OmniCAT Final Report

[GitHub][Live Application]

Lok Jian Ming A0236537Y

Ong Pei Cong Lester A0245751A

Shirshajit Sen Gupta A0244147H

1. About OmniCAT

OmniCAT is an omni-modal, collaborative annotation tool designed to simplify the data annotation process and alleviate the workload of subject matter experts. The application enables experts to delegate repetitive and time-consuming tasks to annotators with less specialised knowledge, such as students. Through a multi-stage pipeline, users progressively refine annotations with increasing levels of expertise, ultimately producing more precise and accurate datasets. Currently, OmniCAT specialises in 3D annotations, supporting any 3D model uploaded in the .obj format. It also integrates machine learning inference capabilities, allowing users to perform tasks such as segmentation on uploaded models using user-provided, pre-trained machine learning models and accompanying inference scripts.

2. Existing Applications

OmniCAT is the first tool to offer a multi-stage annotation workflow, designed to streamline the process of data annotation across various stages. Unlike other applications that are primarily built for collaborative annotation processes, our system is distinct in that it enables the structured progression of tasks through multiple stages, offering a more organised and systematic approach to annotation. While other tools may allow for simultaneous collaboration, they do not inherently provide the same level of sequential workflow management, which is a key differentiator of our platform. In addition, OmniCAT is also one of the few tools with a built-in machine learning inference pipeline. The following sections compare OmniCAT to some of the existing tools.

2.1 SuperAnnotate

SuperAnnotate is a platform designed for data annotation and model evaluation, catering to artificial intelligence (AI) pipelines in computer vision (CV), natural language processing (NLP), and generative AI tasks. It supports data connection from local and cloud storage with GCP, AWS, Azure, and Databricks integration, advanced annotation tools, and project management features. SuperAnnotate's platform is particularly valued for its high-quality annotations, ISO-27001 and HIPAA compliance, support for multiple use cases, and project management and LLM features for labelling datasets.

Pros

- **Efficiency with advanced annotation tools**: SuperAnnotate claims up to 60% efficiency improvements in reducing annotation cycle times, offering tools that automatically assist users with initial annotations.
- **Support for multiple file types**: SuperAnnotate supports many different kinds of inputs including images, videos and audio, spanning across multiple AI domains, e.g., robotics, aerial imagery, NLP, sports, insurance, and healthcare domains.
- **Security and Compliance**: Industry-standard compliance with the EU (GDPR), medical (HIPAA), SOC 2 Type II, and ISO 27001.

Cons

- **Cost**: The platform's pricing may be prohibitive for small teams or individual users, especially when handling large datasets or requiring frequent annotations.
- **Annotation Workforce Dependency**: Despite the availability of trained workforce, outsourcing annotations may introduce variability in quality or timelines depending on the annotation team's expertise and workload.
- **No support for 3D files**: SuperAnnotate does not support annotations on 3D files.
- **No multi-stage workflow**: SuperAnnotate does not support a multi-stage workflow.

2.2 3D Slicer

3D Slicer is a free, open-source software for visualisation, processing, segmentation, registration, and analysis of medical, biomedical, and other 3D images and meshes; and planning and navigating image-guided procedures. It is widely used in research, clinical applications, and education. The platform supports multi-modal imaging, segmentation, registration, and 3D reconstruction, making it a powerful tool for healthcare professionals, researchers, and developers.

Pros

- **Multi-modal Data Support**: Offers tools to create and manipulate 2D images and 3D models for surgical planning, anatomical studies, and more, including DICOM files.
- **Image Segmentation Capabilities**: 3D Slicer provides automatic AI-assisted segmentation, and a built-in segment editor.
- **Free and Open Source**: 3D Slicer is free and open-source, and has an active community on the forum for users to seek help.

Cons

- **Specialised Expertise Required**: To fully leverage advanced modules such as SlicerRT, users often need a solid understanding of concepts like DICOM, dose-volume histograms, and image registration, which may not be common knowledge.
- **Limited Documentation and Support**: While 3D Slicer has a wiki and user forums, users may find the documentation insufficient for troubleshooting specific issues. This can be problematic when dealing with advanced features or custom workflows.
- **Steep Learning Curve**: While 3D Slicer is powerful, its extensive feature set can make it overwhelming for beginners, particularly those without prior experience in medical imaging software. The interface, though functional, is not always intuitive for new users.
- **No multi-stage workflow**: 3D Slicer does not support a multi-stage workflow.

2.3 Labelbox

Labelbox is a platform designed to streamline the process of creating training data for machine learning models. It provides tools for data annotation, management, and iteration, making it a comprehensive solution for labelling workflows. With features like model-assisted labelling, active learning, and API integration, Labelbox helps reduce the time and cost associated with preparing datasets. It offers collaboration tools and quality review mechanisms, enabling teams to work efficiently on labelling projects across domains like healthcare, retail, and robotics.

