Cheetah Sign

Team Information

Capstone Team

Alec Burch-DeWitt

Christian Johnson

Christopher Ortega

Dominic Szpak

Trinity Washington

Client Information

cheetah

an accutech company

- A software company that manages workflow processes
- Document signing (Document Sign, Hello Sign)
- People:
  - Josh Rittenhouse
  - Gabe Chandler
  - Ryland Babusiak

Business Requirement

Business Requirement

We are continuing the development of the Cheetah Sign application with the goal of implementing the needed features required to deploy the app in the Cheetah Flow system as a replacement for expensive and proprietary alternatives.

IN WITNESS WHEREOF, the parties have executed this Agreement as stated below:

Client

Name: William Jackson

Title: VP of Marketing, Hanford Inc

Date: 08 / 10 / 21

Consultant

Click to sign *

Name:

Title:

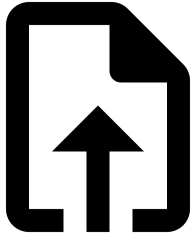Date: ___ / ___ / ___

Use Cases

Use Cases

UC1: Manage Documents
➜ *Administrators*
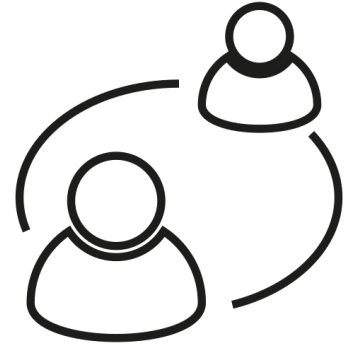
Upload

Favorite

View History

UC2: Sign Documents
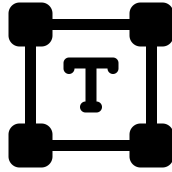➜ *Admins and Clients*

Legal Signature

Designate Order

Use Cases

UC3: Prepare and Send Documents
➜ *Administrators*

Drag and Drop Entry Fields

Pre-Filled Fields
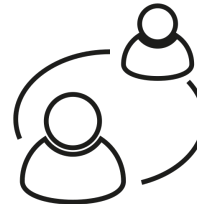
Send Prepared Documents

UC4: Sign and Return Documents
➜ *Clients*

Receive Documents

Sign Marked Fields
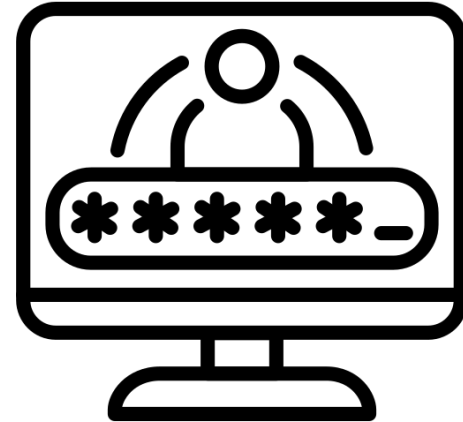
Send Signed Documents

Use Cases

UC5: Audit Trail of Document Status
➜ *Administrators*

UC6: Login System
➜ *Administrators*

Requirements

Requirements

High-Level FR

- System will be able to handle uploading, preparing, sending, tracking, and managing documents with full audit trails.
- Admins will have full access
- Clients will receive and sign documents via email.
- Multi signer, multi-document workflows, with audit trails

