

**Security Fundamentals and Development (H7SFD)**

**BSHC3**

**CA1 – Group Project Report**

**Group:** E

**Members:**

Eoin Fitzsimons x23151374

Eskandar Atrakchi x23137517

David O’Connor x23153784

Lecturer: Kamil Mahajan



**National College of Ireland**

**Project Submission Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** | Eoin Fitzsimons | | |
| **Student ID:** | X23151374 | | |
| **Programme:** | Computing | **Year:** | 3 |
| **Module:** | Security Fundamentals | | |
| **Lecturer:** | Kamil Mahajan | | |
| **Submission Due Date:** | 15-11-24 | | |
| **Project Title:** |  | | |
| **Word Count:** |  | | |

**I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.**

**ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.**

|  |  |
| --- | --- |
| **Signature:** | ……………………………………………………………………………………………………………… |
| **Date:** |  |

**PLEASE READ THE FOLLOWING INSTRUCTIONS:**

1. Please attach a completed copy of this sheet to each project (including multiple copies).

2. Projects should be submitted to your Programme Coordinator.

3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.

4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**

5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

|  |  |
| --- | --- |
| **Office Use Only** | |
| Signature: |  |
| Date: |  |
| Penalty Applied (if applicable): |  |

# AI Acknowledgement Supplement

1. **Cybersecurity Fundamentals**

# CA1 – Group Project Report

|  |  |  |
| --- | --- | --- |
| **Your Name/Student Number** | **Course** | **Date** |
|  |  |  |

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click [here](https://libguides.ncirl.ie/useofaiinteachingandlearning/studentguide).

# AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

|  |  |  |
| --- | --- | --- |
| **Tool Name** | **Brief Description** | **Link to tool** |
|  |  |  |
|  |  |  |

# Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

|  |  |
| --- | --- |
| **[Insert Tool Name]** | |
| [Insert Description of use] | |
| [Insert Sample prompt] | [Insert Sample response] |

# Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

# Additional Evidence:

[Place evidence here]

# Additional Evidence:

[Place evidence here]

**Overall page limit – 5 pages per member in a group (excluding front pages and references/annexure)**

# Co**ntributions of each member of the Group**

Describe the contribution or tasks of each member of the group (e.g., investigation, testing, coding of the application like GUI or cryptographic mechanisms, etc.).

|  |  |  |
| --- | --- | --- |
| **StudentID** | **StudentName** | **Tasks** |
| x23151374 | Eoin Fitzsimons | Nothing |
| x23137517 | Eskandar Atrakchi | Everything |
| x23153784 | David O’Connor | Something |

# Link to the video demo

This is a link to a 5-minute video demonstrating the application and a quick walkthrough of the code. Make sure the video has the appropriate permissions to be accessed.

# Link to the GitHub project **(Make sure the accessibility is public)**

This is a link to your GitHub project which contains the source code of your application.

[EskandarAtrakchi/multi-format-encryption-decryption](https://github.com/EskandarAtrakchi/multi-format-encryption-decryption)

# Manual

Describe the instructions to download, install and run your application.

--------

The Project will be deployed on GitHub using GitHub Pages Jekyll by GitHub Actions package a Jekyll site with GitHub Pages dependencies preinstalled meaning no installing required, only a website link anyone, anywhere can access the project by clicking on the link.

# Summary of the Application

**Chosen Use Case Scenario:** [20. Secure storage using caching]

# Business Functionality

The purpose is to provide a secure web-based platform that allows users to store their data locally on their devices, minimizing the risk of unauthorized access or data theft. The application ensures that sensitive information is encrypted on the user's machine using encryption protocols. Additionally, users must create a PIN to add an extra layer of security, giving them full control over their data privacy.

## Value to Users:

This application offers the value of strapping local data encryption, authorizing users to protect their sensitive information even on shared computers. The core features include:

[AES-GCM Encryption](https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto/encrypt) for strong data protection.

[Data Hashing with SHA-256](https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto/digest) for secure PIN management.

[Crypto API](https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto) (window.crypto.subtle) in JavaScript, leveraging the latest in-browser cryptography standards.

## Use Cases:

Privacy on shared devices: When working on a personal document, such as a love letter, they can save it securely on a family home computer. The app prevents unauthorized access from family members, ensuring classified information stays private.