Pros

- **Multi-modal Data Support**: Natively supports image, video, text, PDF document, tiled geospatial, medical imagery such as DICOM, and audio data.
- **Collaboration-Friendly Design**: Multi-step workflows with expert AI trainers, AI feedback and automated quality controls to produce quality data at scale
- **Active Learning & Model Feedback**: Improves the annotation process by iterating on AI-generated pre-labels.

Cons

- **Complex Interface**: Despite its positive reviews, some users have expressed difficulty in navigating through Labelbox's interface. These users believe there is room for improvement in making the UI more approachable.
- **Lacklustre Support**: A common grievance is the delay in receiving technical support from Labelbox. Additionally, some users pointed out occasional maintenance disruptions, emphasising the need for enhanced technical assistance and reduced downtime.

- **Cost**: Labelbox's pricing has been a point of contention. Some users, especially from startups and smaller enterprises, find it a bit steep.
- **No multi-stage workflow**: Labelbox does not support a multi-stage workflow.

2.4 V7 Darwin

V7 Darwin is a suite of annotation tools which supports workflows with conditional logic, real-time collaboration, and a consensus review system to ensure high data quality. V7 Darwin is particularly favoured for its intuitive interface, extensive API, and versatility across domains such as medical imaging, autonomous vehicles, and retail.

Pros

- **AI-Assisted Annotation**: It has a user-friendly annotation feature with auto-annotation capabilities which utilises deep learning models for automatic segmentation. This allows for the labelling of even complex datasets swiftly and accurately.
- **Specialised Annotation Tools**: The polygon tool, image manipulation options, copy annotations feature, model-assisted polygon annotation, and subtract and merge tools make data annotation efficient and precise.
- **Multi-modal Data Support**: Handles a wide variety of datasets, including 2D images, 3D point clouds, videos, and medical imaging data, making it versatile across industries.
- **Secure**: SOC 2 Type II and HIPAA compliant.

Cons

- **Pricing for Small Teams**: The platform's cost structure may not be affordable for small teams or early-stage startups with limited budgets.
- **Occasional Annotation Variability**: Despite strong tools for automation and quality control, some users report inconsistencies in annotations, particularly in edge cases or complex scenarios.
- **No multi-stage workflow**: Labelbox does not support a multi-stage workflow.

3. Review of Milestones and Timeline

Our final milestone timeline is as follows. The ~~redacted~~ points indicate features that were scrapped or changed.

Week	Tasks	Remarks
1 (7 Oct - 13 Oct)	1. Design UI/UX on Figma 2. Present proposed UI/UX redesigns to our project mentor 3. Backend architecture review 4. Implement frontend authentication UI 5. Unify authentication database	Tasks 1 to 3 were completed as scheduled. Tasks 4 to 5 spanned the first 2 weeks due to the complexity involved in role-based access control.
2 (14 Oct - 20 Oct)	1. Rebuild project management UI 2. Backend restructuring 3. User management feature	The tasks in this week were completed on time. AWS S3 implementation was moved to the following week, as we felt it made more sense to include it after we built the 3D annotation functionality.
3 (21 Oct - 27 Oct)	1. Task management and annotation UI improvements ~~2. Implement 2D image annotation~~ 3. Testing core annotation functionality 4. Conduct integration testing ~~5. Gather user feedback for 3D annotation~~ 6. AWS S3 migration for storage	Task 1, 3, 4 and 6 were completed as scheduled. The review of workflow with project mentor was deferred to next week to allow more time for completing the application's core features.
4 (28 Oct - 3 Nov)	1. Enhancements based on feedback 2. Review workflow with project mentor 3. Live deployment of frontend and backend 4. Model inference pipeline 5. Build CI/CD pipeline	Task 1 and 4 were completed as scheduled. Task 3 was completed over week 4 and 5 due to challenges with running the provided inference scripts.

5 (4 Nov - 10 Nov)	1. Additional 3D annotation user testing after enhancements made
	2. AWS-backed model inference pipeline
	3. User testing
	4. ~~2D annotation user testing~~ Feature flaw fixes responding to user testing findings
	5. ~~Stretch goals: Work on optional features, such as:~~
	   ~~a. Identifying a better 3D visualisation library~~
	   ~~b. Expanding annotation tools beyond brushes~~
	   ~~c. Enabling local hosting of annotated datasets for users requiring on-premises storage due to data privacy concerns~~
	   ~~d. Real-time annotations~~
		Tasks 1 and 2 were completed as scheduled.
		
		The original Task 4 was removed as 2D annotations were not implemented on time. The updated Task 4 was completed as planned.

