Attack Flow 10

Final Report of Group AttackFlow10

Gia Bao Hoang (a1814824)

Project Vision

In our initial vision for Attackflow, we aspired to bridge the knowledge gap in understanding sophisticated cyberattacks, catering especially to non-cybersecurity experts. The primary objective was to encapsulate these intricate cyberattacks into open-source educational content that was both simplified and accessible. As envisaged, Attackflow was conceptualized as a web platform, ensuring maximum reach and scalability.

Reflecting upon the final product, we are proud to note that our end result largely aligns with our original aspirations. The platform effectively captures and represents sequences of attacks in a user-friendly manner, providing non-cybersecurity experts with an insightful gateway into the realm of cyber threats. The base functionalities laid out in the original vision have been faithfully replicated, and we have even managed to introduce enhanced features that further bolster the tool's utility.

That said, while we have met most of our envisioned goals, the journey wasn't without its challenges. However, the end product is not only a testament to our technical prowess but also to our adaptability and commitment to the original vision. The platform now stands as a valuable tool for generating case studies on cybersecurity breaches, enabling users worldwide to share and gain critical knowledge on cyber threats.

Customer Q&A

Summary of Key Questions Asked During Client Meetings:

1. Platform Choice: Do we have to carry out all discussions on Slack, given that Discord is our team's preferred communication platform?

- Client Response: Slack was chosen as the primary communication platform for ease of communication with the client.

2. Accessibility: Is the client envisioning a platform where all annotated and approved cyberattack flows are freely visualized by users without authentication, given the open-source community orientation?

- Client Response: Yes, the platform was imagined to promote open-source accessibility.

3. Feedback on Current Progress: What is your opinion about the current state of the project? Which aspect would you like to explore more, and which do you consider not as a primary feature? Do you have any questions or suggestions regarding our current progress?

- Client Response: Feedback varied per meeting but was instrumental in refining our project direction, emphasizing essential features, and clearing ambiguities.

Reflection on Client Meetings:

During our project's journey, the client meetings on Microsoft Teams emerged as pivotal milestones, offering me crucial insights into the trajectory we should follow. As I took on the role of Scrum Master in Sprint 2, the responsibility of orchestrating these meetings landed on my shoulders. This opportunity enabled me to bridge the gap between our team's aspirations and the client's expectations. Direct interactions with the client solidified my understanding of their requirements, helping me craft acceptance criteria for our user stories with precision.

My tenure as the Scrum Master was not just about leading; it was about listening. Through these meetings, I delved deep into the client's vision, ensuring we tailored our efforts to what truly mattered. The feedback we received was invaluable, refining our direction and bolstering our confidence that we were indeed aligned with the client's vision.

However, like any journey, ours was not devoid of challenges. While we made substantial strides, there were moments when we couldn't fully align with the client's expectations or timelines. Yet, the transparent communication channels we established ensured that we swiftly realigned our goals, minimizing potential roadblocks.

Reflecting on this experience, I realize the importance of initiating feedback loops earlier in our development process. A more proactive approach could have further narrowed any gaps in understanding. Another key takeaway was recognizing the imperative of complete team involvement. As certain members couldn't attend the kickoff due to other commitments, it underscored the significance of collective participation to foster a comprehensive understanding and foster effective decision-making.

Users and User Stories

Final User Roles

1. Data Annotators

- Responsibilities & Actions: Data Annotators play a crucial role in interpreting and summarizing cyberattacks into an understandable format. They upload incident reports, create and save detailed annotations, and collaborate with other annotators to refine attackflow projects.
- Key User Stories:
    - 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.

- 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.

- 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.

2. Unauthenticated Users

- Responsibilities & Actions: These users leverage Attackflow to educate themselves about cybersecurity through accessing and visualizing annotated cyberattacks. They can view and download approved projects without needing an account.
- Key User Stories:

- 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.

- 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.

3. Admin
- Responsibilities & Actions: Admins ensure the quality and accuracy of content on Attackflow. They review, approve, or reject submitted attackflow projects, playing a gatekeeper role to maintain the platform's integrity.
- Key User Story:
    - 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.

Reflection on Role Evolution from Initial Report

In our initial report, Attackflow was envisioned to cater mainly to two types of users: Data Annotators and Unauthenticated Users. The project has since evolved to include a third crucial role: the Admin. This inclusion has enhanced the platform's integrity and content quality, ensuring that the information disseminated is not only educational but also accurate and reliable.

The development journey of Attackflow brought several refinements to our understanding of user roles and their interactions with the platform. Initially, we underestimated the complexity and necessity of having a robust moderating role, like that of an Admin. This role has become pivotal in maintaining the standard and reliability of content, a feature that was initially overlooked but later realized as critical for the project's success.

The evolution of these roles from our initial vision to the final implementation reflects our growing understanding of the users' needs and the dynamic nature of cybersecurity education. The expanded roles and enhanced functionalities indicate a more mature, user-centric approach, aligning better with our mission to demystify cybersecurity for non-experts.

Software Architecture

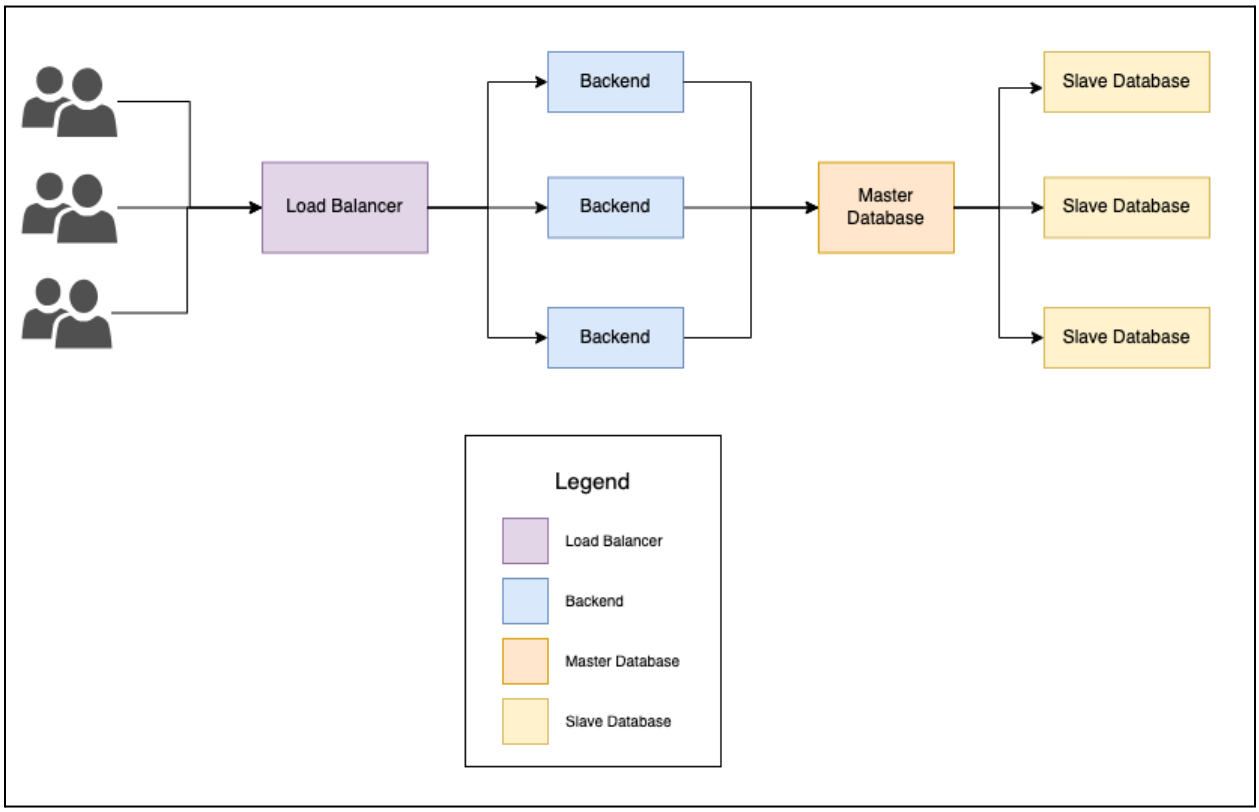


Diagram 1: Software Architecture

Final Architecture Sketch

The provided diagram illustrates the final architecture of our software application. At a high level, the architecture can be understood through its major components and their interactions.

Major Components:

1. Load Balancer: Efficiently routes incoming user requests to the backend servers.
2. Backend Servers: These are the main application servers where most of the processing occurs.
3. Master Database: The primary storage for application data.
4. Slave Databases: Used primarily for read operations, improving efficiency and reducing load on the master database.

Interactions:

1. User Requests: All incoming user requests are initially processed by the Load Balancer. Depending on the nature of the request and the current load on the backend servers, the load balancer routes the request to an appropriate Backend Server.

2. Backend Servers: Upon receiving a request, the backend server processes it and, if required, interacts with the Master Database for write operations or Slave Databases for read operations.

3. Data Replication: Changes made to the Master Database are replicated across Slave Databases to ensure data consistency and availability.

Justification for the Architecture

The architecture for our application was meticulously chosen to prioritize high scalability, availability, and resilience.

1. Scalability & Availability: The Load Balancer ensures that no single backend server is overwhelmed with requests. This allows our system to handle a surge in user traffic without any performance degradation.

2. Data Management & Consistency: The choice of having a Master-Slave Database setup allows us to manage write operations efficiently with the master database, while the read-heavy operations are distributed among the slave databases. This ensures that the system remains performant even during high data retrieval activities.

3. Resilience & Data Protection: With multiple Slave Databases, the system offers a level of redundancy. In the rare event of the master database facing issues, one of the slave databases can be promoted to serve as the new master, ensuring continuous system operation and data protection.

The decision to structure our software architecture in this manner aligns perfectly with the vision and requirements of our application. It not only provides a seamless and efficient user experience but also ensures that the system remains robust and scalable in the face of evolving user demands.

Tech Stacks and Standards

Final Tech Stack:

1. Back-end:

- Node.js: Remaining our primary server-side runtime environment, facilitating scalability and non-blocking asynchronous operations.

2. Frameworks:
- Express.js: Our choice for the backend framework remained consistent due to its flexibility, efficiency, and speed.
- Docker: Used for containerization of our application, ensuring consistent environments and scalability.
- Jest: Employed for automated testing to ensure software reliability.

3. Front-end:
- React.js: Chosen for its virtual DOM and component-based architecture, it allowed us to build a responsive and dynamic user interface.
- Vite: This build tool enables faster refresh and optimized builds.
- TailwindCSS: Offers utility-first CSS framework for rapid UI development.
- Auth0: Handled user authentication securely, integrating seamlessly with React.

4. Programming Languages:
- JavaScript: The bedrock of our application. Given its versatility and compatibility with our chosen frameworks and tools, it was the natural choice.

5. Libraries:
- Axios: Employed for making HTTP requests. Its promise-based structure simplified error handling.

6. Dropped from Initial Stack:
- TypeORM: We found that our needs were sufficiently addressed using Axios and other native methods, reducing complexity.

Communication & Development:

- Slack: Remained as our professional communication platform with external stakeholders like our proxy client.
- Discord: A favored platform for internal team discussions, brainstorming, and quick chats.
- GitHub Projects: Allowed us to streamline our project management process, integrating issue tracking, pull requests, and code repositories.
- VSCode: Continued to be our IDE of choice. Its vast array of extensions, debugger, and Git capabilities streamlined our development process.

Coding Standards:

- AirBNB’s JavaScript Style Guide: Given its popularity and comprehensiveness, this remained our go-to guide.
- ESLint: This tool ensured that our code consistently adhered to the style guide.

- .vscode settings: Ensured uniformity in code appearance across all developers' machines.

Reflection and Justification:

Our decisions regarding the tech stack and standards were rooted in a combination of team familiarity, industry trends, and the specific needs of our application. Sticking mostly to the initial plan speaks volumes about the foresight we had, ensuring minimal disruptions in the development process. The decision to eliminate TypeORM and embrace Axios was driven by our endeavor to keep the stack lean and minimize overheads. Our communication tools streamlined interactions, with distinct channels for client and internal communications. The consistency in coding standards, achieved through tools like ESLint and .vscode settings, ensured that the team produced uniform, clean, and maintainable code.

In contrast to our initial report, the evolution of our tech stack was minor but significant. We observed that a well-thought-out initial plan minimized the need for drastic changes down the line. The success of our application can be attributed to the blend of robust tech choices and rigorous coding standards that we adhered to from the start.

Group Meetings and Team Member Roles

Frequency and Duration of Meetings:

Our team maintained a consistent rhythm with our meetings, encompassing:
1. Daily Stand-Up Meetings: These were brief, around 10 minutes, typically scheduled post-7:30 pm to ensure everyone's availability.
2. Weekly Progress Meetings: Held every Thursday from 3:30 pm to 4:30 pm, where we delved deeper into the week's progress, challenges, and next steps.
3. Fortnightly Sprint Retrospective Meetings: These were scheduled every alternate Tuesday from 7:00 pm to 7:25 pm. This time was dedicated to reflections on the past sprint, understanding what worked, what didn't, and how we could improve.

Timeboxing and Meeting Effectiveness:

We initially planned for meetings to last between 1.5 to 2 hours weekly. Over time, we've seen the benefits of timeboxing, especially with our stand-up meetings and sprint retrospectives. By keeping these meetings succinct, we ensured focused discussions, efficient use of time, and maintained momentum throughout our sprints. Though there were instances when we extended slightly beyond our scheduled time, it was always to ensure clarity and consensus.

Feedback Channels:

Apart from meetings, we established a regular communication channel with our client through Slack. This ensured quick feedback loops, helping us get timely clarifications, negotiate meeting

timings, and present our development progress. While Microsoft Teams was our chosen platform for official proxy client meetings, Slack played a pivotal role in facilitating informal yet crucial interactions.

Scrum Masters:

For each sprint, the role of the Scrum Master was rotated to ensure different perspectives and to distribute leadership responsibilities:
- Sprint 1: Jie Shen Beh
- Sprint 2: Gia Bao Hoang
- Sprint 3: Hung Yee Wong
- Sprint 4: Jiajun Yu
- Sprint 5: Jian Zhe Chan

Personal Reflection:

Looking back, our meeting structure significantly contributed to our software development process. The scheduled meetings, paired with effective time management, ensured alignment with our Agile development principles. My role in the project was multifaceted. Administratively, I managed snapshots, sprint retrospectives, and reports. I dedicated significant effort into defining clear user stories, setting the stage for development. On the technical side, my research and experimentation with tools were crucial to actualizing the 'annotate' and 'visualize' functions. Designing the annotation and visualization pages was both a challenge and a reward. As a Scrum Master during Sprint 2, I could experience firsthand the responsibility of steering the team, ensuring we stayed true to our goals and principles. These roles, combined, have provided me with a holistic view of the project, its challenges, and triumphs, reinforcing my belief in the power of team synergy and the Agile process.

