Project Name	AI Agent Definition and Generation Framework
Online team meeting	https://fau.zoom-x.de/j/66967665829
Production system (if any)	...
Test system (if any)	Github Actions: https://github.com/amosproj/amos2024ss06-health-ai-framework/tree/7f68876bb8cb0176c2791674ac83e60a27d3d715/.github
GitHub repository	https://github.com/amosproj/amos2024ss06-health-ai-framework
GitHub feature board	https://github.com/orgs/amosproj/projects/53
GitHub impediments backlog	https://github.com/orgs/amosproj/projects/64
Team T-shirt (white)	https://www.shirtinator.de/s/nDefN-9-QpugOXEh5oJIOA
Team T-shirt (black)	https://www.shirtinator.de/s/Xeu6K1B5RrSUf5lw05rzSg
Additional materials	...
Team maling list	oss-amos-proj6@lists.fau.de

Last Name	First Name	GitHub User Name	Email Address	Comment
Enescu	David	EnescuDavid	david.enescu@fau.de	
Karategos	Georgios	tubamos	amosproject@proton.me	Email used ONLY for GitHub and the Happy-AMOS tool.
Negasa	Gemechis	Ghemechis	gemechismelkamu@gmail.com	
Potthoff	Jan	potthoffjan	potthoff.jan@googlemail.com	
Sandt	Eloi	eloinoel	eloi.sandt@campus.tu-berlin.de	
Varga	Lukáš	lukas-varga	lukas.varga128@gmail.com	
Zimmermann	Simon	Tims777	tim.simon.zimmermann@fau.de	
Gupta	Manik	manikg08	manik.gupta@fau.de	
Vadaliya	Preet	preetvadaliya	preet.vadaliya@fau.de	

#	Meeting Day	Product Owners	Software Developer	Release Manager	Scrum Master	Comment
	2024-04-17	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	N/A	Simon	
1	2024-04-24	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	David	Simon	irr
	2024-05-01	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Manik	Simon	irr
2	2024-05-08	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Preet	Simon	-
3	2024-05-15	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Jan	Simon	-
4	2024-05-22	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Eloi	Simon	irr
5	2024-05-29	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Lukáš	Simon	irr
6	2024-06-05	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	David	Simon	Mid-term due
7	2024-06-12	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Manik	Simon	-
8	2024-06-19	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Preet	Simon	-
9	2024-06-26	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Jan	Simon	-
10	2024-07-03	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Eloi	Simon	irr
11	2024-07-10	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Lukáš	Simon	irr
12	2024-07-17	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Jan	Simon	Demo day!
13	2024-07-24	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Preet	Simon	Retrospective
Product owners, software developers, and Scurm Master are set and ideally don't change over time; the critical part is the Release Manager role you need to define here						

Goals	- learn and share learning on tech and collabration tasks
	- keep Andi Zink happy
	- finish tasks for each sprint on time (or communicate early why it isnt possible or if you need help)
Meeting norms	- be on time
	- be informed on topics relevant for the meeting (eg. lecture)
	- if you are late or cant make it, give a heads up (eg. on slack)
	- active participation
	- let's not digress too much while key points haven't been gone through yet
Working norms	- ask for help
	- decisions made in mutual agreement
	- deliver constructive criticism with the goal of helping others and yourself grow
	- for code: settle on conventions
	- good comments/documentation of work
	- be considerate of varying sets of experiences and skills
Coordination norms	- the Scrum Master keeps the meeting on track and PO leads the meeting
	- the POs (at least, devs are always welcome to join) meet with the partner at regular intervals and communicate / clarify feature requests to the rest of the team.
Communication norms	- be welcoming and nice
	- dont interupt others
	- communicate illnesses and expected work downtime and what it means for others
	- communication through slack student channel
Consideration norms	- if I disagree, I should think of an alternative/provide constructive reasoning
	- both sides should be heard and find a consense together
	- prefer to speak in public over speaking privately
Cont. improvement norms	- performance evaluated based on weekly sprint target compared with git pull requests
	- automated tests?
Rewards	- after successful release, we can go to mensa (separetly in Erlangen / Berlin)
	- consider meeting in person after the project is completed (in Berlin or Erlangen)
Sanctions	- Start with a warning. Persistent issues can lead to report.

