comp90082-2024-na-koala

Team

Team

Y Owned by yucpeng1 • Updated on 22 Mar 2024

Name Student ID Email Position Responsibility Bowen Fan 1035162 bffa@student.unimelb.edu.au Product Manager/Frontend Developer Leads project direction and user interface development, ensuring alignment with user needs and seamless backend integration. Tianqi Wang 1045939 tww2@student.unimelb.edu.au Scrum Master/ROS Analyst Facilitates Agile process

Confluence Open preview

Project Brief

Project Brief

Y Owned by yucpeng1 • Updated 5 days ago

🤓 Client Requirement https://chri-lab.notion.site/ROSAnnotator-A-Web-Application-for-ROSBag-Data-Analysis-in-Human-Robot-Interaction-e856d9fb941f4ceca0b4c53a77e71d03 📔 Overview & Background https://comp90082-2024-na-koala.atlassian.net/wiki/spaces/comp900822/pages/3735565 🎯 Requirement Analysis https://comp90082-2024-na-…

Confluence Open preview

Requirements & Design

Requirements & Design

Y Owned by yucpeng1 • Updated yesterday

🤓 Client Requirements https://chri-lab.notion.site/ROSAnnotator-A-Web-Application-for-ROSBag-Data-Analysis-in-Human-Robot-Interaction-e856d9fb941f4ceca0b4c53a77e71d03 \uD83D\uDDD2 Project Overview The ROSAnnotator project aims to develop a standalone web application specifically designed to enhance the analysis of Robot Operating System Bag (ROSBag

Confluence Open preview

Sprint Goals

Sprint Management with Trello Board

Y Owned by yucpeng1 • Updated yesterday

Sprint Task Tracking (Trello Board) Note: For each task breakdown, please read the ID as: "Epic Id . UId . Taks Id" (where UId is the user story Id. Please refer to the User Story Table) The Epic assignment is not displayed properly in the Confluence embedded link, please directly visit our Trello board to view the Epic assignment . https://trello

Confluence Open preview

Meeting Notes

Meeting Notes

TW Owned by Tianqi Wang • Updated on 19 Mar 2024

Meeting Notes Incomplete Tasks Decisions

Confluence

Open preview

Stand Up Meetings

Stand Up Meetings

Y Owned by yucpeng1 • Updated on 19 Mar 2024

Daily Stand Up

Confluence

Open preview

🧑‍💻 Team

Name	Student ID	Email	Position	Responsibility
Bowen Fan	1035162	bffa@student.unimelb.edu.au	Product Manager/Frontend Developer	Leads project direction and user interface development, ensuring alignment with user needs and seamless backend integration.
Tianqi Wang	1045939	tww2@student.unimelb.edu.au	Scrum Master/ROS Analyst	Facilitates Agile processes and analyzes ROSBag data to guide development with actionable insights.
Guanqin Wang	1074138	guanqinw@student.unimelb.edu.au	DevOps Manager/Backend Developer	Manages infrastructure and backend development, focusing on performance, data processing, and system integration.
Yujie Zheng	1290137	yujiezheng@student.unimelb.edu.au	Backend Developer	Develops server-side logic and database interactions, ensuring efficient data handling and API functionality.
Yuchen Song	1377108	yuchsong2@student.unimelb.edu.au	Frontend Developer	Creates and improves the web application's visual elements and user experience, ensuring an intuitive and engaging interface.
Yucheng Peng	1382861	yucpeng1@student.unimelb.edu.au	ROS Analyst	Specializes in ROSBag data analysis, developing tools and algorithms for effective data interpretation to inform application features.
Abhishek Tummalapalli	1066956	atummalapall@student.unimelb.edu.au	ROS Analyst	Specializes in ROSBag data analysis, developing tools and algorithms for effective data interpretation to inform application features.

ROSAnnotator: A Web Application for ROSBag Data Analysis in

https://chri-lab.notion.site/ROSAnnotator-A-Web-Application-for-ROSBag-Data-Analysis-in-Human-Robot-Interaction-e856d9fb941f4ceca0b4c53a77e71d03

Project Overview, Background and Goal

Project spec: ROSAnnotator: A Web Application for ROSBag Data Analysis in Human-Robot Interaction | Notion

Project Overview

The ROSAnnotator project aims to develop a standalone web application specifically designed to enhance the analysis of Robot Operating System Bag (ROSBag) data, with a particular emphasis on Human-Robot Interaction (HRI). ROSBags are a crucial element in robotics research, serving as a standard format for logging and replaying messages within the ROS ecosystem. These logs can include a wide variety of data types, such as video streams, 3D point clouds, and custom messages tailored for HRI studies. The application is envisioned to be a versatile tool for researchers, enabling the loading of multiple ROSBags, integration with auto-transcription tools for audio data processing, and the provision of a synchronized, interactive dashboard for intuitive data visualization.

Background

Human-Robot Interaction (HRI) is a growing field of study that explores how humans interact with robots in various contexts, from industrial applications to personal assistance and beyond. The analysis of HRI data is complex, involving multiple modalities such as visual data, audio communications, and sensor data from the robot. The ROS ecosystem provides a flexible framework for robot development and research, but the analysis of ROSBag data, especially from HRI experiments, requires specialized tools. Existing tools like Elan offer some capabilities for annotation and analysis but may not fully meet the unique needs of HRI research, such as handling specific ROS data types or synchronizing multiple data streams.

Goal

The primary goal of ROSAnnotator is to fill this gap by providing a comprehensive tool that facilitates the detailed analysis of HRI experiments. The project focuses on several key objectives:

- **Data Handling Capabilities:** To build robust support for reading and parsing the diverse data types contained in ROSBags, ensuring researchers can access and analyze all relevant data components of their HRI experiments.
- **Synchronised Interactive Dashboard:** To develop an intuitive, user-friendly interface that displays various data streams in a synchronized manner, allowing researchers to interact with and analyze data more effectively.
- **Annotation Features:** To enable detailed annotations of HRI interactions, including custom scales for states (e.g., "child looking at the robot") and events (e.g., "child entering input on the table"), thereby enhancing the depth and specificity of analysis.
- **Integration and Automation:** Although given less priority initially, integrating with auto-transcription tools for audio data and exploring automatic annotation methods based on audio transcripts are also envisioned to streamline the annotation process and make the tool more versatile.

Project Specification with detailed Sprint planning

The project spec for ROSAnnotator outlines a comprehensive plan to develop a standalone web application for the analysis of ROSBag data, specifically focusing on Human-Robot Interaction (HRI). The objectives and methodology indicate a multi-phase approach to the project, emphasizing data handling, user interaction, and innovative features for annotation and analysis.

Specific Requirements & Breakdown

Overall Project Requirements (for reference and future sprints)

1. **Data Handling Capabilities**
   - Develop parsing capabilities for video streams, 3D point clouds, and `hri_msgs`.
2. **Multiple ROSBags Loading**
   - Implement a feature to manage and analyze multiple ROSBags.
3. **Integration with Auto-Transcription**
   - Connect with an auto-transcription tool for audio to text conversion.
4. **Synchronised Interactive Dashboard**
   - Design and implement a user interface for data visualization.
5. **Annotation Features**
   - Develop custom scales and annotation tools for users.
6. **Automatic Annotation Exploration**
   - Research and prototype automatic annotation methods.

> Please navigate to a detailed requirement specification and distribution using the below table.

Sprint 1	**Sprint 1 Planning: Inception** HW Owned by Harry Wang • Updated 5 days ago Sprint 1: Requirement Analysis and Planning Phase Gather Detailed Requirements and Specifications Meetings with PhD students for requirements gathering. Identifying user stories based on these requirements. Survey of Existing Annotation Tools Research and document features of existing tools (e.… Confluence Open preview
Sprint 2	**Sprint 2 Planning: Development** YZ Owned by Yujie Zheng • Updated 2 days ago Sprint 2 Planning: Development (March 25 - April 26) Overview Sprint 2 will span from March 25 to April 26, focusing on enhancing the core functionality and user interface of our audio-to-transcript application with speaker separation. This period aims to solidify our backend infrastructure, streamline front-end… Confluence Open preview
Sprint 3	**Sprint 3 Planning: Development** YZ Owned by Yujie Zheng • Updated yesterday Objective: Focus on refinement, and addressing any identified issues or improvements. Duration: 4 weeks Integration with Auto-Transcription Integrate auto-transcription service for audio to text conversion. Develop annotation functionalities. Duration: 1 Week Annotation Features Develop custom scales and… Confluence Open preview
Sprint 4	

Sprint Planning Notes

- **Sprint Duration**
    - Inception: 2 weeks
    - Development: 4 weeks
    - Handover: 2 weeks
- **Daily Standups** to discuss progress, obstacles, and next steps.
- **Review Meeting** at the end of Sprint 1 to evaluate the deliverables and plan for Sprint 2, focusing on System Design.
- **Retrospective Meeting** to discuss what went well, what didn’t, and how processes can be improved moving forward.

🎯 Requirements & Design

🤓 Client Requirements

ROSAnnotator: A Web Application for ROSBag Data Analysis in Human-Robot Interaction | Notion

Created by Wafa

Project Description

CHRI on Notion

Notion

🗒️ Project Overview

The ROSAnnotator project aims to develop a standalone web application specifically designed to enhance the analysis of Robot Operating System Bag (ROSBag) data, with a particular emphasis on Human-Robot Interaction (HRI). ROSBags are a crucial element in robotics research, serving as a standard format for logging and replaying messages within the ROS ecosystem. These logs can include a wide variety of data types, such as video streams, 3D point clouds, and custom messages tailored for HRI studies. The application is envisioned to be a versatile tool for researchers, enabling the loading of multiple ROSBags, integration with auto-transcription tools for audio data processing, and the provision of a synchronized, interactive dashboard for intuitive data visualization.

🕵️**Background**

Human-Robot Interaction (HRI) is a growing field of study that explores how humans interact with robots in various contexts, from industrial applications to personal assistance and beyond. The analysis of HRI data is complex, involving multiple modalities such as visual data, audio communications, and sensor data from the robot. The ROS ecosystem provides a flexible framework for robot development and research, but the analysis of ROSBag data, especially from HRI experiments, requires specialized tools. Existing tools like Elan offer some capabilities for annotation and analysis but may not fully meet the unique needs of HRI research, such as handling specific ROS data types or synchronizing multiple data streams.

🎯 Objective

The goal of this project is to implement ROSAnnotator; a standalone web application designed to facilitate the analysis of ROSBag data, with a special focus on Human-Robot Interaction (HRI) captured in various formats, including video streams, 3D point clouds, and custom HRI messages. The application will support loading multiple ROSBags, connecting to an auto-transcription tool for processing audio data, and providing a synchronised, interactive dashboard for data visualisation. Users will be able to manually annotate interactions with custom scales, annotating states (e.g., "child looking at the robot") and events (e.g., "child entering input on the table"). The project will also explore the potential for automatic annotation leveraging audio transcripts.

✏️ Analysis

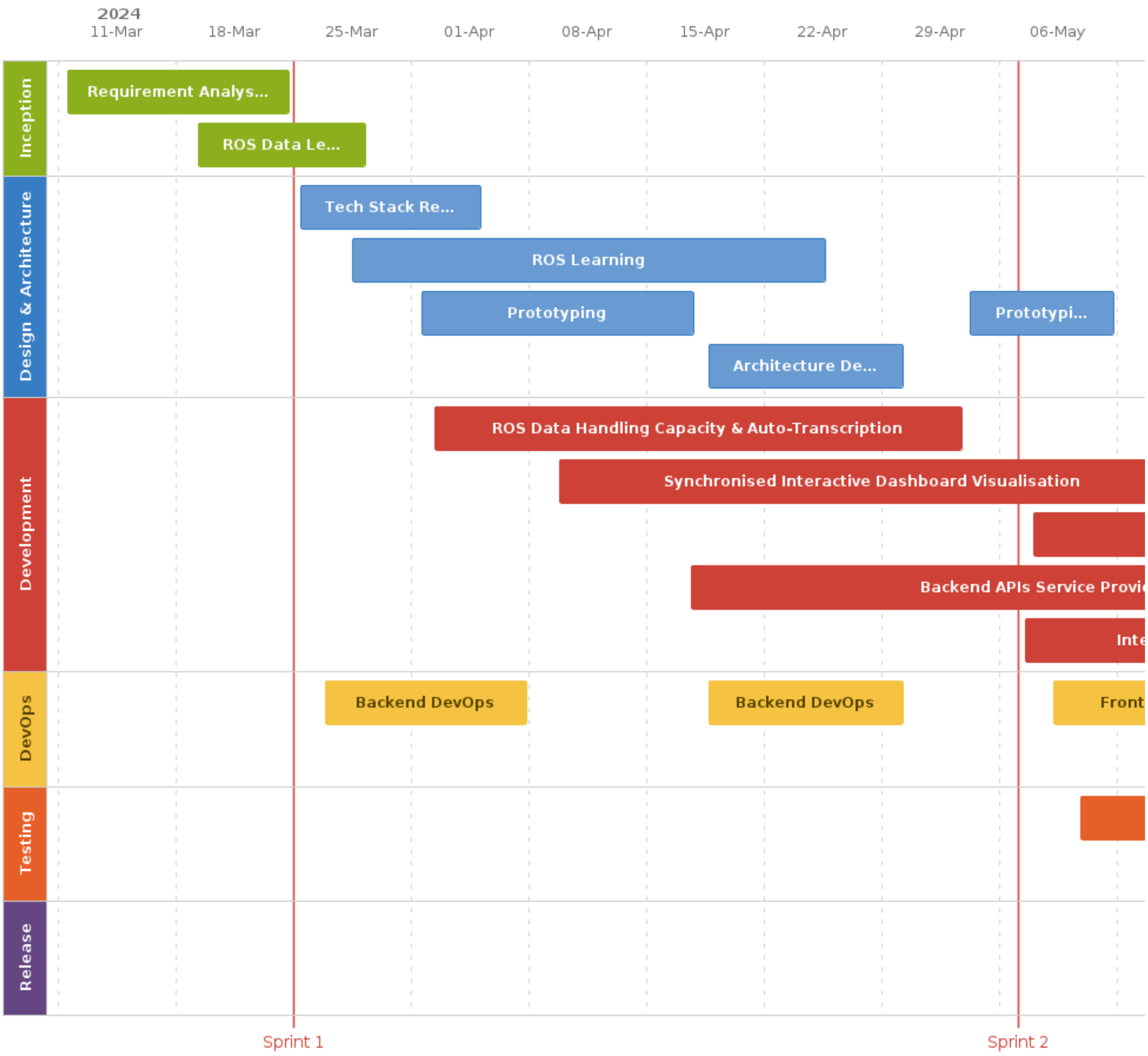
Motivational model	Personas	User Stories	Use Cases	Prototype

📊 Success metrics

Goal	Metric

Data Handling Capabilities	Built robust support for reading and parsing the diverse data types contained in ROSBags, ensuring researchers can access and analyse all relevant data components of their HRI experiments.
Synchronised Interactive Dashboard	Developed an intuitive, user-friendly interface that displays various data streams in a synchronised manner, allowing researchers to interact with and analyze data more effectively.
Annotation Features	Enable detailed annotations of HRI interactions, including custom scales for states (e.g., "child looking at the robot") and events (e.g., "child entering input on the table"), thereby enhancing the depth and specificity of analysis.
Integration and Automation	Integrating with auto-transcription tools for audio data and exploring automatic annotation methods based on audio transcripts are also envisioned to streamline the annotation process and make the tool more versatile.
Non Functional Requirments	Have the app running on any platform and system without heavy manual configuration. The client needs to be able to start the app with only a few simple commands.

🌟 Milestones



⚠ Out of Scope

Due to the complexity of the synchronized data visualization component in the frontend, the handling of complex data types, and the limited timeframe, we have been actively discussing with our client in sprints 1 & 2 to determine what is feasible within our resource constraints. Consequently, we have identified the following five major functional areas to be excluded from the current project scope. Priorities have been assigned to some tasks to ensure appropriate progress.

