Attack Flow

Retrospective Sprint 3 of Group Attackflow 10

Jie Shen Beh (a1834032)
Jian Zhe Chan (a1813851)
Gia Bao Hoang (a1814824)
Marcus Hoang (a1814303)
Guan Chern Liew (a1837053)
Vinh Diem Nguyen (a1838114)
Hoang Nam Trinh (a1807377)
Hung Yee Wong (a1815836)
Jiajun Yu (a1806320)

Individually Written:

What went well in the sprint?

During the recent sprint, numerous factors underscored our successes, and they all revolved around the unyielding team synergy. The collaboration and the shared drive among members played a pivotal role in amplifying the development process. One prime example was our proactive integration of Auth0. Instead of waiting for security issues to arise, we took the initiative, demonstrating both foresight and a commitment to enhancing our software's security and reliability. This proactiveness wasn't limited to just Auth0; our team was agile enough to identify and rectify the Vite import bug promptly. Addressing this not only patched a technical hiccup but also streamlined the workflow for the entire team, eliminating unnecessary delays.

Furthermore, our emphasis on quality assurance was apparent through the comprehensive test cases developed specifically for Auth0. These tests ensured that while we were advancing in terms of features and security. A significant achievement was our renewed focus on user experience. We redesigned our homepage and had a guide demo video for users.

In sum, the successes of this sprint, rooted deeply in our collaborative approach and preemptive measures, not only smoothed our development trajectory but also elevated the end product. This holistic view and understanding of software development, where we seamlessly married technical implementations with user needs and team processes, made the sprint a resounding success.

What could be improved?

During our sprint, there were areas that required enhancement. One significant challenge was the time constraint due to overlapping assignments from other courses. Further complicating matters was the GitHub license issue, which caused notable delays. In hindsight, we could've adopted alternative strategies like working solely locally, utilizing classic file-sharing methods like zipping, or increasing our face-to-face interactions.

On the technical front, although our annotation was nearing completion, we couldn't finalize it because of misunderstandings in task allocations and minor glitches. This led to additional days spent and an unexpected pair-programming session to troubleshoot. More detailed task planning and regular catch-up sessions could've mitigated this. The looming deadlines from other courses also influenced this oversight.

Additionally, we should've sought more user feedback. It would have offered critical insights to refine our user interface and experience. Nevertheless, considering the challenges, our team demonstrated resilience. We managed to navigate through the hurdles, staying within the estimated completion times for most tasks.

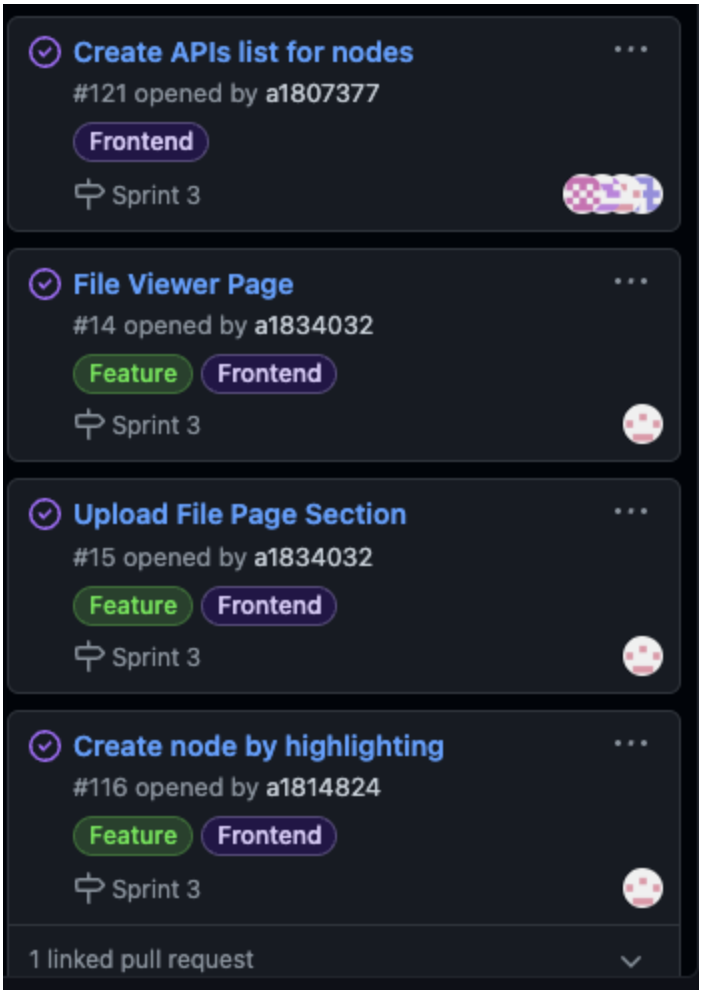
What will the group commit to improve in the next sprint?

In the upcoming sprint, several improvements are paramount. First and foremost, we must refine our task allocation strategy to leverage each member's strengths better. This adjustment will optimize our workflow, ensuring we expedite and finalize tasks, especially considering this sprint's crucial nature where project finalization coincides with exam preparations for other courses. Thus, meticulous planning is necessary, supplemented by increased meetings to ensure alignment and prevent any deviations from our core objectives.

On the project's technical side, we'll embark on developing the attack flow visualizations and strategizing the version control feature. We've identified a promising library to aid in the visualization process. While we possess preliminary concepts for the version control feature, it mandates a comprehensive discussion before initiation. Additionally, our target is to conclude the annotation task within the sprint's first week, following which we will focus on devising more test cases.

In essence, while challenges persist, our trajectory remains positive. With a touch more diligence and concerted effort, we're confident in delivering the project within the stipulated time frame.

