Initial Report of

Group AttackFlow4

Building a dataset of real-world cyber-attacks with Attack Flow

Team members:

Nathaniel Chang | a1821595
Emily Miller | a1799248
An Nguyen | a1840557
Elana Parnis | a1831872
Ethan Pratt | a1828540
Angelos Sohan | a1825116
Wei Long Wan | a1801549
Jinyang Li | a1832744
Xizi Wang | a1824060

Project Vision

Background:
Attackers use a range of methods and tactics to target systems, while those who defend these systems focus on tracking one specific type of attack behaviour. This difference in approach has led to an imbalance between those seeking to breach systems and those guarding them, leading to a widening gap in cybersecurity.

Aims:
This project aims to create a system that supports defenders by constructing a platform for people to upload real-world cyber-attack reports and allow them to annotate the document with tags. It converts the annotated files into a visualisation of connections by using attack flow diagrams, enhancing the understanding of attack methods. These files can be saved and searched with tag filters, making it easier for cybersecurity professionals to collate similar attack methods and support their understanding, and therefore support their work in protecting assets. There is also the potential to streamline the entire process through the employ of semi- or fully-automated systems.

Restrictions:
For this project, the only restriction is the use of proprietary data and tools, a limitation put in place to protect sensitive information and comply with specific licensing agreements. Open-source data and tools are allowed, offering flexibility in alignment with the project's needs.

Customer Q & A

7 August 2023

Can we use Notion for project planning/ management?
Yes we can, the project just needs to be managed in Github.

Can point 4 please be redefined? What is 'to be inserted' and what is it asking?
To validate the system we are creating, attack flow files need to identify the user's knowledge to verify the output:

File upload by client → validation by admin → if valid then it is saved in the database.

Does "annotate" only refer to tags? Or other annotations.
Will be clarified by the client at a later date

Can we be provided with example input and output?
There will be some sample incident report files and their attack flow files

Are we expected to use an attack flow library?
No we aren't.

14 August 2023

We have previously asked 'Does "annotate" only refer to tags? Or other annotations.' 2 weeks ago and were told this would be clarified at a later date. Has this been answered? Can we have an example of an annotated document?
Focusing on the tags such as incident type, date and time, affected systems, and the initial response actions. Told that examples were already provided in the attackflow repo that was initially shared with us.

What are the restrictions of the project?
No restrictions except for not allowed to use proprietary data and tools.

When will we be provided the 'standardised format example' and the input and output examples?
A document is being prepared with more explanations and relevant resources. (Will be uploaded shortly). Told that Prof. Hung asked to use the same format as the MITRE attackflow repo for the standardised format. However this would cause some conflicts from one to another??
Additionally we were able to organise an impromptu meeting with the client (Prof. Hung), a summary of questions and answers are provided below:

15 August 2023
What is the end goal for our project?
We want to produce a large set of quality Attack Flow in an efficient matter

Could you clarify what information we need to include when we generate an attack flow?
Our metadata includes information such as: Who reported the attack, the source of where we got the attack from, what the attack is (e.g. malware on organisation by an attacking group), who the victim and perpetrators are, the date and time of the incident, the location of the attack and relevant documents needed to produce the attack flow including additional sources of the incident on the internet.

Could you explain the process taken from getting the input to producing an attack flow?
We will need to identify all the source documents and put them into a repository. We organise these documents by metadata and generate a document feeder that feeds the documents through. We will need to create a tool for the program to take in data for annotations. Next, we will generate an Attack Flow and organise them plus the supporting documents into another folder. Finally we will need a way to verify what we produced is up to quality and export it to a certain format

How do we know what is a quality attack flow?
To make sure the Attack Flow is faithfully represented, we need to create a notion of quality control. This can be done through a standard format with strict requirements (e.g. action → phase → action). Additionally, we will need to ensure that the system is efficient for humans to use, the user able to create annotations with minimal errors,

What are the input and output types/formats you expect?
Answer: The input is the incident report. We need a component where we scrape the document (input) and annotate it. The output is an Attack Flow report in an XML/AFB file format similar to that of the github project.

Summary/Reflection of kickoff meeting:
In general we felt that the kickoff meeting went poorly. The sheer volume of students attending made it difficult to ask clear questions to the tutor, leading to miscommunications about what was being asked. Additionally, other groups would randomly jump in, trying to clarify their understanding, creating increased difficulty to receive direct responses from the proxy client. The outcome of the kickoff meeting indicated to us our preference for in person meetings, and with only our group present. This is especially true due to lack of response towards questions we asked in the chat.

