

Project Name	...
Online team meeting	https://fau.zoom.us/j/68283073150
Production system (if any)	...
Test system (if any)	...
GitHub repository	https://github.com/amosproj/amos2023ws05-pipeline-config-chat-ai
GitHub feature board	https://github.com/orgs/amosproj/projects/29/views/2
GitHub impediments backlog	https://github.com/orgs/amosproj/projects/35
Team T-shirt (black)	https://www.shirtinator.de/loadBasket/rhFtkf_qdIL
Additional materials	...
Team email list	oss-amos-proj5@lists.fau.de
Team logo	https://www.shirtinator.de/loadBasket/rhFtkf_qdIL
Happiness index	https://happy-amos.appspot.com/Project?project=6222853397544960&course=6210557241720832

Last Name	First Name	GitHub User Name	Email Address
Katziuk	Avi	avikatziuk	avi.katziuk@fau.de
Betancourt Barrita	Cecilia	ceciliabetb	cecilia.betancourt@fau.de
El Brak	Sara	SaraElBrak	sara.el@fau.de
Hazzouri	Zain Alabden	bergzain	zain.hazzouri@gmail.com
Najar	Linda	lyndanajjar	linda.najar@fau.de
Smadi	Obeidah	obismadi99	obeidah0smadi@gmail.com
Gtari	Nahrain	Nahrain1	nahraingtari1@gmail.com

#	Meeting Day	Product Owner	Software Developer	Release Manager	Scrum Master	Comment
1	2023-11-01	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	zain hazzouri	Sara El Brak	
2	2023-11-08	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	Obi Smadi	Sara El Brak	
3	2023-11-15	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	Nahrain Gtari	Sara El Brak	
4	2023-11-22	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	Linda Najar	Sara El Brak	
5	2023-11-29	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	zain hazzouri	Sara El Brak	
6	2023-12-06	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	Obi Smadi	Sara El Brak	
7	2023-12-13	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	Nahrain Gtari	Sara El Brak	Mid-term due
8	2023-12-20	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	Linda Najar	Sara El Brak	
9	2023-12-27	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	zain hazzouri	Sara El Brak	
10	2023-01-11	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	Obi Smadi	Sara El Brak	
11	2023-01-18	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	Nahrain Gtari	Sara El Brak	
12	2023-01-25	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	Linda Najar	Sara El Brak	
13	2023-02-01	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	zain hazzouri	Sara El Brak	
14	2023-02-08	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	Obi Smadi	Sara El Brak	Demo day!
15	2023-02-15	Cecilia Betancourt Barrita, Avi Katziuk	Everyone else	Nahrain Gtari	Sara El Brak	Retrospective

Goals	Create a ChatBot per the company's requirements in a successful way
	Maintain an agile manner of work in a good atmosphere
Meeting norms	The cameras should be turned on for better human communication
	Meetings must not exceed 30 min over time, otherwise schedule another meeting.
	Delays should be informed and handled on agile way (write in the telegram group)
	weekly meeting
Working norms	Contributing and communicating openly
	Keep respect during possible conflicts (on technical issues z.B)
Coordination norms	Everyone sticks to assigned roles
Communication norms	communicate in advance and don't leave it for the last minute (don't be ashamed of asking simple questions)
Consideration norms	We communicate conflicts openly
	Team final decisions are based on voting
Cont. improvement norms	Improvement suggestions are welcomed
	Happiness index and standup emails to be reviewed in case of non-satisfaction
Rewards	Obi makes the team a cake - final reward
	Have coffee and bring games after the mid release
Sanctions	Team will decide on the memeber's sanction accordingly
Signatures	
Scrum Master	Sara El Brak
Product owner	Cecilia Betancourt Barrita
Product owner	Avi Katziuk
Software developer	zain hazzouri (TU Berlin)
Software developer	Obi Smadi (TU Berlin)
Software developer	Nahrain Gtari
Software developer	Linda Najar

