Report:

Title Page:

- Title: Retrospective Sprint 2 of Group AttackFlow1
- Building a dataset of real-world cyber-attacks with Attack Flow
- List of team members with their a-numbers:
  - Se Jin Yoon: a1706219
  - Ting-Wei Chin: a1782423
  - Faisal Hawsawi: a1822781
  - Lina Nehme: a1802697
  - Ran Qi: a1675122
  - Joseph Toubia: a1753547
  - Zemin Wong: a1780385
  - Jixing Ye: a1798631
  - Yu Zheng: a1739446

Snapshots (Group):

I attended the sprint planning meetings on August 18&23&25&30 and September 1&6&8 with the teammates and meetings on August 23 and September 6 with tutor, here are the 2 snapshots we submitted:

Product Backlog (See Figure 1&2)

Task Board (See Figure 3&4)

Sprint Backlog (See Figure 5)

User Stories & DoD

After gathering user stories in the first sprint and having a sprint planning meeting with the tutor, we selected some user stories from the product backlog, which are the most fundamental functions of the project – upload and annotate. After selecting user stories, we were able to derive tasks for this sprint:

• "As a data annotator, I want to upload an incident report and save it so that I can further edit the file."

– Related tasks:

task 1.1: build a web interface of home page task 1.1.2: Build a Mockup to show the layout of front end and user interactions

task 1.2: upload function

task 1.3.1: build a database schema

task 1.3.2: implement database to store original documents

• "As an annotator, I want to annotate incident report so I can generate attack flow."

– Related tasks:

task 2.1.1: decide how user annotate

task 2.1.2: implement user annotation function

• "As a viewer, I want to track the changes made by different users, so I compare the difference and revert changes if need to."

– Related tasks:

task 3.1: implement version control function in front end (I.e. allow users to retrieve history version or roll back annotations)

• "User can log in as administrator, annotator, and general public."

– Related tasks:

task 4.1: implement log in

task 4.2: implement different user roles in the database

Definition of Done

Snapshot 2:

The individual DoD of each task can be seen on the screenshot of the task board.

In summary, the goals are:

• Testing the interface works as we expected.

• Explain and ensure that all the scrum members understand how to build up the implementation.

• Explain and declare the tasks to each scrum member regarding the web design.

Snapshot 3:

- The user interface and user experience adhere to the provided design specifications.
- The user story mockup is ready for presentation in the sprint review
- Sufficient documentation is provided.
- Coding standards are used.

Summary of Changes:

Snapshot 2:

The last snapshot was taken in the first sprint, which was mainly about collecting user stories, researching, and setting up the environment. This snapshot shifts the focus on implementing user stories gathered from the first sprint, emphasising the most fundamental functional requirements: upload and annotate.

- Planned a different set of tasks. We are pleased that we successfully completed all tasks from the previous snapshot. In this snapshot, We selected a new collection of user stories from the product backlog and we are embarking on a fresh set of tasks, with a strong emphasis on implementing the web UI and database components.
- Added a new user story. During our meeting with the tutor, we discovered a new user story about allowing the admin to review user annotation and approve or reject based on a set of rules. We added the new user story to the product backlog.
- Started on the mockup. Before jumping into coding, we need to communicate with the clients and within the team as well in terms of what the product should look like, how will the customer interact with it, etc. We decided to use Figma to build a simple mock up for this purpose.
- Drafted database schema The database schema has been drafted and is pending review by the team members at the moment.
- Started coding. In this second snapshot, The team has been actively working on the web front-end(interface and implementation) and backend(database and express.js). Currently, we don’t have a mockup yet so we can only write some skeleton code and set up the frameworks.

Snapshot 3:

Last week’s snapshot had a focus on the most fundamental functional requirements: upload and annotate. We allocated tasks on database design and front-end coding but quickly realized that we couldn’t jump into coding without agreeing on a general design first. So this week we made a mockup in preparation for the coming client meeting with the tutor, where we will demonstrate the website design and validate user requirements before the building starts.

- A mockup was made using Figma. The mockup shows the layout of the website homepage, the user interactions when uploading files, searching in the attackflow database, and annotating reports (not yet finished).

We discussed the design during a team meeting and were able to reach an agreement.

• The front-end development environment is set up. Vue, node.js are installed and skeleton code is written for the website homepage based on the design of the mockup. The front-end code is uploaded to the git repository and read-me files were added/edited.
• Database initial setup is complete. A database schema is made and a database was setup using MariaDB.

What went well in the sprint?

We not only achieved a harmonious collaboration but also laid a robust groundwork for our project, setting a positive trajectory for future developments. The meticulous planning and execution of tasks, coupled with a clear vision delineated through the mockup, are testament to a sprint well executed. Especially on:

Mockup Development

Before delving into coding, we recognized the importance of having a clear vision of what the end product should encompass. Hence, we utilized Figma to create a mockup, illustrating the layout of the website homepage and depicting user interactions for various functionalities such as uploading files and searching in the attack flow database. This mockup served as a blueprint, facilitating discussions and agreements during team meetings.