We did learn that it may be beneficial to have variations of the same question, in the case that our question is not initially understood or misunderstood by the client. It would also help our team be clearer and more defined with what we do and do not understand, and the nuances we need clarification on. As a result, we will be more efficient in our meetings and have a clearer vision of the project.

Users

The primary group of users of the Attackflow generator software is the defenders and leaders that want to understand how adversaries operate and compose atomic techniques into attacks to better understand defensive posture. Next, we provided a more detailed analysis of all possible users:

Role: Cybersecurity Defender
Responsibilities: Provide accurate reports from a valid source.
Potential actions: Upload an incident report file (of any type), annotate a file, save a file, download attack flows, add/remove tags on the uploaded document.

Role: Admin
Responsibilities: Quality control, easy tag navigation.
Potential actions: View submissions, view and edit annotations by any user, inspect and make changes to the AttackFlow database.

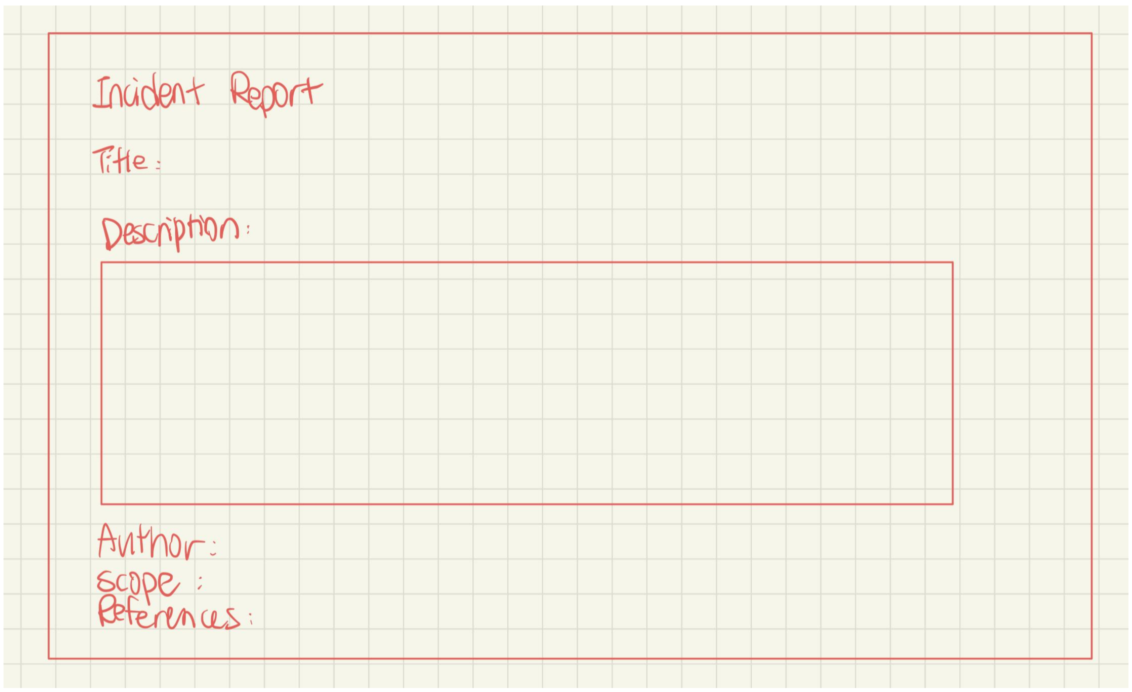
Role: Researcher
Responsibilities: Progress understanding of cyber attacks.
Potential actions: Export the generated attack scenarios and results in formats suitable for academic publications.

Role: Developer
Responsibilities: Create a stable and reliable tool that produces accurate results based on the reports fed into the system.
Potential actions: Extract and prioritise vulnerabilities from the uploaded report, suggest relevant tags, highlight the most relevant information.