Temporary Secure Storage for Sensitive Notes: If a user needs to type delicate details (like a password or financial details) for future reference, they can store this information temporarily without the worry of unauthorised access by others who may use the same device.

## Provenance of Source Code:

The encryption is built upon the AES-GCM algorithm and the window.crypto.subtle API in JavaScript for secure, in-browser encryption and decryption. These are industry-standard functions that ensure efficient performance and impregnable security. Our custom modifications were made to integrate user-specific PIN creation and management, these modifications complemented AES encryption by adding a user-defined layer of protection.

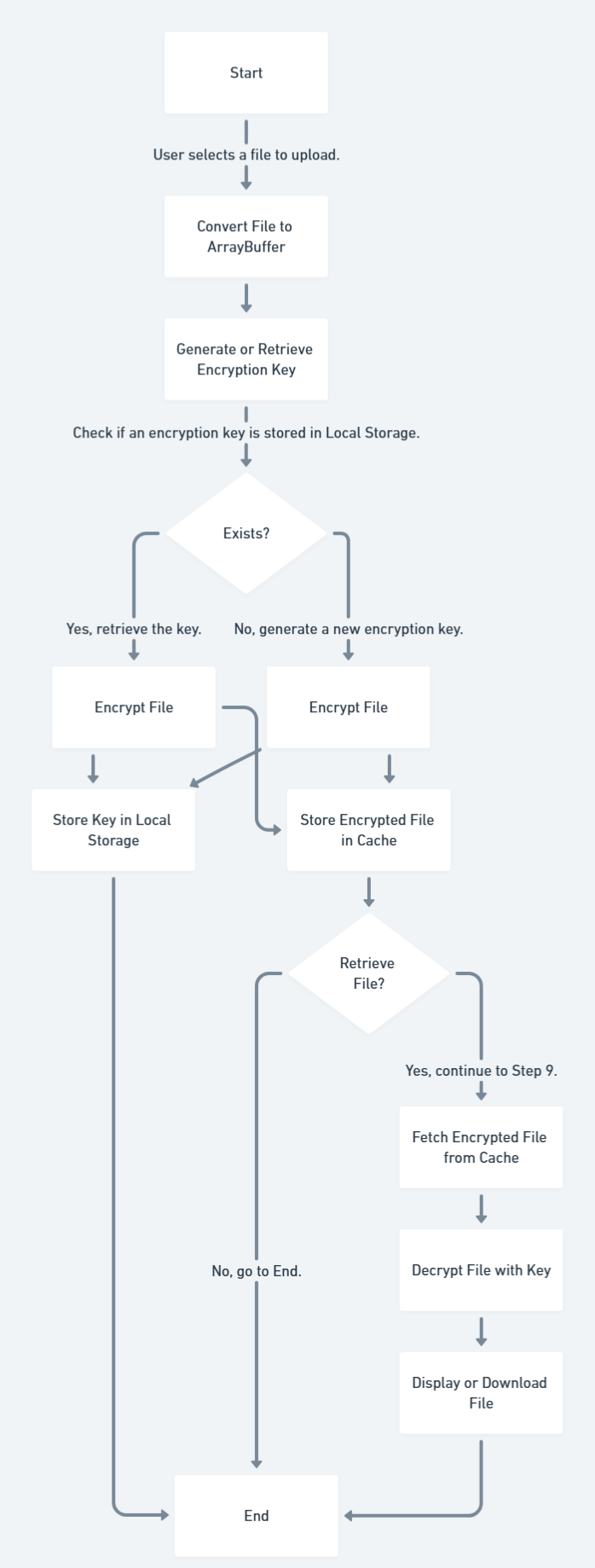
# Design of the Application

The diagram we drew after the team discussion is a chart flow that describes the flow of the project.