Snapshot

Attack Flow

Snapshot Week 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)

Product Backlog and Task Board

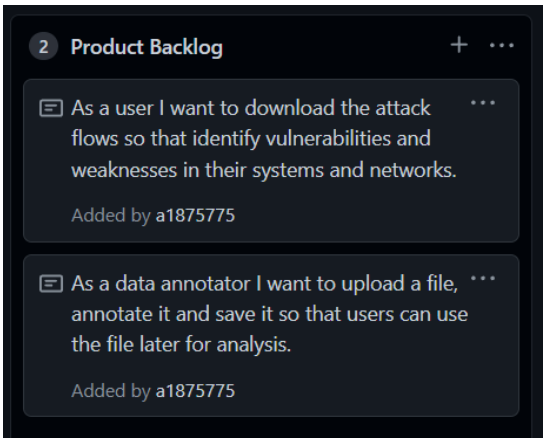


Image 1: Snapshot 1.1 Product Backlog

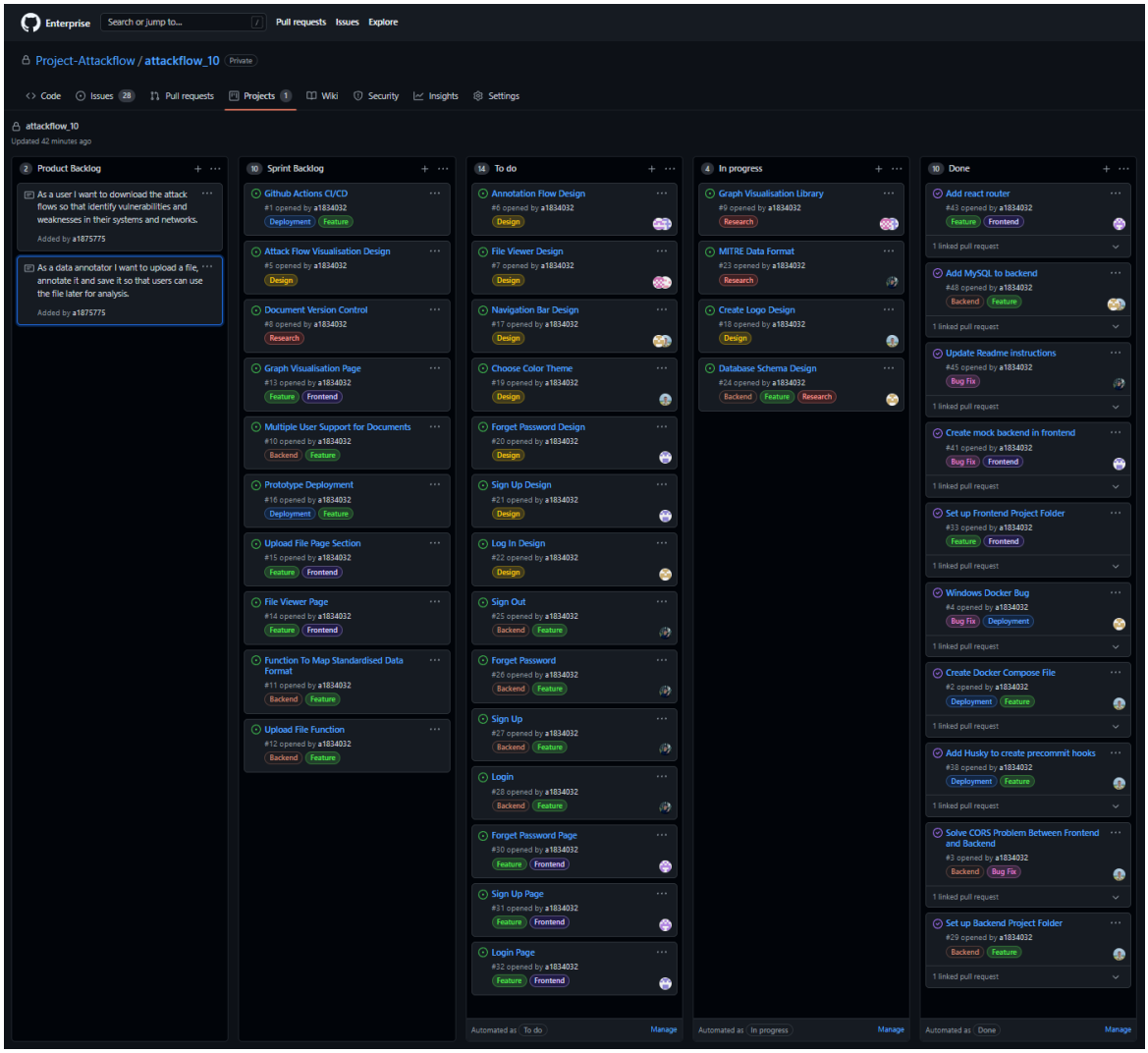


Image 2: Snapshot 1.1 Task Board

Sprint Backlog and User Stories

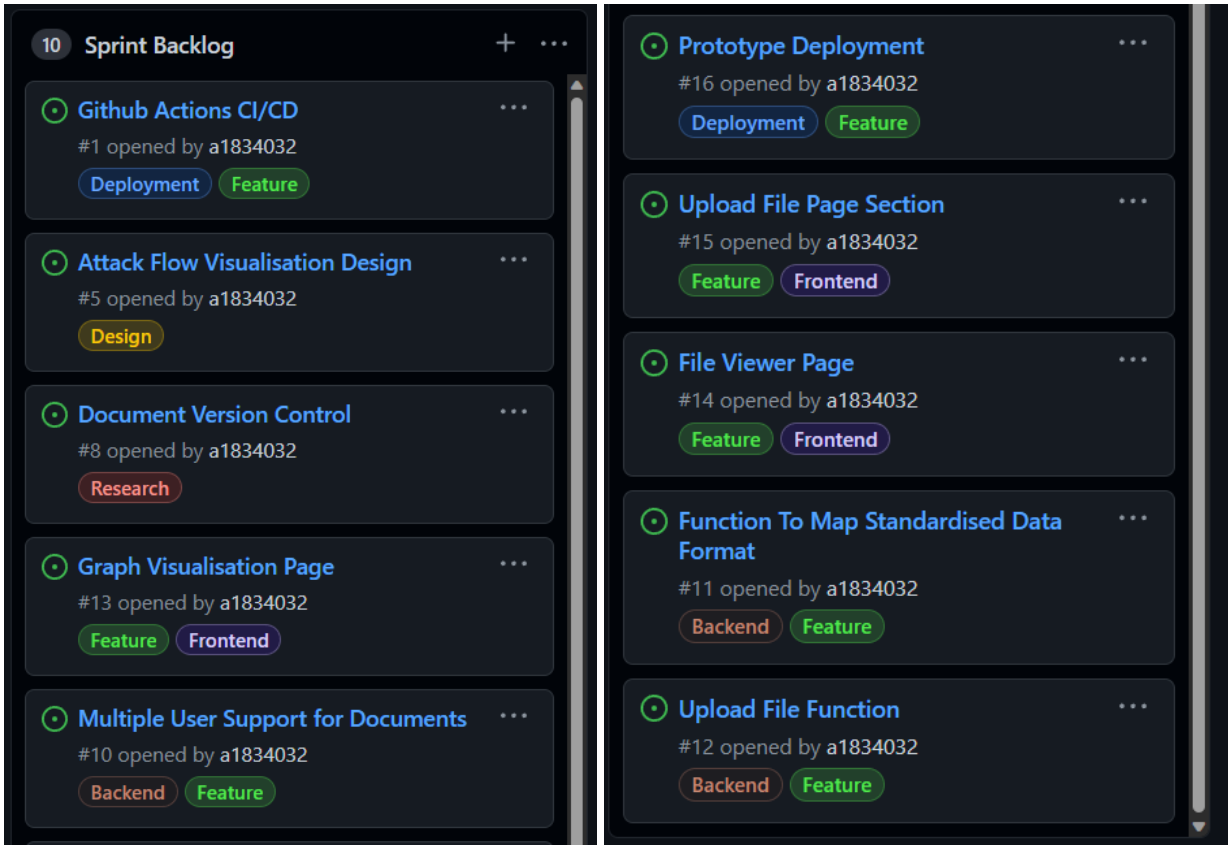


Image 3: Snapshot 1.1 Sprint Backlog

User Stories

1. **As a user I want to download the attack flows so that I can identify vulnerabilities and weaknesses in their systems and networks.**
    a. Description: The user will be able to download the attackflows after it has been published. This feature enables them to simulate potential attack scenarios and strengthen their defenses for prevention.
    b. Related tasks include researching on MITRE Data Format and backend function to map standardized data format.
2. **As a data annotator I want to upload a file, annotate it and save it so that users can use the file later for analysis.**
    a. Description: The data annotator will be able to create annotations using their cybersecurity knowledge on an uploaded incident report. This feature will create a standardized data format for visualization. Then, users can analyze the visualization and prevent future cyberattacks in their systems.

b. Related tasks include setting up the frontend project folder, designing the file viewer after a user uploads the file and researching on the document version control.

Note:

- The user stories are highlighted in **bold** because they are first introduced in this sprint.
- For detailed insights into the Task Board, Product Backlog, and Sprint Backlog, please click here.

Definition of Done

Our definition of done is split into two parts for the first snapshot, which are general goals and specific goals.

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.

Specific Goals:

- **User Registration**: A new user can register using an unused email address without any glitches.
- **User Authentication**: Users can log in using their valid registered email addresses.
- **Password Recovery**: Registered users who forget their passwords have a clear and secure process to reset them.
- **User Logout**: Logged-in users can easily and securely sign out of their accounts.

Summary of Changes

In this first snapshot, our team has started working on the AttackFlow project. Since this is just the beginning, there haven't been any changes or updates yet.

