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)

What went well in the sprint?

This sprint witnessed our team effectively marrying the Scrum process with key technical advancements, resulting in a highly productive phase. One poignant example was our strategic decision to integrate Auth0. Beyond the technical enhancement, this move epitomized our adherence to the Scrum tenet of iterative feedback, reflecting our commitment to addressing anticipated user security concerns. Addressing the Vite import issue demonstrated our Scrum agility. Instead of passively grappling with the bug, we proactively streamlined our workflow, embodying the Scrum value of flexibility and adaptability. This not only optimized our development pace but also harmonized our team's collaboration. Moreover, our emphasis on quality assurance, showcased by crafting rigorous test cases for Auth0, mirrors Scrum's focus on delivering value.

Additionally, our approach to enhancing the user experience, such as the homepage redesign, was in direct response to user stories, a foundational Scrum artifact. Our progress in annotation and visualization resonates with the Scrum practice of incremental delivery, ensuring each feature is user-ready. Collectively, our achievements underscore a strong alignment with the Scrum framework, enabling us to deliver a user-centric, efficient software product.

What could be improved?

During this sprint, while we encountered progress in several areas, certain challenges did come to the forefront. A primary setback was the unforeseen GitHub license issue. The delay it caused in merging to the main branch brought to light the gaps in our contingency planning within the Scrum process. As Scrum encourages foresight and adaptability, we recognized the need to diversify our tool dependencies and potentially embrace pair programming or face-to-face coding sessions, especially when facing such technical issues.

Another observation was the partial completion of the annotation functionality. In the Scrum framework, sprint goals aim to be achievable; thus, better task breakdown or improved backlog prioritization could possibly have expedited this feature’s rollout. The impending semester end, with its academic demands, underscored the essence of continuous sprint reviews and the importance of granular task allocation for balanced team workload.

Furthermore, though we made advancements in the user interface, the Scrum principle of iterative development suggests we could gain more by embedding continuous user feedback mechanisms. As we reflect, the lessons are clear: enhancing our feedback loops, bolstering contingency measures, and ensuring every sprint goal is distinctly achievable will be instrumental for our team's future success.

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

As we pivot into our next sprint, our team has mapped out targeted commitments that aim to fortify our development approach. Firstly, task allocation will undergo a structured revision. We plan to create a comprehensive skills matrix that transparently identifies each member's expertise, ensuring tasks are assigned where proficiency is highest. This method, rooted in Scrum's ethos of leveraging team strengths, aims to accelerate task completion.

Addressing challenges from the previous sprint, we'll establish "Focused Integration Sessions." These will allow members to collaboratively address technical bottlenecks like the GitHub issue, using pair programming or live code reviews. With the semester's peak approaching, we'll initiate "Progress Pulse Checks"—brief yet focused meetings twice a week—to monitor task progress, identify barriers, and adjust workloads.

A pivotal commitment is the early-stage roll-out of the visualization functionality. By front-loading its development, we pave the way for more in-depth user testing and refinements in subsequent sprints. Using iterative design principles, we'll incorporate regular feedback loops, conducting prototype testing with users to ensure their needs are front and center in our design decisions. Collectively, these improvements are engineered to propel our software development process forward, emphasizing agility, user feedback, and adaptive planning.