6 (11 Nov - 17 Nov)	1. ~~Second round of 2D annotation user testing~~ Complete basic implementation of project staging editing feature 2. STEPS marketing and presentation 3. Refine the pitch based on feedback from the final presentation and testing sessions	The original Task 1 was removed since 2D annotations were not implemented. The updated Task 1, along with Tasks 2 and 3, were completed as planned.
7 (18 Nov - 22 Nov)	1. Final report: Complete final report to be submitted	All tasks were completed as scheduled.

Table 1. Table of milestones and timelines

Despite delays in most of our initial milestones due to the complexity of the project, unclear project requirements and unfortunate decisions we had to make such as scraping the previous codebase, all core functionality features of the product were successfully delivered, including 3D annotations, dataset uploading, AI model and inference script uploads, on-demand machine learning inference, and comprehensive project and task management.

After navigating the complexities of task management functionality and the challenges of 3D modelling on a web app, the team decided to concentrate on delivering a viable standalone product. As a result, we chose not to pursue the 2D annotation as well as the other stretch goals within the 6-week development timeline. Instead, we choose to allocate the remaining time upon finishing a viable standalone product to debug and refine the existing features of our product with the aim of enhancing the overall user experience.

User testing was also postponed to accommodate application readiness and scheduling. Despite the shift, we successfully addressed several feature flaws identified during testing towards the end of the timeline, further improving the product's functionality and usability.

4. Individual Contributions and Roles

4.1 Jian Ming

Jian Ming was a full-stack engineer on the team. His contributions include:

- Initial project setup with Turborepo, nginx and Docker
- Implemented Role-Based Access Control using Clerk for the frontend and backend
- Data fetching for frontend using React Query on all pages
- 3D annotations (core functionality)
- CI/CD pipeline with GitHub Actions
- Multi-stage pipeline
- Deployment on Digital Ocean
- STePS marketing efforts, including LinkedIn post and poster design

4.2 Lester

Lester was a UI/UX designer and frontend engineer on the team. His contributions include:

- Design of the application on Figma
- Implemented the general structure of all frontend pages
- 3D annotations (user interface, export functionality)
- Help page documentation
- STePS marketing efforts, including promotional video and poster design

4.3 Shirsho

Shirsho was a backend engineer on the team. His contributions include:

- Implemented the general structure of the backend
- AWS S3 setup for models and datasets

- AWS EC2 and Lambda for inference pipeline
- STePS marketing efforts, including the design of the poster and creation of slides for pitch.

4.4 Resources

- 3D annotations were built upon https://github.com/CadanoX/Mesh-Painter.
- Dr. Jeffry Hartanto provided us with the data, inference, and users for testing.
- Uncle Soo provided us with feedback and insights about the different milestones of our project, and free marketing via his LinkedIn.

5. Application Design

5.1 Use Case Diagrams

Due to the sensitive nature of patient data, our application implements role-based access control. There are two roles available in OmniCAT as follows:

- Admin: Manage all projects, users, datasets and models for an organisation
- User: Manage projects they own, and perform annotation tasks

