

**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 Fundementals**
2. **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).

1. **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** |
|  |  |  |
|  |  |  |

1. **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] |

1. **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.

1. **Additional Evidence:**

[Place evidence here]

1. **Additional Evidence:**

[Place evidence here]

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

1. **Contributions 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** |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Link to the demo video**

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.

1. **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)

1. **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.

1. **Summary of the Application**

**Chosen Use Case Scenario:** [ID and name]

**6.1 Business Functionality**

Briefly describe the purpose of the application and what is the value offered by it in terms of functionality and features present. Illustrate with use cases as appropriate.

If you are using an existing source code as the basis for your project, clearly indicate it here describing its provenance (where the code comes from) and what were the modifications/novel parts or contributions to this specific project.

--------

The purpose of the project is that any user can use the localStorage of any current used device to store any format of data securely by decrypting the data, the data then can be retrieved securely by the user only and only if the conditions meet

1. The data is not altered at all.
2. The user is authorized (The user is the one meant to retrieve).

Where did we get the code from?  
We as a team got the code from [developer.mozilla](https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto) by reading the documentation and following of how to implement the code to get the following

* [AES-GCM Encryption](https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto/encrypt)
* [Crypto API](https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto): The window.crypto.subtle API in JavaScript
* [Data Hashing with SHA-256](https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto/digest)
* [File Handling in JavaScript](https://developer.mozilla.org/en-US/docs/Web/API/FileReader)

By following these instructions, we have built the project using plain JS with HTML.

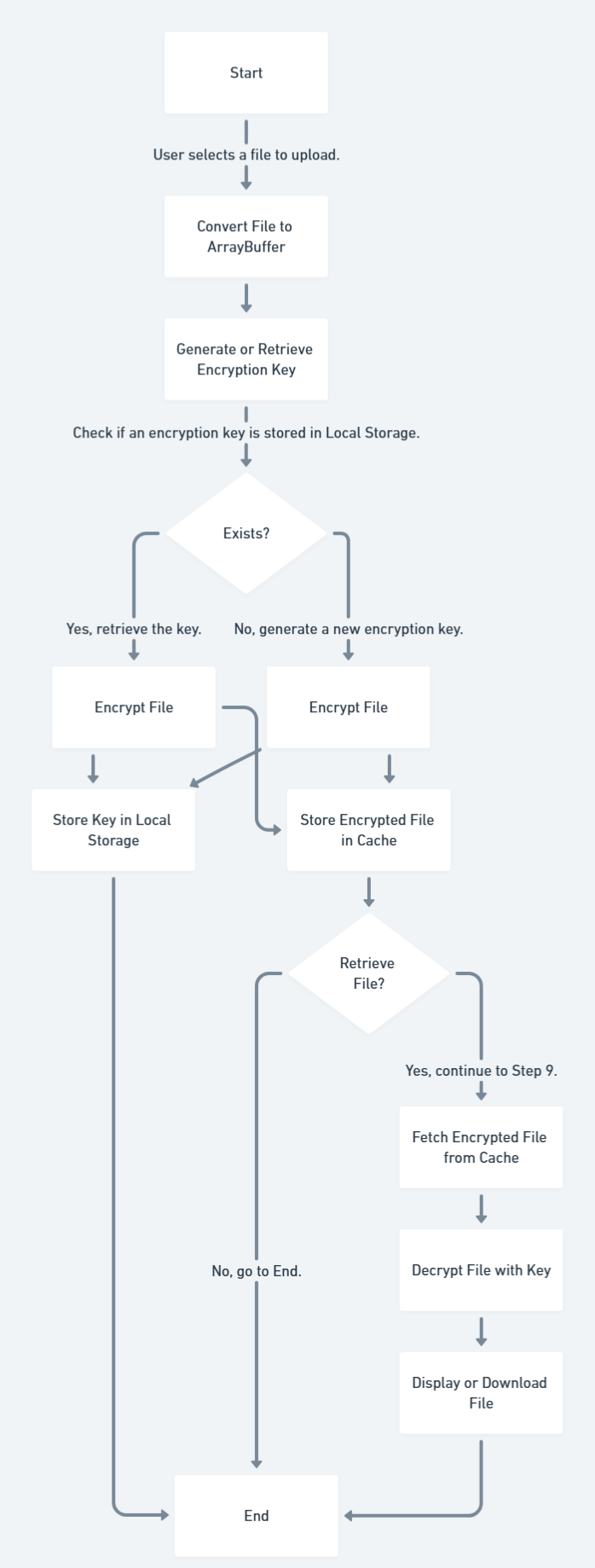
* 1. **Design of the Application**

Create a diagram that illustrates the architecture of your application (including inputs, outputs, and main components like GUI, databases, etc.). Also include a diagram that illustrates the application's operating process (e.g., using flowcharts, activity diagrams, sequence diagrams).