Signatures	
Scrum Master	Simon Zimmermann
Product owner	Georgios Karategos
Product owner	Gemechis Negasa
Software developer	Manik Gupta
Software developer	David Enescu
Software developer	Eloi Sandt
Software developer	Lukáš Varga
Software developer	Preet Vadaliya
Software developer	Jan Potthoff

Product Vision	Project Mission
Ailixir” empowers users to create and incrementally refine custom AI agents that are specialized in specific domains. It assists users in gaining useful, reliable and timely answers to domain specific questions they are interested in, eventually becoming a dependable companion in the journey to navigate effectively within the field of their choice.	The mission of Ailixir is to produce an MVP for a prototyping tool that allows users, who are entrepreneurs / developers to create, refine and compare the results of custom AI agents. Ailixir can be thought of a combination of three pieces that come together to achive its goals: - The first one aims to create an automated, modular pipeline of acquiring, storing and generating current contextual information from handpicked knowledge sources. - The second one uses the acquired data and by utilizing a data pipeline which is modular enough to change various parameters, it produces answers independently of the underlying components. This allows the user to tweak parameters or replace components with the aim of finding an optimum combination that produces scientifically accurate and useful results. Important at this step is the ability to trace the sources that were used to generate the results. - The final piece of the project is the creation of a user-facing app that allows users to interact with the data pipeline via modalities such as text and voice.

Sprint #	Sprint goal
1	None
2	None
3	None
4	Progress data acquisition and storage functionality, start with Langchain and app framework
5	Finalise the data acquisition functionality, progress with Langchain and setup the basics for the app
6	Finalise data acquisition, finalise basic LangChain pipeline and continue setting up the app
7	Build frontend components
8	
9	
10	
11	
12	
13	

Term	Definition
config	a way to configure a handpicked list of sources to be used as targets for data scrapping or access to 3rd party APIs
orchestrator	A piece of logic that coordinates / orchestrates the data acquisition process from the scrapping targets / API providers based on the configurator

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			190	190		
Sprints						
1	Basic formal documentation requirements		0	190	0	190
2	Project set-up and start with data acquisition		29	190	29	190
3	CI/CD pipeline set-up, progress with data acquisition and local storage		35	161	30	161
4	Progress data acquisition and storage functionality, start with Langchain and app framework		72	126	62	131
5	Finalize data acquisition functionality, progress langchain and set up the basics for app		54	54	0	69
6	Finalise data acquisition, finalise basic LangChain pipeline and continue setting up the app		0	0	0	69
Features						
1	Basic formal documentation requirements					
		No items / commits				
2	Project set-up and start with data acquisition					
		Extract transcripts from podcasts	8		13	
		Extract transcript from a single YouTube video locally	8		3	
		Extend transcript extraction from YouTube	5		5	
		Create concept for the sources input functionality (config)	5		5	
		Setup Development Environment	3		3	
3	CI/CD pipeline set-up, progress with data acquisition and local storage					
		Implement the "orchestrator" functionality	8		8	
		Create storage solution for extracted data in the server of the FAU HPC server	8		3	
		Create CI/CD pipeline for Github	3		3	
		Extract data from academic repository PubMed Central (PMC) of the NIH	8		8	
			8		8	
4	Progress data acquisition and storage functionality, start with Langchain and app framework					
		Enhance Allrecipes scraper to include the latest orchestrator methods	3		3	
		Migration of the YouTube data extraction pipeline to GC	8		3	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
		LangChain - Query Contruction	8		5	
		LangChain - get on the same page	5		5	
		Extract recipe data from recipe website Allrecipes	8		8	
		Extract text content from academic repository: Arxiv	8		5	
		Create the DB for extracted data in Google Cloud	8		5	
		Enhance Arxiv scraper to include the new orchestrator methods	3		5	
		Enhance YouTube scraper to include the new orhestrator methods	3		5	
		Familiarise with the RAG technique and the LangChain framework	5		5	
		Extract blog posts from the website Nutritionfacts	8		8	
		Create system architecture for the app	5		5	
5	Finalize data acquisition functionality, progress langchain and set up the basics for app					
		Extend the "save" functionality of the base-scraper	5		0	
		Error handling in the orchestrator	5		0	
		Finalize LangChain query construction	5		0	
		Integrate the nutritionfacts blog posts scraper into the new orchestrator structure	3		0	
		Integrate the podcast scraper into the latest version of the orchestrator	3		0	
		LangChain - Basic retrival and generation	8		0	
		Create frontend interaction wireframes for the user app	8		0	
		Setup backend	8		0	
		Create automated PR	1		0	
		Integrate the NIH-PubMed scraper into the new orchestrator structure	3		0	
		Create build process video	5		0	
6	Finalise data acquisition, finalise basic LangChain pipeline and continue setting up the app					
		Setup react-native dev-environment	0		0	
		Setup a Firestore instance (pair)	0		0	
		Create tests for the orchestrator + config functionality	0		0	
		Setup and write project documentation in Github wiki	0		0	
		Write unit tests for orchestrator/config functionality	0		0	
		Implement the appropriate JSON chunking technique	0		0	
		Implement the LangChain pipeline in Typescript	0		0	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			0	0		
Sprints						
7	Build frontend components		0	0	0	0
8			0	0	0	0
9			0	0	0	0
10				0		0
11						
12				0		0
Features						
7	Build frontend components					
8						