Product Vision	Project Mission
<p>UPDATED 03.12.23 Introducing ChatRTDIP. By leveraging the power of large language models and generative AI, this product aims to offer an intuitive, user-friendly interface for establishing RTDIP configurations. The focus is on simplicity and efficiency, allowing users from various backgrounds, technical or otherwise, to interact with and create complex data extraction configurations with ease. The integration of the RTDIP and RTDIP SDK as a knowledge base, combined with a chat engine, will result in an accessible and adaptable product, ultimately reducing the learning curve, time, and effort required in analyzing real-time sensor data.</p>	<p>UPDATED 03.12.23 The mission of the ChatRTDIP project, in collaboration with Shell and under the AMOS framework, is to develop a cutting-edge tool that seamlessly blends chat AI technology with the robust data processing capabilities of RTDIP. The project is dedicated to creating a platform where users can effortlessly configure data pipelines using a conversational interface.</p>

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release							
Total				215	215	178	
Sprints							
1	Project Kickoff and Planning			5	215	5	215
2	Chatbot Software Architecture and Framework			33	210	27	210
3	Natural Language Processing Integration			8	177	8	183
4	User Interaction and Model Enhacement			10	169	10	175
5	User Interface and Conversation Flow			13	159	13	165
6	Refine Interactive Communication			28	146	25	152
7	Begin testing and simplify UI			34	118	34	127
8	Continue query testing and prepare for advanced queries input			20	84	19	93
9	Query, performance testing and improvements			8	64	0	74
10	Farther speed improvments and UI adjustments			12	56	15	74
11	Test of unsupported queries and edge cases			25	44	22	59
12	Quality of life features			0	19	0	37
13	Prepare for final release			19	19	0	37
...					118		127
Features							
1	Project Kickoff and Planning						
		Establish project infrastructure and team roles					
			Create bill of materials	3		3	
			Define regular meeting with IP	1		1	
			Build process review - Homework	1		1	
2	Chatbot Software Architecture and Framework						
		Define the software architecture and prepare the necessary data for training the chatbot					
			Connect the bot to the RAG Hub	2		2	
			Pipeline workshop with IP	1		1	
			Initialize a ReadMe File	2		2	
			Collect the data for the RAG content Hub	2		2	
			Initialize a RAG content Hub	8		5	

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Define a third meeting with IP	1		1	
			Create a pre-alpha version of UI	8		5	
			Create the first output	3		3	
			Environment set up (pipeline)	3		3	
			Create a Software Architecture Document	3		3	
3	Natural Language Processing Integration						
			Develop the core NLP model for chatbot understanding				
			Create and configure inference end point for the model	5		5	
			Source processing power	3		3	
4	User Interaction and Model Enhancement						
			Create wireframes and UI mockups.				
			Create a docker image to run the inference endpoint	5		5	
			Test Plan for Prompt Engineering	2		2	
			Initialize a Domain-Specific Dataset (DNS)	3		3	
5	User Interface and Conversation Flow						
			Design the chatbot's user interface and conversation flow.				
			Finding and Configuring an alternative UI	3		3	
			Inference Endpoint and Docker Container Creation	3		3	
			Text generation inference configuration	2		2	
			Create a new docker file	5		5	
6	Refine Interactive Communication						
			Detail on how to communicate with the bot				
			Prepare 7 testing queries for testing week 1	2		2	
			Add OpenAI API Key to the UI	3		3	
			Add response speed measurement to test how long does it take for the response	3		2	
			Implement Caching for Faster Responses	3		3	
			Create requirement.txt	2		2	
			Enable iterative communication with the bot	5		5	
			Set up a query catalog - single source of truth	5		3	
			Developing a Model-Agnostic Approach for Enhancing Speed Performance	5		5	
7	Begin testing and simplify UI						
			Boost reliability, UX through testing, UI simplification for usability.				
			Q[1,a] - Query testing, optimization and verification	3		3	
			Clear the left UI bar	2		2	
			Add a title and Github link on the left of the UI	3		3	
			Streaming response model support	3		3	

