Attack Flow

Retrospective Sprint 4 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)

What went well in the sprint?

In reflecting on sprint 4, our team's productivity and alignment were evident throughout weeks 9 and 10. Our decision to revisit MITRE's sample attackflows in Week 9 underscored our commitment to build on established benchmarks. This wasn't just a technical exercise; it exemplified our dedication to understanding foundational principles before software execution. The development of the graph visualization page, rooted in previous research, showcased our team's forward-thinking approach. This blend of research and implementation helped us navigate the challenges more efficiently.

Further, our enhancements in admin functions and backend routes aligned seamlessly with our goals of functionality and user experience. It was heartening to see features like annotation and collaboration not just implemented but also refined to perfection. The culmination of the annotation feature—with functionalities like file reading and node relationship establishment—was a testament to our team's technical prowess.

Our responsiveness to challenges was another highlight. By Week 10, we had not only fixed notable bugs like the schema foreign key issue but also furthered our software's features. As end-of-semester pressures loomed, our strategic focus on refining existing features, instead of just adding new ones, was a smart move, reinforcing our commitment to quality over quantity.

What could be improved?

Looking back on sprint 4, several areas stand out where improvement is paramount. Foremost is our alignment with client expectations. The feedback highlighted that while we made significant advancements like the intricate graph visualization and annotation enhancements, overlooking the timely completion of a primary feature they anticipated created ripples in our project trajectory. This omission not only shifted our internal targets but also cast a shadow on our team's spirit and pace.

From a technical perspective, our focus on graph visualization and backend development showed promise, but balancing this with client priorities is crucial. For instance, our strides in creating the annotation side menu or fixing schema foreign key bugs, though noteworthy, should parallelly align with client-preferred functionalities to prevent any disparity.

Furthermore, the pressing academic demands on several team members posed challenges, with many of us finding it hard to maintain project commitments. This delicate balance influenced our software development pace, as evidenced by delays in functionalities like incident report annotation. Going forward, enhancing communication, setting clear priorities, and managing time better will be pivotal. This would ensure that both our technical advancements and process efficiency harmonize with client and team aspirations.

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

In our next sprint, our primary focus will be on the efficient rollout of the annotation tool. Recognizing its importance and addressing past delays, we aim to ensure that its implementation aligns more closely with both user and client expectations. To streamline our development process, we're introducing structured daily stand-ups. These sessions will serve as regular touchpoints to keep the team aligned, monitor progress, and swiftly tackle any challenges. This step is expected to enhance workflow efficiency and maintain clear communication amongst all members.

Furthermore, we're establishing weekly feedback sessions. This proactive measure is aimed at strengthening team unity and fostering an environment where members feel valued, heard, and involved. These sessions will serve as platforms for team members to voice concerns, address potential roadblocks, and collaboratively brainstorm solutions. With the project presentation nearing and time running out, our emphasis will shift towards refining and finalizing our existing features to ensure a compelling showcase for the client. Our overarching goal, through these targeted improvements, is not only to enhance the technical facets of our software but also to refine our software development approach, ensuring it remains agile, client-centric, and goal-oriented.

