Project Name	Ailixir - 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	Preet	Simon	Mid-term due
7	2024-06-12	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Jan	Simon	-
8	2024-06-19	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	David	Simon	-
9	2024-06-26	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Eloi	Simon	-
10	2024-07-03	George, Gemechis	David, Jan, Eloi, Lukáš, Manik, Preet	Manik	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

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 data acquisition functionality, progress langchain and set up the basics for app
6	Finalise data acquisition, finalise basic LangChain pipeline and continue setting up the app
7	Develop basic app components
8	Develop app components
9	Component integration and parallel answers
10	Voice Integration, modular initialisation prompt
11	Modular initialisation prompt, multiple LLMs, memory
12	Stability, handoff
13	AMOS Closing

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
Embeddings	numerical representations of data (such as words) that capture their semantic meanings and relationships, enabling their processing and analysis by machine learning algorithms
Text Chunking	the process of dividing a body of text into smaller pieces, like sentences or paragraphs, to facilitate the processing by NLP models
VectorStore	a specialized database that stores high-dimensional vectors, enabling efficient similarity searches and retrievals based on the semantic content of the data
LLM	an AI model that has been trained on large text datasets and can understand, generate, and predict text based on input prompts.
RAG	(Retrieval-Augmented Generation): a technique that combines retrieval of relevant documents from a database with generative capabilities to improve the accuracy and relevance of generated text
LangChain	a framework for developing applications powered by language models, enabling the integration of these models with various data sources and computational workflows.
Hugging Face	a platform that hosts tools, pre-trained models and datasets useful for building AI enabled applications
React Native	a framework that facilitates the creation of cross-platform mobile / web applications
Expo	a framework / platform for React Native that provides tools and services to make the development, building, and deployment of mobile applications easier.
Formik	a framework that facilitated the integration of forms with comprehensive frontend validation in an app.
Agent	a software designed to interact with users, understand queries, and provide responses using natural language processing and machine learning techniques
Conversation / Chat	a collection of user questions and agent answers within a context that is usually centered around a main thematic topic.
Jotai	State management for React native.
Prompt	a text input or query given to an AI model with the goal of generating a specific response or output based on its training data.
Prompt Engineering	the process of designing and refining prompts to elicit the most accurate, relevant, or desired responses from AI models.
React Native Component	a reusable, self-contained piece of code that defines the appearance and behavior of a part of the user interface in a React Native application
Firebase	an app development platform by Google that provides various tools and services, such as real-time databases, authentication, and cloud storage
Firestore	a cloud-based NoSQL database offered by Google Firebase for storing and syncing data across mobile and web applications
App navigation drawer	a UI component that slides in from the side of the screen, providing access to different sections or features of an application
Initialisation Prompt	A prompt controled by the entrepreneur-user which is tasked with asking the common-user basic questions about medical conditions, nutritional, fitness particularities and goals. This is done during an initialisation phase in the app right after the new user has created their account.
Entrepreneur User	Writes and administers the initialisation prompt for each instance of the app by changing the content of a google docs document.
Common User	In the current instance of the app: the user that utilises the app to answer their queries about lifestyle advice for people with persistent health conditions.

					Est. Size	Est. Remaining	Real Size	Real Remaining
				Mid Project Release	232	232	211	211
				Sprints 01-06	232	232	211	211
Sprint	**Start Date**	**End Date**	**Sprint Goal**	**Feature Name**		**Est. BD**		**Real BD**
1	2024-04-17	2024-04-24	AMOS Kickoff		0	232	0	211
2	2024-04-24	2024-05-08	Project set-up and start with data acquisition		32	232	32	211
3	2024-05-08	2024-05-15	CI/CD pipeline set-up, progress with data acquisition and local storage		27	200	22	179
4	2024-05-15	2024-05-22	Progress data acquisition and storage functionality, start with Langchain and app framework		72	173	62	157
5	2024-05-22	2024-05-29	Finalise data acquisition functionality, progress langchain and set up the basics for app		40	101	35	95
6	2024-05-29	2024-06-05	Finalise data acquisition, finalise basic LangChain pipeline and continue setting up the app		61	61	60	60
Sprint	**Start Date**	**End Date**	**Sprint Goal**	**Feature Name**	**Est.Size**		**Real Size**	
1	2024-04-17	2024-04-24	AMOS Kickoff					
				No items / commits	0		0	
					0		0	
2	2024-04-24	2024-05-08	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	
				Project Setup	3		3	
3	2024-05-08	2024-05-15	CI/CD pipeline set-up, progress with data acquisition and local storage					
				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	
				Implement the "orchestrator" functionality	8		8	
4	2024-05-15	2024-05-22	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	
				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	2024-05-22	2024-05-29	Finalise data acquisition functionality, progress langchain and set up the basics for app					
				LangChain - Basic retrival and generation	8		8	
				Create frontend interaction wireframes for the user app	8		8	
				Finalize LangChain query construction	5		3	
				Create build process video	5		5	