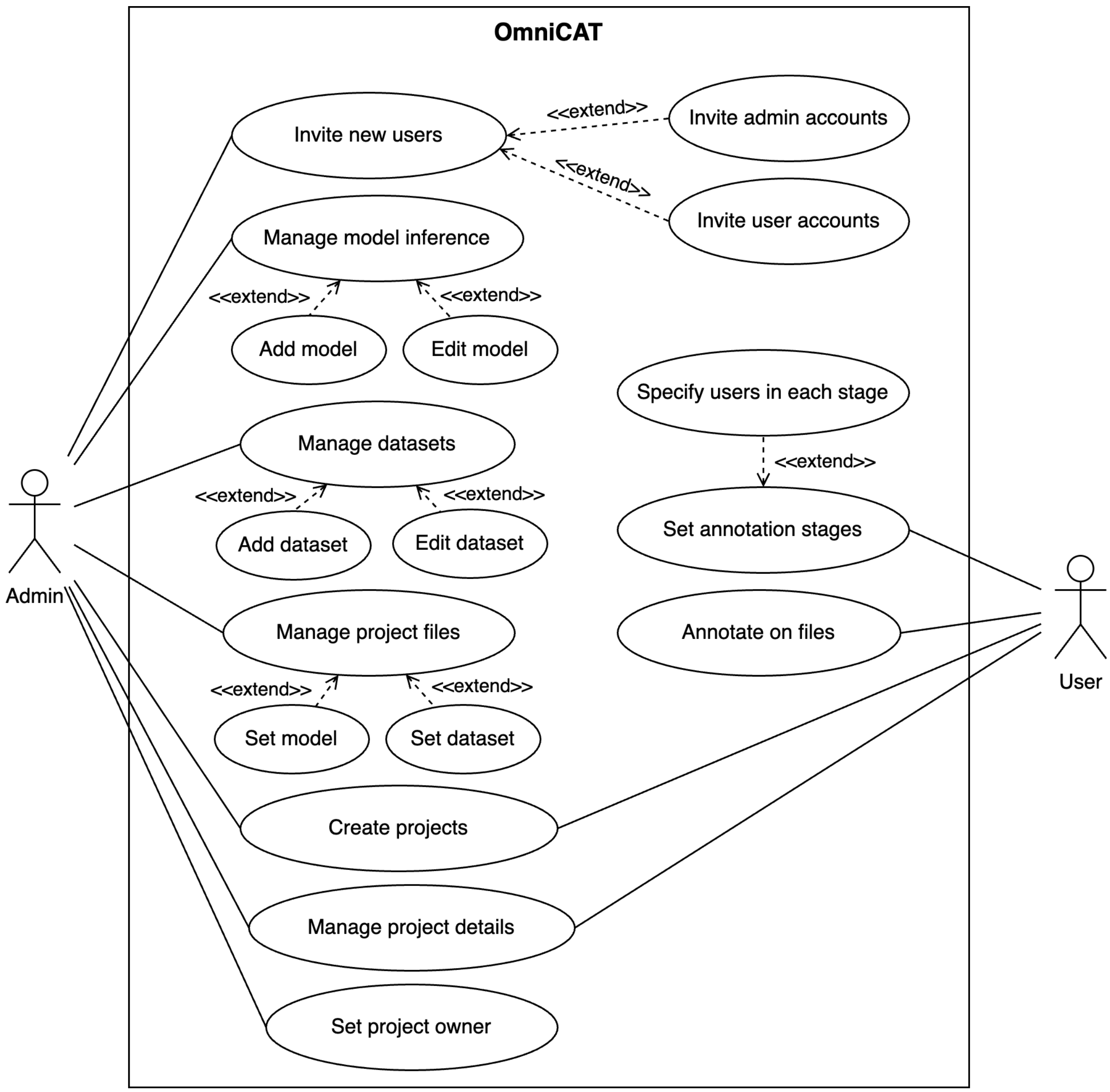

Figure 1. Use Case Diagram

5.2 User Flow

The user flow below outlines a typical multi-stage annotation process.