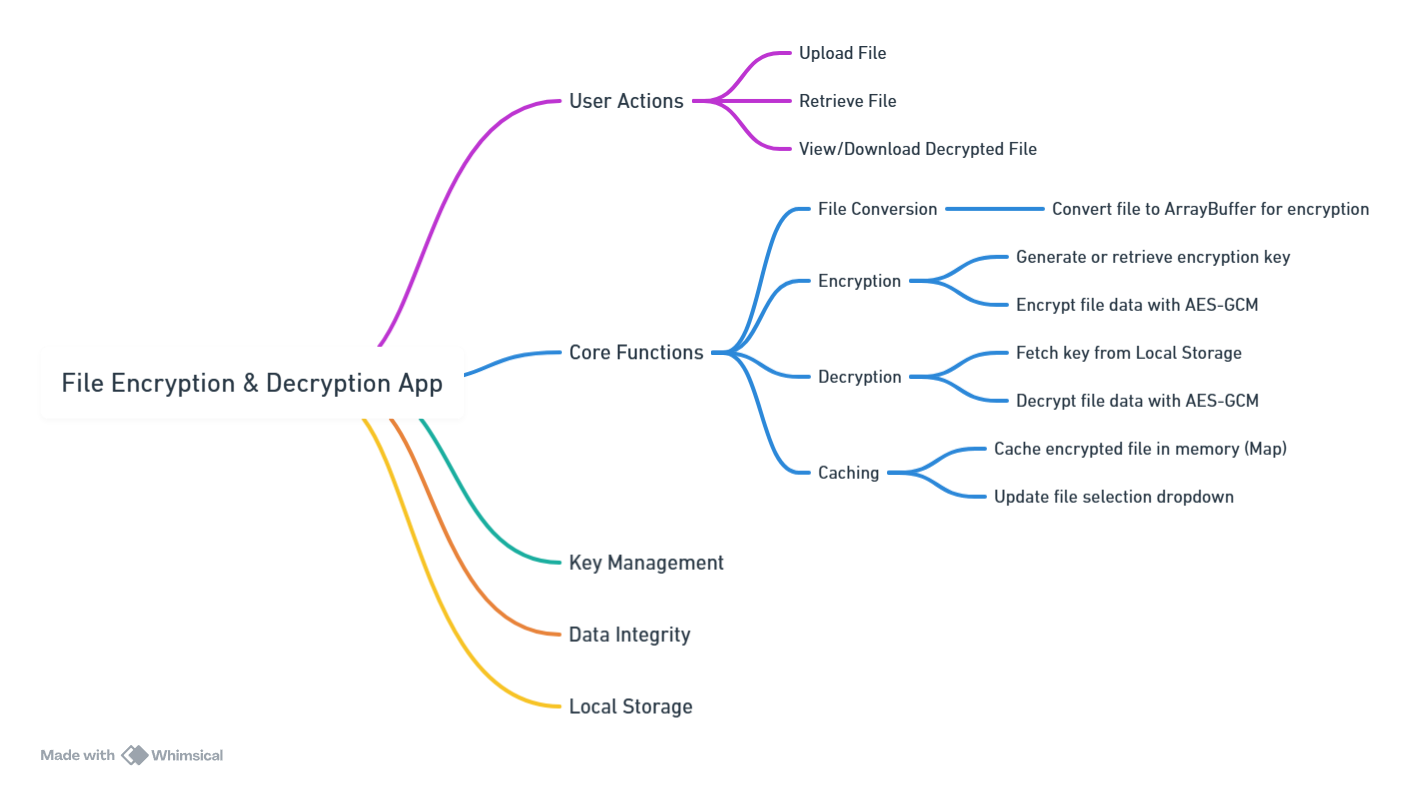
----------

The diagram we draw as after team discussion is chart flow that describes the flow of the project.