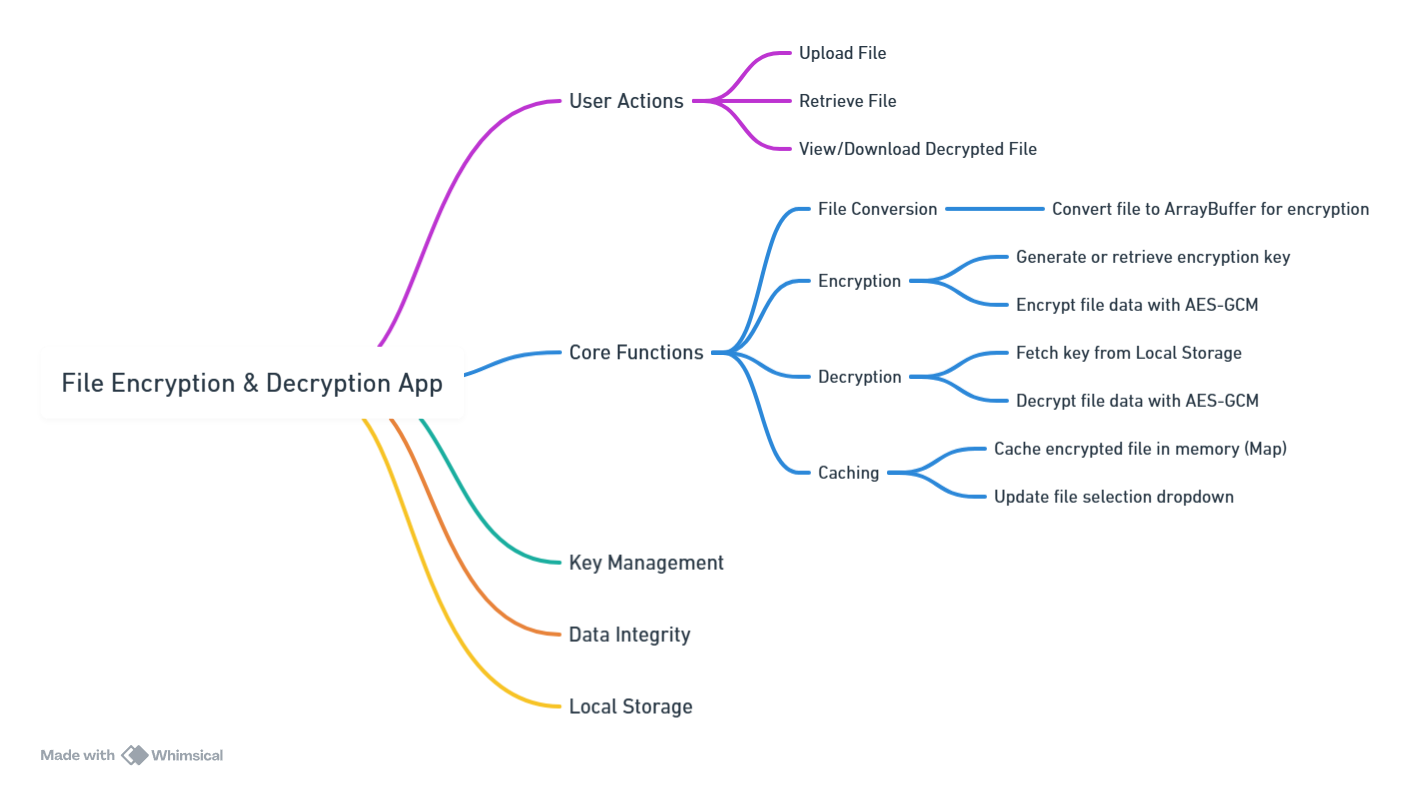
This is how we structured the diagrams for

