UNIVERSITY
OF WOLLONGONG
AUSTRALIA

CSCI321 - Final Year Project:
Secure Remote Authentication using Visual Cryptography

TECHNICAL MANUAL

Version 1.4

Presented by: **Team SS19/1C**			
	UOW ID	**Name**	**Email Address**
1	5710741	Amit Singh Hundal	hundal004@mymail.sim.edu.sg
2	5710972	Erwin Leonardy Musa	elm001@mymail.sim.edu.sg
3	5710996	Harpreet Singh Kang	kang019@mymail.sim.edu.sg
4	5711642	Yeo Wei Wen Matthew	wwmyeo001@mymail.sim.edu.sg

Revision History 4

Executive Summary 5

Introduction 5

Mission 5

Vision 5

Document Objectives 5

Introduction 5

Literature Review 5

Project Scope 5

Requirement Specification 6

Design Specification 6

Project Proposal 6

Development Methodology 6

Risk Assessment 6

Project Website 6

Roles and Responsibilities 8

Existing Applications 9

CASE STUDY 1: Double Helix Industries Visual Cryptography 9

CASE STUDY 2: BotDetect Captcha 9

CASE STUDY 3: VisualCrypto 10

Feature Gathering 10

Application Comparison 12

Literature Review 13

Understanding Visual Cryptography 13

What is Encryption? 13

How does Encryption work? 13

What is Visual Cryptography? 13

How does it differ from other forms of Cryptography? 13

What are the different Visual Cryptographic methods available? 13

How these methods are implemented? 13

What are the key aspects of visual cryptography? 14

Project Scope 15

Background 15

Description 15

Deliverables 15

Constraints and Exclusions 16

Assumptions and Dependencies 16

Operating Environment 16

Stakeholders 16

Bank Employee *16*

Bank Customer *16*

Project Build Team 16

Project Proposal **17**

Business Model 17

Problems *18*

Target *18*

Solution *18*

Reaching and acquiring customers *18*

Revenue generation 18

Target User Domain 18

Product Validation 18

Requirement Specification **19**

Interfaces 19

User Interface *19*

Hardware Interface *19*

Software Interface *19*

Communication Interface 19

Functional Requirements 19

Provide Share of Signature *19*

Drop-in Signature for Encryption *19*

Create Shares *20*

Generate Key Image *20*

Resize Image *20*

Send Shares to Customer & Bank *20*

Add Share to Document/Cheque *21*

Remove Noise 21

Non-Functional Requirements 21

Performance Requirements *21*

Safety Requirements *21*

Security Requirements *21*

Software Quality Attributes *22*

Other Requirements 22

Design Specification **23**

Product Architecture 23

Use Case Diagram 23

Use Case Textual 24

Class Diagram 26

Sequence Diagrams 27

Provide Share of Signature (1.0) 27

Drop-in Signature for Encryption (2.1) 27

Create Shares (3.1) 28

Generate Key Image (3.1.1) 28

Resize Image (3.1.2) 29

Email Shares to Customer and Bank (3.2) 29

Add Share to Document/Cheque (3.3) 30

Remove Noise (3.3.1) 30

Activity Diagrams 31

Create Shares (3.1) 31

Send Shares to Customer and Bank (3.2) 32

Add Share to Document/Cheque (3.3) 32

Development Methodology 34

Waterfall Model 34

Project Timeline 35

Development Tools 35

Programming Language and Associated Libraries 35

Web Development Tools and Libraries 35

Version Control 35

Risk Assessment 36

Revision History

Name	Date	Reason for changes	Version
Amit, Erwin, Harpreet, Matthew	18 March 2019	Initial version of the Technical Manual	1.1
Amit, Harpreet, Matthew	29 March 2019	Updated Technical Manual to include required information and Class, Use Case, Activity and Sequence Diagrams	1.2
Amit, Harpreet, Matthew	2 April 2019	Update Formatting and Layout of Technical Manual, Risk Assessment, Waterfall Model, Development Tools	1.3
Amit, Erwin, Harpreet, Matthew	4 April 2019	Updating of Technical Manual, Product Architecture, Market Research	1.4

Executive Summary

The basis of our project is to define a secure authentication mechanisms for users to access remotely, through the use of visual cryptography. To achieve this, we have devised a system that operates on the basis of user signatures, where the signatures will be used to authenticate a legal document. Through an online platform, we allow users to provide an image of a signature, from which two shares that appear as random noise emerge. One of the shares is to be kept by the user, while the other is sent to a party that requires authentication. In order to provide the authentication, the party requiring authentication on the legal document sends in their complete document, inclusive of the share, while the recipient will in turn provide their share to complete the authentication.

For our final product, VSignIT, our primary focus will be electronically authenticating signatures on cheques. However, while the focus of the project is in the banking domain, the usage of the application can be generalized to serve as an authentication for any legal document requiring signatures within an organisation.

Introduction

Mission

To inspire trust in our systems ability to remotely authenticate documents and to provide our community with the ease of mind knowing that their documents are secure going through our system, despite having to traverse through an online portal.

Vision

Apply visual cryptography that provides users an authentication mechanism that is both remote and seamless.

Document Objectives

Introduction

To provide background information on the context of the project, as well as to provide the structure of the documentation. This section serves to provide information on how the workload was distributed amongst the group members, the roles each member holds, and the project website, where all the deliverables of the project will be hosted.

Literature Review

This section serves to provide an understanding of the fundamentals of encryption and more specifically, how visual cryptography functions. This section is for individuals who do not have an understanding of the stated concepts, and can be overlooked by parties that already have knowledge of these concepts.

Project Scope

This section provides details on our product, the research on the market domain, as well as the stakeholders involved in our product, currently specifically catered towards the banking domain.

Requirement Specification

This section covers the interfaces involved in the product, functional requirements, non-functional requirements, and any other requirements that are to be included into the final product.

Design Specification

This portion includes the relevant diagramming necessary for the development of the product. It includes the use case diagram, class diagram, sequence diagrams and activity diagrams.

Project Proposal

This section encompasses how the product is to be marketed to potential customers. A business proposal is provided.

Development Methodology

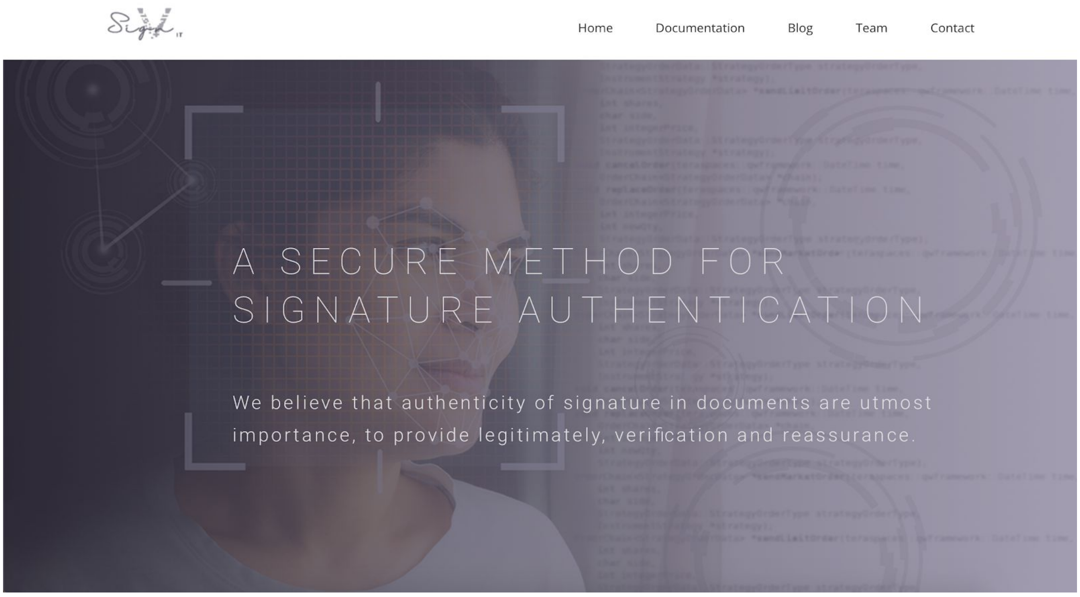
This section covers the methodology used to develop this project.

Risk Assessment

This section covers the risks involved in the development of the product.

Project Website

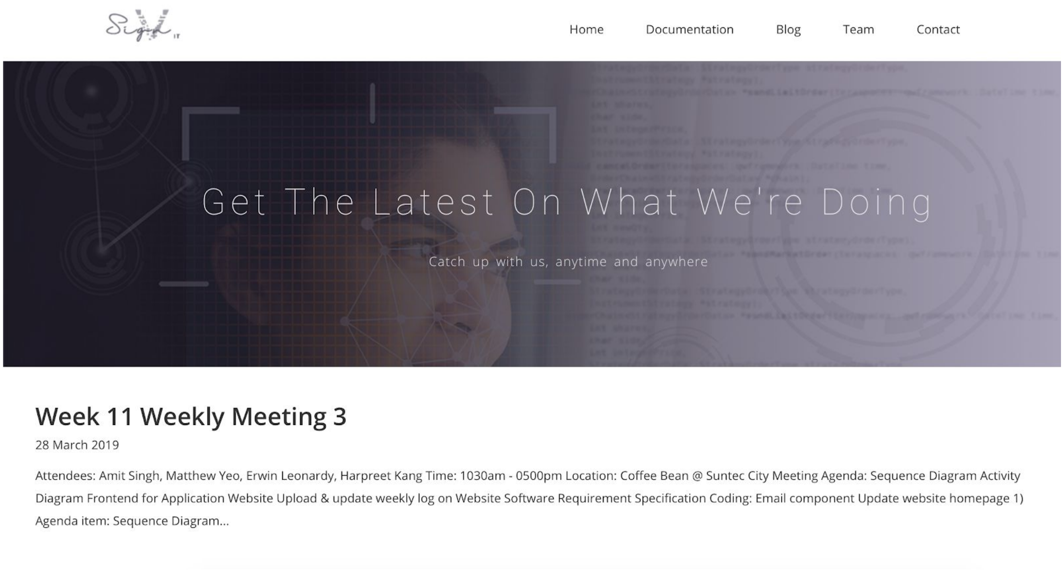
This website serves as a medium for the prospective users to know more about our product and it also provides answers to the questions of our current users. This website would also provide images and video tutorials of the application in the future. It also provides all the documentation and deliverables necessary for the project.



A SECURE METHOD FOR SIGNATURE AUTHENTICATION

We believe that authenticity of signature in documents are utmost importance, to provide legitimately, verification and reassurance.

