Media Engineering and Technology Faculty
German University in Cairo

Handwritten Fonts

Bachelor Thesis

Author: Mahmoud Mohammad Abou Eleneen
Supervisors: Dr. Wael Abouelsaadat

Submission Date: 23 June, 2024

Media Engineering and Technology Faculty
German University in Cairo

Handwritten Fonts

Bachelor Thesis

Author: Mahmoud Mohammad Abou Eleneen

Supervisors: Dr. Wael Abouelsaadat

Submission Date: 23 June, 2024

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowlegement has been made in the text to all other material used

Mahmoud Mohammad Abou Eleneen
23 June, 2024

Acknowledgments

I would like to express my sincere gratitude to my thesis supervisor, Dr. Wael Abouelsaadat. Dr. Wael provided invaluable guidance, feedback and expertise that helped shape the research and system into its final form for this thesis.

Abstract

Technology is often criticized as being impersonal and cold. The emergence of digital communication in the 21st century, with billions of users engaging in online interactions, underscores the need for personalized experiences. While personalization efforts exist in various digital platforms, a key element of human communication remains largely absent: the personal touch conveyed through handwriting. This thesis explores the concept of bridging this gap by developing a software system that leverages a user's individual handwriting for web-based messaging, by allowing users to send messages in their own handwriting on the web.

The project aims to move beyond the impersonal nature of standardized fonts by enabling users to incorporate their own handwriting into their digital communication. Furthermore, the software will explore the potential for users to express emotions by generating different emotional variations of their handwritten fonts, that can indicate emotions such as happiness or anger, as opposed to the user's normal, neutral handwritten font. This research delves into the potential for such technology to humanize and personalize the digital communication experience, fostering a more nuanced and potentially emotionally richer connection between individuals.

By investigating the integration of handwriting into web-based messaging and exploring the expression of emotions through such personalized fonts, this thesis seeks to contribute to a more human-centric approach to digital communication.

Contents

Chapter 1

Introduction

The digital landscape has fundamentally transformed communication in the 21st century. Billions of users engage in online interactions daily, fostering connections and communities across geographical boundaries. However, a crucial element of human connection often gets lost in the digital realm – the personal touch conveyed through handwriting.

This thesis delves into the potential for bridging this gap by exploring the development of a software system that integrates a user's individual handwriting into their web-based messaging experience. We move beyond the impersonal nature of standardized fonts by allowing users to express themselves through their own handwriting in messages they send over the web. Furthermore, the project investigates leveraging different emotional variations of the users' handwriting to express specific emotions within their messages.

This research investigates the potential for such technology to humanize and personalize digital communication. The integration of handwriting and the exploration of emotional expression through personalized fonts hold the promise of fostering richer connections and a more nuanced communication experience in the digital world.

1.1 Motivation

Standardized fonts currently used in digital communication lack the individuality and emotional depth inherent in handwriting. Handwritten messages, with their subtle variations and nuances, add a layer of personality and emotional depth that can be missing in online interactions. This project is driven by the desire to address this shortcoming.

With the use of emojis, color themes, stickers, and GIFs as means for customizing messaging experiences and enhancing expressiveness, this project aims to introduce the next major step in advancing web-based communication. By incorporating a user's unique handwriting into web-based messaging, we can create a more engaging and emotionally resonant experience. Furthermore, the exploration of expressing emotions through variations in handwriting presents a revolutionary opportunity to enhance the expressiveness of digital communication.

This research is not intended to replace normal texting entirely, which remains a valuable tool for its speed and convenience. However, in specific situations, a handwritten message carries a personal touch that normal text messages lack. This project aims to bring the same personal element to the digital realm of web-based messaging.

1.2 Aim of the Project

This thesis focuses on developing a software system that enables users to leverage their own handwriting for web-based messaging. The project will explore various methods for capturing and integrating user handwriting into the communication platform. Additionally, we will investigate the potential for users to create and utilize variations of their handwriting to express specific emotions within their messages.

The scope of this research does not encompass the development of a standalone messaging web application. Rather, the project relies on existing messaging applications, social media platforms, and email services and attempts to extend them to demonstrate the feasibility and potential benefits of integrating handwriting and emotional expression in web-based messaging.

1.3 Thesis Organization

This thesis is structured into several chapters, each focusing on a different aspect of the research. Chapter two provides background information for understanding the context of the research, and covers previous works and relevant literature. Chapter three covers the design and implementation of the system. It details the design goals, system overview and design, the details of implementing the system, and the flow of the system as a whole. It also covers an evaluation conducted for the system and its findings. Chapter four is the conclusion, providing a summary of the research, the limitations of the study, and suggestions for future work.

Chapter 2

Background

2.1 Background

2.1.1 Personalization in Digital Experiences

Technological advancements of the twenty-first century have brought about a revolution in digital communication, with digital text being used as a means to communicate messages over messaging applications, email services, and social media platforms. As of April 2024, there are 5.44 billion internet users worldwide, amounting to 67.1 percent of the global population, and of this total, 5.07 billion, or 62.6 percent of the world's population, are users of social media platforms [23]. We can thus interpret that over half of the global population has been and continues to be exposed to some sort of digital communication over the web – which in turn brings forth the need for the personalization of this communication experience.

With technological revolution, there can be an anti-human tendency that emerges with an increase in consumerism that leads to the depersonalizing of the technology at hand. As Stroud [24] interprets Kant's views, technology, while a manifestation of human freedom, can also become a form of domination. This domination can lead to a depersonalization of technology, where it no longer serves as a tool for human freedom, but rather dictates our behaviors and choices. Similarly, Bock [3] discusses Hegel's perspective on the mechanization of labor in industrial society. This mechanization, often driven by consumerism, can lead to a loss of individuality and freedom, reducing human roles to mere cogs in the machine. In this context, technology can become alienating, stripping away the human touch and leading to a sense of depersonalization, and as we continue to advance technologically, it is crucial to remember the human element, to ensure that technology serves us, rather than the other way around.

As digital text becomes more prevalent, the human touch in communication can become lost. This is particularly evident in the use of standardized fonts across various platforms, which lack the personal nuances and individuality inherent in human handwriting. This depersonalization can lead to a sense of detachment in digital communication, potentially reducing the emotional impact and personal connection that are often present in face-to-face interactions.

The need for personalization of digital experiences is highlighted via the increasing adoption of personalized experiences in various digital platforms, such as personalized recommendations on streaming platforms, personalized learning paths in online education, and personalized user interfaces in software applications. Also indicative of the importance of personalization is the valuation of the worldwide market for personalization software which, in 2022, was estimated to be over 943 million U.S. dollars and is expected to rise from 1.16 billion dollars in 2023 to an estimated 5.16 billion dollars by 2030 [22].

Digital communication experiences are thus inherently vital to be personalized. This is evident in the widespread use of custom themes in chat applications, emojis, stickers, and GIFs, intended to allow users to express their personalities and emotions in a more nuanced way than plain text allows. However, while these elements enhance the expressiveness of digital communication, they do not fully address the need for personalization and conveying unique personal touches in on-screen text.

2.1.2 Bitcoin and the Blockchain

The concept of a blockchain was originally conceptualized in 2008 by Satoshi Nakamoto along with the proposal of Bitcoin in the original Bitcoin white paper [18]. Bitcoin was intended to be a peer-to-peer, electronic cash system, allowing the exchange of currency from one party to another without the need for an intermediate financial body to process the transaction. The underlying infrastructure associated with this proposed Bitcoin system is known as a blockchain, a decentralized and distributed digital ledger that records transactions across many computers (nodes) so that any involved record cannot be altered retroactively, without the alteration of all subsequent blocks.

In 2009, Bitcoin was finally launched as the first cryptocurrency that utilized blockchain technology, with one of its primary innovations being that it allowed two parties to transact with each other directly, without the need for a trusted third party like a bank. This was made possible by the use of complex mathematical and cryptography algorithms and consensus protocols that ensure all transactions are verified and recorded on the blockchain in a secure and transparent manner. As of 2024, Bitcoin has indicate the success here with a reference. The success of Bitcoin sparked a wave of innovation in the field of cryptocurrencies and blockchain technology. Thousands of alternative cryptocurrencies and blockchain-based networks have been created since then, each with its own unique features and uses.

2.1.3 Ethereum

One of the most notable among these successors of Bitcoin is Ethereum, which was proposed by Vitalik Buterin [5] in 2014 and launched in 2015. Unlike Bitcoin, which is primarily a digital currency, Ethereum is a more advanced blockchain project that enables the development of Decentralized Applications (dApps) using smart contracts.

Smart contracts are self-executing contracts with the terms of the agreement directly written into lines of code. They automatically execute transactions when their conditions are met, eliminating the need for a trusted third party. They can be considered as decentralized, immutable back-ends. Each smart contract comes with its own storage and its own set of functions that can be invoked to perform specified operations. The storage is where the contract's data is stored, and it persists between function calls and transactions.

Smart contracts can interact with other smart contracts, send and receive Ether (the native cryptocurrency of the Ethereum network), and even create new smart contracts. They are executed by the Ethereum Virtual Machine (EVM), which runs on every node in the Ethereum network.

The code of a smart contract is immutable once it's deployed to the Ethereum network. An advantage of such is that it means that once a contract is deployed, its behavior can't be changed, which provides a strong guarantee to all parties interacting with the contract that it will continue to behave as expected. This is particularly important in scenarios where trust is essential, such as financial transactions or when dealing with sensitive user data. However, this immutability also means that if there are bugs or vulnerabilities in the contract code, they can't be fixed by simply updating the code. Therefore, it's crucial to thoroughly test smart contract code before deploying it to ensure it behaves as expected and does not contain any exploitable bugs.

Smart contracts can be written in a few different languages, but one of the most common one is Solidity [1]. Solidity is a statically typed programming language designed specifically for writing smart contracts on Ethereum. It's influenced by C++, Python, and JavaScript, and is designed to target the EVM.

Using smart contracts, dApps can be developed in cases where a decentralized back-end is needed, as it would run on the Ethereum blockchain, leveraging its decentralized nature and the capabilities of smart contracts. Web pages for a web application, for example, can be hosted on a traditional server, but communicate with a smart contract to fetch data instead of a traditional API server. To interact with the Ethereum blockchain from a browser in this case, however, users typically need a blockchain wallet, such as MetaMask.

MetaMask is a browser extension that allows users to interact with the Ethereum blockchain, including its smart contracts, directly from their browser. It acts as a bridge that connects users to the Ethereum network without running a full Ethereum node. It also securely manages the user's private keys, can sign blockchain transactions, and send and receive Ether.

In the context of Ethereum, a provider is an abstraction layer for a connection to the Ethereum blockchain. It's responsible for transporting requests and responses between an application and the Ethereum network. When an application wants to read data from or write data to the Ethereum blockchain, it does not communicate with the Ethereum network directly. Instead, it sends a request to a provider. The provider then communicates with the Ethereum network, gets the response, and sends it back to the application.

MetaMask injects a global API into all websites visited by its users at window.ethereum. This allows the development of websites that can use this API to request users' Ethereum accounts, read data from the Ethereum network the user is connected to, and ask the user to accept the transactions using their wallet. The presence of the provider object indicates an Ethereum user. Thus, providers like MetaMask play a substantial role in the Ethereum ecosystem and the development and use of dApps. It is worth noting that transactions to interact with the blockchain, like writing some data to the blockchain, require an account to be sent from. This is where MetaMask comes in, as it also acts as a wallet that holds Ether currency that is used to pay for these transactions.