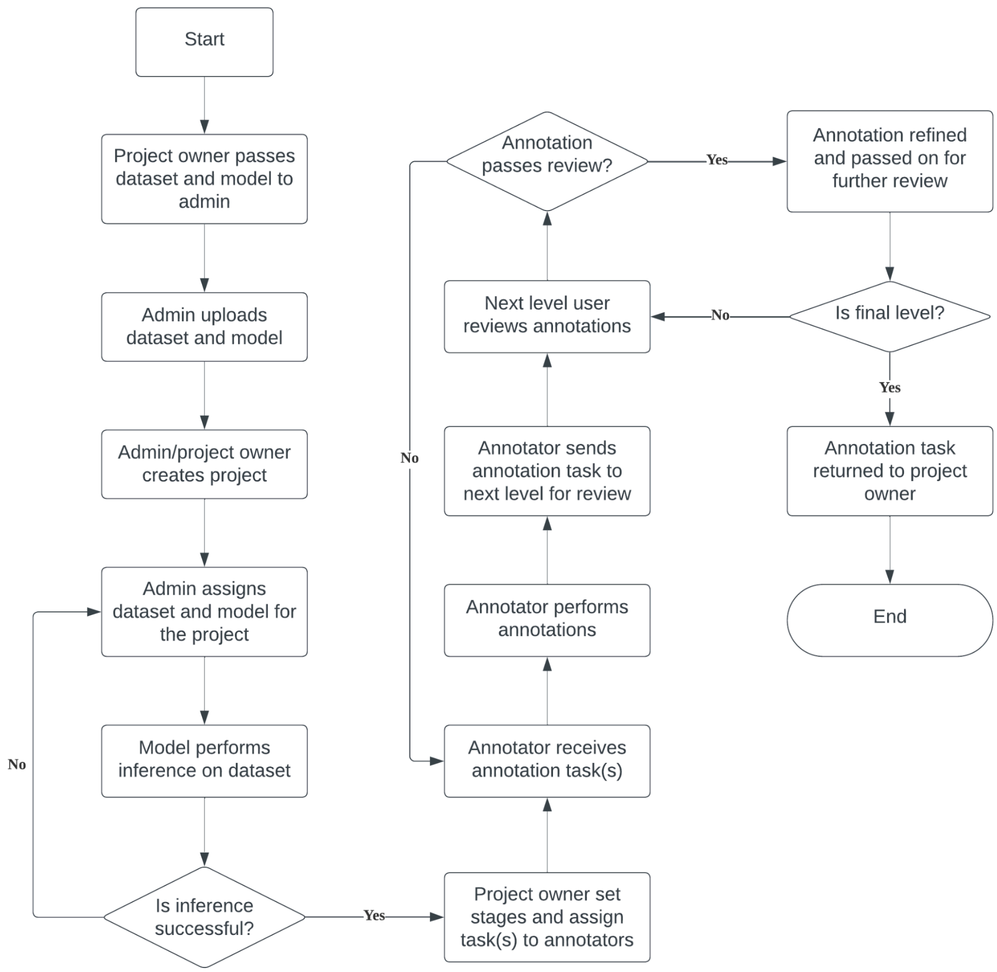

Figure 2. User Flow

5.3 Tech Stack

Our application is structured as a monorepo using Turborepo, where the code for the main application frontend, backend, and user documentation site is stored within one repository. The tech stack for each application is described below.

Main application frontend: React, Next.js, TailwindCSS and Three.js

- React, Three.js was used to maintain continuity with the frontend stack from the original source code, and also our team's familiarity with React.
- TailwindCSS was used to speed up the styling process.

Backend: Django and Python

- Django was used as we wanted to maintain continuity with the backend stack from the original source code. It also allows us to leverage on its powerful Object-Relational Mapping (ORM) capabilities to effectively manage database interactions, enabling seamless integration between our relational database and the application's data models. Django REST framework with function based views was also used which allows us to write simplified code for our different endpoints.

Documentation site: Docusaurus and React

- A documentation site was necessary to quickly create documentation using Markdown.
- A React-based framework was preferred for the documentation site for consistency with our main tech stack. Docusaurs was picked for its wider feature set in comparison to other options like Nextra.

Authentication: Clerk

- Authentication for the main application is implemented using Clerk for both the frontend and the backend. Role-based access control in our application is implemented through the inclusion of roles in the public metadata embedded as part of the JSON Web Token (JWT) returned for each user. We currently have two roles defined for our main application, admin and user.

Database and Storage: PostgreSQL and AWS S3

- PostgreSQL was chosen for its robust capabilities as a relational database, particularly in managing the complex relationships between projects, stages, tasks, and annotators within our application. Additionally, PostgreSQL's scalability and support for transactional integrity ensure data consistency and reliability across the application for better scalability.
- AWS S3 was chosen for the storage of models, datasets, and inference results to streamline our infrastructure and unify cloud services under a single provider. By leveraging AWS S3, we benefit from its highly scalable, durable, and cost-effective storage solutions, ensuring that large datasets and model files can be securely stored and accessed with minimal latency. S3's integration with other AWS services, such as AWS Lambda for automated processing also provide us with a reliable and flexible platform for managing data throughout its lifecycle.

Deployment: Caddy, Docker and DigitalOcean

- Caddy is used as a reverse proxy in our project to manage incoming traffic and efficiently route requests to the appropriate Docker containers. As a modern, high-performance web server, Caddy provides automatic HTTPS with its built-in support for SSL/TLS, ensuring secure connections without manual certificate management.
- Docker is utilised in our project to ensure environment consistency across development, testing, and production. By containerizing applications and their dependencies, Docker eliminates discrepancies between environments and simplifies deployment.
- DigitalOcean was chosen for its user-friendly interface and its ability to facilitate rapid deployment through our CI/CD pipeline via GitHub Actions. The platform's intuitive dashboard makes it easy to manage infrastructure, allowing our team to focus more on development rather than complex server configurations.