Comment on your progress this sprint

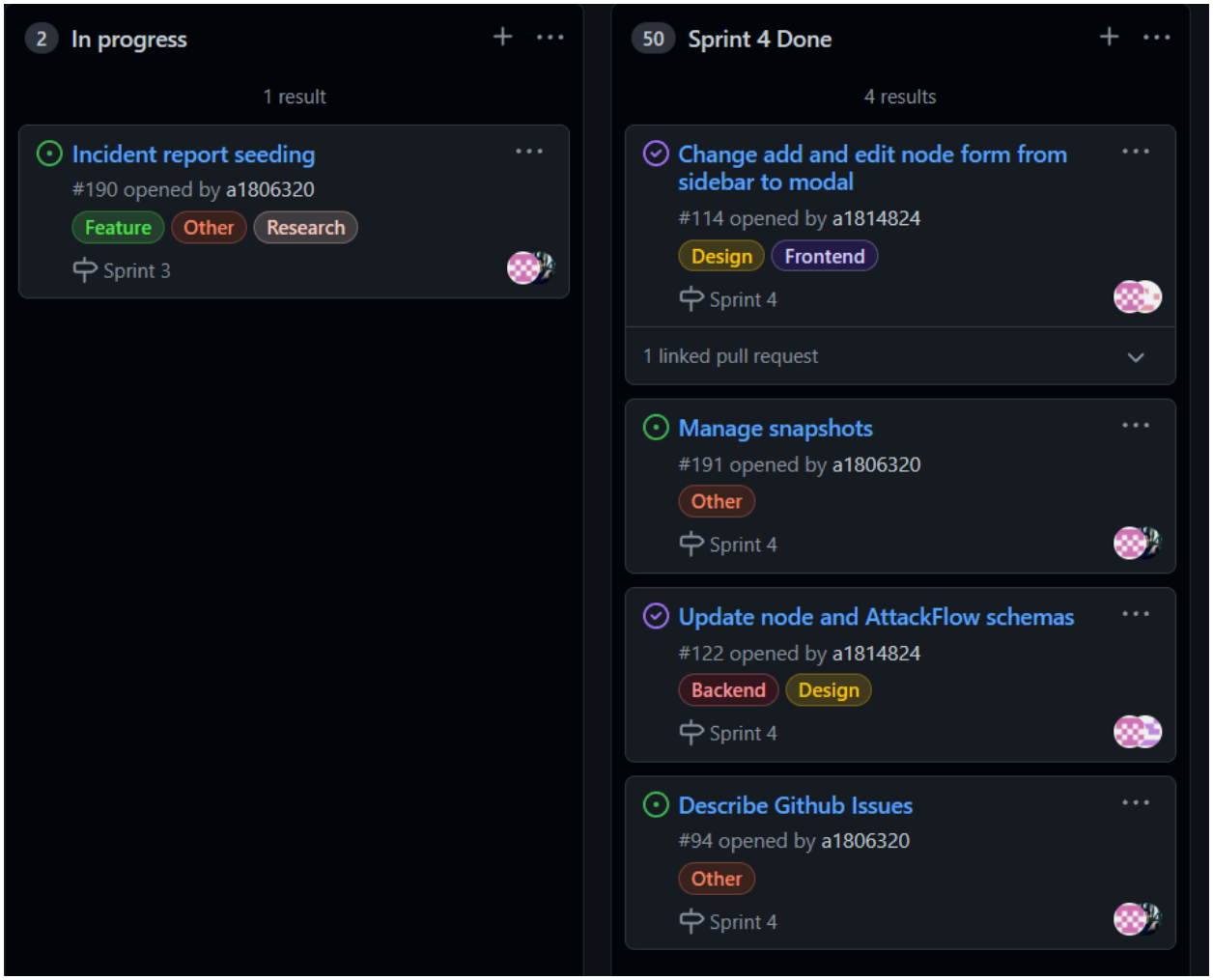

Image 1: Issues for Sprint 4

Sprint 4 was a culmination of relentless dedication and technical intricacy. Among the pivotal tasks I undertook, the continuation of incident report seeding was foremost. This wasn't just about feeding data; it involved a deep understanding of the software architecture and ensuring that our application could reliably and efficiently process the input for testing. The evolution of this feature demonstrates our commitment to laying a robust foundation for our product.

In collaboration with a teammate, we took on the challenge of reimagining the edit note functionality. Transitioning it from a sidebar to a modal required not only technical expertise but also an eye for user experience. This move enhanced the application's intuitiveness, making user interactions more streamlined and purposeful.

Managing the snapshots for sprint 4 demanded a blend of organizational skill and project foresight. This responsibility emphasized my commitment to ensuring our team remained on track and that our sprint objectives were clear and achievable.