Database Schema

We drafted a database schema, which is currently under review by team members. This schema will play a crucial role in the subsequent stages of development, ensuring a structured approach to database management.

What could be improved?

Although we did much work on frontend, jobs on debugging and connection with backend are still under discussion. The good news is we had agreed to go with Express for the backend framework, MySQL as our database, Sequelize as our ORM, and Axios for client-server communication.
From the task board snapshot, it is observed that there are tasks in various stages - to do, in progress, and done. Streamlining the task allocation process and adhering to the estimated deadlines would foster a more productive work environment. The task of implementing a function to retrieve data from the database to the website was estimated to take 2 hours but is still in progress. Enhancing the accuracy of time estimates and ensuring timely completion of tasks would be beneficial.

What will the group commit to improve in the next sprint?

We are committed to fostering a collaborative work environment, where each team member actively contributes to the project's success. This includes ensuring that all scrum members understand how to build up the implementation and assigning tasks to each member based on their expertise and the project's requirements.

As we prepare for the upcoming client meeting with the tutor, we are committed to ensuring that our presentation is well-prepared, showcasing our mockup and elucidating the user requirements to gather valuable feedback and insights.

In essence, our commitments for the next sprint are centered around enhancing the user experience, leveraging technological advancements, and fostering a collaborative work environment. We are poised to not only address the areas of improvement identified in the previous sprint but also to set new benchmarks of excellence

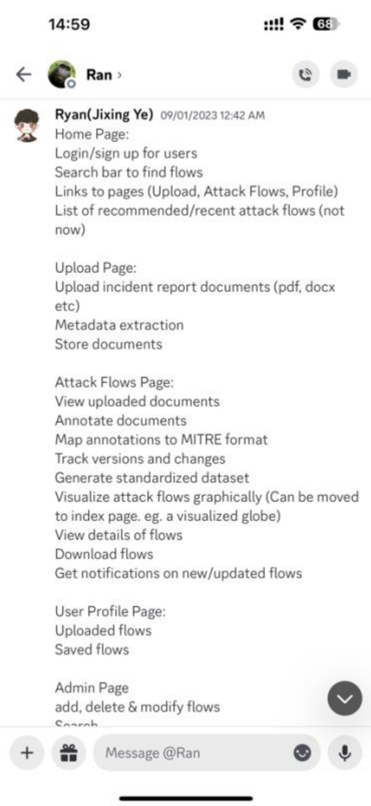
in our project development journey.

Comment on your progress this sprint:

What I have done:

1. Participated in all the meetings and engaged in sprint discussions.
2. Work with Ran Qi to build the website mockup with Figma.

Evidence of our communication: (Since both of us are Chinese)



14:59

Ran ›

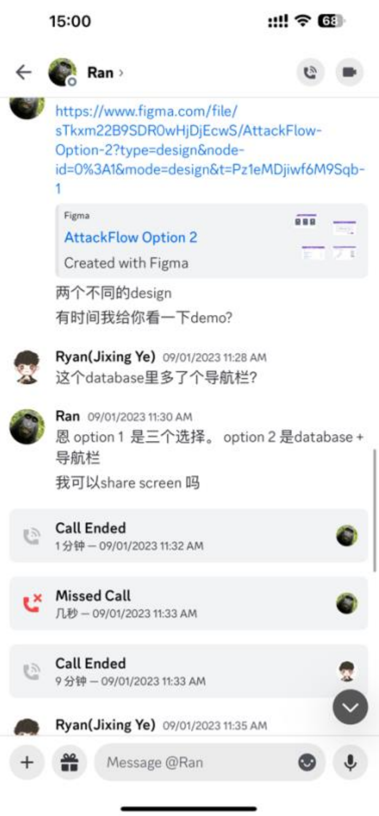
Ryan(Jixing Ye) 09/01/2023 12:42 AM
Home Page:
Login/sign up for users
Search bar to find flows
Links to pages (Upload, Attack Flows, Profile)
List of recommended/recent attack flows (not now)

Upload Page:
Upload incident report documents (pdf, docx etc)
Metadata extraction
Store documents

Attack Flows Page:
View uploaded documents
Annotate documents
Map annotations to MITRE format
Track versions and changes
Generate standardized dataset
Visualize attack flows graphically (Can be moved to index page. eg. a visualized globe)
View details of flows
Download flows
Get notifications on new/updated flows

User Profile Page:
Uploaded flows
Saved flows

Admin Page
add, delete & modify flows
Search



15:00

Ran ›

https://www.figma.com/file/sTkxm22B9SDR0wHjDjEcwS/AttackFlow-Option-2?type=design&node-id=0%3A1&mode=design&t=Pz1eMDjiwf6M9Sqb-1

Figma
AttackFlow Option 2
Created with Figma

两个不同的design
有时间我给你看一下demo?

