Team 28: DocuSign Integration

3 Design

3.1 Design Context

3.1.1 Broader Context

The broader context for our project is that it is set in a time when the housing market is expanding and there is a greater need for BuilderTrend's services in a socially distanced way. The communities we are designing for are people who are in the real estate market. A societal need our project addresses is the need to stay socially distanced in this pandemic.

Area	Description	Examples
Public health, safety, and welfare	This project allows for all parties to sign their legal documents electronically	Reducing Covid-19 exposure
Global, cultural, and social	It will help communication between builders, homeowners, and contractors as well as keep everyone accountable for completing their tasks in a timely manner	Implementation would reduce face to face interaction
Environmental	Reduce material usage as all documents would now be stored electronically	Decrease usage of recyclable materials
Economic	This project will be viable for BuilderTrend and will cut down on their material costs.	Product will remain affordable for BuilderTrend and allow for more accessible use of their product

3.1.2 User Needs -Justin

Builders: Builders need a way to manage, sign, and review contracts between themselves, the homeowner, and contractors. This is because a builder should facilitate the relationship between a homeowner and a contractor to complete a construction project.

Homeowners: Homeowners need a way to review and sign contracts related to the construction of their home, as well as viewing these documents at any time. This is because a homeowner will need to make decisions regarding their new home, and those decisions need to be formally agreed upon via a contract that can be accessed at any time.

Contractors: Contractors need a way to create, share, and sign documents related to their work on a house or construction project. This is because the contractor will complete specific parts of a project, and they need to receive payment according to a contract.

3.1.3 Prior Work/Solutions

For our project, there is not a lot of background/literature other than understanding what BuilderTrend does and who will be using our product. This information can be found in other sections. However, since we

will be relying on the DocuSign API for most of our project (They have multiple APIs, but we will likely be using their [eSignature API](#) since we need to capture actual signatures). There are many, many alternatives to DocuSign, but many of them are targeted towards smaller businesses. A few examples of these are HelloSign and PandaDoc HelloSign's specific advantages are it is easier to set up, it offers more features in its free plan than DocuSign, and supports in-person signatures.. DocuSign's advantage is it is targeted for large commercial companies who need to process many signatures and documents (Similar to what BuilderTrend needs). DocuSign also offers a mobile app in its paid tiers. Below is a feature comparison chart taken from competitor [PandaDoc's website](#).

	DocuSign	PandaDoc
Legally-binding eSignatures with Audit Trail	100	Unlimited plans
Templates	✓	✓
Activities Notifications	✓	✓
Payments Collection	✓	All plans
Automated Workflows	$	✓
Custom Branding for Documents, Templates, and Emails	$	✓
CRM Integrations	$	✓
eSignature API (including Sandbox API)	✓	✓
Document Generation and Editing	X	✓
Document Analytics and Insights	X	✓
Free trial	✓	✓
24/7 customer service	$	✓

It is also important to note that BuilderTrend explicitly stated that they want us to use DocuSign, so we will not be using a competitor. As far as following previous work, we will need to integrate our code into their pre-existing platform. We do not have access to their code base right now, so we do not know exactly what that will look like.

3.1.4 Technical Complexity

- All group members are software engineering majors. Thus the technical complexity is limited to that field, but that is an appropriate scope given the experience of the team members.

- The specific technical difficulty of this project comes from BuilderTrend's need for Docusign to be integrated into multiple parts of the system. What follows are some examples.
  - Conditional Event Triggers. Website events can only be triggered after a document has been properly signed by all necessary parties.
  - Multiple file formats.
  - Single or multiple parties may need to sign one document, perhaps multiple devices.
  - Documentation needs to be written for continued use of Docusign by BuilderTrend.
  - Page integrity. Will reloading or losing connection disrupt the state of a document.
  -

3.2 Design Exploration

3.2.1 Design Decisions

- DocuSign has multiple APIs: eSignature, Click, Rooms. Each one caters to a specific need. We need to decide which ones suit the project the best, or if we should make use of more than one.
- BuilderTrend has multiple places on their website where Docusign could be of use. Would it be better to add docusign into one place and then generalize it, or to attempt to make a plug and play solution for all possible locations.
- Does DocuSign work best as a website element, or would it work better if we made an endpoint for distributing a DocuSign document across emails.

