# Electronic Signatures Service

|  |  |
| --- | --- |
| **Target release** | 11/27/2020 |
| **Epic** |  |
| **Document status** | DRAFT |
| **Document owner** | [Scott Rosenberg](https://loffacorp.atlassian.net/wiki/people/5d5c918c41ab580c26f89795?ref=confluence)[Israel Perez](https://loffacorp.atlassian.net/wiki/people/557058:56e07ded-07b5-4bca-a8b1-662cca9fc56e?ref=confluence) |
| **Designer** | [Israel Perez](https://loffacorp.atlassian.net/wiki/people/557058:56e07ded-07b5-4bca-a8b1-662cca9fc56e?ref=confluence) [Scott Rosenberg](https://loffacorp.atlassian.net/wiki/people/5d5c918c41ab580c26f89795?ref=confluence) |
| **Tech lead** | [Yogesh Doke](https://loffacorp.atlassian.net/wiki/people/5f8dac3a9c31840076d1d1ba?ref=confluence) |
| **Technical oversight** | [Israel Perez](https://loffacorp.atlassian.net/wiki/people/557058:56e07ded-07b5-4bca-a8b1-662cca9fc56e?ref=confluence) |
| **QA** | [Sagar Ambi](https://loffacorp.atlassian.net/wiki/people/5f65091acacd83007786fbfc?ref=confluence) |

## Objective

Currently, all Loffa products send out documents via email as PDF attachments. The goal is to allow for the option to send via a URL to then access that document in a web application. The customer can then electronically sign the PDF and submit the form. The respective Loffa system will then update the response from the customer within its transactional database.

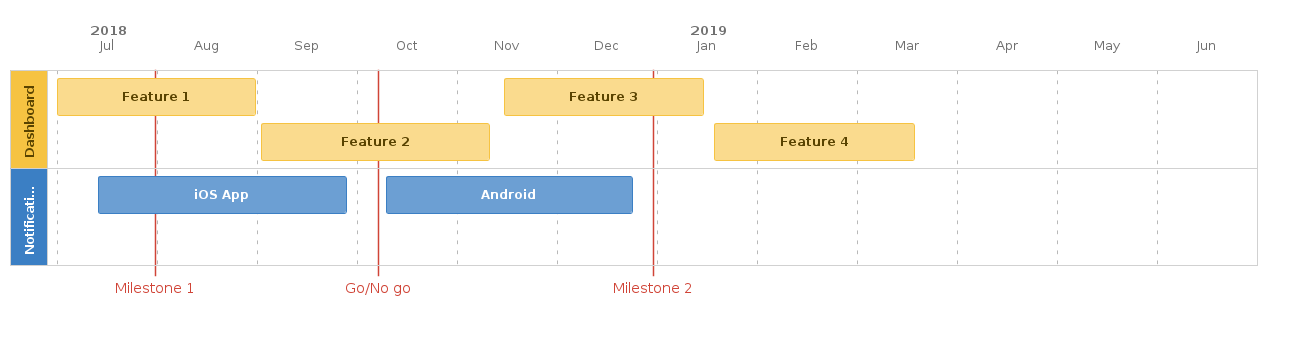
The objective is to develop a service that can work with all Loffa products and allow for sending a data-driven fillable PDF to specific email addresses in the form of a URL that will then take the client to a webpage that allows them to view and ‘e-sign’ the PDF.

## Success metrics

|  |  |
| --- | --- |
| **Goal** | **Metric** |
| Have a microservice that allow us to send Esig documents | All current applications should be able to use the microservice, that is: send the document and receive updates of it. |
| Send Esig document to Recipients that can be signed and responded back. | Recipients that are not part of the Loffa Network should be able to receive ESig Documents, Interact with them and respond them in a dynamic and secure wat. |

## Assumptions

## Milestones



## Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Requirement** | **User Story** | **Importance** | **Jira Issue** | **Notes** |
| 1 | All Products should be able to send E-Sig documents | As an Administrator User I want to be able to configure if email documents are sent out using Esig or normal email so that I can decide how my recipients receive this documents | HIGH |  |  |
| 2 | All Products should be able to send E-Sig documents | As a Developer I want to have an API where to send PDFs for Esig so that I can provide with this option in my application |  |  |  |
| 3 | All Products should be able to send E-Sig documents | As a Developer I want to have an API that notify my application when an event happened to the Esig document that I previously sent so that I can update the application item information accordingly. |  |  |  |
| 4 | All Products should be able to send E-Sig documents | As a Developer I want to have an API that allows me to indicate what controls the Esig document that I send will have so that I can configure the input parameters |  |  |  |
| 5 | All Products should be able to send E-Sig documents | As a Developer I want to have an API that allows me to indicate the position of controls in the Esig document that I send out so that I can create predefined configurations for them |  |  |  |
| 6 | Recipient should get a ESignature document with the ability to read it, sign it and respond it in a secure way. | As a Recipient I want to receive my Esig document via email so that I can read it and determine my respond. |  |  |  |
| 7 | Recipient should get a ESignature document with the ability to read it, sign it and respond it in a secure way. | As a Recipient I want to be able to sign the document using hand written signature or type signature so that my counter party can confirmed that I signed the document. |  |  |  |
| 8 | Recipient should get a ESignature document with the ability to read it, sign it and respond it in a secure way. | As a Recipient I want to be able to access the Esig document using a link to a secure portal so that I can read it, sign it and/or respond it. |  |  |  |
| 9 | Esignature Microservice should keep record of all transactions | As an IT user I want to be able to audit the microservice so that I can troubleshoot in case of an incident. |  |  |  |
| 10 | Microservice should follow TDD, DDD and REST API principles | As a Developer I want to unit test the important business logic parts so that I can deliver a service with less bugs |  |  |  |
| 11 | Microservice should follow TDD, DDD and REST API principles | As a Developer I want to develop the service using DDD so that I can keep the business logic encapsulated and decouple so it becomes more easy to maintain |  |  |  |
| 12 | Microservice should follow TDD, DDD and REST API principles | As a Developer I want to develop the server using REST API principles and follow the Richardson Maturity levels (<https://restfulapi.net/richardson-maturity-model/>) so that the API becomes more self explanatory. |  |  |  |

## User interaction and design

* Solution Structure
  + repository
  + Storage: Database
  + Sql Local
  + Code first
  + Save document : TBA
  + database
  + file system
  + Azure
* Rest API:
  + End Points
  + Queries
  + Document information
  + Document Status
  + Document History
  + Commands
  + Developer options
  + Upload Document
  + Add controls to documents
  + Add Responses to Documents
  + Send and Resend
  + Freeze document to only allow Recipient Interaction
  + Recipient Options
  + Update Control Value
  + Update Document Status
* AMQP ( Async Messages )
  + Azure Service Bus
  + RabbitMQ
  + Endpoint subscriber--
* References :
  + <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/>
  + <https://github.com/dotnet-architecture/eShopOnContainers>
* Send email with link to recipient
* PDF Visualizer ( secure portal )
  + Secure PDF visualizer
  + PDF.JS for front end
  + ASPOSE PDF for backend
  + Pdf viewer with controls
  + Interactive controls
  + Response options
  + Link validation
* Integration with Current Applications
  + Administration user configuration
  + Service connection
  + Send pdf email
  + receive update statuses

## Service API / Applications interaction mockup

## Open Questions

|  |  |  |
| --- | --- | --- |
| **Question** | **Answer** | **Date Answered** |
|  |  |  |

## Out of Scope