Ryan(Jixing Ye) 09/01/2023 11:28 AM
这个database里多了个导航栏?

Ran 09/01/2023 11:30 AM
恩 option 1 是三个选择。 option 2 是database + 导航栏
我可以share screen 吗

Call Ended
1 分钟 — 09/01/2023 11:32 AM

Missed Call
几秒 — 09/01/2023 11:33 AM

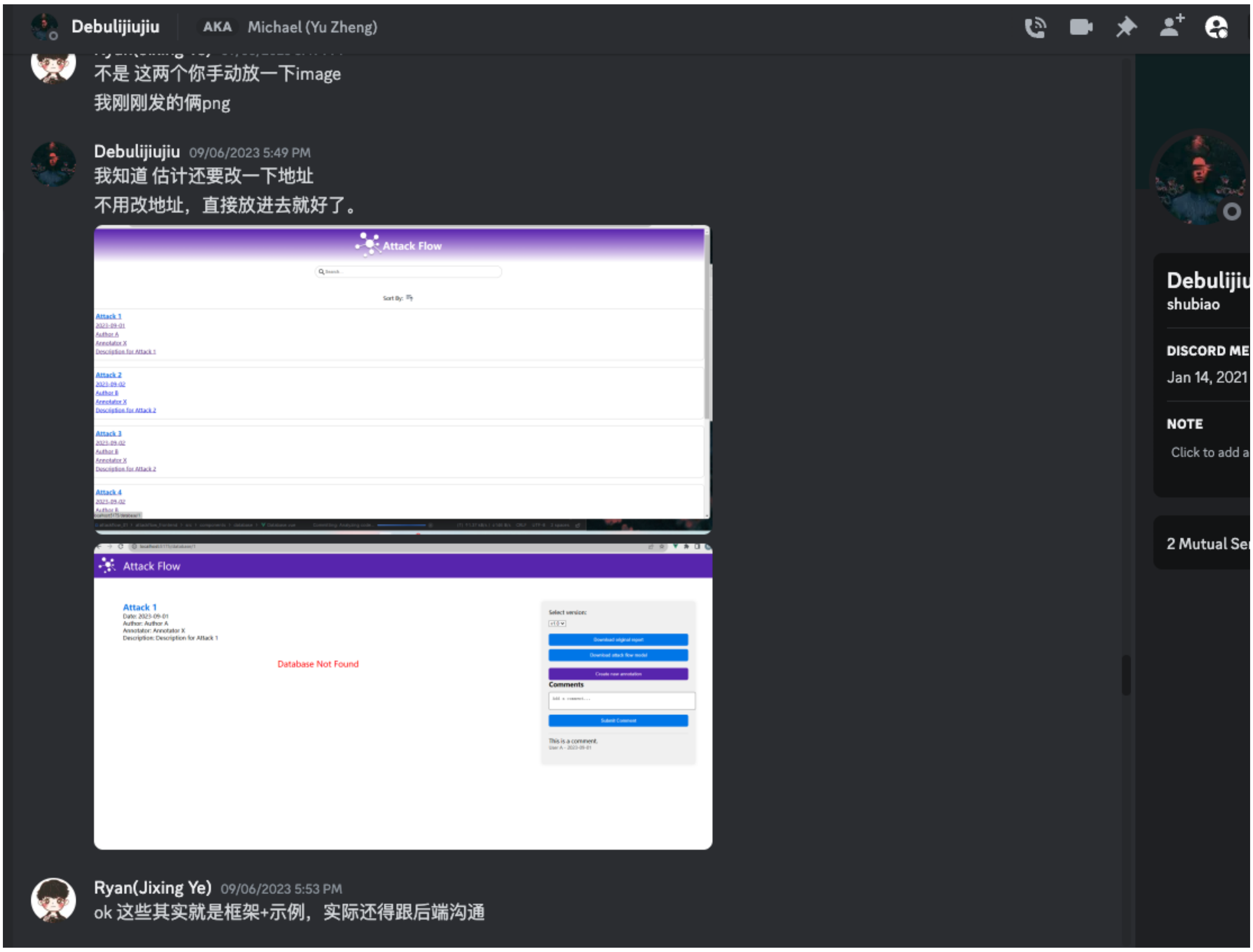
Call Ended
9 分钟 — 09/01/2023 11:33 AM

Ryan(Jixing Ye) 09/01/2023 11:35 AM

3. Work with Yu Zheng to build the web interface by using IntelliJ Idea. I build the database part with Database.vue, Detail.vue and Posts.vue to achieve a well-structured, user-friendly web application for viewing a database of "attacks." The main component is Database.vue, which serves as a container and master layout for listing all the posts. It imports and uses Header.vue and Posts.vue components to build its structure. The Database.vue implements features such as a search bar, sort functionality, and pagination, with a pre-defined list of "attack" posts. The Posts.vue component is designed to display individual posts succinctly, utilizing Vue's router-link feature to link to a detailed view of each post, implying a single-page application (SPA) architectural design. This detailed view is constructed in the Detail.vue file, which provides a comprehensive look at a single post, and facilitates user interaction such as selecting different versions of a post and commenting functionality, leveraging Vue's reactivity system through the ref function to handle dynamic data.