2.1.4 InterPlanetary File System (IPFS)

The InterPlanetary File System (IPFS) is an open-source protocol and peer-to-peer network that utilizes a distributed file system for storing and distributing data on the internet. Instead of centralized storage, content published on the IPFS is stored and duplicated on multiple nodes on the network. This addresses the problem of censorship and targeted removal of certain content since the removal of content from the IPFS network would require the removal of that content from all nodes that host it. Instead of using a URL to request a specific resource, such as a file, IPFS relies on content-addressing by assigning unique Content Identifiers (CIDs) to all published content, and thus the resource is requested with that specific identifier (which is in the form of a hash). The IPFS network then finds the fastest, closest, or least expensive way to get that file to you [12].

2.2 Related Works

In typography, the choice of typeface is not merely a matter of aesthetics. Research has increasingly highlighted the psychological impact of typefaces, suggesting that they can carry inherent personality traits, invoke human emotions, and significantly influence consumer behavior. The following studies provide compelling evidence of these phenomena.

2.2.1 Font Effects on Perceived Personality Traits

Research papers have consistently shown evidence of the relation between certain typefaces or fonts and perceived personality traits, and fonts can even be clearly categorized based on their perceived traits, as has been shown by Brumberger [4], Shaikh, Chaparro, and Fox [20], and Mackiewicz and Moeller [17]. Brumberger has also found that "the data supports theoretical perspectives that suggest that visual language is analogous to verbal language in carrying connotations" [4]. The finding that the font of a certain body of on-screen text can carry its own meaning and can affect the reader's perception is a recurring one. For instance, in a study by Fox, Shaikh, and Chaparro, it was found that the use of certain fonts in documents affects how the reader perceives both the document itself and its author [9]. In another study done by Mackiewicz and Moeller, it was found suggestively that people naturally differentiate between various typefaces and the personality traits they are related to, and that this is often done in an unstructured way, relying more on intuition than analytical evaluations of a typeface's anatomical features [17]. Li and Suen have also proven the association of personality traits with typefaces and the potential association between those traits and the typeface's design and aesthetic characteristics [16].

	Top Three		
Stable	TNR	Arial	Cambria
Flexible	Kristen	Gigi	Rage Italic
Conformist	Courier New	TNR	Arial
Polite	Monotype Corsiva	TNR	Cambria
Mature	TNR	Courier New	Cambria
Formal	TNR	Monotype Corsiva	Georgia
Assertive	Impact	Rockwell Xbold	Georgia
Practical	Georgia	TNR	Cambria
Creative	Gigi	Kristen	Rage Italic
Happy	Kristen	Gigi	Comic Sans
Exciting	Gigi	Kristen	Rage Italic
Attractive	Monotype Corsiva	Rage Italic	Gigi
Elegant	Monotype Corsiva	Rage Italic	Gigi
Cuddly	Kristen	Gigi	Comic Sans

Figure 2.1: Top three fonts for each personality adjective from Shaikh, Chaparro, and Fox [20]

2.2.2 Product Perception Effects of Fonts

Research has demonstrated the effects of using handwritten fonts on consumers and the marketing domain, which can be through affecting the perception of products. Flinger [8] has shown that consumers can reliably discern semantic attributes, such as naturalness, linked to a specific font style, and the implications derived from the typefaces used on product packaging do influence consumers' perception and assessment of the product.

Following such a train of thought, the use of handwritten fonts for the presentation or packaging of products can be examined. Multiple research papers have discussed the effects of using handwritten fonts in particular in product perception.

Schroll, Schnurr, and Grewal have shown that using handwritten fonts increases consumers' positive evaluation of products by creating human presence perceptions which, in turn, increases emotional connections to products [19]. For products such as souvenirs, Guo, Cui, and Zhao have shown that the use of handwritten typefaces for their packaging design enhances tourists' emotional attachment to them, resulting in more favorable evaluations of such products [10]. The authors state that these findings are "consistent with the notion that handwriting, accompanied by sincerity, effort, and genuineness, signals authenticity" and "demonstrate the importance of the sense of human presence as a mediator". Other products have also received similar research with similar results. For example, it has been found that the use of a certain font in a restaurant menu can cause a heightened sense of openness in the environment, influencing human personality ratings [25]. In the education domain, a study done by Cross et al. [6] has shown that individuals prefer the use of a combination of handwriting and machine typefaces over using pure handwriting or machine typefaces in online learning videos, making the content more digestible and interactive for the learners.

Figure 2.2: Product Packaging Featuring a Handwritten (Left) and Machine-Written (Right) Typeface [19]

Figure 2.3: Product Packaging Featuring a Handwritten (Left) and Machine-Written (Right) Typeface [19]

Figure 2.4: Product Packaging Featuring a Handwritten (Left) and Machine-Written (Right) Typeface [19]

Figure 2.5: Product Packaging Featuring a Handwritten (Left) and Machine-Written (Right) Typeface [19]

Figure 2.6: Product Packaging Featuring a Handwritten (Left) and Machine-Written (Right) Typeface [19]

2.2.3 Congruity Between Fonts and Products

Doyle [7] has shown that the congruity between fonts and products resulted in rating the products as more appropriate and that individuals were more likely to pick a certain brand if the used font had an "obvious" association with the product's category. Flinger [8] has shown that the impact of the font is influenced by whether the associations derived from the font align with the information conveyed by the product's "intrinsic and extrinsic cues".

2.2.4 Persuasive Effects of Fonts

In the marketing domain, it has been shown by Kim, Jung, and Kim [14] that consumers can perceive text in an advertisement that is displayed in a font such as Sans Serif to be more "powerful" than a handwritten typeface, meanwhile the message is seen as more engaging and with a higher probability to elicit engagement from the consumers if it uses handwriting, and this engagement would then increase the favorable behavior towards it.

2.2.5 Emotional Effects of Fonts

Juni and Gross [13] have also stated that the subtle meanings conveyed by different font styles can influence how people interpret and feel about the text. This can subtly enhance the persuasive power of a text, which has important implications for fields like marketing and advertising. Furthermore, Hazlett et al. [11] have indicated that a typeface that aligns with some on-screen text's emotional meaning can facilitate the affective classification of that text and delivery of its intended meaning. Furthermore, research has built upon these established findings, as Kulahcioglu and de Melo 2020 [15] have established a system for searching for fonts using a reference font as a starting point, and a search query such as "fonts like this, but happier" - further solidifying the association of fonts with emotions. In addition, Bianchi, da Hora Rodrigues, and de Almeida Neris have shown that font sizes in combination with font styles trigger various emotions in users and affect their decision-making, preferences, and viewpoints [2].

2.3 Summary

In this chapter, we have discussed, in Section 2.1, background information regarding personalization of digital experiences, Bitcoin, blockchain technology, Ethereum, and IPFS. In Section 2.2, we have discussed related works that demonstrate font effects on perceived personality traits, product perception effects of fonts, congruity between fonts and products, persuasive effects of fonts, and emotional effects of fonts. The discussed related works present findings that indicate the ability of digital messages displayed in certain fonts to invoke certain emotional and personal correlations in individuals, as well as carrying additional context and meaning in the message. This makes way for the use of handwritten fonts in digital communication. The personal and emotional connections that handwritten fonts can create, as evident by the works discussed herein, can be used to make digital communication more personal and engaging, and also bring up the possibility of not only creating one font for each individual but also creating different variants of the user's handwriting to match certain emotions such as happiness or anger, that is different from the neutral font of the user's handwriting.

Chapter 3

System Design and Implementation

3.1 Design Goals

The design goal was to implement a complete system, with a heavy prioritization on security, to generate a handwritten font from the user's handwriting, as well as emotional variants of the generated font, and allow the user to send and view messages in these generated fonts. Due to the nature of the project in dealing with font files corresponding to real individuals' handwriting, which is to be considered sensitive user data, the security requirements of the developed system dictated the design goals and choices.

3.2 Design Strategy

3.2.1 Font File Generation

The design goals dictated that a system be implemented that eliminated the use of server-side actions with the aforementioned sensitive user data, which would otherwise require client-server communication with back-and-forth communication to send the user data over the network, which, in turn, posed a security risk to such user data. The use of encrypted communication over Hypertext Transfer Protocol Secure (HTTPS) or developing a system to manually encrypt the data being sent over the network still exposes the user's data while the transfer is occurring. Thus, the generation of the fonts needed to be done locally on the user's machine, where no outgoing network requests would be performed and all the data would be inputted and processed directly on the user's device.

3.2.2 Font File Storage

The problem that follows the processing and generation of font files would be where to store the files for retrieval and use in our application. Using a centralized database would pose the risk of potentially exposing the font files of all system users in the event of a data leak. Furthermore, a centralized database would act as a single point of failure in which the persistence of the users' data depends on the reliability of the database systems, and incidents of data loss or corruption could mean the loss of all user data. Finally, utilizing a centralized database would dictate that one entity is to be entrusted with the entirety of the user's handwritten font data, which the user has no say over in how it is stored, encrypted, or used. For these reasons, a decentralized file storage system was deemed fit.

3.2.3 Back-end Verification

Closely tied to the problem of storage is the problem of processing requests. Sending and receiving messages requires maintaining records of these sent messages and managing permissions to allow the intended recipient to fetch the font file that the message is intended to be displayed in, which requires the implementation of a back-end that cannot be tampered with on the client-side to serve such requests. Due to the sensitive nature of handwritten font files, an immutable and predictable back-end needed to be developed – one that users would know wouldn't unexpectedly change its behavior or policies after the user data had already been collected, all of which would be violated by creating a server. Thus, smart contracts on blockchain were the solution to the problem, due to their immutable nature and the availability of their byte code permanently on the blockchain, visible to all individuals.

3.2.4 Web-based User Interface

What remained was a tool to be developed to provide the users with a web-based interface to interact with the rest of the system components. This would include handling the uploading of font files, sending messages, and displaying messages in handwritten fonts. A browser extension was thus seen as the solution to this problem, allowing the displaying of handwritten messages to be done on any website, utilizing the scripting privileges of browser extensions. The idea is to utilize the existing infrastructure of messaging applications, email services, social media platforms, and more, and develop a method for sending messages through these applications attached to some metadata that indicates their handwritten font to be displayed in, with the help of the browser extension.

3.3 System Overview

The system consists of four major components intended to achieve the aforementioned design goals and aligning with the highlighted design strategy.

3.3.1 Desktop Application

The first major component of the system is a Desktop Application that is responsible for the generation of the handwritten font file from the user's handwriting as well as emotional variants of that font. The application does all the data processing and generation locally on the user's machine without sending or receiving any data over the network. This achieves the intended design goal of eliminating the process of sending user data to be processed on a server – eliminating the use of a server entirely.

3.3.2 InterPlanetary File System (IPFS)

The second major component of the system is the IPFS, which is used by the user to upload the generated font files to. This achieves the intended design goal of using a decentralized file storage system, due to the decentralized, peer-to-peer nature of the IPFS, and thus also removing the need for a server.

3.3.3 Ethereum Smart Contracts

The third major component of the system is the Ethereum Blockchain. Smart contracts on the Ethereum Blockchain were utilized as a verifiable, immutable back-end that is intended to achieve the design goal of developing a back-end to verify message sending between users and allowing only the intended recipients to view messages in the intended fonts, all while avoiding the use of a centralized server.

3.3.4 Chrome Extension

The final major component of the system is a Chrome extension developed for the Chrome browser. The choice of developing the extension for Chrome was due to the fact that the Google Chrome browser is the most used browser worldwide [21]. It achieves the design goal of developing a web-based user interface that interacts with the rest of the system components and allows the user to upload font files, and send and display messages in handwritten fonts.

3.4 System Design

The user's interaction with the system begins with the desktop application. The user is greeted with a simple home page, wherein they can navigate to the first page for font creation. They begin by saving the template to their device as shown in Figure 3.1.