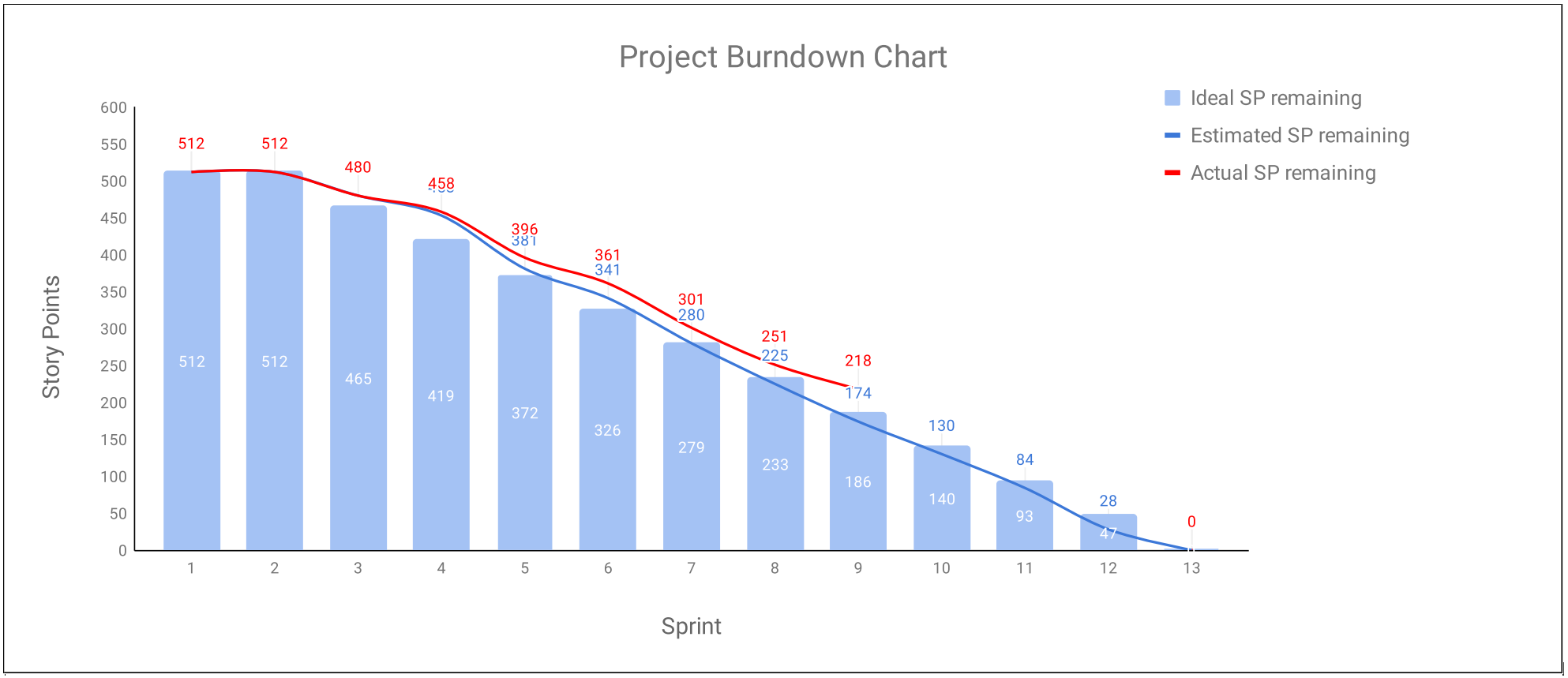
					Est. Size	Est. Remaining	Real Size	Real Remaining
				Mid Project Release	232	232	211	211
				Sprints 01-06	232	232	211	211
Sprint	**Start Date**	**End Date**	**Sprint Goal**	**Feature Name**		**Est. BD**		**Real BD**
1	2024-04-17	2024-04-24	AMOS Kickoff		0	232	0	211
2	2024-04-24	2024-05-08	Project set-up and start with data acquisition		32	232	32	211
3	2024-05-08	2024-05-15	CI/CD pipeline set-up, progress with data acquisition and local storage		27	200	22	179
4	2024-05-15	2024-05-22	Progress data acquisition and storage functionality, start with Langchain and app framework		72	173	62	157
5	2024-05-22	2024-05-29	Finalise data acquisition functionality, progress langchain and set up the basics for app		40	101	35	95
6	2024-05-29	2024-06-05	Finalise data acquisition, finalise basic LangChain pipeline and continue setting up the app		61	61	60	60
Sprint	**Start Date**	**End Date**	**Sprint Goal**	**Feature Name**	**Est.Size**		**Real Size**	
				Error handling in the orchestrator	5		3	
				Extend the "save" functionality of the base-scraper	5		2	
				Integrate the podcast scraper into the latest version of the orchestrator	3		5	
				Create automated PR	1		1	
6	2024-05-29	2024-06-05	Finalise data acquisition, finalise basic LangChain pipeline and continue setting up the app					
				Create short video as status update for IP and the Mid-Project product demo	3		3	
				Implement the LangChain pipeline in Typescript	8		8	
				Complete the Wireframe task to include the document download functionality	3		3	
				Setup a Firestore within Firebase	8		5	
				Write unit tests for orchestrator/config functionality	5		5	
				Integrate the nutritionfacts blog posts scraper into the new orchestrator structure	3		5	
				Setup react-native dev-environment	8		8	
				Implement the appropriate JSON chunking technique	5		5	
				Integrate the NIH-PubMed scraper into the new orchestrator structure	5		5	
				Patch for the PubMed Scraper pdf downloading problems	3		3	
				Catch errors from missing api-keys in .env file	1		1	
				Orchestrator: automatically create data/ folder and config.json	1		1	
				Setup and write project documentation in GitHub Wiki	8		8	