Comment on your progress this sprint

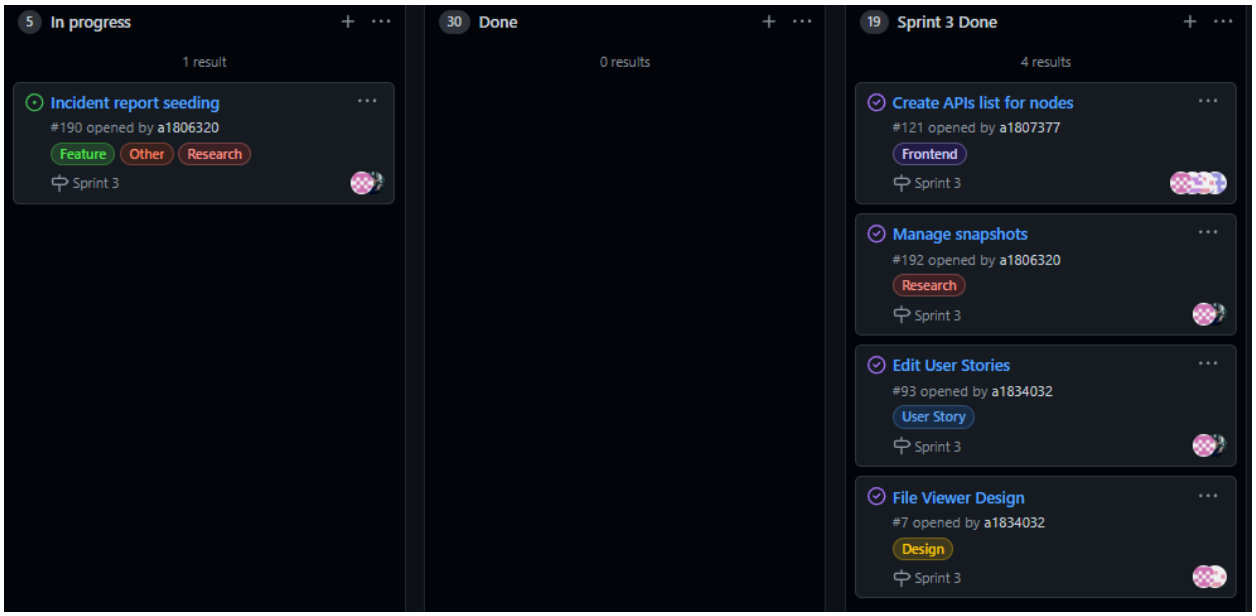

Image 1: Issues for Sprint 3

Over the course of this sprint, I've taken on a series of pivotal roles that contributed significantly to our project's momentum. My involvement in co-creating the API list for nodes, in tandem with the team, was crucial. Designing an API list involves a deep understanding of the system's requirements and ensuring that each API corresponds to a specific functionality. This task, while seemingly straightforward, is laden with intricacies, given the need to anticipate future requirements and scalability.

Furthermore, managing snapshots, along with a teammate, involved meticulous oversight. Ensuring that each snapshot accurately captured our progress, addressed setbacks, and laid out clear future directions was imperative. This task went beyond mere documentation, serving as a compass for our team's direction and approach.

The task of editing and updating user stories, coupled with adding acceptance criteria, required a fine balance between technical understanding and user-centric design. Each user story, while being rooted in technology, had to reflect the user's journey, ensuring the software's relevance and efficacy.

Designing the file viewer page and functionality meant delving deep into user experience design while ensuring the functionality remained seamless.

Lastly, collaborating on collecting, sampling, and replicating MITRE attack flows is a testament to the depth of work undertaken. Given the granularity and detail inherent in MITRE attackflows, this task is not just lengthy but also complex. The precision required ensures that our software is aligned with industry standards, enhancing its credibility and utility. Considering the volume and intricacy of this task, its continuation into sprint 4 is a reflection of our commitment to quality and depth.

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 03/10/2023 with the tutor.