Attack Flow

Snapshot Week 5 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

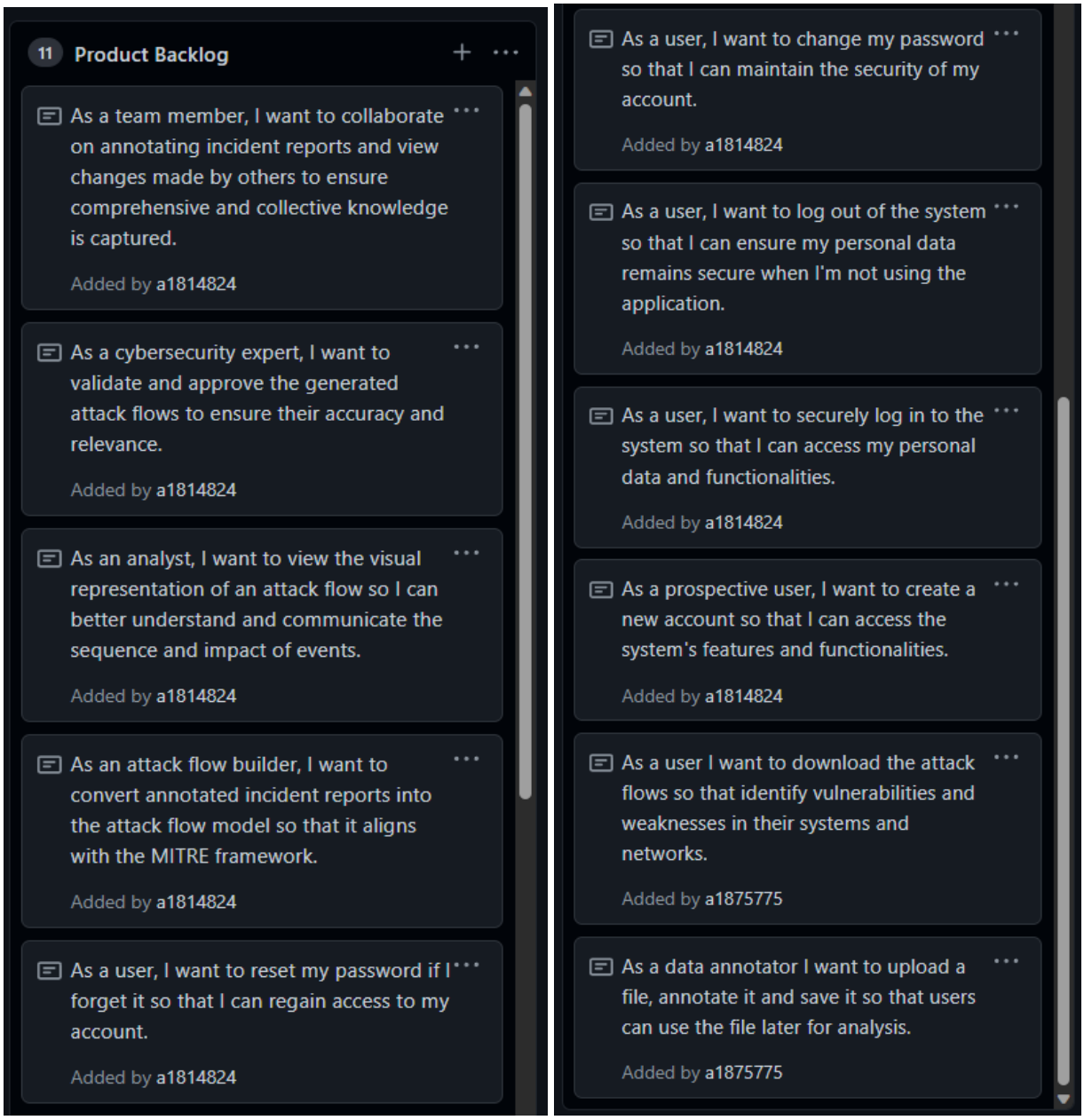


Image 1: Snapshot 2.1 Product Backlog

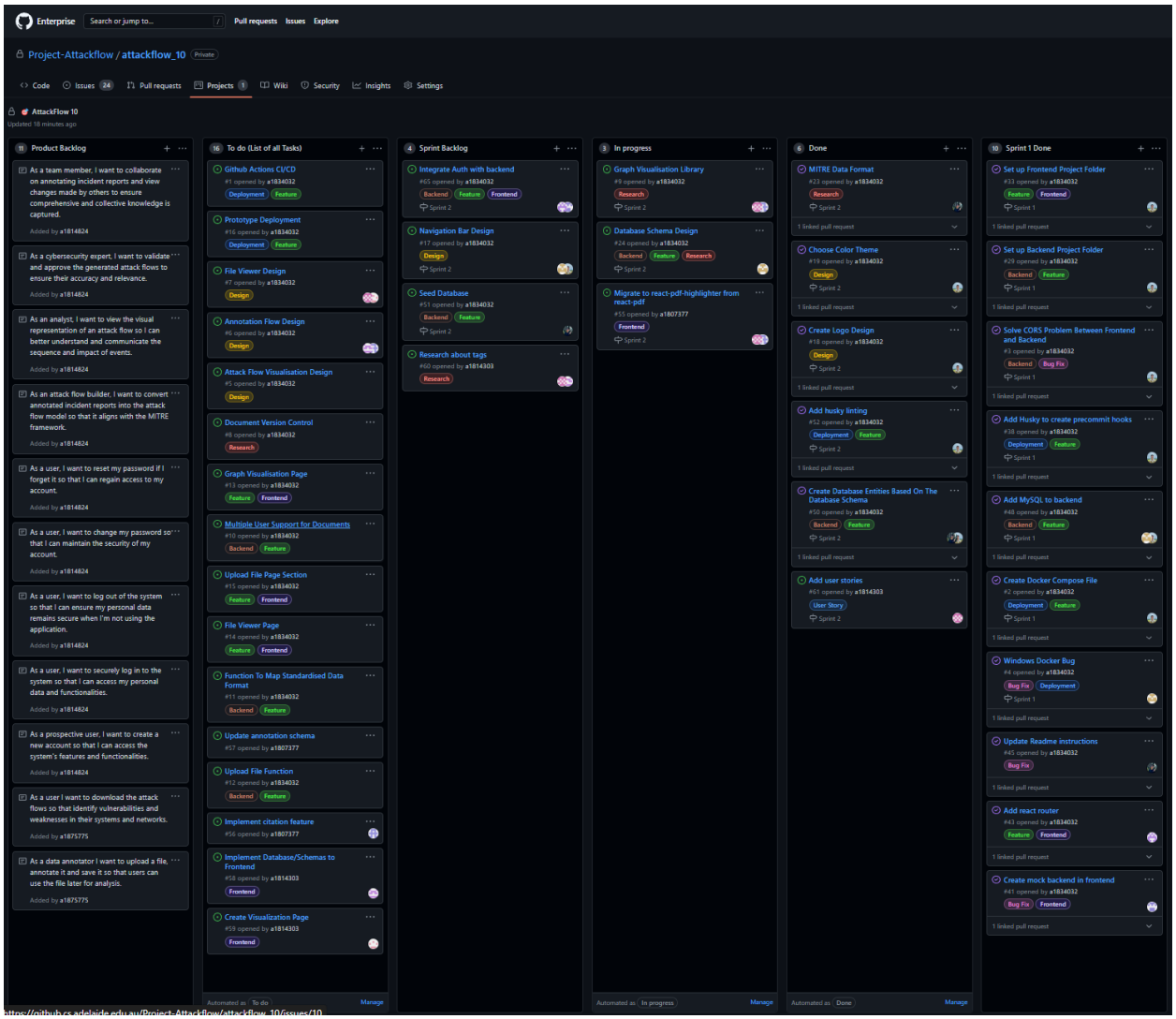


Image 2: Snapshot 2.1 Task Board

Sprint Backlog and User Stories

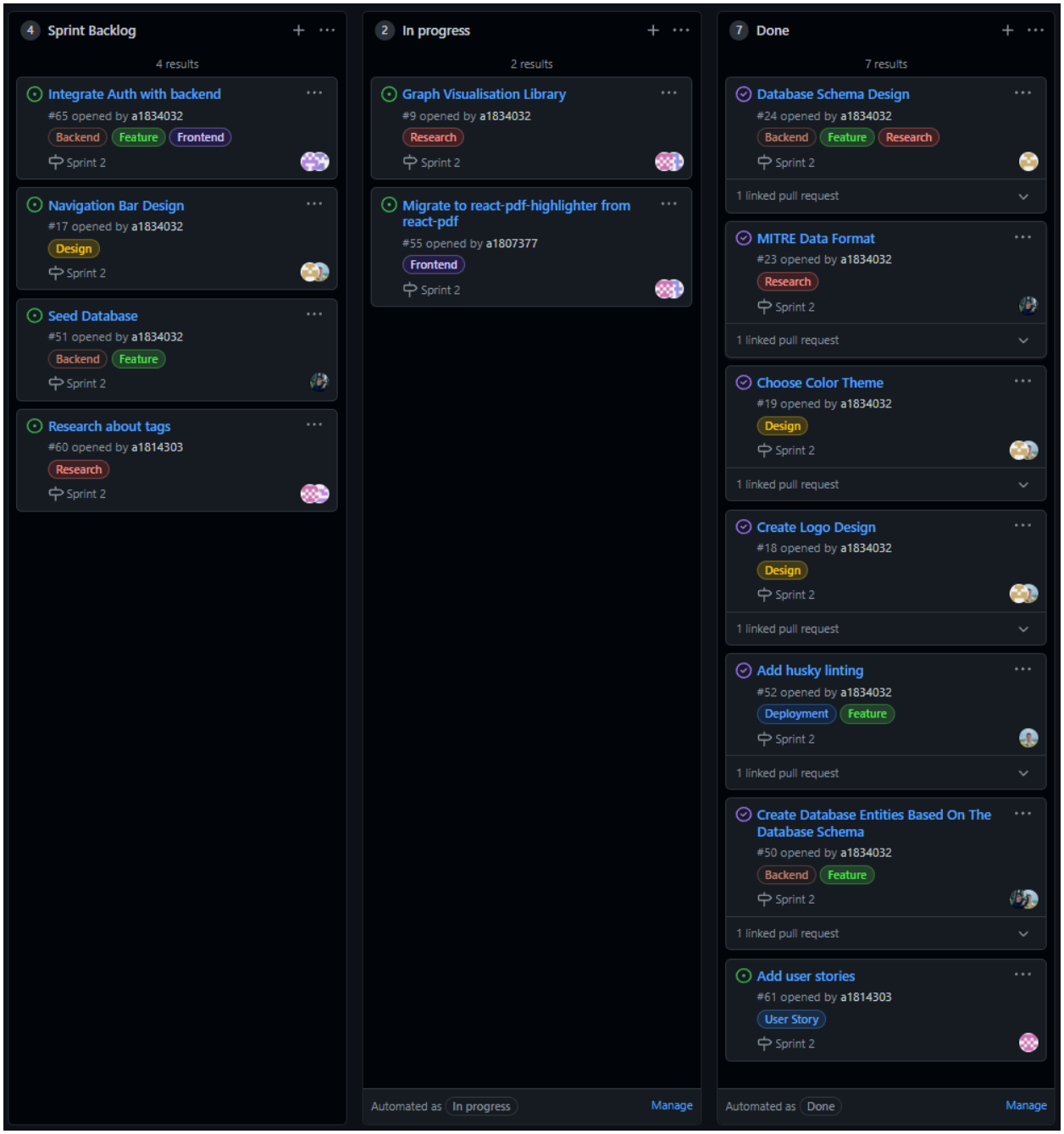


Image 3: Snapshot 2.1 Sprint Backlog

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

1. As a user I want to download the attack flows so that I can identify vulnerabilities and weaknesses in their systems and networks.
    a. Description: The user will be able to download the attackflows after it has been published. This feature enables them to simulate potential attack scenarios and strengthen their defenses for prevention.
    b. Related tasks include researching on MITRE Data Format and backend function to map standardized data format.
2. As a data annotator I want to upload a file, annotate it and save it so that users can use the file later for analysis.
    a. Description: The data annotator will be able to create annotations using their cybersecurity knowledge on an uploaded incident report. This feature will create a standardized data format for visualization. Then, users can analyze the visualization and prevent future cyberattacks in their systems.
    b. Related tasks include setting up the frontend project folder, designing the file viewer after a user uploads the file and researching on the document version control.
3. **As a prospective user, I want to create a new account so that I can access the system's features and functionalities.**
    a. Description: The user should be presented with a registration form that captures all necessary details for account creation. Upon successful registration, they should receive a confirmation email or notification.
    b. Related Tasks:
        - Design the registration UI. (Done)
        - Implement backend logic for account creation.
        - Research and integrate email notification systems for account confirmations.

4. **As a user, I want to securely log in to the system so that I can access my personal data and functionalities.**
    a. Description: The user should be presented with a simple and intuitive login form that asks for their credentials. Any unsuccessful login attempts should provide feedback to guide the user.
    b. Related Tasks:
        - Design the login UI. (Done)
        - Implement backend authentication logic.
        - Integrate error handling and feedback mechanisms.

5. **As a user, I want to log out of the system so that I can ensure my personal data remains secure when I'm not using the application.**
    a. Description: The user should easily find and use the logout functionality. After logging out, their session should be terminated.
    b. Related Tasks:

- Add the logout button to the UI. (Done)
- Remove JWT Token after logout.

6. **As a user, I want to change my password so that I can maintain the security of my account.**
    a. Description: The user should be able to easily locate the change password option, be prompted for their current password, and then specify a new password.
    b. Related Tasks:
        - Design update password page. (Done)
        - Implement backend logic for password updates.
        - Return message and status of request for successful or unsuccessful password changes.

7. **As a user, I want to reset my password if I forget it so that I can regain access to my account.**
    a. Description: The user should find a "Forgot Password" option in the login screen, which guides them through the process of resetting their password via their registered email.
    b. Related Tasks:
        - Design UI for password reset. (Done)
        - Implement backend logic to send reset password emails.
        - Ensure secure token generation for password reset.

8. **As an attack flow builder, I want to convert annotated incident reports into the attack flow model so that it aligns with the MITRE framework.**
    a. Description: The system should interpret annotations and use them to create a standardized attack flow model.
    b. Related Tasks:
        - Understand and implement the MITRE framework format.
        - Design algorithms or tools to convert annotations into attack flow models.

9. **As an analyst, I want to view the visual representation of an attack flow so I can better understand and communicate the sequence and impact of events.**
    a. Description: Once the attack flow model is generated, the system should provide a visualization tool or integrate with the MITRE visualization tool.
    b. Related Tasks:
        - Integrate with the MITRE visualization tool or develop a custom visualization module.
        - Ensure visual clarity and interactive features for better analysis.

10. **As a cybersecurity expert, I want to validate and approve the generated attack flows to ensure their accuracy and relevance.**
    a. Description: Users should be able to review, modify, and approve attack flows to ensure they accurately represent the incident report.
    b. Related Tasks:
        - Design a review and validation interface.
        - Implement feedback feature for continuous improvement.

11. **As a team member, I want to collaborate on annotating incident reports and view changes made by others to ensure comprehensive and collective knowledge is captured.**
    a. Description: The platform should support multiple users annotating a document and provide version control to track changes.
    b. Related Tasks:
        - Integrate collaboration tools or features.
        - Implement a version control system for documents.

Note: The user stories are highlighted in **bold** because they are first introduced in this sprint.

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

Definition of Done

For our second 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.

Specific Goals:

- **User Registration**: A new user can smoothly register using an unused email address, receiving a confirmation notification after successful registration.
- **User Authentication**: Users can log in securely using their registered credentials. Additionally, the system provides clear feedback for any unsuccessful login attempts.
- **Password Management**: Users can change their passwords without any hassle. Moreover, if they forget their passwords, they can initiate a clear and secure recovery process.
- **User Logout**: Users can easily log out, ensuring their session is terminated and their personal data remains secure.
- **File Upload and Annotation**: Data annotators can upload files and annotate specific sections within them. The annotations should be clear and easily modifiable.
- **Attack Flow Integration**: The system can interpret annotations from uploaded files to generate attack flow models aligned with the MITRE framework.
- **Visualization**: Analysts and other users can view a clear visual representation of any attack flow, providing an easy understanding of sequences and impact.
- **Validation and Collaboration**: Cybersecurity experts can review and validate generated attack flows. Team members can collaborate on annotations and utilize a version control system to track any changes to incident reports.
- **Download Capability**: Users can download the attack flows for offline analysis or sharing. The downloaded format should be in line with the MITRE Data Format.

Summary of Changes

Since our last update, we've been making significant progress. We've built a strong foundation for the system by carefully creating a solid database structure. Our new authentication system is now operational, enhancing security for user interactions. The real achievement was seamlessly connecting the backend with the frontend, resulting in a user-friendly experience that flows smoothly. To give our project a unique identity, we've crafted an eye-catching logo and selected colors that truly reflect its essence. And that's not all – we've also successfully developed two key interface elements: the file viewer page and attack flow nodes list. Additionally, we've integrated a side menu to enhance navigation. These improvements showcase our dedication to both functionality and aesthetics, marking a clear advancement since the last snapshot.