Figure 1: Homepage of Website



Get The Latest On What We're Doing

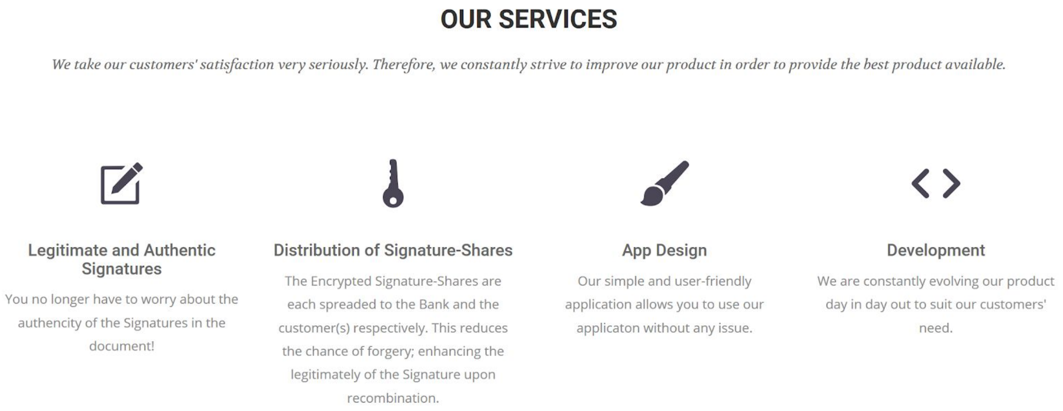
Catch up with us, anytime and anywhere

Week 11 Weekly Meeting 3

28 March 2019

Attendees: Amit Singh, Matthew Yeo, Erwin Leonardy, Harpreet Kang Time: 1030am - 0500pm Location: Coffee Bean @ Suntec City Meeting Agenda: Sequence Diagram Activity Diagram Frontend for Application Website Upload & update weekly log on Website Software Requirement Specification Coding: Email component Update website homepage 1) Agenda item: Sequence Diagram...

Figure 2: Website Blog Page



OUR SERVICES

We take our customers' satisfaction very seriously. Therefore, we constantly strive to improve our product in order to provide the best product available.

Legitimate and Authentic Signatures

You no longer have to worry about the authencity of the Signatures in the document!

Distribution of Signature-Shares

The Encrypted Signature-Shares are each spreaded to the Bank and the customer(s) respectively. This reduces the chance of forgery; enhancing the legitimately of the Signature upon recombination.

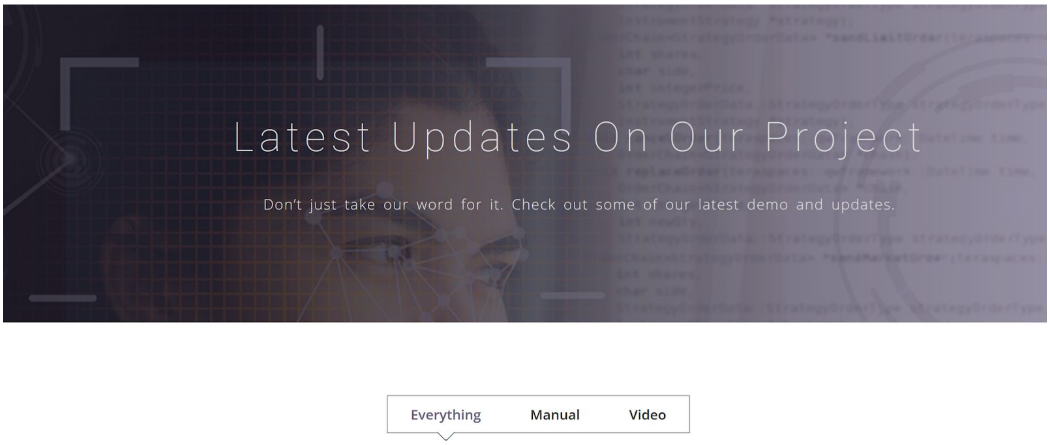
App Design

Our simple and user-friendly application allows you to use our applicaton without any issue.

Development

We are constantly evolving our product day in day out to suit our customers' need.

Figure 3: Service Specification



Latest Updates On Our Project

Don't just take our word for it. Check out some of our latest demo and updates.

Figure 4: Documentation Page

Roles and Responsibilities

	Amit Singh *Designer*	**Erwin Leonardy** *Team Leader*	**Harpreet Singh** *Programmer*	**Matthew Yeo** *Tester*
Background Research	✓	✓	✓	✓
Idea generation	✓	✓	✓	✓
Gathering requirements	✓	✓	✓	✓
Project Diary				✓
Meeting Minutes			✓	✓
Website		✓		✓
Project Proposal	✓	✓	✓	✓
Project Timeline	✓			
SRS	✓	✓	✓	✓
Diagramming	✓	✓	✓	✓
Product Design	✓	✓	✓	✓
Technical Design Manual	✓	✓	✓	✓
User Manual		✓		
Code Implementation		✓		
Testing Implementation	✓	✓	✓	✓
Videography		✓		

Existing Applications

CASE STUDY 1: Double Helix Industries Visual Cryptography

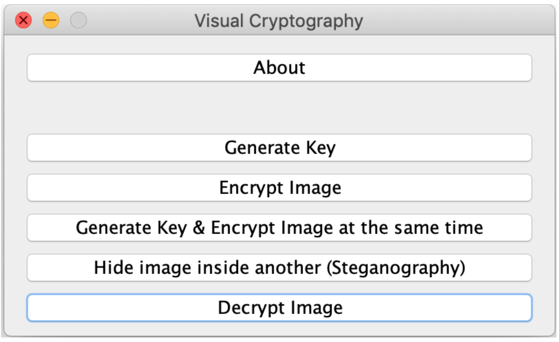
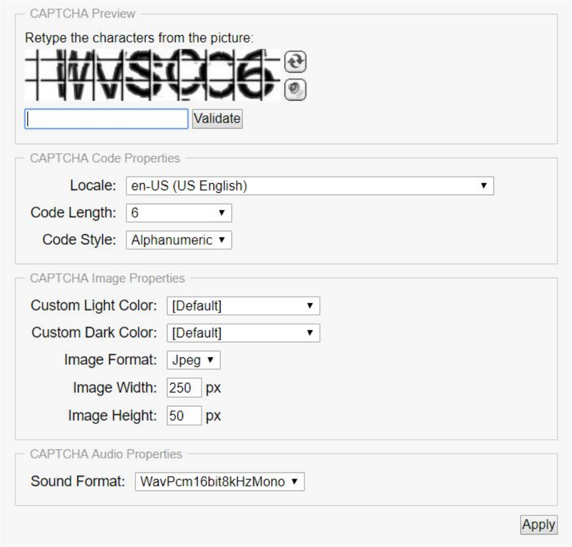

Visual Cryptography
About
Generate Key
Encrypt Image
Generate Key & Encrypt Image at the same time
Hide image inside another (Steganography)
Decrypt Image

Figure 5: Double Helix Industries

The Double Helix Industries desktop visual cryptography provides key generation, image encryption and decryption with the appropriate supplied images. It generates a key image, and creates shares with the key image and an image supplied by the user. The key and and shares are both stored locally. Decryption is carried out by inputting key image and encrypted image. It also provides steganography by allowing for images to be hidden within other images.

CASE STUDY 2: BotDetect Captcha


CAPTCHA Preview

Retype the characters from the picture:

CAPTCHA Code Properties

Locale: en-US (US English)

Code Length: 6

Code Style: Alphanumeric

CAPTCHA Image Properties

Custom Light Color: [Default]

Custom Dark Color: [Default]

Image Format: Jpeg

Image Width: 250 px

Image Height: 50 px

CAPTCHA Audio Properties

Sound Format: WavPcm16bit8kHzMono

Figure 6: BotDetect Captcha

https://captcha.com/demos/features/captcha-demo.aspx

Captcha is a program used to distinguish between human users and machines. It is used as method of preventing spam and DOS. BotDetect is a fully customizable program that allows us to modify the parameters and combinations of the captcha.

CASE STUDY 3: VisualCrypto

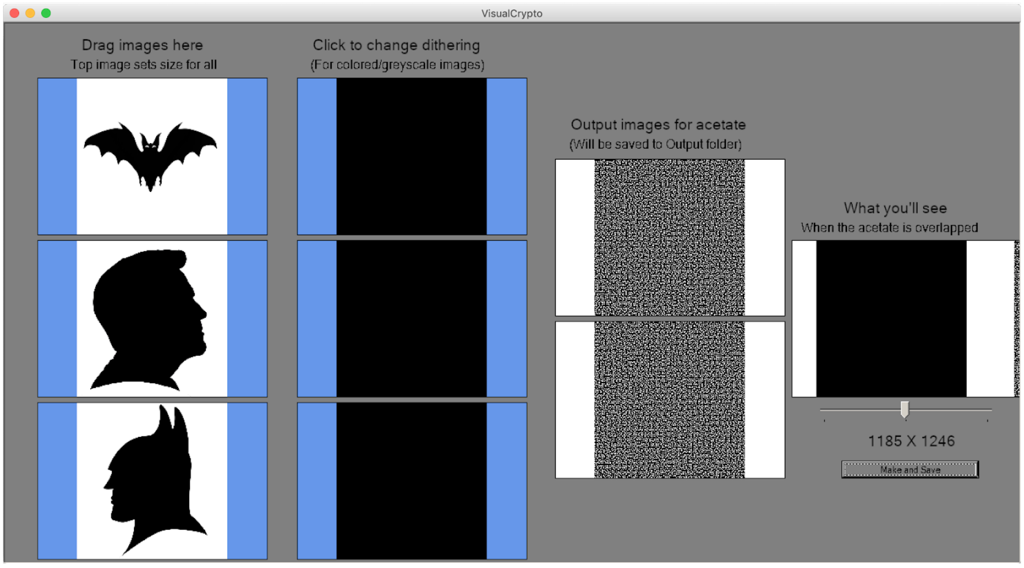

VisualCrypto
Drag images here
Top image sets size for all
Click to change dithering
(For colored/greyscale images)
Output images for acetate
(Will be saved to Output folder)
What you'll see
When the acetate is overlapped
1185 X 1246
Make and Save

Figure 7: VisualCrypto

VisualCrypto is an implementation of visual cryptography which provides a detailed interface that explains each step of the cryptographic process. It begins with providing images to create the key image and the encrypted image. It allows the user to contrast the RGB elements for colored image and respectively shows the shares created, and the result of overlapping these shares. It also provides the user to create the shares such that the output of the decryption will be of a specific size.

Feature Gathering

Based on the existing visual cryptographic applications that were sourced, inclusive of the case studies shown above, the following is the list of features gathered:

- Generate Key Image (with a given dimension)
- Encrypt and generate N shares
- Decrypt N shares with any k shares
- Steganography
- Add watermark
- Send shares to different recipients
- Resize the image
- Additional file support (PNG, JPG, etc)
- Remove noise (Threshold filtering)
- Dithering (RGB parts of coloured or grayscale images) / convert to black/white
- GPU encryption/decryption
- Print on transparencies

Based on the features gathered, the next step to perform before carrying out the ideation process of the project was to define the features that were to be implemented within the system to provide an edge in comparison to the case studies that were gathered. Figure 8, as shown below, illustrates the features that were selected to be included in the system.

