Project Name	...
Production system (if any)	...
Test system (if any)	...
GitHub repository	https://github.com/amosproj/amos-ss2021-synthetic-file-system
GitHub kanban board (project)	https://github.com/amosproj/amos-ss2021-synthetic-file-system/projects/1
Team T-shirt (white)	https://www.shirtinator.de/loadBasket/x7rBfP1clL-
Team T-shirt (black)	...
Additional materials	...

Last Name	First Name	GitHub User Name	Email Address
Mallick	Tuhin	tuhinmallick	tuhin.mallick@fau.de
Abraham	Finley	finleymercyabraham	finley.abraham@fau.de
Weghorn	Marlon	marlonweghorn	marlon.weghorn@fau.de
Bafna	Vaidehi	vaidehi-for-coding	vaidehi.bafna@fau.de
Schoepf	Dominik	DOMEscho	dominik.schoepf@fau.de
Schulze	Matti	ma-schulze	matti.schulze@fau.de
Srikhaolan	Charinee	CharineeSrikhaolan	Charinee.srikhaolan@fau.de
Arbo	Sandra	sandraaocode	sandra.arbo@fau.de

Goals	
	Implement a successful project by adhering to the project plan and roadmap
	Conduct team meetings
	Learn and execute the tasks and responsibilities of Agile Methods
	Get an insight into the project roles
	Gather full business requirements of the training app from industry owner
Meeting norms	
	Mandatory meetings from Wednesday 12.30, everybody should participate in meetings regularly
	One additional weekly meeting like Sunday, everybody should participate in general (at least two people)
	Inform other team members if someone will be late and take accountability
Working norms	
	If one cannot fulfill the task due to the time constraints or for any other reason, then informing the team well ahead would be expected
	Difficult tasks can be solved as a group, but split the tasks as far as possible to work independently
	Raise project related queries to the product owner and seek regular confirmations and approvals
	Finish all the assigned tasks meeting the project deadlines
	Pull principle : Everyone can take the task that they like to do
Coordination norms	
	Team Coordination over WhatsApp and MS Teams
	Project Owner acts as the Leader of the team meetings and ensures the project runs on track.
	Allocate assignments: Nobody has to do all the tasks, which he or she does not want to do. However, keep track on the tasks which nobody wants to do it, allocate the task to the right person.
	Be responsible and respectful to each other
Communication norms	
	Communicate over MS Teams and WhatsApp (and Zoom in case of connection issues)
	Upload the project documentation in the designated project Tool
	A quick response or acknowledgement is expected. However, inform team members well ahead if someone is not available, quick answers to a topic should be possible for everybody
	Inform team members if possible, the day before if one cannot participate in the meeting
	Communicate beforehand your personal issues (e.g., Examination period, excuses afterwards are not accepted)
Consideration norms	
	Discuss relevant topics during the meetings
	Have a vote on agreement / disagreement in case of differences in ideas and opinions
Cont. improvement norms	
	Track team’s progress by conducting status update meetings regularly (protocol)
	Upload / share lecture slides, documents, and other information in MS Teams
Rewards	
	Celebrate special efforts or MVP of every sprint in the HALL OF FAME
Sanctions	
	We will make a list of the backlogs, and at the end of the project bring chocolates, make their place in the HALL OF SHAME.

#	Meeting Day	Comment	Coach	Product Owner	Software Developer	Scrum Master	Release Manager
1	2021-04-14		Yes		Everyone else	N/A	N/A
2	2021-04-21		Yes	Tuhin	Everyone else	Charinee	Dominik
3	2021-04-28		Yes	Finley	Everyone else	Sandra	Matti
4	2021-05-05			Tuhin	Everyone else	Marlon	Vaidehi
5	2021-05-12		Yes	Finley	Everyone else	Tuhin	Sandra
6	2021-05-19			Tuhin	Everyone else	Matti	Charinee
7	2021-05-26	Mid-term due	Yes	Finley	Everyone else	Dominik	Marlon
8	2021-06-02			Tuhin	Everyone else	Charinee	Dominik
9	2021-06-09			Finley	Everyone else	Vaidehi	Matti
10	2021-06-16		Yes	Tuhin	Everyone else	Finley	Vaidehi
11	2021-06-23			Finley	Everyone else	Sandra	Charinee
12	2021-06-30			Tuhin	Everyone else	Matti	Sandra
13	2021-07-07		Yes	Finley	Everyone else	Dominik	Marlon
14	2021-07-14	Demo day!					
15	2021-07-21	Retrospective					