Attack Flow

Snapshot Week 6 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

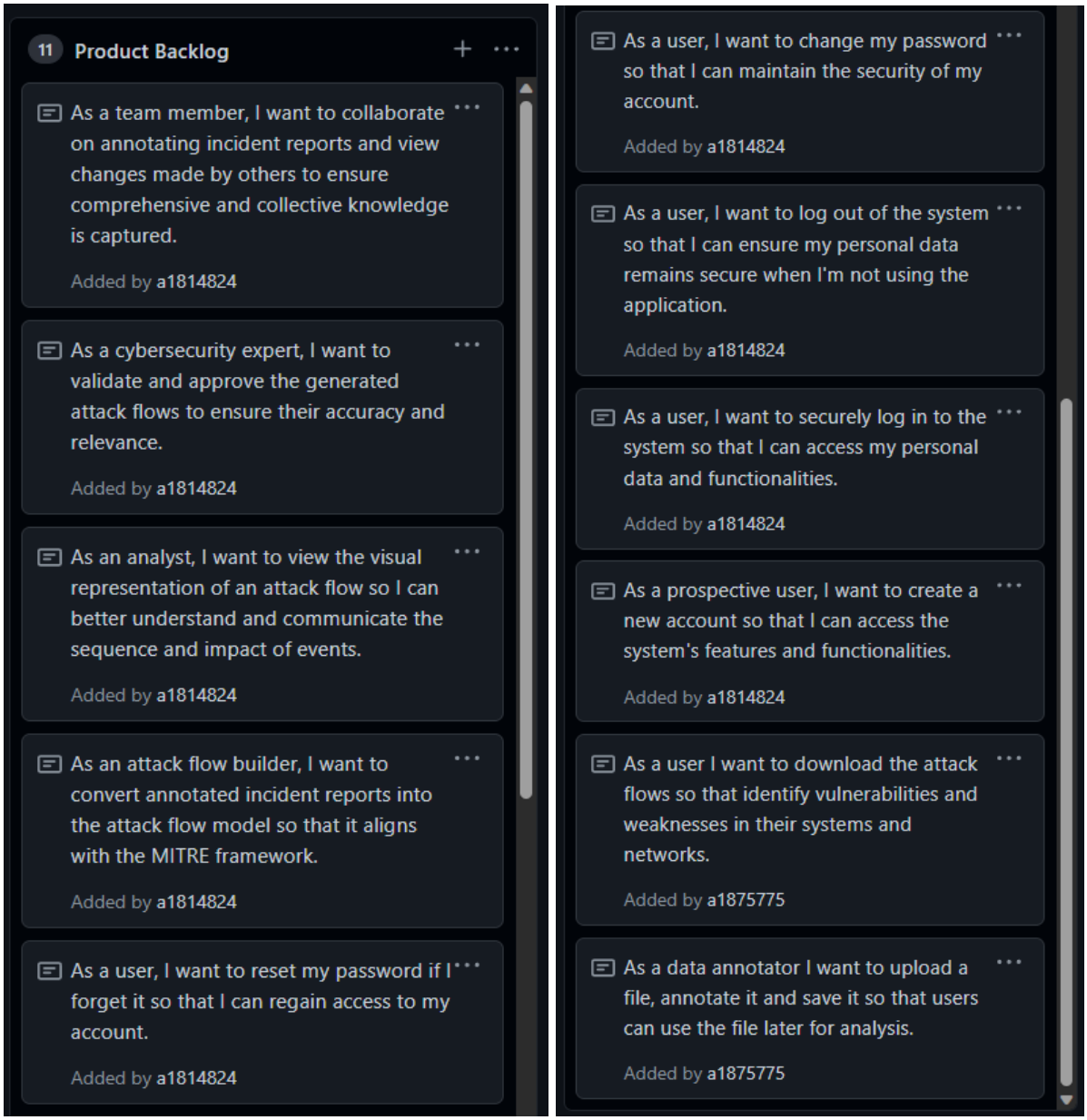


Image 1: Snapshot 2.2 Product Backlog

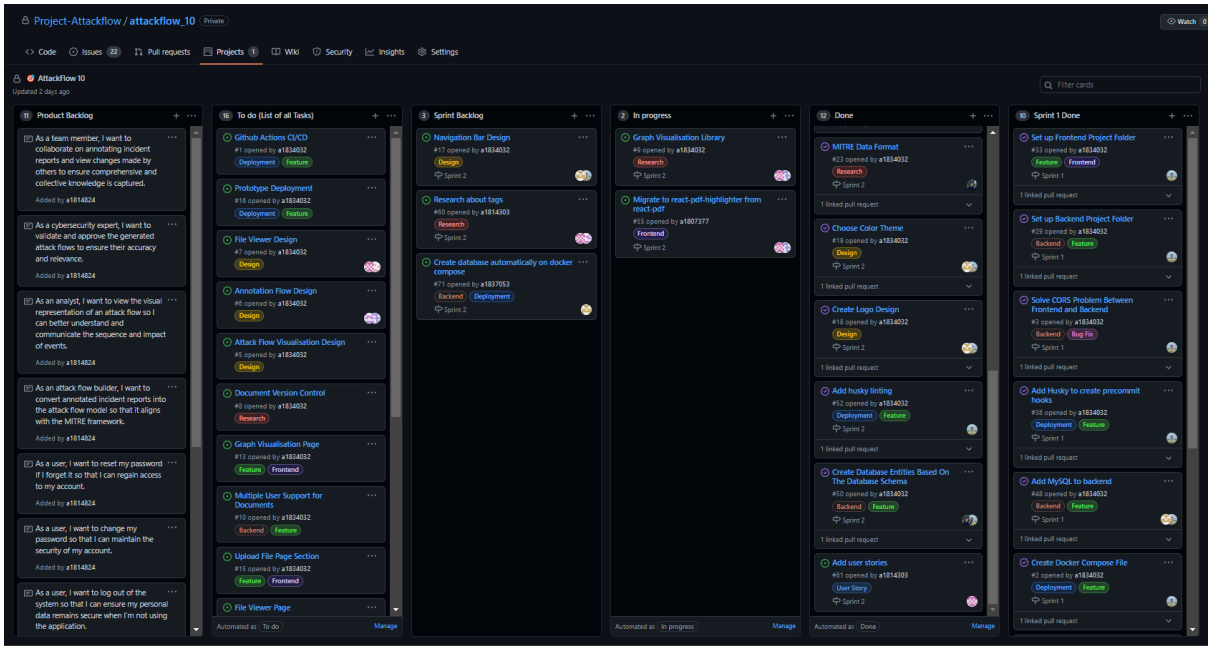
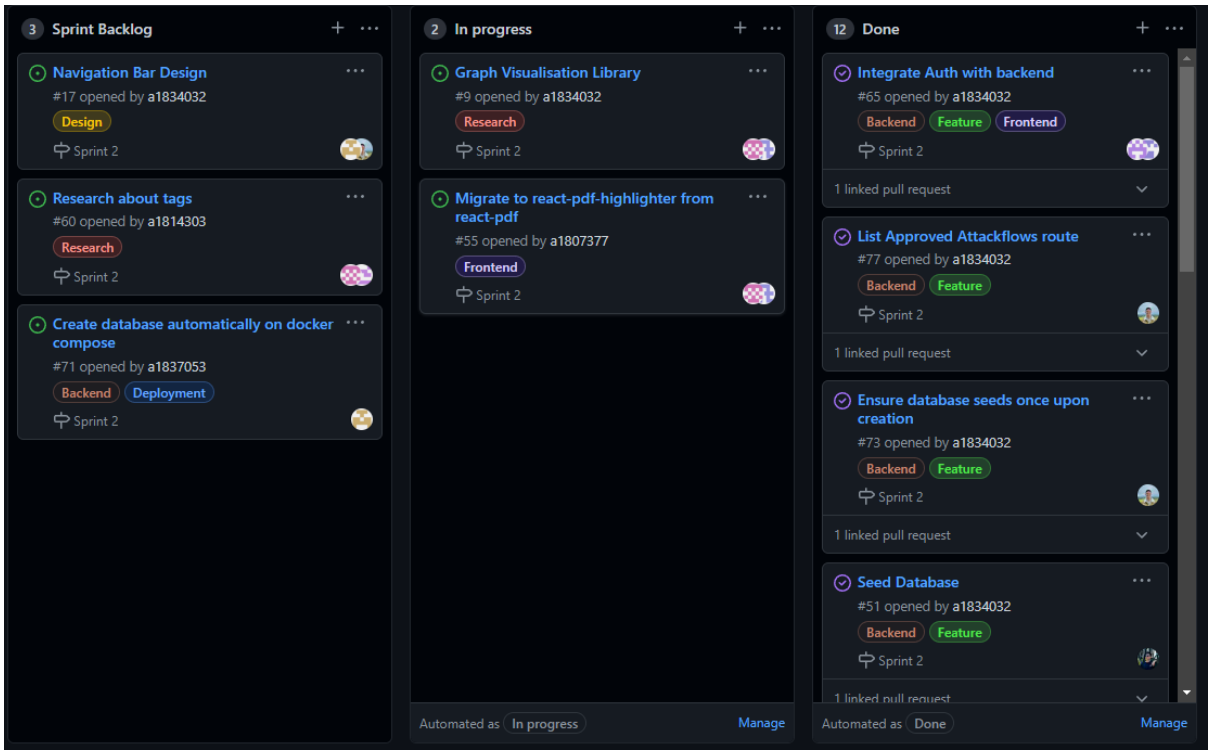


Image 2: Snapshot 2.2 Task Board

Sprint Backlog and User Stories



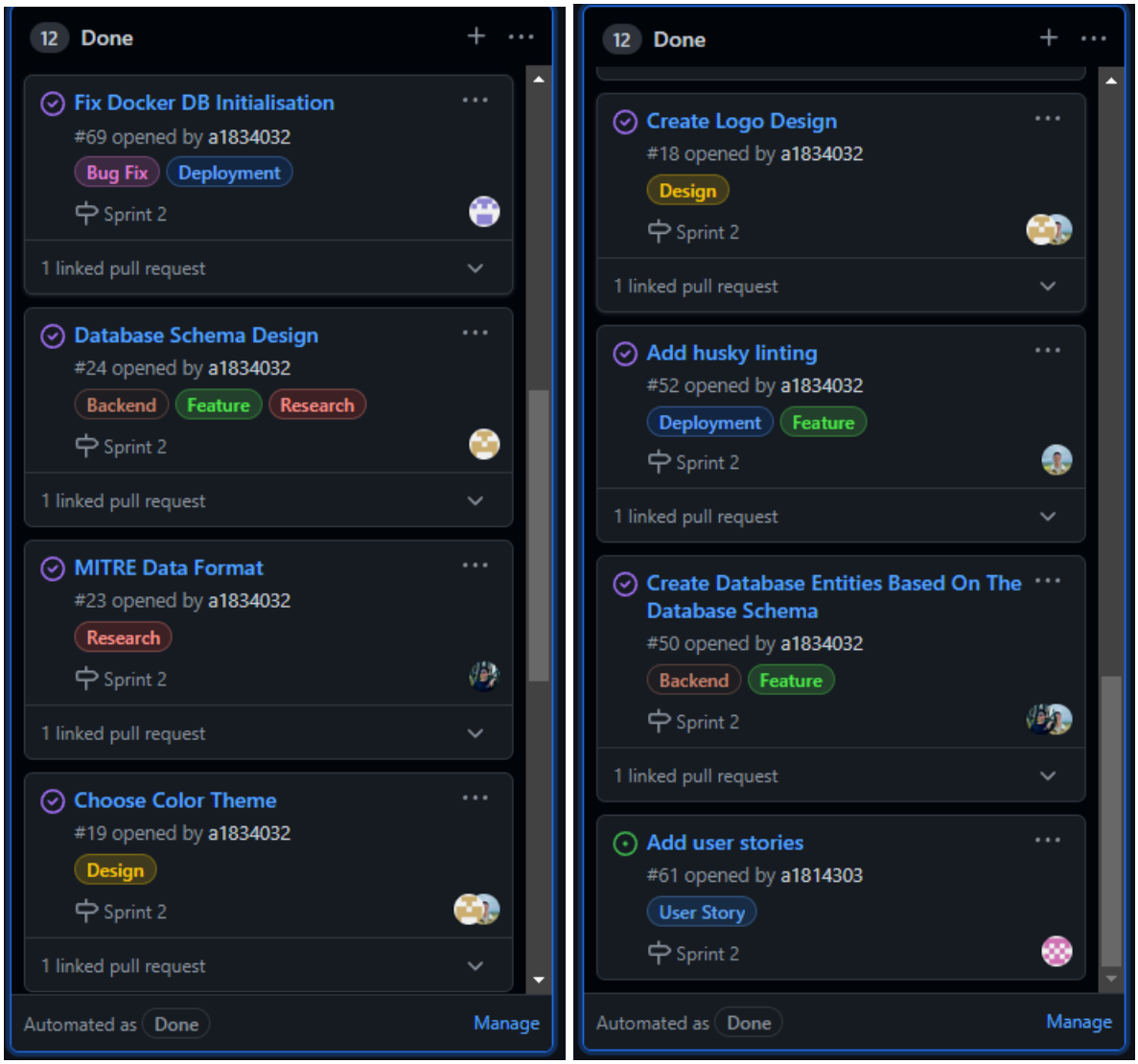


Image 3: Snapshot 2.2 Sprint Backlog

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. As a prospective user, I want to create a new account so that I can access the system's features and functionalities.
    a. Description: The user should be presented with a registration form that captures all necessary details for account creation. Upon successful registration, they should receive a confirmation email or notification.
    b. Related Tasks:
        - Design the registration UI. (Done)
        - Implement backend logic for account creation. (Done)
        - Research and integrate email notification systems for account confirmations.

2. As a user, I want to securely log in to the system so that I can access my personal data and functionalities.
    a. Description: The user should be presented with a simple and intuitive login form that asks for their credentials. Any unsuccessful login attempts should provide feedback to guide the user.
    b. Related Tasks:
        - Design the login UI. (Done)
        - Implement backend authentication logic.
        - Integrate error handling and feedback mechanisms.

3. As a user, I want to log out of the system so that I can ensure my personal data remains secure when I'm not using the application.
    a. Description: The user should easily find and use the logout functionality. After logging out, their session should be terminated.
    b. Related Tasks:
        - Add the logout button to the UI. (Done)
        - Remove JWT Token after logout.

4. As a user, I want to change my password so that I can maintain the security of my account.
    a. Description: The user should be able to easily locate the change password option, be prompted for their current password, and then specify a new password.
    b. Related Tasks:
        - Design update password page. (Done)
        - Implement backend logic for password updates.
        - Return message and status of request for successful or unsuccessful password changes.

5. As a user, I want to reset my password if I forget it so that I can regain access to my account.

a. Description: The user should find a "Forgot Password" option in the login screen, which guides them through the process of resetting their password via their registered email.
b. Related Tasks:
    - Design UI for password reset. (Done)
    - Implement backend logic to send reset password emails.
    - Ensure secure token generation for password reset.

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

Definition of Done

For our third 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:

- **User Registration**: A new user can smoothly register using an unused email address, receiving a confirmation notification after successful registration.
- **User Authentication**: Users can log in securely using their registered credentials. Additionally, the system provides clear feedback for any unsuccessful login attempts.

- **Password Management**: Users can change their passwords without any hassle. Moreover, if they forget their passwords, they can initiate a clear and secure recovery process.
- **User Logout**: Users can easily log out, ensuring their session is terminated and their personal data remains secure.
- **File Upload and Annotation**: Data annotators can upload files and annotate specific sections within them. The annotations should be clear and easily modifiable.
- **Attack Flow Integration**: The system can interpret annotations from uploaded files to generate attack flow models aligned with the MITRE framework.
- **Visualisation**: Analysts and other users can view a clear visual representation of any attack flow, providing an easy understanding of sequences and impact.
- **Validation and Collaboration**: Cybersecurity experts can review and validate generated attack flows. Team members can collaborate on annotations and utilise a version control system to track any changes to incident reports.
- **Download Capability**: Users can download the attack flows for offline analysis or sharing. The downloaded format should be in line with the MITRE Data Format.

Summary of Changes

After a highly productive week, we seamlessly implemented the database structure into the backend system and thoughtfully populated it with some seeding data. All data entities were meticulously initialised with some formatted value to facilitate fundamental testing. After thorough testing, we unearthed some significant bugs that were promptly fixed to ensure smooth development. Notably, we optimised our data seeding process and database creation process to execute only once upon container launch using Docker, eliminating redundancy.

In parallel, we also fortified our security framework with connecting the frontend authentication to the backend, establishing a robust system that grants access exclusively to authorised users for designated routes. This enhances the overall security and reliability of our application, contributing to its seamless functionality.