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

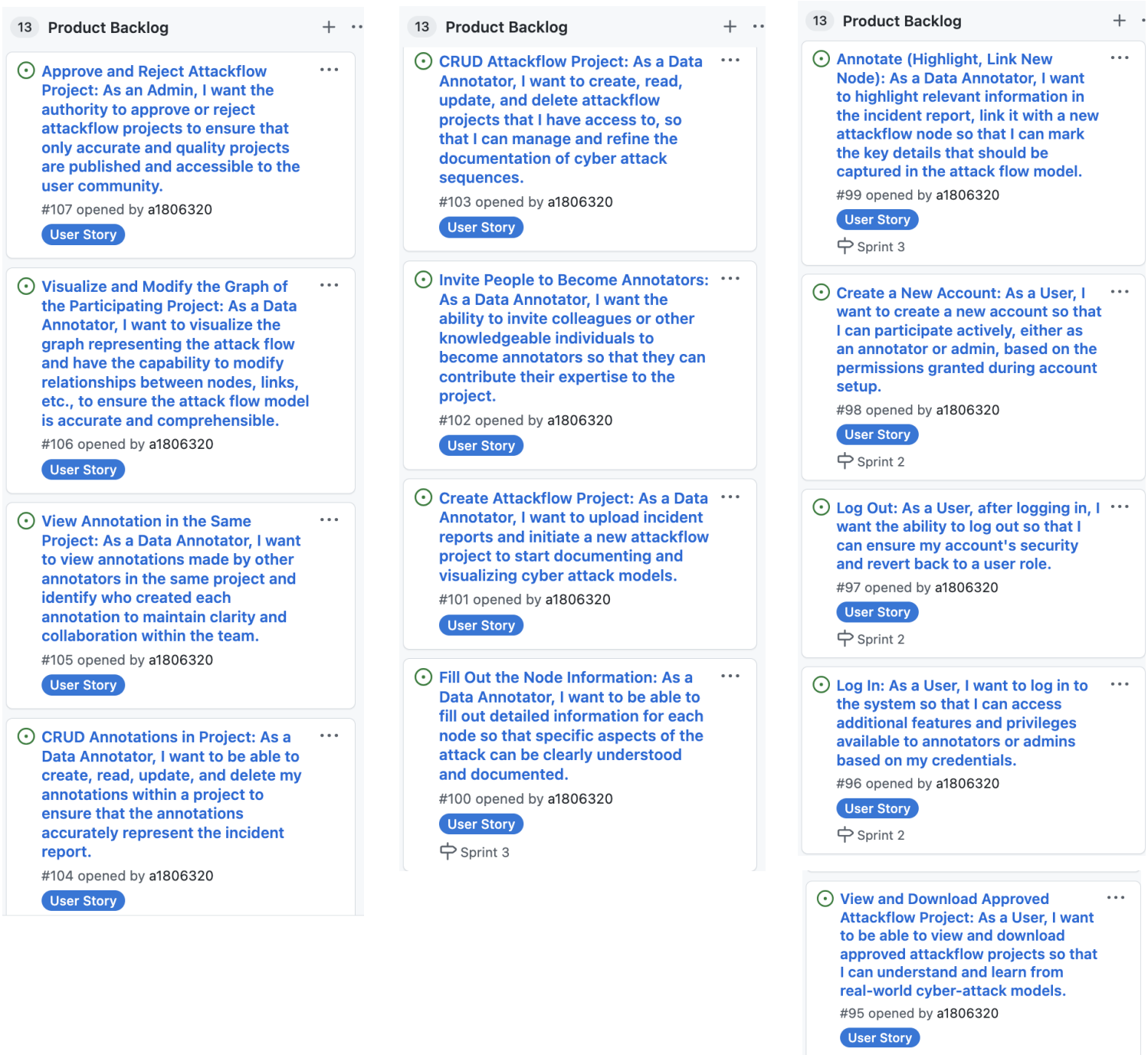

Image 1: Snapshot 3.1 Product Backlog