Comment on your progress this sprint


Create APIs list for nodes
#121 opened by a1807377
Frontend
Sprint 3
File Viewer Page
#14 opened by a1834032
Feature
Frontend
Sprint 3
Upload File Page Section
#15 opened by a1834032
Feature
Frontend
Sprint 3
Create node by highlighting
#116 opened by a1814824
Feature
Frontend
Sprint 3
1 linked pull request

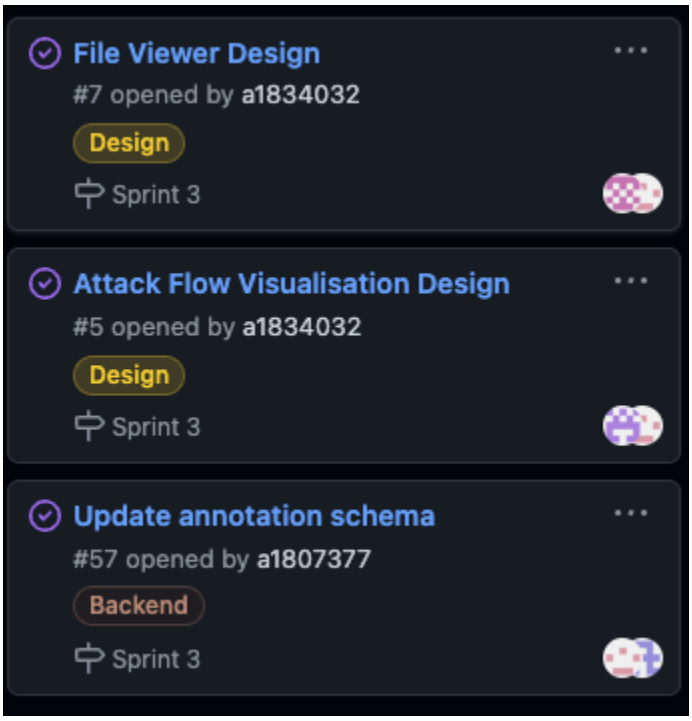

File Viewer Design
#7 opened by a1834032
Design
Sprint 3
Attack Flow Visualisation Design
#5 opened by a1834032
Design
Sprint 3
Update annotation schema
#57 opened by a1807377
Backend
Sprint 3

Image 1: Issues in Sprint 3

During this sprint, I feel my contributions have been significant, particularly concerning the annotation segment which is nearing completion. Collaborating closely with Hung Yee Wong and Hoang Nam Trinh, I played a pivotal role in crafting the File Viewer Page and the Upload File Page Section. A notable challenge arose when integrating these pages with the 'react-pdf-highlighter' library. Hoang Nam Trinh and I had to engage in two intensive pair programming sessions to iron out integration issues.

Furthermore, to enhance user-friendliness, we introduced a feature allowing nodes to be created by highlighting specific text segments. Crafting this solution posed a unique challenge since the library didn’t natively support it, but with perseverance, a workaround was achieved. After feature finalization, I collaborated with three team members to articulate the required APIs for nodes, which we subsequently communicated to the backend team. This feature evolution also necessitated an update to the Annotation schema; I took charge of the draft version, liaising with the database team for its implementation.

Preparing for our next steps, Hung Yee Wong and I proactively identified a suitable library for visualization and undertook preliminary UI design tasks.

Additionally, I lent support to Vinh Diem Nguyen on his assignment and assisted others with frontend challenges. These collaborative instances were not only about contribution but also served as invaluable learning experiences, providing insights into the intricacies of managing extensive projects.

To summarize, I feel confident about my performance this sprint, having successfully navigated individual responsibilities while also fostering team synergy to surmount challenges.

Snapshots

At 7:00pm 03/10/2023, my team and I had a sprint review meeting with the tutor where we present what we've done in the previous meeting and clarify some information for the project. We received some feedback about our progress and we will try to fix it in the next sprint.

Attack Flow

Snapshot Week 7 of Group AttackFlow 10

Jie Shen Beh (a1834032)
Jian Zhe Chan (a1813851)
Gia Bao Hoang (a1814824)
Marcus Hoang (a1814303)
Guan Chern Liew (a1837053)
Vinh Diem Nguyen (a1838114)
Hoang Nam Trinh (a1807377)
Hung Yee Wong (a1815836)
Jiajun Yu (a1806320)