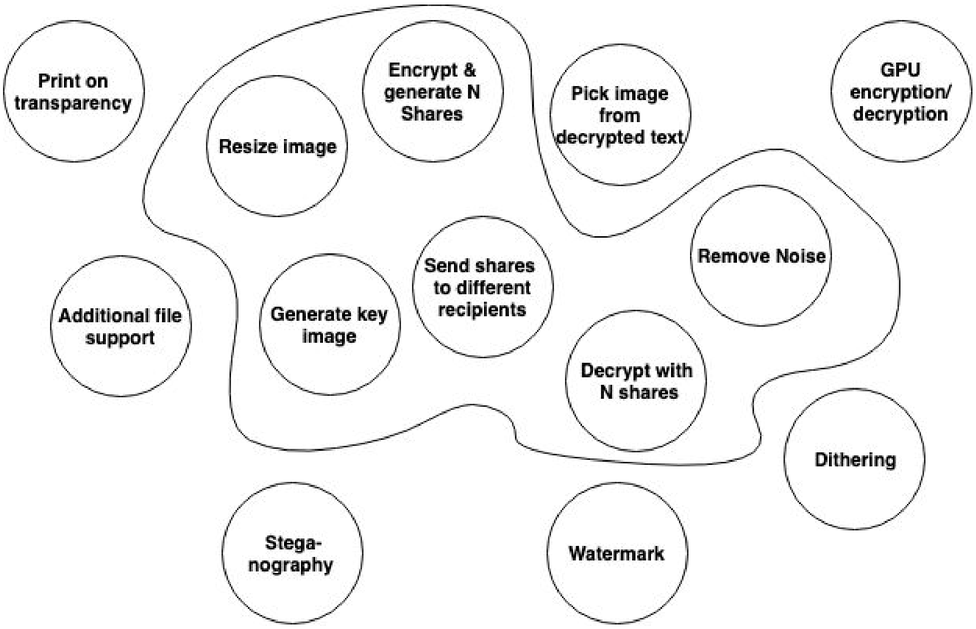


Figure 8: mind map of features gathered and selected

Based on the features selected, they were then further classified to provide a clearer image of how they would correspond with each other. The two main features presented, as shown in figure 9 below, are decryption with N shares (no less than N), and encryption with generation of N shares. Their sub-features are shown as follow.

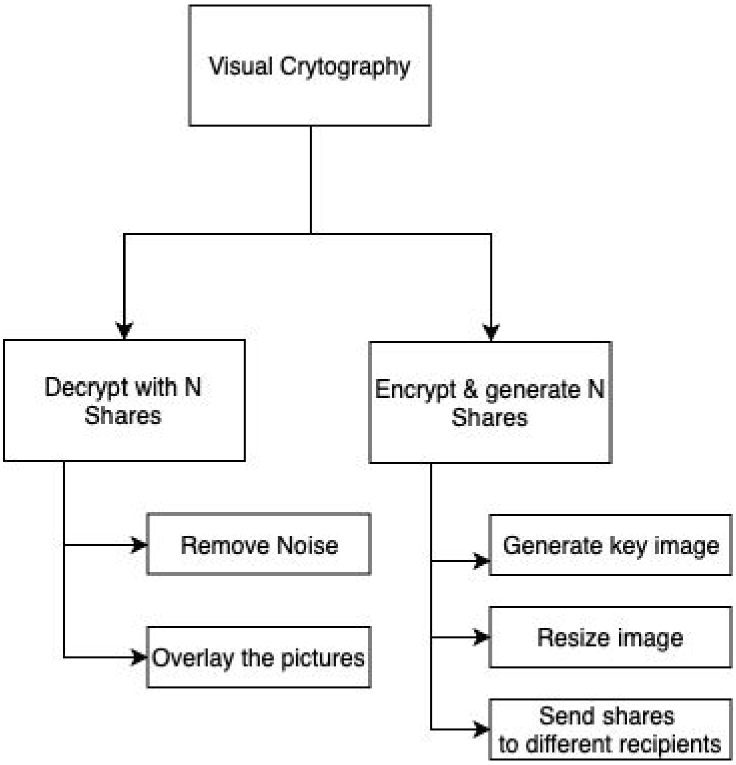


Figure 9: Application Chart

Application Comparison

Features	Double Helix Bridge	BotDetect Captcha	VisualCrypto	Our Product
Open Source	✓	✓	✖	✖
Easy to use	✓	✓	✖	✓
Generate Key Image (with a given dimension)	✓	✖	✓	✓
Encrypt and generate N shares	✓	✖	✖	✓
Decrypt N shares with any k shares	✖	✖	✖	✖
Steganography	✓	✖	✖	✖
Add watermark	✖	✖	✖	✖
Send shares to different recipients	✖	✖	✖	✓
Resize the image you want to encrypt	✖	✓	✓	✓
Additional file support (PNG, JPG, etc)	✖	✓	✖	✓
Remove noise (Threshold filtering)	✓	✖	✖	✓
GPU encryption/decryption	✖	✖	✖	✖
Dithering	✖	✖	✓	✖
Mandatory n shares decryption	✖	✖	✖	✖

Literature Review

Understanding Visual Cryptography

What is Encryption?

Encryption is the process where by a message (plaintext) is encoded (scrambled and unintelligible) to become an encrypted message (ciphertext). Decryption is the opposite, where the ciphertext is unscrambled to recover the plaintext. The purpose of encryption is to prevent an unauthorised person/s from reading the message.

How does Encryption work?

Encryption makes use of an algorithm to scramble/encrypt the plaintext to transform it into a ciphertext. This can be described as using a key to lock up the message, the message can then be unlocked or decrypted by using the key to transform it back into a plaintext. There are 2 types of encryption, symmetric key and public key cryptography.

- Symmetric Key: It works with only one key, the same key is used to encrypt and decrypt a message.
- Asymmetric (or Public) Key: Each user has a set of public and private keys. If A wants to send a message to B, A will take B's public key and encrypt the message by combing A's private key and B's public key. B will then decrypt the message by using their private key.

What is Visual Cryptography?

Visual Cryptography is a technique that allows information (images, text, diagrams..) to be encrypted using an encoding system that can be decrypted by the eyes. No computer required to decode.

How does it differ from other forms of Cryptography?

As opposed to other forms of cryptography, visual cryptography requires the participation of the user. Decryption of the shares provide information that is only perceived visually, thus deterring automated attacks. It is akin to One Time Pad (OTP) that provides perfect secrecy, this is due to the incomplete information the attacker has to break the cipher.

What are the different Visual Cryptographic methods available?

- Halftone Visual Cryptography
- Colour Visual Cryptography
- General Access Structure Visual Cryptography
- Random Grid Based Visual Cryptography
- Extended Visual Cryptography
- Hierarchical Visual Cryptography [1]

How these methods are implemented?

Halftone Visual Cryptography

The secret image is encoded into halftone shares. This technique uses blue noise halftoning principle. Visual quality of obtained halftone shares are better than other methods.

Colour Visual Cryptography

Secret colour image hides itself in two arbitrary colour images, this can be constructed and then kept by 2 participants, separately. The processed image is known as camouflage image.

General Access Structure Visual Cryptography
Uses optimisation technique. Improves visual quality of worse image; no need for codebook or basis metrics as well as it reduces the pixel expansion problem. The recovered image has better display quality than original image.

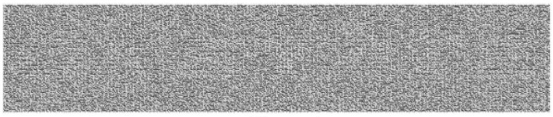
Random Grid Based Visual Cryptography
Probable allocation method to produce the best contrast in the share image as well as the stack images. The size of revealed image is the same as original secret image. This is highly secure because of the randomness.

Extended Visual Cryptography
This is being applied for colour image which uses VIP synchronisation and error diffusion for visual quality improvement. It uses artificial bee colony algorithm where halftoning process is applied over the colour image, then embedded process is applied.

Hierarchical Visual Cryptography
Encrypts the secret images in to levels. It hides the secret information into number of levels. Expansion ratio is 1:4.

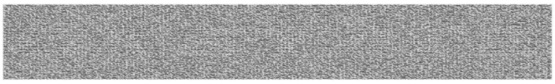
What are the key aspects of visual cryptography?

Each pixel in an image is divided into consistent blocks. It always has the same number of black and white subpixels. If the same pixel is divided into 4 blocks, 2 of them would be white and the other 2 would be black. The same applies for pixels with 2 subpixels.



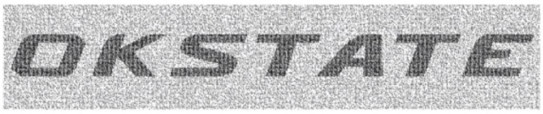
Share 1

Figure 10: Share 1



Share 2

Figure 11: Share 2



Overlay (Share 1 + Share 2)

Figure 12: Decryption of shares

In the examples on top, we have demonstrated how Visual Cryptography works. It would reveal the secret image when we stack two of the seemingly random shares on top of each other.

Project Scope

Background

With this project, our aim is to provide an authentication mechanism which allows users to sign documents in both a secure and seamless manner. Physical signatures introduce limitations when introduced to the digital world. Their origin is difficult to verify, i.e we do not know if the actual person has signed the document.

The reason why this scheme is superior to digital signature (Public Key Signature) is because there is a need to have a trusted third party in the scheme. The trusted third party may abuse the right bestowed to them by the users. In this scheme, the original document is held by the user and the shares are disseminated to the shareholders.

Currently in a bank, when an authorised signatory is created, they must first physically verify their signature with either a bank staff, lawyer or public notary. The bank keeps this signature for future verification purposes. When a client wants to sign a document for the bank, they will sign the original document, scan it and email it to the bank. The banker will now have to physically verify the documents signature with the stored signature from the database; this is when visual cryptography kicks in. When a client wishes to sign a check, they must provide the bank with a physically signed "wet-ink" check. The banks do not allow digital checks.

With our visual cryptographic signature scheme, we aim improve the life of the clients by allowing them to use digitally signed documents and checks. The banks have a streamlined authentication process and have the added assurance that the signatures are legitimate.

Description

Our proposed solution is to re-innovate the signing of documents with visual cryptography. When a customer create an account with the bank, they will also create a copy of their signature. The bank staff facilitating this procedure will input the customer's signature into our program to create 2 shares. 1 share would be kept by the bank and the other given to the customer.

When the customer wishes to sign a document or check, all they would have to do is use our program to "sign" their document using their signature share. The bank would then be able to verify the authenticity of the signature by using out program to overlay the 2 shares. If the signature is visible the signature can be considered authenticated, if not then they would have to inform to the customer.

Deliverables

1. Project Diary
2. User Manual and Technical Design Manual
3. Project Website
4. Prototype
5. Final Product
6. Video Demonstration of Product

Constraints and Exclusions

For confidentiality reasons, we cannot store the original documents of the clients in our database. This includes original signatures, signature shares and documents.

A dedicated space on documents must be allocated for signature purposes. As multiple parties will be using the signature space, documents must have a reserved location for the signature visual cryptographic functions to take place.

Assumptions and Dependencies

Our main assumption is that or program will be a tack-on feature. The login and shares storage will be done by the bank. We will depend on the bank to do the user authentication, the banks staff will have to verify the customers identity. Our program is the middle man that creates the signature shares and signs the documents and checks.

Our program will fit into the Cheque Truncation System (CTS) currently used by banks. The CTS digitises physical cheques to increase processing efficiency. Using our program in addition to the CTS, we are able to fully use cheques digitally.

Operating Environment

The program will be developed as a web application, both bank staff and their customer will be able to access the web application. The Program will be written in Python so as to facilitate the web application services. We will be using Flask as a micro web framework to run the program. Javascript will be used to build the front end of the web application.

Stakeholders

Bank Employee

These entities will be responsible for communicating with the system to produce the shares for authentication. Through the database held by the bank, these parties will have to submit the customers signature to allow the system to create the appropriate shares. One share will be sent back to be stored in the banks database, while the other share is emailed to the respective customer.

They will also provide the share kept within their database to the system when the customer sends them a legal document for verification. The system takes their share and reconstructs the final signature, from where they are to verify the legitimacy of the signature visually.