The tasks involving schema updates, particularly for nodes and AttackFlow, were especially intricate. Partnering with a teammate, we delved into the underpinnings of our system, making necessary adjustments to align with the evolving software requirements. This effort was testament to our dedication to building a scalable and efficient system.

Lastly, clarifying and providing context to several GitHub issues was a crucial step in ensuring our team's alignment. By detailing out issues and offering a clearer roadmap, we ensured that every team member was equipped with the knowledge to progress effectively.

In summary, Sprint 4 was an exemplar of technical depth and cohesive teamwork. Through meticulous task management and collaboration, we not only addressed the sprint's challenges but also set the stage for continued success.

Snapshots

I attended the sprint review/planning meeting on 22/08/2023 with the tutor.
I attended the sprint review/planning meeting on 05/09/2023 with the tutor.
I attended the sprint review/planning meeting on 17/10/2023 with the tutor.

Attack Flow

Snapshot Week 9 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

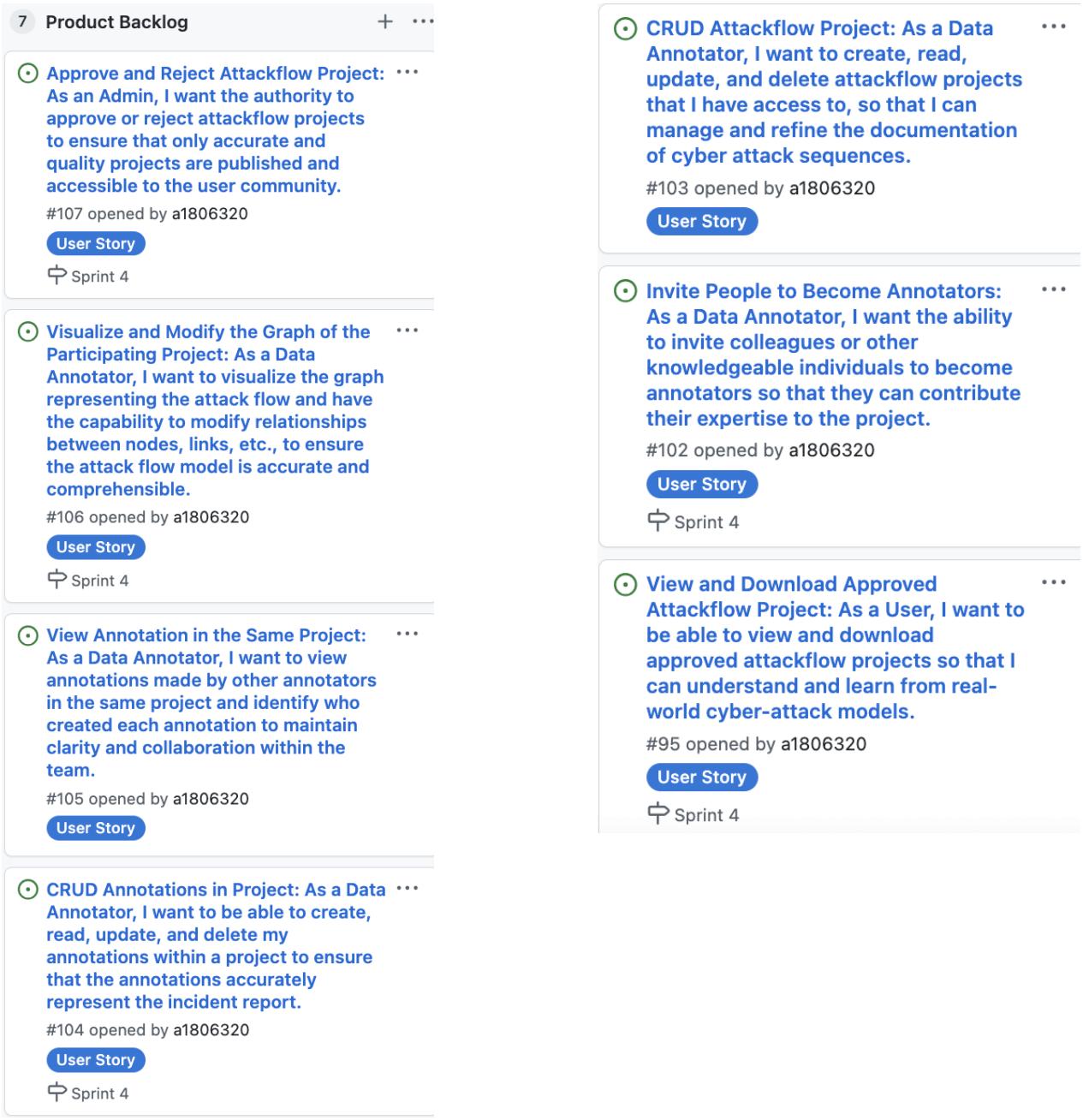

Image 1: Snapshot 4.1 Product Backlog

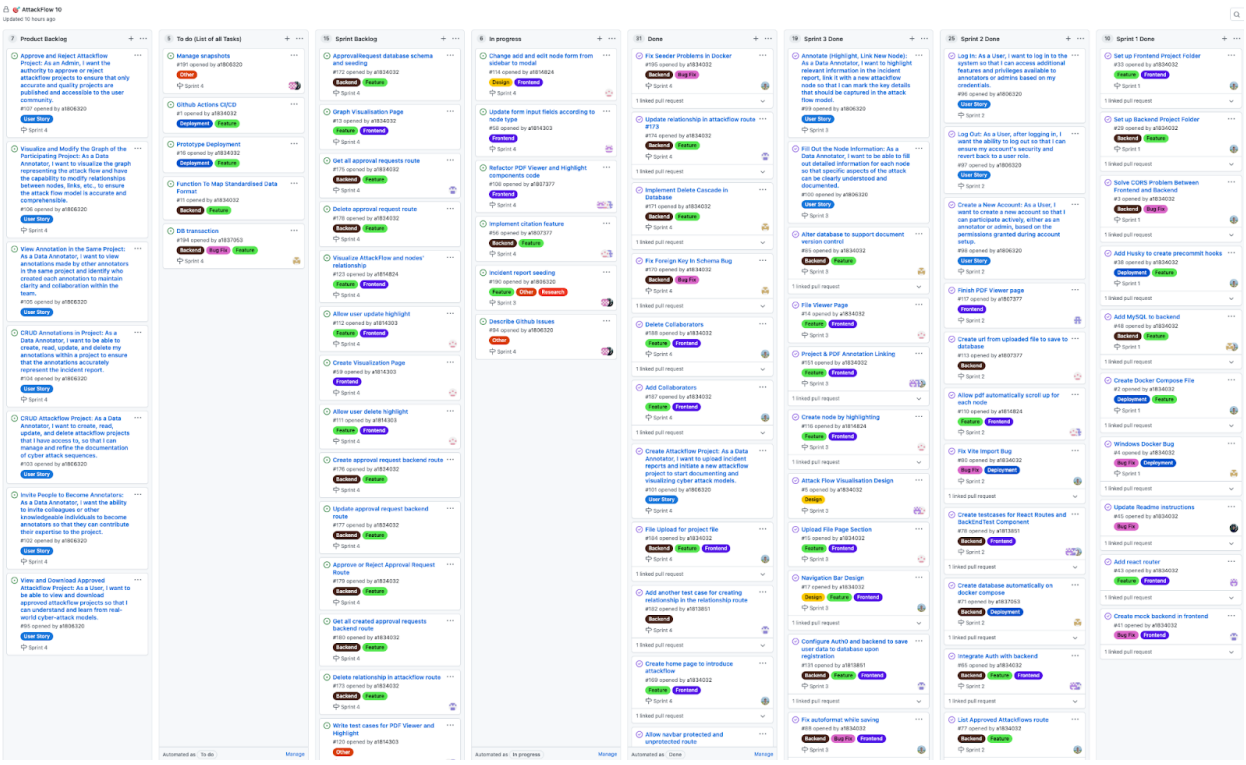

Image 2: Snapshot 4.1 Task Board

Sprint Backlog and User Stories

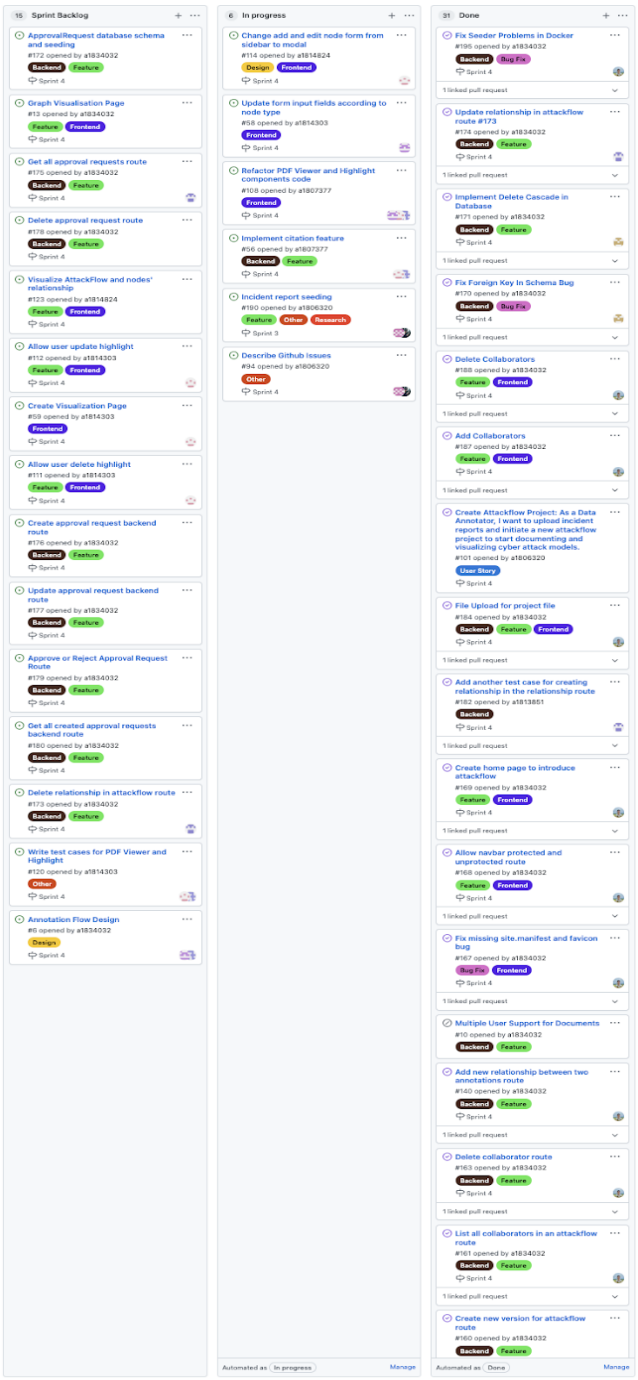

Image 3: Snapshot 4.1 Sprint Backlog

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 ninth 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

In this highly productive week, our team was continually making steady progress towards our final goal of developing this attackflow project. Here is the list of our achievements this week:

Incident Report Seeding: This week, we came back to our starting point and studied a few MITRE's sample attackflows reports. The objective was to comprehend and subsequently replicate the visualising results to ensure that in the end, our software can generate visualised outcomes which will align with established benchmarks.

Graph Visualisation Page: We embarked on the development of the graph visualisation page. This task relies on the research conducted in prior weeks and is driven by the chosen library's capabilities.

Admin Functions Enhancement: We have also finished the administration functionalities. With this, admins can now approve projects and make them accessible for broad viewing.

Annotation and Highlight Finalisation: We have also successfully completed the development of the annotation function. This module now includes features like file reading, highlighting, feeding node information via a side menu, and node relationship establishment.