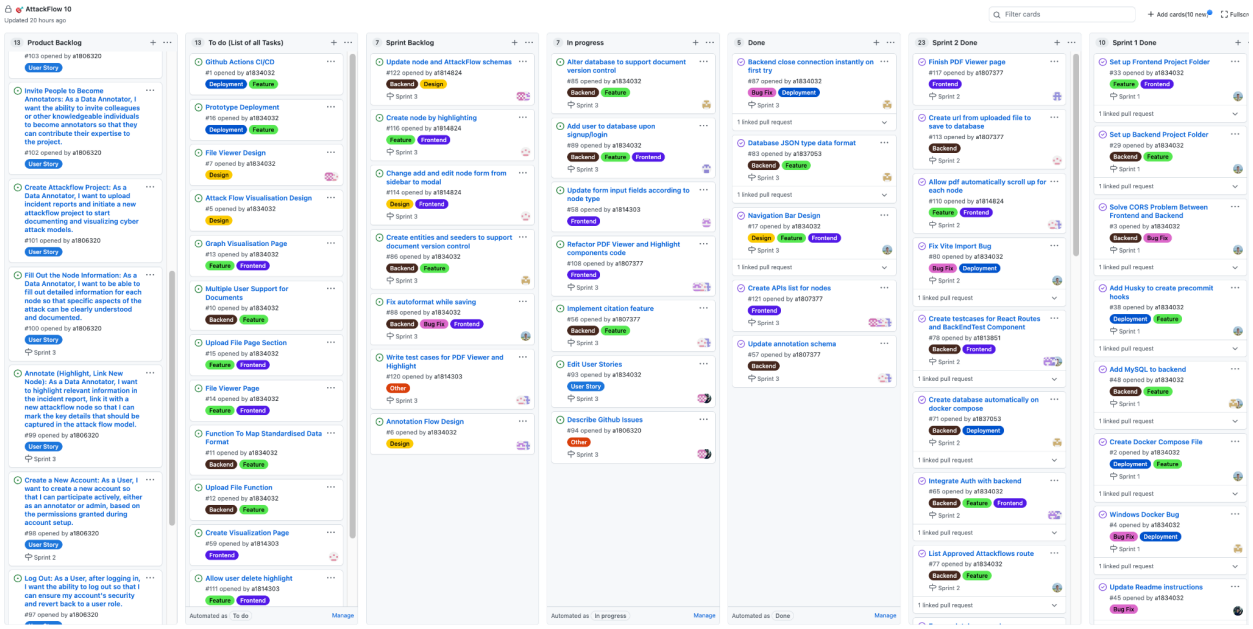

Image 2: Snapshot 3.1 Task Board

Sprint Backlog and User Stories

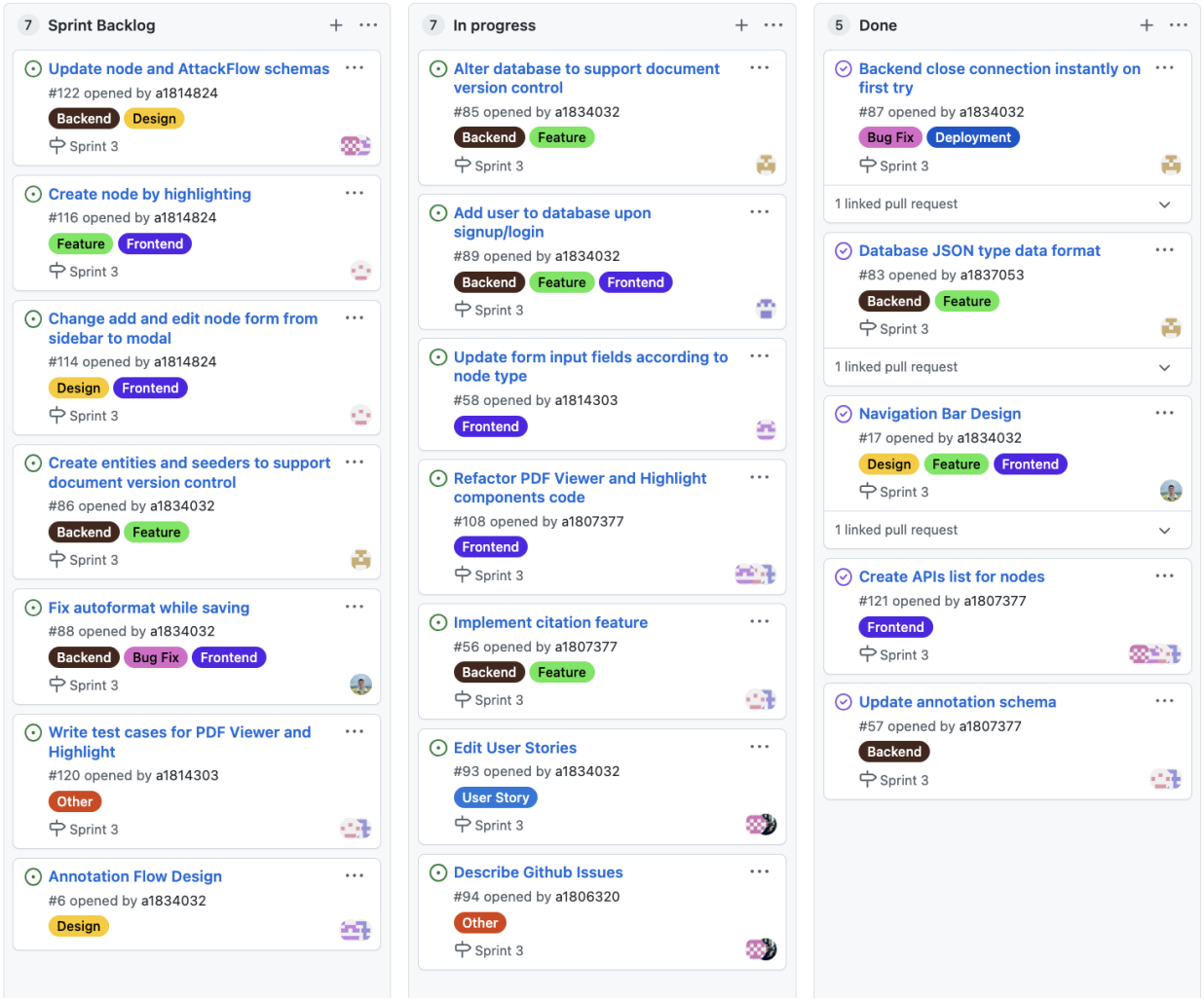