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

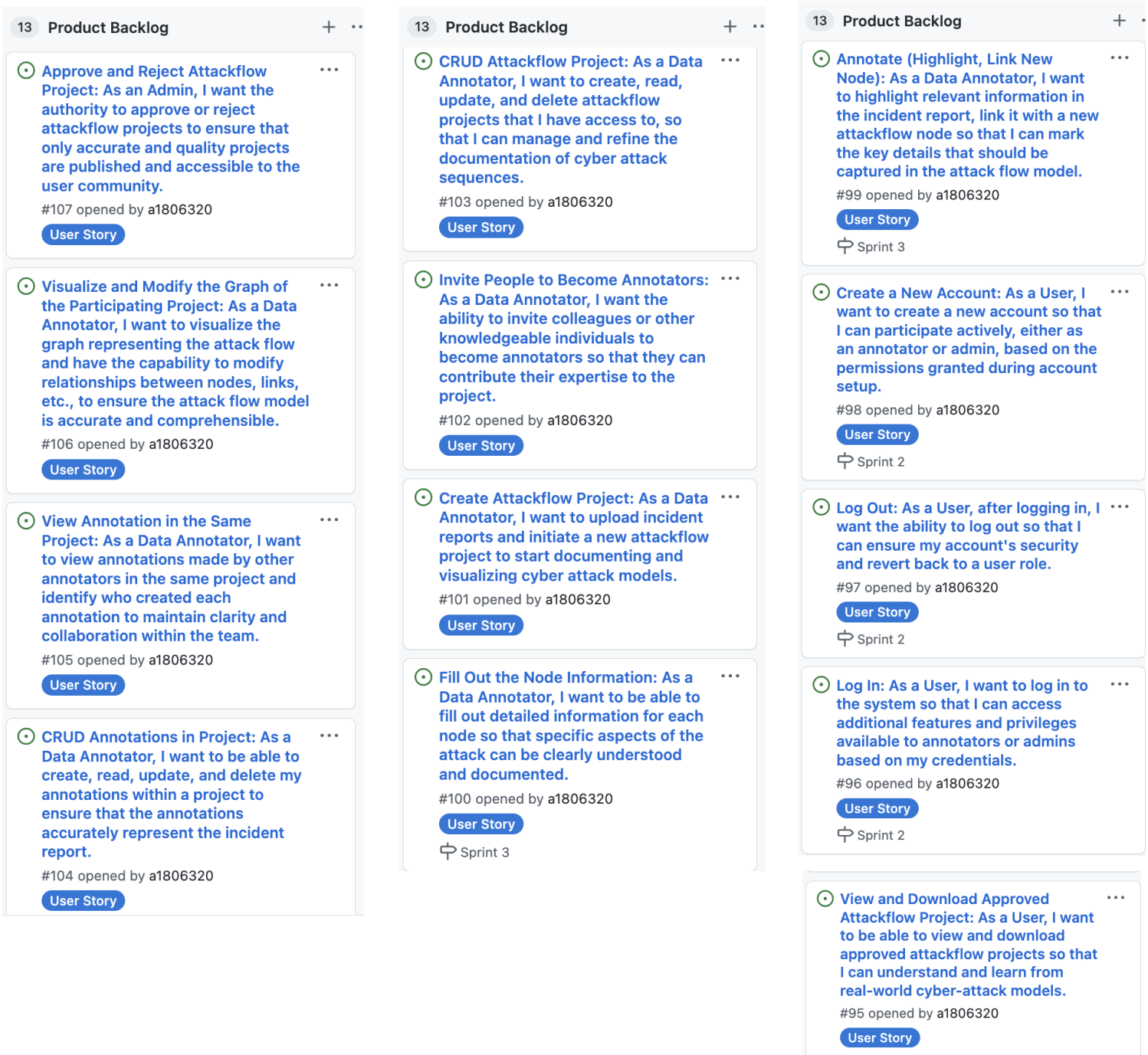


Image 1: Snapshot 3.1 Product Backlog

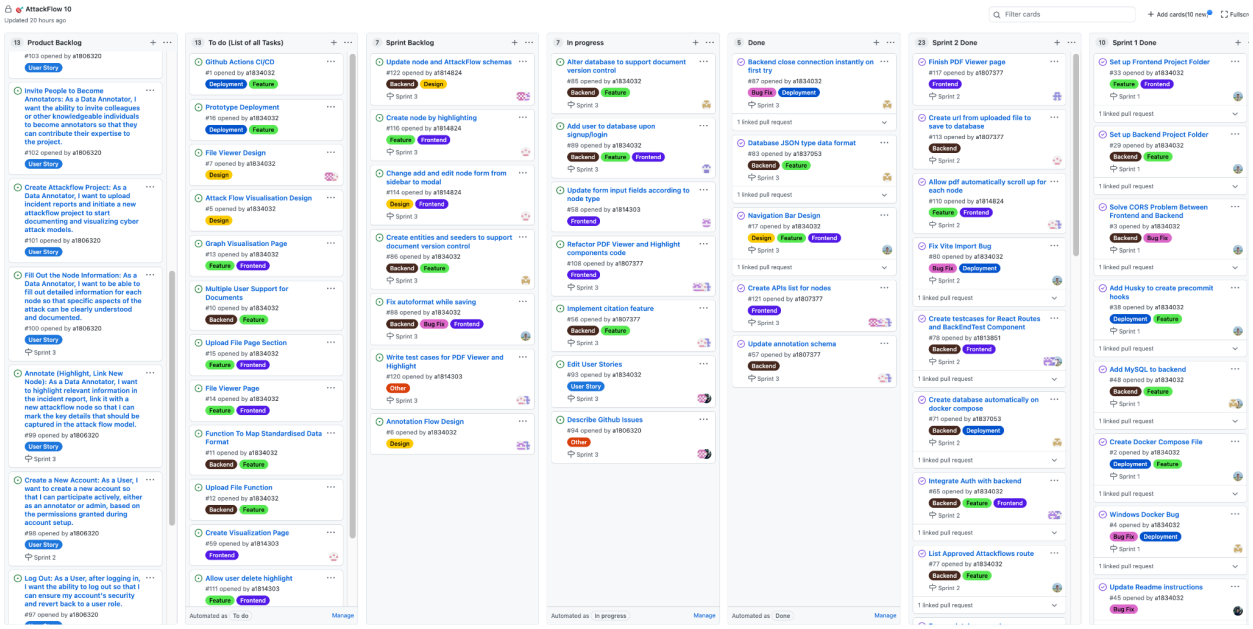


Image 2: Snapshot 3.1 Task Board

Sprint Backlog and User Stories

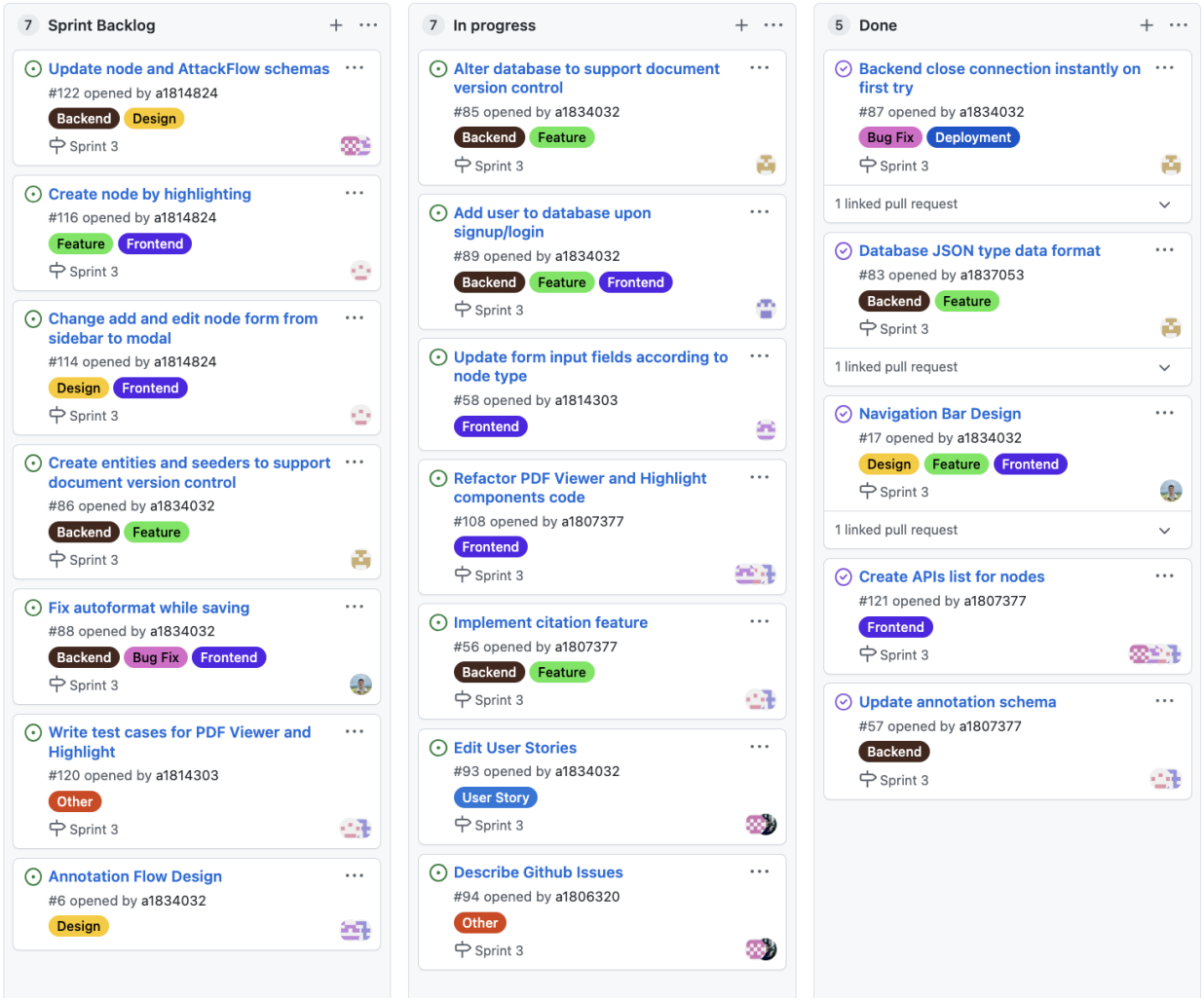


Image 3: Snapshot 3.1 Sprint Backlog

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

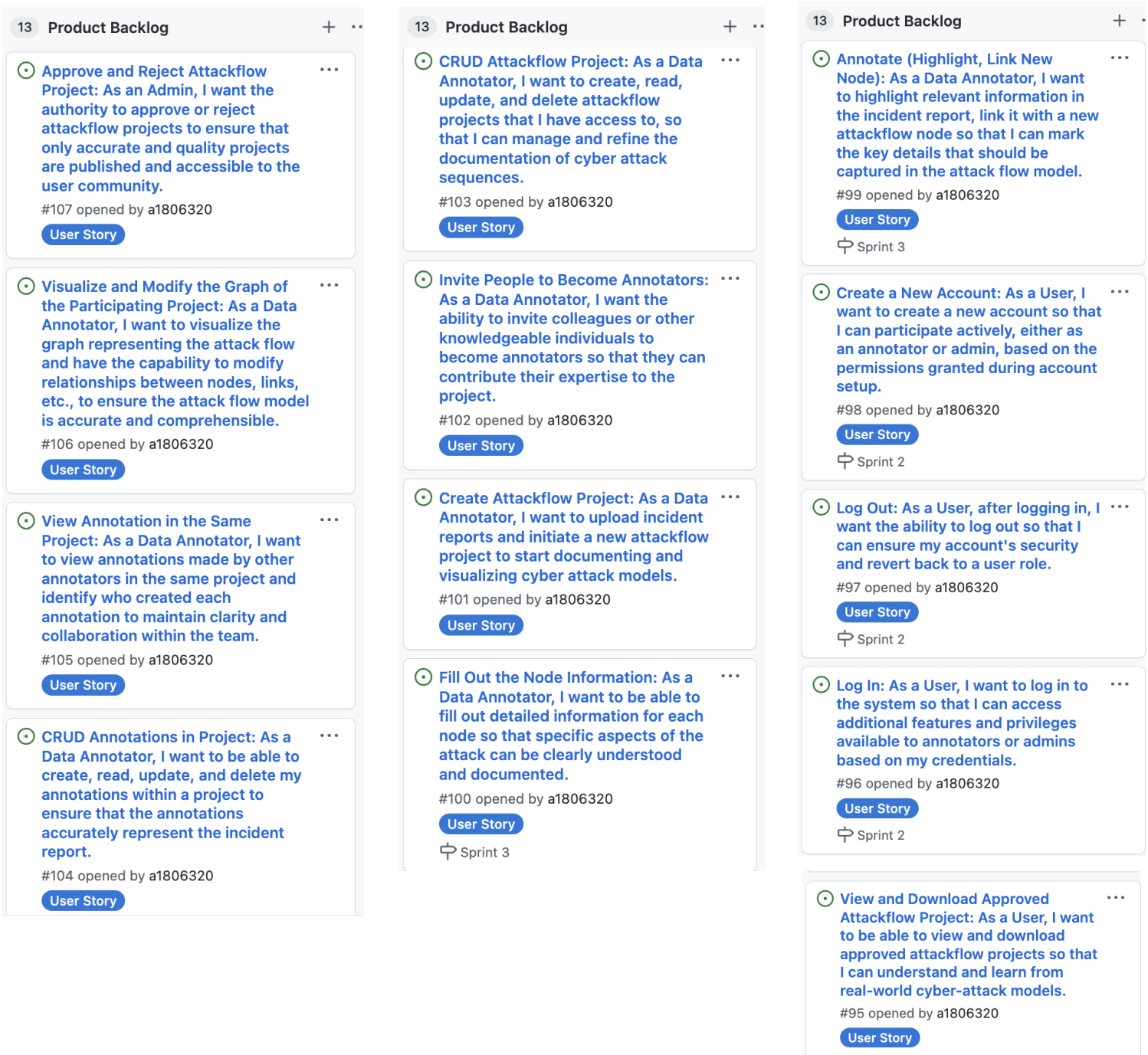


Image 1: Snapshot 3.2 Product Backlog

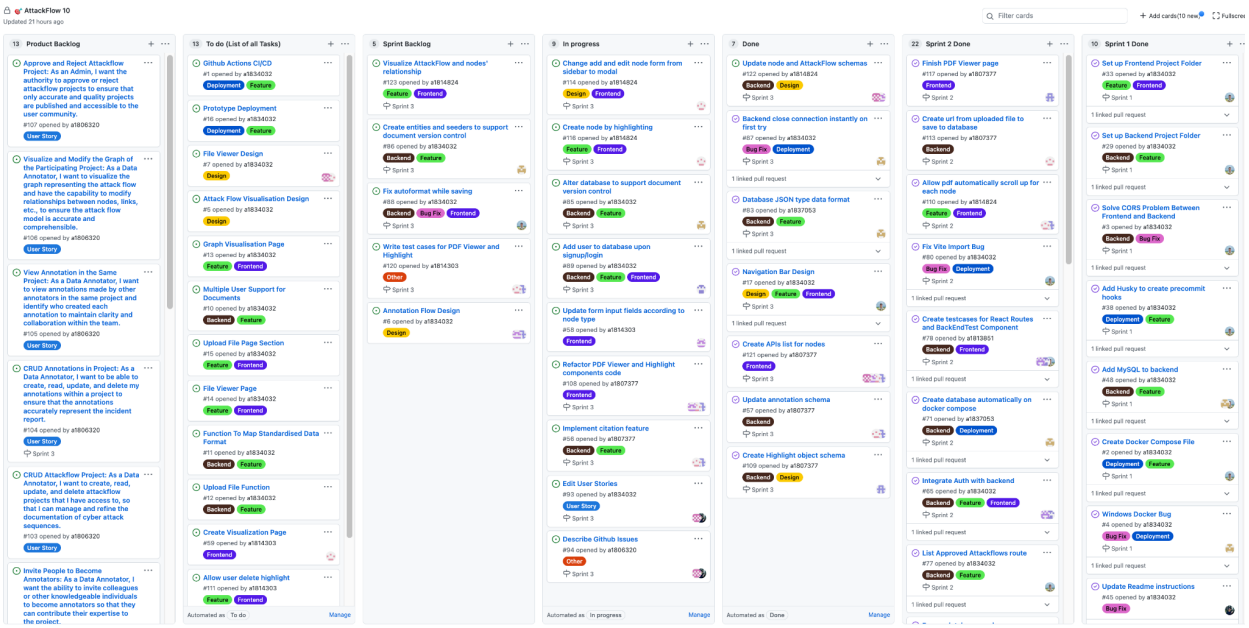


Image 2: Snapshot 3.2 Task Board

Sprint Backlog and User Stories

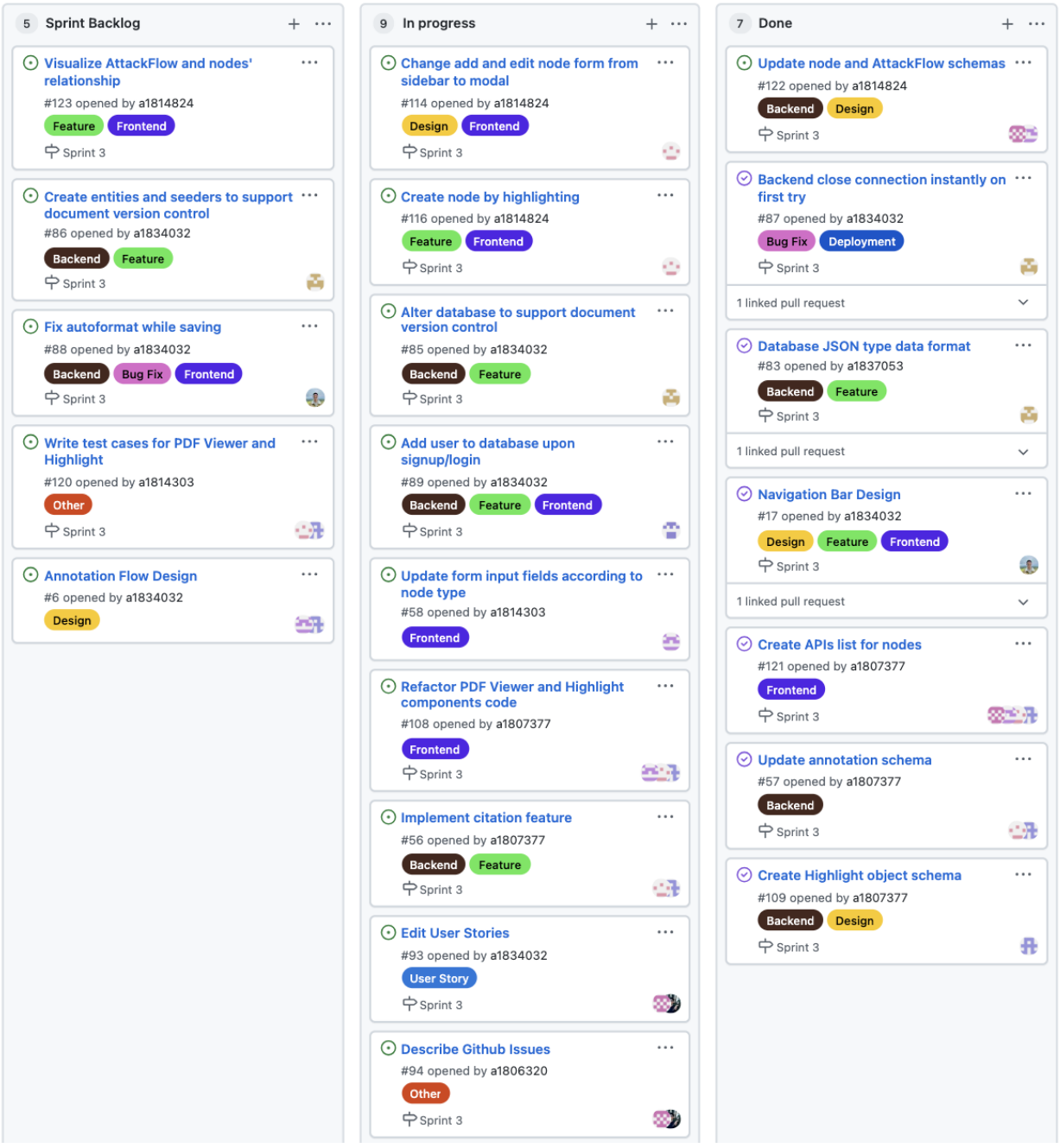


Image 3: Snapshot 3.2 Sprint Backlog

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.