   We also enhanced a trial on annotating the attackflow data with generated AI.

Evidence of our communication and my work: (Since both of us are Chinese)



不是 这两个你手动放一下image
我刚刚发的俩png

Debulijiujiu 09/06/2023 5:49 PM
我知道 估计还要改一下地址
不用改地址，直接放进去就好了。

Attack 1
2023-09-01
Author A
Annotator X
Description for Attack 1

Attack 2
2023-09-02
Author B
Annotator X
Description for Attack 2

Attack 3
2023-09-02
Author B
Annotator X
Description for Attack 2

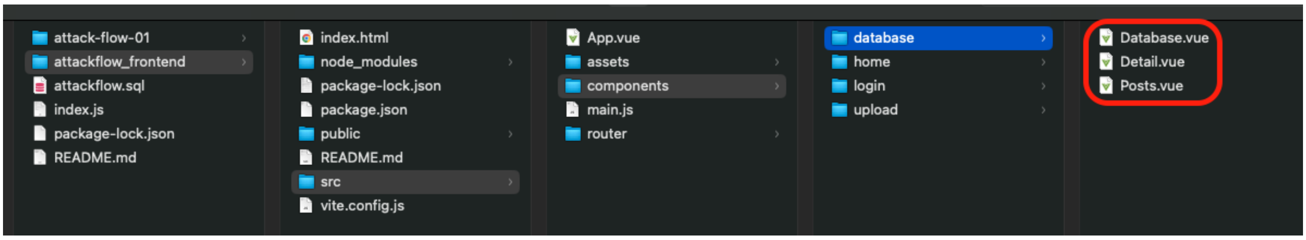
Attack 4
2023-09-02
Author B

Attack 1
Date: 2023-09-01
Author: Author A
Annotator: Annotator X
Description: Description for Attack 1

Database Not Found

This is a comment.
User A - 2023-09-01

Ryan(Jixing Ye) 09/06/2023 5:53 PM
ok 这些其实就是框架+示例，实际还得跟后端沟通



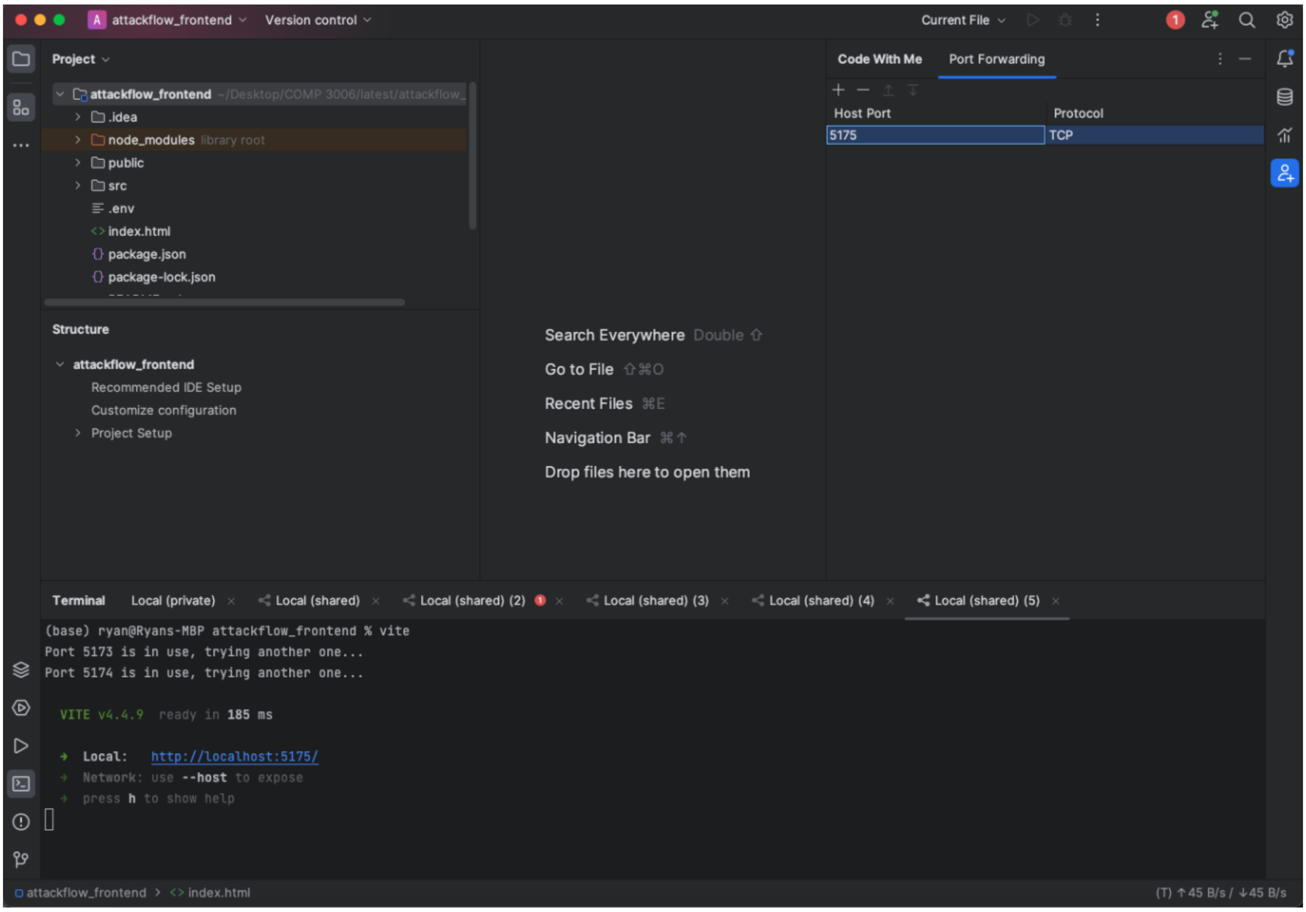
attack-flow-01
attackflow_frontend
attackflow.sql
index.js
package-lock.json
README.md