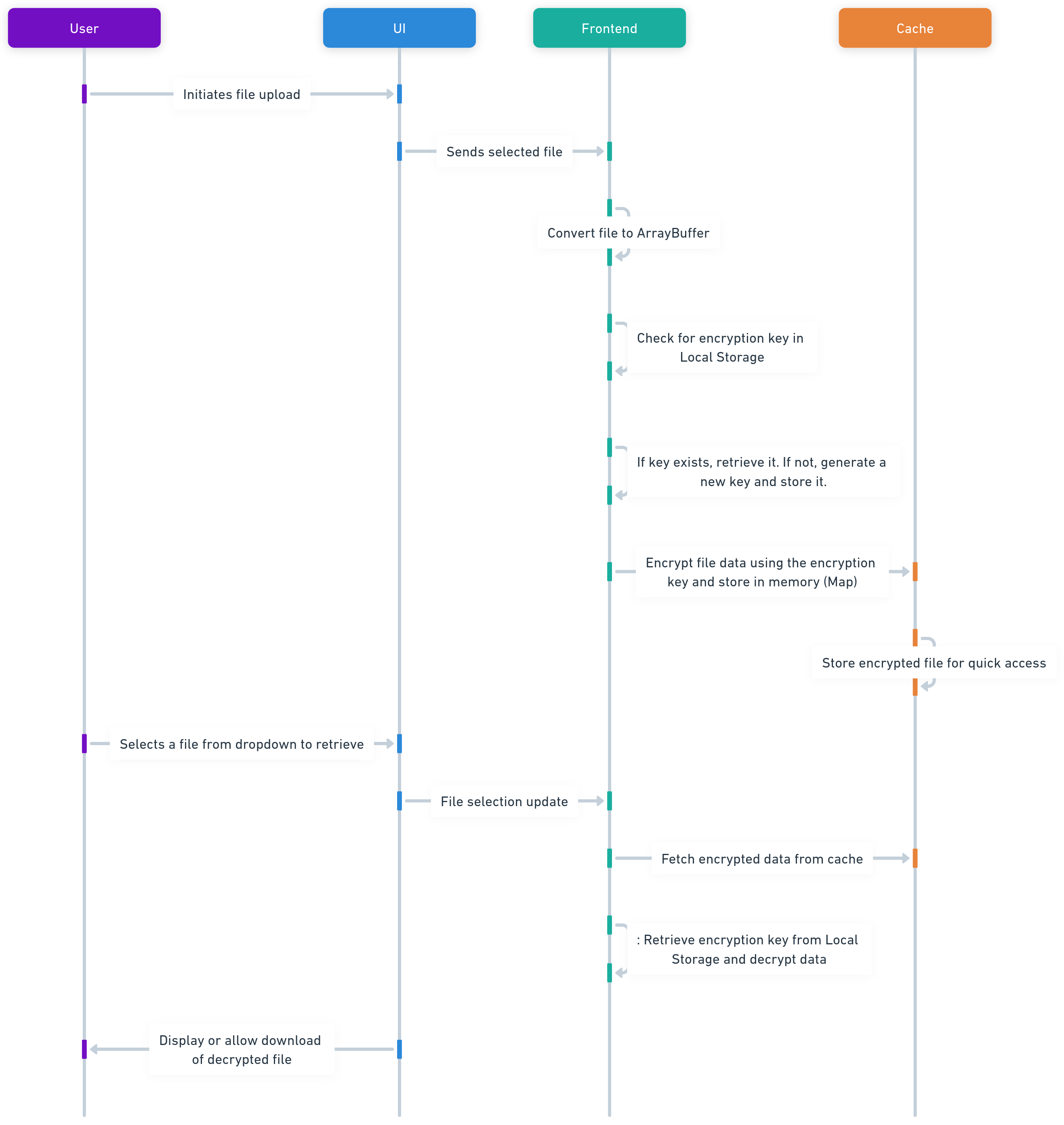
Flowchart helped us to understand the process of our project which helped us in structuring the files to begin building (coding) on GitHub.