Non-Functional Requirements

- When the system opens, the login screen will be the first page visible to the admins
- When a file is sent over email, the admin will be able to send a full email that also includes the link, instead of blank email with the link.
- Documents can populate multiple lists, as there is the original version of document and the signed version that are stored separately.
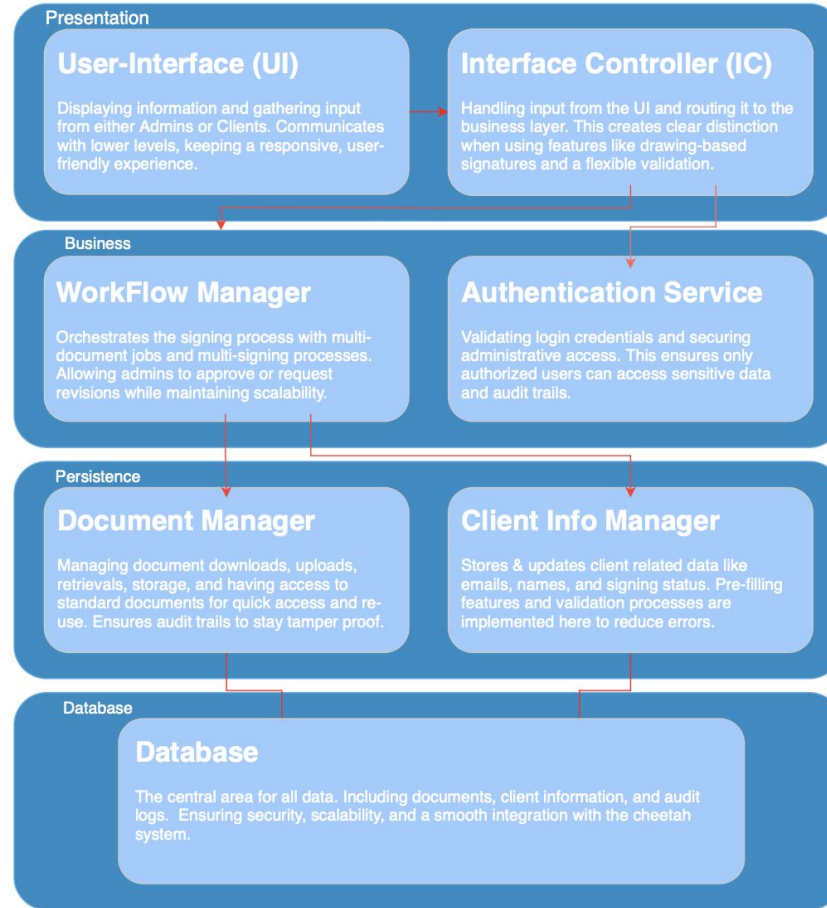- The audit trails will not be visible directly within the document, but rather in the admin tools.

Architecture

Software Architecture

- Admin uploads package

- Workflow manager creates a signing job/sends to client

- Authentication service verifies the client

- Client sign via email

- Document manager saves & logs audit trail in the database

Presentation

User-Interface (UI)

Displaying information and gathering input from either Admins or Clients. Communicates with lower levels, keeping a responsive, user-friendly experience.

Interface Controller (IC)

Handling input from the UI and routing it to the business layer. This creates clear distinction when using features like drawing-based signatures and a flexible validation.

Business

WorkFlow Manager

Orchestrates the signing process with multi-document jobs and multi-signing processes. Allowing admins to approve or request revisions while maintaining scalability.

Authentication Service

Validating login credentials and securing administrative access. This ensures only authorized users can access sensitive data and audit trails.

Persistence

Document Manager

Managing document downloads, uploads, retrievals, storage, and having access to standard documents for quick access and re-use. Ensures audit trails to stay tamper proof.

Client Info Manager

Stores & updates client related data like emails, names, and signing status. Pre-filling features and validation processes are implemented here to reduce errors.