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

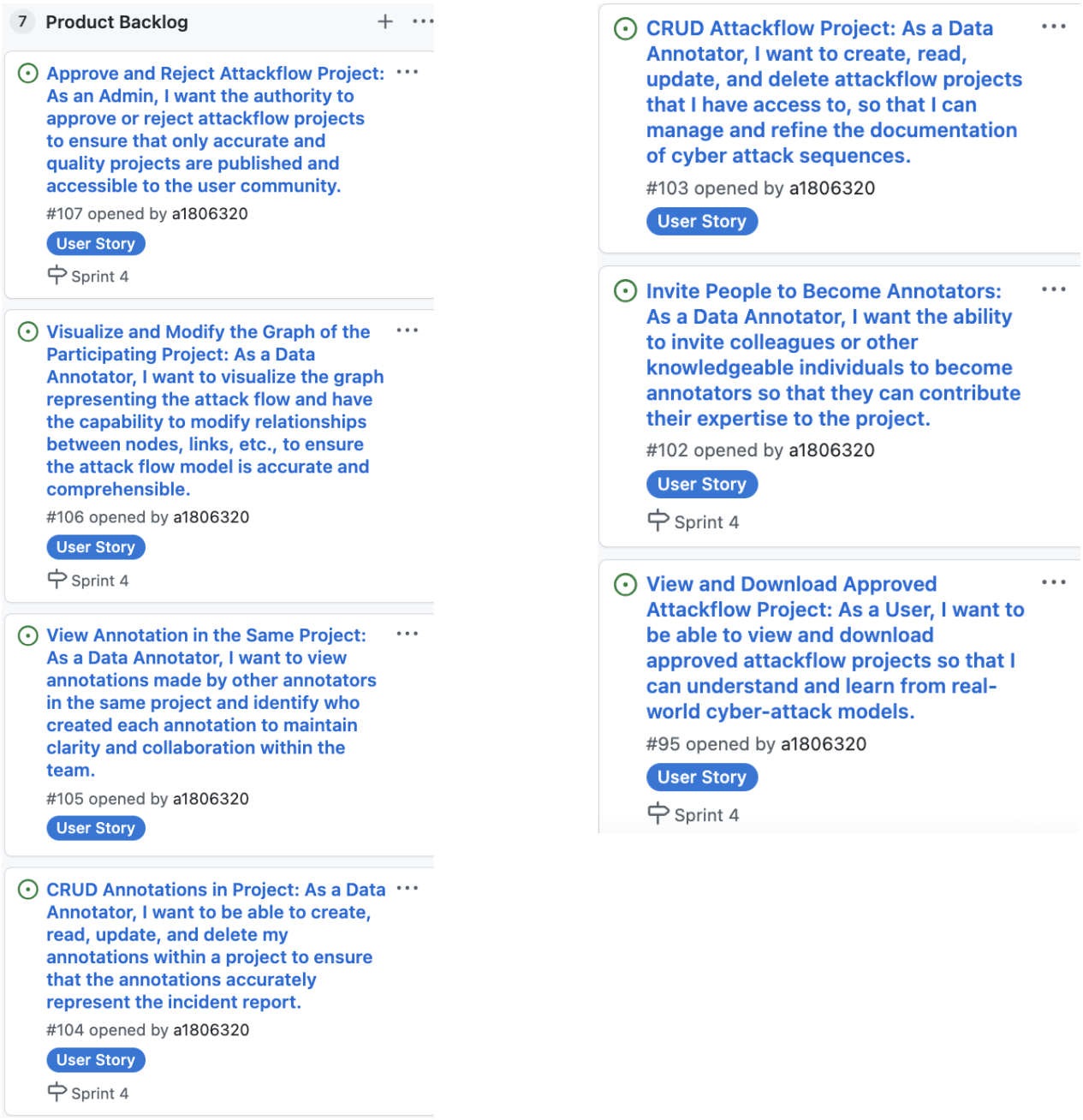


Image 1: Snapshot 4.1 Product Backlog

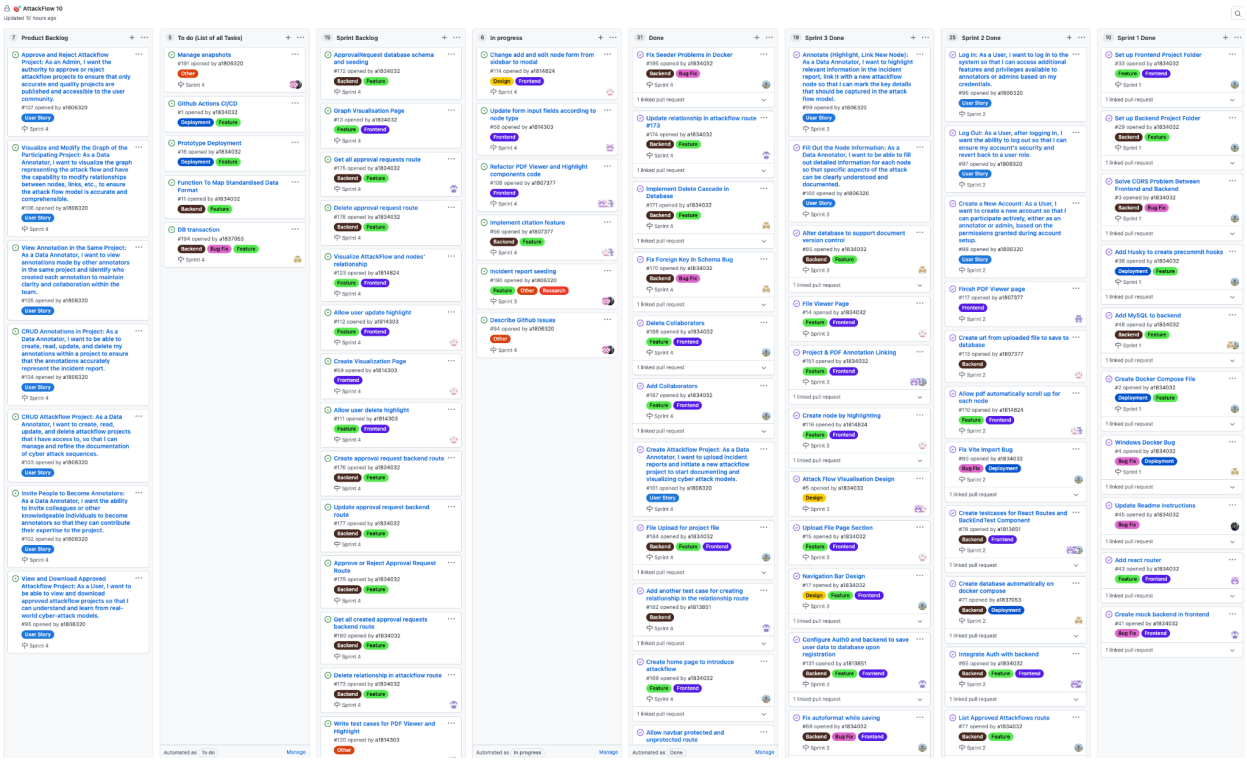


Image 2: Snapshot 4.1 Task Board

Sprint Backlog and User Stories

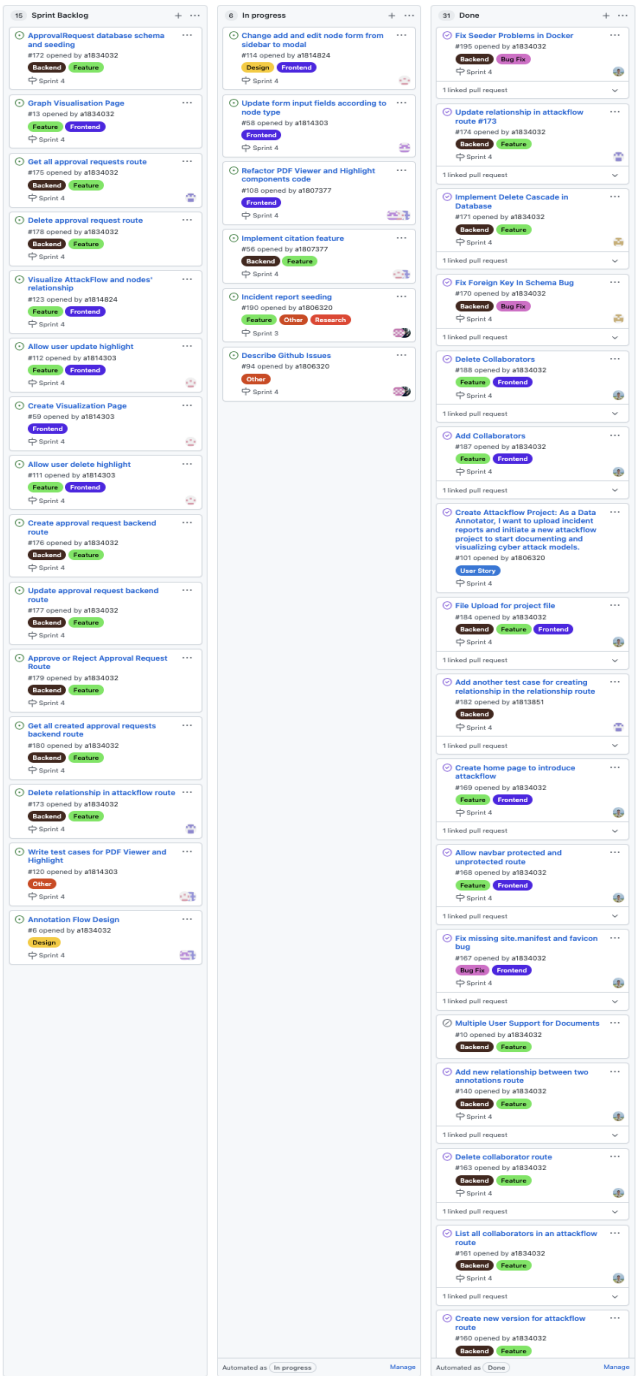


Image 3: Snapshot 4.1 Sprint Backlog

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

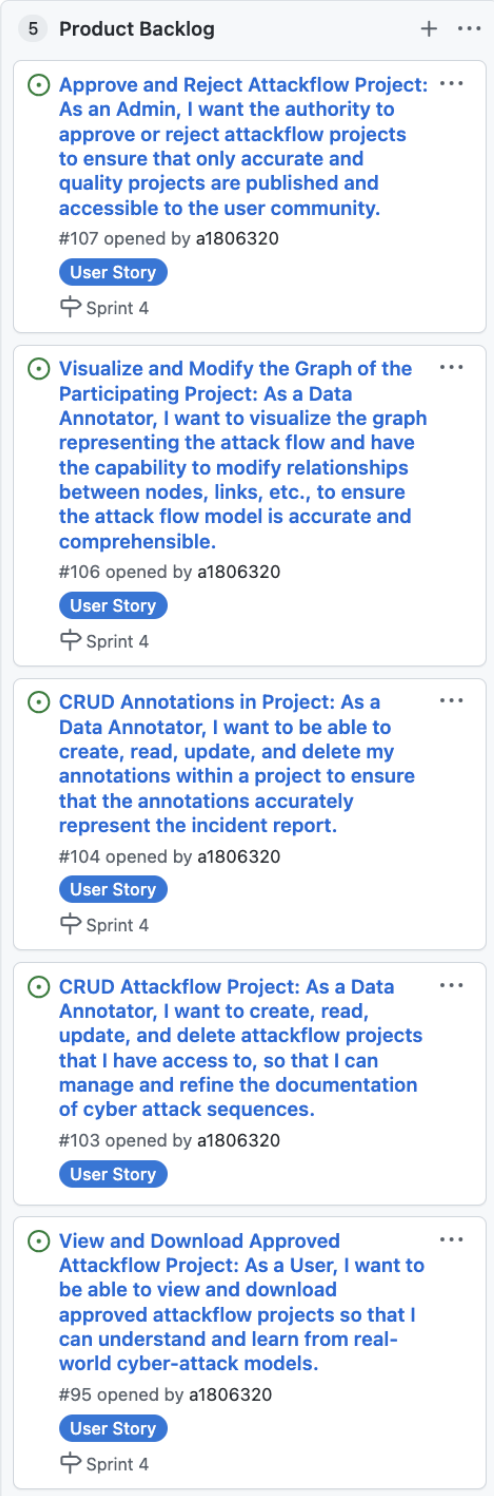


Image 1: Snapshot 4.2 Product Backlog

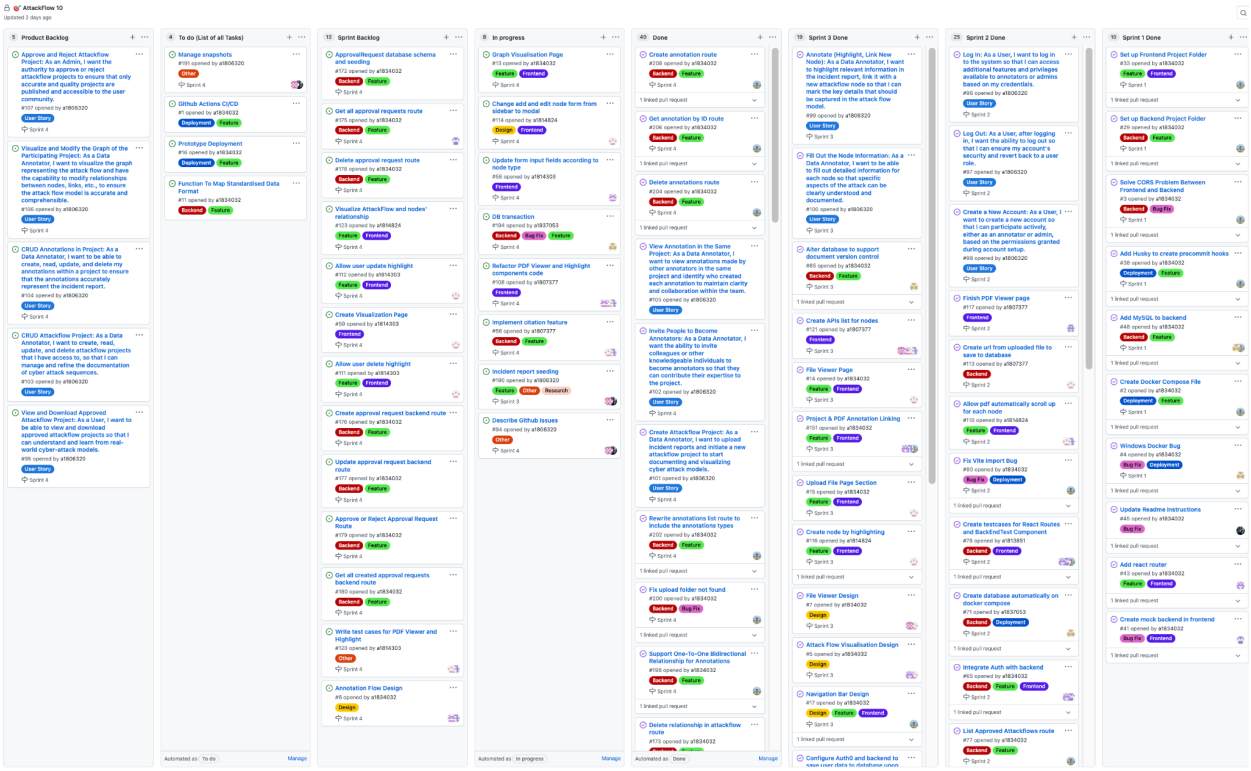


Image 2: Snapshot 4.2 Task Board

Sprint Backlog and User Stories

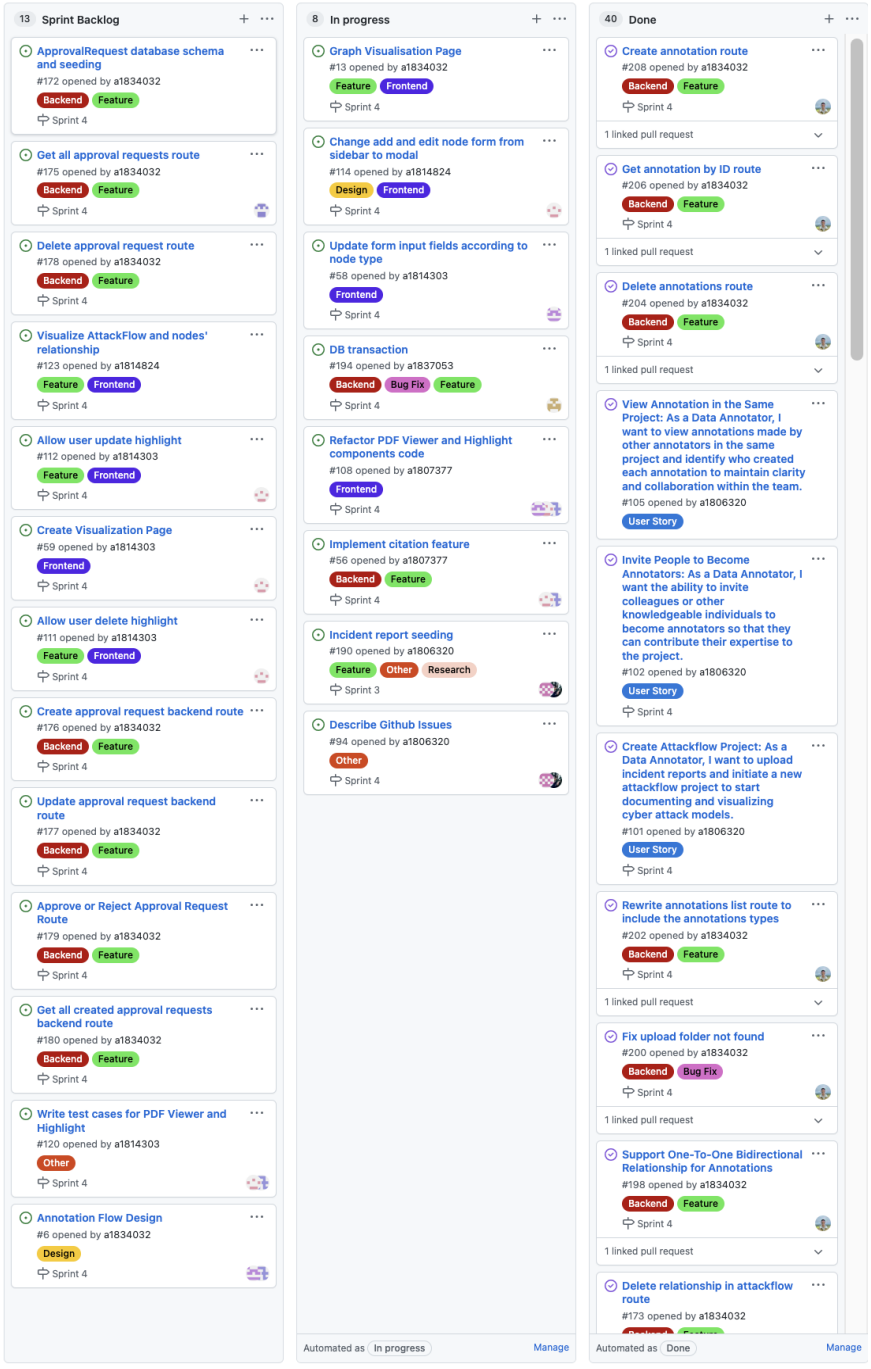


Image 3: Snapshot 4.2 Sprint Backlog

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.