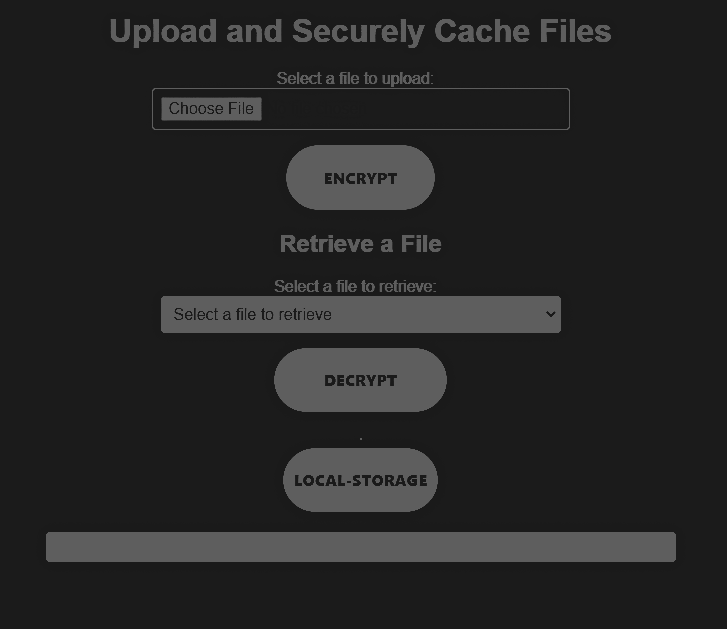
Second chart we have created is mind map which we have voted on using it because it focuses on the big components and functionalities of the project, going up from central concept and this helped us to understand what to build (code) first and settled down the priorities



Now, after we understand how to structure our project, and we settled the priorities to code the next big step is to understand the sequence of our project because without knowing what is happening behind clicking buttons this would lead to unsustainable project and might fail, or the project does not do its intended purpose.



For the design before building the project: our purpose was to establish simple project with 4 buttons and one dropdown list of the stored files as our focus was on how to create diagrams for the project to know how to code it and achieve the Confidentiality, Integrity, and Availability (CIA) concept.



* 1. **Technical Implementation**

Describe how you implemented technically the application. Discuss any technologies involved, programming languages, libraries, and other algorithms with emphasis on the cryptography mechanisms used (elaborate on how the mechanism is used in the use case and provide a rationale for choosing those specific mechanisms).

----------

1. Technologies and Languages: after a discussion if we would use viteJs TypeScript or JavaScript with HTML we settled on

* JavaScript: for client-side logic
* HTML && DOM manipulation: was used for UI components for many things in the project for example uploading files, dropdown menu, and display decrypted files.
* Local Storage: we have used this technology to store the encryption key and encrypted file data in uninterrupted way.

1. Web Crypto API: provides secure cryptographic functions for encryption, decryption, and hashing.
2. File Encryption and Decryption Workflow:

* AES-GCM (Advanced Encryption Standard - Galois/Counter Mode): we have chosed this after research as AES-GCM is efficient and authenticated for encryption properties, the AES-GCM helps us achieve the C in CIA which is confidentiality by encrypting file data with a secure key.
* IV (Initialization Vector): after research we found out that must hack of the data happens because of patterns in decrypted data, so using IV for randomness and preventing patterns in encrypted data.

How did we implement it?

* Is by retrieving or generating a 256-bit AES-GCM encryption key which is stored as JSON in local storage
* Convert the file to an ArrayBuffer, encrypt it with the key, and store the result (encrypted data and IV) in Local Storage and memory cache.

1. Key Management:

* Local Storage: is used to store the encryption key, so it can be retrieved at any moment on the same current used machine.
* Key Persistence: for the first time the application will generate the key for the user and save it in local storage, for the future uses the key will not be generated again, rather, it will retrieve achieving project consistency and sustainability.

1. Data Integrity Mechanism: we have used SHA-256 Hashing to create hash for the file data before encryption happens then the same hash is used later to verify the file’s integrity upon decryption by comparing the hash of the decrypted data (achieving integrity).

What's the team logic using this?

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

1. File Storage and Retrieval

* File Storage in Cache: we decided to use local storage allowing us for fast ease of store and retrieval.
* Decryption Process: on decryption, the application gets the encrypted data and then using the stored key to decrypt it, verifies integrity using SHA-256 hash and display the file only and only if the integrity is confirmed.

1. format usage after file conversion: to display the file after decrypt we have used blob, URLs, and DOM elements like img, iframes or even download links.

Cryptographic Mechanisms - Rationale

* AES-GCM: Selected for secure and authenticated encryption. The combination of encryption and authentication helps protect data from unauthorized access and tampering. Good to mention that this kind of encryption is symmetric meaning uses one key to encrypt and decrypt, because it is efficient, faster, allow more storage, and trust-less as we are using localStorage();
* SHA-256: Ensures data integrity, preventing accidental or malicious data corruption intentionally or unintentionally.
* IV (Initialization Vector): Introduces randomness, preventing repeated data patterns and making ciphertexts unique for each encryption, even with the same data and that the reason we decided to use it.

Conclusion: to achieve the CIA we followed this:

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.