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Prepare testing queries for testing week 2,3,4	2		2	
			Q[1,d] - Query testing, optimization and verification	3		3	
			Q[1,f] - Query testing, optimization and verification	3		3	
			Q[1,g] - Query testing, optimization and verification	3		3	
			Q[1,c] - Query testing, optimization and verification	3		3	
			Q[1,b] - Query testing, optimization and verification	3		3	
			Update ReadMe File	2		2	
			Cleaning up the repo	2		2	
			Q[1, a - g] Query testing, optimization and verification	2		2	
8			Continue query testing and prepare for advanced queries input				
			Advance query testing, prepare for complex query inputs continuation.				
			Implement a script for queries construction	3		3	
			Script for Dockerizing & App Launching	3		3	
			Q[2, a - g] Query testing, optimization and verification	3		2	
			Experiment: Improve response times with a database	3		0	
			API error handling	2		3	
			Improve context awareness	3		5	
			Parsing for incrementally generated text	3		3	
9			Query, performance testing and improvements				
			Enhance query performance through testing and optimizations.				
			Q[3, a - g] Query testing, optimization and verification	2			
			Initialize and test Q3 queries manually with PyTest	3			
			Script for updating the RAG content store in case changes are made to the RTDIP repo	3			
10			Farther speed improvments and UI adjustments				
			Create a new set of improvements and new performance for the Interface with the user.				
			Fix bug in chat display	2		2	
			Implement RAGUpdate trigger and modification date retrieval	2		3	
			Documentation for Shell	3		5	
			Explore a solution for fixing Q2 and Q3 queries (only for website component queries)	5		5	
11			Test of unsupported queries and edge cases				
			Draft for demo day slide and demo day video				
			Display sessions history - backend	5		5	
			Update bill of materials	1		1	
			Testing output for consistency	3		3	
			Testing output for accuracy	3			

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Solution #2: Retrievers merger	8		8	
			Solution #1: Lean content hub	5		5	
12	Quality of life features						
13	Prepare for final release						
			Create the final Software Architecture				
			Testing output for accuracy (solution 2)	3			
			UI adjustments	3			
			Find a solution for surfaced consistency issues	8			
			Display sessions history - frontend	5			

Term	Definition

[illegible]

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release							
Total				215	215	178	
Sprints							
1	Project Kickoff and Planning			5	215	5	215
2	Chatbot Software Architecture and Framework			33	210	27	210
3	Natural Language Processing Integration			8	177	8	183
4	User Interaction and Model Enhancement			10	169	10	175
5	User Interface and Conversation Flow			13	159	13	165
6	Refine Interactive Communication			28	146	25	152
7	Begin testing and simplify UI			34	118	34	127
8	Continue query testing and prepare for advanced queries input			20	84	19	93
9	Query, performance testing and improvements			8	64	0	74
10	Farther speed improvements and UI adjustments			12	56	15	74
11	Test of unsupported queries and edge cases			25	44	22	59
12	Quality of life features			0	19	0	37
13	Prepare for final release			19	19	0	37
...					118		127
Features							
1	Project Kickoff and Planning						
			Establish project infrastructure and team roles				
			Create bill of materials	3		3	
			Define regular meeting with IP	1		1	
			Build process review - Homework	1		1	
2	Chatbot Software Architecture and Framework						
			Define the software architecture and prepare the necessary data for training the chatbot				
			Connect the bot to the RAG Hub	2		2	
			Pipeline workshop with IP	1		1	
			Initialize a ReadMe File	2		2	
			Collect the data for the RAG content Hub	2		2	
			Initialize a RAG content Hub	8		5	

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Define a third meeting with IP	1		1	
			Create a pre-alpha version of UI	8		5	
			Create the first output	3		3	
			Environment set up (pipeline)	3		3	
			Create a Software Architecture Document	3		3	
3			Natural Language Processing Integration				
			Develop the core NLP model for chatbot understanding				
			Create and configure inference end point for the model	5		5	
			Source processing power	3		3	
4			User Interaction and Model Enhancement				
			Create wireframes and UI mockups.				
			Create a docker image to run the inference endpoint	5		5	
			Test Plan for Prompt Engineering	2		2	
			Initialize a Domain-Specific Dataset (DNS)	3		3	
5			User Interface and Conversation Flow				
			Design the chatbot's user interface and conversation flow.				
			Finding and Configuring an alternative UI	3		3	
			Inference Endpoint and Docker Container Creation	3		3	
			Text generation inference configuration	2		2	
			Create a new docker file	5		5	
6			Refine Interactive Communication				
			Detail on how to communicate with the bot				
			Prepare 7 testing queries for testing week 1	2		2	
			Add OpenAI API Key to the UI	3		3	
			Add response speed measurement to test how long does it take for the response	3		2	
			Implement Caching for Faster Responses	3		3	
			Create requirement.txt	2		2	
			Enable iterative communication with the bot	5		5	
			Set up a query catalog - single source of truth	5		3	
			Developing a Model-Agnostic Approach for Enhancing Speed Performance	5		5	
7			Begin testing and simplify UI				
			Boost reliability, UX through testing, UI simplification for usability.				
			Q[1,a] - Query testing, optimization and verification	3		3	
			Clear the left UI bar	2		2	
			Add a title and Github link on the left of the UI	3		3	
			Streaming response model support	3		3	