	Out of Scope Feature	Description	Reference
1	**3D Point Cloud**	All features related to 3D point clouds, including visualization and data processing, will not be implemented.	see user story 2.1

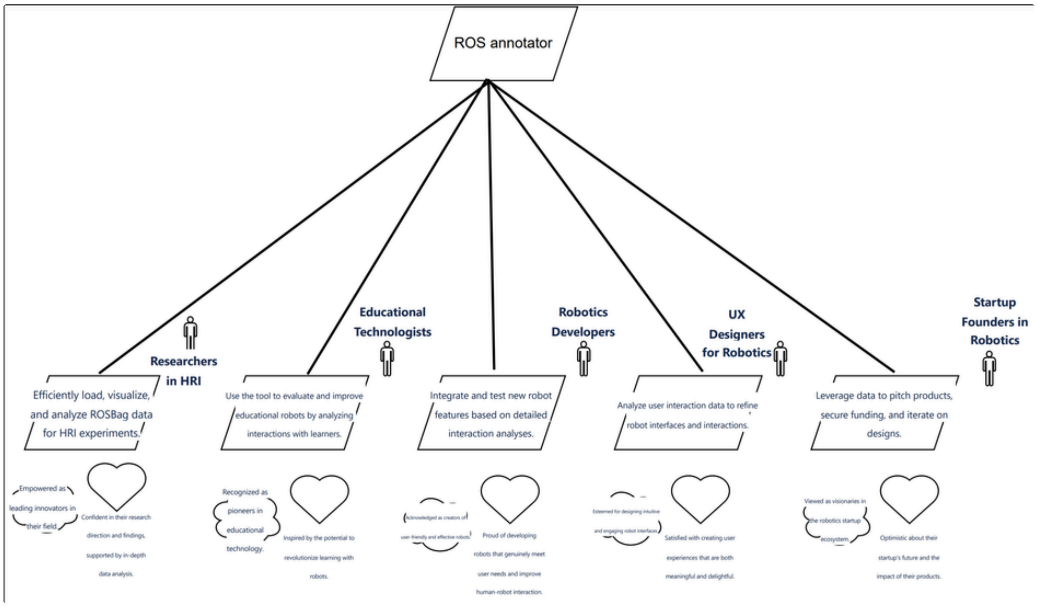
2	**Complex Operations on Frontend Data Display**	The ability to perform complex operations on displayed data in the frontend, such as filtering and toggling data, and video frame manipulation.	see user stories 1.4, and 2.4 - 2.9
3	**Persistent Storage for Processing Data or Annotations**	Interactions with databases for storing the progress of users or processing data will not be included in this project phase.	see user stories 3.6 - 3.7
4	**Batch & Auto Annotations**	The development of features for batch processing and automatic annotations	see user stories 6.1, 6.2, and 6.4
5	**Manual Correction of Auto-Generated Transcripts**	The functionality for manual corrections of automatically generated transcripts	see user story 4.3

These exclusions are aimed at focusing our efforts on deliverables that are achievable within the given constraints, ensuring that we deliver quality and value where it matters most.

Motivational model

Goal Model

Reference: DO-BE-FEEL


ROS annotator
Researchers in HRI
Educational Technologists
Robotics Developers
UX Designers for Robotics
Startup Founders in Robotics
Efficiently load, visualize, and analyze ROSBag data for HRI experiments.
Use the tool to evaluate and improve educational robots by analyzing interactions with learners.
Integrate and test new robot features based on detailed interaction analyses.
Analyze user interaction data to refine robot interfaces and interactions.
Leverage data to pitch products, secure funding, and iterate on designs.
Empowered as leading innovators in their field.
Confident in their research direction and findings, supported by in-depth data analysis.
Recognized as pioneers in educational technology.
Inspired by the potential to revolutionize learning with robots.
Acknowledged as creators of user-friendly and effective robots.
Proud of developing robots that genuinely meet user needs and improve human-robot interaction.
Esteemed for designing intuitive and engaging robot interfaces.
Satisfied with creating user experiences that are both meaningful and delightful.
Viewed as visionaries in the robotics startup ecosystem.
Optimistic about their startup's future and the impact of their products.

DO-BE-FEEL

DO-BE-FEEL table for ROSAnnotator Project

Reference : Motivational model

Users	DO (Functional Goal)	BE (Quality Goal)	FEEL (Emotional Goal)
Researchers in HRI	Efficiently load, visualize, and analyze ROSBag data for HRI experiments.	Empowered as leading innovators in their field.	Confident in their research direction and findings, supported by in-depth data analysis.
Educational Technologists	Use the tool to evaluate and improve educational robots by analyzing interactions with learners.	Recognized as pioneers in educational technology.	Inspired by the potential to revolutionize learning with robots.
Robotics Developers	Integrate and test new robot features based on detailed interaction analyses.	Acknowledged as creators of user-friendly and effective robots.	Proud of developing robots that genuinely meet user needs and improve human-robot interaction.
UX Designers for Robotics	Analyze user interaction data to refine robot interfaces and interactions.	Esteemed for designing intuitive and engaging robot interfaces.	Satisfied with creating user experiences that are both meaningful and delightful.
Startup Founders in Robotics	Leverage data to pitch products, secure funding, and iterate on designs.	Viewed as visionaries in the robotics startup ecosystem.	Optimistic about their startup's future and the impact of their products.

Personas

ℹ Note:

- We've developed four personas for this project, tailored to its specialized use scenario, as the initiative is primarily aimed at research purposes.
- You can explore detailed information about each persona by clicking on their names.
- Clicking on the "User End Goal" will reveal their expected outcomes and what they hope to achieve with this product.
- The final column provides references to a list of user stories that may be applicable to each persona.

⭐ Persona ⭐	Name	End Goal	User Story
	Dr. Emily Nguyen	📄 User Persona 1 | :light_bulb_on: Expectation	1.1 ~ 1.2 2.1 ~ 2.2 3.1 ~ 3.2 5.1
	Alex Rivere	📄 User Persona 2 | :light_bulb_on: Expectation	1.1 ; 1.3 ; 1.4 2.2 ~ 2.4 3.2 ~ 3.4 5.2
	Samira Campbell	📄 User Persona 3 | :light_bulb_on: Expectation	1.1 ~ 1.2 2.5 ~ 2.7 3.5 ~ 3.6 4.1 ~ 4.3

	Rahul Sharma	User Persona 4 | :light_bulb_on: Expectation	1.3 2.1 ; 2.2 ; 2.8 ~ 2.9 3.3 ; 3.7 ~ 3.8 5.2 ~ 5.3 6.1 ~ 6.4

User Persona 1

Persona name	**Dr. Emily Nguyen**
Persona role	Assistant Professor
Job description	Dr. Nguyen is an Assistant Professor in Robotics Engineering, specializing in Human-Robot Interaction (HRI) research, focusing on educational robots and their impact on children's learning.

🏛️ Institution

Institution name	TechnoFuture University
Institution size	Medium (500-1000 employees)
Industry	Education & Research in Robotics
Institution Description	TechnoFuture University, nestled in a bustling urban area, stands at the forefront of innovation in education and research, particularly in the field of robotics engineering. It offers a fertile ground for pioneering work in Human-Robot Interaction (HRI).

👥 Demographic information

Age	35
Gender	Female
Income	$80,000 annually
Education level	Ph.D. in Robotics Engineering
Residential environment	Urban

🖋️ Personal quote

- "Understanding the subtle cues in human-robot interaction can revolutionize educational technology."

📜 Biography

Dr. Emily Nguyen has been fascinated by robotics since her undergraduate years. After completing her Ph.D., she dedicated herself to research in HRI, specifically focusing on how robots can enhance learning experiences for children. She balances her time between teaching, conducting research, and contributing to academic journals. Emily is always on the lookout for innovative tools to better analyze and interpret interaction data between children and robots.

Professional goals	**Motivators**
• To pioneer research that improves robot design for educational purposes. • To develop methodologies for assessing and enhancing robot-child interaction.	• Passion for robotics and education. • Desire to make a significant contribution to the field of HRI. • The quest for finding efficient tools to streamline research processes.
Challenges	**Sources of information**
• Limited time due to teaching responsibilities. • Need for a comprehensive tool that simplifies the analysis of complex HRI data.	• Academic journals and conferences in robotics and HRI. • Online forums and communities dedicated to robotics research. • Collaboration with other researchers and industry professionals.

💡 Expectation

1. Professional ROS Data Analysis:

- Ability to efficiently load and parse ROSBag data containing diverse data types from HRI experiments, particularly those involving educational robots and children.
- Streamline process for identifying and analyzing human-robot interaction that can impact children's learning.

2. Synchronized Display of Multimodal Data:

- Seamless integration and synchronized display of video streams, 3D point clouds, and custom HRI messages to facilitate comprehensive analysis of robot-child interactions.
- High-performance visualization tools that can handle complex datasets without significant lag or processing delays.

3. Intuitive and Customizable Annotation:

- User-friendly interface for manual annotation of interactions, enabling detailed documentation of states (e.g., "child engaged with the robot") and events (e.g., "learning milestone achieved").
- Flexibility in defining and modifying annotation scales to suit specific research needs and objectives.

4. Data Preservation and Export:

- After completing annotations, the expectation is to easily save the annotated datasets in ROSBag format for future usage, sharing with the research community and school, or further analysis.
- Assurance of data integrity and the preservation of all original and annotated data elements within the ROSBag format.

5. Enhanced Research Productivity:

- Reduction in the time and effort required to analyze complex HRI data, allowing for more focus on deriving meaningful insights and developing educational robots.
- A tool that fits seamlessly into the research workflow, complementing existing methodologies and enhancing the ability to conduct rigorous scientific studies.

User Persona 2

Persona name	**Alex Rivere**
Persona role	Ph.D candidate
Job description	Alex is a Ph.D. candidate researching the impact of social robots in healthcare settings, with a focus on elderly care. His work involves understanding how interactions with robots can enhance engagement and companionship for the elderly.

🏢 Company

Company name	HealthTech Robotics Lab
Company size	Small (50-100 employees)
Industry	Healthcare & Robotics Research
Company Description	The HealthTech Robotics Lab is a pioneering research facility specializing in the intersection of healthcare and robotics. It is dedicated to developing advanced robotic systems designed to improve the lives of the elderly. The lab is renowned for its development of day-care robots that offer companionship and support to seniors living alone, addressing critical challenges in elderly care with cutting-edge technology.

👥 Demographic information

Age	29
Gender	Male
Income	$30,000 annually (stipend)
Education level	Pursuing Ph.D. in Robotics and Healthcare Interaction

Residential environment	Urban, shared apartment near the university

🖋️ Personal quote

- "Robots have the potential to transform elderly care, making companionship accessible to all."

📜 Biography

Alex's journey into robotics began during his undergraduate studies in biomedical engineering. Inspired by the possibilities of improving lives through technology, he decided to pursue a Ph.D. focusing on the role of robots in elderly care. His research aims to identify factors that promote positive engagement between the elderly and robots, hoping to improve adherence to therapy and overall quality of life. Alex is dedicated to his research but faces challenges in analyzing complex interaction data efficiently.

Professional goals	**Motivators**
• To identify key factors that promote positive engagement between the elderly and social robots. • To contribute to the development of robots that can provide companionship and support to the elderly.	• A desire to improve the quality of life for the elderly. • The challenge of integrating technology and healthcare in meaningful ways. • The opportunity to contribute to groundbreaking research in HRI.
Challenges	**Sources of information**
• Analyzing complex and voluminous interaction data within limited time frames. • Finding user-friendly tools to facilitate the comparison of data across multiple sessions and participants.	• Academic journals in robotics, healthcare technology, and gerontology. • Conferences and workshops on robotics and healthcare innovations. • Online communities and forums dedicated to robotics research and healthcare technology.

💡 Expectation

1. Professional ROS Data Management:

- Alex expects the ability to efficiently load and analyze multiple ROSBag datasets from various scenarios collected in his lab.
- This feature is essential for conducting comparative analyses across different sessions and identifying patterns or discrepancies in the interaction data.
- The tool should accommodate the complexity and volume of data specific to elderly care scenarios, enabling streamlined analysis workflows.

2. Advanced Visualization Tools:

- He seeks advanced visualization tools capable of displaying data in a manner that allows for easy identification of key moments and patterns.
- Alex needs the capability to view 3D models of the environment to better understand the spatial dynamics between the elderly and robots, especially in contexts where the robot's mission includes preventing collisions with obstacles.
- The ability to select specific time points for detailed examination is crucial for dissecting interactions and environmental navigation.

3. Annotation and Improvement Tracking:

- Alex requires a user-friendly interface to annotate moments when the robot's performance did not meet expectations.
- This functionality should enable him to document instances of suboptimal interaction or navigation, facilitating a focused review and iteration process.

- The goal is to leverage these annotations to refine robot behaviour, ensuring safer and more effective companionship and support for the elderly.
- The tool must allow for easy modification and tracking of these annotations over time to measure progress and implement enhancements effectively.

User Persona 3

Persona name	**Samira Campbell**
Persona role	UX Designer
Job description	UX Designer at a large robotics company, specializing in designing intuitive interfaces and interactions for industrial robots to enhance safety and productivity on the factory floor.

🏢 Company

Company name	RoboTech Innovations
Company size	Large (Over 5,000 employees)
Industry	Robotics & Manufacturing
Company Description	RoboTech Innovations is a leading company in the field of industrial robotics, known for its commitment to manufacturing the most usable and safe robots on the market. Situated in a technologically advanced industrial park, RoboTech Innovations is dedicated to pushing the boundaries of robotics engineering, focusing on creating robots that enhance productivity and user experience in industrial settings.

👥 Demographic information

Age	32
Gender	Female
Income	$95,000 annually
Education level	Master’s Degree in Human-Computer Interaction

Residential environment	Suburban, owns a home

🖋️ Personal quote

"Designing for humans means understanding every gesture and glance. Our robots need to be partners, not obstacles."

📜 Biography

Samira Campbell has always been passionate about the intersection of technology and human interaction. With a background in computer science and a master's in human-computer interaction, she embarked on a career in UX design focused on robotics. At RoboTech Innovations, Samira is tasked with making industrial robots not just tools, but collaborative partners for workers. Her challenge is to design systems that are both efficient and intuitive, ensuring safety and productivity in high-stakes environments. She relies on detailed HRI data to understand user needs and improve design continuously.

Professional goals	**Motivators**
• To innovate in the field of UX design for robotics, making machines more accessible and intuitive for human workers. • To leverage data from HRI studies to inform design decisions and enhance robot-user interaction.	• A strong belief in the potential of robots to transform industries and improve human work environments. • The desire to make technology accessible and useful for people of all skill levels. • The challenge of translating complex robotic capabilities into simple, intuitive interactions.
Challenges	**Sources of information**
• Accessing detailed, actionable HRI data to inform design decisions. • Balancing the technical capabilities of robots with the intuitive expectations of human users. • Ensuring safety and productivity through design in diverse industrial environments.	• Industry conferences on robotics and human-computer interaction. • Journals and publications on UX design, robotics, and ergonomics. • User feedback sessions and field studies in industrial settings.

💡 Expectation

1. Detailed Interaction Visualisation:

- Ability to review video recorded from her company's robots frame by frame, enabling her to observe users' reactions and report on the UX quality of new robots.
- This granular insight is crucial for identifying subtle nuances in human-robot interaction that can significantly impact the user experience.

2. Expert Annotation Capabilities:

- Facility to annotate different moments of the video and audio using expert domain knowledge on UX.
- This includes labelling interactions to feed into future machine learning training, enhancing the robot's behaviour and future user experience.
- The ability to easily categorize and label data for analysis and training purposes is vital for refining robot responsiveness and intuitiveness.

3. Audio Transcript Conversion:

- Access to audio transcripts of interactions, allowing for a detailed analysis of the robot's verbal responses.
- This feature is essential for identifying which replies may be inappropriate or confusing, enabling targeted improvements to robot communication strategies to enhance overall user experience.

