

URL

<http://classwork.engr.oregonstate.edu:9865/index.html>

Executive Summary

Over the course of this project (approximately 10 weeks) our team implemented peer feedback at each step along the way. This feedback did not majorly impact our original plan but mostly consisted of fixing typos and elements within our website to be fully functional. Below are the specific changes we made at each step.

Step 1:

Our biggest changes at this point were swapping the ERD and Schema (which we had confused earlier. We also standardized the names of our entities and attributes to be cleaner and align with the teaching material instructions.

Step 2:

In step two, we made some final alterations to our website layout to match the ERD and Schema in places where they were inconsistent.

Step 3:

At this point, we added sample data for each of our tables/entities. We also started implementing the front-end user interface and added buttons for delete and edit functionality.

Step 4:

In step four, we were really just working on the CRUD functions for each page. We were able to generate data from our DDL and DML files and sample data for most pages (Read), and implemented the Delete function for a couple pages. We also had an add (Create) button working completely for one page. Our biggest hurdle was the update function.

Step 5:

At this point we really were just still struggling with update functionality as that is the hardest element of CRUD to get working. We were able to get edit buttons working on a couple pages and took the option off all other pages so as not to confuse users.

Step 6:

We finished up as many loose ends as possible so that the website felt complete and functional.

In summary, this process taught us a lot about databases and working with a team to build out a website from outline, to draft, to final working UI.

Project Outline

Inked In Overview

Inked In is a specialized e-Signature service that is designed to address the needs of companies with extensive e-file storage and high volumes of document processing requirements. Many companies manage over 10,000 digital documents and process around 200 signed forms daily. It is essential to implement an efficient and secure e-signature solution. With Inked In, companies can store up to 400,000 digital documents while processing 1,000 forms and signatures per week ensuring smooth document handling and efficient processing. This service allows companies to organize and track their files as well as securely collaborating with a team to streamline their e-filing processes, and enhance analytical capabilities.. The service will handle and manage relationships between Users to their Files, and Files to their Forms and Signatures, ensuring traceability.

To address the challenges presented currently by companies unable to keep up with digital documents, we present a database-driven e-signature service that is designed to effectively and efficiently manage large volumes of files. The proposed database solution will support Inked In in achieving the following goals:

- Comprehensive file management, allowing users to store, access, and organize large volumes of documents.
- A signature system that links forms to specific files, ensuring traceability and secure document handling.
- Real-time tracking of document statuses, user activity, and signature completion.
- The ability to process 200 or more signed forms per day, ensuring scalability for large organizations and efficient workflow.

Overall Objective

The goal is to create an easy-to-use and reliable e-signature system that helps companies manage large amounts of files while making it simpler to sign documents. This system will keep files organized to simplify the signing process while providing clear reports for businesses to easily manage.

Database Outline

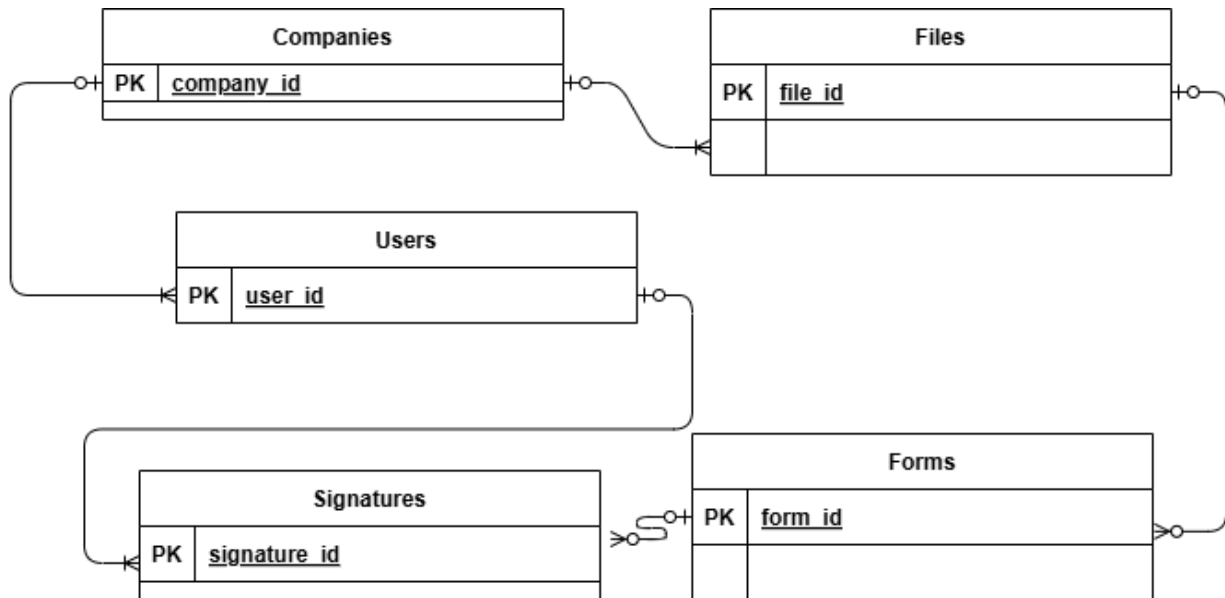
Inked In Database Outline

Entities:

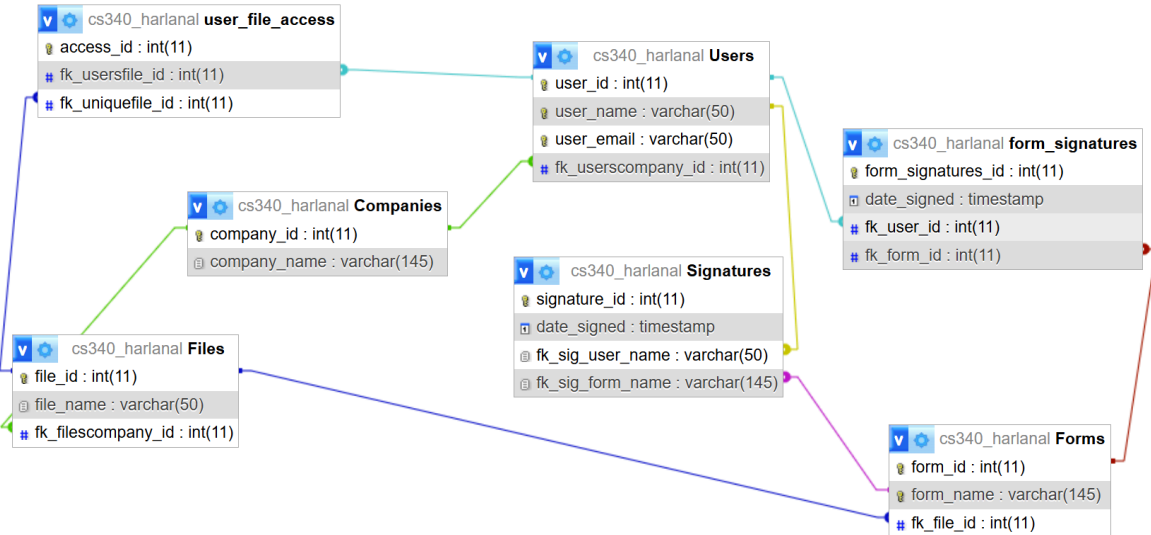
- Companies: records a company to which many employees (Users) belong
 - company_id: int, auto_increment, unique, not NULL, PK
 - company_name: varchar(145), not NULL
 - Relationship:
 - 1:M between Companies and its employees (User)
- Users: records the details about Users of the e-signature service
 - user_id: int, auto_increment, unique, not NULL, PK
 - user_name: varchar(50), not NULL
 - user_email: varchar(50), unique, not NULL
 - fk_userscompany_id: int, auto_increment, unique, not NULL, FK(References Companies)
 - Relationship:
 - 1:M One of the Users belongs to one of the Companies which can have many Users
 - M:N A user can sign multiple Forms and Forms can have Signatures from many Users
 - M:N A user can access multiple Files and Files can be used by many Users
- Forms: represent documents to be stored within Files and signed multiple times by Users
 - form_id: int, auto_increment, unique, not NULL, PK
 - form_name: varchar(145), not NULL
 - fk_file_id: int, auto_increment, unique, not NULL, FK(referencing Files)
 - Relationship:
 - 1:M - one to many, because a Forms can have multiple Signatures being signed by several Users
 - 1:M - one to many, because a Files can be made up of multiple Forms but Forms will only be used once in given Files
 - M:N - many to many, because Forms can be accessed by many Users and a Users can access multiple Forms
- Files: records the details of Files; files can have multiple Forms, multiple Users from the same Company can be given access to the same Files
 - file_id: int, auto_increment, unique, not NULL, PK
 - file_name: varchar(50), not NULL
 - fk_filescompany_id: int, unique, not NULL, FK(Referencing Companies)
 - Relationship:
 - 1:M - A File can contain multiple Forms, serves as storage for multiple documents

- M:N - Users can access multiple files, and files can be accessed by multiple users.
- Signatures : Records an instance of a User signing a Form, includes date signed.
 - signature_id: int, auto_increment, unique, not NULL, PK
 - date_signed: timestamp, not NULL (Note: use timestamp instead of datetime or date so that each Signatures have a unique timestamp and cannot be duplicated)
 - fk_usersig_id: int, NULL, FK(referencing Users)
 - fk_usersform_id: int, NULL, FK(referencing Forms)
 - Relationship:
 - 1:M - Forms can have multiple Signatures, but each of the Signatures is a unique signing event with a secure timestamp
- user_file_access: A join table that will manage access rights for users to files
 - access_id: int, auto-increment, unique, not NULL, PK
 - fk_usersfile_id: INT, NULL, FK (referencing Users)
 - fk_uniquefile_id: INT, NULL, FK(referencing Files)
 - Relationships:
 - M:N - indicates which users can access which files
- 1. form_signatures: A join table that establishes the relationship between a form and the signatures on it.
 - a. form_figntures_id: INT, NOT NULL, PK
 - b. date_signed: timestamp, not NULL
 - c. fk_user_id: INT, NULL, FK (referencing Users)
 - d. fk_form_id: INT, NULL, FK (referencing Forms)
 - e. Relationship:
 - i. 1:M - indicates that a form can have signatures from multiple users