Product Vision	Project Mission
The rapidly growing mass of data requires further refinements and new technologies in order to find the right data in this deluge of information. The creation and management of metadata is decisive for representing the content of stored objects and files. This metadata is held in a database for instant retrieval. Lists can be constructed from these databases to find files and objects in general, but they do not yet afford access to the real data. For this reason, a synthetic file system is considerably useful. A synthetic file system enables to access data, chosen by queries in the database of metadata. Retrieval of project-related data is done through a distributed database and a virtual file system that permits a single namespace for all relevant data designated by their metadata.	Due to the current Corona pandemic, as much data as possible is to be analyzed and evaluated with the help of artificial intelligence (AI). This requires a central intelligence to collect and interpret all accessible data distributed over several facilities. The key issue is that the data is organized and saved in different systems according to different storage types, structures, formats and criteria. The task, or rather mission is now to first make the data obtainable or readable via a uniform mechanism within the project time-frame. This would have the benefit of not having to adapt each application to the different memory types. The synthetic file system is a so-called adapter for each type of memory, so that a unified namespace can be formed from it.

Term	Definition
Filesystem	Procedure/protocol to store and retreive a computer file
FUSE Filesysten in userspace	An interface through which a developer can communicate with the Kernel Modus without knowing "Kernel' language "
Meta data	Data about a computer file ex. date of creation, modification or deletion, type of file
POSIX compliant	The Portable Operating System Interface is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems

#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down
1	Project Organisational Features				29		29	92
		Provide the organisational features						
			Information on prior experience	2		2		
			Role Assignment	1		1		
			T-Shirt Logo	8		8		
			Team Contract	13		13		
			Additional Team Meetings	2		2		
			Clone Client Repository	3		2		
2	Setting up the Docker				10		10	63
		Creating the DOCKER						
			Set up Docker Skeleton (Linux fuse) resembles	3		3		
			Configure the docker to allow X forwarding	5		5		
			Setup a Feature branch in GIT	2		2		
3	Setting up the FUSE				17		8	55
		Creating the basic Framework of FUSE						
			Implement basic directory functionalities in the FUSE	8		5		
			Create a FUSE metaHub bridge	5		3		
4	MetaData Hub				10		6	49
		Getting the new FUSE Code repository						
			Migrate the code base to the new metadata hub	5		3		
			Metadata Hub	5		3		
5	Automated Testing				8		8	41
		Testing should be automated to reduce redudancy						
			Create Unit Tests that cover the most important functionality.	5		5		
			Project linter and CI setup	3		3		

#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down
1	Project Organisational Features				29		29	92
		Provide the organisational features						
			Information on prior experience	2		2		
			Role Assignment	1		1		
			T-Shirt Logo	8		8		
			Team Contract	13		13		
			Additional Team Meetings	2		2		
			Clone Client Repository	3		2		
2	Setting up the Docker				28		28	63
		Creating the DOCKER						
			Set up Docker Skeleton (Linux fuse) resembles	3		3		
			Configure the docker to allow X forwarding	5		5		
			Setup a Feature branch in GIT	2		2		
		Update DOCKER						
			Update DOCKER file and DOCKER compose running	5		5		
			Integration tests for py-test for docker and docker-compose	13		13		
3	SFS Functionalities				31		18	55
		Creating the basic Framework of FUSE						
			Implement basic directory functionalities in the FUSE	8		5		
			Create a FUSE metaHub bridge	5		3		
			Creating test for FUSE (read access)	5		3		
			Improve Error handling	5		5		
		Filter Implementation						
			Implementation of filtering dataset prototype	8		5		
			Implement functionalities to open files from/within the FUSE	5		3		
			Create an updated start script for the SFS	3				
			Make the building of the file tree modular	5				
			Implement WRITE support for SFS	5				
			Parser from Config toml to Graph-QL/Modular	5				
4	MetaData Hub				18		11	49
		Getting the new FUSE Code repository						
			Migrate the code base to the new metadata hub	5		3		
			Metadata Hub	5		3		
			Creating a modular interface for supporting multiple backends	8		5		
			Exception handling and Test for MDH query	3				
			Create an xattr command/script	5				
5	Automated Testing				8		8	41
		Testing should be automated to reduce redudancy						
			Create Unit Tests that cover the most important functionality.	5		5		
			Project linter and CI setup	3		3		
6	Debian System							
		Setting up the debian system			16		8	
			Configure and familiarise with the first Debian System	8		8		
			Set-up automatic testing on the Debian system	8				
7								41