Product Backlog and Task Board

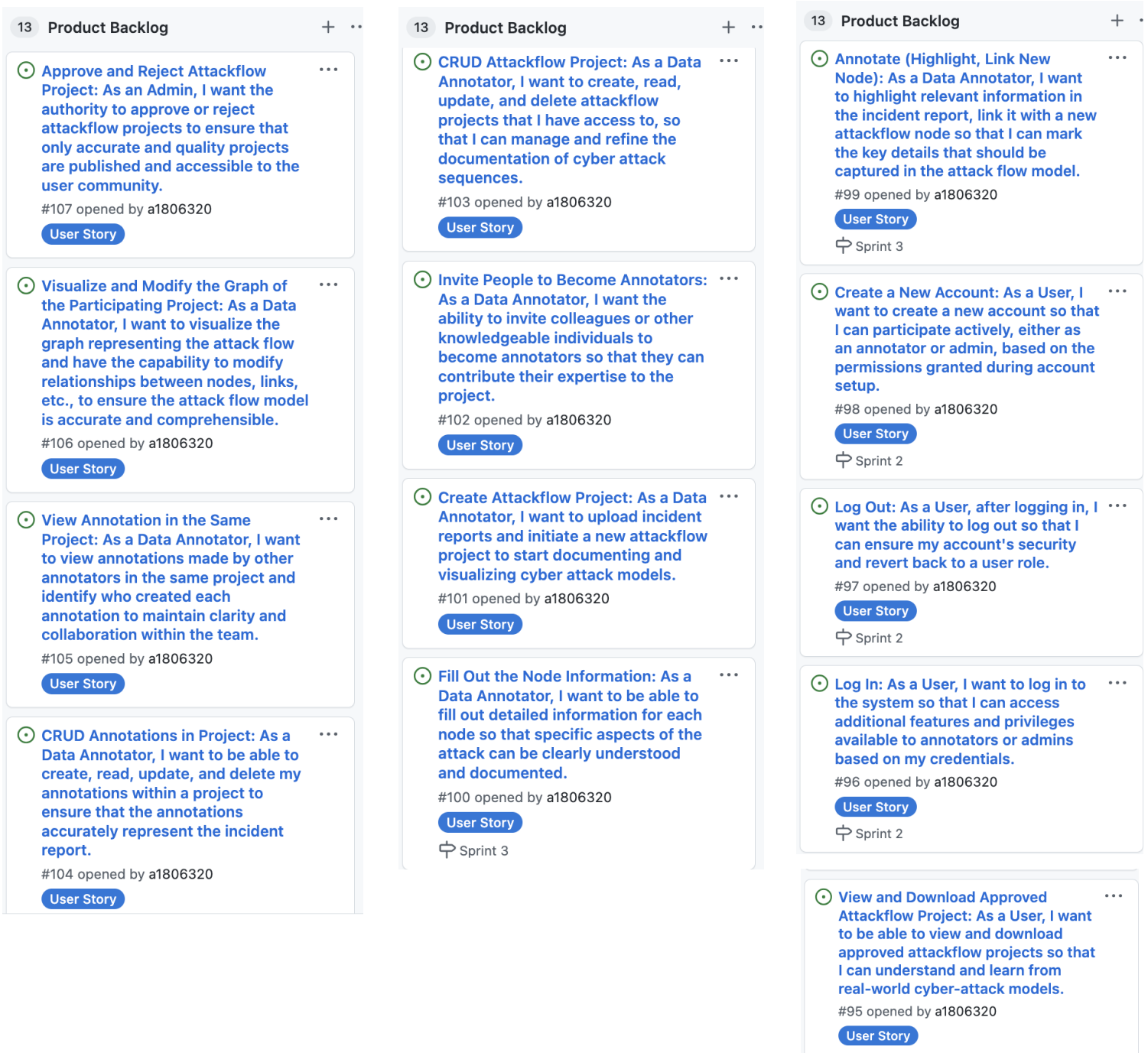


Image 1: Snapshot 3.1 Product Backlog

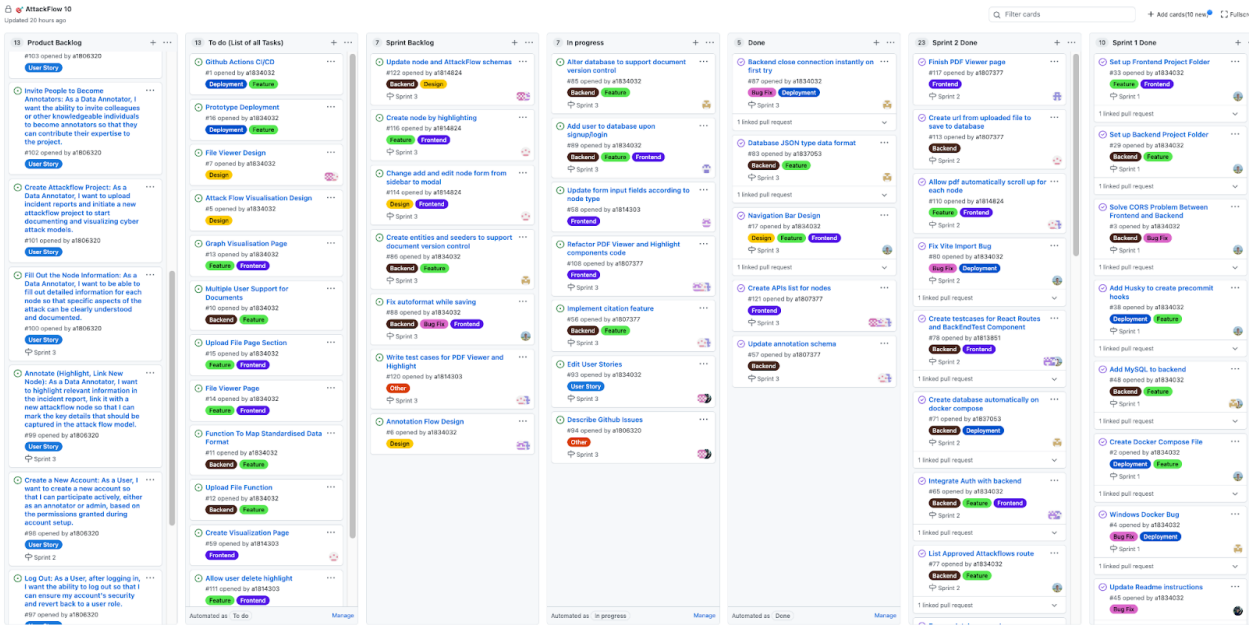


Image 2: Snapshot 3.1 Task Board

Sprint Backlog and User Stories

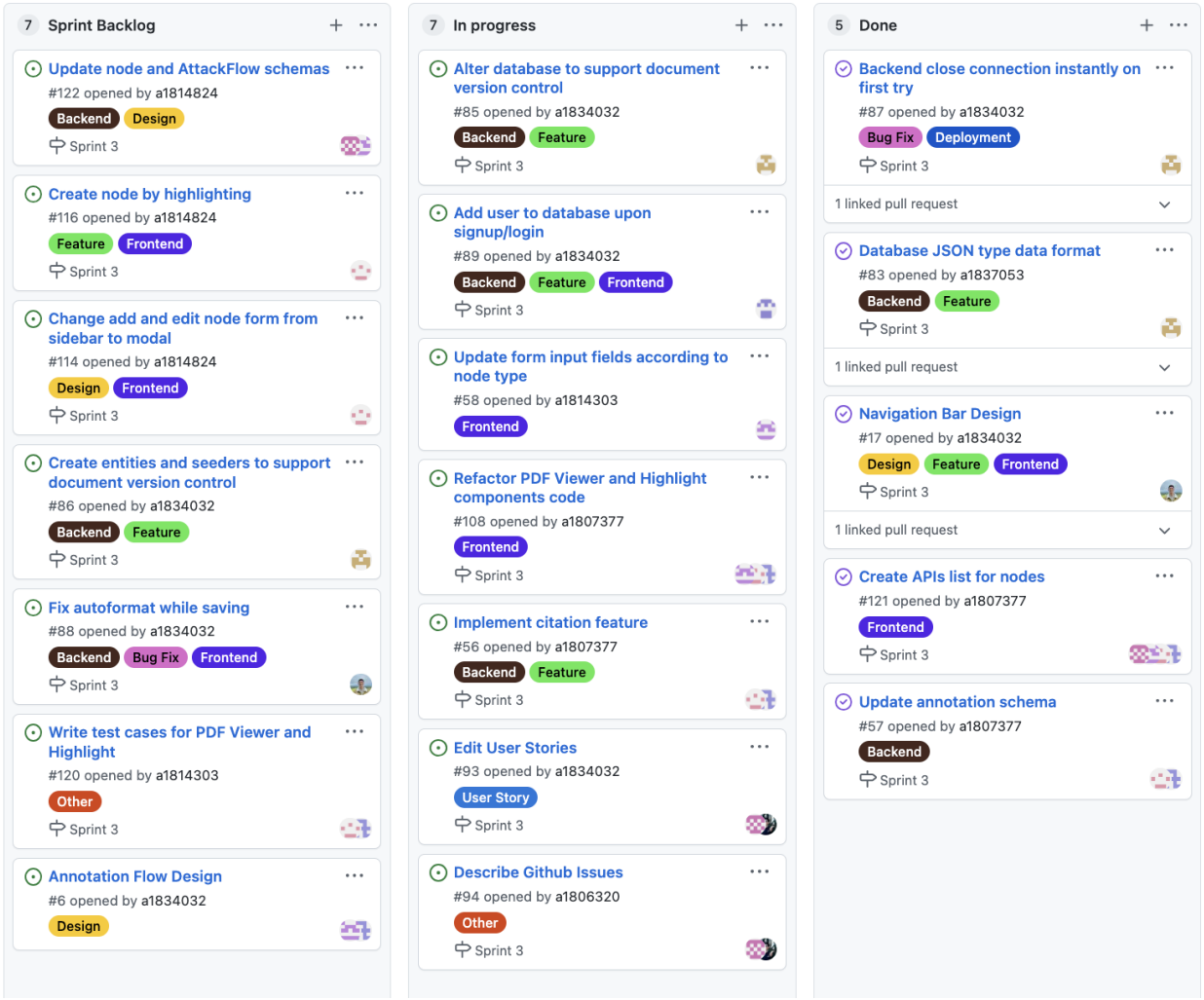


Image 3: Snapshot 3.1 Sprint Backlog

This screenshot depicts the three distinct stages of our sprint backlog: tasks queued in the backlog (Sprint Backlog), tasks currently underway (In progress), and tasks that have been successfully completed (Done).

User Stories of Current Sprint

1. **View and Download Approved Attackflow Project: As a User, I want to be able to view and download approved attackflow projects so that I can understand and learn from real-world cyber-attack models.**
    - Acceptance Criteria:
        - Given: I am a User visiting the platform and there are approved attackflow projects available.
        - When: I navigate to the list of approved attackflow projects.
        - Then: I should be able to view the details and download the project for my reference.

2. **Log In: As a User, I want to log in to the system so that I can access additional features and privileges available to annotators or admins based on my credentials.**
    - Acceptance Criteria:
        - Given: I am a User with valid credentials to the platform.
        - When: I input my username and password on the login page.
        - Then: I should be granted access and redirected to the dashboard or relevant page based on my role (annotator or admin).

3. **Log Out: As a User, after logging in, I want the ability to log out so that I can ensure my account's security and revert back to a user role.**
    - Acceptance Criteria:
        - Given: I am a User currently logged into the platform.
        - When: I click on the "log out" button or option.
        - Then: I should be logged out and returned to the platform's main or login page as a basic user without any specific privileges.

4. **Create a New Account: As a User, I want to create a new account so that I can participate actively, either as an annotator or admin, based on the permissions granted during account setup.**
    - Acceptance Criteria:
        - Given: I am a User on the platform's main or sign-up page.
        - When: I provide the required details to create a new account and submit the form.
        - Then: I should receive a confirmation message and, upon approval, gain the privileges of an annotator or admin based on the permissions granted during account setup.