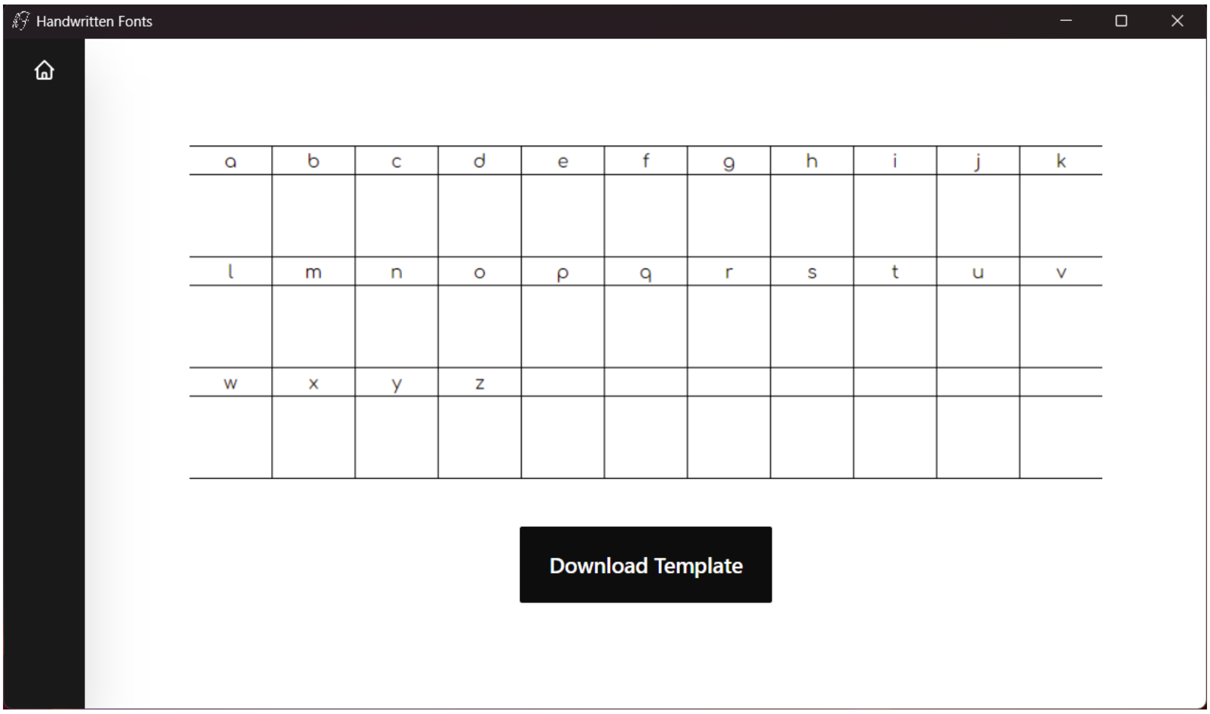

Figure 3.1: Saving Template Image

The user prints out this template and fills it with their handwritten characters. Next, the user uploads the filled template image to the application as shown in Figure 3.2.

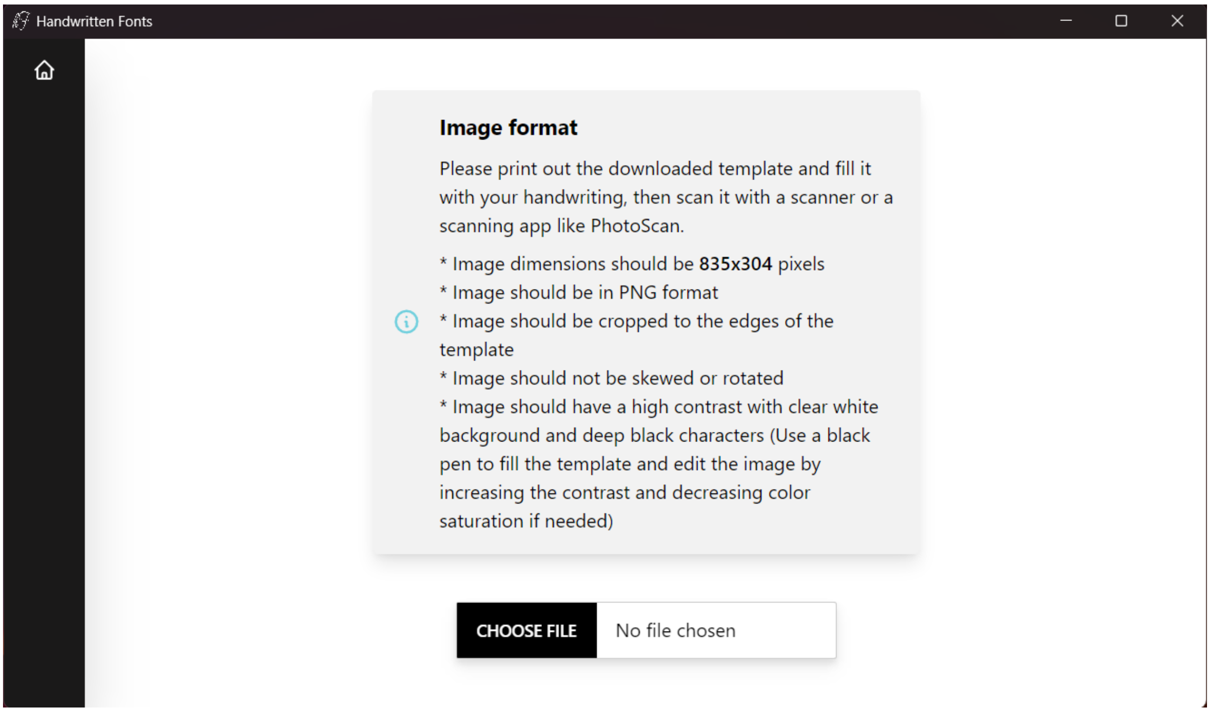

Figure 3.2: Uploading Filled Template Image

The application then proceeds to process the selected image with no needed input from the user. Once this processing is done, the user is navigated to a new page where they can select their desired flavors/variants of the font. They can choose the neutral font, which is the identical one to their handwriting, as well as emotional fonts which are altered versions of the user's handwriting with an added touch of a certain emotion, based on certain reference fonts. See Figure 3.3.

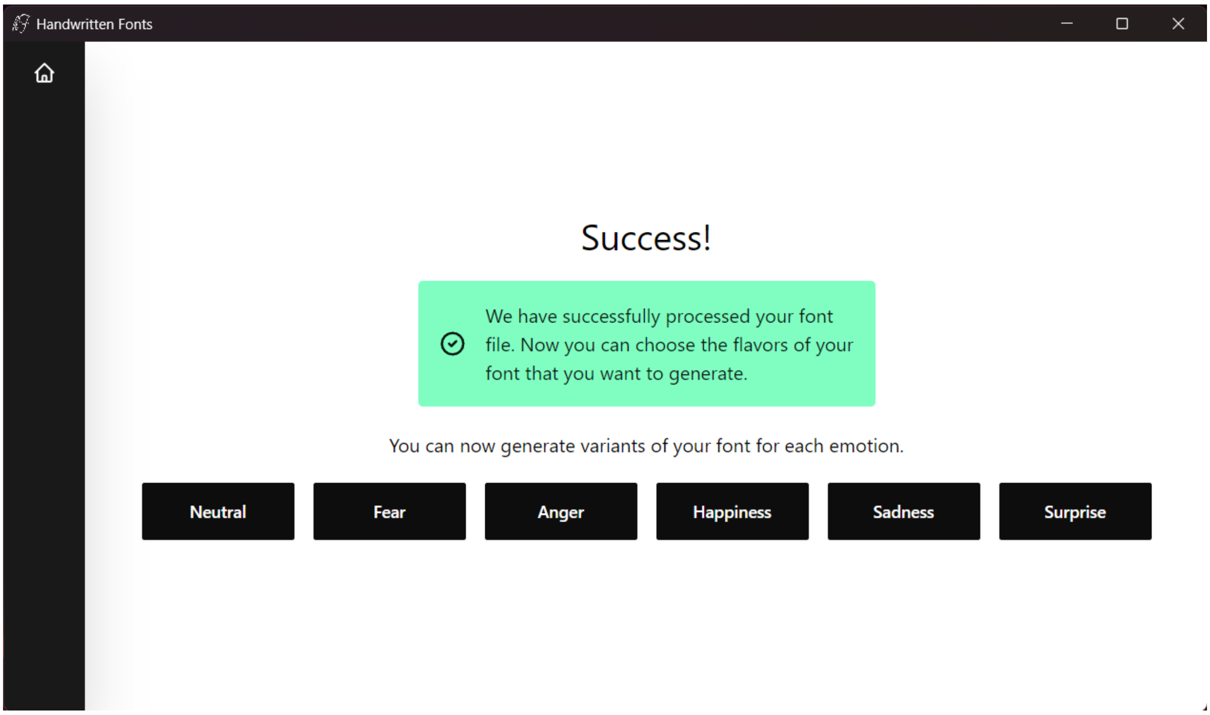

Figure 3.3: Selecting Font Variants

That concludes the desktop application's role in the system. The user then moves on to their Google Chrome browser to use the system's Chrome extension. When the user first opens the extension, it checks for the installation and presence of the MetaMask extension on the user's browser and requests that the user connect their MetaMask account(s).

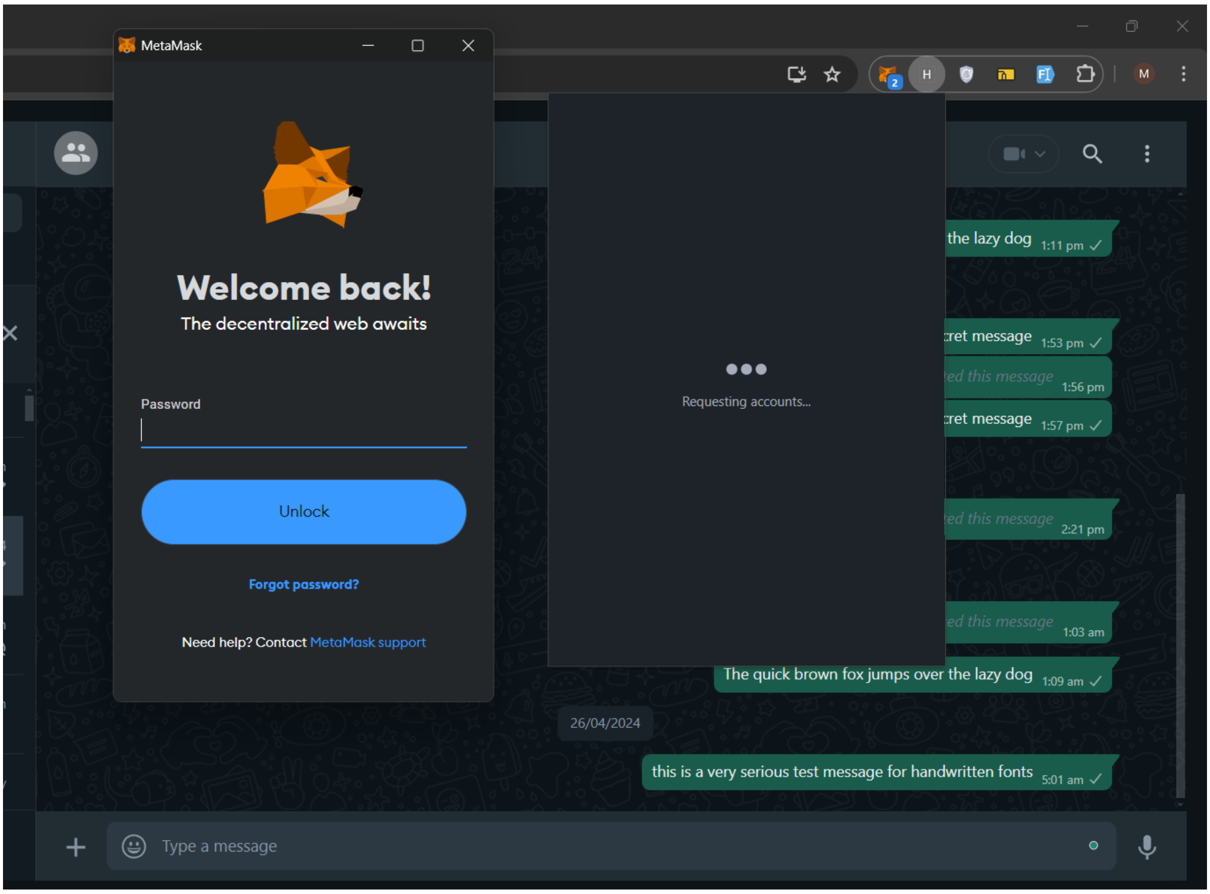

Figure 3.4: Starting up the extension and connecting to MetaMask (WhatsApp web in the background)

Upon logging into MetaMask and connecting at least a MetaMask account, the user is able to go proceed in the developed extension. Any selected accounts will appear in a drop-down for the user to select from in the extension, in which he will be able to use as an account to login or sign-up with in the Handwritten Fonts extension.