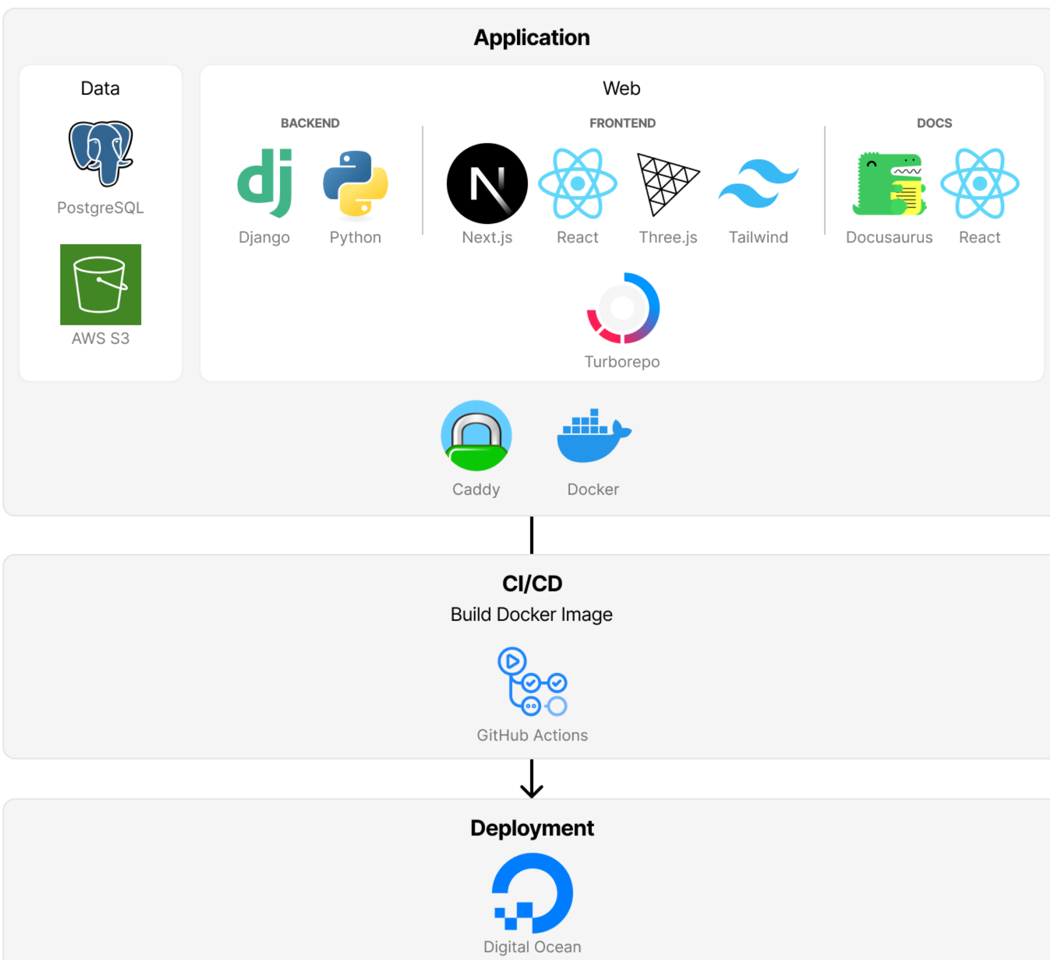

Figure 3. Tech Stack

5.4 Database Schema

The schema for our PostgreSQL database for our backend is shown below and can alternatively be accessed here. The key icon indicates the primary key for that table.

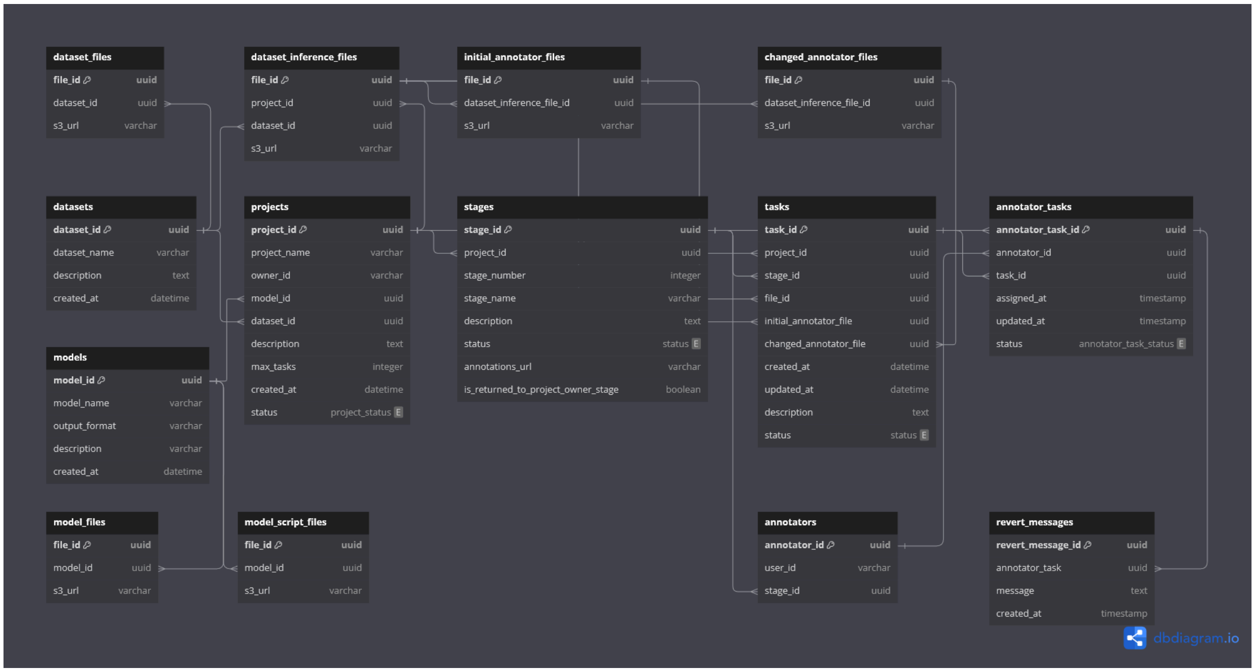

Figure 4. Database Schema

6. Report on Users

Since our application is still in the testing and development phase with the National Dental Centre of Singapore and our project mentor. As it has not yet been released to the public, we did not include any analytics tools in our platform at this stage. However, we conducted one round of user testing with 6 Oral Health Therapy students from Nanyang Polytechnic on 7th November 2024 at the National Cancer Center of Singapore.