index.html
node_modules
package-lock.json
package.json
public
README.md
src
vite.config.js

App.vue
assets
components
main.js
router

database
home
login
upload

Database.vue
Detail.vue
Posts.vue



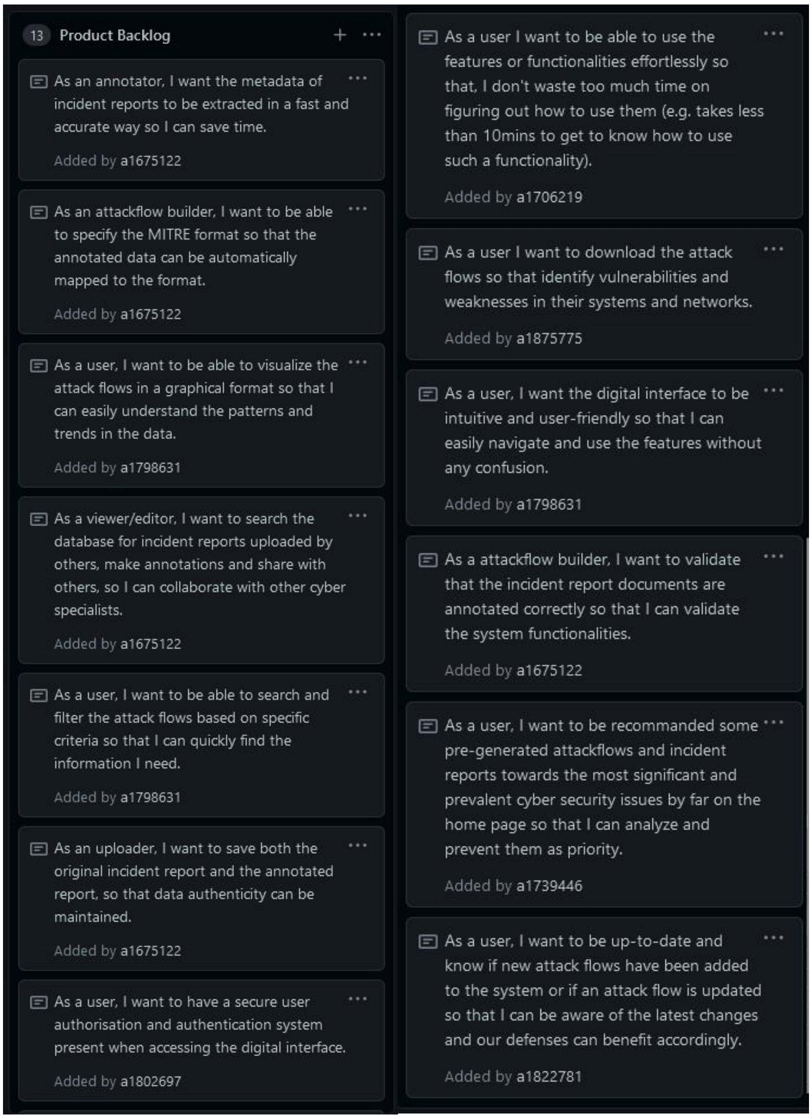
Host Port	Protocol
5175	TCP

```
(base) ryan@Ryans-MBP attackflow_frontend % vite
Port 5173 is in use, trying another one...
Port 5174 is in use, trying another one...

  VITE v4.4.9  ready in 185 ms

  ➜  Local:   http://localhost:5175/
  ➜  Network: use --host to expose
  ➜  press h to show help
```

Appendix



As an annotator, I want the metadata of incident reports to be extracted in a fast and accurate way so I can save time.