ERD



Schema



Sample Data

Sample Companies Data

company_id	company_name
1	McDonalds
2	Starbucks
3	Black Bear Diner

Sample Users Data

user_id	user_name	user_email	fk_userscompany_id
1	Grimace	grim@mickeydees.com	1
2	Ronald McDonald	ronnie@mickeydees.com	1
3	Birdie the Early Bird	mcdonaldsbirdie@mickeydees.com	1

Sample Files Data

file_id	file_name	fk_filescompany_id
1	Menus	1
2	Employee Agreements	1
3	Release of Liability	1

Sample Forms Data

form_id	form_name	fk_file_id
1	\$1 Menu	1
2	Burger Flipper Agreement	2
3	PlayPlace Release of Liability	3

Sample Signatures Data

signature_id	date_signed	fk_usersig_id	fk_usersform_id
1	2024-10-31 18:53:15	1	1
2	2024-10-31 18:53:15	2	2
3	2024-10-31 18:53:15	3	3

Screen Captures of Pages' UI

Home Screen : **Read**/Browse/Display current companies and related actions for each page

Inked-in E-Signature Service

Manage Companies	Manage Users	Manage Files	Manage Forms	Manage Signatures
McDonalds	McDonalds Users	McDonalds Files	McDonalds Forms	McDonalds Signatures
Starbucks	Starbucks Users	Starbucks Files	Starbucks Forms	Starbucks Signatures
Black Bear Diner	Black Bear Diner Users	Black Bear Diner Files	Black Bear Diner Forms	Black Bear Diner Signatures
Subway	Subway Users	Subway Files	Subway Forms	Subway Signatures
Voodoo Donuts	Voodoo Donuts Users	Voodoo Donuts Files	Voodoo Donuts Forms	Voodoo Donuts Signatures

Manage Companies Page: Browse/**Read**/Display current companies

Supported actions: **Create** a company, **Read** all Companies **Update** a Company, **Delete** a company

Select Company To Manage

[Back to Companies](#)

Current Companies

Company ID	Company Name	Actions
1	McDonalds	Delete
2	Starbucks	Delete
3	Black Bear Diner	Delete
4	Subway	Delete
5	Voodoo Donuts	Delete

Add Company

Name:

[Add Company](#)

Update Company

Name: Company ID:

[Save Update Company](#) [cancel](#)

Manage McDonalds' Users Page: Browse, **Read**, Display McDonald's Users
Supported Actions: **Delete** a User, **Create** a User

Manage Users

[Back to Companies](#)

Users for Company:

User ID	User Name	Email	Actions
2	Ronald McDonald	ronnie@mickeydees.com	Delete
15	Hamburglar	hamburglar@mickeydees.com	Delete
16	Grimace	grimace@mickeydees.com	Delete
17	Birdie the Early Bird	birdietheearlybird@mckeydees.com	Delete
18	Captain Crook	captaincrook@mckeydees.com	Delete
Jser Name: <input type="text"/> Email: <input type="text"/> Add User			

Manage Starbuck's Files: Browse, Read, Display Starbuck's Files

Browse, **Read**, Display **User-File Access (Intersection Table)**

Supported Actions: **Create** a File, **Delete** a File,

Create User-File Access using **drop down menus** for each key, View User-File Access
foreign-key with drop-down menu

Select File To Manage

[Back to Companies](#)

Current Files

File ID	File Name	Company ID	Users with Access	Actions
5	Secret Drink Menu	2	<input type="text" value="Susie Perkins"/>	Delete
6	Human Resources	2	<input type="text" value="Susie Perkins"/>	Delete

Add File

Name:

[Add File](#)

Grant Access to File

Select File:

Select User:

[Grant Access](#)

Manage Black McDonald's Forms and **Form Signatures (Intersection Table)**
Supported Actions: **Read** Forms, **Delete** Forms, **Create** Forms

Select Form To Manage

[Back to Companies](#)

Current Forms

Form ID	Form Name	File Id	Actions
1	\$1 Menu	1	Delete
2	Burger Flipper Agreement	2	Delete
3	PlayPlace Release of Liability	3	Delete

Form Signatures

Signature ID	User Name	Date Signed	Form ID
--------------	-----------	-------------	---------

[Add Form](#)

Name:

File:

Select a file

[Add Form](#)

Manage McDonald's Signatures
No Functionality except *filling out the add form*

Manage Signatures

[Back to Companies](#)

Signatures for Company:

Signature ID	Date Signed	User Name	Form Name
--------------	-------------	-----------	-----------

Add New Signature

Name:

Date Signed:

Form Name:

[Add Signature](#)