User Persona 4

Persona name	**Rahul Sharma**
Persona role	Co-founder and HCI Engineer at AutoTransTech
Job description	Co-founder and HCI Engineer at an innovative startup, AutoTransTech, which develops advanced, completely human-free parcel stations powered by robotics for sorting, transporting, and preparing goods.

🏢 Company

Company name	AutoTransTech Startup
Company size	Small (30 employees)
Industry	Logistics and Robotics in the Automotive Transportation Sector
Company Description	AutoTransTech is an innovative startup transforming the logistics industry through the development of fully autonomous parcel stations. Leveraging advanced robotics and cutting-edge human-computer interaction principles, the company specializes in creating efficient, human-free systems for sorting, transporting, and preparing goods. AutoTransTech is at the forefront of redefining automotive transportation logistics to meet the evolving demands of global commerce and e-commerce growth.

👥 Demographic information

Age	40
Gender	Male
Income	Variable, dependent on startup success and funding rounds
Education level	Bachelor's Degree in Mechanical Engineering, self-taught in programming and business management
Residential environment	Urban, in a tech-centric city area

🖋️ Personal quote

"Revolutionizing logistics with robotics, we're creating seamless, efficient, and entirely autonomous parcel stations."

📜 Biography

Rahul Sharma transitioned from a mechanical engineer to an entrepreneur with a vision for transforming the logistics industry. At AutoTransTech, Rahul leads the development of autonomous parcel stations that utilize sophisticated robotics to handle sorting and transportation tasks without human intervention. This venture represents a bold step into the future of automotive transportation, where efficiency, speed, and reliability are significant. Rahul's deep involvement in HCI ensures that the interactions between the robots are as seamless and efficient as possible, focusing on optimizing the system's overall performance and reliability.

Professional goals	**Motivators**
• To innovate in the logistics industry by introducing fully autonomous parcel stations that redefine efficiency and reliability. • To overcome the challenges of designing highly efficient, cooperative robotic systems for logistics applications. • To leverage technology for creating scalable, future-proof solutions in logistics, meeting the growing demands of e-commerce and global trade.	• A deep desire to make a positive impact in the field of robot automation industry through technology. • The entrepreneurial challenge of building a successful business that delivers value to society. • The technical challenge of developing robots that are both reliable and commercialisable for business.
Challenges	**Sources of information**
• Balancing rapid development and iteration with thorough and insightful analysis of HRI data to inform design decisions. • Limited resources for dedicated data analysis personnel and tools. • Ensuring the robots are reliable for independent workflow and continue add value to business.	• Industry reports and research papers on robotics and automation within logistics. • Feedback and performance data from the robotic systems in operation. • Conferences, workshops, and seminars focused on robotics technology, and human-computer interaction.

💡 Expectation

1. Behavioural Analysis Capability:

- Rahul expects the ability to use the ROSAnnotator to analyze and study the behaviour of the robots in the parcel stations, identifying areas for optimization and enhancement.

2. Synchronized Multi-Robot Data Visualization:

- He requires the functionality to load multiple ROS data streams from different robots into one synchronized dashboard, enabling the visualization of their cooperative efficiency and performance.

3. Custom Annotation Tools:

- The system should allow for the creation of custom labels for annotating ROS data.
- Rahul aims to define his own scales and event types specific to AutoTransTech's operations, facilitating a tailored analysis approach.

4. Batch and Automatic Annotation:

- Given the high volume of data generated by the robots, Rahul needs the capability for batch and automatic annotation of multiple robots' data simultaneously. This feature is crucial for efficient data processing and analysis.

5. Future AI Integration:

- Looking ahead, Rahul plans to integrate AI into AutoTransTech's systems. The ROSAnnotator's ability to support this integration is essential, as it will enable more sophisticated analysis and predictive modelling of robot behaviour and station efficiency.

User Stories

📊 User Story Table

Please refer to the user story table for a detailed breakdown of our requirements into specific user stories.

User Story Table

HW Owned by Harry Wang • Updated 5 days ago

Note: For the end users, please refer back to the list of Personas . We do not have points assigned for “Out of Scope“ features (indicates as out), and may not have points assigned for every lowest priority feature. The blue rows indicate the name of each Epic. Epic ID User Story ID As a <User> I want to <Do Something> So that <achieve goals> Velo

Confluence

Open preview

📚 Epics

We categorize our user stories and development functional areas using epics. We have divided the project requirements into 7 epics, labelled with IDs from 1 to 7. Each epic represents a cohesive functional area with its own responsibilities. For each epic, both frontend and backend cooperation is required. For a complete view of all epics, please refer to the User Story Table and the "Epic Board" list on our Trello.

🔢 Story Points

Justification

We assign story points to each user story primarily based on the difficulty level of implementation. We use 3 and 5 as the base units for forming points. Larger stories are estimated by adding the repetitions of these numbers together, following this sequence: 3, 5, 6, 8, 11, 13, 15, 16, etc. (*e.g. 16 = 3 x 2 + 5* x *2*). We do not assign more than 16 points to a single story; if necessary, we break it down into smaller stories.

Each story is further divided into smaller, actionable tasks assigned to individual team members during a sprint. The total story points for a user story are simply the sum of all the story points of its subtasks.

The reason we choose to use this approach is because this method helps manage workload, adjust velocity, and track the completion of points by each developer more easily, as tasks in this project can vary significantly in difficulty.

ℹ Please visit our Product Backlog list on Trello to view the user story distributions and the Sprint Task Tracking Board to see the task breakdown.

Story Point Assignment Approach:

For ongoing tasks within a sprint, if we find that the initial estimation significantly deviates from the actual workload, we adjust the story points accordingly. For future sprints, we first estimate the number of subtasks for each story and then use the Planning Poker method to estimate the points for each task according to their difficulties during a planning meeting. The estimation that receives the highest vote will be assigned.

🚄 Velocity Estimation

Background

The project team is engaged in an Agile development process, with sprints as the primary time management structure for delivering new features and updates. The team's current velocity—an estimate of the amount of work they can complete in a single sprint—is a critical factor for planning and forecasting future work.

Justification

Using the smallest story points as a unit, we estimate that a task worth **3 points** will take a developer one week to complete. With approximately 5 developers actively working on the project each week, this amounts to **3 x 5 = 15** points per team per week. Each development sprint lasts 4 weeks, so we estimate that about **60 points** can be completed per sprint.

Please refer to the "Sprint Management Doc -> Sprint Retro" for velocity adjustments.

⚠ Priority

All priorities are assigned and discussed with our client during meetings. From top to low, the priorities range from the most expected / necessary features that the client wants to the features that can be omitted. We aim to address all MUST HAVE features, complete some COULD HAVE features, and, if time permits or the effort is minimal, we will tackle some NICE TO HAVE features.

Please see the table below for the priority legend.

Priority Legend	
MUST HAVE	Essential features that are critical for the software's basic functionality and core objectives.
COULD HAVE	Desirable features that enhance user experience or functionality but are not critical for the initial release.
NICE TO HAVE	Features identified as lowest priority, excluded from the current development cycle but potentially revisited later.

User Story Table

Note:

- For the end users, please refer back to the list of Personas.
- We do not have points assigned for “Out of Scope“ features (indicates as `OUT`), and may not have points assigned for every lowest priority feature.
- The blue rows indicate the name of each Epic.

Epic ID	User Story ID	As a <User>	I want to <Do Something>	So that <achieve goals>	Velocity Estimation (Story Points)	Priority
ROS & HRI Data Handling Capability						
1	**1.1**	Dr. Emily Nguyen Alex Rivere Rahul Sharma	efficiently **load and parse** various ROSBag data containing diverse data types, such as video streams, and audios *(Note: 3D point clouds is out of scope feature)*	I can efficiently access and analyze the interaction data from the HRI experiments without manual data conversion	15	MUST ...
	1.2	Dr. Emily Nguyen	handle data variabilities in ROSBags, such as **different** message **types** and **formats**	work with data from various sources without needing to preprocess extensively	8	MUST ...
	1.3	Alex Rivere Rahul Sharma	handle **large volumes** of ROSBag data efficiently, ensuring quick loading and parsing times	quickly find relevant data points for my analysis	6	NICE T...
	1.4	Alex Rivere	**filter and search** within the loaded ROSBags for specific data types or messages	I can quick allocte / lookup a recipe	OUT	COUL...
	1.5	Dr. Emily Nguyen Rahul Sharma	have robust support for parsing and visualizing custom **HRI messages** `hri_msgs`	I can deeply understand robot-child interactions and their impacts on learning	OUT	NICE T...
Synchronised Interactive Dashboard						
2	**2.1**	Dr. Emily Nguyen Rahul Sharma	**visualize 3D point clouds** from ROSBag data within the application	analyze spatial aspects of human-robot interactions in educational settings	OUT	NICE T...
		Dr. Emily Nguyen	load and play **video streams** from ROSBags	observe and analyze the dynamics of robot-child interactions in real-time	3	MUST ...

	2.2	Dr. Emily Nguyen Alex Rivere Rahul Sharma	all visualized data streams (video) to be **synchronized** in **real-time** on the dashboard *(Note: 3D point clouds, custom HRI messages are out of scope features)*	observe and analyze how different elements of human-robot interactions interplay simultaneously	8	MUST ...
	2.3	Alex Rivere	**interact** with the dashboard, such as **pausing**, **rewinding**, and **fast-forwarding** through data streams	closely examine specific moments or interactions of interest	15	MUST ...
	2.4	Alex Rivere	**toggle** the **visibility** of different data types on the dashboard	focus on specific aspects of the data without distraction from other information	OUT	COUL...
	2.5	Samira Campbell	review video recordings from industrial robots **frame by frame** within the application	closely observe and analyze users' reactions and interactions with the robots	OUT	COUL...
	2.6	Samira Campbell	**adjust playback speed** and navigate through video recordings easily	focus on critical moments of interaction without missing any details that could inform design improvements	OUT	COUL...
	2.7	Samira Campbell	**zoom** and enhance features on video footage	examine intricate details of user expressions and movements that might indicate usability issues or areas for enhancement	OUT	NICE T...
	2.8	Rahul Sharma	visualize **spatial relationships** and distribution among robots in a shared space	optimize layout and assignments for increased productivity	OUT	NICE T...
	2.9	Rahul Sharma	**highlight** and focus on data **for specific data types** or from specific robots or tasks	drill down into detailed analyses of individual performances within the context of the team's overall operation	OUT	NICE T...
Annotation Features						
3	**3.1**	Dr. Emily Nguyen	**intuitive** and **interactive** interface for annotating video streams and 3D point clouds	precisely mark and note relevant interactions between children and robots without extensive technical training	11	MUST ...
	3.2	Dr. Emily Nguyen Alex Rivere	easily **save** and **export** annotated **datasets** in a common format (e.g., ROSBag)	share findings with the academic community, preserve data for future analysis, or integrate insights into my research workflow	6	MUST ...
	3.3	Alex Rivere Rahul Sharma Samira Campbell	manually annotate the data(video and audio) on a **timeline** with **detailed labels** using my domain knowledge, such as "user	categorize interactions for in-depth analysis and future machine learning model training	5	MUST ...

			confusion", "successful interaction", or "navigation issue"			
	3.4	Alex Rivere	easily **edit** or **delete** annotations	refine my analysis and keep my data organized and accurate as my understanding evolves	5	MUST ...
	3.5	Samira Campbell	annotate the **transcribed text** with **custom** tags or notes	highlight and categorize important verbal interactions or phrases relevant to my research	6	COUL...
	3.6	Samira Campbell	**track** changes and **updates** to **previous annotations** over time	monitor the evolution of user interaction trends and the impact of future modifications	OUT	NICE T...
	3.7	Rahul Sharma	flexibility to **modify** and **expand** my **annotation schema** as our understanding of optimal robot behavior evolves	ensuring our analytics capabilities grow with our company	OUT	NICE T...
	3.8	Rahul Sharma	**store custom annotation** sets within the system	promoting a standardized approach to analyzing and discussing robot performance across our development	6	NICE T...
	3.9	Dr. Emily Nguyen	**synchronize annotations** across different data streams (video, audio transcripts, 3D point clouds)	maintain a cohesive understanding of interactions and events	8	MUST ...
Integration with Auto-Transcription						
4	**4.1**	Samira Campbell	automatically **transcribe audio data** from the ROSBags into text transcripts	analyze verbal interactions without the need for manual transcription, saving time and increasing accuracy	6	COUL...
	4.2	Samira Campbell	**correlate** transcribed text with **specific moments** and events **in the video**	I can gain insights into the verbal aspects of human-robot interactions	6	COUL...
	4.3	Samira Campbell	option to **manually review** and **correct** the auto-transcribed text	ensuring high accuracy and reliability of the data for analysis	OUT	NICE T...
Multiple ROSBags Loading						
5	**5.1**	Dr. Emily Nguyen	manage and **switch** between **multiple ROSBags** easily within the application	I can conduct comparative and composite analysis without significant downtime	8	COUL...
	5.2	Alex Rivere Rahul Sharma	easily **load multiple ROSBags** into the application	perform comparative analyses across different datasets to identify patterns or discrepancies in HRI data.	OUT	COUL...
	5.3	Rahul Sharma	**compare** different **data** format from different operational scenarios **side by side**	assess how well our robots work together under varying conditions	OUT	NICE T...

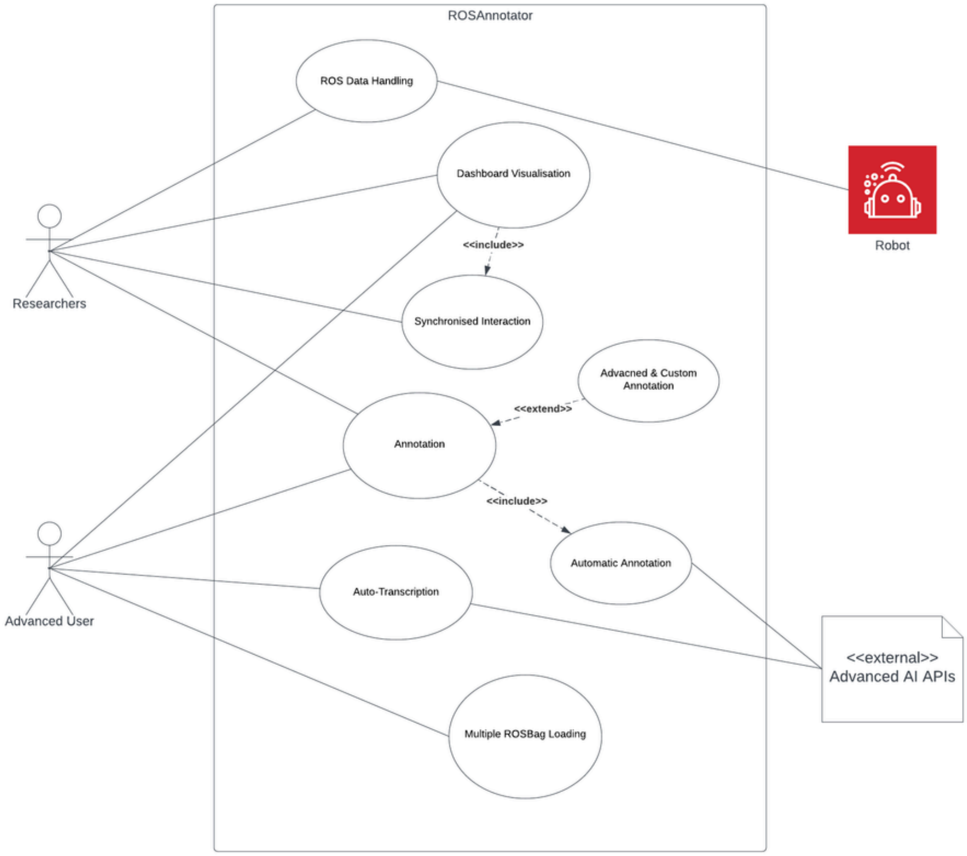
				and identify areas for synchronization improvements		
Automatic Annotation						
6	**6.1**	Rahul Sharma	**batch annotation** capabilities to efficiently process large volumes of data	allowing for quick categorization and analysis of robot performance across multiple parcel stations	OUT	NICE T...
	6.2	Rahul Sharma	**automatic annotation** of common events and behaviors detected in ROS data	streamline the initial stages of data analysis and focus on in-depth evaluation of anomalies and optimization opportunities	OUT	NICE T...
	6.3	Rahul Sharma	**customize rules** for automatic annotation	learn and adapt to our specific operational metrics and goals	6	NICE T...
	6.4	Rahul Sharma	**review and adjust** automatically generated annotations	ensuring accuracy and relevance to our constantly evolving understanding of optimal robot behavior	OUT	NICE T...
Non-Functional Requirements						
7	**7.1**	All	the app can be run on **any platform** without heavy manual configuration	I can use any device to run the system	15	MUST ...
	7.2	All	start using the app with only **a few simple commands** after pull down from GitHub	I can easily start using the app without complex configuration	11	MUST ...