Added by a1675122

As an attackflow builder, I want to be able to specify the MITRE format so that the annotated data can be automatically mapped to the format.

Added by a1675122

As a user, I want to be able to visualize the attack flows in a graphical format so that I can easily understand the patterns and trends in the data.

Added by a1798631

As a viewer/editor, I want to search the database for incident reports uploaded by others, make annotations and share with others, so I can collaborate with other cyber specialists.

Added by a1675122

As a user, I want to be able to search and filter the attack flows based on specific criteria so that I can quickly find the information I need.

Added by a1798631

As an uploader, I want to save both the original incident report and the annotated report, so that data authenticity can be maintained.

Added by a1675122

As a user, I want to have a secure user authorisation and authentication system present when accessing the digital interface.

Added by a1802697

As a user I want to be able to use the features or functionalities effortlessly so that, I don't waste too much time on figuring out how to use them (e.g. takes less than 10mins to get to know how to use such a functionality).

Added by a1706219

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

Added by a1875775

As a user, I want the digital interface to be intuitive and user-friendly so that I can easily navigate and use the features without any confusion.

Added by a1798631

As a attackflow builder, I want to validate that the incident report documents are annotated correctly so that I can validate the system functionalities.

Added by a1675122

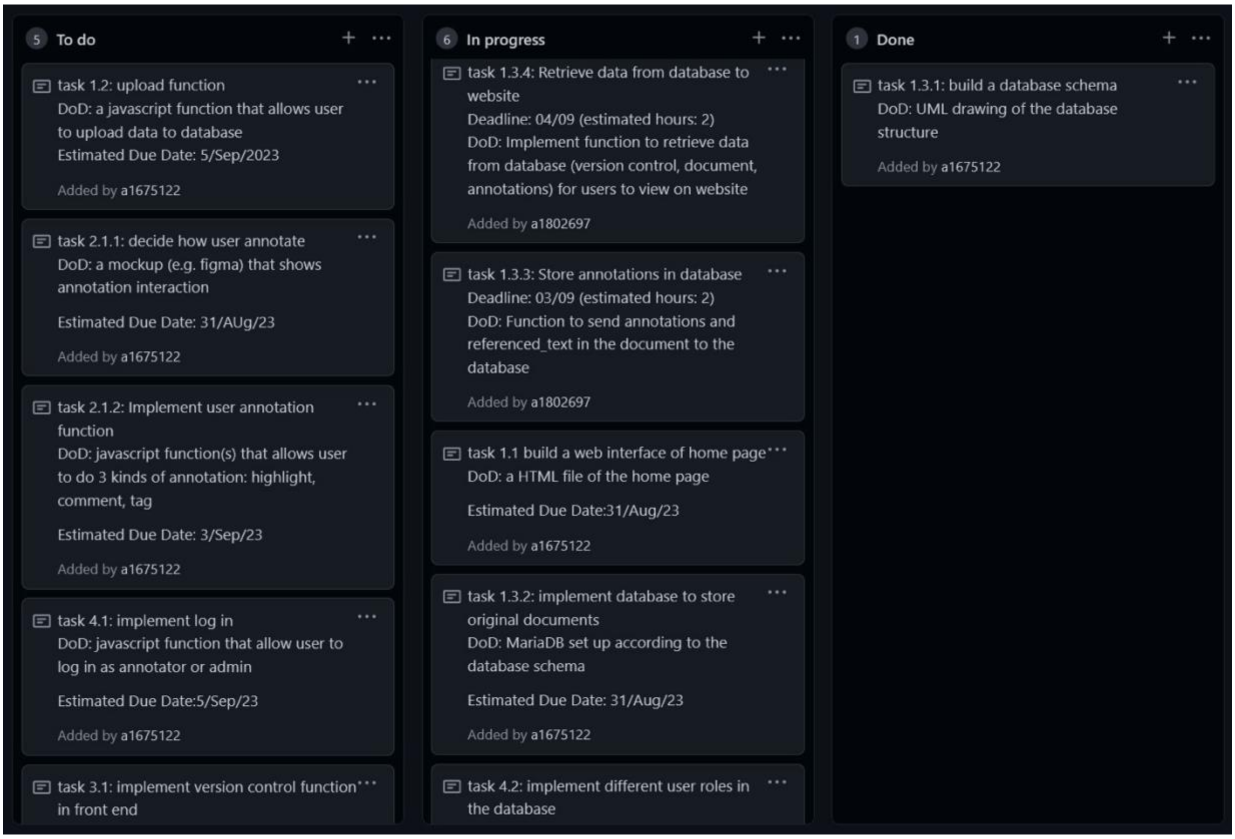
As a user, I want to be recommanded some pre-generated attackflows and incident reports towards the most significant and prevalent cyber security issues by far on the home page so that I can analyze and prevent them as priority.