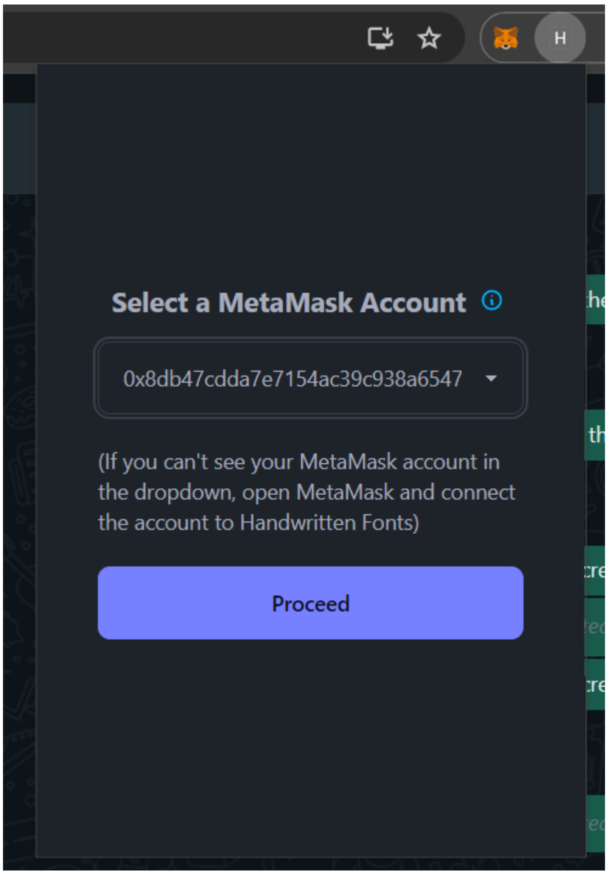

Figure 3.5: Selecting a MetaMask account in the Handwritten Fonts extension

After logging in, the user can use the extension. The next step is to add a contact through the contacts page. A contact is identified with a MetaMask account address and a friendly name that the sender can remember.

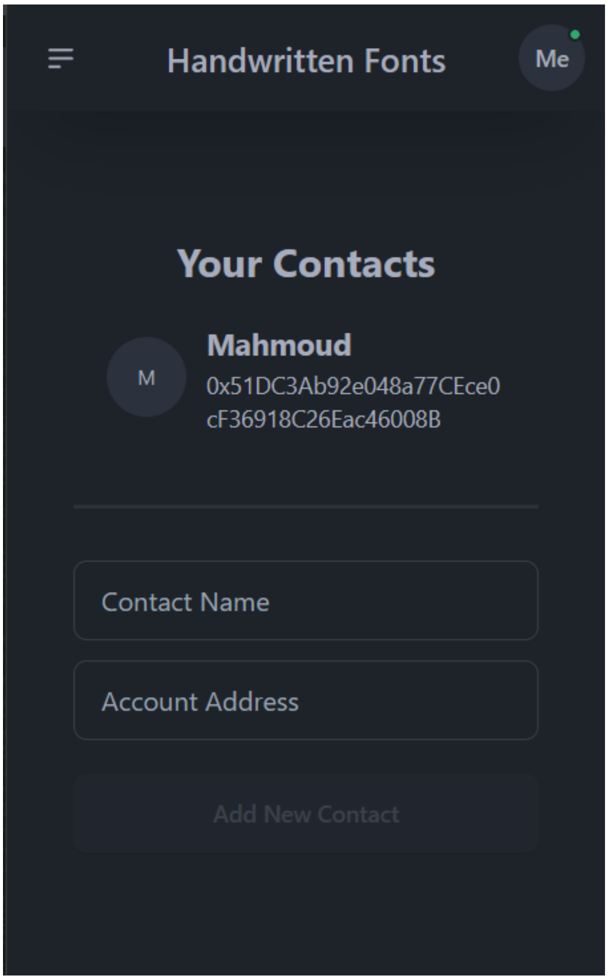

Figure 3.6: Contacts Page

Next, the user needs to upload at least one font file before being able to send messages. The user can upload one font file for each of the emotions supported by the system, as well as the neutral font.

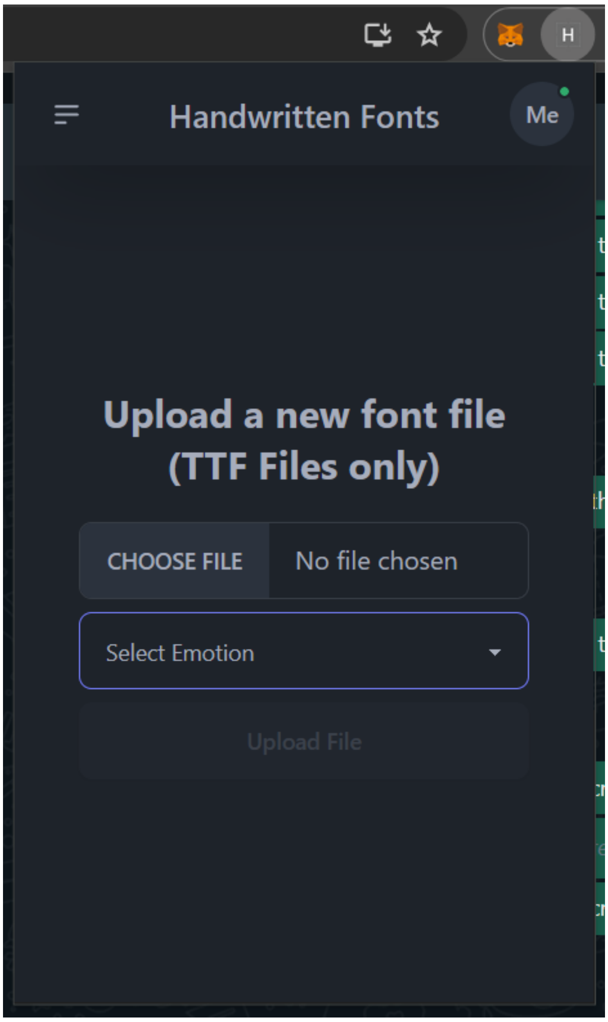

Figure 3.7: Uploading a new font

Next, the user needs to "send a message" to a recipient contact. This is effectively just recording a transaction of sending a certain message to a certain recipient, allowing only that recipient to view that message in the handwritten font, but it does not send them the message. Instead, it generates a message with appended metadata that the sender can then copy and send via any messaging platform.

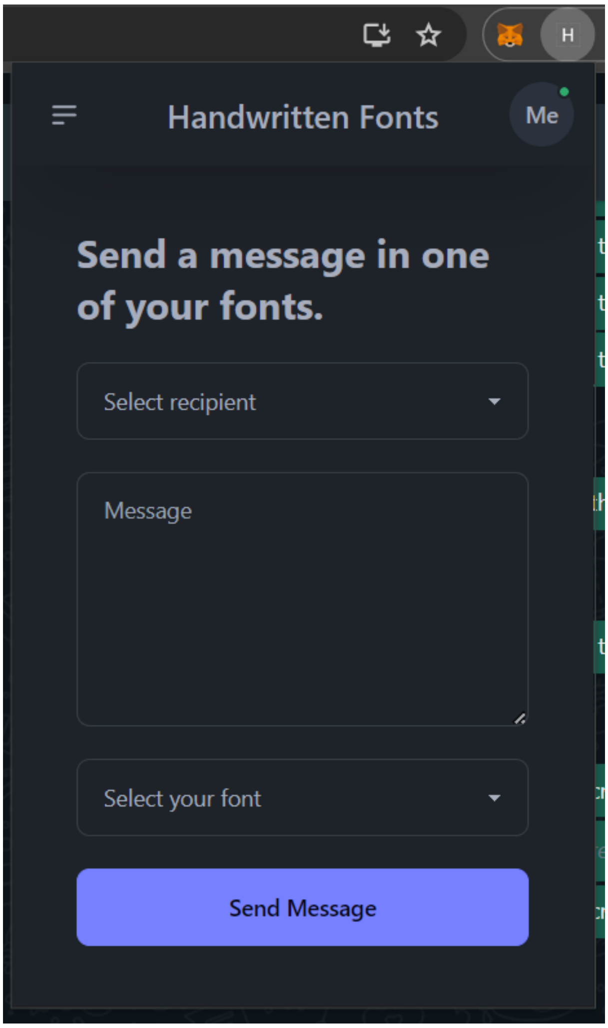

Figure 3.8: Logging a message-sending transaction to a certain recipient MetaMask account

Figure 3.9: Generated message with appended metadata, to be copied by the sender

The recipient can then load up any web page that contains that sent message, and the extension will be able to detect it. The recipient must also have MetaMask and the Handwritten Fonts extensions installed and used. The recipient can then click a button to display any detected messages on the current web page in their intended handwritten font. Finally, the message gets displayed in the intended font.

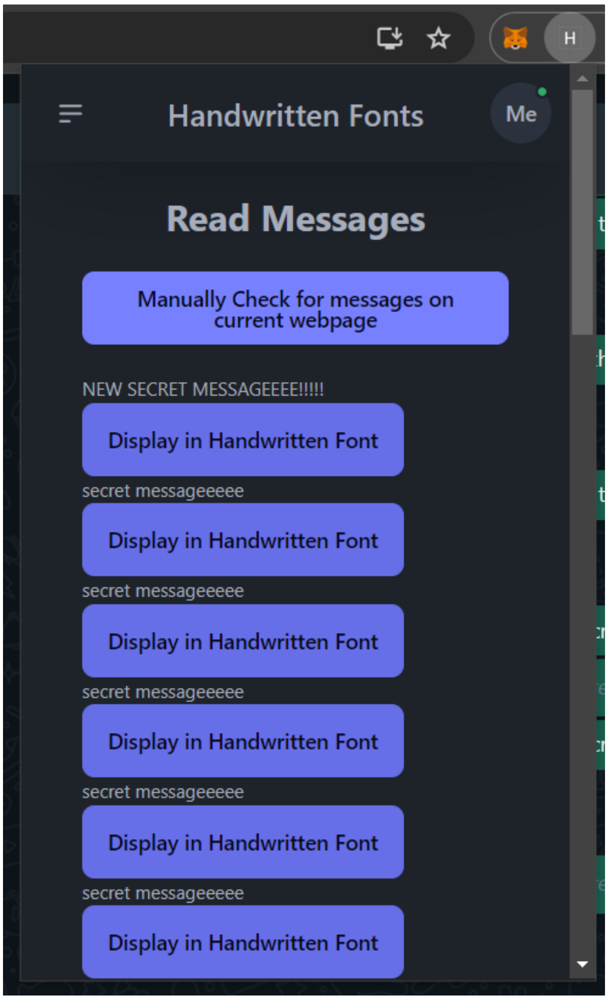

Figure 3.10: Detected Handwritten Fonts messages on the current web page

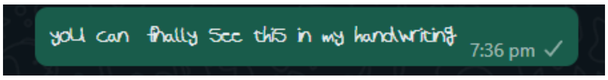

Figure 3.11: Handwritten Message on WhatsApp Web.