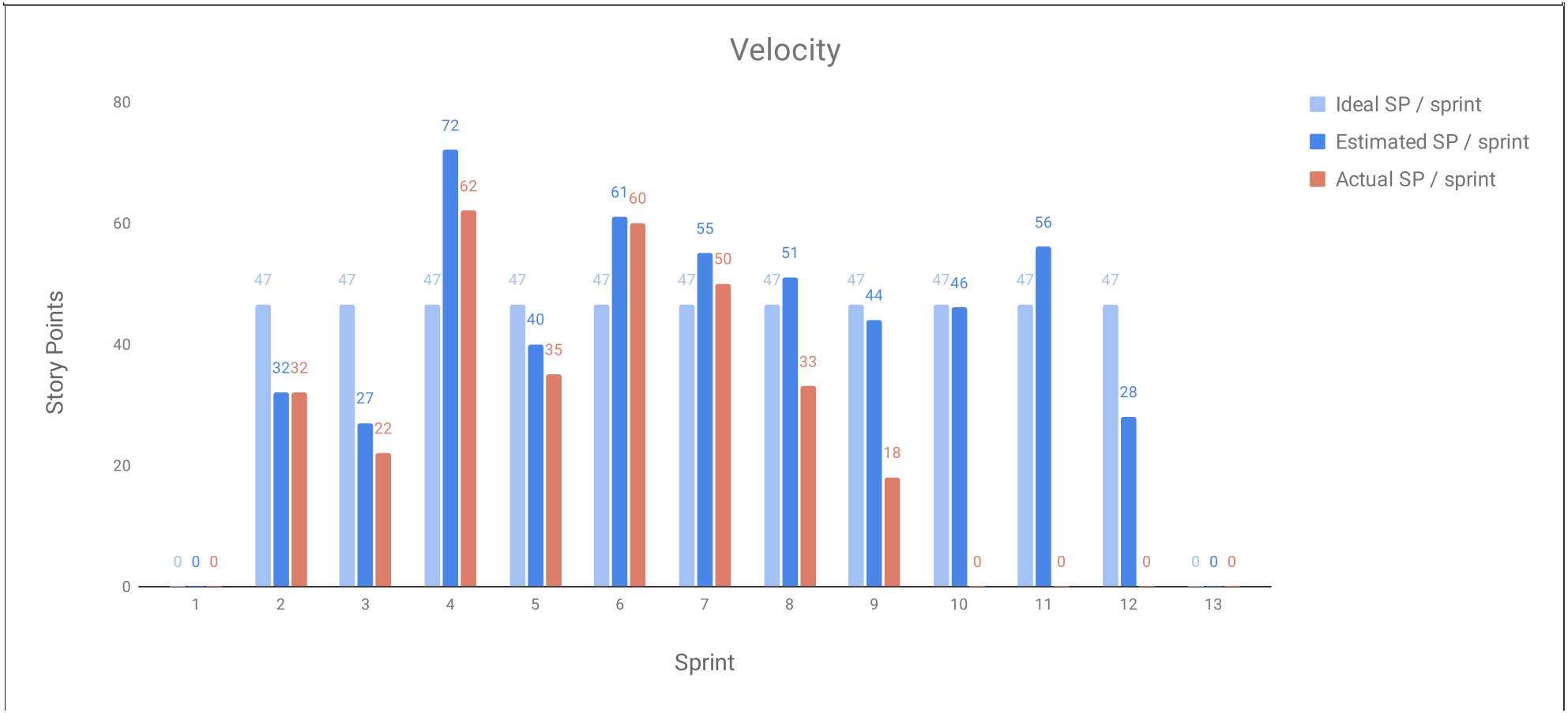
					Est. Size	Est. Remaining	Real Size	Real Remaining
				Complete Project	512	232	312	312
				Final Project Release	280	280	101	101
				Sprints 07-12	280	280	101	101
Sprint	**Start Date**	**End Date**	**Sprint Goal**	**Feature Name**		**Est. BD**		**Real BD**
7	2024-06-05	2024-06-12	Develop basic app components		55	280	50	101
8	2024-06-12	2024-06-19	Develop app components		51	225	33	51
9	2024-06-19	2024-06-26	Component integration and parallel answers		44	174	18	18
10	2024-06-26	2024-07-03	Voice Integration, modular initialisation prompt		46	130	0	0
11	2024-07-03	2024-07-10	Modular initialisation prompt, multiple LLMs, memory		56	84	0	0
12	2024-07-10	2024-07-17	Stability, handoff		28	28	0	0
13	2024-07-17	2024-07-24	AMOS Closing		0	0	0	0
Sprint	**Start Date**	**End Date**	**Sprint Goal**	**Feature Name**	**Est.Size**		**Real Size**	
7	2024-06-05	2024-06-12	Develop basic app components					
				Add a logging functionality for failed scrapping tasks	5		5	
				Create components for the chat screen	8		8	
				Create short video as status update for IP and the Mid-Project product demo	3		3	
				Write unit tests for orchestrator/config functionality	5			
				Create components for basic visual navigation	8		8	
				Create components for login with email	8		8	
				Implement deep-linking	5		5	
				Implement form framework for the login workflow	5		5	
				Setup user authentication	8		8	
8	2024-06-12	2024-06-19	Develop app components					
				Write unit tests for orchestrator/config functionality				
				Present references in agent answers	8			
				Create UI components to allow user to manually store custom instructions	8		8	
				Save conversations in the DB	5		5	
				Create option to change LLM	5			
				Create option to save / export the conversation as formated txt	5			
				Present recent conversations	5		5	
				Create documentation on how the user can run the build and start testing the app	5		5	
				Create dropdown UI for conversation options	5		5	
				Integrate the app components with the navigation component	5		5	
9	2024-06-19	2024-06-26	Component integration and parallel answers					
				Write unit tests for orchestrator/config functionality				
				Present references in agent answers				
				Create option to change LLM				
				Create option to save / export the conversation as formated txt				
				Create documentation in the wiki for devs on how to use the Firestore DB	3		3	
				Query and display multiple LLM responses simultaneously - frontend	8			
				Integrate chat in the main app	5		5	
				Integrate custom user context input components into main navigation	5			
				Implement multiselect option for LLM selection dropdown	5		5	
				Implement fundamental prompts	5			
				Data retrieval - recipes from Nutritionfacts	8			
				Scrum spike: Research prompt engineering techniques	5		5	