Added by a1739446

As a user, I want to be up-to-date and know if new attack flows have been added to the system or if an attack flow is updated so that I can be aware of the latest changes and our defenses can benefit accordingly.

Added by a1822781

Figure 1&2: Product Backlog (Sprint 2&3)



5 To do

task 1.2: upload function
DoD: a javascript function that allows user to upload data to database
Estimated Due Date: 5/Sep/2023

Added by a1675122

task 2.1.1: decide how user annotate
DoD: a mockup (e.g. figma) that shows annotation interaction

Estimated Due Date: 31/AUg/23

Added by a1675122

task 2.1.2: Implement user annotation function
DoD: javascript function(s) that allows user to do 3 kinds of annotation: highlight, comment, tag

Estimated Due Date: 3/Sep/23

Added by a1675122

task 4.1: implement log in
DoD: javascript function that allow user to log in as annotator or admin

Estimated Due Date:5/Sep/23

Added by a1675122

task 3.1: implement version control function in front end

6 In progress

task 1.3.4: Retrieve data from database to website
Deadline: 04/09 (estimated hours: 2)
DoD: Implement function to retrieve data from database (version control, document, annotations) for users to view on website

Added by a1802697

task 1.3.3: Store annotations in database
Deadline: 03/09 (estimated hours: 2)
DoD: Function to send annotations and referenced_text in the document to the database

Added by a1802697

task 1.1 build a web interface of home page
DoD: a HTML file of the home page

Estimated Due Date:31/Aug/23

Added by a1675122

task 1.3.2: implement database to store original documents
DoD: MariaDB set up according to the database schema

Estimated Due Date: 31/Aug/23

Added by a1675122

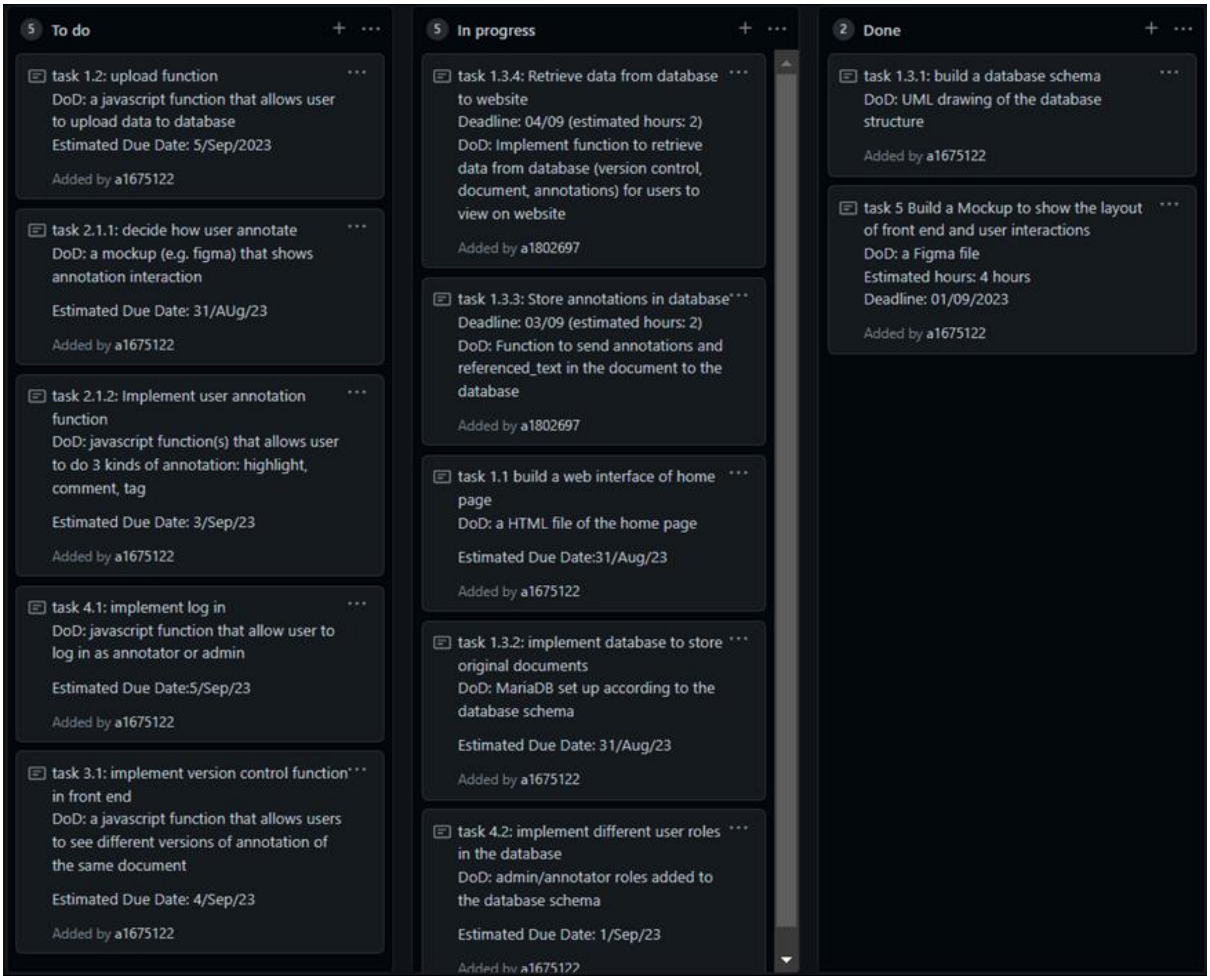
task 4.2: implement different user roles in the database

