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_qdlL
Additional materials	...
Team email list	oss-amos-proj5@lists.fau.de
Team logo	https://www.shirtinator.de/loadBasket/rhFtkf_qdlL
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
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.	**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.

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release							
Total				154	154	116	
Sprints							
1	Project Kickoff and Planning			6	154	6	154
2	Chatbot Software Architecture and Framework			29	148	23	148
3	Natural Language Processing Integration			21	119	21	125
4	User Interaction and Model Enhacement			16	98	16	104
5	User Interface and Conversation Flow			16	82	16	88
6	Refine Interactive Communication			15	66	15	72
7	Begin testing and simplify UI			34	51	19	57
8	Continue query testing and prepare for advanced queries input			17	17	0	38
9	Query, performance testing and improvements			0	0	0	38
10	Farther speed improvments and UI adjustments			0	0	0	38
11	Test of unsupported queries and edge cases			0	0	0	38
12	Quality of life features			0	0	0	38
13	Prepare for final release			0	0	0	38
...					51		57
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	
			Pipeline workshop 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	
			Collect the data for the RAG content Hub	2		2	
			Initialize a RAG content Hub	8		5	
			Create a pre-alpha version of UI	8		5	

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Create the first output	3		3	
			Environment set up (pipleline)	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	
			Developing a Model-Agnostic approach for Enhacing Speed Performance	5		5	
			Finding and Configuring an alternative UI	3		3	
			Initialize the full version of the bot	5		5	
4	**User Interaction and Model Enhacement**						
		Create wireframes and UI mockups.					
			Create a docker image to run the inference endpoint	5		5	
			Write a document with steps we need to take to get the bot working	3		3	
			Test Plan for Prompt Engineering	2		2	
			Initialize a Domain-Specific Dataset (DNS)	3		3	
			Initialize a README File	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	
			Connect the endopoint to the original 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**						
			Create requirement.txt	2		2	
			Display code on the right	3		3	
			Enable iterative communication with the bot	5		5	
			Set up a query catalog - single source of truth	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			
			Q[1,f] - Query testing, optimization and verification	3			
			Q[1,g] - Query testing, optimization and verification	3			
			Q[1,c] - Query testing, optimization and verification	3			
			Q[1,b] - Query testing, optimization and verification	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.				
			Parsing for incrementally generated text	3			
			Improve context awareness	3			
			API error handling	2			
			Experiment: Improve response times with a database	3			
			Q[2, a - g] Query testing, optimization and verification	3			
			Script for Dockerizing & App Launching	3			
9		**Query, performance testing and improvements**					
			Enhance query performance through testing and optimizations.				
			Q[3, a - g] Query testing, optimization and verification				
			Plan how to test context awareness feature				
			Implement speed improvement solutions				
			Implement a "new conversation" button				
			Placeholder for fixing failed tests				
10		**Farther speed improvments and UI adjustments**					
			Create a new set of improvements and new performance for the Interface with the user.				
			Q[4, a - g] Query testing, optimization and verification				
			Test speed improvement solution				
			Adjust UI margins				
			Test context awareness				
			Placeholder for fixing failed tests				
11		**Test of unsupported queries and edge cases**					
			Q[4, a - g] Query testing, optimization and verification				
			Enable reading API from config file				
			Enable message editing				

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Placeholder for fixing failed tests				
12		**Quality of life features**					
			Enable message editing				
			Enable flipping between edited messages				
			Test response for non-supported queries				
			Placeholder for fixing failed tests				
13		**Prepare for final release**					
			Finalize the RAG				
			Close any unfixed issues from testing				
			Update all documentation files				

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining

Term	Definition

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

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
1	Code review has been completed and code has been merged.	Project builds, deploys, and tests successfully	User interaction tests pass on all major browsers
2	User interaction tests pass on all major browsers.		Design documentation has been updated
3			
4			
5			
6			
7			