Database

Database

The central area for all data. Including documents, client information, and audit logs. Ensuring security, scalability, and a smooth integration with the cheetah system.
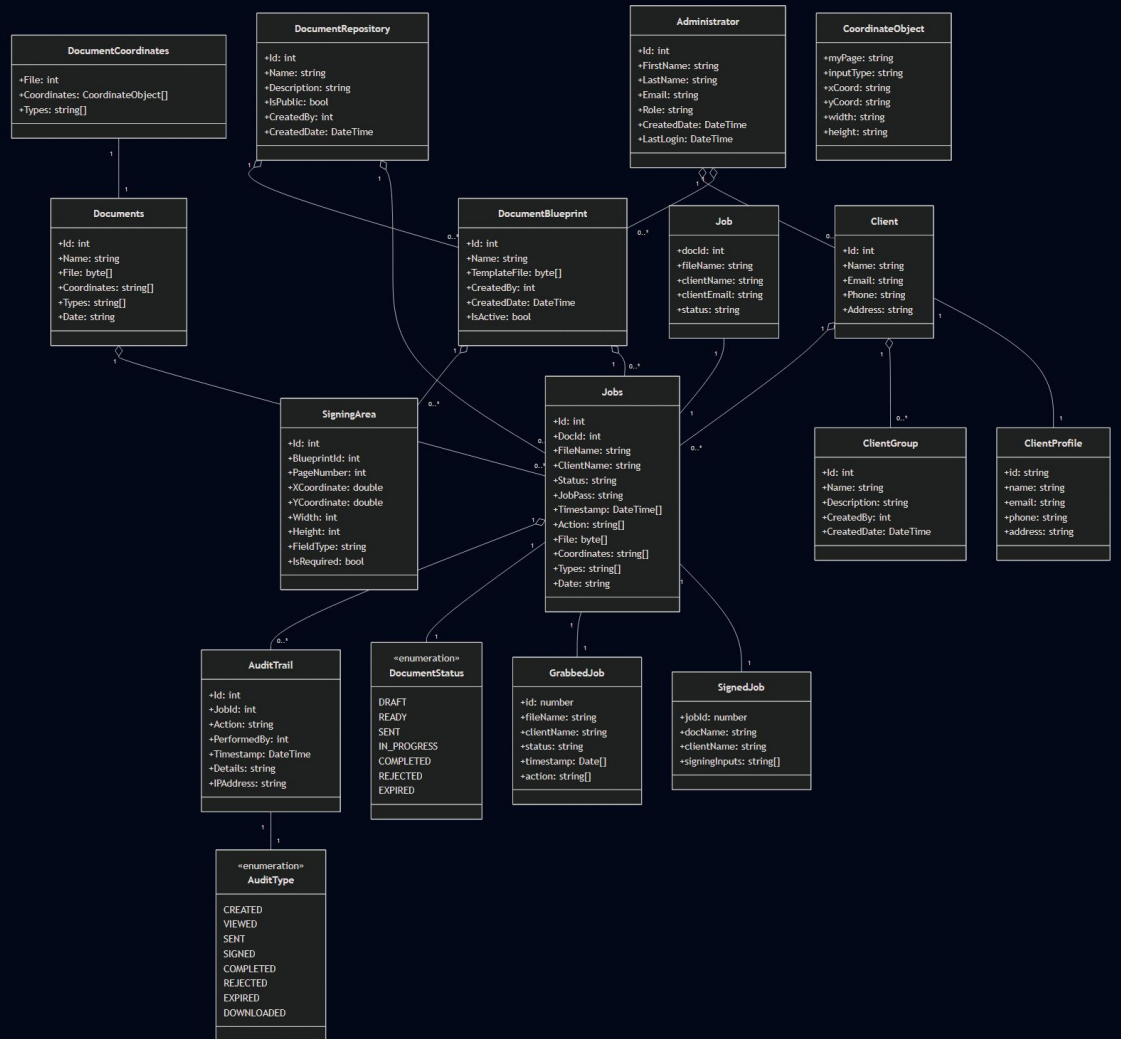
Domain Model

Domain Model

AuditType - Types of actions that can be tracked:

- CREATED: New document job created
- SENT: Document sent to client
- VIEWED: Client opened document
- SIGNED: Client signed document
- COMPLETED: All actions finished
- REJECTED: Document rejected
- EXPIRED: Document expired

DocumentStatus - Current state of a document:

- DRAFT: Still being prepared
- READY: Ready to send
- SENT: Sent to client
- IN_PROGRESS: Client is signing
- COMPLETED: Fully processed
- REJECTED: Rejected by client
- EXPIRED: Past due date Frontend Models