					Est. Size	Est. Remaining	Real Size	Real Remaining
				Complete Project	512	232	312	312
				Final Project Release	280	280	101	101
				Sprints 07-12	280	280	101	101
Sprint	**Start Date**	**End Date**	**Sprint Goal**	**Feature Name**		**Est. BD**		**Real BD**
7	2024-06-05	2024-06-12	Develop basic app components		55	280	50	101
8	2024-06-12	2024-06-19	Develop app components		51	225	33	51
9	2024-06-19	2024-06-26	Component integration and parallel answers		44	174	18	18
10	2024-06-26	2024-07-03	Voice Integration, modular initialisation prompt		46	130	0	0
11	2024-07-03	2024-07-10	Modular initialisation prompt, multiple LLMs, memory		56	84	0	0
12	2024-07-10	2024-07-17	Stability, handoff		28	28	0	0
13	2024-07-17	2024-07-24	AMOS Closing		0	0	0	0
Sprint	**Start Date**	**End Date**	**Sprint Goal**	**Feature Name**	**Est.Size**		**Real Size**	
10	2024-06-26	2024-07-03	Voice Integration, modular initialisation prompt					
				Create option to change LLM				
				Present references in agent answers				
				Create option to save / export the conversation as formated txt				
				Query and display multiple LLM responses simultaneously - frontend				
				Integrate custom user context input components into main navigation				
				Implement fundamental prompts				
				Data retrieval - recipes from Nutritionfacts				
				Voice interaction - agent reply voice output - TTS	8			
				Voice Interaction - user voice input	8			
				Develop option to create new chat	5			
				Fix issue with custom context retrieval	5			
				Enable the fetching of an initialisation prompt from Google Docs	5			
				Create an initialisation prompt	5			
				Save answers provided to the initialisation prompt as custom user-context	5			
				Enable the change of the initialisation prompt	5			
11	2024-07-03	2024-07-10	Modular initialisation prompt, multiple LLMs, memory					
				Write unit tests for orchestrator/config functionality				
				Enable the fetching of an initialisation prompt from Google Docs				
				Create an initialisation prompt				
				Save answers provided to the initialisation prompt as custom user-context				
				Enable the change of the initialisation prompt				
				Query and display multiple LLM responses simultaneously - backend	8			
				Save responses from multiple LLMs	8			
				Pass the custom context from the common-user input to the LLM	8			
				Develop workaround for Pubmed API	5			
				Integrate LLM3: Llama	8			
				Integrate LLM4: Claude	5			
				Implement conversational memory	8			
				Deploy Docker Container in Google Cloud	3			
				Run scrapers in Google Cloud	3			
12	2024-07-10	2024-07-17	Stability, handoff					
				Update wiki documentation with new app functionality	5			
				Bug fixing	13			