Use Cases

Main Functional Requirements

Use Case	Step & Flow	Priority
ROS Data Handling	ROS Data Handling	MUST HAVE
Dashboard Visualisation	Dashboard Visualisation	MUST HAVE
Synchronised Interaction	Synchronised Interaction	MUST HAVE
Annotation Features	Annotation Features | 1. Basic Annotation	MUST HAVE
Custom Annotation	Annotation Features | 2. Advanced Customisation and Labeling	COULD HAVE
Automatic Annotation	Annotation Features | 3. Automatic Annotation	NICE TO HAVE
Auto-Transcription	Auto-Transcription	COULD HAVE
Multiple ROSBags Loading	Multiple ROSBags Loading	COULD HAVE

UseCase Diagram



Note:

- The two primary actors only represent different groups of users, rather than corresponding to specific roles.

ROS Data Handling

Title: Efficient ROSBag Data Loading and Parsing

Actors: Dr. Emily Nguyen, Alex Rivere, Rahul Sharma

Preconditions: ROSBag data is available and accessible by the application.

Main Flow:

1. The user selects the ROSBag file they wish to load into the application.
2. The application parses the ROSBag data, recognizing and notifying the user of various data types (e.g., video streams, 3D point clouds, audio).
3. The application efficiently loads the data onto the dashboard without any errors, making it readily accessible for analysis.
4. The user accesses and analyzes the HRI data, focusing on interactions without needing manual data conversion.

Postconditions: The user has successfully loaded and parsed ROSBag data, ready for in-depth analysis.

Dashboard Visualisation

Title: Visualizing 3D Point Clouds and Video Streams

Actors: Dr. Emily Nguyen, Rahul Sharma

Preconditions: 3D point cloud and video stream data are loaded into the system

Main Flow:

1. The user selects the ROSBag file containing 3D point cloud and video stream data ready for display.
2. The application loads and displays the 3D point cloud and video streams on the dashboard in a way that the user can interact with the data.
3. The user analyzes spatial aspects and dynamics of human-robot interactions by observing the synchronized data.
4. The user can click on each section of the window to change focus and drag the video or 3D objects to change the viewpoints.

Postconditions:

- All data has been displayed aligned to the same time points, the dashboard interface is intuitive and easy to navigate.
- The user has visualized and analyzed the 3D spatial and video data in synchronization.

Synchronised Interaction

Title: Interacting with the Synchronized Dashboard

Actors: Alex Rivere

Preconditions: Multiple data streams of different types are visualized on the dashboard.

Main Flow:

1. The user views synchronized data streams (video, 3D point clouds) on the dashboard.
2. The user interacts with the dashboard, using controls to pause, rewind, or fast-forward through the video data. Meanwhile, all other data streams (including 3D objects and audio) change correspond to the synchronised timeline.
3. The user focuses on specific moments or interactions of interest for closer examination.

Postconditions: The user has interactively examined specific data points across synchronized data streams.

Annotation Features

1. Basic Annotation

Title: Annotating Video Streams and 3D Point Clouds for State and Event

Actors: Dr. Emily Nguyen

Preconditions: Video and 3D point cloud data are loaded onto the dashboard and in a state ready to be modified.

Main Flow:

1. The user selects a segment of the video or a section of the 3D point cloud for **event** annotation. The application highlights the selected parts and prompts for actions.
2. The user chooses the annotation options, optionally adds in text, and inserts an annotation for the highlighted selection, noting relevant interactions or features.
3. The application shows the message of annotated success or failure.
4. Annotations are synchronized across different data streams for cohesive analysis.
5. The user clicks “save” to save or download the annotated data for future reference or sharing.
6. The system pops up the loading window and prompts for successful download.

Alternative Flow:

1. * The user selects a particular frame of the video or one timepoint of the 3D point cloud for **status** annotation. The application highlights the selected parts and prompts for actions.

Postconditions:

- The user has annotated and synchronized notes across multiple data types.
- The newly labelled data has been downloaded in the ROSBag format for future use.

2. Advanced Customisation and Labelling

Title: Customizing Annotated Datasets

Actors: Alex Rivere, Rahul Sharma

Preconditions: Video and 3D point cloud data are loaded onto the dashboard and in a state ready to be modified.

Main Flow:

1. Refer to Basic Annotation Steps 1 & 2.
2. The user selects the option to customise an annotation with self-defined scales or specific state names or events.
3. (Optional) The application prompts the user to save the self-defined label into the system.
4. The user chooses a labelled timepoint and easily edits or deletes annotations to refine their analysis.
5. The annotated dataset is saved in a common format, like ROSBag, for future loading or analysis.
6. The user shares the annotated dataset with the academic community or team members.

Postconditions: Customized annotations have been added and saved for further analysis or integration into research workflows.

3. Automatic Annotation

Title: Annotating Video Streams and 3D Point Clouds for State and Event

Actors: Dr. Emily Nguyen

Preconditions: Video and 3D point cloud data are loaded onto the dashboard and in a state ready to be modified.

Main Flow:

1. The user selects a segment of the video or a section of the 3D point cloud. The application highlights the selected parts and prompts for actions.
2. The user selects the “Automatic Annotation“ option on the toolbar in the application.
3. The system pops up the loading window to inform the user the application is conducting AI analysis on the selected portion.
4. The application displays and highlights all the newly labelled time points and indicates the automatic annotation has finished.
5. The user can select one or more annotations to refine the labelling. Actions include editing, deleting and adding new annotations.
6. Refer to Basic Annotation Steps 4 - 6.

Postconditions:

- The user has annotated and synchronized notes across multiple data types.
- The newly labelled data has been downloaded in the ROSBag format for future use.

Auto-Transcription

Title: Transcribing Audio Data from ROSBags

Actors: Samira Campbell

Preconditions: ROSBag with audio data is loaded into the application.

Main Flow:

1. The user selects the ROSBag file containing audio data for transcription.
2. The application displays the data as an audio waveform in the dashboard.
3. The user selects the “Auto-transcription“ button on the toolbar.
4. The application pops up the loading window and prompts the user that “AI is Calculating”.
5. The system automatically transcribes the audio data into text transcripts and displays the transcript in an independent window for further actions.
6. The user can review and, if necessary, correct the auto-transcribed text for accuracy. The user clicks the “Confirm” button to insert the audio transcript.
7. The transcribed text is correlated with video and 3D point cloud data in the synchronised timelines for comprehensive analysis, enhancing the user's analysis capabilities.

Postconditions:

- Verbal interactions from the ROSBag are transcribed and displayed on the dashboard.
- The data is aligned with other data types on the dashboard for future annotations.

Multiple ROSBags Loading

Title: Managing and Analyzing Multiple ROSBags

Actors: Dr. Emily Nguyen, Alex Rivere, Rahul Sharma

Preconditions: Multiple ROSBags are available for analysis.

Main Flow:

1. The user selects multiple ROSBags for loading into the application.
2. The application efficiently loads the selected ROSBags, ready for analysis.
3. The user selects one displaying option from various view options, including side-by-side, synchronised, controlled variables, and so on.
4. The user performs comparative analysis across the different datasets, identifying patterns or discrepancies.
5. The user performs annotations on multiple datasets. Refer to Auto-Transcription
6. Insights from the comparative analysis inform further research or operational improvements.

Postconditions: The user has successfully managed and analyzed multiple ROSBags, gaining valuable insights for research or operational enhancements.

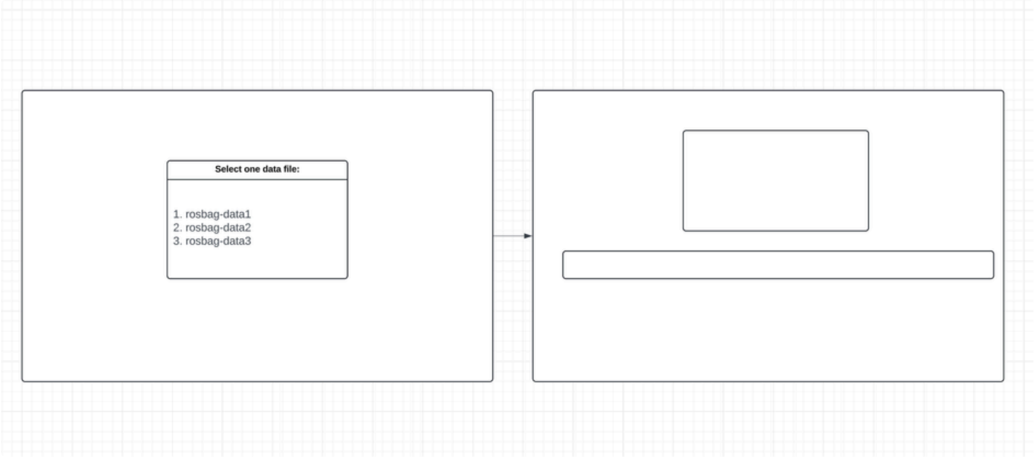
Prototype

https://www.figma.com/file/xUryyBhgdlA6OjIhQglWT5/Untitled?type=design&node-id=20%3A206&mode=design&t=4VONMFhcNv1gw6oI-1

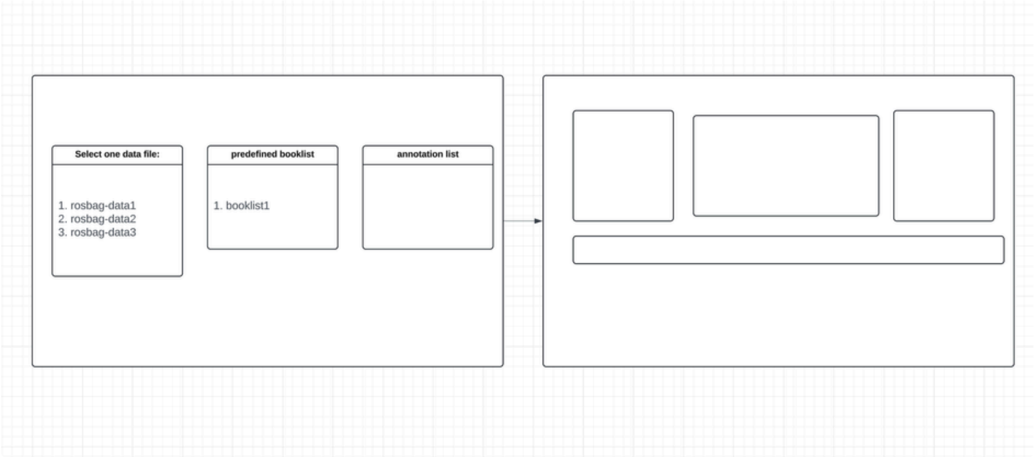
Additional Helper Page

🧠 Idea Perceptions:

1.



2.



💻 Development

GitHub Link:

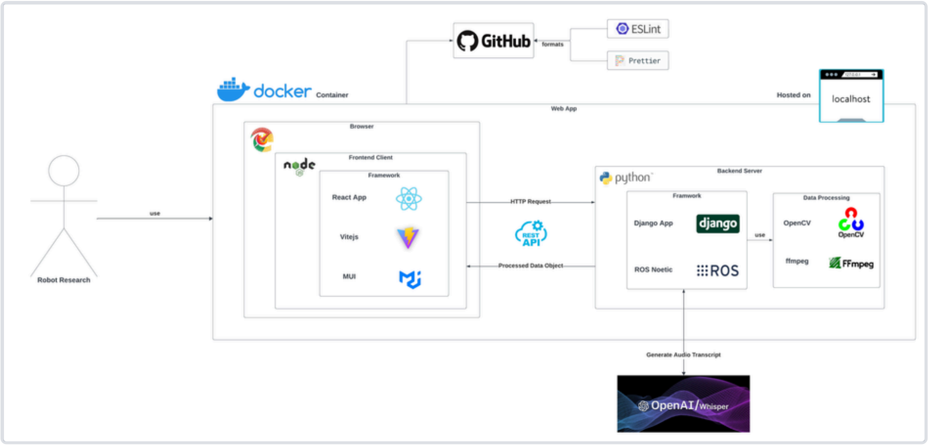
COMP90082-2024-SM1

GitHub is where COMP90082-2024-SM1 builds software.

GitHub

Architecture Diagram

📐 Architecture



Architecture flow

Please check our demonstration video:
https://drive.google.com/file/d/19A8k8uDG5NgB5xpRQiOwexR_nYtFT-9I/view?usp=sharing Connect your Google account

🚀 Deployment strategy

Deployment region: localhost

Instructions: Refer to our Deployment Instructions doc.

⌨️ Rest API details

Refer to our API Documentation doc.

Workflow Guide

Introduction

This guide aims to streamline our team's use of GitHub for managing software development projects. It covers essential Git practices, naming conventions, and workflow strategies to ensure consistency, efficiency, and clarity throughout our projects.

1. Git Workflow

- **Cloning a Repository**: Start by cloning the repository to your local machine using `git clone <repository-url>`.
- **Branching Strategy**:
  - Always create a new branch for your work, no matter how small.
  - Use the naming conventions outlined above.
  - Ensure you are branching off from the correct base branch (usually `main`).
- **Commit Messages**:
  - Write clear, concise commit messages in the imperative mood, e.g., "Add payment processing module."
  - Begin the message with a capital letter and do not end it with a period.
  - Use the body of the commit message to explain "what" and "why", not "how".

2. Pull Requests (PRs) and Code Reviews

- **Creating Pull Requests**:
  - PRs should be made against the `main` branch.
  - Include a clear title and a detailed description of the changes.
  - Reference related issues using hashtags, e.g., `Fixes #123`.
- **Code Review Process**:
  - At least one team member must review the PR before it can be merged.
  - Reviewers should check for code quality, adherence to project standards, and overall integration.
  - Use GitHub's commenting features to ask questions or suggest improvements.

3. Merging and Deployments

- **Merging PRs**:
  - Use the "Squash and Merge" option for small features or fixes to keep the history clean.
  - Use "Merge Commit" for significant features where preserving the detailed history of changes is valuable.
- **Deployment**:
  - After merging, changes should be tested in a staging environment before deploying to production.
  - Use tags and releases in GitHub to manage deployment versions.

4. Handling Issues

- **Creating Issues**:
  - When encountering a bug or a needed improvement, create an issue.
  - Provide a descriptive title and a detailed description including steps to reproduce, expected outcome, and actual outcome.
- **Assigning Issues**:
  - Assign issues to the appropriate team member.
  - Tag issues with labels such as `bug`, `feature`, `urgent`, etc., to help categorize and prioritize.

5. Additional Resources

- Git Documentation
- GitHub Flow

Frontend Documentation

The frontend of our web-app is built upon Node.js and Vite, which enable rapid development with its extensive built-in features, fast development process, and provides a solid foundation for our web-app development.

ⓘ Use this page to navigate through different part of frontend documentations

💻 Frontend Tech Stack A comprehensive guide for setting up the frontend environment and tech stack used.	Development Environment
🗒️ Coding Standards Frontend development obeys certain coding standards, please read through this document before contributing to the project	Frontend Coding Standards

Development Environment

This document outlines the development environment setup for our project, which utilizes Node.js and React for frontend development, and Python with Django for the backend. This guide is intended to help new contributors get their development environment up and running smoothly.