Bank Customer

These entities are responsible for providing the documents they wish to authenticate, together with the share that was provided to them to fulfill the authentication. Once provided to the system, the document will be returned to the user with the imprint of their share. Their next responsibility is to send the bank the image that was returned to them, on their own time and discretion.

Project Build Team

These entities are responsible for the development of the system. Their task is to provide a system that works seamlessly and does not compromise the security of the documents being handled within the application. After the deployment of the system, upon the realisation of any security loopholes, their responsibility is to ensure that fixes are provided in a timely fashion.

Project Proposal

Business Model

Key Partners	➢ Nil
Key Activities	➢ Developing a web application to implement a secure method to sign a document by using visual cryptography.
Key Resources	➢ Hosting platform (Servers) ➢ Application
Value Propositions	➢ Ensures authentication of the signature by requiring the clients to provide their shares ➢ Provides another layer of integrity of the signature by including a checksum in every signature ➢ Shares distributed to n parties for overlapping responsibility ➢ Zero hassle configuration as no programs are needed to be installed
Relationships	➢ Online Manual will be available to clients
Customer Segments	This application can be utilized under several different scenarios. For instance, it can be applied in: ➢ Banking : To provide authentication for signatures on cheques and legal documents pertaining to banks ➢ Insurance : Insurance agent requires the policyholder to sign the insurance documents for claims with required parties ➢ Legal Document : Legal documents that require signatures benefit from the authentication mechanism
Channels	➢ Company computers ➢ Employee workstations ➢ Server communicating with web application
Cost Structures	➢ Infrastructure and environment ➢ Development ➢ Marketing ➢ Deployment
Revenue Streams	➢ Application available for a month trial for evaluation ➢ Monthly subscription scheme

Problems

Physical signatures that are currently being used are susceptible to forgery. To reduce the likelihood of forgery, we propose a secure scheme to sign documents with an added layer of authentication and cryptographic security.

Target

Organisations that wish to provide an authentication mechanism for electronic signatures, primarily banks.

Solution

Clients go through a registration phase where they provide the bank with a written signature. This signature is scanned and stored online. A key image is produced and the 2 shares for the client's signature are produced. One of the shares is stored in the clients account while the other is stored in the banks database. When the bank wants the client to sign their signature, they will send the client the document they wish signed with their share attached, onto which the client provides their share. If the client wishes to send a document, without the initiation of the bank, they provide their share on the document and the bank imposes their share to reconstruct the signature.

Reaching and acquiring customers

We would also actively approach potential clients by reaching out to them electronically and physically; this includes, but not limited to, web short video introduction on how our web application works, the simplicity and the ease of use of it. The brand's identity and reputation amongst its competition will slowly be built from small organisations. Through exposure gained from these organisations, we will continue to spread the usage of the product to larger scale organisations and thus gain their trusts.

Revenue generation

The web application can be trialed for 14 days; after which clients would be required to have a monthly subscription in order to continue using it. The monthly subscription will also include product support via email.

Target User Domain

The application aims to provide authentication and verification of the signature used to sign a document. The product is projected under a generic domain but leans in the favour of the banking domain, where signatures on legal documents and cheques are critical components of their workflow. The project provides a means for organisations to get signatures from their designated parties globally whilst still providing integrity for the signature.

Product Validation

The idea was floated amongst several banking professionals and has garnered interest. The current banking standards do not have the option for digital signatures on checks and documents. Our product will fill the gap where documents and checks can be signed for eBanking.

Requirement Specification

Interfaces

User Interface

The user interface has to be straightforward, such that respective users should be able to easily comprehend what their next course of action should be when interacting with the system. The interface should not be cluttered and should provide feedback to the customer to show that their actions have been processed. It should provide ease of use when users have to administer uploads or downloads of documents.

Hardware Interface

The product does not have a physical interface.

Software Interface

Website

Communication Interface

Internet

Functional Requirements

Provide Share of Signature

Bank and customer provides shares to be used to recover the original signature

Priority :	HIGH
Benefit :	Allows for the reconstruction of the signature
Penalty :	Without the shares provided by bank and customer, original signature does not get reconstructed
Cost :	No authentication can be carried out
Risk (1-9) :	9

Drop-in Signature for Encryption

Bank will provide customer's signature for share creation

Priority :	HIGH
Benefit :	Allow application to carry out encryption and creation of shares based on signature provided
Penalty :	Will not be able to carry out encryption process without signature
Cost :	Removes the purpose of the application
Risk (1-9) :	9

Create Shares

Shares created to be used for verification purposes later by the Bank and the Customer

Priority :	HIGH
Benefit :	Allow for confidentiality and signing of documents by both bank and customer
Penalty :	Signature verification can't be carried out.
Cost :	Removes the functionality of the application
Risk (1-9) :	9

Generate Key Image

This function generates a random layer that would be overlayed on top of the original signature

Priority :	HIGH
Benefit :	Facilitates in application of visual cryptography
Penalty :	If this feature is not implemented, the user won't be able to create shares of the signature
Cost :	People would stop using our application
Risk (1-9) :	9

Resize Image

All signatures will be resized before share creations occurs to a default size

Priority :	HIGH
Benefit :	Resize image based on to the specific requirement.
Penalty :	Users would have to manually resize the document before encrypting the document with our application.
Cost :	Users would have to download a photo editing application to resize the image; this will reduce the productivity of the users.
Risk (1-9) :	7

Send Shares to Customer & Bank

Once the shares are created, the shares are sent to the respective parties

Priority :	HIGH
Benefit :	We are not required to store the shares, we can save on storage space.
Penalty :	Users would have to manually send the share to the shareholders
Cost :	The Share would be at one spot for a moment, compromising security
Risk (1-9) :	8

Add Share to Document/Cheque

Layering of shares to authenticate the document/cheque

Priority : HIGH

Benefit : Provides mechanism for remote authentication and does not require all participants to be active at the same time

Penalty : Makes functionality of application redundant

Cost : Without binding the signature to the document, application does not serve purpose

Risk (1-9) : 9

Remove Noise

To effectively reconstruct the original signature, noise will be removed from the shares.

Priority : HIGH

Benefit : The image would have noises in it when it is reconstructed from the shares. This would make the deciphering of the signature clearer.

Penalty : Elements of the signature might be obscured by the noises, hence affecting the authenticity of the signature.

Cost : This would render our application useless because we are unable to verify the signature used as certain vital elements might not be shown clearly.

Risk (1-9) : 9

Non-Functional Requirements

Performance Requirements

- The program should be able to split the signature into shares within 3 seconds
- The program should be able to sign the document within 3 seconds.

Safety Requirements

- It should work as a middleware layer.

Security Requirements

- Security and Authenticity is an important aspect of the program
- The original signature (plaintext) will never be stored by us
- The shares generated by us will never be stored
- Signed Documents will never be stored by us

Software Quality Attributes

Adaptability : Available as a web application; independent of operating system.

Availability : Available 24/7

Flexibility : The Bank has the ability to submit a signature to the application to undergo encryption and allocate the shares back to the Banks and the Customer(s). Customers who wants to sign, will input the shares into the document and will be sent to the bank to be verified. The Bank will reconstruct (decryption) and verify the signature from the document sent by the Customer.

Maintainability: Future updates are provided by developers on the web application

Portability : Portable

Reliability: Redundant servers are to be put in place to account for failure, and to serve as either a backup or load sharing server in case of heavy traffic or an outage

Reusability: Application is reusable

Robustness: Robust

Usability: Requires the Bank to be familiar with the process of submitting signature(s) for encryption and verification (after decryption) of the received documents.

The Customers to be familiar with the process of providing the shares to sign the document to be sent to the Bank for decryption.

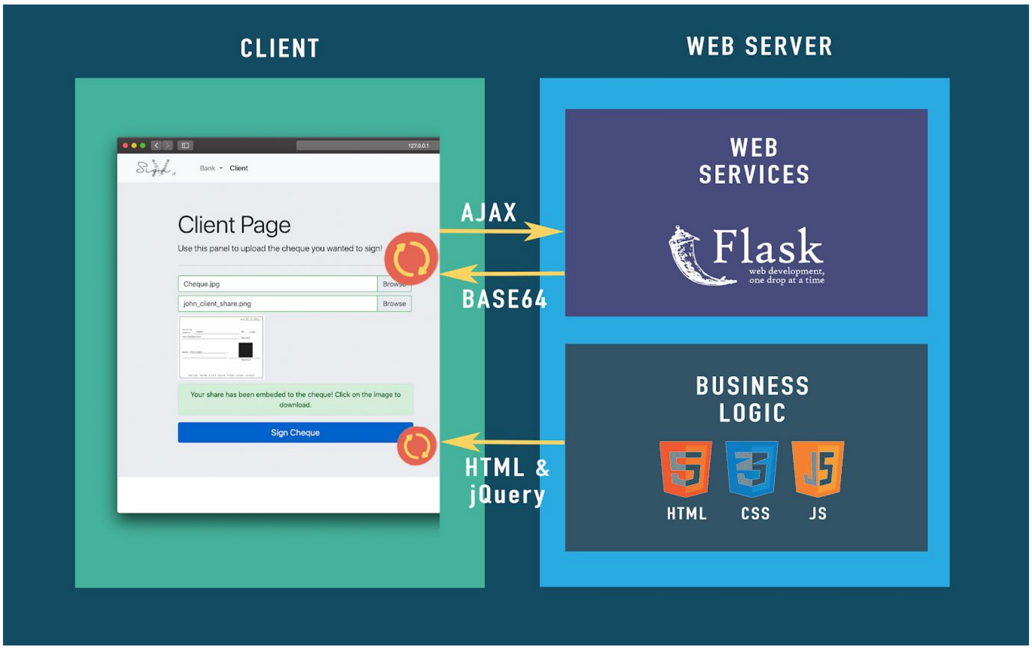
These processes are very intuitive and the usage does not require the participants to be familiar with the inner workings of the encryption and decryption process. The Customers simply have to be able to understand how to drop-in their signatures/shares on to the documents and sent it to the Bank while the Bank knowing how to encrypt and decrypt using the application.

Other Requirements

- NIL

Design Specification

Product Architecture



Client Page

Use this panel to upload the cheque you wanted to sign!

Your share has been embeded to the cheque! Click on the image to download.