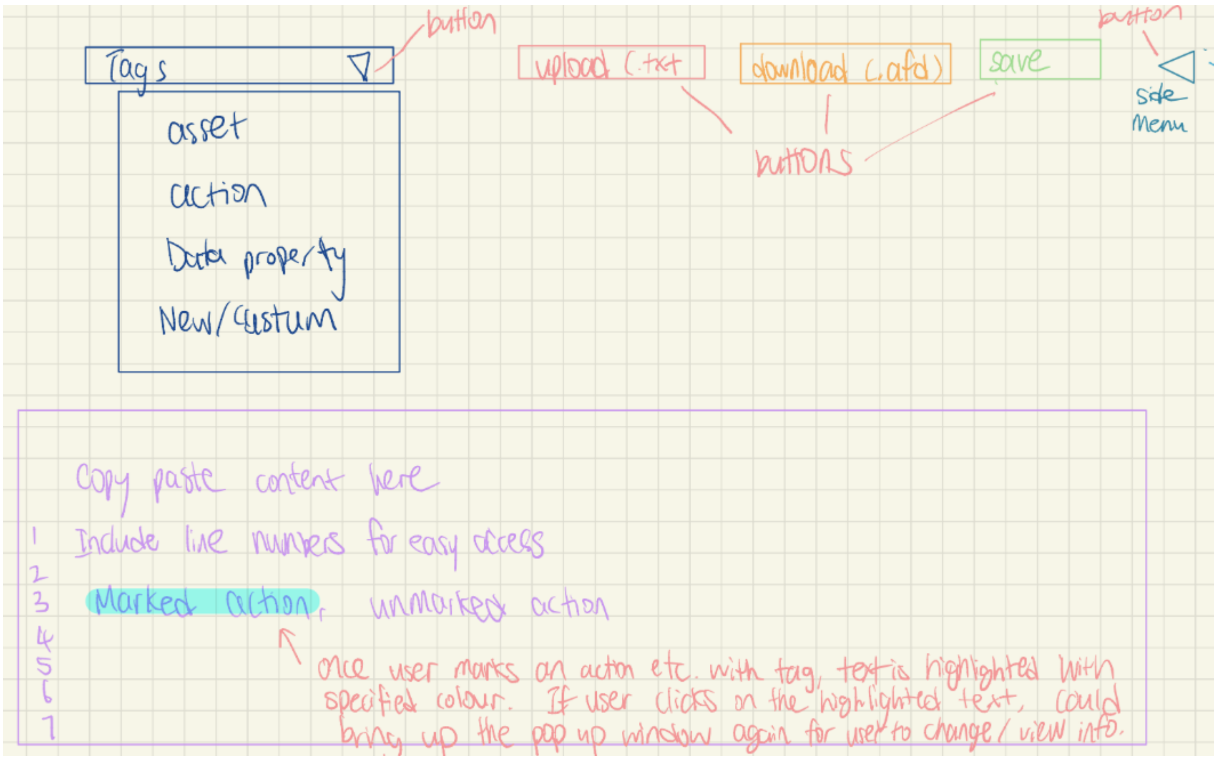
Role: Compliance operator
Responsibilities: Regulatory compliance.
Potential actions: keep track of the uploaded security reports and the corresponding attack scenarios generated.