Frontend Development

Node.js and React

Our frontend application is built using React, a popular JavaScript library for building user interfaces, particularly single-page applications. Node.js is used for running JavaScript code server-side, and npm (Node Package Manager) is used to manage project dependencies.

Prerequisites:

- **Node.js**: Ensure you have the latest LTS (Long-Term Support) version of Node.js installed, which includes npm.
- **Vite**:
  - **Purpose**: Vite is a modern, fast frontend build tool that significantly improves the development experience by providing instant server start and hot module replacement.
  - Start the development server provided by Vite:

    `npm run dev` accessible at `localhost:3000` (your localhost).

Core Libraries:

1. **React**:
   - **Description**: React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”.
2. **React Player**:
   - **Description**: A React component that provides a simple player for various types of media including video and audio. It supports multiple streaming sources and provides a uniform API across different browser implementations.
3. **Material-UI (@mui/material and @mui/system)**:
   - **Description**: Material-UI provides a robust, customizable, and accessible library of foundational and advanced components, enabling you to build your own design system and develop React applications faster.
4. **Styled-components**:
   - **Description**: Allows you to write actual CSS code to style your components. It removes the mapping between components and styles, enabling complete and complex component styles within your JavaScript files.
5. **React LRC**:
   - **Description**: React LRC is a React component designed to display lyrics synchronized with audio or video playback. It is ideal for projects that require dynamic lyric or caption synchronization with media content, enhancing the multimedia experience.

Frontend Directory Structure:

- `src/`: Contains all source files.
- `components/`: Contains reusable React components.
  - `Block/`: Manages annotation blocks on the timeline.
  - `CustomSnackbar/`: Provides feedback through snackbar notifications.
  - `Timeline/`: Handles the video timeline and interactions.
  - `Transcript/`: An autoscroll transcript that synchronizes with the video.

- `App.tsx`: The entry component that integrates various components.
- `main.tsx`: Responsible for rendering the React application to the DOM.

Styles Management

This project uses styled-components for styling the React components, allowing for scoped and reusable style definitions.

Example of Styled Component:

```
const ScrollableTimelineContainer = styled('div')({
  overflowX: 'scroll',
  overflowY: 'hidden',
  width: '100%',
  cursor: 'pointer',
  padding: '10px 0',
});

```

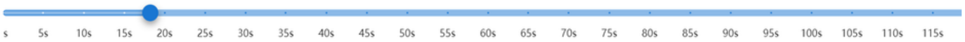
Key Components

Each component serves a specific purpose in the application, facilitating modularity and maintainability. Here's an overview of some critical components:

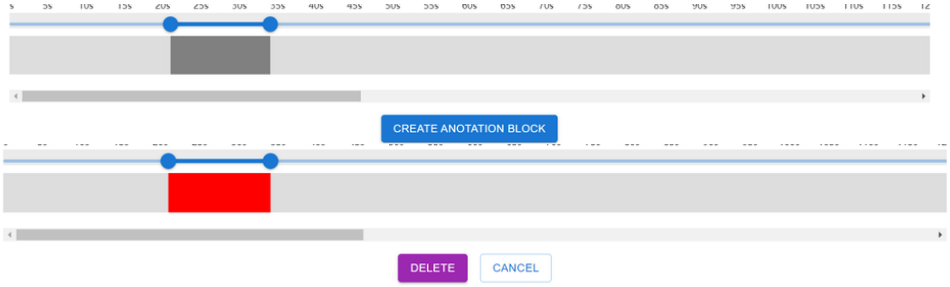
Timeline Component

The Timeline component is the central interface for video navigation and annotation management, providing essential functionality for interacting with video content and annotations.

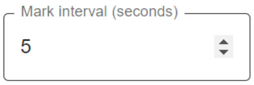
- **Features**:
  - **Navigation and Seeking**: Users can navigate through the video by moving a draggable slider along the timeline.



  - **Annotation Interaction Block**: Supports the addition, selection and deletion of annotations block directly on the timeline.

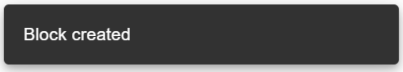


  - **Customizable Mark Interval**: Allows users to adjust the interval at which marks appear on the timeline, enabling a more tailored view according to the specific needs of the video or annotations.



Mark interval (seconds)
5

  - **Utilizes Snackbar**: Provides user feedback for actions such as creating, adjusting, or deleting annotations via the CustomSnackbar component.

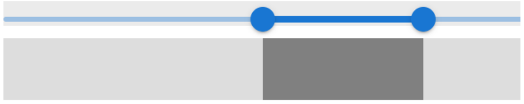


Block created

Blocks Component

The Block component represents annotations on the timeline, providing visual and interactive markers that the user can manipulate to annotate specific parts of the video.

- **Features**:
  - Interactivity: Blocks can be created with a double slider, allowing the user to adjust their start and end times directly on the timeline.



  - **Selection and Deletion**: Selecting a block reveals options for resizing or removing it, with deletions confirmed via the Snackbar.（Displayed in the timeline section）
  - **Color Coding**: Blocks are color-coded to indicate their selection status, enhancing visual clarity and user interaction.（Displayed in the timeline section）

CustomSnackbar Component

The CustomSnackbar provides essential feedback to users, confirming the results of their interactions within the Timeline. It plays a critical role in enhancing user experience by providing immediate, contextual information on actions performed.

- **Features**:
  - **Feedback on User Actions**: It displays notifications for various user actions like adding, deleting, or modifying Blocks. This helps confirm that the intended actions have been successfully completed.
  - **Error Handling**: The Snackbar also serves to inform the user of any errors or issues that might occur during interaction, ensuring clear communication of system states.（Refinement in sprint3）
  - **Customizable Appearance**: The duration and appearance of the Snackbar messages are customizable, allowing alignment with the overall design and user experience strategy of the application.（Refinement in sprint3）

Transcript Component

The Transcript component is an autoscroll screen that contains the transcript of the video. The transcript will scroll automatically in-sync with the video and adjusts to timeline changes accordingly.

- **Features**:
  - **Autoscroll**: The Transcript automatically scrolls according to the video timeline.
  - **Highlighted Current Line**: The line that is currently being spoken is highlighted in blue in accordance with the video player and moves onto the next line when finished.

Integrated Functionality

The design ensures seamless interaction between components:

Timeline and Blocks: The Timeline controls the rendering and manipulation of Blocks, handling their placement based on user actions and video playback.
Timeline and CustomSnackbar: The Timeline triggers the Snackbar for providing feedback related to annotation actions, ensuring users receive immediate and relevant feedback on their interactions.

Frontend Coding Standards

1. **Folder Structure**
    - **Directory Per Component:** Each React component should have its own directory with the same name as the component.
        - Example: `UserProfile/UserProfile.tsx`
    - **Test Files:** Place test files next to their corresponding component and use `.test.tsx` suffix.
        - Example: `UserProfile/UserProfile.test.tsx`
    - **Styles:** If using CSS modules, the styles file should also be named after the component.
        - Example: `UserProfile/UserProfile.module.css`

2. **File Naming Convention**
    - **Components:** Use PascalCase for React components. Files should match the name of the default export.
        - Example: `UserProfile.tsx`
    - **Hooks:** Prefix custom hooks with `use` and use camelCase.
        - Example: `useUser.ts`
    - **Utilities and Helpers:** Use camelCase for utility files.
        - Example: `arrayHelpers.ts`
    - **Type Definitions:** Use PascalCase and include `Type` suffix if it helps in clarity, otherwise just PascalCase.
        - Example: `UserProfileType.ts`, `Theme.ts`

3. **Function and Method Naming Convention**
    - **Regular Functions:** Use camelCase for function names. The name should start with a verb expressing what the function does.
        - Example: `calculateAge`, `fetchUserData`
    - **Event Handlers:** Start with `handle` followed by the action and the element.
        - Example: `handleButtonClick`, `handleChange`

3. **Functionality**
    - **Clarity and Descriptiveness:** Names should clearly and accurately describe their functionality or role.
    - **Consistency:** Apply naming conventions consistently across the project.
    - **Simplicity:** Prefer simplicity over complexity in naming. Avoid abbreviations unless widely understood.

Backend Documentation

The backend of our web-app is built upon Django, which enable rapid development with its extensive built-in features, scalability to handle increased data, and provides a solid foundation for our web-app development.

> ℹ Use this page to navigate through different part of backend documentations

💻 Backend Deployment Instruction A comprehensive guide for setting up the development environment and workflow for the backend of the project	Deployment Instructions
🔌 API Documentation A list of provided APIs with parameter requirements and response format.	API Documentation
🛠️ Function Documentation Functions used within each API are listed with its details.	Function Documentation
🗒️ Coding Standards Back-end development obeys certain coding standards, please read through this document before contributing to the project	Backend Coding Standards

Deployment Instructions

This document provides a comprehensive guide for running (or setting up) the development environment and workflow for the backend of our project. It includes instructions for preparing all the `ROS` dependencies, starting the backend `Django` server, developing within the environment, and placing `rosbag` files.

- The backend is running on Python version **3.9** & Django version **4.2**
- The `rosbag` analysis is done by using **ROS1 Noetic** and related Python packages provided by RoboStack. Please refer to the ROS Environment section for more details.
- Base image for Docker: `ubuntu:jammy` (Ubuntu 22.04)

Prerequisites

Docker Installation

- Before you begin, ensure that you have Docker installed on your machine. If not, download and install Docker from Docker's official site.
- Verify that the **Docker daemon** is running by executing `docker info` in your terminal. This should return information about the Docker client and server. If not, please start the Docker daemon.

Backend Directory Structure

- `ros_annotator/` : The main app with the same name as the project.
- `rosbag_processing/` : The app for ROSbag data processing tasks.
- `processed_data/` : This folder is used to store all processed files, including audio, video, waveform, and transcription.
- `rosbag-data/` : This folder is for user to store their ROSbag files.
- `manage.py` : The Django command-line utility.
- `env_config/` : This folder contains environment-specific configuration settings

Starting the Backend Server

To start the backend server, follow these steps:

1. **Navigate to the Backend Directory**

   Change to the backend folder by running:

```
cd src/backend
```

2. **Start the Server**

   Use Docker Compose to start the backend services:

```
docker-compose up --build
```

This command will build the image using `Dockerfile` from ground up and start all the required backend services as defined in the `docker-compose.yml` file. It also installs all the `ROS` dependencies.

Note: The image size is about 4.84 GB, so it may take a while for the first time to download and build the image. The speed depends on your computational resources and internet connection.

Use the below command to check the image and the status of running container:

```
# you should see a image with the name 'backend:ros-annotator'
docker image ls
# you should see a running container with the name 'ros-annotator'
docker ps
```

3. **Access the Backend Server**

Once the server is up and running, you can access the backend server at `http://localhost:8000/`. You should see the Django REST framework interface.

Note: The backend server is running in development mode, and the backend code base has been map to the container through Docker volume, so it will automatically reload whenever you make changes to the code.

Visit `http://localhost:5678/` for debugging purposes.

Development Setup Guide & Work Flow

To develop on the backend, proceed with the following steps (or follow the gif below for a visual guide):

1. **VSCode Extensions**: Ensure you have the Dev Container extension installed in your VSCode. The Dev Container extension allows you to develop inside a Docker container.
2. **Attach to the Running Container**: In VSCode, use the Dev Container extension to attach to the running Docker container. This can typically be done through the command palette (Ctrl+Shift+P or Cmd+Shift+P) and selecting "Attach to Running Container."
3. **Install Necessary Extensions in Dev Container**: Within the dev container environment in VSCode, install the Python (and Jupyter extensions if you are going to run Jupyter Notebook) to facilitate development. (*Note: This only need to be done once when you attach the container for the first time.*)

Using ROSbag Datas

To work with `rosbag` files:

- **Data Placement**: Place your rosbag data into the `src/backend/rosbag-data` folder. This location is set up to be accessible within the backend environment using docker volume.

API Documentation

Functionality APIs

1.Data Processing APIs (/rosbag)

Path	/process_rosbag
Type	POST

Request	`{`
	`  "bag_filename": "example.bag"`
	`}`

Response	On Success:
	`{`
	`  "video_data": "<base64_encoded_video_data>",`
	`  "audio_data": "<base64_encoded_audio_data>",`
	`  "waveform_image_data": "<base64_encoded_waveform_image_data>",`
	`  "message": "Processing complete"`
	`}`
	On Error:
	`{`
	`  "error": "No bag_filename provided"`
	`}`
	`{`
	`  "error": "Invalid request method"`
	`}`

Function Documentation

This document serves as a reference for the functions used within each APIs

API: /process_rosbag

1. **extract_images_from_rosbag**
   - **Purpose**: Extract images from a ROS bag file containing RGB image messages and save them to the specified output folder.
   - **Inputs**: `bag_filename` (str), `output_folder` (str)
   - **Outputs**: None
   - **Dependencies**: `cv2`, `Bag` from `rosbag.bag`, `CvBridge`
2. **extract_video**
   - **Purpose**: Convert extracted images into a video file.
   - **Inputs**: `bag_filename` (str), `output_folder` (str)
   - **Outputs**: None
   - **Dependencies**: `subprocess`, `ffmpeg`
3. **extract_audio**
   - **Purpose**: Extract audio data from the ROS bag file and save it as an MP3 file.
   - **Inputs**: `bag_filename` (str), `output_folder` (str)
   - **Outputs**: None
   - **Dependencies**: `Bag` from `rosbag.bag`
4. **plot_waveform**
   - **Purpose**: Plot the waveform of an audio file and save it as an image.
   - **Inputs**: `audio_path` (str), `output_filename` (str), `start_sec` (float, optional), `end_sec` (float, optional)
   - **Outputs**: None
   - **Dependencies**: `AudioSegment` from `pydub`, `matplotlib`
5. **combine_video_audio**
   - **Purpose**: Combine video and audio files into a single video file.
   - **Inputs**: `output_folder` (str)
   - **Outputs**: None
   - **Dependencies**: `subprocess`, `ffmpeg`
6. **generate_srt**
   - **Purpose**: Generate an SRT file containing speech transcripts with timestamps.
   - **Inputs**: `uri` (str), `output_folder` (str)
   - **Outputs**: `srt_file_path` (str)
   - **Dependencies**: `speech_v1p1beta1` from `google.cloud`, `format_time`
7. **format_time**
   - **Purpose**: Format a duration into HH:MM:SS,MMM format.
   - **Inputs**: `duration` (datetime.timedelta)
   - **Outputs**: Formatted time string (str)
   - **Dependencies**: None
8. **encode_file_base64**
   - **Purpose**: Encode a file as a base64 string.

- **Inputs**: `file_path` (str)
- **Outputs**: Encoded base64 string (str)
- **Dependencies**: `base64`

Backend Coding Standards

1. **Folder Structure**
    - All src files should reside in a `src/` directory
    - Create and manage the Django app based on functionality needed
    - Use descriptive folder names to represent the main functionality of the module.

2. **Naming Convention**
    - **Files**: snake_case
    - **Functions**: snake_case
    - **API endpoints**: snake_case

3. **Functionality**
    - Each function should only be responsible for a single task
    - Use descriptive function names that clearly convey their purposes
    - Keep solution simple

4. **Error Handling**
    - Return HTTP status codes consistent with the nature of the error (e.g., `400` for bad requests, `405` for invalid request methods)
    - Implement appropriate error handling mechanisms for all potential failure points, including file I/O operations and data processing

Technologies Used

1. **CVBridge**

A ROS (Robot Operating System) package that provides a bridge between ROS image messages and OpenCV image formats. It allows ROS nodes to efficiently convert image messages from ROS topics into OpenCV-compatible data structures, and vice versa. This enables seamless integration of ROS-based robotic systems with OpenCV-based computer vision algorithms and applications. CVBridge is commonly used for tasks such as image processing, object detection, navigation, and perception in robotic applications.

2. **FFmpeg**

A powerful multimedia framework that can decode, encode, transcode, mux, demux, stream, filter, and play almost any type of audio and video files. It is widely used for tasks such as video and audio editing, format conversion, streaming media, and batch processing.