					Est. Size	Est. Remaining	Real Size	Real Remaining
				Complete Project	512	232	312	312
				Final Project Release	280	280	101	101
				Sprints 07-12	280	280	101	101
Sprint	**Start Date**	**End Date**	**Sprint Goal**	**Feature Name**		**Est. BD**		**Real BD**
7	2024-06-05	2024-06-12	Develop basic app components		55	280	50	101
8	2024-06-12	2024-06-19	Develop app components		51	225	33	51
9	2024-06-19	2024-06-26	Component integration and parallel answers		44	174	18	18
10	2024-06-26	2024-07-03	Voice Integration, modular initialisation prompt		46	130	0	0
11	2024-07-03	2024-07-10	Modular initialisation prompt, multiple LLMs, memory		56	84	0	0
12	2024-07-10	2024-07-17	Stability, handoff		28	28	0	0
13	2024-07-17	2024-07-24	AMOS Closing		0	0	0	0
Sprint	**Start Date**	**End Date**	**Sprint Goal**	**Feature Name**	**Est.Size**		**Real Size**	
				Update user / wiki documentation for Handoff	5			
				Component integration	5			
13	2024-07-17	2024-07-24	AMOS Closing					
				Project Report				
				Project Retrospective				

Sprint	Sprint Goal	Sprint Start Date	Sprint Finish Date	Ideal SP / sprint	Ideal SP remaining	Estimated SP / sprint	Estimated SP remaining	Actual SP / sprint	Actual SP remaining
1	AMOS Kickoff	2024-04-17	2024-04-24	0	512	0	512	0	512
2	Project set-up and start with data acqui	2024-04-24	2024-05-08	47	512	32	512	32	512
3	CI/CD pipeline set-up, progress with dat	2024-05-08	2024-05-15	47	465	27	480	22	480
4	Progress data acquisition and storage f	2024-05-15	2024-05-22	47	419	72	453	62	458
5	Finalise data acquisition functionality, p	2024-05-22	2024-05-29	47	372	40	381	35	396
6	Finalise data acquisition, finalise basic	2024-05-29	2024-06-05	47	326	61	341	60	361
7	Develop basic app components	2024-06-05	2024-06-12	47	279	55	280	50	301
8	Develop app components	2024-06-12	2024-06-19	47	233	51	225	33	251
9	Component integration and parallel ans	2024-06-19	2024-06-26	47	186	44	174	18	218
10	Voice Integration, modular initialisation	2024-06-26	2024-07-03	47	140	46	130	0	
11	Modular initialisation prompt, multiple L	2024-07-03	2024-07-10	47	93	56	84	0	
12	Stability, handoff	2024-07-10	2024-07-17	47	47	28	28	0	
13	AMOS Closing	2024-07-17	2024-07-24	0	0	0	0	0	0




Velocity
Ideal SP / sprint
Estimated SP / sprint
Actual SP / sprint
Story Points
Sprint
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10 11 12 13
0 0 0
47 32 32
47 27 22
47 72 62
47 40 35
47 61 60
47 55 50
47 51 33
47 44 18
47 46 0
47 56 0
47 28 0
0 0 0

Type	Link / reference
Product Wiki	https://github.com/amosproj/amos2024ss06-health-ai-framework/wiki

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
1	The feature has been fully implemented.	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 feature utility, function and usage have been documented in the respective project wiki on github.		

#	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
24	App development	react-native-paper	5.12.3	MIT	App UI
25	App development	@react-navigation/native	6.1.17	MIT	App Navigation
26	App development	@react-navigation/drawer	6.6.15	MIT	App Drawer Navigation
27	App development	react-native	0.74.2	MIT	Basic react native functionality
28	App development	react-native-safe-area-context	4.10.4	MIT	Make app respect phone specific boundaries
29	App development	react	18.3.1	MIT	Basic react functionality
30	App development	react-native-vector-icons	10.1.0	MIT	App Icons
31	App development	@expo/vector-icons	14.0.2	MIT	App Icons
32	App development	react-native-element-dropdown	2.12.0	MIT	Dropdown for selecting
33	App development	react-native-gesture-handler	2.16.1	MIT	ScrollView
34	App development	@react-native-google-signin/google-signin	12.2.0	MIT	Google Signin
35	App development	@react-navigation/native-stack	7.0.0-alpha.20	MIT	useNavigation hook --> reset
36	App development	@react-native-firebase/auth	20.1.0	Apache-2.0	firebase authentification
37	App development	reactfire	4.2.3	MIT	useUser, useAuth hooks
38	App development	@react-native-voice/voice	3.2.4	MIT	voice input and recognition
39	App development	expo-speech	12.0.2	MIT	voice output for chats
40	App development	expo-clipboard	6.0.3	MIT	copy chat to clipboard
41	App development	expo-file-system	17.0.1	MIT	saving chat to storage
42	App development	expo-media-library	16.0.4	MIT	saving chat to storage
43	App development	react-native-fs	2.27.1	MIT	saving chat to storage