Image 3: Snapshot 3.1 Sprint Backlog

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

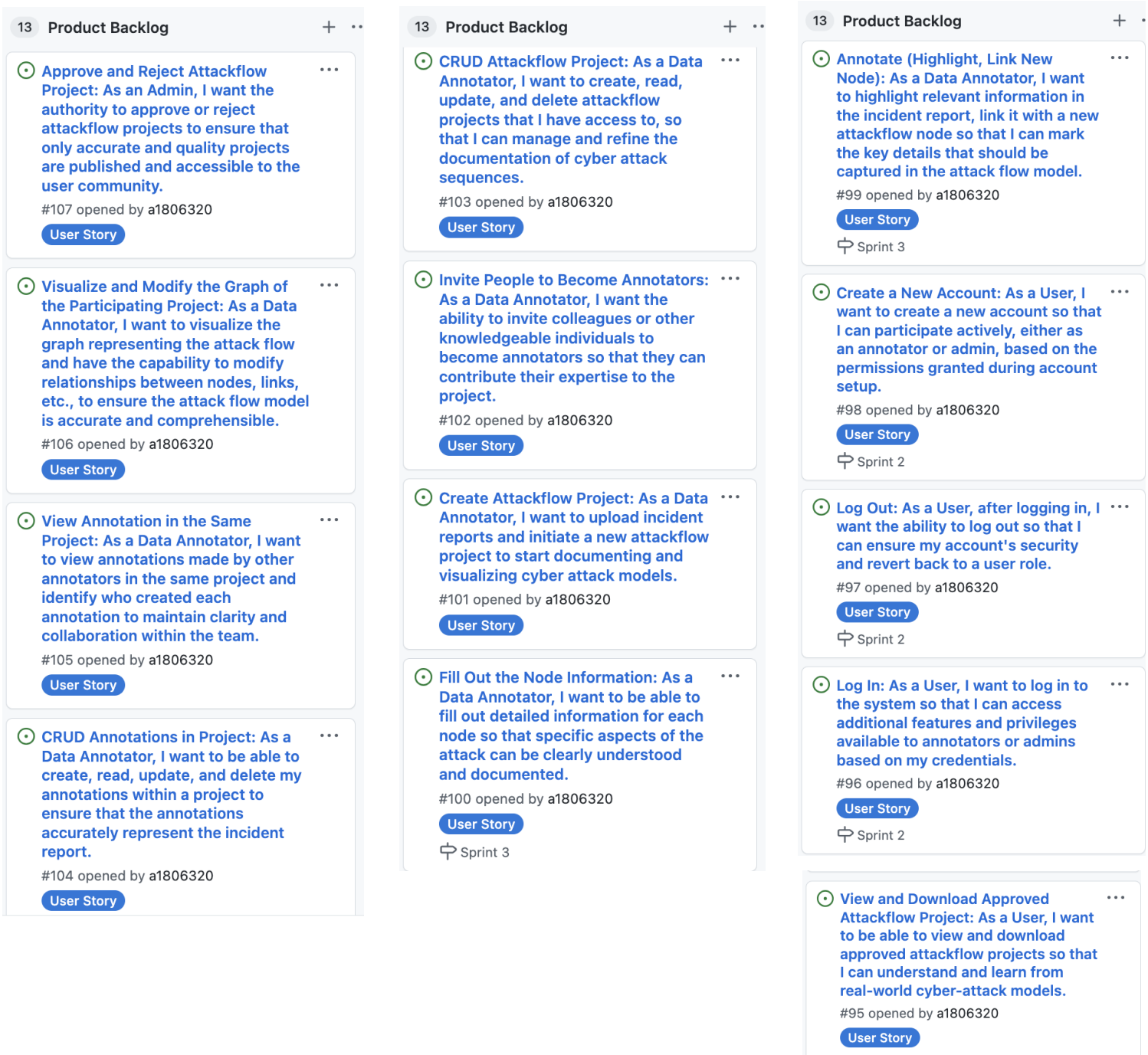