3. **OpenCV**

A library of programming functions mainly aimed at real-time computer vision. It provides tools and algorithms for tasks such as image and video processing, object detection and tracking, facial recognition, and machine learning-based vision applications. OpenCV is widely used in various fields including robotics, augmented reality, medical imaging, and surveillance.

4. **Whisper**

A library for converting spoken language into text. It utilize machine learning algorithms to analyze audio data and transcribe it into written text. This is useful for application that need automatic transcription of audio recordings.

5. **Google Cloud Platform Services (Speech-to-Text)**

Another library for converting spoken language into text transcripts using machine learning algorithms.

Testing Documentation

1. Data Loading Test Cases

Test Case 1.1: Load a Single ROSBag

- **Description:** Verify the application can load a single ROSBag file.
- **Steps:**
  a. Open the application.
  b. Select a ROSBag file.
  c. Click "Load."
- **Expected Outcome:** The ROSBag data is loaded and displayed on the dashboard.

Test Case 1.2: Load Multiple ROSBags

- **Description:** Verify the application can load and manage multiple ROSBag files simultaneously.
- **Steps:**
  a. Open the application.
  b. Select multiple ROSBag files.
  c. Click "Load."
- **Expected Outcome:** All selected ROSBag files are loaded and displayed separately on the dashboard.

Test Case 1.3: Handle Corrupted ROSBag

- **Description:** Verify the application handles corrupted or invalid ROSBag files gracefully.
- **Steps:**
  a. Open the application.
  b. Select a corrupted ROSBag file.
  c. Click "Load."
- **Expected Outcome:** An appropriate error message is displayed, and the application remains stable.

2. Auto-Transcription Test Cases

Test Case 2.1: Transcription of Audio Stream

- **Description:** Verify the audio transcription feature works correctly.
- **Steps:**
  a. Load a ROSBag containing an audio stream.
  b. Click "Transcribe."
- **Expected Outcome:** The transcription text appears and aligns with the audio stream.

Test Case 2.2: Handle Missing Audio Stream

- **Description:** Verify the application handles missing audio streams gracefully.
- **Steps:**
  a. Load a ROSBag without an audio stream.
  b. Click "Transcribe."
- **Expected Outcome:** An appropriate error message is displayed.

3. Annotation Test Cases

Test Case 3.1: Manual Annotation

- **Description:** Verify manual annotation capabilities.
- **Steps:**
  a. Load a ROSBag with video/audio data.
  b. Select a time segment.
  c. Add a manual annotation.
- **Expected Outcome:** The manual annotation appears on the dashboard at the specified time.

Test Case 3.2: Automatic Annotation

- **Description:** Verify automatic annotation based on audio transcription.
- **Steps:**
  a. Load a ROSBag with an audio stream.
  b. Click "Transcribe" to generate the transcription.
  c. Click "Auto Annotate."
- **Expected Outcome:** The automatic annotations appear based on the audio transcription.

Test Case 3.3: Edit Annotations

- **Description:** Verify the ability to edit existing annotations.
- **Steps:**
  a. Add or auto-generate an annotation.
  b. Edit the annotation's content or time.
- **Expected Outcome:** The annotation is updated as expected.

4. Dashboard and Synchronization Test Cases

Test Case 4.1: Dashboard Visualization

- **Description:** Verify the dashboard displays synchronized visualizations.
- **Steps:**
  a. Load a ROSBag with audio, video, and other streams.
  b. Play the streams from the dashboard.
- **Expected Outcome:** All data streams are synchronized and visible on the dashboard.

Test Case 4.2: Synchronization of Annotations

- **Description:** Verify annotations synchronize with data streams.
- **Steps:**
  a. Add annotations to a loaded ROSBag.
  b. Play the data streams.
- **Expected Outcome:** The annotations appear at the correct times on the dashboard.

5. Security and Compliance Test Cases

Test Case 5.1: Role-Based Access Control

- **Description:** Verify only authorized users can access sensitive data.
- **Steps:**
  a. Log in as an unauthorized user.
  b. Attempt to access sensitive data.
- **Expected Outcome:** Access is denied, and an error message is displayed.

Test Case 5.2: Data Encryption

- **Description:** Verify data is encrypted in transit.

- **Steps:**
  a. Load a ROSBag file.

  b. Monitor the data being transmitted over the network.
- **Expected Outcome:** Data is encrypted during transmission.

Test Case 5.3: Consent Management

- **Description:** Verify consent management is respected.
- **Steps:**
  a. Load a ROSBag file with missing consent.

  b. Attempt to process the data.
- **Expected Outcome:** An error message is displayed, preventing the use of data without consent.

Code Review

This guideline assists in documenting automated code reviews effectively on Confluence. It ensures a comprehensive record of what code was reviewed, the criteria for selection, participants, findings, and follow-up actions.

Selection Criteria of Reviewed File

Files with the most centralised logic is selected for review, this includes:

1. Frontend components
2. Frontend pages
3. Backend views
4. Backend models

Code Review Sessions

2024-04-22 Code Review (Peer Review)

2024-05-01 Code Review (Frontend ChatGPT Review)

2024-05-02 Code Review (Backend ChatGPT Review)

2024-04-22 Code Review (Peer Review)

Date

22 Apr 2024

Participants

- @Tianqi Wang
- @yucpeng1
- @Harry Wang

Summary

In this peer-to-peer review, participants evaluated the usability and design of the frontend components of a video editing timeline interface. Key improvements were discussed and decisions made without the involvement of ChatGPT.

Topics Reviewed

Topic	Issue	Decision
Frontend “Timeline” Component	The timeline represents the video and needs to be easier for the user to scroll. Currently feels clumsy and too squished.	Use another UI design for the timeline such as a material-ui slidebar. The timeline length on the screen needs to be increased.
Frontend “Block” Component	The “Block” component needs to represent an interval on the timeline. It currently cannot have overlapping start and finish times.	Fix the issue of the block not being able to have overlapping start and finish times. The creation logic of the block needs to be modified to fit research needs.
Frontend “Create block” button	The create block button slides with the sliding timeline and creates an uneasy to use experience for clicking.	The “Create Block” button needs to be placed outside of the sliding window and be at a fixed position regardless of where the timeline is currently at. UI arrangement needs to be fixed on the “Timeline” component.

Decisions

- Ability to overlap block
- Fix bug so block can have the same start and end time
- Extend timeline length
- Place create block out of sliding timeline

2024-05-01 Code Review (Frontend ChatGPT Review)

Date

1 May 2024

Participants

- @Bowen Fan
- @yuchsong2

Pull Request

https://github.com/COMP90082-2024-SM1/ros-annotator/pull/4 Connect your Github account

Summary

In this review of the pull request, ChatGPT GitHub action is used to generate prompts during the review. The main inspected files are the files that contain our Components. We have decided that the most important and logic-intense portion of the code is within each frontend component and the component files ending with “.tsx” are in most need of an automatic review. More specifically, “Timeline.tsx”, “Block.tsx” and “Transcript.tsx” are reviewed in this pull request.

ChatGPT has spotted the lack of commenting in some vital areas and praised everything else in the frontend. No improvements are suggested in any other fields including Visual Representation, Structure, Resource, User Input, Logic and so on.

Topics Reviewed

Topic	Issue	Decision
Documentation and Comments	There appears to be not enough comments in Transcript.tsx and Block.tsx component according to ChatGPT.	While this is true in some areas of the components such as in some areas in Transcript.tsx, there is not enough supporting arguments to add more comments in Block.tsx as the code are all self-explanatory. Block is a simple rectangular component that represents an interval. We decided to add more commenting within Transcript.tsx and leave Block.tsx as is.
Testing and Edge Case Considerations	ChatGPT mentioned adding testing and edge case considerations to the components to ensure security.	Testing is done seperately from the component files, which is not sent to the ChatGPT prompt with this Github Action. For now, this suggestion is taken in mind and safely set aside.

Decisions

Add more comment in Transcript.tsx

Review testing considerations

💡 Conclusion

While suggesting about commenting and documentation helped us realise some areas of improvements in our codebase, ChatGPT seems to not be able to provide any other suggestions for this frontend review specifically, except for some generic prompts. At the frontend, more manual peer-to-peer reviewed is definitely required to make further improvements in the future, much like the 22/04 review we conducted earlier.

2024-05-02 Code Review (Backend ChatGPT Review)

📅 Date

2 May 2024

👥 Participants

- @Yujie Zheng
- @Harry Wang

🔗 Pull Request

https://github.com/COMP90082-2024-SM1/ros-annotator/pull/5 Connect your Github account

ℹ️ Summary

In this review of the pull request, ChatGPT GitHub action is used to generate prompts during the review. The main inspected files are the files that contain our Views. We have decided that the most important and logic-intense portion of the code is within “views.py” file that contains most of the logic and functions.

🗣️ Topics Reviewed

Topic	Issue	Decision
File Error Handling	“Ensure that error handling mechanisms are in place for potential exceptions during file processing.”	Add error handling “try” and “except” clauses when handling files, including opening “.ros”, “.mp3”, “.txt” files and so on when processing the intermediate files.
Error Messages	“Consider adding more detailed error messages in the response to provide better feedback to users.”	Following the above topic and issue, add detailed print and error logging messages when encountering an error.
Code Refactoring	“Refactor the code to separate concerns and improve modularity for better maintainability.”	We discussed the possibility of separating the functions into seperate files for ease of reading. However, for now, there is not enough files for this to be practical. File refactoring will be set aside until more content is added.
Documentation and Comments	“Add more detailed documentation to explain the purpose of each function and the overall workflow of the code.”	The documentation on the workflow is not present in the code and is put onto Confluence. However, suggesting that each function to be more detailed is a right direction. Next, we will add docstrings to the most important functions to give more detailed description and comment.

⤴️ Decisions

Add error handling when opening files

- Log detailed error messages
- Consider refactoring code into different files in later sprints
- Add docstrings to important functions

Decision Making

Decision	Date
Decision on Speech-To-Text problem TW Owned by Tianqi Wang • Updated on 10 Apr 2024 Status Complete Impact High Driver @yucpeng1 Approver client Contributors @Bowen Fan @yucpeng1 @Tianqi Wang Informed client Due date Apr 10, 2024 Resources https://comp90082-2024-na-… Confluence Open preview	10 Apr 2024
Decision on Predefined Booklist TW Owned by Tianqi Wang • Updated on 24 Apr 2024 Status In progress Impact High Driver @Yujie Zheng Approver Contributors Informed client Due date Apr 23, 2024 Resources \uD83D\uDCD8 Background The client requests a pre-defined booklist for the annotation task. However, as … Confluence Open preview	23 Apr 2024
Decision on intro page TW Owned by Tianqi Wang • Updated on 22 Apr 2024 Status Not started / In progress / Complete Impact High / Medium / Low Driver Approver Contributors Informed Due date Resources \uD83D\uDCDA Relevant data \uD83D\uDCD8 Background \uD83C\uDF08 Options considered Option … Confluence Open preview	

Decision log

Create decision

Decision	Status	Stakeholders	Outcome	Due date	Owner
Decision on Predefined Booklist	IN PROGRESS			23 Apr 2024	
Decision on intro page	NOT STARTED / IN PROGRESS / COMPLETE				
Decision on Speech-To-Text problem	COMPLETE			10 Apr 2024	

Decision on Speech-To-Text problem

Status	COMPLETE
Impact	HIGH
Driver	@yucpeng1
Approver	client
Contributors	@Bowen Fan @yucpeng1 @Tianqi Wang
Informed	client
Due date	10 Apr 2024
Resources	2024-04-10 Meeting notes

📚 Relevant data

Speech-to-Text AI: speech recognition and transcription

Introducing Whisper

📘 Background

This decision is about the speech-to-text requirement. The client requires the transcript for the video and would like to separate the speaker in the transcript. We tried out 2 ways of STT API, Google Cloud and OpenAI whisper. They have their pros and cons. After meeting with the client, we decided to stick with the whisper.

🌈 Options Considered

	Option 1	Option 2
Description	**Using Google Cloud API to do STT task.**	**Using OpenAI Whisper API to do STT task.**
Pros and cons	⊕ Can separate speaker. ⊖ Can not hear the robot. ⊖ Accuracy is low.	⊕ Accuracy is high ⊕ Can hear the robot clearly ⊖ Can not separate speaker
Estimated cost	MEDIUM	MEDIUM

✅ Action items

☑ Discuss with the client which option they prefer.

🌟 Outcome

Our development group will use Whisper API to do the STT task. Meanwhile, we will keep looking for better tools to separate the speaker.

On the other hand, the client provided an idea to polish and enhance the voice of the robot and to see if Google Cloud will work, this will be tested during development.

Decision on Predefined Booklist

Status	IN PROGRESS
Impact	HIGH
Driver	@Yujie Zheng
Approver	
Contributors	
Informed	client
Due date	23 Apr 2024
Resources	

📘 Background

The client requests a pre-defined booklist for the annotation task. However, as a web app without a database, we have no ability to save data. As a result, the pre-define booklist can either be uploaded or manually imported by the user in the format of special rules(for example CSV).

🌈 Options considered

	Option 1 upload and download booklist	**Option 2 manually imported in text editor**
Description	In this option, users are provided with the ability to upload a file containing their predefined annotation booklist. This file can be in a variety of formats, such as CSV, JSON, or XML, which the web app will parse to utilize the contained data. Users can also download their current booklist in one of these formats, which they can edit offline and re-upload at their convenience.	This option involves a text editor integrated into the web application, where users can directly type their annotation booklist using a specific format, such as CSV. This format will allow the app to differentiate between various entries properly. The use of a text editor simplifies direct manipulation of the booklist on the web interface itself without the need for handling files.
Pros and cons	⊕ Users can transfer their booklists between different systems or back them up locally. ⊖ Users are responsible for maintaining their own files, which could lead to issues if files are lost or corrupted.	⊕ Eliminates the need for users to manage physical files, reducing the risk of data loss due to file corruption or misplacement. Users interact with a straightforward text-editing environment, which is easier to use. ⊖ As the size of data grows, the text editor might become cumbersome to manage,

	Handling file uploads and downloads adds additional complexity to the web app's backend, requiring more robust error handling and security measures.	making it harder to navigate and edit extensive lists.
Estimated cost	LARGE	SMALL

✅ Action items

- □ Discuss with the client during the meeting on April 23th.

🌟 Outcome

Option 2 is good enough.