3.5 Implementation

The implementation details for the major components of the system shall be highlighted herein, by following the flow of the whole system from the generation of the font files till sending messages and examining the underlying operations at each stage. It will also cover in detail the heavy encryption system that is in place for the security of the system.

3.5.1 Generation of Font Files

The standalone desktop application was developed to handle font file generation locally on the user's machine as follows. The user saves a template image from the desktop application that contains spaces for the twenty-six lowercase English letters, then prints it out and fills it with their handwritten characters. Then the user scans the image with a scanner (a physical scanner or a scanner app like PhotoScan) and then uploads the scanned image to the desktop application. The application then proceeds to perform preprocessing on the image before automatically cropping out the letters from the image, including binarization of the image to only have black and white colors.

The application defines preset locations on the image using pixel coordinates to determine where the letters are, which means that the scanned image should be nearly identical in dimensions to the original template image and should not have any skewing or rotation. The cropped letters are then processed individually to remove any extra white space around them and then converted to glyphs and used to generate a [**?**] file, which is the file extension for fonts that we use for this project. This generated font file is considered as the "Neutral" font for the user, and then the user is prompted to select which other emotional variants of the font he would like to generate, with the available variants being for the following emotions: Happiness, Sadness, Anger, Fear, and Surprise. Once a variant is selected, the generated neutral font file is then combined with a predefined reference font file for the selected emotion, and an intermediate font is generated that resembles both the user's original handwritten font and the reference font, adding a touch of the desired emotion. To do this mixing between the two fonts, a technique known as interpolation is carried out between the characters of both fonts to generate intermediate characters.

The desktop application's views were created with React.js, a popular front-end library for web development, and the font generation was handled by a standalone Java application using the Fontastic library, which is used to generate font files. The FontForge library was used for font interpolation, specifically leveraging its scripting language for interpolating two fonts. Electron was utilized as the framework to build the desktop application, combining web-based front-end technologies (React.js), the font generation logic (contained in the Java application), and the interpolation logic (contained in FontForge scripts) by allowing the JavaScript code running on the desktop application's front-end to run the Java application or the FontForge script as a child process on the user's Operating System and pass necessary data and communication via command-line arguments.

3.5.2 Handwritten Fonts Smart Contract

The smart contract entitled "HandwrittenFontsContract" was created on the Ethereum network to serve as the needed back-end for the system. It was written using the Solidity language and deployed on the Ethereum Sepolia test network. It was used instead of the Ethereum main network as the main network uses a real currency that amounts to an actual value in the real world, whereas test networks allow us to use fake currency to test out our applications. The source code for the contract can be seen below.

```
// SPDX-License-Identifier: MIT

pragma solidity 0.8.9;

/**
 * @title HandwrittenFonts
 * @dev A contract for the Handwritten Fonts project
 * @author Mahmoud Abou Eleneen
 */
contract HandwrittenFonts {
    struct MessageRelatedData {
        string encryptedSymmetricKey;
        string encryptedCidOfEncryptedFontFile;
        string encryptedFilenameOfEncryptedFontFile;
    }

    // Mapping from message hash to encrypted data relevant to the message
    mapping(string => MessageRelatedData) public messageHashToEncryptedData;

    // Mapping from account address to password creation status
    mapping(address => bool) private passwordCreated;

    // Mapping of address to their public key
    mapping(address => string) public addressToPublicKey;

    /**
     * @dev Store the encrypted data needed to display a message in a certain font
     * @param _messageHash the hash of the message
     * @param _encryptedSymmetricKey the encrypted symmetric key used to encrypt the font file
     * @param _encryptedCidOfEncryptedFontFile the encrypted IPFS CID of the font file this message should be displayed in
     * @param _encryptedFilenameOfEncryptedFontFile the name of the font file this message should be displayed in
     */
    function storeEncryptedData(
        string calldata _messageHash,
        string calldata _encryptedSymmetricKey,
        string calldata _encryptedCidOfEncryptedFontFile,
        string calldata _encryptedFilenameOfEncryptedFontFile
    ) external {
        messageHashToEncryptedData[_messageHash] = MessageRelatedData(
            _encryptedSymmetricKey,
            _encryptedCidOfEncryptedFontFile,
            _encryptedFilenameOfEncryptedFontFile
```

```
        );
    }

    /**
     * @dev Update the password creation status for an account
     * @param _status the new status of the account's password
     */
    function updatePasswordStatus(bool _status) external {
        passwordCreated[msg.sender] = _status;
    }

    /**
     * @dev Check the password creation status of the sender's account address
     */
    function checkPasswordStatus() external view returns (bool) {
        return passwordCreated[msg.sender];
    }

    /**
     * @dev Store the public key of the sender's account address
     * @param _publicKey the public key to be stored
     */
    function storePublicKey(string calldata _publicKey) external {
        addressToPublicKey[msg.sender] = _publicKey;
    }
}
```

The details of the smart contract will be discussed more along with each of the following subsections to maintain a clear purpose of its parts.

3.5.3 Login and Sign Up

A Chrome extension was created to act as the interface for the user to interact with the main system functionalities. It was created using the React framework. Its functionality depends on the user having the MetaMask extension installed to have an Ethereum wallet and method of interaction with the Ethereum network, and the Handwritten Fonts extension uses this MetaMask provider to connect to the Ethereum network and make calls to the deployed smart contract. The developed extension uses the "Web3.js" JavaScript library as a portal into the Ethereum network, using the MetaMask provider.

The user is first prompted in the extension to select a MetaMask account to use as his Handwritten Fonts account. The extension prompts the user, through the MetaMask popup, to select an account or more to connect to the Handwritten Fonts extension. The extension will then be able to detect the accounts that the user connected, and the user can create a Handwritten Fonts account with any of them. In essence, MetaMask accounts were used along with their account addresses as unique identifiers for accounts in the developed extension.

Once the user selects a MetaMask account to use as his Handwritten Fonts account, they are prompted to create a password, and a call to the smart contract is made to update a mapping of account addresses to account creation status. We use this mapping to later check if the user should be signing in or creating a new password for a given account address. This call, and other calls to the Ethereum network, require the use of the MetaMask provider as it will incur any gas fees on the user.

Upon creation of the password, a public and private RSA key pair is generated by the extension using a JavaScript library entitled "JSEncrypt", where the public key gets stored on the Ethereum network in a mapping of account addresses to public keys, and the private key gets encrypted by the created password and stored on the extension's local storage on the user's device.

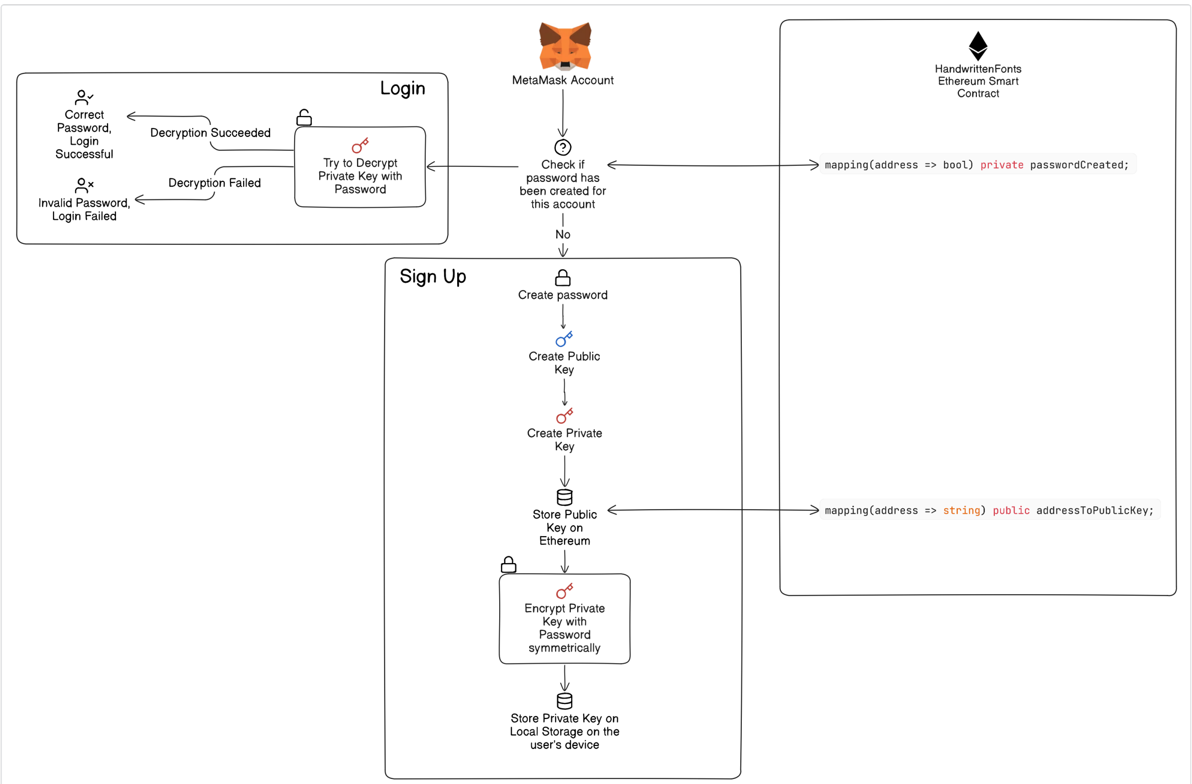

Figure 3.12: Login and Sign Up Flow

3.5.4 Uploading Font Files

For font file storage, the IPFS was used as a decentralized, peer-to-peer file storage system. A service called "NFT.Storage" was used to connect to a node on the IPFS network. It provides a "pinning" service by running multiple dedicated nodes on the IPFS network and creating copies of the uploaded content on these nodes, effectively pinning the uploaded content. This is necessary as there must at least be one node on the network that has a resource for it to be accessible.

The user is able to upload a generated font file for each emotion. The upload process works by first generating a symmetric key using a JavaScript library entitled ”CryptoJS”, and encrypting the font file with this symmetric key. The extension then proceeds to upload the encrypted font file to the IPFS network through the API Gateway of the ”NFT.Storage” service. The symmetric key is then encrypted along with the returned content identifier (CID) of the stored file on the IPFS network and the file name, and then they are all stored in the local storage of the extension.

After uploading at least one font file, the user can send messages to certain individuals who have also installed and linked an account on the developed extension, by referencing their MetaMask account address.

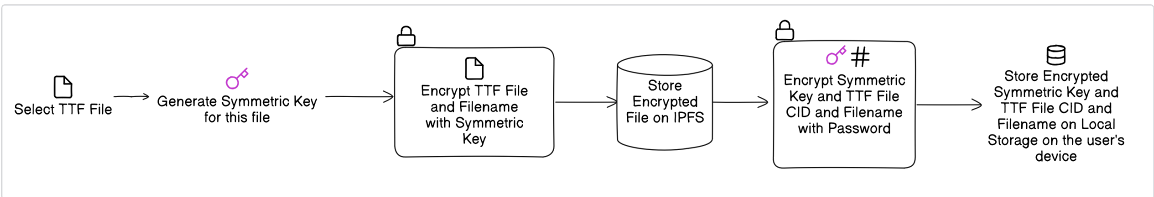

Figure 3.13: Font File Upload Flow

3.5.5 Sending Messages

A contacts feature was developed to improve user experience and limit the chance of error in entering the account addresses of recipients. The user can create a contact with an account address and an easily recognizable name. The user can then select a recipient contact, type out the desired message to send, and select the font in which the message should be displayed.