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release							
Total				154	154	116	
Sprints							
1	Project Kickoff and Planning			6	154	6	154
2	Chatbot Software Architecture and Framework			29	148	23	148
3	Natural Language Processing Integration			21	119	21	125
4	User Interaction and Model Enhacement			16	98	16	104
5	User Interface and Conversation Flow			16	82	16	88
6	Refine Interactive Communication			15	66	15	72
7	Begin testing and simplify UI			34	51	19	57
8	Continue query testing and prepare for advanced queries input			17	17	0	38
9	Query, performance testing and improvements			0	0	0	38
10	Farther speed improvments and UI adjustments			0	0	0	38
11	Test of unsupported queries and edge cases			0	0	0	38
12	Quality of life features			0	0	0	38
13	Prepare for final release			0	0	0	38
...					51		57
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	
			Pipeline workshop 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	
			Collect the data for the RAG content Hub	2		2	
			Initialize a RAG content Hub	8		5	
			Create a pre-alpha version of UI	8		5	

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Create the first output	3		3	
			Environment set up (pipleline)	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	
			Developing a Model-Agnostic approach for Enhacing Speed Performance	5		5	
			Finding and Configuring an alternative UI	3		3	
			Initialize the full version of the bot	5		5	
4		**User Interaction and Model Enhacement**					
			Create wireframes and UI mockups.				
			Create a docker image to run the inference endpoint	5		5	
			Write a document with steps we need to take to get the bot working	3		3	
			Test Plan for Prompt Engineering	2		2	
			Initialize a Domain-Specific Dataset (DNS)	3		3	
			Initialize a README File	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	
			Connect the endopoint to the original 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**					
			Create requirement.txt	2		2	
			Display code on the right	3		3	
			Enable iterative communication with the bot	5		5	
			Set up a query catalog - single source of truth	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			
			Q[1,f] - Query testing, optimization and verification	3			
			Q[1,g] - Query testing, optimization and verification	3			
			Q[1,c] - Query testing, optimization and verification	3			
			Q[1,b] - Query testing, optimization and verification	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.				
			Parsing for incrementally generated text	3			
			Improve context awareness	3			
			API error handling	2			
			Experiment: Improve response times with a database	3			
			Q[2, a - g] Query testing, optimization and verification	3			
			Script for Dockerizing & App Launching	3			
9		**Query, performance testing and improvements**					
			Enhance query performance through testing and optimizations.				
			Q[3, a - g] Query testing, optimization and verification				
			Plan how to test context awareness feature				
			Implement speed improvement solutions				
			Implement a "new conversation" button				
			Placeholder for fixing failed tests				
10		**Farther speed improvments and UI adjustments**					
			Create a new set of improvements and new performance for the Interface with the user.				
			Q[4, a - g] Query testing, optimization and verification				
			Test speed improvement solution				
			Adjust UI margins				
			Test context awareness				
			Placeholder for fixing failed tests				
11		**Test of unsupported queries and edge cases**					
			Q[4, a - g] Query testing, optimization and verification				
			Enable reading API from config file				
			Enable message editing				

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
			Placeholder for fixing failed tests				
12		**Quality of life features**					
			Enable message editing				
			Enable flipping between edited messages				
			Test response for non-supported queries				
			Placeholder for fixing failed tests				
13		**Prepare for final release**					
			Finalize the RAG				
			Close any unfixed issues from testing				
			Update all documentation files				

Sprint	Topic	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining

Type	Link / reference

Last Name	First Name	Value					
Katziuk	Avi			**8.00**	**OK**		
Betancourt Barrita	Cecilia						
El Brak	Sara						
Hazzouri	Zain Alabden	8					
Najar	Linda	8		0	No size		
Smadi	Obeidah	8		1	Trivial size		
#REF!	#REF!	8		2	Small size		
#REF!	#REF!	8		3	Medium size		
Gtari	Nahrain	8		5	Large size		
				8	Very large size		
				13	Too large (size)		

#	Context	Name	Version	License	Comment
1	1st level dependecy	Python	3.11	OSI-approved	
2	1st level dependecy	PySpark		Apache License	
3	1st level dependecy	RTDIP SDK	v0.5.0	Apache 2.0 License	
4	1st level dependecy	Git	3.10.3	GNU	
5	1st level dependecy	HTML	5		
6	1st level dependecy	CSS			