Prior to user testing, we briefed the participants on the context of OmniCAT. We explained that the application's primary purpose is to serve as a multi-stage annotation tool, designed to support users with varying levels of expertise. We emphasised how the tool helps to streamline collaboration. We also encouraged them to draw comparisons with any existing tools they currently use and to provide honest and constructive feedback, ensuring we capture valuable insights that could inform future improvements.

During user testing, each participant conducted different tasks, depending on their role. There were 3 roles involved: project owner, first-stage annotator and second-stage annotator.

Project owner tasks

- Create a dataset
- Create a model
- Create a project with 2 stages, and 2 annotators in each stage
- View the completed annotation

First-stage annotator tasks

- Register for an account
- Annotate on an assigned task
- Send the task to the next stage (second stage annotator)

Second-stage annotator tasks

- Register for an account
- Return the task to the first stage annotator
- Annotate on an assigned task
- Send the task to the next stage (back to the project owner)

Our user testing indicates that OmniCAT is highly intuitive and user-friendly. The majority of participants were able to navigate and begin to use the application with little to no effort, demonstrating a quick learning curve. Most users were able to complete their tasks independently, without needing assistance or guidelines. A few participants even expressed positive feedback, highlighting the simplicity and ease of use as key strengths of the application.

However, several participants noted the lack of some features they were accustomed to in their current tools. While the app's simplicity and ease of use were appreciated, some users felt that certain capabilities were missing or underdeveloped, which could limit its appeal for those seeking more robust or specialised functionality. For example, their existing tools offered capabilities such as annotations automatically snapping to edges and more advanced annotation features such as bounding boxes. With the feedback received, we continued iterating on OmniCAT and developed our future plans and strategies.

7. Future Plans and Strategies

At the moment, we are still discussing with our project mentor, Dr. Jeffry Hartanto, about the future direction of the project. Below are some of the plans and strategies we are considering with his guidance.

A key focus for the future is enhancing the annotation capabilities of our application. This includes introducing additional 3D annotation tools and supporting more file types. Some of the 3D annotation tools we are considering include text annotations, bounding boxes and an eraser tool. We have plans to support more 3D file types such as .gltf, .stl, as well as other image file types such as .png, .jpg, .dcm[1] for 2D annotation functionality. Alongside these upgrades, we plan to refine the UI/UX and conduct extensive user testing to ensure these features integrate seamlessly without overcomplicating the application. By expanding and optimising our annotation tools, we aim to position OmniCAT as a versatile and user-friendly solution for a wide range of annotation tasks, ultimately attracting a broader user base and unlocking new potential in the data annotation tools market which continues to expand rapidly.

Another key focus is integrating pre-trained machine learning models for general inference tasks on user-uploaded datasets, such as the YOLO11 model for detection, segmentation, or classification tasks on 2D images. This feature would enable users without pre-trained models to leverage the platform's machine learning capabilities, significantly reducing setup time. Instead of training and uploading their own models, users could immediately begin the annotation process. Feedback from our STePS presentation indicated confusion among some audience members about the absence of such models, underscoring the importance of this feature in enhancing usability and meeting user expectations.

We also aim to enhance the collaborative aspect of our application by enabling multiple users to work on a single annotation task simultaneously. This feature would facilitate real-time collaboration, making OmniCAT suitable for handling more complex annotation tasks that require diverse expertise. For example, multiple users could annotate different aspects of a dataset, such as labelling objects, adding detailed classifications, or refining segmentations concurrently. This would not only accelerate the annotation process but also improve the quality and consistency of the resulting datasets. By supporting such advanced workflows, OmniCAT

[1] .dcm is the file type for DICOM files, commonly used in medical images.

could cater to specialised applications in fields like medical imaging, autonomous systems, and advanced research, further solidifying its position as a multi-functional and powerful annotation tool.