5. **Annotate (Highlight, Link New Node): As a Data Annotator, I want to highlight relevant information in the incident report, link it with a new attackflow node so that I can mark the key details that should be captured in the attack flow model.**
    - Acceptance Criteria:

        - Given: I am a Data Annotator viewing an incident report in the system.
        - When: I highlight text and opt to link it to a new attackflow node.
        - Then: The highlighted text should be connected to a new node in the attack flow model.

6. **Fill Out the Node Information: As a Data Annotator, I want to be able to fill out detailed information for each node so that specific aspects of the attack can be clearly understood and documented.**
    - Acceptance Criteria:
        - Given: I am a Data Annotator and have created a new node in the attack flow model.
        - When: I fill out the detailed information fields for that node.
        - Then: The node should update to reflect the new details.

7. **Create Attackflow Project: As a Data Annotator, I want to upload incident reports and initiate a new attackflow project to start documenting and visualizing cyber attack models.**
    - Acceptance Criteria:
        - Given: I am a Data Annotator on the platform's project creation page.
        - When: I upload an incident report and initiate a new attackflow project.
        - Then: A new project should be created and I should be able to start adding nodes and annotations.

8. **Invite People to Become Annotators: As a Data Annotator, I want the ability to invite colleagues or other knowledgeable individuals to become annotators so that they can contribute their expertise to the project.**
    - Acceptance Criteria:
        - Given: I am a Data Annotator in an existing attackflow project.
        - When: I send an invitation through the system to potential new annotators.
        - Then: The invitees should receive an invitation and, upon acceptance, join the project as annotators.

9. **CRUD Attackflow Project: As a Data Annotator, I want to create, read, update, and delete attackflow projects that I have access to, so that I can manage and refine the documentation of cyber attack sequences.**
    - Acceptance Criteria:
        - Given: I am a Data Annotator in an existing attackflow project.
        - When: I perform create, read, update, or delete actions on the project.
        - Then: The project should reflect these changes accordingly.

