services that scale automatically.

- **AWS Lambda** hosts all processing logic for metadata extraction, Bates number parsing, file renaming, and error notifications. Each Lambda function runs independently and is invoked only when needed, reducing cost and improving scalability.
- **AWS API Gateway** provides the public HTTPS endpoint that receives webhook events from Box.com and triggers the first Lambda function.
- **AWS Secrets Manager** stores Box API credentials securely and provides them to Lambda functions at runtime.
- **Box Custom Skill** triggers the pipeline whenever a PDF is uploaded into the designated folder in Box.com.

- **AWS CloudWatch** records execution logs, performance metrics, and error reports for all Lambda functions.

Because the system is fully serverless, no container orchestration, local servers, or manual deployment environments are required.

3 User Interface

3.1 Overview

The Box Discovery Bates Namer does not require a custom user interface. All user interaction occurs through the existing Box.com platform. Users upload discovery folders and PDF files directly into a designated skill-enabled folder in Box, and the system automatically processes each file through AWS Lambda without requiring further user input.

3.2 How Users Interact With the System

- Users log into Box.com and navigate to the skill-enabled "Skills Applied" folder.
- The user uploads a folder named using the required format (e.g., `PD251234_02`).
- As files are uploaded, the system automatically extracts Bates numbers, renames files, and moves them to the appropriate destination folder.
- If errors occur (such as missing Bates stamps or incorrect folder naming), the user receives an email notification with details and a link to the affected file.

3.3 User Interface Screens

The user experience consists of standard Box.com screens, including:

- The Box folder view where discovery files are uploaded.
- The upload interface used to add PDF files to the designated folder.

All processing occurs automatically in the background; therefore, no additional UI elements are required.

3.4 Data Storage and Management

The system does not use a traditional relational database. Instead, it relies on cloud-managed storage services to maintain the information required for processing and auditing:

- **Box.com** acts as the primary storage system for discovery files. It stores the original PDFs, renamed documents, and the folder structure used to organize files by PD Case Number and Disc Number.
- **AWS Secrets Manager** stores sensitive credentials such as Box API keys and authentication tokens. These values are securely retrieved by Lambda functions at runtime.
- **AWS CloudWatch Logs** serves as the operational record-keeping system, storing processing logs, error reports, execution traces, and audit information for each file that moves through the pipeline.

Together, these services fulfill the system's storage, security, and traceability needs without requiring a standalone database engine.

4 Glossary

Acronym	Definition
UI	User Interface
API	Application Programming Interface
DB	Database
SRS	Software Requirements Specification
SDD	Software Design Document

Key Deffinitions	Definition
Bates Number	Commonly used in the discovery phase of legal cases, bates numbers are a sequential identifier assigned to each page of a legal document for tracking and referencing
Box.com	A cloud-based file storage platform used for document organization and collaboration
AWS Lambda	A serverless computing service provided by Amazon Web Services that executes code in response to events, eliminating the need to manage and maintain servers

5 References

References

[1] Santa Barbara Public Defender's Office. *Software Design Document for Box Discovery Bates Namer Integration into Cloud Environment*. Version 1.1.2, April 10, 2025. https://ascent.cysun.org/project/project/view/221