Decision on intro page

Status	NOT STARTED / IN PROGRESS / COMPLETE
Impact	HIGH / MEDIUM / LOW
Driver	
Approver	
Contributors	
Informed	
Due date	
Resources	

📚 Relevant data

📘 Background

🌈 Options considered

	Option 1	Option 2
Description		
Pros and cons	⊕ ⊖	⊕ ⊖
Estimated cost	LARGE	MEDIUM

✅ Action items

□

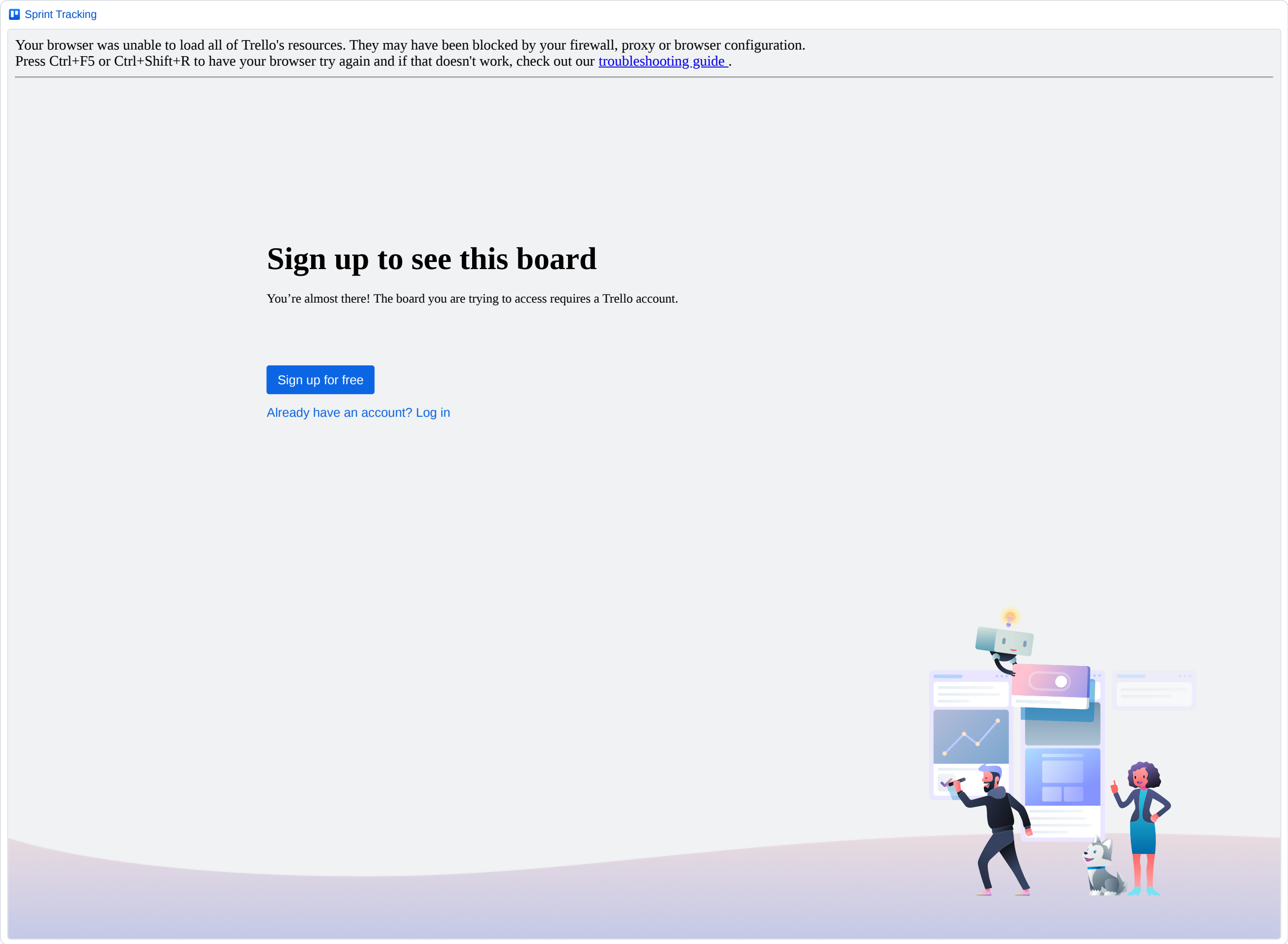
🌟 Outcome

🚀 Sprint Management with Trello Board

Sprint Task Tracking (Trello Board)

Note:

- *For each task breakdown, please read the ID as:* ***"Epic Id . UId . Taks Id"*** *(where UId is the user story Id. Please refer to the User Story Table)*
- *The Epic assignment is not displayed properly in the Confluence embedded link, please directly visit our Trello board to view the Epic assignment.*

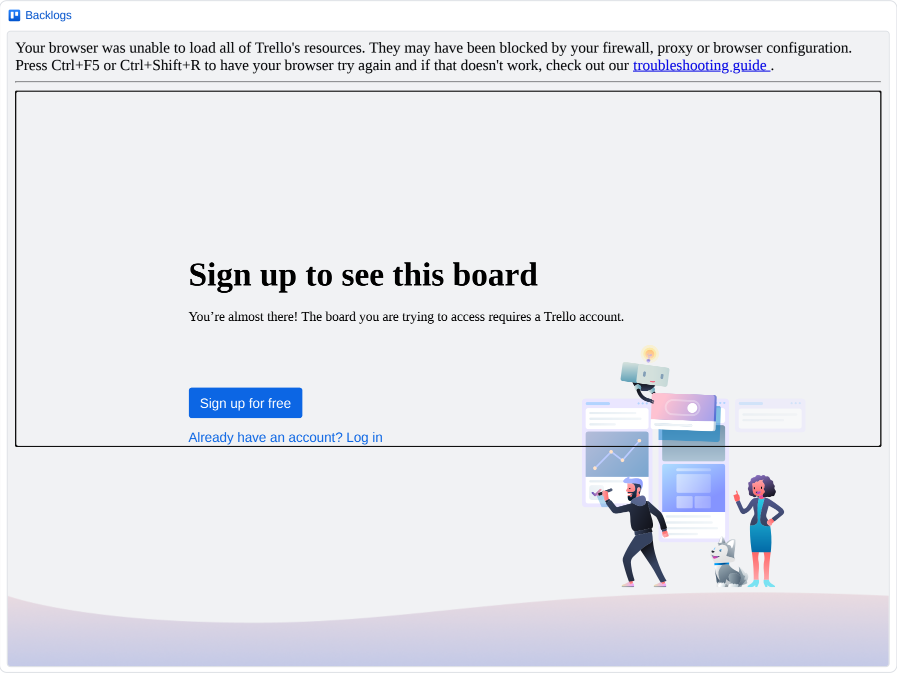


Your browser was unable to load all of Trello's resources. They may have been blocked by your firewall, proxy or browser configuration.
Press Ctrl+F5 or Ctrl+Shift+R to have your browser try again and if that doesn't work, check out our troubleshooting guide .

Sign up to see this board

You're almost there! The board you are trying to access requires a Trello account.

Backlog & Epics



Backlogs

Your browser was unable to load all of Trello's resources. They may have been blocked by your firewall, proxy or browser configuration. Press Ctrl+F5 or Ctrl+Shift+R to have your browser try again and if that doesn't work, check out our troubleshooting guide .

Sign up to see this board

You're almost there! The board you are trying to access requires a Trello account.

Sprint 1 Deliverables

Sprint 1

TW Owned by Tianqi Wang • Updated on 21 Mar 2024

Sprint 1 check-list https://comp90082-2024-na-koala.atlassian.net/wiki/spaces/comp900822/pages/3014657 Sprint 1 task distribution https://comp90082-2024-na-koala.atlassian.net/wiki/spaces/comp900822/pages/6324238 Sprint 1 Retrospective Summary https://comp90082-2024-na-koala.atlassian.net/wiki/spaces/comp900822/pages/6160408

Confluence

Open preview

Sprint 2 Deliverables

Sprint 2

YZ Owned by Yujie Zheng • Updated yesterday

Sprint 2 Planning https://comp90082-2024-na-koala.atlassian.net/wiki/spaces/comp900822/pages/7864332 Sprint 2 check-list https://comp90082-2024-na-koala.atlassian.net/wiki/spaces/comp900822/pages/16482305 Sprint 2 Retrospective summary https://comp90082-2024-na-koala.atlassian.net/wiki/spaces/comp900822/pages/28835851

Confluence

Open preview

Sprint 1

Sprint 1 check-list

Sprint 1 check-list

TW Owned by Tianqi Wang • Updated on 21 Mar 2024

Sprint 1 (checklist) This checklist helps you double-check your work for Sprint 1. ## Background description, client goals, motivation [✅] Project overview, background and goals were created. (not mandatory but recommended for all projects - these resources would help with the design sprint and project overview) [✅] DO-BE-FEEL list and GOAL

Confluence Open preview

Sprint 1 task distribution

Sprint 1 task distribution

TW Owned by Tianqi Wang • Updated on 19 Mar 2024

User stories, use cases, Personas. @Harry Wang Confluence: @yuchsong2 @yucpeng1 meeting notes Sprint planning/retrospective Daily Standups Project overview, project goal, DO-BE-FEEL list, goal model @Tianqi Wang GitHub structure and Readme. @Bowen Fan Trello board. @Yujie Zheng

Confluence Open preview

Sprint 1 Retrospective Summary

Sprint 1 Retrospective Summary

Y Owned by yuchsong2 • Updated on 19 Mar 2024

@Tianqi Wang @Yujie Zheng @Bowen Fan @Harry Wang @yuchsong2 @yucpeng1 @Jinhao He Sprint Number : 1 Duration : 4 March 2024 - 22 March 2024 What went well? Teamwork : Collaboration and support among team members facilitates the smooth running of the work. Task completion was good : the sprint completed all planned tasks and team members performed we

Confluence Open preview

Sprint 1 Planning: Inception

Sprint 1: Requirement Analysis and Planning Phase

1. **Gather Detailed Requirements and Specifications**
   - Meetings with PhD students for requirements gathering.
   - Identifying user stories based on these requirements.
2. **Survey of Existing Annotation Tools**
   - Research and document features of existing tools (e.g., Elan).
   - Identify innovative features and best practices that can be integrated.

Sprint 1 Tasks

For Sprint 1, the focus is on the initial phase of Requirement Analysis and Planning. Here's a detailed breakdown of tasks:

- **Task 1: Initial Team Meeting**
  - Objective: Align the team on the project's goals and methodologies.
  - Duration: 1 day
- **Task 2: Requirement Gathering Meetings**
  - Objective: Conduct meetings with PhD students to gather detailed requirements.
  - Duration: 1 week
  - Deliverable: A requirements document outlining user needs and expected features.
- **Task 3: Research on Existing Tools**
  - Objective: Survey existing annotation tools and document their features.
  - Duration: 1 week
  - Deliverable: A comparison report highlighting potential features for ROSAnnotator.

Sprint 1 check-list

Sprint 1 (checklist)

This checklist helps you double-check your work for Sprint 1.

Background description, client goals, motivation

- [✅] Project overview, background and goals were created.

(*not mandatory but recommended for all projects - these resources would help with the design sprint and project overview*)

- [✅] DO-BE-FEEL list and GOAL MODEL were created.
- [✅] The goal model is consistent with the client understanding of the problem and with DO-BE-FEEL list.
- [✅] Personas are based on the research done by students and the discussion with industry partners.
- [✅] Personas are inclusive and diverse.

Analysis of requirements (User Stories or Use Cases)

- [✅] The analysis of requirements was performed on most of the existing requirements.
- [✅] The [new set of] requirements is consistent to the scope of the project, completely cover the new capabilities required by the client and are well documented/structured/organised on Confluence.
- [✅] The requirements can be documented in the form of user stories or use cases, supplementary specification of design/implementation/deployment requirements, prototypes, and others. It may also be necessary to be explicit about what is not in scope to define the scope boundary more clearly.
- [✅] We used ChatGPT to generate user stories to our project. On Confluence Space, we documented the prompt we've used, what user stories were generated WITH and WITHOUT ChatGPT.

Development environment

- [✅] Confluence is organised (cover page, project details, requirements, technical details about the project, meeting minutes and so on).
- [✅] Trello (or Github projects or JIRA) is created, structured and organized.
- [✅] Previous/existing project is deployed and could be used/tested as part of this requirements engineering phase.
- [✅] README file is updated and provide details about the project, workflow (branches/naming conventions and so on).

Plan

- [✅] A plan (or discussion on what to do next) was provided (requirements to develop, technologies to use, infrastructure to deploy the project) for Sprint 2 and Sprint 3.
- [✅] Requirements were estimated and prioritised.
- [✅] Backlog items can be found in Trello (or Github project or JIRA).

Meetings

- [✅] Meetings are recorded in Confluence and only. They were NOT exported to Github as they're part of internal process.

GitHub

- [✅] Folders are structured (On Canvas, visit Assignment -> "Sprint 1: Confluence Space, project background and elicitation documents" page: you can find requirements for folders' structure.)

- [✅] Sprint 1 documents were exported from Confluence and added to the repository (and are updated)
- [✅] README file is updated and explain the team's repository
- [✅] A baseline tag was generated for this Sprint (On Canvas, visit Assignment -> "Sprint 1: Confluence Space, project background and elicitation documents" page: you can find requirements for the baseline tag)

Additional Information

do you have any other additional information you'd like to share with us? Please add it here.

Sprint 1 task distribution

1. User stories, use cases, Personas. @Harry Wang
2. Confluence: @yuchsong2 @yucpeng1
   a. meeting notes
   b. Sprint planning/retrospective
   c. Daily Standups
3. Project overview, project goal, DO-BE-FEEL list, goal model @Tianqi Wang
4. GitHub structure and Readme. @Bowen Fan
5. Trello board. @Yujie Zheng

Sprint 1 Retrospective Summary

- @Tianqi Wang
- @Yujie Zheng
- @Bowen Fan
- @Harry Wang
- @yuchsong2
- @yucpeng1
- @Jinhao He

Sprint Number: 1

Duration: 4 March 2024 - 22 March 2024

What went well?

- **Teamwork**: Collaboration and support among team members facilitates the smooth running of the work.
- **Task completion was good**: the sprint completed all planned tasks and team members performed well on the tasks.
- **Active participation in the discussion**: the group members were actively involved in the discussion, presenting different points of view and discussing them.
- **Efficient completion of tasks**: everyone completed their part of the task efficiently.

Areas for improvement

- **Scheduling of meetings**: Meetings were not scheduled appropriately and there were occasional absences.

action plan

Adjust Meeting Times: Re-evaluate and adjust the scheduling of team meetings to reduce absences, and emphasize the meeting time in Slack to ensure that all team members receive the notification.

Summary

The team excelled in communication, task completion and quality control in this sprint. The active participation of members and efficient execution were key to this success. However, the meeting schedule needs to be improved to ensure that all members can attend. In the next sprint, we will adjust the meeting times to better accommodate team members' schedules.

Sprint 2

Sprint 2 Planning

Sprint 2 Planning: Development

YZ Owned by Yujie Zheng • Updated 2 days ago

Sprint 2 Planning: Development (March 25 - April 26) Overview Sprint 2 will span from March 25 to April 26, focusing on enhancing the core functionality and user interface of our audio-to-transcript application with speaker separation. This period aims to solidify our backend infrastructure, streamline front-end interactions, and explore advanced d

Confluence

Open preview

Sprint 2 check-list

Sprint 2 checklist

TW Owned by Tianqi Wang • Updated 2 hours ago

Each group is to submit ONE (and only) checklist here on Canvas. Team-based submission (not individual-based). ✅ ## Confluence (infra) Make sure that your Confluence satisfies the following criteria: [✅] The students have produced an excellent structure of the project on Confluence. [✅] Easy to find contents on pages. [✅] Most of the conte

Confluence

Open preview

Sprint 2 Retrospective summary

Sprint 2 Retrospective

TW Owned by Tianqi Wang • Updated 2 days ago

@Tianqi Wang @Yujie Zheng @Bowen Fan @Harry Wang @yuchsong2 @yucpeng1 @ABHISHEK TUMMALAPALLI Sprint Number : 2 Duration : 29 March 2024 - 2 May 2024 What went well? Effective Communication : Clear and open communication among team members enhanced understanding and coordination throughout the project. High-Quality Deliverables : All output met or

Confluence

Open preview

Sprint 2 Planning: Development

Sprint 2 Planning: Development (March 25 - April 26)

Overview

Sprint 2 will span from March 25 to April 26, focusing on enhancing the core functionality and user interface of our audio-to-transcript application with speaker separation. This period aims to solidify our backend infrastructure, streamline front-end interactions, and explore advanced deployment options, ensuring a robust foundation and an intuitive user experience.

Justification

The enhancement of the core functionality and user interface in this sprint is essential for meeting the evolving demands of our target users, who require a robust and intuitive audio-to-transcript application. The introduction of speaker separation is particularly vital for contexts where multiple speakers are present, such as meetings, interviews, and conferences, ensuring that the transcription accurately reflects who said what. This feature not only enhances user satisfaction but also increases the usability of the application in professional and academic settings.

Moreover, the development of a comprehensive architecture diagram and the careful planning of backend and frontend integrations are designed to minimize potential technical debt and streamline future development efforts. This strategic approach allows for more predictable scaling and maintenance of the application as user demand grows. Early focus on these areas will prevent costly reworks and ensure that the product can efficiently handle increased workloads as it matures.

The exploration of advanced deployment options like ROS in Docker containers highlights our commitment to leveraging cutting-edge technology to enhance our application's capabilities. While not immediately critical, this investigation is an investment in future-proofing the application and exploring possibilities that could set our product apart from competitors.