The needed font file CID, filename, and symmetric key are then fetched from the user’s local storage and decrypted with his password. Then, a call to the Ethereum network is made to retrieve the public key of the recipient and encrypt the aforementioned decrypted data. This data will then only be decrypted using the recipient’s private key.

A mapping of message hash to encrypted font data was then utilized on the smart contract for this data retrieval by the recipient. A Universally Unique Identifier (UUID) is generated and concatenated with the message then hashed to form a unique hash using the SHA-3 algorithm through the ”CryptoJS” JavaScript library. The UUID is necessary to be used in combination with the message to ensure a unique hash that only identifies this message is guaranteed.

Thus, this message hash is stored in the mapping to the data encrypted with the recipient’s public key. Finally, the sender is given the message, presented in a specific format that includes the message, a special tag, and the UUID. The sender can then proceed to copy this message from the extension and send it over any web application with messaging services to the recipient.

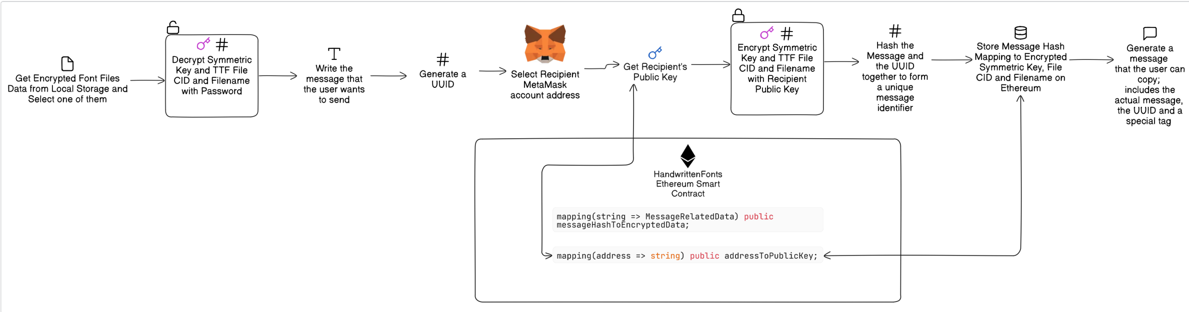

Figure 3.14: Message Sending Flow

3.5.6 Displaying Messages

The recipient can then load up any web page that displays the message sent by the sender in the specified format, and it will be detected automatically by the developed extension in the background. The recipient is then prompted with the detected messages and whether they would like to view them in the handwritten font.

In such case, the recipient's private key is decrypted using his password, then the message gets combined with the UUID sent as part of the message to produce the same message hash to fetch the encrypted font data related to this message from the smart contract. The fetched font file CID, filename, and symmetric key are then decrypted using the recipient's private key, and the font file is fetched from the IPFS network using the CID and filename via the "NFT.Storage" API Gateway. Finally, the fetched font file is then decrypted using the symmetric key, and the font gets applied to the message.

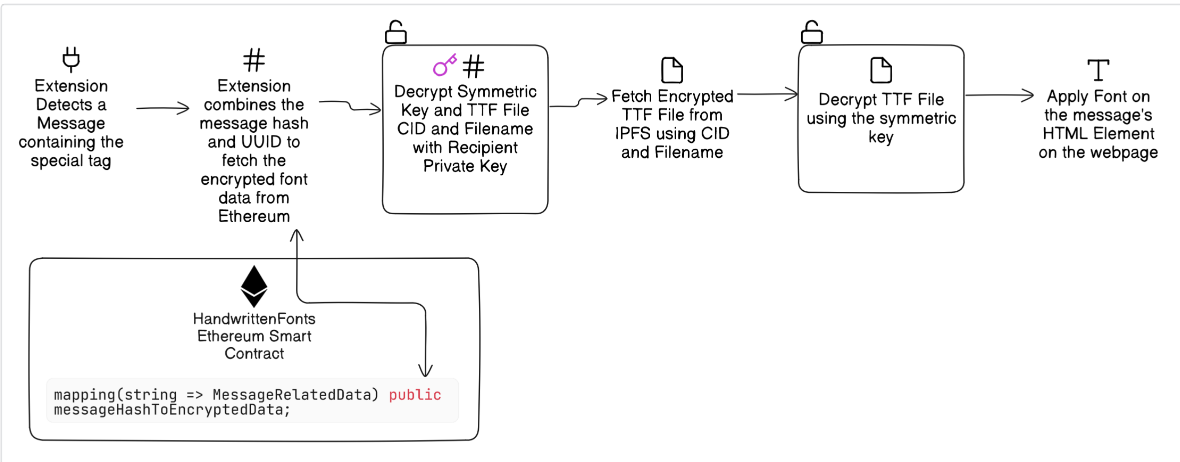

Figure 3.15: Message Displaying Flow

3.5.7 Integration between MetaMask and Handwritten Fonts Extensions

The integration between MetaMask and a normal website is straightforward. By default, MetaMask injects a window.ethereum object in each web page that the user visits, and thus a website can be easily developed to interact with that API to use the user's MetaMask account and interact with the Ethereum networks. However, the integration between MetaMask and another Chrome extension is not at all straightforward.

First of all, Chrome extensions run in their own JavaScript context. For this reason, Chrome provides a scripting API that extensions can call in their code to execute some script on the current web page. In order for the Handwritten Fonts extension to interact with MetaMask, it needs to access the window.ethereum object present on the current web page. It would then be reasonable to assume that by using the Chrome scripting API, the extension would then be able to access this window.ethereum object injected into the web page. However, this API only allows the extension to run things on the current web page, but without giving the extension access to any JavaScript variables available on the web page. For these reasons, the extension had no way of interacting with the user's MetaMask extension and thus no way of interacting with Ethereum.

The solution was to develop a communication protocol between the Handwritten Fonts extension and the MetaMask extension. First, an "injectedScript.ts" file was created. This script gets actually injected as a normal JavaScript script into the web page by the Handwritten Fonts extension and runs as an in-line script using the ¡script¿¡/script¿ HTML tag; so it has access to the actual variables in the JavaScript context of the web page. Therefore, this injected script has access to the window.ethereum object injected into the web page by MetaMask, and thus we use it to set up a configuration for communicating with the deployed Ethereum smart contract through the user's MetaMask. After setting up the configuration, the injected script sets an event listener to listen for communication from the Handwritten Fonts extension, execute whatever method calls on the Ethereum blockchain that are needed, and respond back to the extension with the response. This involved much complexity in essentially developing a communication protocol between the injected script and the extension for effective transfer of data. A simplified portion of the injected script's source code can be seen below.

```
// Use the window.ethereum object injected by MetaMask
const web3Instance = new Web3(window.ethereum);
const contractInstance = new web3Instance.eth.Contract(
    compiledContract.abi,
    config.deployedContractAddress
    );

window.addEventListener("message", (event) => {
  if (event.source !== window) return; // We only accept messages from ourselves
```

```
  if (event.data.type) {
    if (event.data.type === "CALL_ETHEREUM_METHOD") {
      callEthereumMethod(
        event.data.method,
        ...event.data.options.methodArgs
      );
    } else if (event.data.type === "CALL_CONTRACT_METHOD") {
      callContractMethod(
        event.data.method, "call",
        event.data.options.methodArgs,
        event.data.options.callArgs
      );
    } else if (event.data.type === "SEND_CONTRACT_METHOD") {
      callContractMethod(
        event.data.method,
        "send",
        event.data.options.methodArgs,
        event.data.options.sendArgs
      );
    }
  }
});

function callEthereumMethod(method: string, ...args: any[]) {
    ...
}

function callContractMethod(method: string, type: "call" | "send",
    methodArgs: any[], callArgs: any[]) {
    ...
}
```

A ”content script” is used by the extension to inject the injected script. A content script is a key part of Chrome extensions. It is a JavaScript file that runs in the context of web pages. Using the Document Object Model (DOM), it can read details of the web pages the browser visits, make changes to them, and pass information to their parent extension. This content script acts as the interface between the injected script and the actual extension, and relays requests and responses between the extension and the injected script.

```
// Inject the injectedScript into the current web page.
injectScript(chrome.runtime.getURL("assets/injectedScript-CVX7b_kC.js"));

chrome.runtime.onMessage.addListener((request, _sender, sendResponse) => {
```

```
if (request.method) {
  let messageType;
  if (request.method === "request") {
    messageType = "CALL_ETHEREUM_METHOD";
  } else {
    messageType = request.options.type === "call"
      ? "CALL_CONTRACT_METHOD" : "SEND_CONTRACT_METHOD";
  }

  window.postMessage(
    {
      type: messageType,
      method: request.method,
      options: request.options
    },
    "*"
  );

  // Set up a one-time listener for the response from the injected script
  const responseListener = (event) => {
    if (event.source !== window) return;

    if (event.data.type && event.data.type.endsWith("_RESULT")) {
      // Convert BigInt values (non-serializable) to strings
      const result = JSON.parse(
        JSON.stringify(
          event.data.result,
          (_, v) => (typeof v === "bigint" ? v.toString() : v)
        )
      );
      console.log("event.data.result:", result);
      sendResponse({ result: result });
      window.removeEventListener("message", responseListener);
    } else if (event.data.type && event.data.type.endsWith("_ERROR")) {
      console.log("event.data.error:", event.data.error);
      sendResponse({ error: event.data.error });
      window.removeEventListener("message", responseListener);
    }
  };

  window.addEventListener("message", responseListener);

  // Return true to indicate that the response will be sent asynchronously
  return true;
```

```
  }
});
```

The extension's react code can then define a hook to interact with the injected script to communicate with the Ethereum contract and back, through the content script's interface.

```
const useEthereum = () => {
  const callMethodInTabContext = (method: string, options?: Options) => {
    return new Promise<any>((resolve, reject) => {
      chrome.tabs.query({ active: true, currentWindow: true }, (tabs) => {
        chrome.tabs.sendMessage(tabs[0].id!, { method, options },
        (response) => {
          if (response.error) {
            reject(new Error(response.error));
          } else {
            resolve(response.result);
          }
        });
      });
    });
  };

  const ethereum = {
    request: (...args: any[]) => callMethodInTabContext(
        "request", { methodArgs: args }),
    contractCall: (method: string, methodArgs: any[], callArgs: any[]) =>
      callMethodInTabContext(method, { type: "call", methodArgs, callArgs }),
    contractSend: (method: string, methodArgs: any[], sendArgs: any[]) =>
      callMethodInTabContext(method, { type: "send", methodArgs, sendArgs })
  };

  return ethereum;
};
```

Finally, we can call methods on the Ethereum contract in the react code as follows:

```
await ethereum.contractSend(
    "storeEncryptedData",
    [messageHash, encryptedSymmetricKey, encryptedCidOfEncryptedFontFile,
        encryptedFileNameOfEncryptedFontFile], [{ from: selectedAccount }]
);
```

We can see that, while it is inherently complex to establish communication between the MetaMask extension and another extension, it is indeed possible. Other solutions were reviewed and yielded no results, such as attempting to pass the window.ethereum object from the web page to the extension, which would not work as this object contains functions that cannot be serialized when sending the object as a message between these different run times.

3.6 Evaluation

An evaluation of the system was carried out by having ten participants try out the system and interviewing them one week later. Participants were from the ages of 18-24 with current education stages being high school and university students from different majors (9 male, 1 female). Each participant had the desktop application installer downloaded on his machine as well as other needed dependencies, and the developed extension as well as MetaMask was added to his browser. A walk-through and an explanation of how the system works and how to use it was provided in order for the user to not run into any issues.