Figure 13: Product Architecture

Use Case Diagram

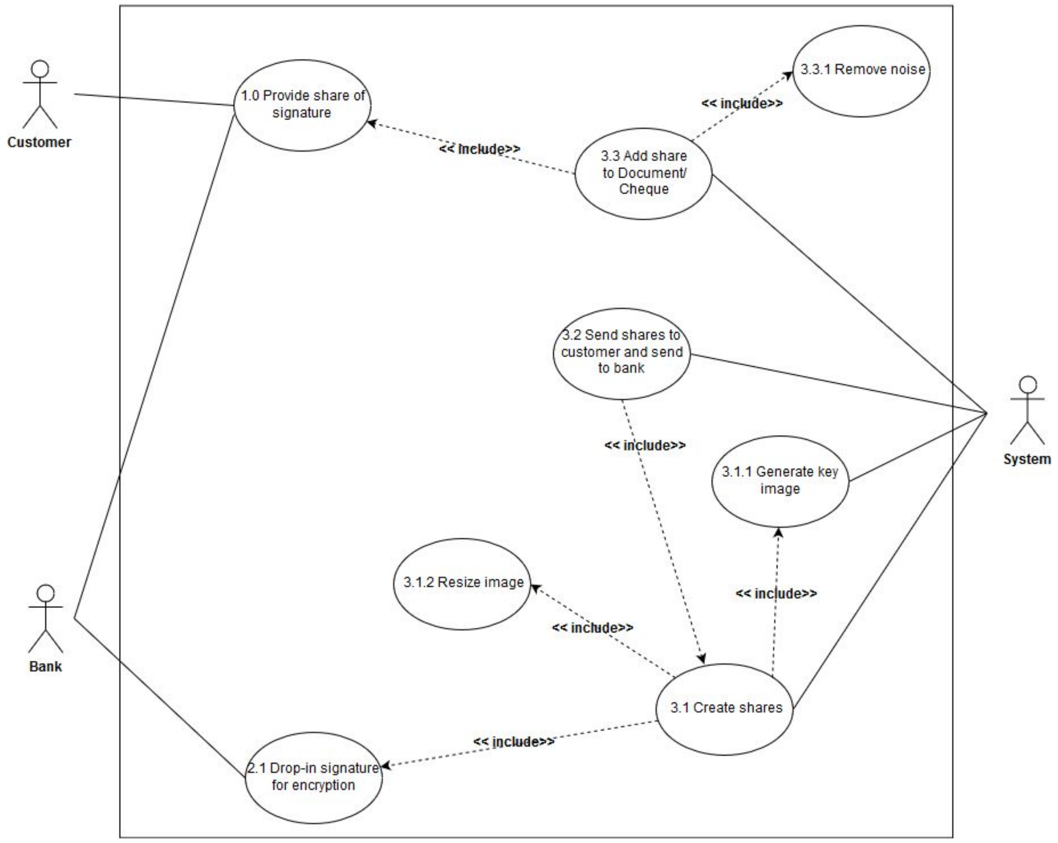


Figure 14: Use Case Diagram

Use Case Textual

Provide share of signature	
UC-ID	1.0
Description	Shares of signature must be provided for the document/cheque decryption and signature verification purposes.
Actor(s)	Customer and Bank
Main Scenario	1) Customer and Bank clicks on the button to provide their share of the signature to the web application.

Drop-in Signature for Encryption	
UC-ID	2.1
Description	Bank drop-in the signature (image) of the Customer for Encryption.
Actor(s)	Bank
Main Scenario	1) Bank click on the button to drop-in their signature (image).

Create Shares	
UC-ID	3.1
Description	Shares created by the System (web application) upon drop-in of signatures.
Actor(s)	System
Main Scenario	1) Bank click on the button to drop-in their signature (image) into the System. 2) The System (web application) will resize image of the signature. 3) The System (web application) will generate a key image to be used for the encryption of the image. 4) The System (web application) will encrypt the signature with the generated key image. 5) The System (web application) will then create a share each for the Customer and the Bank.

Generate Key Image	
UC-ID	3.1.1
Description	Key image generated to be used for encryption with the signature (image).
Actor(s)	System
Main Scenario	1) The System (web application) will generate a key image.

Resize Image	
UC-ID	3.1.2
Description	Image (signature) is resize automatically by the System (web application) before it can be used for the creation of shares.
Actor(s)	System
Main Scenario	1) The System (web application) will automatically resize the Signature (image) before it is being used for the encryption and creation of shares.

Email Shares to Customer and send to Bank	
UC-ID	3.2
Description	Created shares of the signature will be send to the Customer and the Bank respectively via Email.
Actor(s)	System
Main Scenario	1) The System (web application) sends the created shares of the signatures to the Customer and the Bank respectively.

Add Share to Document/Cheque	
UC-ID	3.3
Description	The provided share of the signature by the Customer will be added into the document/cheque.
Actor(s)	System
Main Scenario	1) The System (web application) adds the provided share of the signature by the Customer into the document/cheque.

Remove noise	
UC-ID	3.3.1
Description	To effectively reconstruct the original document, noise will be removed from the shares.
Actor(s)	System
Main Scenario	1) Remove grids from reconstructed image.

Class Diagram

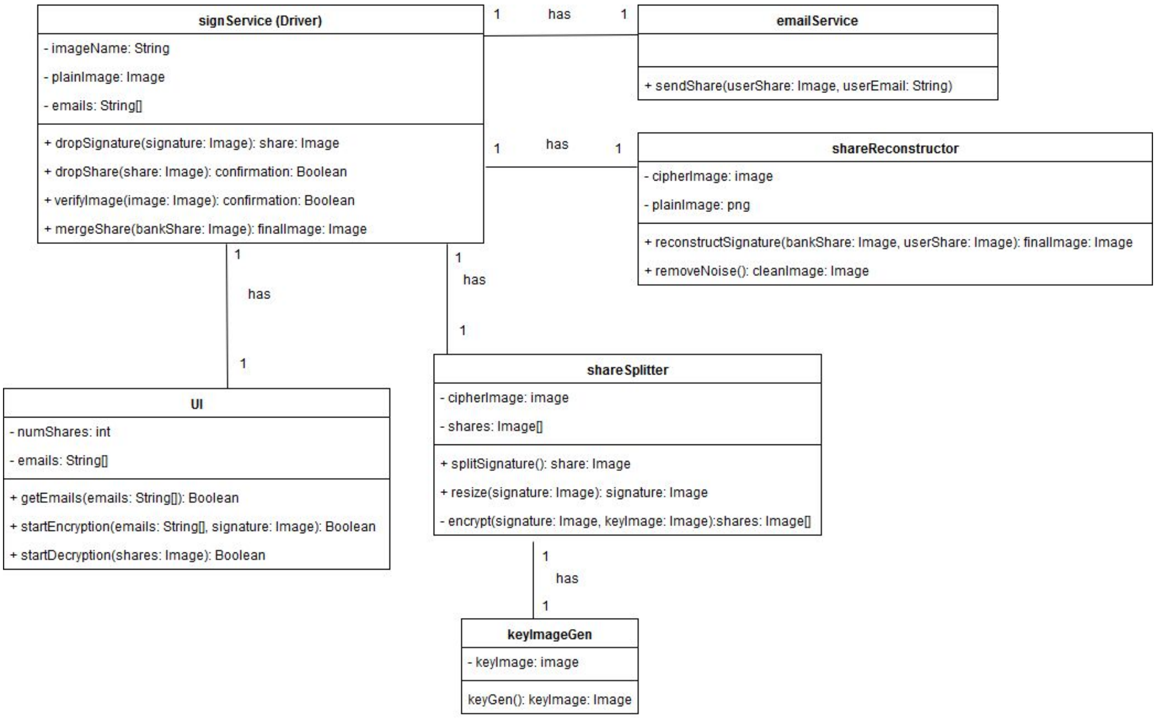

signService (Driver)
- imageName: String
- plainImage: Image
- emails: String[]
+ dropSignature(signature: Image): share: Image
+ dropShare(share: Image): confirmation: Boolean
+ verifyImage(image: Image): confirmation: Boolean
+ mergeShare(bankShare: Image): finalImage: Image
1
has
1
emailService
+ sendShare(userShare: Image, userEmail: String)
1
has
1
shareReconstructor
- cipherImage: image
- plainImage: png
+ reconstructSignature(bankShare: Image, userShare: Image): finalImage: Image
+ removeNoise(): cleanImage: Image
1
has
1
1
has
1
UI
- numShares: int
- emails: String[]
+ getEmails(emails: String[]): Boolean
+ startEncryption(emails: String[], signature: Image): Boolean
+ startDecryption(shares: Image): Boolean
shareSplitter
- cipherImage: image
- shares: Image[]
+ splitSignature(): share: Image
+ resize(signature: Image): signature: Image
- encrypt(signature: Image, keyImage: Image):shares: Image[]
1
has
1
keyImageGen
- keyImage: image
keyGen(): keyImage: Image

Figure 15: Class Diagram

Sequence Diagrams

Provide Share of Signature (1.0)

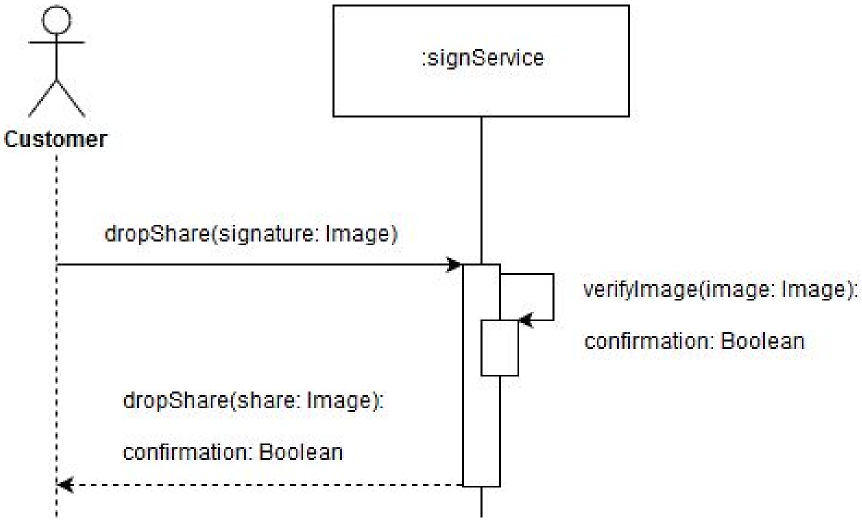


Figure 16: Provide Share of Signature (1.0)

Drop-in Signature for Encryption (2.1)

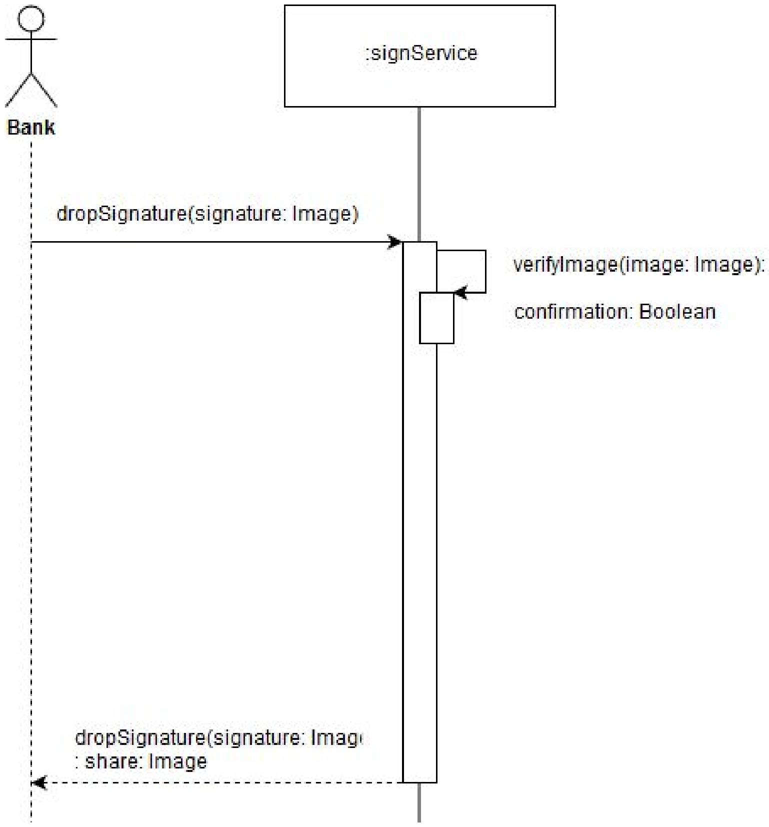

:signService
Bank
dropSignature(signature: Image)
verifyImage(image: Image):
confirmation: Boolean
dropSignature(signature: Imag
: share: Image