Image 1: Snapshot 3.2 Product Backlog

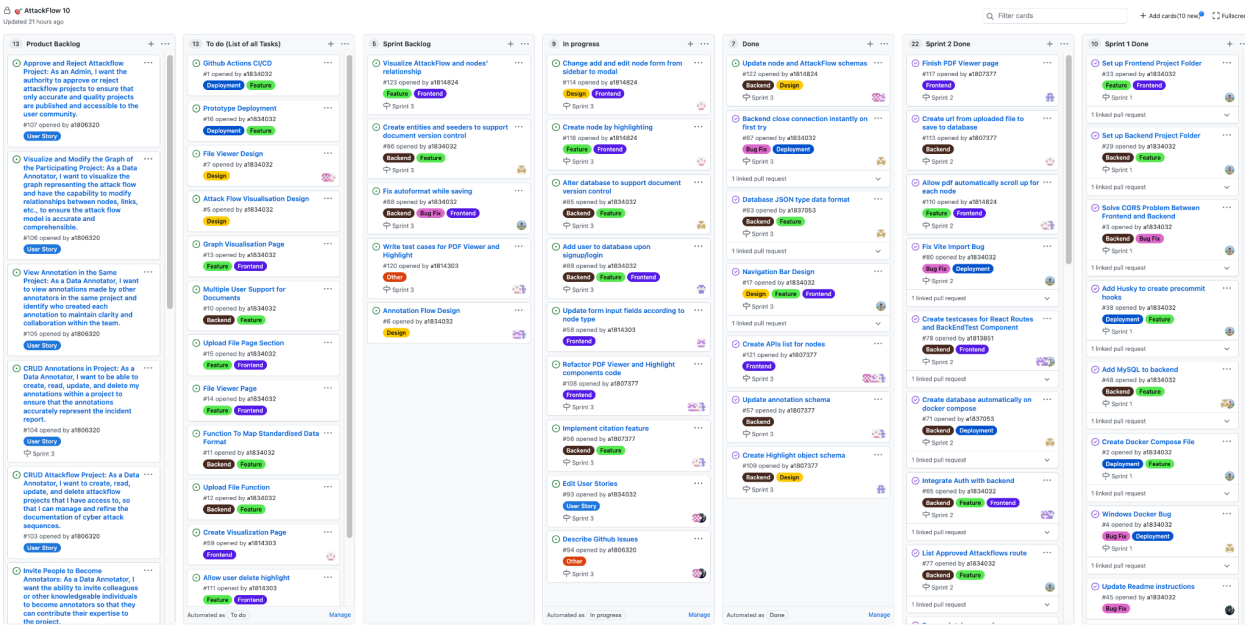

Image 2: Snapshot 3.2 Task Board

Sprint Backlog and User Stories