1. Flowchart
2. Mind map
3. Sequence

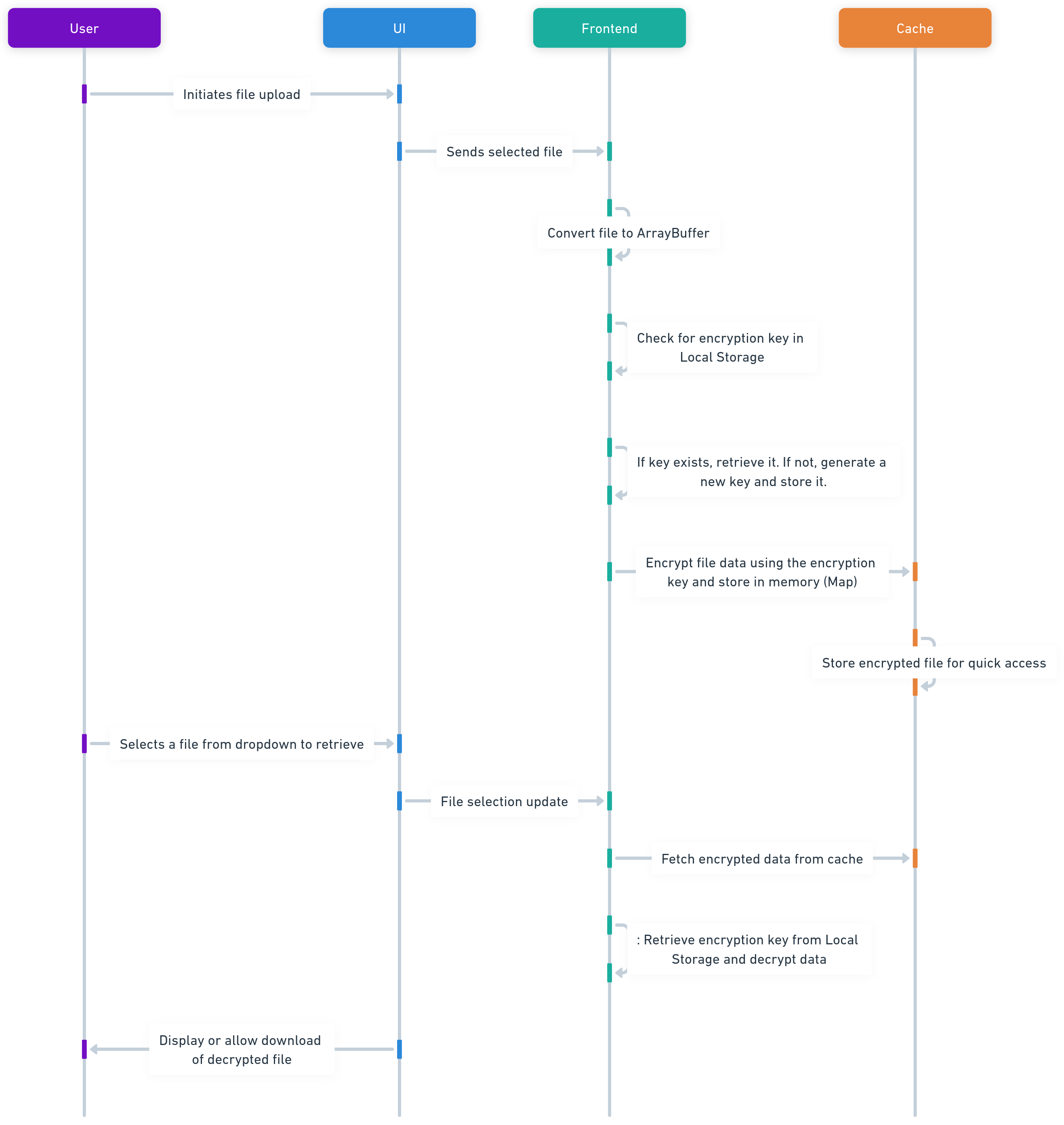
We used a flowchart to understand the process of our project and structure the files to begin building (coding) on GitHub.

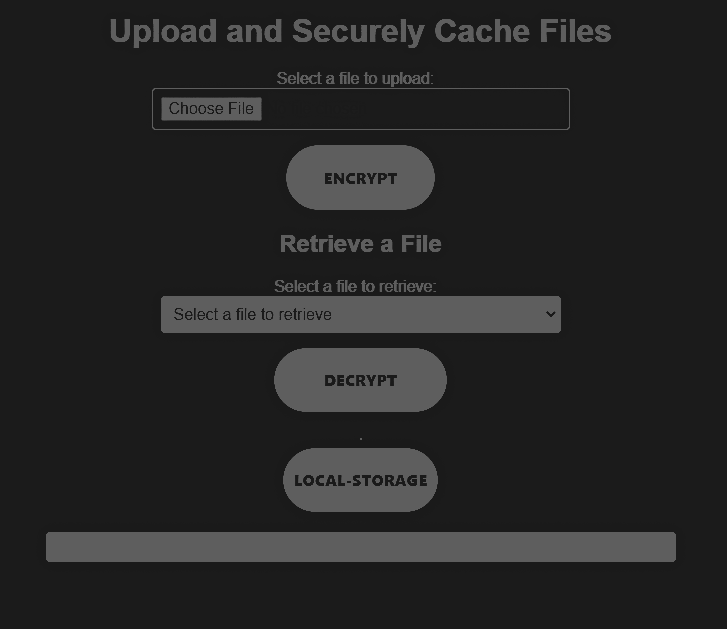


We created a mind map because it focuses on the big components and functionalities of the project, going up from the central concept, this helped us understand what to build (code) first and settle down the priorities.