Attack Flow

Snapshot Week 11 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

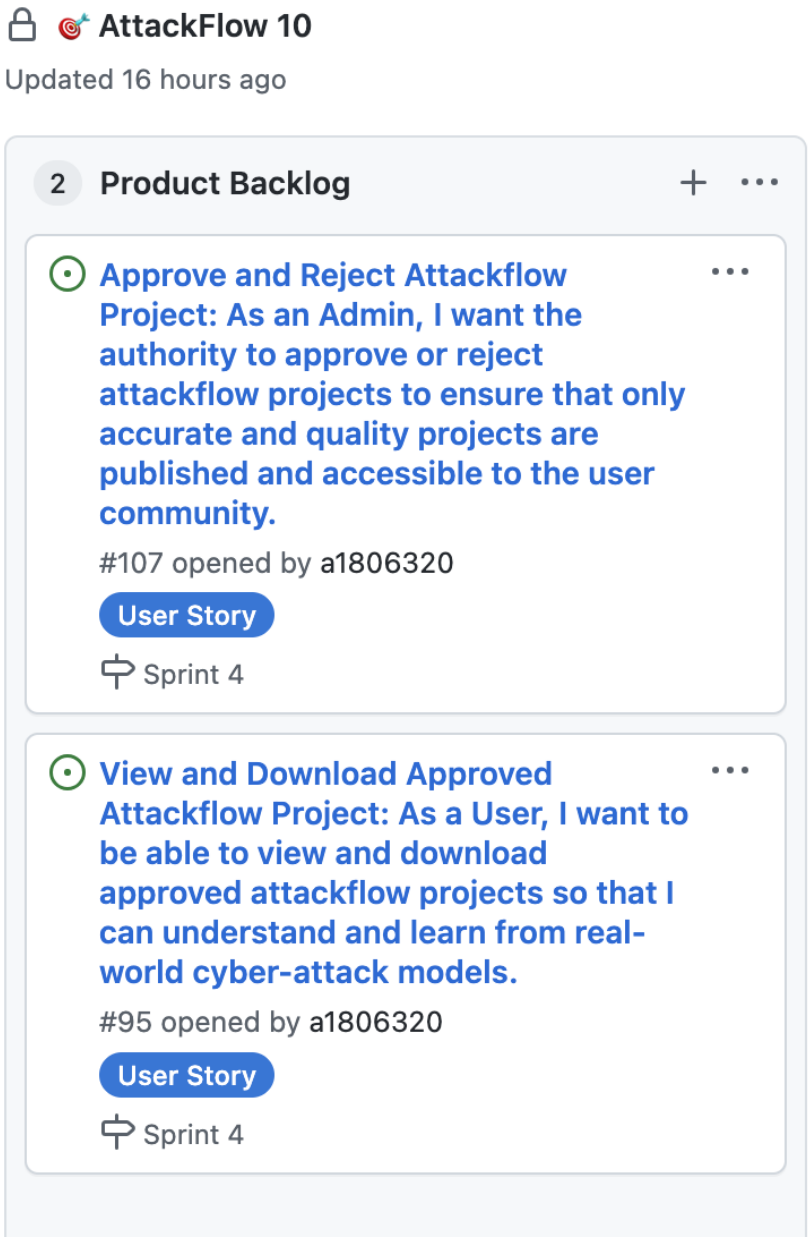


Image 1: Snapshot 5.1 Product Backlog

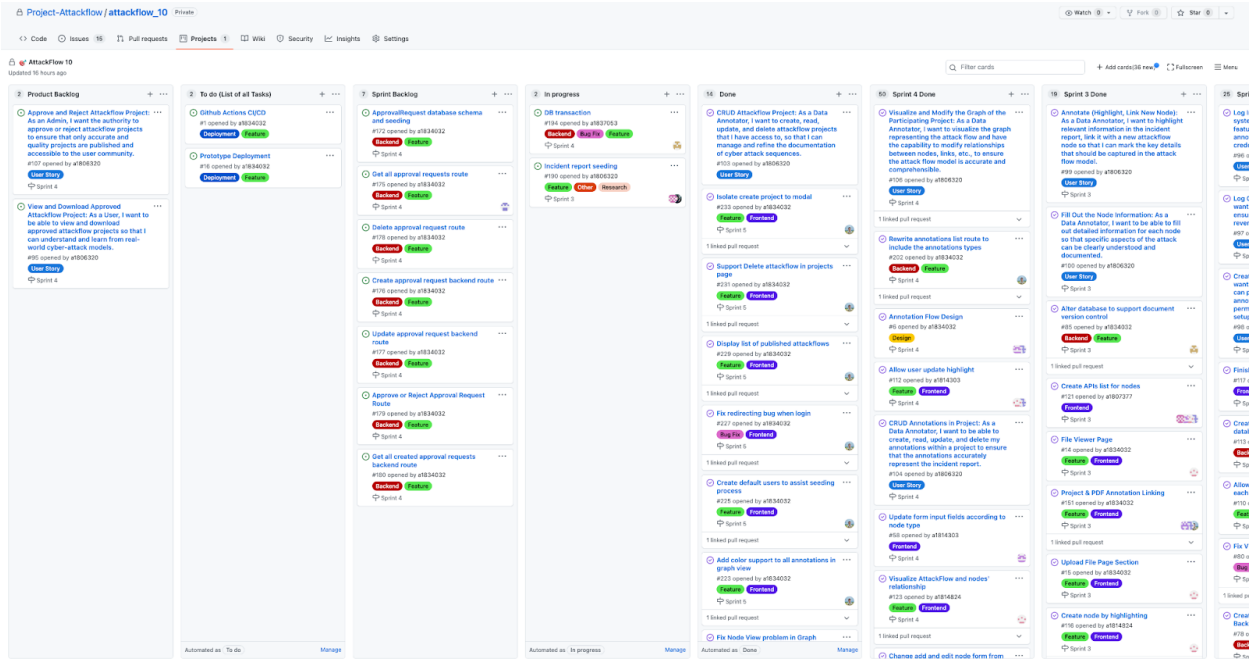


Image 2: Snapshot 5.1 Task Board

Sprint Backlog and User Stories

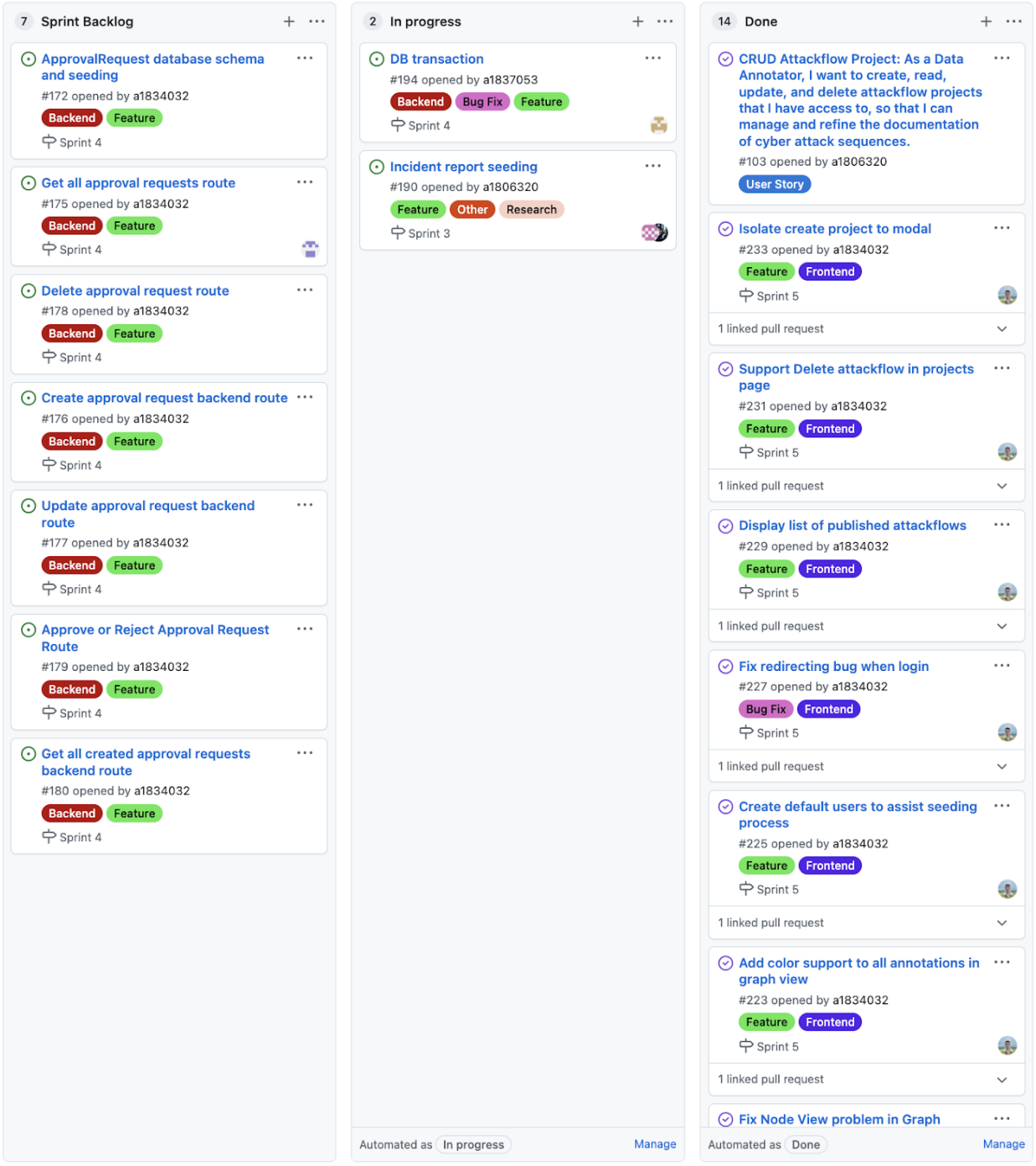


Image 3: Snapshot 5.1 Sprint Backlog

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. 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 eleventh 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

As we are approaching the end of the semester, our attackflow project has undergone significant enhancements, both technically and non-technically. Here's an organised breakdown of the week's progress:

Database Enhancements: We have fortified our project's backbone by seeding our database specifically for incident reports, paving the way for reliable data representation and handling.

Graph Visualisation: Building upon our prior front-end accomplishments, the graph visualisation has now reached its culmination. It is fully integrated with the backend, thereby providing users with a seamless and interactive visual experience. Moreover, minor node view bugs within the graph have been addressed, ensuring the presentation is glitch-free.

Annotations Overhaul: This week our team has a comprehensive update in the annotations segment:

- Introduced support for all 17 annotation types in the form.
- Implemented a feature to list all annotations, with each one being colour-coded based on the annotating user.
- Extended colour support to all 17 annotation types.
- Fixed a pivotal bug to ensure annotations and their respective highlights are immediately displayed as the file is accessed.

User Interface Upgrades: Our interface has become more intuitive with the addition of features that list annotations in distinct colours, simplifying user navigation.

Final Touches and Presentation Preparations: With our software nearing its final form, we have delved deep into testing the attackflow application for robustness. Alongside these final

refinements, we're gearing up for our conclusive presentation, ensuring we provide our clients with a coherent and insightful overview of our endeavours.

In essence, the week's alterations and advancements reflect our unwavering commitment to presenting a refined, user-friendly product, underlined by robust technical foundations.

Attack Flow

Snapshot Week 12 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