10. **CRUD Annotations in Project: As a Data Annotator, I want to be able to create, read, update, and delete my annotations within a project to ensure that the annotations accurately represent the incident report.**

- Acceptance Criteria:
  - Given: I am a Data Annotator in a project with existing annotations.
  - When: I perform create, read, update, or delete actions on my annotations within the project.
  - Then: The annotations should be created, displayed, updated, or deleted as per my actions.

11. **View Annotation in the Same Project: As a Data Annotator, I want to view annotations made by other annotators in the same project and identify who created each annotation to maintain clarity and collaboration within the team.**
    - Acceptance Criteria:
      - Given: I am a Data Annotator in a project with annotations from multiple users.
      - When: I view the list of annotations.
      - Then: I should see who created each annotation for clarity and collaboration.

12. **Visualize and Modify the Graph of the Participating Project: As a Data Annotator, I want to visualize the graph representing the attack flow and have the capability to modify relationships between nodes, links, etc., to ensure the attack flow model is accurate and comprehensible.**
    - Acceptance Criteria:
      - Given: I am a Data Annotator viewing the attack flow model graph.
      - When: I choose to modify relationships between nodes, add links, etc.
      - Then: The graph should update to reflect these modifications, ensuring the attack flow model is accurate and comprehensive.

13. **Approve and Reject Attackflow Project: As an Admin, I want the authority to approve or reject attackflow projects to ensure that only accurate and quality projects are published and accessible to the user community.**
    - Acceptance Criteria:
      - Given: I am an Admin and reviewing a list of submitted attackflow projects awaiting approval.
      - When: I select a project and choose to either approve or reject it.
      - Then: The project's status should update accordingly. If approved, the project should be accessible to the user community, and if rejected, it should not be published or visible to users.

For detailed insights into the Task Board, Product Backlog, and Sprint Backlog, please click here.

Definition of Done

For our fourth snapshot, our "definition of done" remains divided into two sections: general goals and specific goals. While these largely build upon the foundations set in our first snapshot, they have been expanded to incorporate fresh objectives and other pivotal functionalities that emerged from our recent discussions and user stories.

General Goals:

- **Testing**: All code has undergone rigorous testing and passes all unit tests.
- **Deployment**: The code is successfully deployed to a staging environment, accessible via a public URL.
- **Code Review**: The code has been reviewed, critiqued, and approved by at least one other developer.
- **Feedback Integration**: All feedback from reviewers has been addressed and incorporated.
- **User Experience**: The website provides a seamless experience on both mobile and web platforms, ensuring easy navigation for users.
- **Documentation:** Maintain documentation of project requirements, changes, and decisions. Share this documentation with the client to ensure that both parties have a clear understanding of project scope and goals.

Specific Goals:

- For **Users**:
  - **User Registration**: A potential user should effortlessly register using a unique email address and receive an acknowledgment after successful registration.
  - **User Authentication**: Users must securely log in using their registered credentials and should receive apt feedback for unsuccessful login attempts.
  - **User Logout**: Users should find the logout process straightforward, ensuring their session ends and their data remains secure.
  - **View and Download Approved Attackflow Projects**: Users should easily access, view, and download approved attack flow projects for understanding and reference.

- For **Data Annotators**:
  - **File Upload and Annotation**: Data annotators must be able to upload documents and annotate specific segments within these files without ambiguity.
  - **Attack Flow Integration**: The system should convert annotations from uploaded documents into attack flow models compliant with the MITRE framework.

        - **Visualization**: Data annotators should have tools to visually represent any attack flow, ensuring a coherent understanding of sequences and consequences.
        - **Validation and Collaboration**: Data annotators should have the capability to collaborate on annotations and employ a version control mechanism to monitor modifications to incident reports.
        - **Invite Colleagues**: Data annotators should be able to invite their colleagues to contribute to the platform.

    - For **Admins**:
        - **Project Approval/Rejection**: Admins should have the authority to approve or reject attack flow projects, ensuring only quality projects are available for the user community.

Summary of Changes

After a productive week, our team achieved some significant milestones in our development process. First of all, we successfully integrated Auth0 into our backend, enhancing the security and the reliability of our application. To ensure the robustness of our implementation, we created a few comprehensive test cases for Auth0. In addition, we fixed a critical Vite import bug to streamline our development workflow. Previously, the bug would require us to manually perform two additional dependencies installation tasks, which took up a lot of time and negatively impacted our development efficiency. In response to the data storage requirements, we optimized some data types of our database to better handle 'array'-like data structures so that our program is now able to store array data. What is more, we also improved our backend's performance by implementing a timeout mechanism and a dependency resolver, which allows for an instant connection closure on the first attempt.

Attack Flow

Snapshot Week 8 of Group AttackFlow 10

Jie Shen Beh (a1834032)
Jian Zhe Chan (a1813851)
Gia Bao Hoang (a1814824)
Marcus Hoang (a1814303)
Guan Chern Liew (a1837053)
Vinh Diem Nguyen (a1838114)
Hoang Nam Trinh (a1807377)
Hung Yee Wong (a1815836)
Jiajun Yu (a1806320)