Software Architecture

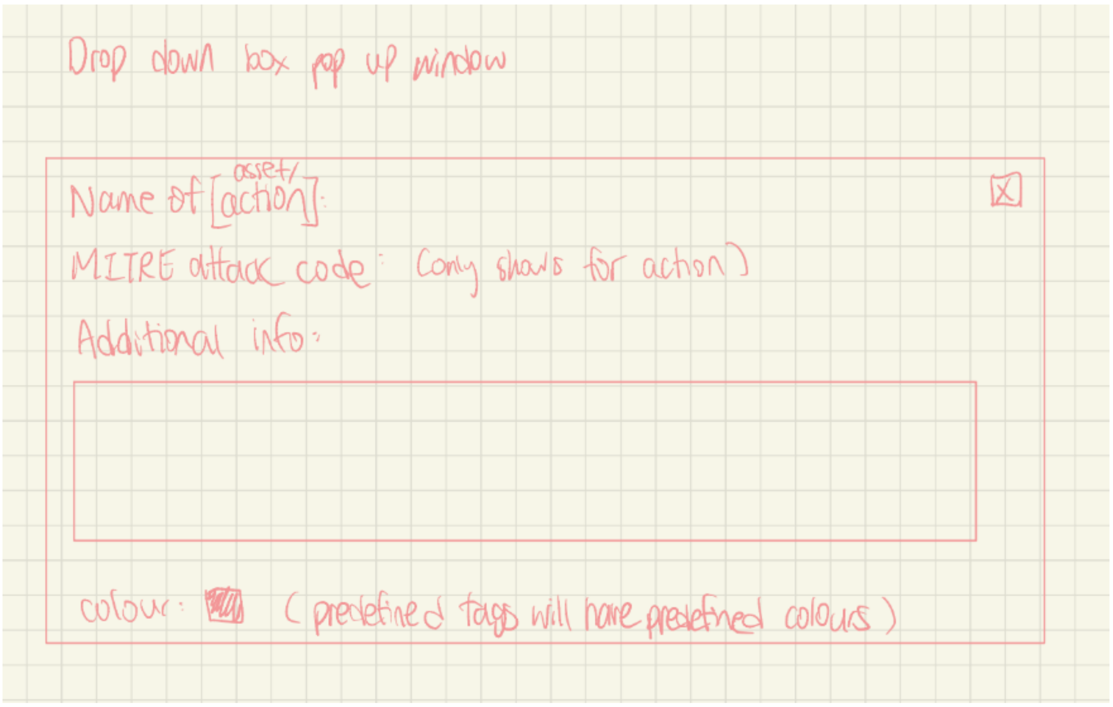
We decided to make a website with a plain text input box, and simple buttons with functions such as tags, upload, download and save. For the incident report, the user could upload a file that gets converted to text via an API, or directly input into the text box. They can enter basic information for the incident report as shown in the red box interface below.



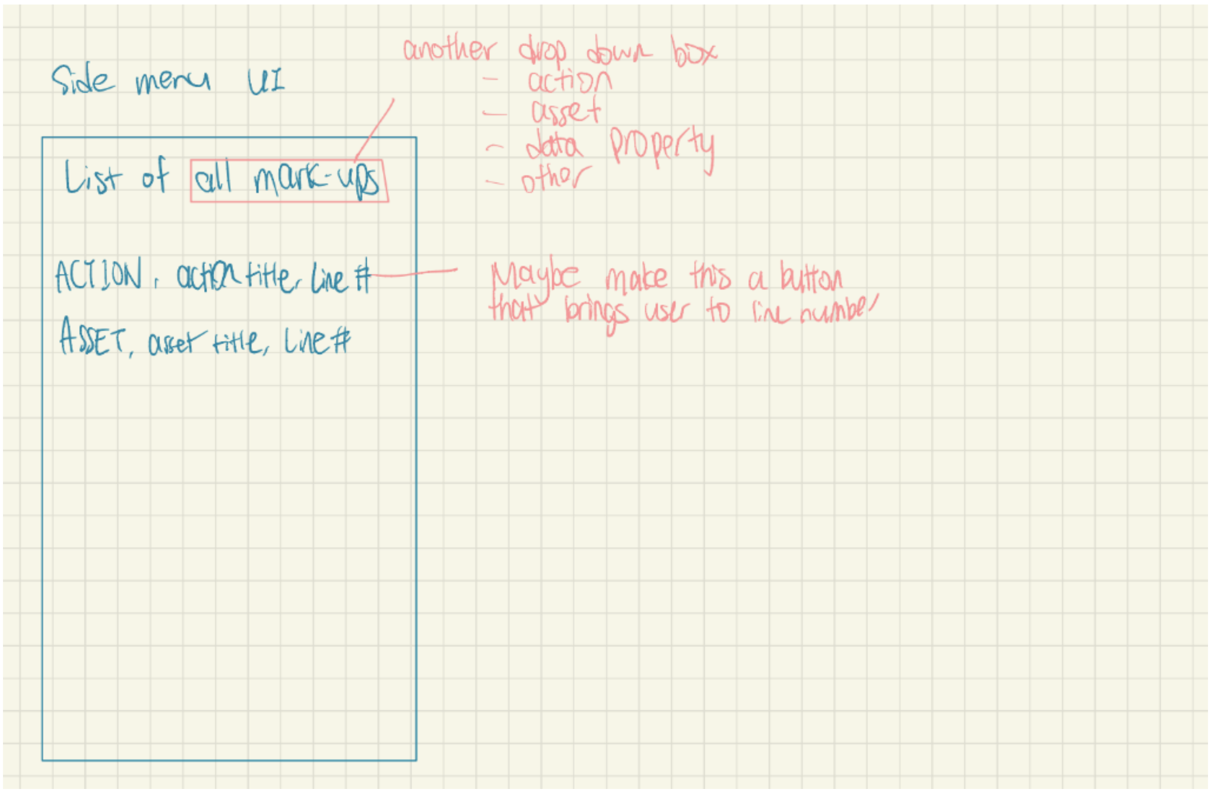
Once the user inputs basic information, they can then choose to highlight important text, and mark them with tags (e.g. Action).