Objectives and Key Deliverables

1. **Audio to Transcript with Speaker Separation**: Prioritize the development of a reliable and accurate system for converting audio to text while identifying and separating speakers. This critical functionality forms the backbone of our application, necessitating an early sprint focus to allow ample time for integration and testing.
2. **Architecture Diagram Creation**: Early establishment of a comprehensive architecture diagram is crucial. It will guide the development process, ensuring all team members understand the system's structure, components, and how they interact. This clarity will facilitate more efficient development and integration across all parts of the project.
3. **Backend Integration for Extracted Code**: Organize the backend to handle audio processing and transcript generation efficiently. This involves creating a scalable and secure architecture that can manage heavy workloads and ensure smooth communication with the front end.
4. **Frontend Prototype Design**: Designing an intuitive and functional frontend prototype is essential for user interaction. This task will follow the backend setup to align the design with the backend capabilities, focusing on displaying the transcript effectively and allowing users to navigate the audio and text seamlessly.
5. **Transcript Scrolling Display with Timeline Jump**: Implement a user-friendly feature for scrolling through the transcript and jumping to specific video timelines. This feature enhances the user experience by providing efficient navigation and interaction with the content.
6. **Video Player Subtitle Toggle**: Add a feature in the video player to toggle subtitles on and off, generated from the transcript. This functionality must be user-centric, offering flexibility in how users engage with the video content.
7. **Exploration of ROS Docker**: Investigate the implementation of ROS (Robot Operating System) within Docker containers. This exploration, while not immediately critical, could provide valuable insights into improving processing capabilities or simulation environments in the future.
8. **Story Point Replan**: Midway through the sprint, reassess and replan story points for tasks based on progress and any unforeseen challenges. This adjustment ensures that the team remains on track and that priorities are correctly aligned with project goals.

Sprint Schedule and Milestones

- **Week 1 (March 25 - March 31)**: Kick-off with audio to transcript development and start architecture diagram. Research speech recognition and speaker diarization technologies.
- **Week 2 (April 1 - April 7)**: Continue backend integration and finalize architecture diagram. Begin frontend prototype design.
- **Week 3 (April 8 - April 14)**: Complete frontend design and start implementation. Initiate transcript scrolling and video subtitle toggle features.
- **Week 4 (April 15 - April 21)**: Conduct ROS Docker exploration. Implement and integrate backend and frontend functionalities. Start comprehensive testing of all features.
- **Week 5 (April 22 - April 26)**: Final adjustments, additional testing, and story point replan. Prepare for end-of-sprint review, focusing on lessons learned and planning for the next development phase.
- **Week 6 (April 29 - May 2): Documentation and Closure:** Extra week for documentation. Detailed documentation of all features developed during the sprint, updates to the architecture diagram, and preparation of end-of-sprint materials.

Sprint 2 checklist

Each group is to submit ONE (and only) checklist here on Canvas. Team-based submission (not individual-based).

- ✅

Confluence (infra)
Make sure that your Confluence satisfies the following criteria:

- [✅] The students have produced an excellent structure of the project on Confluence.
- [✅] Easy to find contents on pages.
- [✅] Most of the contents are visible and editable (no unnecessary attachments).

Confluence (contents and consistency)
Make sure that your Confluence satisfies the following criteria:

- [✅] Contents are available and updated on Confluence (meeting minutes, scope of the project, diagrams, technologies used in the project, user stories, test cases).
- [✅] Contents are consistent with trello (or github project) and with their code repositories.

Task Tracking
Make sure that your task tracking satisfy the following criteria:

- [✅] Students organized a product backlog and a lower-level sprint backlog.
- [✅] Tasks in the sprint backlog were estimated, have an appropriate due date and have a sufficiently low level of granularity.
- [✅] Tasks are also clear, linked to their user stories (Confluence) and offer additional description when necessary.
- [✅] A link to your Trello or JIRA (Anyone with link can access that resource) was made available to your marker.

Code Review (Sprint 2)
Make sure that your code review satisfy the following criteria:

- [ ] Students documented their peer-to-peer or chatgpt code review on GitHub (pull request comment)[THIS ITEM IS OPTIONAL].
- [✅] Students documented their peer-to-peer or chatgpt code review on Confluence (new page on Confluence to document how you performed your code review - who participated in that, when did that happen, number of issues identified and so on).
- [ ] In case you used ChatGPT in this sprint, please disconnect your GitHub repository from our ChatGPT Code Review one so you don't get charged in the future (or, make sure you continue to use this carefully and only when strictly necessary - do not use it for all commits)

Sprint Planning and Review
Make sure that your sprint planning and review are documented on Confluence:

- [✅] Clear indication that sprint planning was followed this sprint and a clear, consistent, updated sprint planning for Sprint 3.
- [✅] Clear indication that sprint review was followed this sprint. Team organised a meeting for this, documented discussions, reflections and next steps to be taken in Sprint 3.

Ethical Considerations
Make sure that your ethical considerations are documented on Confluence:

- [✅] Clear indication that students discussed/reflected on project ethical issues.

Cyber Security
Make sure that your cyber security considerations are documented on Confluence:

- [✅] Clear indication that students discussed/reflected on project cyber security issues.

Product
Make sure that your product satisfy the following criteria:

- [] Product is deployed and an URL is available on Confluence and Github README so client can access current version of this software.
- [✅] FOR PROJECTS YOU CANT DEPLOY IT NOW: Can you emulate the project and demonstrate current progress for us in a short recorded video? we need to be able to measure your development progress in Sprint 2, that's all.

Meetings
Make sure your meetings (team meetings, supervision meetings and meetings with industry partners) are documented in Confluence (and only).

- [✅] Meetings are recorded in Confluence and only. They were NOT exported to Github as they're part of an internal process.

GitHub
Make sure that:

- [✅] Folders are structured.
- [✅] Sprint 2 documents were exported from Confluence and added to the repository (and are updated)
- [✅] README file is updated and explain the team's repository and new release
- [✅] A baseline tag was generated for this Sprint

Additional Information

do you have any other additional information you'd like to share with us? Please add it here.

Sprint 2 Retrospective

- @Tianqi Wang
- @Yujie Zheng
- @Bowen Fan
- @Harry Wang
- @yuchsong2
- @yucpeng1
- @ABHISHEK TUMMALAPALLI

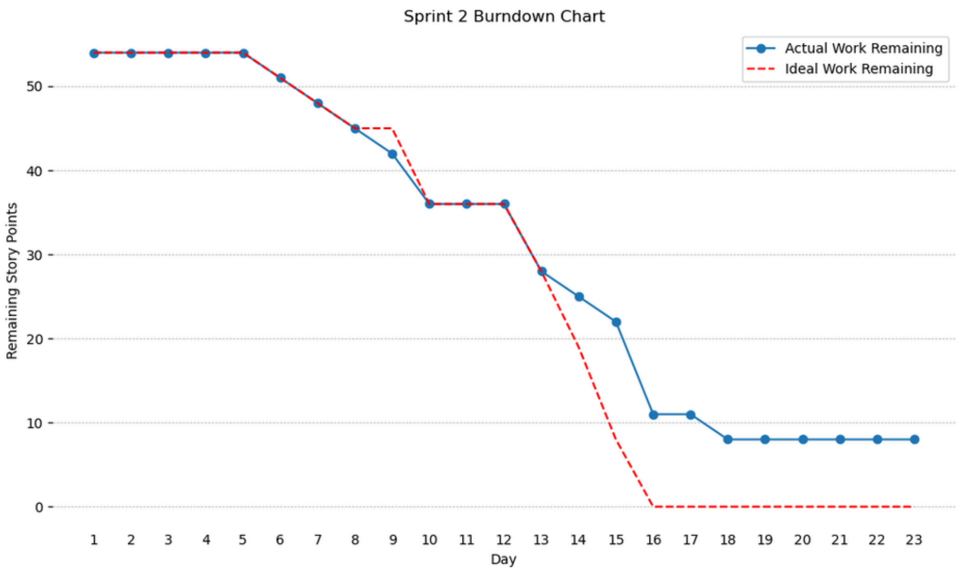
Sprint Number: 2

Duration: 29 March 2024 - 2 May 2024

What went well?

- **Effective Communication**: Clear and open communication among team members enhanced understanding and coordination throughout the project.
- **High-Quality Deliverables**: All output met or exceeded the expected standards, reflecting the team's dedication and expertise.
- **Engagement and Enthusiasm**: Team members showed high levels of engagement and enthusiasm, contributing positively to the team atmosphere.
- **Problem Solving**: Team members effectively addressed and resolved challenges as they arose, maintaining project momentum.

Sprint burndown chart



Areas for improvement

- **Task Completion Timeliness**: There were delays in completing some tasks, which impacted the overall project timeline.
- **Unfinished Tasks**: A few tasks remained incomplete by the end of the sprint, needing reassignment or additional resources to ensure completion.
- **Feedback and Revisions**: The process for gathering feedback and implementing revisions could be streamlined to accelerate project advancement.
- **Resource Allocation**: There were occasional mismatches in resource allocation, which affected the efficiency of task execution.

Action plan

- **Enhanced Project Tracking**: Implement a more robust project management tool to monitor task progress and deadlines more effectively. Ensure that all team members are trained on how to use this tool efficiently to keep tasks on track and address delays promptly.

Summary

During the second sprint, our team excelled in communication, quality of deliverables, engagement, and problem-solving, which were key contributors to the project's overall progress. However, we faced challenges with task completion timeliness, incomplete tasks, and inefficient resource allocation, as indicated by our burndown chart. Moving forward, we plan to implement a robust project management tool to enhance tracking and improve task completion rates, aiming to streamline our processes for better efficiency and output in future sprints.

Sprint 3

Temporary Tasks

1. Predefine booklist
    a. Create booklist
    b. Edit booklist
    c. Analyse booklist
2. Annotation task
    a. C

Sprint 3 Planning: Development

Objective: Focus on refinement, and addressing any identified issues or improvements.

Duration: 4 weeks

1. **Integration with Auto-Transcription**
   - Integrate auto-transcription service for audio to text conversion.
   - Develop annotation functionalities.
   - Duration: 1 Week
2. **Annotation Features**
   - Develop custom scales and annotation tools for users.
   - Duration: 2 Week
3. **Automatic Annotation Exploration**
   - Research and prototype automatic annotation methods.
   - Duration: 1 Week
4. **Front-End Refinement**
   - Add “login page“
   - Gather feedback on the front-end design from users or stakeholders.
   - Implement improvements and refinements to enhance user experience and usability.
   - Duration: Throughout Sprint 3
5. **Testing**
   - Conduct thorough testing, including unit tests, integration tests, and user acceptance tests.
   - Address any issues or bugs identified during testing.
   - Duration: 1 Week

Cybersecurity Considerations

As part of our commitment to integrating robust cybersecurity measures within the ROSAnnotator project, this report outlines key initiatives designed to strengthen the project's security posture. We focus on developing a project-specific threat model, documenting secure design practices, and detailing our data protection strategies. These initiatives are essential to protecting sensitive data, ensuring system integrity, and maintaining user privacy.

Project-Specific Threat Model

The development of a detailed, project-specific threat model is crucial for understanding and mitigating potential security risks associated with the ROSAnnotator project. This model allows us to identify specific threats, assess their potential impact and likelihood, and implement appropriate mitigation strategies. Below is a table that outlines the primary threats identified, along with their respective characteristics and mitigation strategies:

Threat Model Table:

Threat	Description	Impact	Likelihood	Mitigation Strategy
Unauthorized Data Access	Unauthorized access to sensitive video and audio data	High	Medium	Implement robust access controls and MFA
Data Corruption	Accidental or malicious alteration of data	Medium	Low	Use checksums and data integrity checks
Privacy Breaches	Exposure of personal information from datasets	High	Medium	Data anonymization and use of encryption
Injection Attacks	SQL injection, XSS, etc., compromising system integrity	High	High	Employ secure coding practices and input validation

Secure Design Documentation

The secure design of the ROSAnnotator project is documented comprehensively to outline how security considerations are integrated into the architecture. This documentation includes:

- **Architecture Diagrams**: Visual diagrams that detail the secure architecture of the system, highlighting critical security controls at each layer of the application.
- **Secure Coding Practices**: A list of secure coding practices adopted by the development team, such as input validation, use of parameterized queries, and comprehensive error handling to prevent common vulnerabilities like SQL injection and cross-site scripting (XSS).

Data Protection Strategies

The following strategies have been implemented to protect the integrity, confidentiality, and availability of data within the ROSAnnotator project:

- **Encryption Protocols**: Specifications of encryption protocols used for securing data at rest and in transit, such as TLS for data transmission and AES for data storage.

- **Access Control Policies**: Detailed descriptions of role-based access control (RBAC) policies that ensure only authorized personnel can access specific types of data.
- **Data Integrity Mechanisms**: Implementation of mechanisms like checksums and digital signatures to maintain and verify data integrity.

Conclusion

The cybersecurity enhancements outlined in this report are tailored specifically to the needs of the ROSAnnotator project. By implementing these targeted strategies, we not only enhance the security of the system but also build trust with our users and stakeholders. Regular updates and revisions to our security documentation on Confluence will continue to reflect the evolving nature of our project and the cybersecurity landscape.

Ethical Considerations

The ROSAnnotator project, designed to enhance the analysis of ROSBag data for Human-Robot Interaction (HRI), presents unique ethical challenges that must be addressed to ensure the responsible handling of sensitive data, equitable access, and unbiased analysis. This report outlines the specific ethical issues identified within the project, alongside detailed strategies for their resolution and ongoing management.

Specific Ethical Issues

Privacy and Confidentiality: The project processes potentially sensitive data, including video and audio recordings from HRI environments. Such data can include personal identifiers or sensitive personal interactions, raising significant privacy concerns.

Bias in Automated Systems: The application's reliance on auto-transcription tools and facial recognition technologies poses risks of bias, potentially skewing the analysis of HRI data if the underlying models are not representative of diverse populations.

Informed Consent: Ensuring that individuals whose data is captured and analyzed have provided informed consent is critical, especially given the complex environments in which HRI typically occurs.

Reliability and Accountability: The integrity of data processing and annotation within ROSAnnotator is vital. Errors in data interpretation could misrepresent human behaviors, impacting HRI research and applications.

Access and Equity: It is essential that ROSAnnotator is accessible to all potential users, including those with disabilities, and that it does not create or perpetuate inequalities among user groups.

Resolution Strategies

Enhanced Data Protection Measures: To safeguard privacy, the project implements strict data encryption, anonymizes personally identifiable information where possible, and uses secure data transfer protocols. Access controls are rigorously applied to ensure that only authorized personnel can access sensitive data.

Bias Mitigation: Continuous efforts are made to ensure that the auto-transcription and facial recognition algorithms are free from biases. This involves regular updates, training on diverse datasets, and conducting fairness audits. Documentation of all algorithmic processes is maintained for transparency and is available for review on Confluence.

Robust Consent Management: A consent management framework has been developed within the application, ensuring clear communication of data use policies and easy consent withdrawal mechanisms. This framework is crucial for maintaining trust and ethical compliance in dynamic interaction settings.

Accountability Mechanisms: The application features comprehensive logging and error tracking to swiftly identify and correct any issues in data processing. A feedback mechanism is also in place, allowing users to report inaccuracies or ethical concerns, which the development team promptly addresses.

Ensuring Inclusivity: The development adheres to web accessibility guidelines, ensuring that ROSAnnotator is usable by a diverse audience. Ongoing user experience research helps to break down barriers to access, ensuring the application benefits all users equally.

Team Commitments and Documentation

The development team is committed to regular ethical training to reinforce the importance of ethical design, privacy protection, and unbiased development. Ethical considerations, data handling protocols, and resolution strategies are meticulously documented on Confluence. This documentation is reviewed and updated each sprint to reflect new developments or changes in ethical guidelines. Ethical reviews are incorporated into sprint retrospectives, allowing for continual assessment and improvement of ethical integration in the development process.

Conclusion

The ethical framework established for the ROSAnnotator project ensures that it not only advances HRI research but does so with a firm commitment to upholding high ethical standards. By addressing each ethical challenge with clear strategies and maintaining rigorous documentation and team training, the project sets a standard for responsibility and inclusivity in technological development.