Product Backlog and Task Board

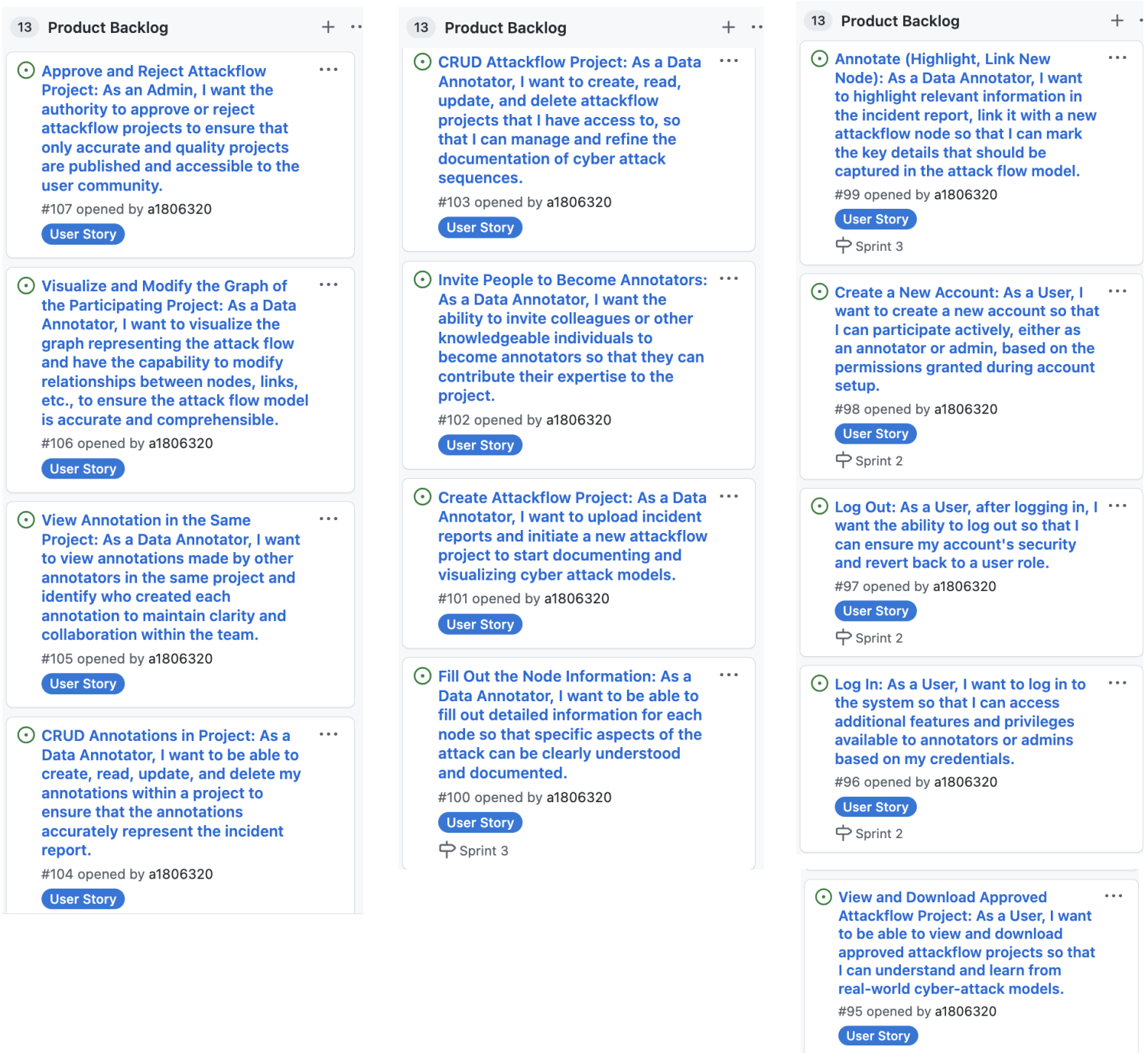


Image 1: Snapshot 3.2 Product Backlog

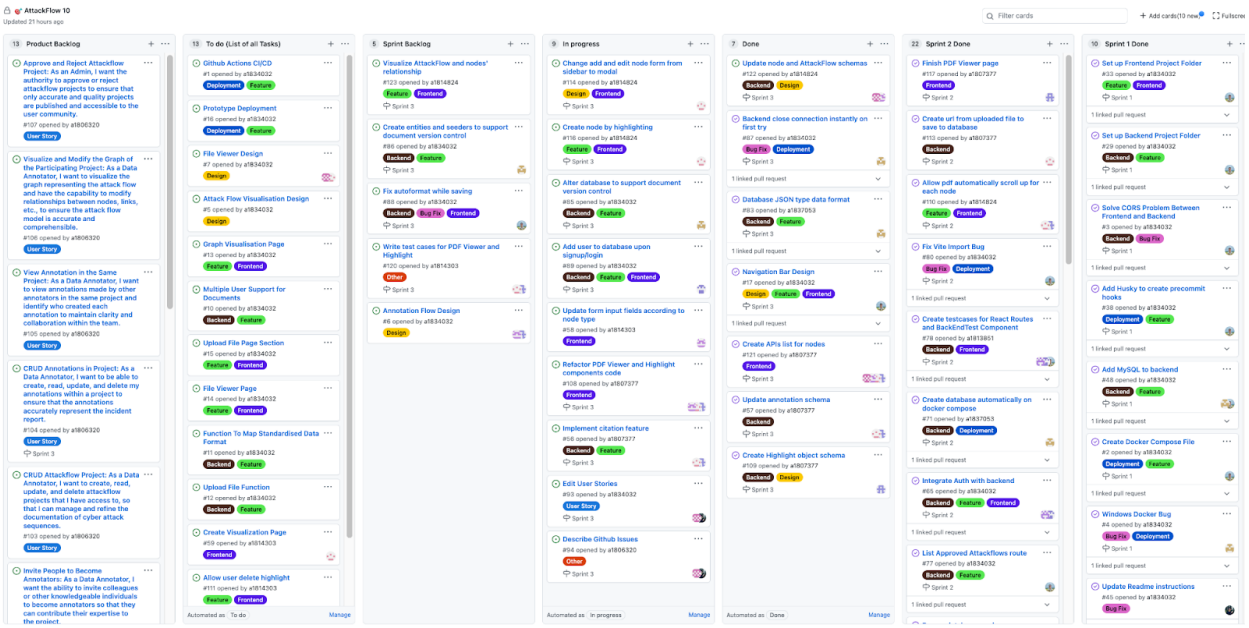


Image 2: Snapshot 3.2 Task Board

Sprint Backlog and User Stories

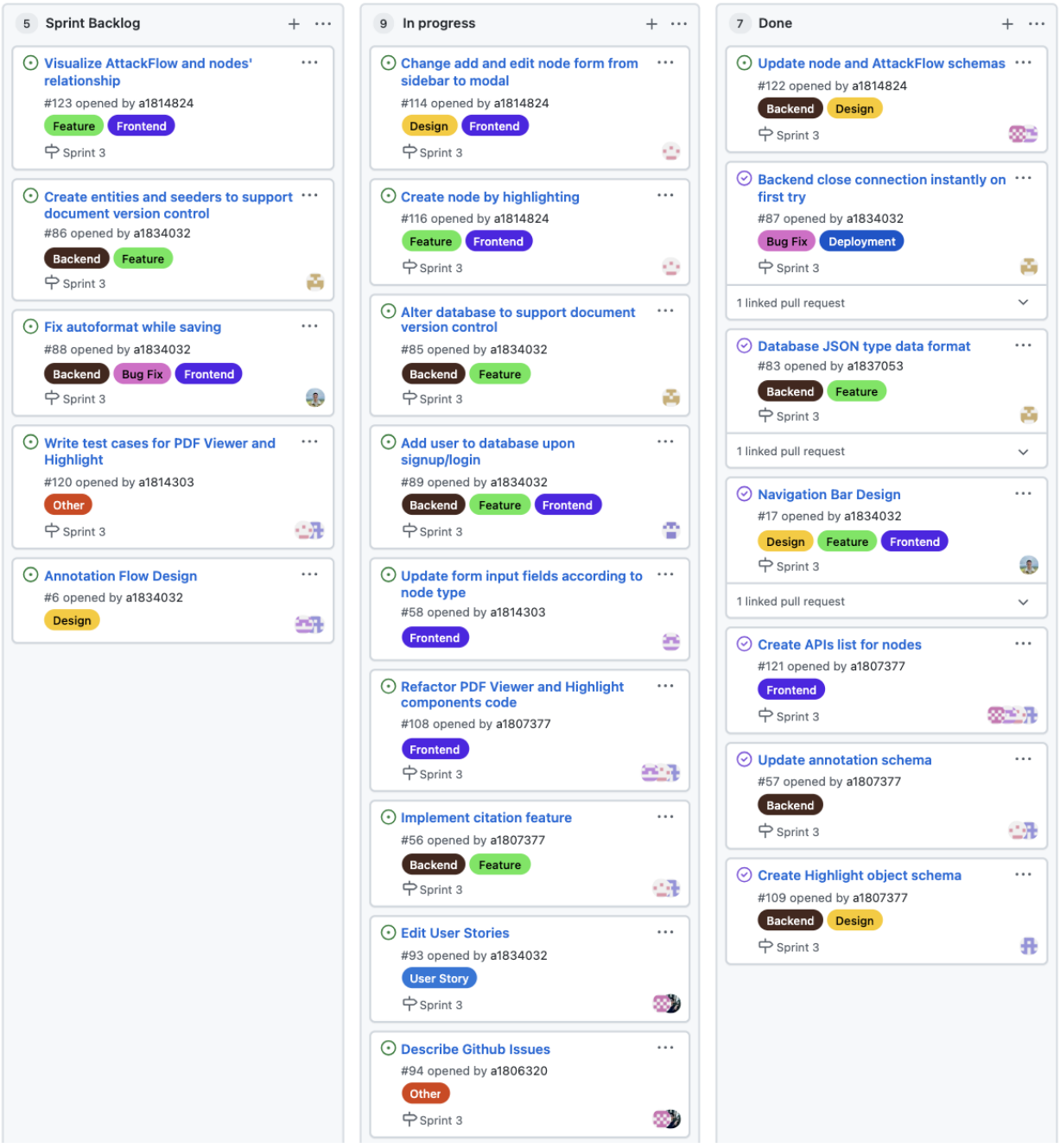


Image 3: Snapshot 3.2 Sprint Backlog

This screenshot depicts the three distinct stages of our sprint backlog: tasks queued in the backlog (Sprint Backlog), tasks currently underway (In progress), and tasks that have been successfully completed (Done).

User Stories of Current Sprint

1. View and Download Approved Attackflow Project: As a User, I want to be able to view and download approved attackflow projects so that I can understand and learn from real-world cyber-attack models.
    - Acceptance Criteria:
        - Given: I am a User visiting the platform and there are approved attackflow projects available.
        - When: I navigate to the list of approved attackflow projects.
        - Then: I should be able to view the details and download the project for my reference.

2. Log In: As a User, I want to log in to the system so that I can access additional features and privileges available to annotators or admins based on my credentials.
    - Acceptance Criteria:
        - Given: I am a User with valid credentials to the platform.
        - When: I input my username and password on the login page.
        - Then: I should be granted access and redirected to the dashboard or relevant page based on my role (annotator or admin).

3. Log Out: As a User, after logging in, I want the ability to log out so that I can ensure my account's security and revert back to a user role.
    - Acceptance Criteria:
        - Given: I am a User currently logged into the platform.
        - When: I click on the "log out" button or option.
        - Then: I should be logged out and returned to the platform's main or login page as a basic user without any specific privileges.