1 Done

task 1.3.1: build a database schema
DoD: UML drawing of the database structure

Added by a1675122

Figure 3: Task board (Sprint 2)



5 To do

task 1.2: upload function
DoD: a javascript function that allows user to upload data to database
Estimated Due Date: 5/Sep/2023
Added by a1675122

task 2.1.1: decide how user annotate
DoD: a mockup (e.g. figma) that shows annotation interaction
Estimated Due Date: 31/AUg/23
Added by a1675122

task 2.1.2: Implement user annotation function
DoD: javascript function(s) that allows user to do 3 kinds of annotation: highlight, comment, tag
Estimated Due Date: 3/Sep/23
Added by a1675122

task 4.1: implement log in
DoD: javascript function that allow user to log in as annotator or admin
Estimated Due Date:5/Sep/23
Added by a1675122

task 3.1: implement version control function in front end
DoD: a javascript function that allows users to see different versions of annotation of the same document
Estimated Due Date: 4/Sep/23
Added by a1675122

5 In progress

task 1.3.4: Retrieve data from database to website
Deadline: 04/09 (estimated hours: 2)
DoD: Implement function to retrieve data from database (version control, document, annotations) for users to view on website
Added by a1802697

task 1.3.3: Store annotations in database
Deadline: 03/09 (estimated hours: 2)
DoD: Function to send annotations and referenced_text in the document to the database
Added by a1802697

task 1.1 build a web interface of home page
DoD: a HTML file of the home page
Estimated Due Date:31/Aug/23
Added by a1675122

task 1.3.2: implement database to store original documents
DoD: MariaDB set up according to the database schema
Estimated Due Date: 31/Aug/23
Added by a1675122

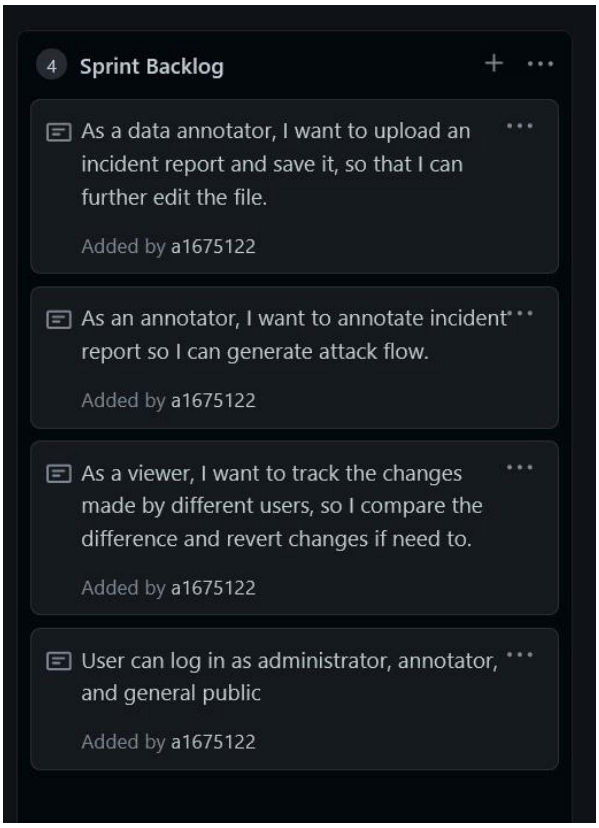
task 4.2: implement different user roles in the database
DoD: admin/annotator roles added to the database schema
Estimated Due Date: 1/Sep/23
Added by a1675122

2 Done

task 1.3.1: build a database schema
DoD: UML drawing of the database structure
Added by a1675122

task 5 Build a Mockup to show the layout of front end and user interactions
DoD: a Figma file
Estimated hours: 4 hours
Deadline: 01/09/2023
Added by a1675122

Figure 4: Task board (Sprint 3)



4 Sprint Backlog

As a data annotator, I want to upload an incident report and save it, so that I can further edit the file.

Added by a1675122

As an annotator, I want to annotate incident report so I can generate attack flow.

Added by a1675122

As a viewer, I want to track the changes made by different users, so I compare the difference and revert changes if need to.

Added by a1675122

User can log in as administrator, annotator, and general public

Added by a1675122

Figure 5: Sprint Backlog (Sprint 2&3)