Backend Routes Update: To support the evolving functionalities of our software, we updated the backend routes. New routes catering to functionalities like version control and collaboration were developed. In addition, we have also maintained some existing annotation routes to enhance the robustness of our project.

In summary, our team has made significant achievements in Week 9, especially in functional requirements. Through these developments, we are confident to ensure that our software not only meets but also surpasses the expectations of this course.

Attack Flow

Snapshot Week 10 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

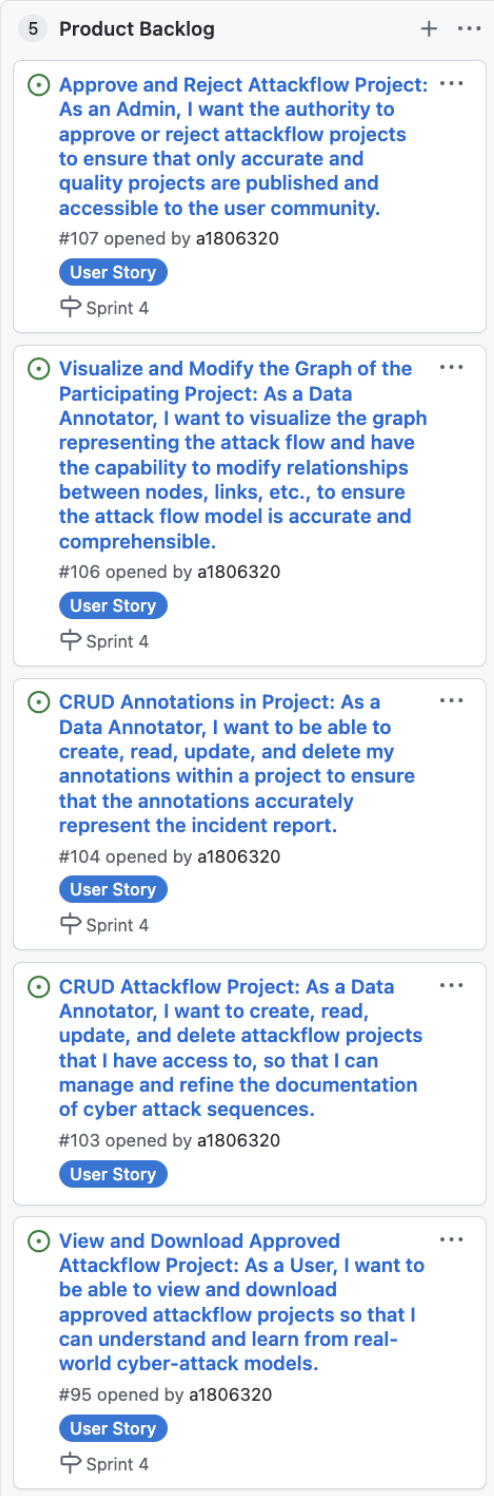

Image 1: Snapshot 4.2 Product Backlog

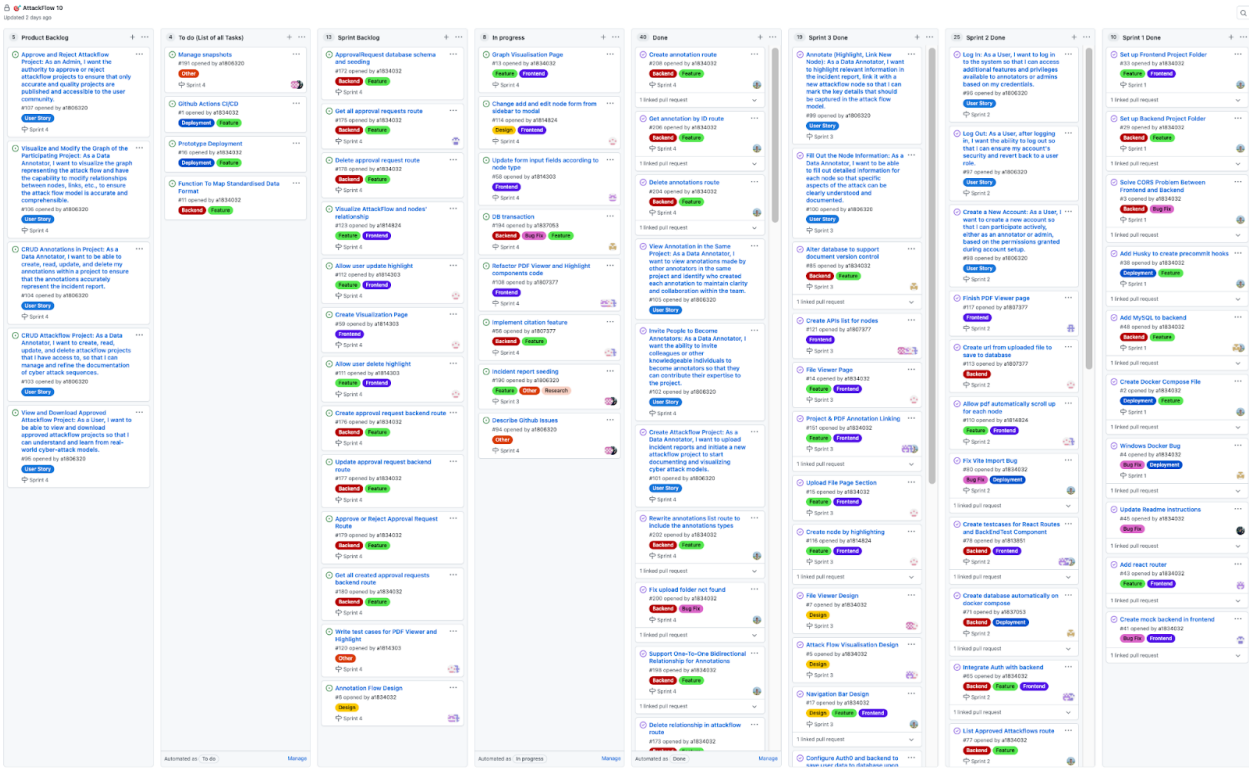

Image 2: Snapshot 4.2 Task Board

Sprint Backlog and User Stories

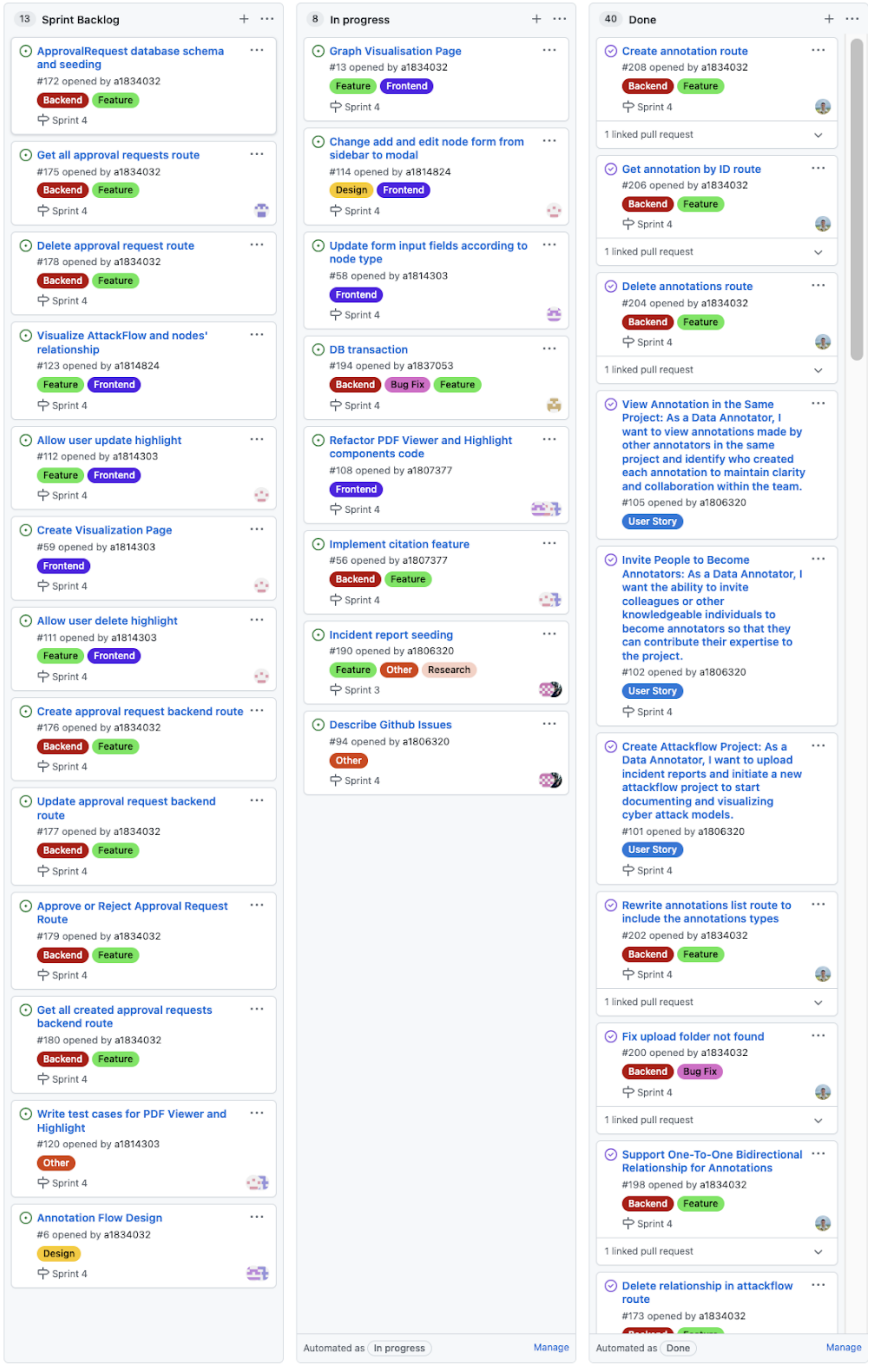

Image 3: Snapshot 4.2 Sprint Backlog

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. Visualise and Modify the Graph of the Participating Project: As a Data Annotator, I want to visualise 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 tenth 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.