button
Tags
asset
action
Data property
New/custum
upload (.txt
download (.afd)
save
buttons
button
side menu
Copy paste content here
1 Include line numbers for easy access
2
3 Marked action, unmarked action
4
5
6
7
once user marks an action etc. with tag, text is highlighted with specified colour. If user clicks on the highlighted text, could bring up the pop up window again for user to change/view info.

Once the user selects from a drop-down Tags label, they will be greeted with the pop-up box as shown below, where they can enter additional information. Each tag has its unique colour, to differentiate between all the different types of tags.


Drop down box pop up window
Name of [asset/ action]:
MITRE attack code: (only shows for action)
Additional info:
colour: (predefined tags will have predefined colours)

Finally, there is a side menu where the user can review and order the list of markups they have completed.


Side menu UI
List of all mark-ups
another drop down box
- action
- asset
- data property
- other
ACTION, action title, Line #
ASSET, asset title, Line #
Maybe make this a button that brings user to line number

Tech Stack and Standards

The planned tech stack for this application:

Front-end languages:
JavaScript/TypeScript, HTML5, CSS

Front-end frameworks:
Vue.js, Tailwind/Bootstrap, Storybook

Back-end languages:
Python, JSON

Back-end framework:
Django

Communication:
Slack, Discord

Notes:
Notion, Google docs (Google drive)

Development:
VS code, Pycharm

Standards:
In code documentation, in code comments, README file, Version control (Github), Github commit messages

Justifications

These proven, supported software choices fulfil project requirements. Vue offers a scalable, performant front-end with component-based design for efficient code reuse and testing. Django's design allows for rapid and effective testing, helping ensure proper interfacing between back and front end code. Additionally, the provided tools for handling HTTP requests will make development quicker and easier while Python's text processing capabilities will be convenient in the course of developing an annotation system.
For integration and development (IDE), we opt for VS Code and PyCharm, supporting the aforementioned chosen frameworks. GitHub is essential for management of this project. Google Docs streamlines documentation with collaborative editing and contributions. Notion helps us organise the project, and Discord aids communication within the team. Slack is used for formal communication with the proxy client.

Group Meetings and Team Member Roles

Each group member has agreed to a daily short conversation approximately 10-15 minutes long, via a shared Discord chat. We will each share what we have achieved, any issues that have popped up, and what we plan to achieve today.
In addition we have agreed to have in person meetings twice a week (Monday and Thursday) for one to two hours each session. During the Monday meeting the old scrum master shares the current state of the project, then passes over to the new scrum master. They will outline the tasks for the new sprint, which are self-allocated by the team members.

Before each retrospective meeting, each group member should reflect on the closing sprint. They should; prepare discussion points, set personal objectives for the outcome of the meeting, prepare questions, and work on bringing a positive mindset to the meeting. They should prepare, so that they can; share the goals that were achieved and by what means, provide feedback on what could have been done better, and discuss what did not work well during the previous sprint, as well as potential methods for resolving these issues.

The communication between the team and the tutor will occur via the Slack channel created by the tutor in order to receive timely responses for questions and clarifications. Additionally, meetings will be organised with the tutor every fortnight. Email is also a viable communication method as a backup, if Slack is not viable for whatever reason may occur.

The scrum master roles will be as follows:

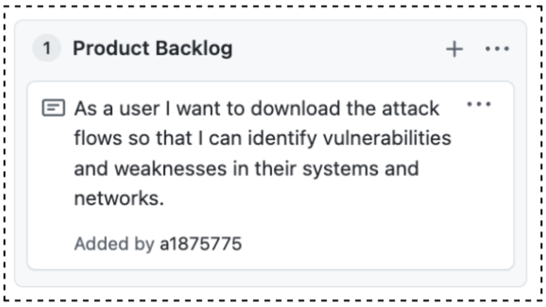
Sprint number:	**Dates:**	**Name:**
1	07/08/23 - 20/08/23	Nathaniel Chang
2	21/08/23 - 10/09/23	Emily Miller
3	11/09/23 - 24/09/23	An Nguyen
4	29/05/23 - 08/10/23	Elana Parnis
5	09/10/23 - 22/10/23	Ethan Pratt
6	23/10/23 - 30/10/23	Angelos Sohan
7	N/A	Wei Long Wan
8	N/A	Jinyang Li
9	N/A	Xizi Wang