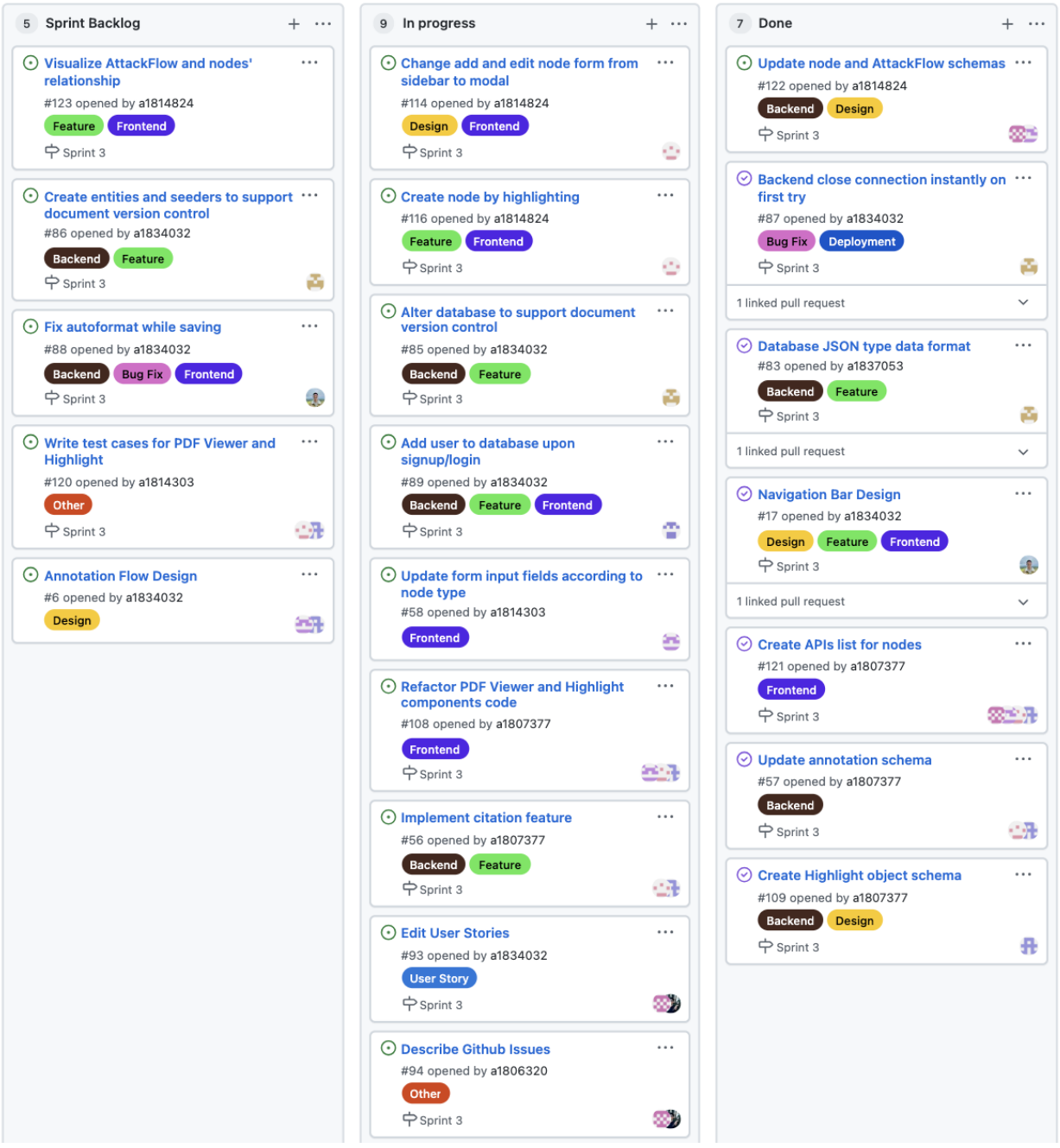

Image 3: Snapshot 3.2 Sprint Backlog

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.