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
9						
10						
11						

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
12						

Type	Link / reference

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
1	The feature has been fully implemented and integrated into the app.	All features planned for the sprint are integrated and functioning as expected.	The app functions correctly and responds appropriately to user inputs.
2	The feature has been manually tested and works as expected without critical bugs.	A manual check confirms that all integrated features work together without critical issues.	Design documentation is updated to reflect the final architecture and system design.
3	The feature code is documented with clear explanations of its functionality and usage.	The app builds successfully and can be deployed.	User documentation is updated to provide guides and instructions on using the app.
4	The feature code has been reviewed and approved by at least one team member.	Database update scripts are successful.	The project release is approved by the product owners and industry partner.
5	The feature branches have been merged into the main branch and closed.	Sprint release notes are written, summarising the new features, changes, and improvements.	
6		The changelog is updated to reflect all modifications made during the sprint.	

#	Context	Name	Version	License	Comment
1	development tooling	ruff	0.4.2	MIT	linter and code formatter
2	development tooling	pre-commit	3.7.0	MIT	pre-commit hooks
3	development tooling	python-dotenv	1.0.1	BSD License (BSD-3-Clause)	reads key-value pairs from a .env file
4	data pipeline development	python-youtube	0.9.4	MIT	youtube data scrapping, wrapper around for YouTube Data API V3.
5	data pipeline development	bs4	0.0.2	MIT License	Website scraping. Parses html and provides functionality to process the html
6	data pipeline development	pydub	0.25.1	MIT License (MIT)	To work on audiofiles
7	data pipeline development	ffmpeg	1.4	LGPL 2.1+/ GPL 2+	Needed to convert mp3 to wav
8	data pipeline development	wave	0.0.2	GNU Library or Lesser General Public License (LGPL)	work with wav files
9	data pipeline development	vosk	0.3.45	Apache Software License	Provides models for speech recognition and transcription
10	data pipeline development	paperscraper	0.2.11	MIT	Scraping text data from pdfs
11	data pipeline development	biopython	1.83	Freely Distributable	Pubmed API access, search queries, metadata scraping
12	data pipeline development	pypdf	4.2.0	BSD	Downloading pdfs from doi links (pubmed)
13	data pipeline development	requests	2.32.2	Apache 2.0	Requests allows you to send HTTP requests for download vosk transcription model
14	data pipeline development	arxiv	2.1.0	MIT	Webscraping tool API for Arxiv
15	deployment tool	apptainer	1.3.1	BSD	container tool for HPC (potentially) or Google Cloud
16	retrieval augmented generation	openai	1.30.1	Apache 2.0	Access to the OpenAI REST API
17	retrieval augmented generation	langchain	0.2.0	MIT	Framework for developing LLM
18	retrieval augmented generation	langchain-community	0.2.0	MIT	Framework for developing LLM
19	retrieval augmented generation	langchain-google-genai	1.0.4	MIT	Google integrations with LangChain
20	retrieval augmented generation	chromadb	0.5.0	Apache 2.0	AI-native open-source embedding database
21	retrieval augmented generation	vertexai	1.49.0	Apache 2.0	Platform for data science and machine learning
22	retrieval augmented generation	tiktoken	0.7.0	MIT	Tokeniser for OpenAI's models
23	retrieval augmented generation	sentence-transformers	2.7.0	Apache 2.0	Multilingual Sentence & Image Embeddings with BERT