        - **Visualisation**: 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

This week marked another progressive phase in our attackflow project journey. As we near the end of the semester, our primary focus shifted to refining and finalising key features, while also introducing new ones that would enhance the usability and functionality of our software. Here's a brief breakdown:

Graph Visualisation: Our team successfully completed the front-end aspect of the graph visualisation, presenting a basic functional graph. With this in place, our energies are now channelled towards developing a robust backend. A significant portion of this phase is to form an intricate schema for nodes within the graph, ensuring they reflect accurate data relationships.

Invitation and Collaboration: The feature enabling users to invite others and collaborate in real-time has reached its final stages. We have wrapped up the primary development and have now shifted our focus to improving the user experience to ensure smooth interactions and easy-to-use interfaces.

Annotation Side Menu: To enhance the clarity of annotations, we have introduced a dropdown list in the side menu. This new addition feature allows users to effortlessly select from different types of nodes, which streamlines the annotation process.

Incident Report Annotation: Our commitment to working on the incident report annotation continues, with dedicated efforts to ensure it aligns with our vision of offering comprehensive and accurate visualisation results for users.

Bug Fixes: To enhance the technical stability of our software, several bugs were identified and addressed. One notable fix was the resolution of an issue about the schema foreign keys, bolstering the reliability of our database relationships.

In essence, this week has been pivotal in shaping our software. With features like graph visualisation moving to advanced stages and collaboration tools being refined, we are approaching closer to a final product that encapsulates our vision and meets user needs.