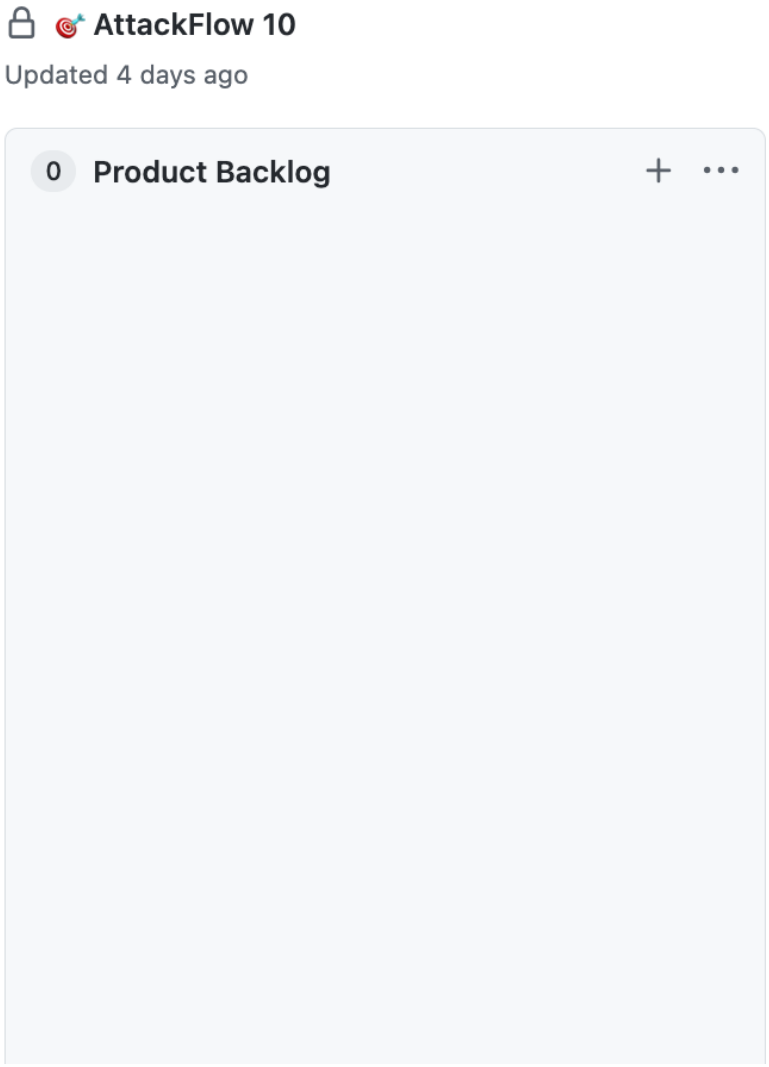


Image 1: Snapshot 5.2 Product Backlog

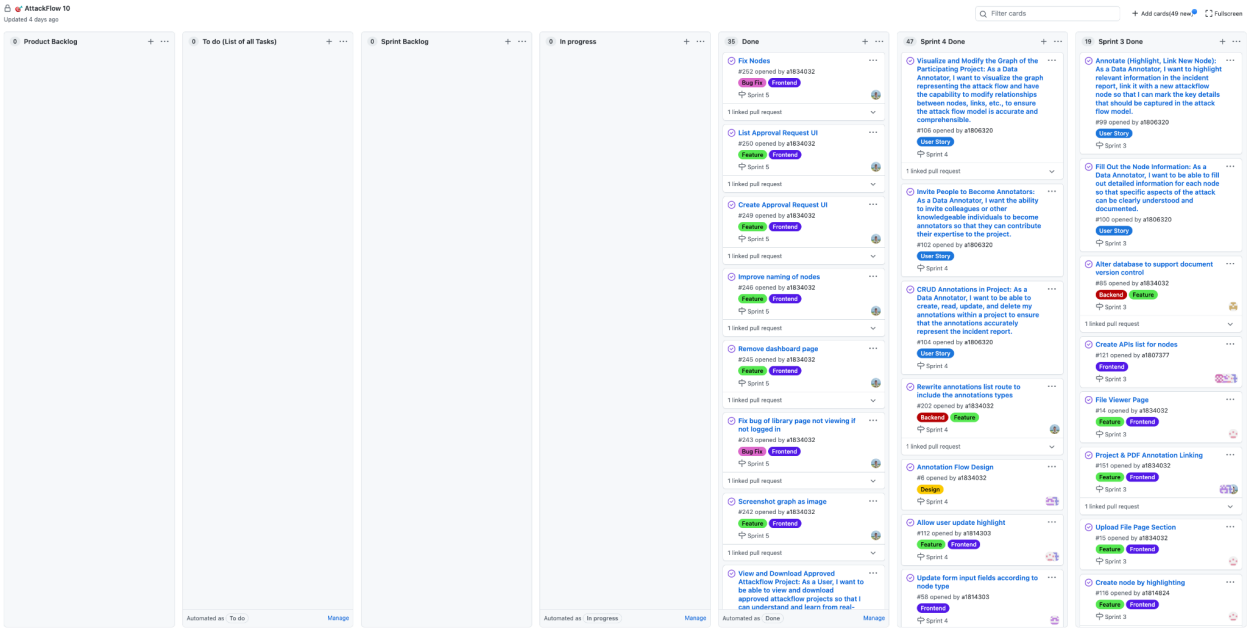


Image 2: Snapshot 5.2 Task Board

Sprint Backlog and User Stories

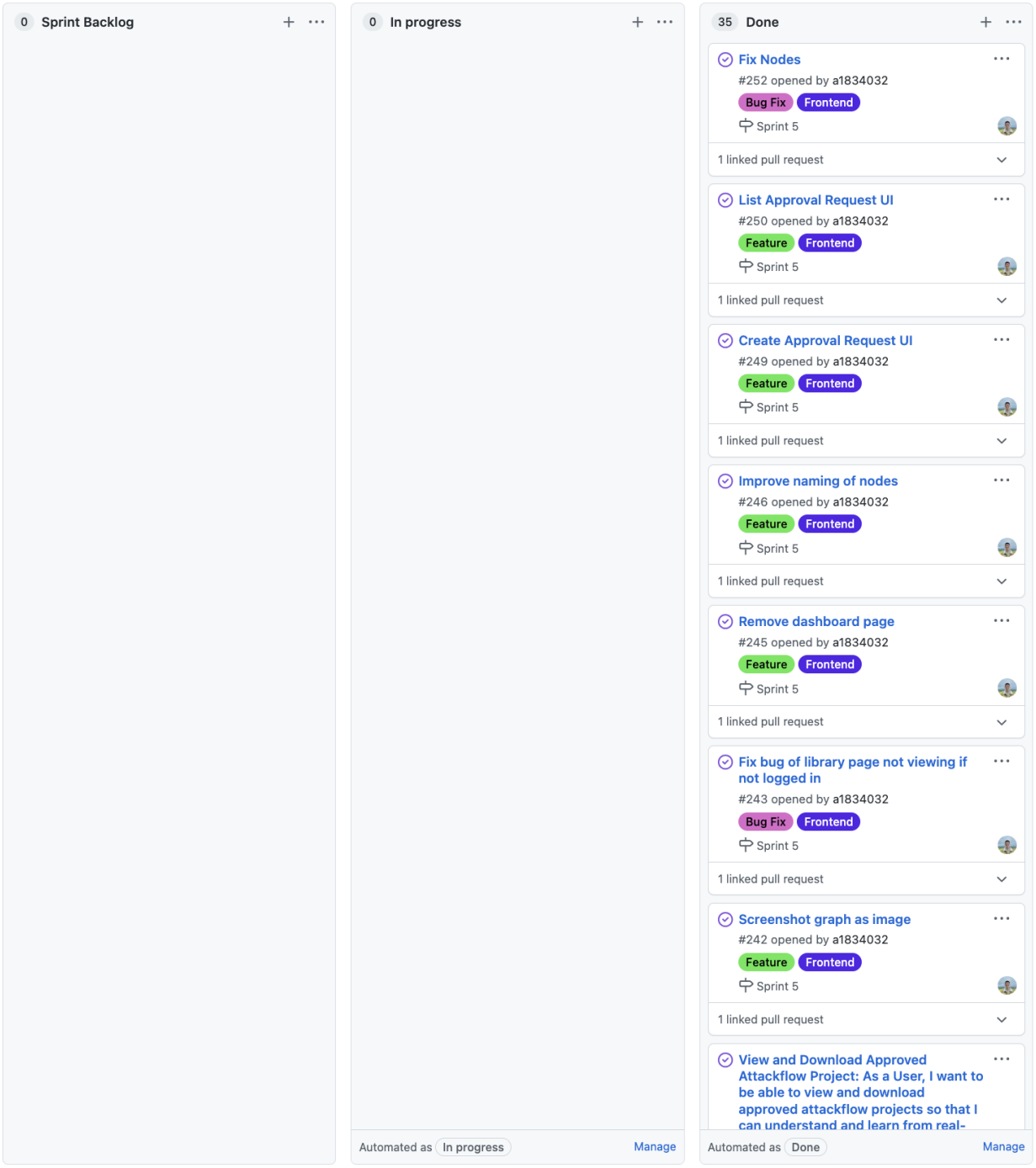


Image 3: Snapshot 5.2 Sprint Backlog
This screenshot depicts the three distinct stages of our sprint backlog: tasks queued in the backlog (Sprint Backlog), tasks currently underway (In progress), and tasks that have been successfully completed (Done).

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 eleventh 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

As we near the semester's culmination, the Attackflow project has witnessed substantial developments and refinements. Here's a recap of our notable progressions this week:

Deployment and Accessibility: Our Attackflow project is now live and accessible to everyone via the URL: https://attackflow.jasonbeh.com/. This deployment marks a pivotal moment, bridging the gap between our development efforts and the end-users.

Incident Report Testing: We have seeded and rigorously tested two incident reports, ensuring that they are available for public view. This enhances our database's content, ensuring users have meaningful data to interact with upon accessing our platform.

User Interface and Visual Features:

- **Graph Visualisation:** An integrated and interactive graphical representation is now in place, linking the front-end with the back-end. Furthermore, any minor discrepancies previously found in node view have been resolved.
- **Annotations:** Our annotations feature has seen a significant overhaul. All 17 annotation types are now supported in the form. Moreover, we have introduced a feature to display a list of annotations, each differentiated by colour based on the annotating user. A crucial bug, which hindered the immediate display of annotations upon file access, has been rectified.

Presentation and Documentation:

- **Slide Preparation:** We have crafted our slides meticulously for both the final product presentation and the process presentation, ensuring we encapsulate our journey and achievements coherently.
- **Video Documentations:** Each team member has successfully completed their video recordings, capturing individual perspectives and contributions.

In conclusion, this week's transitions and enhancements signify our dedication to delivering a polished, user-centric product, underpinned by solid technical underpinnings.