Figure 17: Drop-in Signature for Encryption (2.1)

Create Shares (3.1)

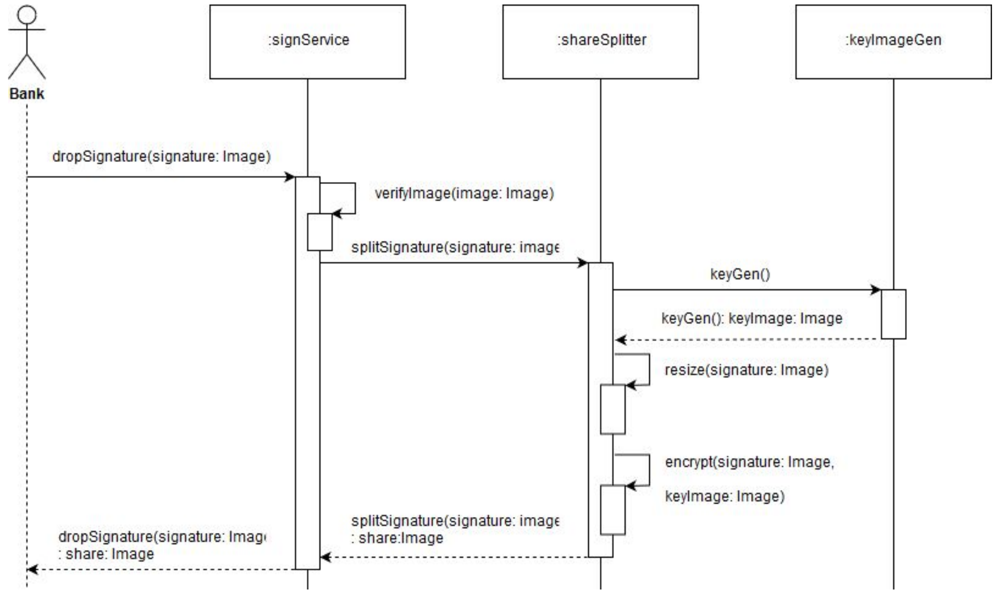

Bank
:signService
:shareSplitter
:keyImageGen
dropSignature(signature: Image)
verifyImage(image: Image)
splitSignature(signature: image
keyGen()
keyGen(): keyImage: Image
resize(signature: Image)
encrypt(signature: Image,
keyImage: Image)
splitSignature(signature: image
: share:Image
dropSignature(signature: Imag
: share: Image

Figure 18: Create Shares (3.1)

Generate Key Image (3.1.1)

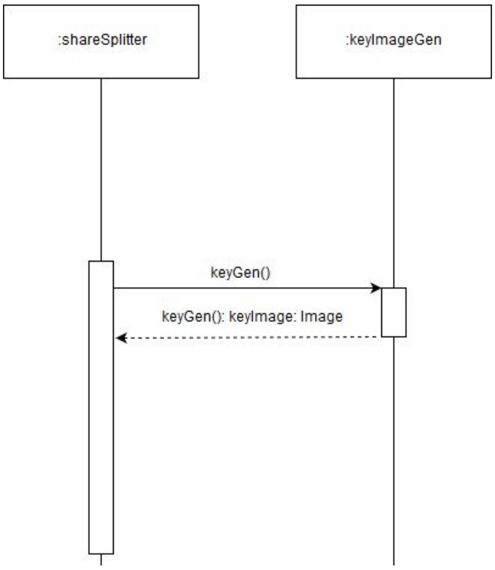


Figure 19: Generate Key Image (3.1.1)

Resize Image (3.1.2)

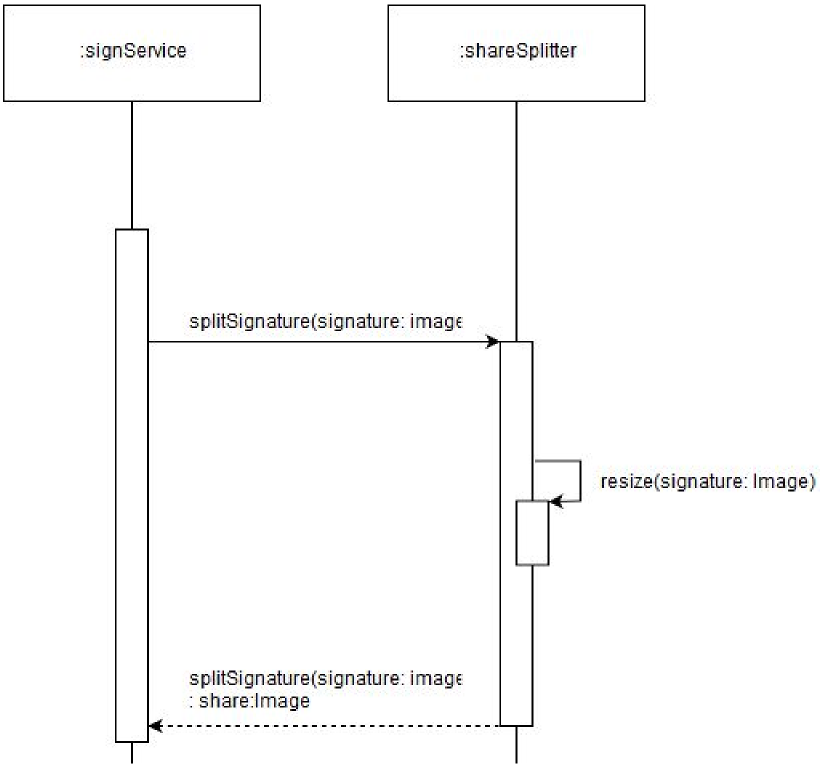


Figure 20: Resize Image (3.1.2)

Email Shares to Customer and Bank (3.2)

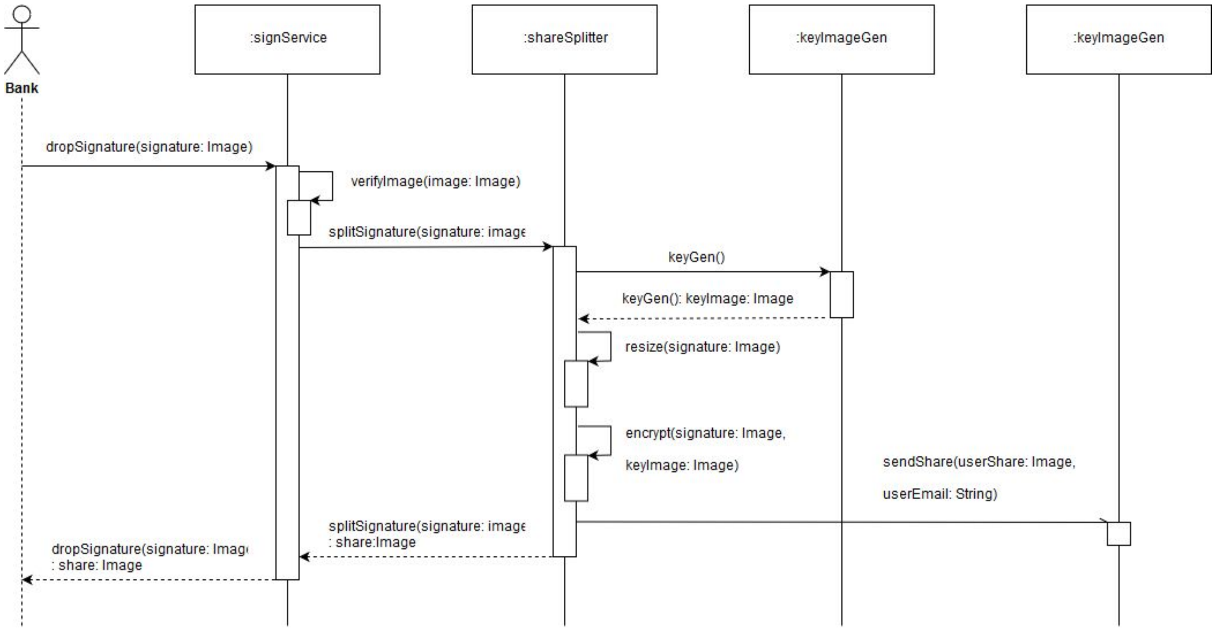

Bank
:signService
:shareSplitter
:keyImageGen
:keyImageGen
dropSignature(signature: Image)
verifyImage(image: Image)
splitSignature(signature: image
keyGen()
keyGen(): keyImage: Image
resize(signature: Image)
encrypt(signature: Image,
keyImage: Image)
sendShare(userShare: Image,
userEmail: String)
splitSignature(signature: image
: share:Image
dropSignature(signature: Image
: share: Image

Figure 21: Email Shares to Customer and Bank (3.2)

Add Share to Document/Cheque (3.3)

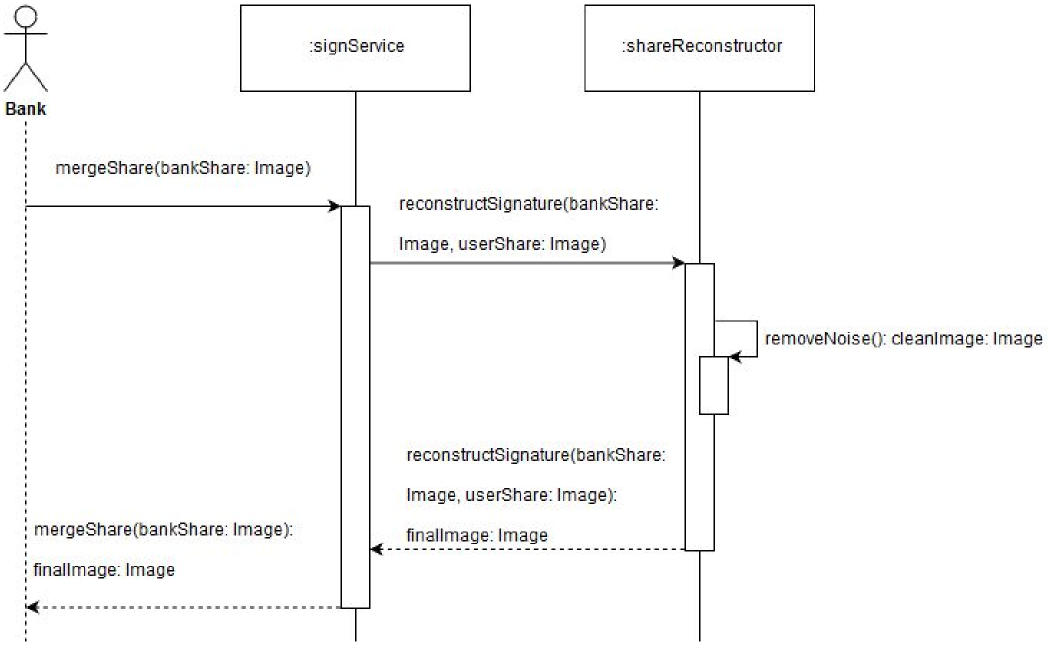

Bank
:signService
:shareReconstructor
mergeShare(bankShare: Image)
reconstructSignature(bankShare: Image, userShare: Image)
removeNoise(): cleanImage: Image
reconstructSignature(bankShare: Image, userShare: Image): finalImage: Image
mergeShare(bankShare: Image): finalImage: Image

Figure 22: Add Share to Document/Cheque (3.3)