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Prepare testing queries for testing week 2,3,4	2		2	
			Q[1,d] - Query testing, optimization and verification	3		3	
			Q[1,f] - Query testing, optimization and verification	3		3	
			Q[1,g] - Query testing, optimization and verification	3		3	
			Q[1,c] - Query testing, optimization and verification	3		3	
			Q[1,b] - Query testing, optimization and verification	3		3	
			Update ReadMe File	2		2	
			Cleaning up the repo	2		2	
			Q[1, a - g] Query testing, optimization and verification	2		2	
8			Continue query testing and prepare for advanced queries input				
			Advance query testing, prepare for complex query inputs continuation.				
			Implement a script for queries construction	3		3	
			Script for Dockerizing & App Launching	3		3	
			Q[2, a - g] Query testing, optimization and verification	3		2	
			Experiment: Improve response times with a database	3		0	
			API error handling	2		3	
			Improve context awareness	3		5	
			Parsing for incrementally generated text	3		3	
9			Query, performance testing and improvements				
			Enhance query performance through testing and optimizations.				
			Q[3, a - g] Query testing, optimization and verification	2			
			Initialize and test Q3 queries manually with PyTest	3			
			Script for updating the RAG content store in case changes are made to the RTDIP repo	3			
10			Farther speed improvments and UI adjustments				
			Create a new set of improvements and new performance for the Interface with the user.				
			Fix bug in chat display	2		2	
			Implement RAGUpdate trigger and modification date retrieval	2		3	
			Documentation for Shell	3		5	
			Explore a solution for fixing Q2 and Q3 queries (only for website component queries)	5		5	
11			Test of unsupported queries and edge cases				
			Draft for demo day slide and demo day video				
			Display sessions history - backend	5		5	
			Update bill of materials	1		1	
			Testing output for consistency	3		3	
			Testing output for accuracy	3			

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Solution #2: Retrievers merger	8		8	
			Solution #1: Lean content hub	5		5	
12	Quality of life features						
13	Prepare for final release						
			Create the final Software Architecture				
			Testing output for accuracy (solution 2)	3			
			UI adjustments	3			
			Find a solution for surfaced consistency issues	8			
			Display sessions history - frontend	5			

Sprint #	Sprint goal
1	-
2	-
3	-
4	Elevate the user experience by improving user interaction and enhancing the underlying model's performance.
5	Set up and clean the user interface (UI) and streamlining conversation flow within our product.
6	Create a more meaningful and compelling communication experience for our users by refining interactive communication.
7	Verify and test individual features for bot actions and natural language understanding.
8	Continue query testing and prepare for advanced queries input
9	Query, performance testing and improvements
10	Farther speed improvments and UI adjustments
11	Test of unsupported queries and edge cases
12	Quality of life features
13	Prepare for final release

[illegible]

#	Context	Name	Version	License	Comment
1	1st level dependency	Python	3.11	OSI-approved	
2	1st level dependency	PySpark		Apache License	
3	1st level dependency	RTDIP SDK	v0.5.0	Apache 2.0 License	
4	1st level dependency	Git	3.10.3	GNU	
5	1st level dependency	HTML	5		
6	1st level dependency	CSS			
7	1st level dependency	Langchain	>= 0.0.184		
8	1st level dependency	Streamlit	1.28.2		
9	1st level dependency	OpenAI			
10	2nd level dependency	Chromadb			

Last Name	First Name	Value					
Katziuk	Avi			0.00	OK		
Betancourt Barrita	Cecilia						
El Brak	Sara						
Hazzouri	Zain Alabden	0					
Najar	Linda	0		0	No size		
Smadi	Obeidah	0		1	Trivial size		
Gtari	Nahrain	0		2	Small size		
				3	Medium size		
				5	Large size		
				8	Very large size		
				13	Too large (size)		