Please note that our team began working unofficially before the first sprint began on 14/08/23, treating the zoom kick-off meeting on the 7th as our start date for our first sprint. Regarding sprint 6, if needed, the scrum master must remain in charge until 30/10/23, despite the final report's due date being 26/10/23. In the industry, sprints 7, 8, and 9 are planned; all team members were intended to be scrum masters but following alphabetical order and course constraints limits this.

Snapshot

Product Backlog and Task Board

The product backlog:

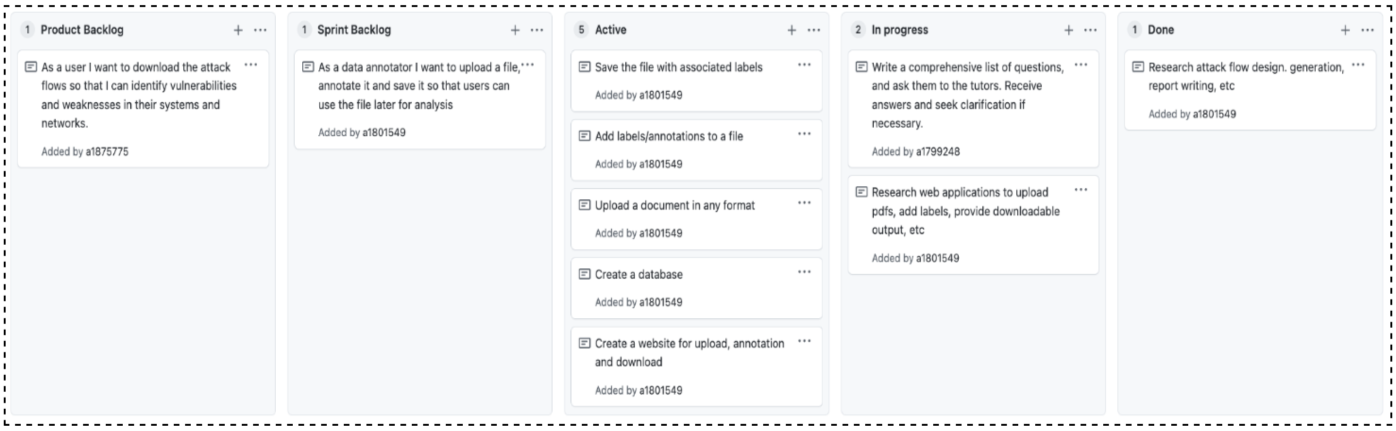


1 Product Backlog

As a user I want to download the attack flows so that I can identify vulnerabilities and weaknesses in their systems and networks.

Added by a1875775

The task board:



1 Product Backlog

As a user I want to download the attack flows so that I can identify vulnerabilities and weaknesses in their systems and networks.
Added by a1875775

1 Sprint Backlog

As a data annotator I want to upload a file, annotate it and save it so that users can use the file later for analysis
Added by a1801549

5 Active

Save the file with associated labels
Added by a1801549

Add labels/annotations to a file
Added by a1801549

Upload a document in any format
Added by a1801549

Create a database
Added by a1801549

Create a website for upload, annotation and download
Added by a1801549

2 In progress

Write a comprehensive list of questions, and ask them to the tutors. Receive answers and seek clarification if necessary.
Added by a1799248