#	Theme	Goal	Feature Name	Est. Size (Feature)	Est. Size (Sprint)	Real Size (Feature)	Real Size (Sprint)	Burn-Down

Sprint	Status	Source	Impediment	Resolution
1	Resolved	Tuhin, Sandra	Meetings are not clearly arranged.	Arranged fixed meetings: (We already have fixed internal meetings (Wednesdays 12:30 and Sundays 16:00) and two fixed external meetings (Mondays at 10:00 and Wednesdays at 15.00)
1	Resolved	Dominik, Vaihedi	Set ups/Requirements for the synthetic file system are not clear.	Refer to the project description, PowerPoint Slides (GRAU Data) and uploaded Protocols (when reading them all together probably it is going to be more clear) and on 26.04 there will be a clarification on the requirements
1	Resolved	Finley, Charinee	No written protocol for meetings are set up.	Sandra has written a protocol for every meeting, and uploaded it into Teams and Google Drive folder
3	Resolved	Finley	Sunday meetings at 16:00 are not possible for him	Tried a doodle survey but due to the very different time schedules it wasn't possible to find a better time and day.
4	Resolved	Charinee	Specific task to be assigned and later be helped out	The task was allocated and later on, if any impedimeent is created, later on helped on
5	Resolved	Tuhin	Change timing of meetings	
5	Resolved	Sandra	Not clear definition of all the issues	
6	Resolved	Finley	POs need one more meeting for themselve in a sprint	One meeting on Thursday afternoon(15:00) to have a quick update on developed features/funtionalities of the program and to prepare for the next meeting with Ulrich

Type	Link / reference
Software Build and Deployment Documen	https://github.com/amosproj/amos-ss2021-synthetic-file-system/wiki/Synthetic-File-System-Build-and-Deployment-Documentation
User Documentation	https://github.com/amosproj/amos-ss2021-synthetic-file-system/wiki/Synthetic-File-System-User-Documentation
Technical Documentation	https://github.com/amosproj/amos-ss2021-synthetic-file-system/wiki/Synthetic-File-System-Technical-Documentation
Software Architecture	https://github.com/amosproj/amos-ss2021-synthetic-file-system/wiki/Software-Architecture-Description

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
1	Feature Code review from peers has been completed and passed	Code review from peers has been completed and passed	Code review from peers has been completed and passed
	a. Code is fully implemented and commented	*a. Code is fully implemented and commented*	*a. Code is fully implemented and commented*
	b. Adheres to coding guidelines	*b. Adheres to coding guidelines*	*b. Adheres to coding guidelines*
	c. Commits are not squashed	*c. Commits are not squashed*	*c. Commits are not squashed*
	d. Code is checked into the repository	*d. Code is checked into the repository*	*d. Code is checked into the repository*
2	Unit tests for feature have been written and are passing	Unit tests/ Integration test have been written and are passing	Unit tests/ Integration test have been written and are passing
3	Cleanliness of the Code (Readability)	Cleanliness of the Code (Readability)	Cleanliness of the Code (Readability)
4	No critical bugs are open	No critical bugs are open	No critical bugs are open
5	Feature branch has been tagged and merged	Database consistency checks in test environment succeeded	Software documentation passes external review
6	Feature Code has been included into the release (candidate)	Code has been included into the release (candidate)	User manual passes external review
7	Code Coverage:	Code Coverage:	The code and readme on Github is well documented and easily readible
	40 % - for Features	50 % - for Sprint Release	Code Coverage:
			60 % - for Product Release

\	Context	Name	Version	License	Comment
1	OS	Docker	20.1	Apache License 2.0	
2	OS	ubuntu	20.04	GPL-3.0 License	
3	Linting	flake8	3.9.1.	MIT License	
4	Testing	pytest	6.2.3	MIT License	
5	Test coverage	pytest-cov	2.11.1	BSD License (MIT)	
6	OS	Docker-compose	1.27	Apache License 2.0	

Last Name	First Name	Value					
Mallick	**Tuhin**			#DIV/0!	#N/A		
Abraham	**Finley**						
Weghorn	**Marlon**						
Bafna	**Vaidehi**						
Schoepf	**Dominik**			0	No effort		
Schulze	**Matti**			1	Minimal effort		
Srikhaolan	**Charinee**			2	Small effort		
Arbo	**Sandra**			3	Medium effort		
				5	Large effort		
				8	Very large effort		
				13	Too large effort		
{__/}	{__/}	{__/}	{__/}	{__/}	{__/}	{__/}	
(● ●)	(● ●)	(● ●)	(● ●)	(● ●)	(● ●)	(● ●)	
(> Want a taco?	(> Want a taco?	(> Want a taco?	(> Want a taco?	(> Want a taco?	(> Want a taco?	(> Want a taco?	