A standard System Usability Study (SUS) comprised of 10 questions was carried out, followed by a questionnaire, and finally an open-ended interview.

3.6.1 System Usability Study

For the SUS, the following questions were asked, with the responses being on a scale from 1 (strongly disagree) to 5 (strongly agree).

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The average rating for each question was calculated by summing the ratings given by all participants and then dividing by the number of participants.

3.6.2 Questionnaire

For the questionnaire, the following questions were asked, including a blend of yes/no questions, statement questions, and choice questions.

1. Do you feel that the use of technology diminishes your personal identity?
2. Do you believe that text feels more personalized and owned when it is in your handwritten font?
3. Do you think a personalized handwritten font makes your digital communications feel more authentic?
4. Do you think that using your handwritten font in digital communications helps you express your emotions more accurately?
5. Do you think that viewing messages from other individuals in their handwriting helps show their personality more?
6. Do you believe that having multiple emotional variants of your generated font is beneficial or not, and why?
7. How important is it for you to maintain the uniqueness of your handwritten font in digital communications?
8. How comfortable are you with your handwritten font being stored on a decentralized network used for digital communication?
9. What are your primary concerns about digitizing your handwriting for use as a font?
10. Do you have any concerns about the long-term storage and management of your digitized handwriting?
11. Do you have any privacy concerns about the potential misuse of your handwritten font, such as the forgery of your signature?
12. In the event of your passing, would you prefer your handwritten font to be deleted to symbolize the end of your existence in the real world? Why or why not?
13. Are there specific digital platforms (e.g., email, social media) where you would prefer to use your handwritten font?

3.6.3 Open-Ended Interview

Open-ended interviews were held with the participants, and discourse analysis was utilized to analyze the participants' responses. Participants were given the chance to freely express their opinions on matters such as the system itself, the topic of technology and its impersonal tendencies, the degree of practicality or usefulness of using handwritten fonts in messaging, and its application in solving the problem of the lack of personalization in online communication.

3.6.4 Findings and Results

The following average ratings were achieved for the SUS.

Question	**Average**
I think that I would like to use this system frequently.	3.5
I found the system unnecessarily complex.	1.5
I thought the system was easy to use.	4.8
I think that I would need the support of a technical person to be able to use this system.	1.2
I found the various functions in this system were well integrated.	5.0
I thought there was too much inconsistency in this system.	1.0
I would imagine that most people would learn to use this system very quickly.	3.5
I found the system very cumbersome to use.	1.0
I felt very confident using the system.	4.6
I needed to learn a lot of things before I could get going with this system.	2.0

Table 3.1: Average Ratings for Usability Questions

From these ratings, we can interpret a few points. For frequent use, an average rating of 3.5 indicates a moderate likelihood that participants would use the system frequently. This suggests that while the system has potential, there may be areas that need improvement to increase user engagement. The low rating of 1.5 for unnecessary complexity and the high rating of 4.8 for ease of use indicates that users generally did not find the system overly complicated, which is a positive aspect of its design. The low score of 1.2 for needing technical support suggests that the system is user-friendly and does not require additional help to operate, which enhances its accessibility for a broader audience. A score of 5.0 for well-integrated functions shows that users felt the system components worked seamlessly together, reflecting good design coherence. The low score of 1.0 for inconsistency implies that users did not perceive significant discrepancies in the system, which contributes to a smoother user experience. An average rating of 3.5 for quick learning indicates that users generally found the system straightforward to learn, though there might be room for simplifying the onboarding process. The low score of 1.0 for cumbersome use shows that users did not find the system difficult to navigate or use, reinforcing its usability. For confidence of use, a high rating of 4.6 indicates that users felt confident using the system, suggesting that it is intuitive and reliable. For the initial learning requirements, the score of 2.0 suggests that there may be some initial learning required, but it is not overly burdensome.

For the questionnaire, the following findings were acquired:

1. 60% of the participants felt that technology diminishes their personal identity.
2. 90% of the participants believed that text feels more personalized and owned when it is in their handwritten font.
3. 100% of the participants thought that a personalized handwritten font makes their digital communications feel more authentic.
4. 70% of the participants thought that using their handwritten font in digital communications helps them express their emotions more accurately.
5. 90% of participants thought that viewing messages from other individuals in their handwriting helps show their personality more
6. Most participants agreed that having multiple emotional variants of their generated font is beneficial, with reasonings such as helping to deliver the intended emotion of the message, not just a plain message.
7. Participants described the importance of the uniqueness of their handwritten font as either moderately important or very important.
8. Most participants were comfortable with their handwritten font being stored on a decentralized network as long as it was secure enough.
9. Participants' main concerns about digitizing their handwriting as a font were related to privacy, data security, the potential misuse or unauthorized access to their personal handwriting, and a signature being copied to steal their identity or commit forgery.
10. The same security concerns were brought up with regard to concerns about the long-term storage and management of digitized handwritten fonts
11. Most participants were concerned about the potential misuse of their handwritten font, such as forgery of their signature.
12. In the event of their passing, half of the participants preferred their handwritten font to be deleted from existence, citing that it represents them as a person and thus it would have to go once they do. Whereas the other half preferred not to remove it, for reasons such as people remembering the individual and praying for them.

13. Participants wished to use this feature on social media platforms to make posts, as well as emails.

As for the open-ended interviews, using discourse analysis helped evaluate the responses from the open-ended interviews, revealing several key themes that emerged. Participants primarily expressed perceptions that centered around two main areas: the system's practicality and usage, and the usefulness and authenticity of the system and its role in solving the personalization problem.

Many participants found the system intuitive and easy to use once explained to them and after trying out a full cycle through the system, aligning with the high usability scores. However, none of the participants were accustomed to Ethereum, blockchain technology, or any related fields, and understandably struggled to adapt and understand the role of such technologies in the system, and the need for an Ethereum wallet such as MetaMask, and cryptocurrency such as Ether to pay for any transactions. Given these technologies are somewhat new, this is understandable and expected.

Participants emphasized the importance of personalized handwriting in digital communication. In particular, some of them highlighted that using handwritten fonts for sending messages was a side of personalization that they did not think of nor expect, but welcomed nonetheless as another dimension of personality in messaging. The participants on the authenticity of the outputted fonts, mostly mentioned that the normal handwritten font was almost identical to the letters they filled in the template, but they did run into challenges in scanning the template image and inserting it properly into the system without any skewing or rotations in order to get accurate results. They also mentioned that the emotional variants of the fonts had the right idea, but the outputted quality was not adequate and did not provide the intended balance between their handwriting and the intended additional emotional touch. They mentioned white-space issues, in which some letters were too highly elevated compared to others. This problem is related to the interpolation technique used to generate the emotional variants, and we discuss it further in the limitations section.

3.6.5 Overall Interpretation of Findings

The findings from the SUS, the questionnaire, and the open-ended interviews provide a comprehensive view of user perceptions of the system and the idea behind it.

The SUS results indicate that the system is generally user-friendly, with high scores for ease of use and integration of functions. The discourse analysis supports this, as participants mentioned the system's ease of use after the initial learning.

Privacy and security concerns were prominent. Participants were wary of the potential misuse of their digitized handwriting, particularly for forgery or unauthorized access, highlighting the utmost importance of security in such a system, and justifying our design choices favoring security and leveraging decentralized technologies. The split opinions on the management of handwritten fonts after one's passing highlight differing views on digital permanence and legacy. This area requires sensitive handling and clear policies to cater to user preferences, and could affect our choice of storage, as it is difficult to remove files once uploaded to the IPFS due to its very nature, as uploaded content gets replicated over the network and to remove content from it would require removing it from every node that is hosting that content on the network.

Finally, and most importantly, there is a strong user preference for personalization in digital communication, with participants overwhelmingly supporting the use of their handwritten fonts to convey authenticity and emotional depth. This aligns with the high agreement in the questionnaire responses regarding the benefits of personalized handwriting, as well as the expressed agreement in the open-ended interviews. The participants did, in fact, appreciate the idea of using handwritten fonts as well as emotional variants of those fonts to send messages over the web, agreeing that this adds an unexpected but welcomed dimension of personalization to messages online. This aligns with our primary goal for this thesis, to provide a solution to the lack of personalization in online communication.

3.7 Comparison with Related Works

The previously discussed related works covered using fonts, and some discussed handwritten fonts, in particular, to enhance the emotional connection between individuals and certain products. They also discussed congruity of the font with the entity in which it is used with to increase appropriateness and connotations, and better deliver intended messages and emotions. In addition, they covered personality traits and emotions and the ability of individuals to make links between them and certain fonts.

Building upon this information, the developed system creates a handwritten font for the user, with the aim of bringing out their personality when communicating on the web. In addition, emotional variants are generated from this handwritten font to better align with the mood of the message and the user when sending that specific message. This is all intended to align with the user's personality and emotions and better expressing them to all recipients of the messages.

3.8 Summary

In this chapter, we have covered the design goals, design strategy, an overview of the major components of the system, the system design, the implementation details, an evaluation of the system, and a comparison between the system and its related works. We have successfully developed a system with a desktop application for generating handwritten fonts and emotional variants from them, as well as infrastructure using the Ethereum blockchain and IPFS for setting up the messaging system, and a Chrome extension to interact with said infrastructure to allow for sending and displaying handwritten messages. An overview of the system in its final state can be seen in Figure 3.16.

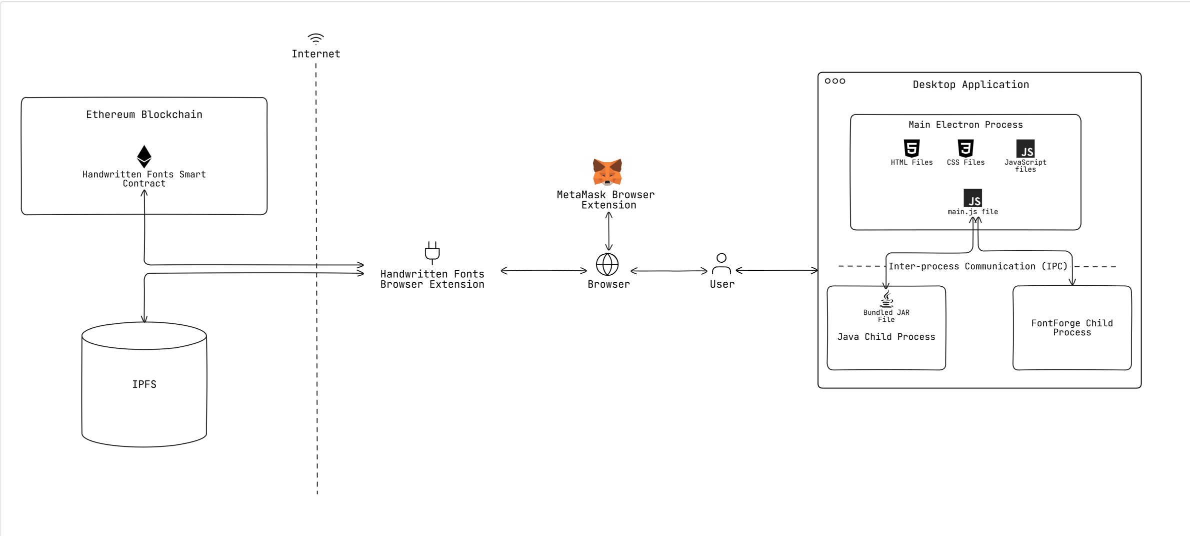

Figure 3.16: System Overview

Chapter 4

Conclusion

4.1 Summary

This thesis investigated the potential for integrating user-generated handwriting into web-based messaging for a more personalized communication experience. A software system has been developed that allows users to create handwritten font files based on their handwriting, as well as create different emotional variants of their fonts. These fonts can then be used for sending messages on web platforms.