8. Insights Gained

One insight we gained was the complexity surrounding the technical skills needed to achieve the intended functionalities of the project. One notable example would be the usage of three.js for 3D rendering. While the library is powerful and capable of creating impressive interactive 3D content, its steep learning curve, limited support for TypeScript and React, and the limited availability of detailed documentation for advanced use cases presented substantial challenges. Extensive troubleshooting and experimentation were often required to suit our use cases. This experience highlighted the importance of carefully evaluating the tools we select for a project, considering not only their capabilities but also their usability, learning resources, and community support. It also underscored the hidden effort required to implement visually impressive features, which may not be immediately apparent to end-users or stakeholders. Moving forward, we recognize the value of balancing technical ambition with practicality, as well as the importance of allocating time for exploring alternatives and prototyping to ensure the chosen tools align with project requirements and team expertise.

Another valuable insight we gained during this project was the importance of evaluating the scale and quality of a project holistically. While functionality may appear seamless in a recorded demo, such demos often fail to reveal critical aspects like code quality, scalability, and maintainability. Upon gaining access to the previous codebase, we discovered significant issues, including poor documentation, disorganised structure, and inefficient implementations. These challenges rendered the codebase difficult to work with and ultimately led us to rebuild the entire project from scratch. This experience emphasised the need for rigorous code reviews and thorough technical assessments during project takeovers to avoid unforeseen obstacles. By occasionally referring to the previous implementation for reference, we were able to learn from its shortcomings and make informed decisions in designing a more robust and efficient solution.

Apart from that, we also gained a deeper appreciation for the importance of clarifying project requirements when working with a client. Clients may sometimes be uncertain about the functionalities they want in an application, making it essential not only to ask questions but to identify the right ones to ask. In our case, unclear requirements during the initial stages led to delays, highlighting the need for effective communication and proactive engagement to align on project goals. Additionally, given our limited timeline, it was crucial to accurately estimate the time required to implement features and to set boundaries when necessary. This included knowing when to say no, particularly when the client was unaware of the complexity involved in certain features. This taught us the importance of balancing client expectations with practical feasibility.

Additionally, we were reminded of the critical importance of conducting multiple iterations of user testing throughout the development process. Our decision to focus on annotating 3D models, while innovative, limited our pool of potential users for testing, as it required expertise or use cases in specialised domains. Compounding this challenge, the delays we faced during development left us with little time to conduct more user testing until the final stages of the project. Despite these constraints, the feedback we received underscored the significant benefits of early and iterative user engagement. Our decision to conduct user testing with Dr Jeffry using our Figma mockup early also allowed us to validate design choices and minimise the need for major overhauls late in the timeline. This experience highlighted the importance of prioritising user feedback as an integral part of the development cycle, even when time and resources are limited. This would help us balance innovation with usability and ensure that the application meets diverse user needs effectively.

We also came to realise the importance of effective project management and how it could have significantly improved our workflow. Due to limited manpower, no one formally assumed the role of a project manager. This lack of centralised coordination led to miscommunications occasionally which disrupted the development process, especially given that we underestimated the challenges we would face throughout the project. In hindsight, having a dedicated project manager could have greatly benefited our team by ensuring clearer task allocation, more accurate timeline estimations, and better alignment of goals. A project manager could have resolved bottlenecks proactively and maintained a clearer overview of the project timeline, especially given our tight deadlines. This role would have also streamlined communication among team members and ensured that decisions were made efficiently, ultimately enabling a more organised and focused development process. This experience underscored the critical role project management plays in driving team coordination and ensuring timely project delivery, particularly in resource-constrained environments.

Lastly, one final insight we gained during the final project was the importance of thoroughly testing not only the application's functionality but also its basic connectivity across different networks. While the School of Computing network allowed access to our domain, other parts of NUS's network flagged our newly registered domain as greyware, completely blocking connections to our website. This issue became evident during user testing at the National Cancer Centre Singapore with Dr Jeffry, where our laptops were unable to access our website, deployed at https://omnic.at when connected to Duke-NUS's WiFi. We resolved this by raising a support ticket through nTouch to NUS IT, which successfully unblocked our website. This experience highlighted the critical need to account for potential network restrictions when deploying applications, especially in institutional environments.

9. Appendix

9.1 Application Details

The user accounts below contain uploaded datasets and models and can be used to test the application. New projects can be created from these datasets and models. The repository is private, and can be found at https://github.com/omnicat-nus/omnicat. The live deployment of the application is found at https://omnic.at.

Username: admin-steps@omnic.at
Password: admin-steps

Username: projectadmin-steps@omnic.at
Password: projectadmin-steps

Username: stage1-steps@omnic.at
Password: stage1-steps

Username: stage2-steps@omnic.at
Password: stage2-steps

Username: stage3-steps@omnic.at
Password: stage3-steps

The model and dataset files are not provided since they are the property of the National Dental Centre of Singapore, i.e., they have been provided to us for this project by Dr. Jeffry Hartanto.

9.2 Customer Contact Reports

- Customer Contact Report 1 can be accessed here.
- Customer Contact Report 2 can be accessed here.
- Customer Contact Report 3 can be accessed here.