Prototype

Prototype

Login

My Documents

Prototype

View Document

Dummy Doc

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Signature #1   Signature #2

Name   Name

Date   Date

Audit Trail

Client Signers: Christian Johnson, Alec Burch-DeWitt

Document Signed: 9/24/25 4:37 PM

Document Opened: 9/24/25 4:36 PM

Document Sent: 9/20/25 2:25 PM

Document Created: 9/20/25 1:50 PM

Document Completed: 9/20/25 4:40 PM

Clients Tab

Sort ⌄

Name	Email	Address	Phone	
Olivia Roussel	oroussel@example.com	456 Study Street Campus Town, ST 12345	(260) 444-4444	🗑
Gabby Spier	gspier@example.com	789 Homework Drive College City, ST 67890	(260) 333-3333	🗑
Emilee Dailey	edailey@example.com	321 Assignment Blvd University Place, ST 11223	(765) 888-8888	🗑

New +

My Documents   Document Repository   Clients

Prototype

Editing a Document

TechStack

Frontend

- **Node.js 20** - Node.js 20 is the runtime we need to run and build the frontend. It also gives us access to npm, which is used to install extra tools and packages. Our client wanted us to use Node.js because it supports Vue and Vite, which are required parts of their setup.

- **Vue 3** - Vue.js is the main framework for the frontend. It lets us build a single-page application where the user can move between screens smoothly without the page reloading. Vue is easier to learn than some other frameworks and is already used in the client's other products.

- **Vite 5** - Vite 5 is the build tool and development server for Vue. It makes starting the app very fast which means we can see code changes right away without restarting. This makes development smoother and was included by the client in the base setup for Cheetah Sign.

- **Visual Studio Code** - VS Code is the IDE chosen by the client for this project. It is lightweight, has extensions for Vue, Node.js, and Docker, and is easy for the team to use together. The client specifically recommended VS Code with the Docker Compose extension for working on Cheetah Sign.

Backend

- **NET 8 SDK** - s the framework we use for the backend services. It is reliable, cross-platform, and supports high-performance web APIs. We will write the backend in C#, which is well supported by .NET. The client wanted us to use .NET because it is already used in other Cheetah projects and integrates well with Docker and PostgreSQL.

Database

- **PostgreSQL** - PostgreSQL is the relational database chosen by our client.
- **PgAdmin** - pgAdmin is the tool we use to manage and check the database.

Development Tools

- **Docker** - Docker is used to containerize the frontend, backend, and database so the system runs the same on every team member's computer. Running docker starts all four parts of the project (API, frontend, Postgres database, and pgAdmin)
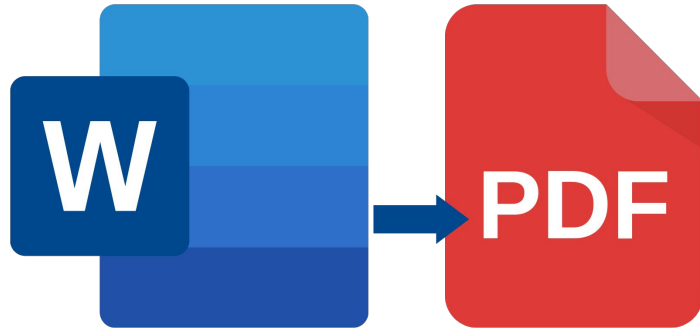
First Iteration Features

First Iteration Features

Field Validation

Document
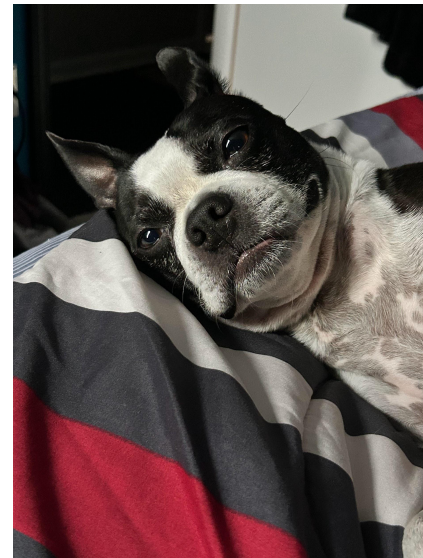Conversion

Client Feedback

Client Feedback

- In our client meeting we went over our documents with the understanding that many of them would likely be similar to the previous team's design, since we are building off of it.
- They said that our documents looked pretty good and seemed to incorporate the new features they set out to achieve for this continuation of the project.

Fun slide!

Q & A