The research addressed the challenge of adding a personal touch to digital communication, which often lacks the individuality and emotional depth conveyed by handwritten notes. The proposed solution involves a software system that enables users to capture their own handwriting by filling out a template, generate handwritten font files based on their handwriting, and integrate the custom font into web platforms for messaging.

An evaluation of the system was carried out by having ten participants try out the system and interviewing them one week later. A standard SUS was used, followed by a questionnaire, and finally an open-ended interview. The findings indicated participants' appreciation of the idea of using handwritten fonts as well as emotional variants of those fonts to send messages over the web, agreeing that this adds an unexpected but welcomed dimension of personalization to messages online.

The developed system succeeded in demonstrating the feasibility of this approach. Users can successfully create fonts based on their handwriting and generate emotional variants from them and potentially enhance the emotional connection in digital communication. This project has taken a significant step towards bridging the gap between the impersonal nature of digital communication and the warmth of handwritten notes. The developed system demonstrates the potential for personalization and emotional expression through the use of handwritten fonts in web-based communication.

4.2 Limitations

There are a few limitations to the current system. The current automatic generation of the font file from an input filled template has limitations due to the need for the input image to be extremely precise; it needs to almost exactly match the width and height of the original downloaded template image so that the program's preset coordinates for each letter are accurate. In addition, the use of interpolation for generating the emotional font variants is too simplistic and linear and does not produce fonts with adequate quality.

4.3 Future Work

Future work on this project should develop and train a custom Optical Character Recognition (OCR) model for detecting the individual characters from the filled template image so that the users can upload filled template images with varying and less strict widths, heights, skewing, rotation, and qualities, and without needing to scan it and worry about accuracy.

In addition, machine learning methods should be used for blending two fonts to generate emotional font variants, as this problem is likely best suited for this domain. It requires a non-linear approach and perhaps a stylistic input to the process to be able to merge two fonts and create an intermediate one, all while resembling the original handwriting enough and also adding an emotional touch to it.

The user experience could also be improved in the messaging system. The user might be allowed to enter the message they want to send directly on the web page of their desired web application, and the text input could be detected and recognized by the Handwritten Fonts extension and a toggle gets added next to it which the user can click if they want to send that message in a handwritten font. This toggle would then open the extension and prompt the user to select the font and recipient and then apply the generated message to the same input element.

The system could also be extended to support more characters, instead of just the twenty-six lowercase English letters. It could support capital letters, other symbols, as well as other language letters such as Arabic.

4.4 Summary

This chapter has covered a summary of the thesis including the research problem, the proposed solution, the project that has been implemented, and the results. It has also covered the limitations of this thesis and future work.

Appendix

Appendix A

Additional Materials

A.1 Quotes from Evaluation

The following is a collection of quotes from the participants of the evaluation carried out for the project, collected during the open-ended interview section, talking about their perceived usefulness of using handwritten fonts and their emotional variants for messaging.

> ”It helps me express a wider range of emotions”

> ”I think it enhances expression and personalization in my messages.”

> ”This gives me more ways to express my thoughts and how I feel.”

> ”It delivers different emotions.”

> ”They reveal how I actually feel regarding what I'm writing.

> ”It helps me explain my thoughts more.”

> ”It shows more emotion and personality.”

> ”My handwriting represents me as a person.

> ”It is part of my identity”

A.2 Screenshots

The following are a few sample screenshots taken from the participants of the evaluation trying out the system using their handwritten font on WhatsApp web and on Outlook email.

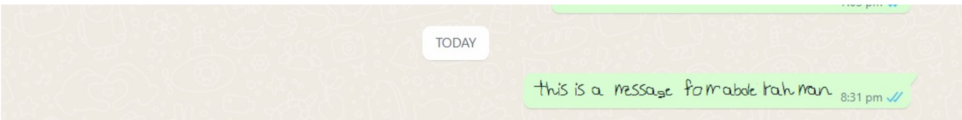

Figure A.1: Sample handwritten message on WhatsApp web

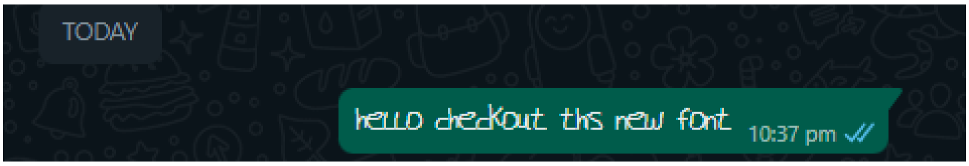

Figure A.2: Sample handwritten message on WhatsApp web

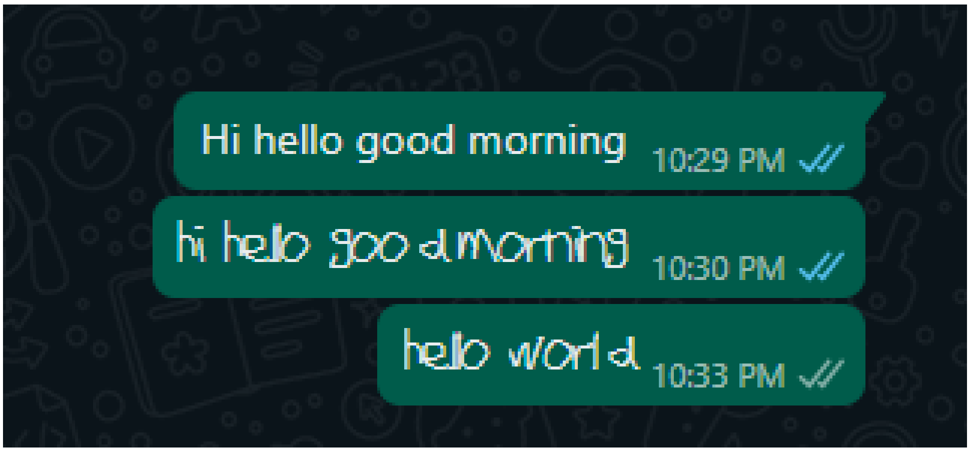

Figure A.3: Sample handwritten message on WhatsApp web

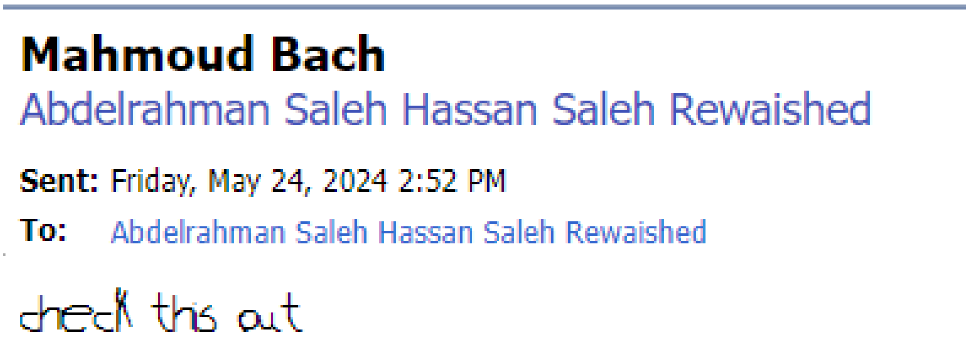

Figure A.4: Sample handwritten message on Outlook

Appendix B

Lists

HTTPS	Hypertext Transfer Protocol Secure
IPFS	InterPlanetary File System
UUID	Universally Unique Identifier
dApps	Decentralized Applications
DOM	Document Object Model
EVM	Ethereum Virtual Machine
CIDs	Content Identifiers
OCR	Optical Character Recognition
SUS	System Usability Study

List of Figures

List of Tables

Bibliography

[1] Smart contract languages, 2024. [Online; accessed 9-May-2024].

[2] Renata Germano Bianchi, Kamila Rios da Hora Rodrigues, and Vânia Paula de Almeida Neris. Emotional responses to font types and sizes in web pages. In *Proceedings of the XX Brazilian Symposium on Human Factors in Computing Systems*, number Article 17 in IHC '21, pages 1–11, New York, NY, USA, October 2021. Association for Computing Machinery.

[3] Joel Bock. Technology, freedom, and the mechanization of labor in the philosophies of hegel and adorno. *Philosophy Technology*, 34:1263–1285, 2021.

[4] Eva R Brumberger. The rhetoric of typography: The persona of typeface and text. *Technical Communication*, 50:206–223, May 2003.

[5] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform, 2014. [Online; accessed 9-May-2024].

[6] Andrew Cross, Mydhili Bayyapunedi, Edward Cutrell, Anant Agarwal, and William Thies. TypeRighting: combining the benefits of handwriting and typeface in online educational videos. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 793–796, New York, NY, USA, April 2013. Association for Computing Machinery.

[7] John R Doyle and Paul A Bottomley. Dressed for the occasion: Font-Product congruity in the perception of logotype. *J. Consum. Psychol.*, 16(2):112–123, January 2006.

[8] Alysha Fligner. The effect of packaging typeface on product perception and evaluation. December 2013.

[9] Doug Fox, A Dawn Shaikh, and Barbara S Chaparro. The effect of typeface on the perception of onscreen documents. *Proc. Hum. Fact. Ergon. Soc. Annu. Meet.*, 51(5):464–468, October 2007.

[10] Yongrui Guo, Xiaotian Cui, and Yuzong Zhao. Handwritten typeface effect of souvenirs: The influence of human presence, perceived authenticity, product types, and consumption goals. *J. Int. Hosp. Leisure Tour. Manag.*, 58:345–353, March 2024.

[11] R Hazlett, K Larson, A D Shaikh, and Barbara S Chaparo. Two studies on how a typeface congruent with content can enhance onscreen communication. *Information Design Journal*, 20:207–219, 2013.

[12] IPFS. An open system to manage data without a central server, 2024. [Online; accessed 9-May-2024].

[13] Samuel Juni and Julie S Gross. Emotional and persuasive perception of fonts. *Percept. Mot. Skills*, 106(1):35–42, February 2008.

[14] Soojin Kim, A-Reum Jung, and Yoojung Kim. The effects of typefaces on ad effectiveness considering psychological perception and perceived communicator's power. *Journal of Marketing Communications*, 27(7):716–741, October 2021.

[15] Tugba Kulahcioglu and Gerard de Melo. Fonts like this but happier: A new way to discover fonts. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, pages 2973–2981, New York, NY, USA, October 2020. Association for Computing Machinery.

[16] Ying Li and Ching Y Suen. Typeface personality traits and their design characteristics. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 231–238, New York, NY, USA, June 2010. Association for Computing Machinery.

[17] J Mackiewicz and R Moeller. Why people perceive typefaces to have different personalities. In *International Professional Communication Conference, 2004. IPCC 2004. Proceedings.*, pages 304–313. IEEE, 2004.

[18] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. [Online; accessed 9-May-2024].

[19] Roland Schroll, Benedikt Schnurr, and Dhruv Grewal. Humanizing products with handwritten typefaces. *J. Consum. Res.*, 45(3):648–672, March 2018.

[20] A D Shaikh, B Chaparro, and D Fox. Perception of fonts: Perceived personality traits and uses. 2006.

[21] StatCounter. Browser market share worldwide, 2024. Accessed: 2024-05-08.

[22] Statista. Global customer experience personalization and optimization software revenue 2024, 2024. Accessed: 2024-05-07.

[23] Statista. Worldwide digital population 2024, 2024. Accessed: 2024-05-07.

[24] Scott R. Stroud. Living large: Kant and the sublimity of technology. *Teaching Ethics*, 2003.

[25] T T L Wang. Fonts and fluency: The effects of typeface familiarity, appropriateness, and personality on reader judgments. 2013.