Remove Noise (3.3.1)

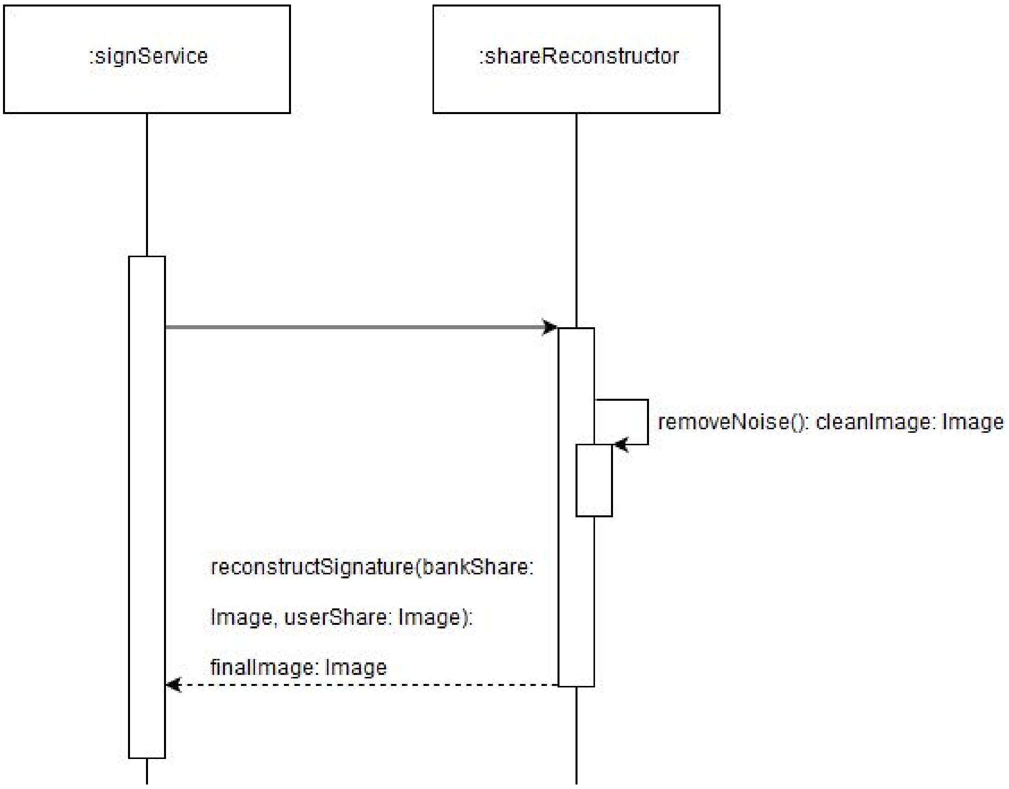

:signService
:shareReconstructor
removeNoise(): cleanImage: Image
reconstructSignature(bankShare: Image, userShare: Image): finalImage: Image

Figure 23: Remove Noise (3.3.1)

Activity Diagrams

The following activity diagrams encompass the activity flow for the entire application, and is categorized into 3 main components. These components are creation of shares, sending shares to bank and customer, and adding of shares to cheque/document.

Create Shares (3.1)

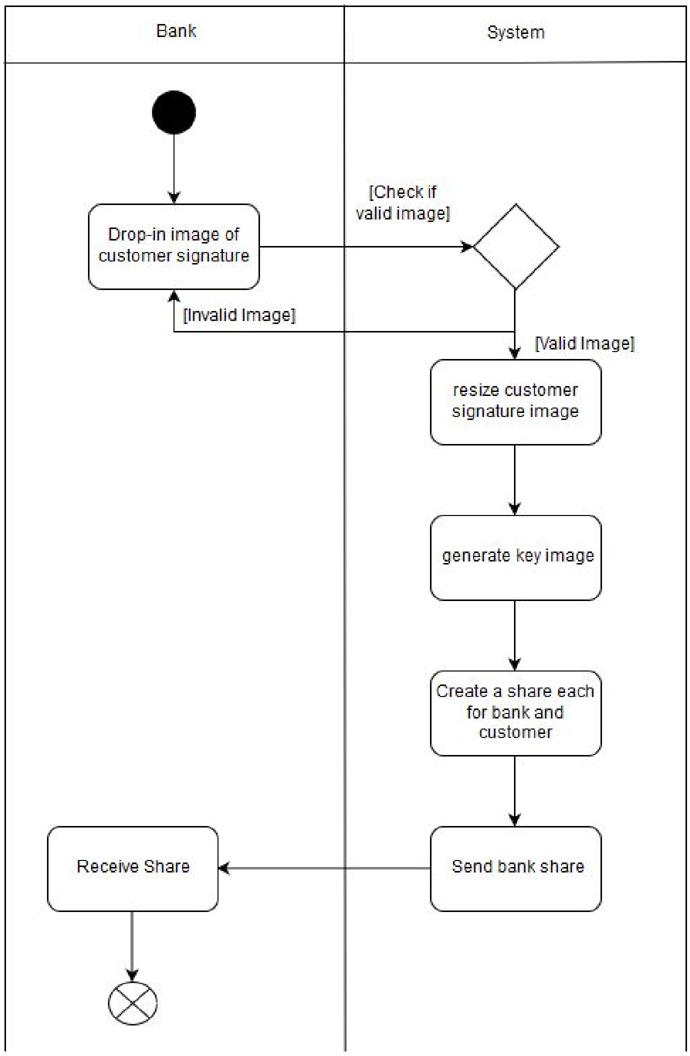

Bank
System
Drop-in image of customer signature
[Check if valid image]
[Invalid Image]
[Valid Image]
resize customer signature image
generate key image
Create a share each for bank and customer
Send bank share
Receive Share

Figure 24: Create Shares (3.1)

Send Shares to Customer and Bank (3.2)

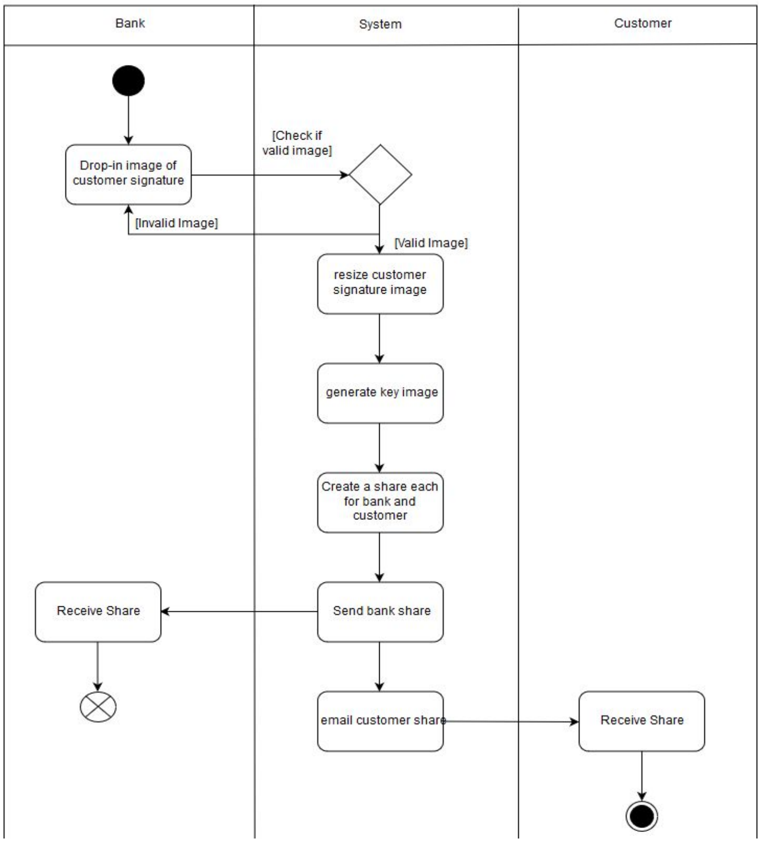


Figure 25: Email Shares to Customer and Bank (3.2)

Add Share to Document/Cheque (3.3)

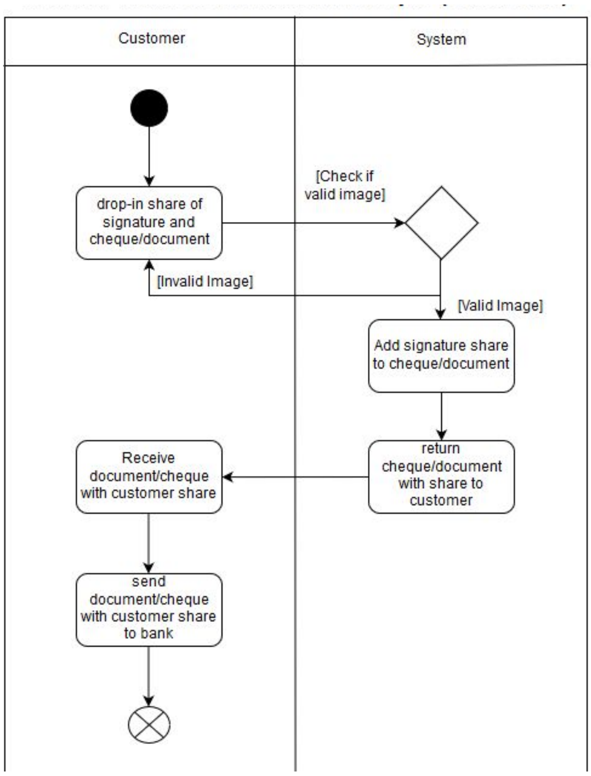

Customer
System
drop-in share of signature and cheque/document
[Check if valid image]
[Invalid Image]
[Valid Image]
Add signature share to cheque/document
Receive document/cheque with customer share
return cheque/document with share to customer
send document/cheque with customer share to bank

Figure 26: Add Share to Document/Cheque (3.3) [Customer perspective]