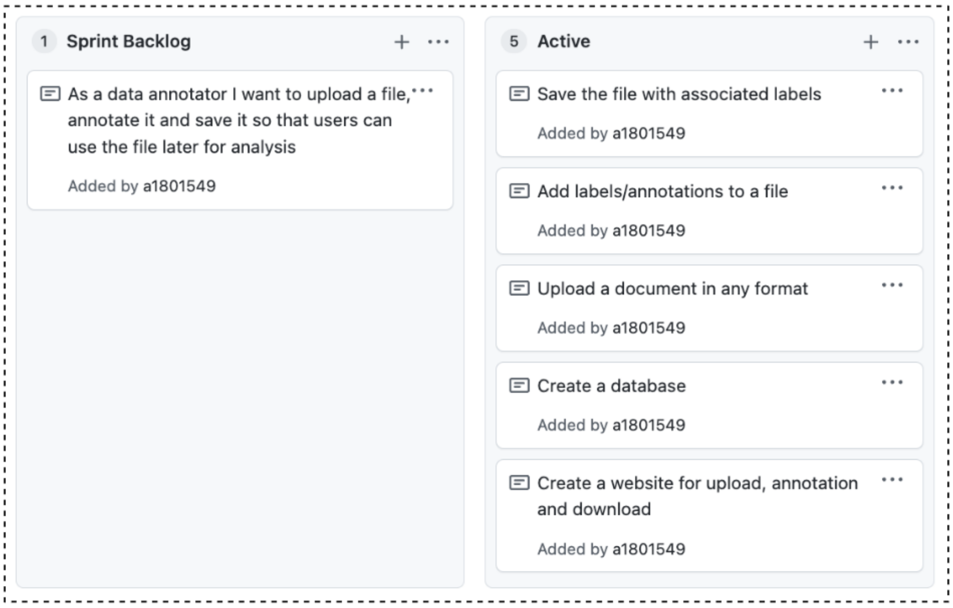
Research web applications to upload pdfs, add labels, provide downloadable output, etc
Added by a1801549

1 Done

Research attack flow design. generation, report writing, etc
Added by a1801549

Sprint Backlog and User Stories

The sprint backlog:


1 Sprint Backlog

As a data annotator I want to upload a file, annotate it and save it so that users can use the file later for analysis

Added by a1801549

5 Active

Save the file with associated labels

Added by a1801549

Add labels/annotations to a file

Added by a1801549

Upload a document in any format

Added by a1801549

Create a database

Added by a1801549

Create a website for upload, annotation and download

Added by a1801549

Description of the user stories selected for the current sprint:**

For the first sprint we have selected the user story;
"As a data annotator I want to upload a file, annotate it and save it so that users can use the file later for analysis"

This describes a user, 'data annotator' which requires the need to upload a data file (of various types), a need to annotate the file (using labels/ tags) and the ability to save the file along with its provided tags for later analysis (which will likely need to be searchable using file name and associated tags). The primary tasks selected from the user story are: the web platform required to access the software, file upload, annotation of files using tags (both predefined and user-defined) and storage of the file with tags in a database.

*** Note that due to delay in receiving information, progress was limited and this week was entirely theoretical in the work achieved. The user story below was unable to be worked on, but was discussed in theory as part of our design.*

Definition of Done

At the conclusion of this project, our application should be ready for deployment and actualisation. Throughout the project, the term "done" denotes the completion of a particular component. It is attained after a thorough review by both a team member and the featured client, has incorporated feedback, and undergone subsequent review. Additionally, a component achieves "done" status when it aligns with the client's brand guidelines, ensures a consistent and integrated visual identity, encapsulates technical proficiency, and adheres to design standards. A more specific definition of "done" for each of the predicted aspects is outlined below:

1. **Website Functionality:**
    - The website is fully functional, allowing users to upload files.
    - Annotations are automatically generated based on the uploaded file's content.
    - Users have the capability to manually add annotations to the file.
    - The website's user interface is responsive, intuitive, and accessible.
    - User reviews all annotations prior to saving data to the database
2. **Attack Flow Creation:**
    - An attack flow is generated accurately based on the report's content.
    - The attack flow represents the sequence of events and vulnerabilities identified in the report.
    - The attackflow generator already built by the client will be used to produce attack flows based on the annotations extracted.
3. **Database Integration:**
    - The annotated report is inserted into the database after manual annotation.
    - Data integrity is ensured during the database insertion process.
4. **Report Download:**
    - Users can successfully download the annotated report from the website.
    - The downloaded report includes all annotations and attack flow visualisations.
5. **Testing and Quality Assurance:**
    - The website functionalities are thoroughly tested to ensure proper behaviour.
    - Any bugs, errors, or inconsistencies are resolved before considering the user story "done."
6. **Documentation:**
    - Comprehensive documentation is provided, including user instructions and technical details.
    - The documentation assists users in understanding how to utilise the website's features effectively.

Summary of Changes

As this is the initial snapshot, there are no changes to summarise. Therefore, its purpose is to mark the foundation of this project, and be a reference point for future developments where alterations may have occurred.