4. Create a New Account: As a User, I want to create a new account so that I can participate actively, either as an annotator or admin, based on the permissions granted during account setup.
    - Acceptance Criteria:
        - Given: I am a User on the platform's main or sign-up page.
        - When: I provide the required details to create a new account and submit the form.
        - Then: I should receive a confirmation message and, upon approval, gain the privileges of an annotator or admin based on the permissions granted during account setup.

5. Annotate (Highlight, Link New Node): As a Data Annotator, I want to highlight relevant information in the incident report, link it with a new attackflow node so that I can mark the key details that should be captured in the attack flow model.
    - Acceptance Criteria:
        - Given: I am a Data Annotator viewing an incident report in the system.

        - When: I highlight text and opt to link it to a new attackflow node.
        - Then: The highlighted text should be connected to a new node in the attack flow model.

6. Fill Out the Node Information: As a Data Annotator, I want to be able to fill out detailed information for each node so that specific aspects of the attack can be clearly understood and documented.
    - Acceptance Criteria:
        - Given: I am a Data Annotator and have created a new node in the attack flow model.
        - When: I fill out the detailed information fields for that node.
        - Then: The node should update to reflect the new details.

7. Create Attackflow Project: As a Data Annotator, I want to upload incident reports and initiate a new attackflow project to start documenting and visualizing cyber attack models.
    - Acceptance Criteria:
        - Given: I am a Data Annotator on the platform's project creation page.
        - When: I upload an incident report and initiate a new attackflow project.
        - Then: A new project should be created and I should be able to start adding nodes and annotations.