3.2.2 Ideation

- "Does DocuSign work best as a website element, or would it work better if we made an endpoint for distributing a DocuSign document across emails?"
- We identified options empathising with end users. Considering that this solution is meant to make the lives of both contractors and clients easier.
- Potential Options:
  - Exclusively Email
  - Exclusively website elements.
  - Website elements exist for single party signatures while any multi party documents are distributed over email.
  - Allow partial completion of documents by available parties and then distribute to remaining parties via email.
  - Associating documents with accounts, distributed over accounts instead of emails regardless of the number of parties involved.

3.2.3 Decision-Making and Trade-Off

- Exclusively Email
  - Pros:
    - Simple to implement
    - Logistically simple.
  - Cons:
    - Requires all parties to sign into email and access the document even if they are all together in person.
    - Possible confusion and missed emails.
- **Exclusively website elements. - Chosen Option.**

- ○ **Chosen as it is most in line with what our BuilderTrend contact has requested.**
  - ○ Pros:
    - ■ Easy to access.
    - ■ All present parties can sign immediately.
    - ■ Links can be shared over any platform.
  - ○ Cons:
    - ■ Page persistence.
    - ■ How to work with parties that are not present
- Website elements exist for single party signatures while any multi party documents are distributed over email.
  - ○ Pros:
    - ■ Simple for the end user.
    - ■ Makes collaboration accessible while making single party signatures quick.
  - ○ Cons:
    - ■ Harder to implement.
- Allow partial completion of documents by available parties and then distribute to remaining parties via email.
  - ○ Pros:
    - ■ Best of both worlds
  - ○ Cons:
    - ■ Not sure if possible with the available DocuSign APIs.
- Associating documents with accounts, distributed over accounts instead of emails regardless of the number of parties involved.
  - ○ Pros:
    - ■ Simple
    - ■ Secure
    - ■ Organized
  - ○ Cons:
    - ■ Hardest solution to implement as it requires deeper integration into BuilderTrends existing website and databases.
    - ■ Not feasible within the time frame.

3.3 Proposed Design

3.3.1 Design Visual and Description

Breakdown of the Basic System:

Component Diagram:
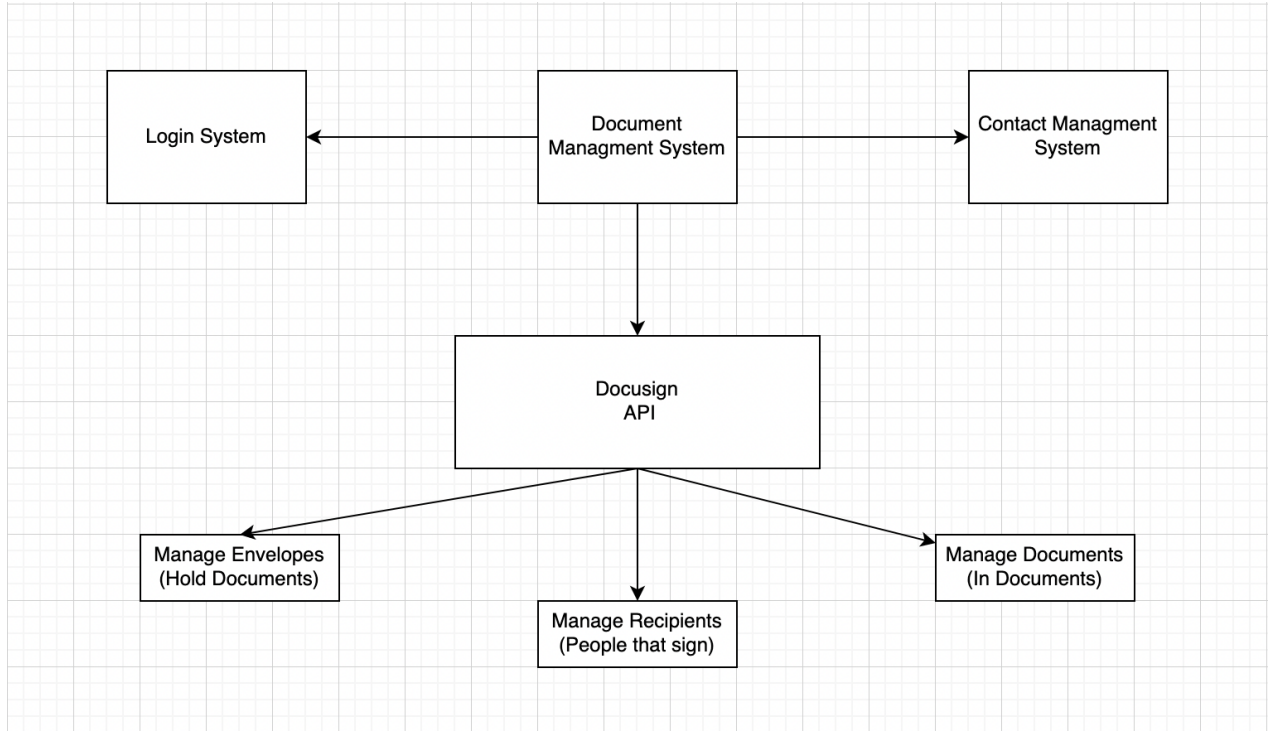
Overall:

This system is built so that it meets the basic Buildertrend Requirements. The details of the system have been broken down into 3 main systems, login system, document management system, and the contact management system.

Login:

While the project isn't solely based on developing a login system, we will need it to manage client accounts with Docusign and how people get documents to sign. This will be as basic as having a simple login/logout section and account preferences (for email notification preferences).

Document Management:

The document management system will take care of uploading/downloading documents, and will serve as a place to get documents into Docusign to start the signing processes. At a basic starting level, we will learn how to upload these documents, and then on a more advanced level of the project we will have documents be contained and sorted by project/customer/client/status. These details of the advanced state of this system will be determined when the project reaches a state in which the basic Docusign API details have been hashed out.

Contact Management:

The contact management system will serve as an interface that displays current signature status on a document from each contact involved in the process. It will also allow administrators to easily contact the involved parties and update them on the status of the

document. The overall goal of this branch is to allow for easy communication between the groups involved.

3.3.2 Functionality -Justin

Our design is a proof-of-concept web application that is accessible to BuilderTrend at any time. This app, combined with our extensive documentation, will provide BuilderTrend a clear path forward to fully integrate DocuSign with their entire platform. We intend to recommend implementations based on their business needs, as well as demonstrate how that implementation might work within our own web app.

This design should satisfy all of our functional requirements, since we can use the DocuSign API to clearly demo each use case that BuilderTrend is considering.

Our app can be implemented with little to no cost, and can satisfy non-fun ctional requirements such as speed and security. However, these don't necessarily translate to the non-functional requirements of BuilderTrend's final implementation.

3.3.3 Areas of Concern and Development -Justin + Alan

Our primary concern with our current design is security. Considering we are dealing with confidential contracts and information, we should prioritize security over accessibility and usability. Our current design utilizes the DocuSign API, which is known for its security, but we need to make sure all parts of our web app are secure. Another concern of ours is we haven't been able to confirm that Docusign's API will be able to meet the requirements presented by Buildertrend based on our current depth of research. If we reach a point where we have decided that Docusign does not meet Buildertrend's requirements, we will have to resort to other options for implementing the use cases given to us by Docusign. Our immediate plans for developing solutions to these concerns are to do lots of research on how to make secure web applications. We will not be afraid to also consult engineers at DocuSign if we feel we are stuck. As for the API limit, that will just be something we will know more about down the road.