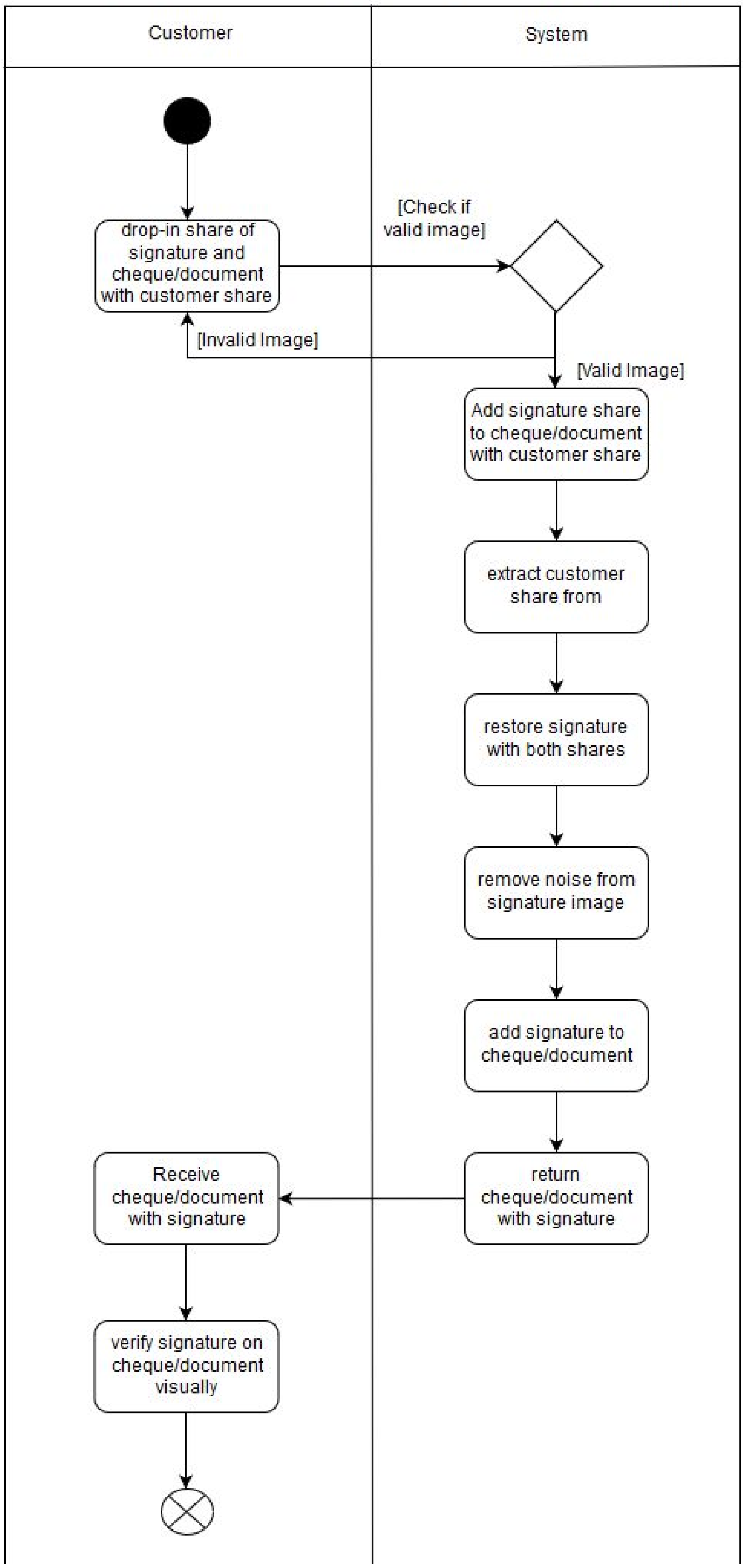

Customer
System
drop-in share of signature and cheque/document with customer share
[Check if valid image]
[Invalid Image]
[Valid Image]
Add signature share to cheque/document with customer share
extract customer share from
restore signature with both shares
remove noise from signature image
add signature to cheque/document
return cheque/document with signature
Receive cheque/document with signature
verify signature on cheque/document visually

Figure 27: Add Share to Document/Cheque (3.3) [Bank perspective]

Development Methodology

Waterfall Model

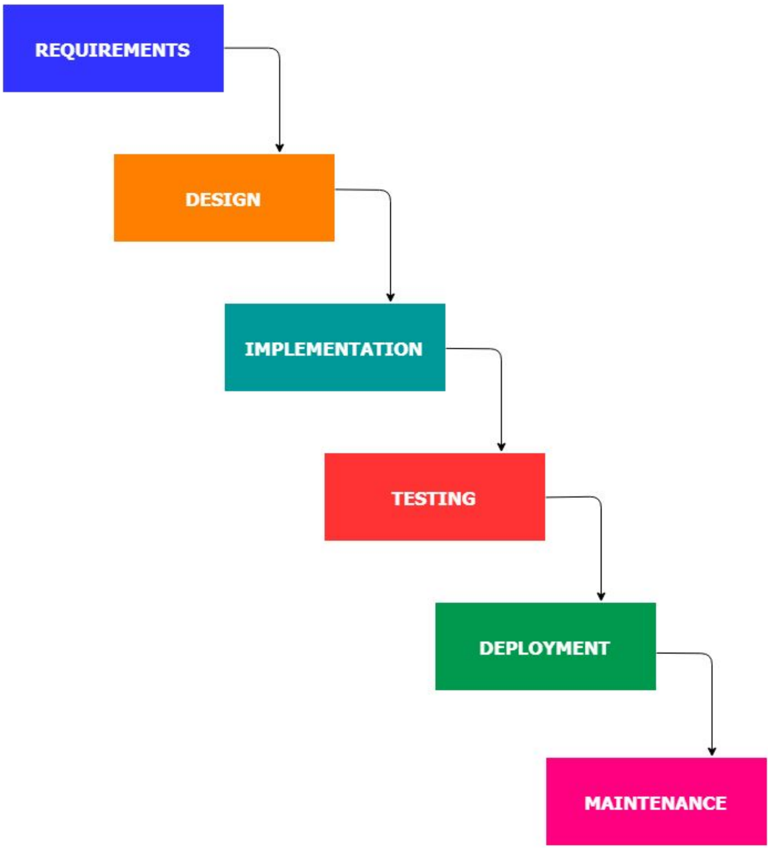
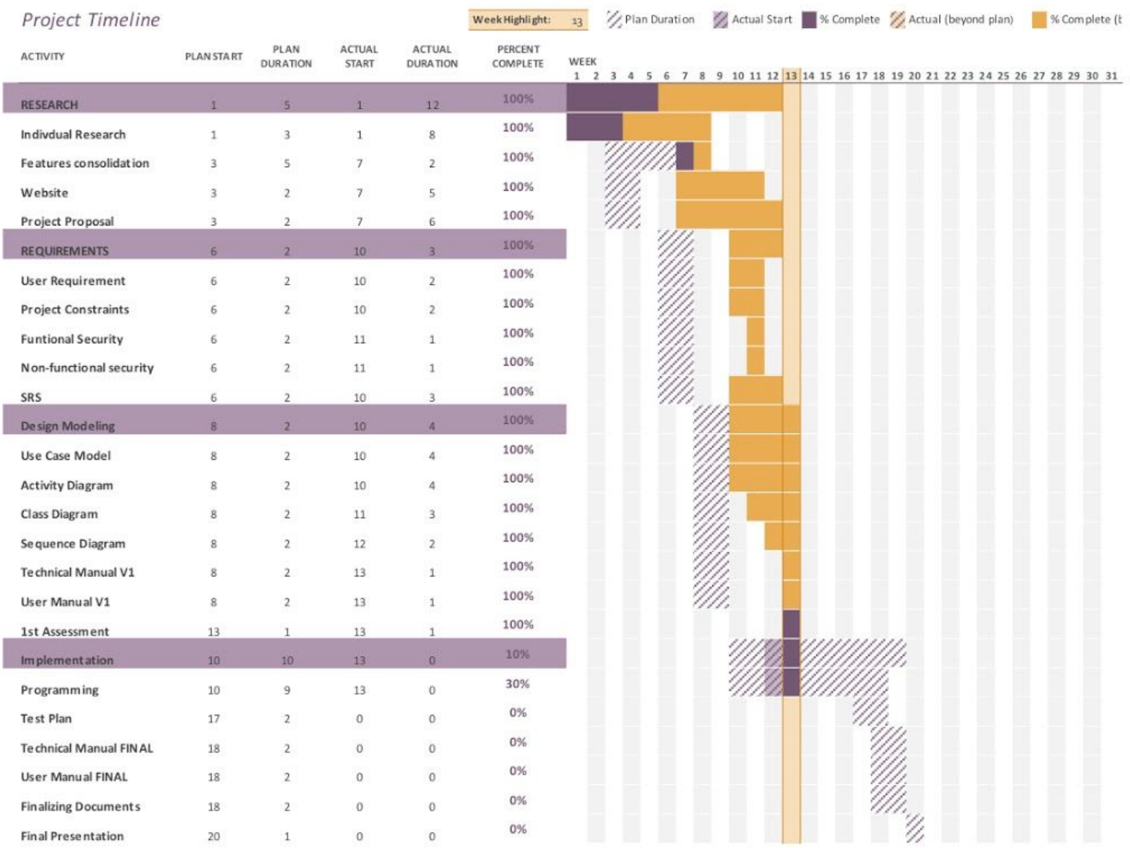


Figure 28: Waterfall Model

We chose waterfall model as it has a strict rigid model. This allows us to set deadlines so that we are able to progress through each phase of the waterfall mode effectively and efficiently. Our development moved from Requirements into Design, Implementation, Testing , Deployment and finally maintenance. The Waterfall model has the advantages of simplicity and easy to use. Each phase is done step-by-step. We have clearly defined phases with absolute goals.

Project Timeline



Project Timeline

Week Highlight: 13

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
RESEARCH	1	5	1	12	100%
Indivdual Research	1	3	1	8	100%
Features consolidation	3	5	7	2	100%
Website	3	2	7	5	100%
Project Proposal	3	2	7	6	100%
REQUIREMENTS	6	2	10	3	100%
User Requirement	6	2	10	2	100%
Project Constraints	6	2	10	2	100%
Funtional Security	6	2	11	1	100%
Non-functional security	6	2	11	1	100%
SRS	6	2	10	3	100%
Design Modeling	8	2	10	4	100%
Use Case Model	8	2	10	4	100%
Activity Diagram	8	2	10	4	100%
Class Diagram	8	2	11	3	100%
Sequence Diagram	8	2	12	2	100%
Technical Manual V1	8	2	13	1	100%
User Manual V1	8	2	13	1	100%
1st Assessment	13	1	13	1	100%
Implementation	10	10	13	0	10%
Programming	10	9	13	0	30%
Test Plan	17	2	0	0	0%
Technical Manual FINAL	18	2	0	0	0%
User Manual FINAL	18	2	0	0	0%
Finalizing Documents	18	2	0	0	0%
Final Presentation	20	1	0	0	0%

Figure 29: Project Timeline

Development Tools

Programming Language and Associated Libraries

Python - Flask

Javascript - jQuery, AJAX

Web Development Tools and Libraries

HTML

CSS

Bootstrap

Version Control

Github

Risk Assessment

	Risk	Risk Statement	Initial Risk (High, Medium, Low)	Control Plan	Final Risk (High, Medium, Low)
1.	Corruption of Shares	If the shares stored by either the customer or bank gets corrupted, the signature cannot be combined for verification	Low	The Customer would have to be issued a new share for signing purposes	Low
2.	Invalid signature after reconstruction	When the Bank reconstructs the signature, the signature doesn't match Bank records	Medium	The Customer needs to ensures their signature is updated with the Bank	Medium
3.	Unable to reconstruct shares	There is an issue during the verification of signature, as the program is unable to combine the shares	Medium	Check of the customer has used the correct share to sign the document	Low
4.	Unable to 'drop-in' Signature	The customer is unable to "drop-in" their signature share to sign documents	Low	We would have to fix the code, to check for any problems	Low
5.	Incorrect customer's email entered	Shares generated for the Customer can't be sent	High	The Bank needs to ensure that the email entered is correct and the email is being used by the Customer	High
6.	Signature size is too large	Signature size is too large; the shares cannot be created	Medium	The Web Application will auto-resize the signature so that the shares can be created	Low
7.	Web Application freezes	The application stops working	Medium	Ensure Web Browser is up-to-date; reload the Web Application	Medium
8.	Ink of signature provided is too thin	Does not allow for effective decryption	High	Customer must re-do their signature	Low

Test Analysis

Test Objective

To deliver a seamless user experience for the Bank and the Customer, we need to do rigorous testing to ensure that the web application is bug-free as possible and it is working as intended.

Test Scope

placeholder

Test Cases

Test ID	
Description	
Browser and Operating System	
Test Date	
Tester	

	Test Case	Expected Results	Actual Results	PASS	FAIL	Priority
1.						