

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

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 Najjar	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 Najjar	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 Najjar	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

<b>Goals</b>	Create a ChatBot per the company's requirements in a successful way
	Maintain an agile manner of work in a good atmosphere
<b>Meeting norms</b>	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
<b>Working norms</b>	Contributing and communicating openly
	Keep respect during possible conflicts (on technical issues z.B)
<b>Coordination norms</b>	Everyone sticks to assigned roles
<b>Communication norms</b>	communicate in advance and don't leave it for the last minute ( don't be ashamed of asking simple questions )
<b>Consideration norms</b>	We communicate conflicts openly
	Team final decisions are based on voting
<b>Cont. improvement norms</b>	Improvement suggestions are welcomed
	Happiness index and standup emails to be reviewed in case of non-satisfaction
<b>Rewards</b>	Obi makes the team a cake - final reward
	Have coffee and bring games after the mid release
<b>Sanctions</b>	Team will decide on the memeber's sanction accordingly
<b>Signatures</b>	
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><b>UPDATED 03.12.23</b> 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><b>UPDATED 03.12.23</b> 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				177	177	156	
Sprints							
1	Project Kickoff and Planning			5	177	5	177
2	Chatbot Software Architecture and Framework			33	172	27	172
3	Natural Language Processing Integration			8	139	8	145
4	User Interaction and Model Enhancement			10	131	10	137
5	User Interface and Conversation Flow			13	121	13	127
6	Refine Interactive Communication			28	108	25	114
7	Begin testing and simplify UI			34	80	34	89
8	Continue query testing and prepare for advanced queries input			20	46	19	55
9	Query, performance testing and improvements			8	26	0	36
10	Farther speed improvements and UI adjustments			12	18	15	36
11	Test of unsupported queries and edge cases			6	6	0	21
12	Quality of life features			0	0	0	21
13	Prepare for final release			0	0	0	21
...					80		89
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	<b>Natural Language Processing Integration</b>						
			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	<b>User Interaction and Model Enhancement</b>						
			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	<b>User Interface and Conversation Flow</b>						
			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	<b>Refine Interactive Communication</b>						
			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	<b>Begin testing and simplify UI</b>						
			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			<b>Continue query testing and prepare for advanced queries input</b>				
			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			<b>Query, performance testing and improvements</b>				
			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			<b>Farther speed improvments and UI adjustments</b>				
			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			<b>Test of unsupported queries and edge cases</b>				
			Draft for demo day slide and demo day video				
			Display sessions history - backend	5			
			Update bill of materials	1			
			Testing output for consistency				
			Testing output for accuracy				



Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Solution #2: Retrievers merger				
			Solution #1: Lean content hub				
12	Quality of life features						
13	Prepare for final release						

Term	Definition

[illegible]

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

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release							
Total				177	177	151	
Sprints							
1	Project Kickoff and Planning			5	177	5	177
2	Chatbot Software Architecture and Framework			33	172	27	172
3	Natural Language Processing Integration			8	139	8	145
4	User Interaction and Model Enhancement			10	131	10	137
5	User Interface and Conversation Flow			13	121	13	127
6	Refine Interactive Communication			28	108	25	114
7	Begin testing and simplify UI			34	80	34	89
8	Continue query testing and prepare for advanced queries input			20	46	19	55
9	Query, performance testing and improvements			14	26	8	36
10	Farther speed improvements and UI adjustments			12	12	2	28
11	Test of unsupported queries and edge cases			0	0	0	26
12	Quality of life features			0	0	0	26
13	Prepare for final release			0	0	0	26
...					80		89
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			<b>Natural Language Processing Integration</b>				
			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			<b>User Interaction and Model Enhancement</b>				
			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			<b>User Interface and Conversation Flow</b>				
			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			<b>Refine Interactive Communication</b>				
			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			<b>Begin testing and simplify UI</b>				
			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			<b>Continue query testing and prepare for advanced queries input</b>				
			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			<b>Query, performance testing and improvements</b>				
			Enhance query performance through testing and optimizations.				
			Permanant store API key	3			
			Testing for speed performance tweak	3			
			Q[3, a - g] Query testing, optimization and verification	2		2	
			Initialize and test Q3 queries manually with PyTest	3		3	
			Script for updating the RAG content store in case changes are made to the RTDIP repo	3		3	
10			<b>Farther speed improvments and UI adjustments</b>				
			Create a new set of improvements and new performance for the Interface with the user.				
			Fix bug chat in display	2		2	
			Update RAG per button	2			
			Documentation for Shell	3			
			Explore a solution for fixing Q2 and Q3 queries (only for website component queries)	5			
11			<b>Test of unsupported queries and edge cases</b>				

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
12	Quality of life features						
13	Prepare for final release						



Type	Link / reference

#	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	2st level dependency	Streamlit	1.28.2		
9	2st level dependency	OpenAI			

Last Name	First Name	Value					
Katziuk	Avi			3.00	OK		
Betancourt Barrita	Cecilia						
El Brak	Sara						
Hazzouri	Zain Alabden	3					
Najar	Linda	3		0	No size		
Smadi	Obeidah	3		1	Trivial size		
#REF!	#REF!			2	Small size		
#REF!	#REF!			3	Medium size		
Gtari	Nahrain	3		5	Large size		
				8	Very large size		
				13	Too large (size)		