

1 Introduction

1.1 Purpose

The purpose of this document is to provide an overview of the proposed project along with the ultimate goal of its implementation. A detailed description of all the components of the project in addition to their inner-workings and how they interact to form one complete system will be included as well.

1.2 Objective

The proposed project's primary objective is to offer an application that allows users to provide proof to another party that the total value of their financial assets is greater than or equal to a given amount without revealing the actual balance. The user of the product will have the ability to share their official financial documents with another party while obscuring only the balances on those documents. All the information that is needed in order to associate that document with the user such as the consumer's name, address, and financial institution will remain visible to the other party. The balance will be compared to the verifying amount, which is the minimum amount determined by the other party that needs to be held in the user's account, and the result will be shared with both parties. The result will consist of a simple true or false value, indicating whether the user has at least the minimum amount in their account or not.

1.3 Description of issue

The project is aimed at delivering a product for users that need to share their financial information with another party for verification purposes, but would prefer not to reveal the total value of all their assets. This involves consumers applying for mortgages, business loans, leases or any other business transaction requiring them to provide proof that they hold a certain amount in assets. These customers are also very concerned about sharing their financial data with another party from both a security standpoint and general privacy sense. Sharing financial information involves risk, as one cannot guarantee who will have access to that data. In addition, revealing financial data is a proposition that many believe involves compromising personal privacy. Having the ability to provide verification of the minimum value of one's financial assets without revealing the actual balances of those assets resolves the conflict of both the security and privacy concerns.

1.4 Users

The users of the system will include both the consumer who wishes to provide proof that their financial asset balance is at least a certain amount and the party that is interested in verifying that information. This will primarily involve financial institutions and consumers applying for loans through those institutions.

1.5 Definitions

- **User** – The person sharing their financial information.
- **Verifier** – The party interested in verifying that the user's asset balance is at least a certain amount.
- **Verification** – The product of a user submitting a request to provide proof that their financial asset balance is at least a given amount to another party.
- **Threshold** – The minimum amount, set by the verifier, that the user's asset balance must equal.
- **UI** – User interface

1.6 Use cases

- The user should be able to enter in all the required information in the user interface.
- The interface should upload a PDF document and display its image within the same window.
- The user should be able to submit a new verification and that verification should be saved to the database.
- The other party should be able to view the verification and accept or decline it.

2 Components

2.1 User interface

The UI is the only component that the user will interact with. The interface can be accessed through any web browser and will require user registration. Once logged in, the user will be directed to the account page. This page will display the user's name and email address. The main menu will be on the left side of the page and will contain links to all the pages. The UI includes the following pages: account, new verification, pending verification, approved verification and declined verification.

The UI will require the user to provide 3 items in order to submit a new verification. These required items include, the name of the verifier, the threshold, and the official financial documents. The UI will have 2 text inputs; one for the verifier name and one for the threshold amount. The name field should accept all characters while the threshold field should only accept digits. There will also be a button to upload local files and a file viewer window that displays an image of the document that was uploaded. The file uploader should only allow PDF documents to be uploaded. Attempting to upload a document format other than PDF should result in an error message. Once an appropriate document has been uploaded, the file viewer window should display an image of the document.

The user will have access to all the verifications that they have submitted through the UI. Verifications will be displayed in the user interface based on 3 classifications: Pending, approved, and declined. Pending verifications are new submissions that have not been processed. Approved verifications are submissions that have been processed and successfully verified. Finally, declined verifications consist of submissions that have been declined due to either unapproved documents or because the asset balance is not greater than or equal to the threshold set. Each of these verification categories will be displayed on their own page and can be accessed through the links on the main menu. All verifications will be listed on their respective page in rows with column headers indicating the following information: date of

submission, date of action (if approved or declined), verifier name and threshold. A link to an image of the document uploaded will be included in each row.

The verifier will have access to a similar interface. The pages for the verifier UI will consist of the new verification page, approved verification page, and the declined verification page. Verifications will be listed in the same fashion as they are in the user's UI. The column headers will list the date of submission, date of action (if approved or declined), name of the user and the threshold amount. The verifier will also have access to the image of the document uploaded. In addition, each row within the new verification page of the verifier UI will contain an accept and reject button.

Once a user submits a new verification, the verifier will receive an alert and will need to log in to their account. The verifier will need to view the new verification page and click on the verification. After confirming that the official financial documents are authentic and that the threshold is the correct amount, the verifier can then click 'accept'. If there is any issue with the uploaded document or threshold amount, the verifier can reject the verification by clicking the 'reject' button.

The verifier will have access to all verifications submitted to them. These verifications will include the new verifications that they have not approved, the approved verifications that were successfully verified, and the declined verifications that either did not have at least the threshold amount or that had unsatisfactory documents.

Development on the user interface began in October and is expected to be completed by the beginning of February. Final testing and integration is being conducted incrementally and should be completed by the middle of February.

2.2 Database

The database will store several tables consisting of several columns that will be key in displaying and sharing information between the user and verifier. These tables will store users, verifiers, pending verifications, approved verifications, declined verifications, document images and cipher text values.

Development on the database began in October and is expected to be completed by the end of January.

2.3 Web API

The API will be providing an OCR service for the user's uploaded documents. The documents will be converted from a PDF document into an HTML string through this web service. The API should accept all documents in PDF format, accurately read all characters on that document and return the precise HTML product of that document. The OCR service should reliably read all documents uploaded without errors.

Development and integration on the Web API began in October and was completed in November. Testing is expected to continue until the middle of January.

2.4 Encryption module

The encryption module is a Java program tasked with encrypting all the balances read from the user's uploaded documents and summing them. The program should read in a text file placed within its

directory named 'plaintexts' and scan integers separated by line breaks. The program should return a cipher text value representing the sum of all the integers within that text file.

The program is also responsible for reading in a text file name 'ciphertext' placed within its directory, scanning the cipher text value in that file, and subtracting a threshold amount from that value. The program should be able to take in the threshold amount as a parameter.

The algorithm should be able to read multiple lines within the text files given and discern an individual value based on it being placed on its own page line.

Development on the program began in October and is expected to finish by the end of January. Testing and integration should be completed around the same time.

2.5 Value extraction module

The HTML string returned by the Web API will be parsed by the value extraction program. All values preceded by a dollar symbol will be replaced with arbitrary characters in order to mask the data intended to be kept private. The values will also be saved to a text file for later processing by the encryption module.

The program should accurately identify all the values following a dollar symbol and accomplish two goals: mask those values by replacing them with arbitrary characters and save those files to a text file.

Development on this module began in November and was completed within the same month. Testing and integration was also completed during the same month.

3 Overview of System