Now that we understand how to structure our project and have settled on the priorities for coding, the next big step is to understand the sequence of our project. Without knowing what happens behind the scenes when clicking buttons, the project could become unsustainable or fail to achieve its intended purpose.



For the design before building the project, our goal was to create a simple project with four buttons and one dropdown list of stored files. Our focus was on designing diagrams to help us understand how to code the project while achieving the Confidentiality, Integrity, and Availability (CIA) concepts.

# **Technical Implementation**

## **Technologies and Languages**

After discussing whether to use Vite.js, TypeScript, or JavaScript with HTML, we decided on the following:

**JavaScript**: for client-side logic.

**HTML & DOM Manipulation**: for UI components in the project, such as uploading files, creating a dropdown menu, and displaying decrypted files.

**Local Storage**: to store the encryption key and encrypted file data persistently.

**Web Crypto API**: to provide secure cryptographic functions for encryption, decryption, and hashing.

### File Encryption and Decryption Workflow:

**AES-GCM (Advanced Encryption Standard - Galois/Counter Mode)**: We chose AES-GCM after researching its efficiency and authentication properties. AES-GCM helps us achieve the *C* in the CIA triad—confidentiality—by encrypting file data with a secure key.

**IV (Initialization Vector)**: After our research, we learned that many data hacks occur due to patterns in decrypted data. Using an IV introduces randomness, preventing patterns in the encrypted data.

## How did we implement it?

We retrieve or generate a 256-bit AES-GCM encryption key, which is stored as JSON in local storage.

The file is converted to an ArrayBuffer, encrypted with the key, and the result (encrypted data and IV) is stored in both Local Storage and memory cache.

### Key Management:

**Local Storage**: Used to store the encryption key, making it available for retrieval at any time on the same device.

**Key Persistence**: On the first run, the application generates the key for the user and stores it in local storage. For future uses, the key is retrieved rather than regenerated, ensuring project consistency and sustainability.

### Data Integrity Mechanism:

We use **SHA-256 Hashing** to create a hash of the file data before encryption. This hash is later used to verify the file’s integrity during decryption by comparing it with the hash of the decrypted data, ensuring data integrity.

## Why did we choose this approach?

The **SHA-256 hash** guarantees data integrity by detecting any accidental or intentional alterations in the file contents.

### File Storage and Retrieval:

**File Storage in Cache**: We opted for local storage to allow for quick and easy file storage and retrieval.

**Decryption Process**: During decryption, the application retrieves the encrypted data, uses the stored key to decrypt it, and verifies the file’s integrity using the SHA-256 hash. The file is displayed only if the integrity check is successful.

**Format Usage after File Conversion**: After decryption, the file is displayed using blobs, URLs, and DOM elements like images, iframes, or download links.

### Cryptographic Mechanisms – Rationale:

**AES-GCM**: Chosen for secure, authenticated encryption. Encryption and authentication protect data from unauthorized access and tampering. AES-GCM is symmetric, meaning it uses one key for both encryption and decryption, which makes it efficient, and faster, and allows for more storage. It is also trustless, as we use local storage for encryption.

**SHA-256**: Ensures data integrity by preventing accidental or malicious corruption of the data.

**IV (Initialization Vector)**: Introduces randomness to prevent repeated patterns in encrypted data, ensuring unique ciphertexts even when encrypting the same data.

# Conclusion:

To achieve the CIA, we followed this: (CIA and STRIDE) \*\* (Davey) I am coming back to this section to work on it.

AES-GCM: to achieve confidentiality.

SHA-256: to achieve integrity.

And if both are achieved successfully the local storage is where the file is available.