8. Invite People to Become Annotators: As a Data Annotator, I want the ability to invite colleagues or other knowledgeable individuals to become annotators so that they can contribute their expertise to the project.
    - Acceptance Criteria:
        - Given: I am a Data Annotator in an existing attackflow project.
        - When: I send an invitation through the system to potential new annotators.
        - Then: The invitees should receive an invitation and, upon acceptance, join the project as annotators.

9. CRUD Attackflow Project: As a Data Annotator, I want to create, read, update, and delete attackflow projects that I have access to, so that I can manage and refine the documentation of cyber attack sequences.
    - Acceptance Criteria:
        - Given: I am a Data Annotator in an existing attackflow project.
        - When: I perform create, read, update, or delete actions on the project.
        - Then: The project should reflect these changes accordingly.

10. CRUD Annotations in Project: As a Data Annotator, I want to be able to create, read, update, and delete my annotations within a project to ensure that the annotations accurately represent the incident report.
    - Acceptance Criteria:

        - Given: I am a Data Annotator in a project with existing annotations.
        - When: I perform create, read, update, or delete actions on my annotations within the project.
        - Then: The annotations should be created, displayed, updated, or deleted as per my actions.

11. View Annotation in the Same Project: As a Data Annotator, I want to view annotations made by other annotators in the same project and identify who created each annotation to maintain clarity and collaboration within the team.
    - Acceptance Criteria:
        - Given: I am a Data Annotator in a project with annotations from multiple users.
        - When: I view the list of annotations.
        - Then: I should see who created each annotation for clarity and collaboration.

12. Visualize and Modify the Graph of the Participating Project: As a Data Annotator, I want to visualize the graph representing the attack flow and have the capability to modify relationships between nodes, links, etc., to ensure the attack flow model is accurate and comprehensible.
    - Acceptance Criteria:
        - Given: I am a Data Annotator viewing the attack flow model graph.
        - When: I choose to modify relationships between nodes, add links, etc.
        - Then: The graph should update to reflect these modifications, ensuring the attack flow model is accurate and comprehensive.

13. Approve and Reject Attackflow Project: As an Admin, I want the authority to approve or reject attackflow projects to ensure that only accurate and quality projects are published and accessible to the user community.
    - Acceptance Criteria:
        - Given: I am an Admin and reviewing a list of submitted attackflow projects awaiting approval.
        - When: I select a project and choose to either approve or reject it.
        - Then: The project's status should update accordingly. If approved, the project should be accessible to the user community, and if rejected, it should not be published or visible to users.

For detailed insights into the Task Board, Product Backlog, and Sprint Backlog, please click here.

Definition of Done

For our fifth snapshot, our "definition of done" remains divided into two sections: general goals and specific goals. While these largely build upon the foundations set in our first snapshot, they have been expanded to incorporate fresh objectives and other pivotal functionalities that emerged from our recent discussions and user stories.

General Goals:

- **Testing**: All code has undergone rigorous testing and passes all unit tests.
- **Deployment**: The code is successfully deployed to a staging environment, accessible via a public URL.
- **Code Review**: The code has been reviewed, critiqued, and approved by at least one other developer.
- **Feedback Integration**: All feedback from reviewers has been addressed and incorporated.
- **User Experience**: The website provides a seamless experience on both mobile and web platforms, ensuring easy navigation for users.
- **Documentation:** Maintain documentation of project requirements, changes, and decisions. Share this documentation with the client to ensure that both parties have a clear understanding of project scope and goals.

Specific Goals:

- For **Users**:
  - **User Registration**: A potential user should effortlessly register using a unique email address and receive an acknowledgment after successful registration.
  - **User Authentication**: Users must securely log in using their registered credentials and should receive apt feedback for unsuccessful login attempts.
  - **User Logout**: Users should find the logout process straightforward, ensuring their session ends and their data remains secure.
  - **View and Download Approved Attackflow Projects**: Users should easily access, view, and download approved attack flow projects for understanding and reference.

- For **Data Annotators**:
  - **File Upload and Annotation**: Data annotators must be able to upload documents and annotate specific segments within these files without ambiguity.
  - **Attack Flow Integration**: The system should convert annotations from uploaded documents into attack flow models compliant with the MITRE framework.

        - **Visualization**: Data annotators should have tools to visually represent any attack flow, ensuring a coherent understanding of sequences and consequences.
        - **Validation and Collaboration**: Data annotators should have the capability to collaborate on annotations and employ a version control mechanism to monitor modifications to incident reports.
        - **Invite Colleagues**: Data annotators should be able to invite their colleagues to contribute to the platform.

    - For **Admins**:
        - **Project Approval/Rejection**: Admins should have the authority to approve or reject attack flow projects, ensuring only quality projects are available for the user community.

Summary of Changes

Throughout a fruitful week, our team has made some great progress on the annotation functionality, particularly putting the efforts on the highlighting feature, while simultaneously initiating the visualisation aspect. However, due to the unexpected GitHub licence issue, we delayed a bit on our development process, and had not merged our code to the main branch. Here's the breakdown of our current progress:

- Annotation Advancements: We have been working on the annotation's implementation for most of our time, and moving closer to a fully functional feature. Part of this progress involved integrating a highlight library to lay the groundwork for an efficient annotation system.

- Schema Refinements: There has been a critical update to both the nodes and AttackFlow schemas. The refinements of the schema ensures that our foundational structures align with the expected functionality of the software.

- API Development: We have wrapped up a complete API list, which is crucial for facilitating interactions with nodes. These APIs will serve as the bridge for multiple functionalities, ensuring smooth and efficient operations.

- User Interface Enhancements: To enhance the user experience, our team has also changed the homepage design of our application to make it more visual-appealing.

- Initiation of Visualisation: We have also started the visualisation functionality implementation. The primary focus here is on effectively displaying the AttackFlow visualisation result. The major challenge of this task is illustrating the intricate

relationships between nodes, making the data presentation more comprehensible and interactive for users.

In conclusion, these advancements mark significant strides towards realising our project goals step-by-step, ensuring that the software is not only functional but also user-friendly and visually appealing.