Project Name	Xcelerator Demo App
Online team meeting	https://fau.zoom-x.de/j/68734239815
Production system (if any)	TBD
Test system (if any)	TBD
GitHub repository	https://github.com/amosproj/amos2024ss01-xcelerator-demo-app
GitHub feature board	https://github.com/orgs/amosproj/projects/43/views/2
GitHub impediments backlog	
Male Team T-shirt (white)	https://www.shirtinator.de/s/ZyMdmUG5TpCQTKgMQLhArQ
Male Team T-shirt (black)	https://www.shirtinator.de/s/Gpdz8IxTT7-4qgmoQZBK5w
Female Team T-shirt (white)	https://www.shirtinator.de/s/rfxFlAlxTlebR-mo-t182g
Female Team T-shirt (black)	https://www.shirtinator.de/s/tBsO50F8STmiLdLYU9WCCQ
Additional materials	/
Team maling list	oss-amos-proj1@lists.fau.de

Last Name	First Name	GitHub User Name	Email Address
Lorenz	Alexander	Hydraneut	alex.lorenz.1@gmx.de
Rajjo	Lama	lamara11	lama.l.rajjo@fau.de
Sternberg	Ingo	IngoSternberg	ingo.sternberg@fau.de
Betancourt Barrita	Cecilia	ceciliabetb	cecilia.betancourt@fau.de
Heisterberg	Jonas	Persists	jonas@mennicke.com
Krug	Maximilian	HaruspexSan	krugm03@gmail.com
Nasir	Shahraz	shahraz1998	nasirsharaz@gmail.com
Schmidt	Patrick	PatrickSchm1dt	patrick.m.schmidt@fau.de
Bojanić	Saša	Sabo2k	saleb2k@gmail.com
Schmidt	David	KonsumGandalf	d.schmidt@campus.tu-berlin.de / david_scl

#	Meeting Day	Product Owners	Software Developer	Release Manager	Scrum Master	Comment
1	2024-04-17	Lama Rajjo, Shahraz Nasir	Everyone else	N/A	Cecilia Betancourt Barrita	
2	2024-04-24	Lama Rajjo, Shahraz Nasir	Everyone else	TBD	Cecilia Betancourt Barrita	
3	2024-05-01	Lama Rajjo, Shahraz Nasir	Everyone else	TBD	Cecilia Betancourt Barrita	
4	2024-05-08	Lama Rajjo, Shahraz Nasir	Everyone else	Saša Bojanić	Cecilia Betancourt Barrita	
5	2024-05-15	Lama Rajjo, Shahraz Nasir	Everyone else	Jonas Heisterberg	Cecilia Betancourt Barrita	
6	2024-05-22	Lama Rajjo, Shahraz Nasir	Everyone else	Maximilian Krug	Cecilia Betancourt Barrita	
7	2024-05-29	Lama Rajjo, Shahraz Nasir	Everyone else	Alexander Lorenz	Cecilia Betancourt Barrita	Mid-term due
8	2024-06-05	Lama Rajjo, Shahraz Nasir	Everyone else	David Schmidt	Cecilia Betancourt Barrita	
9	2024-06-12	Lama Rajjo, Shahraz Nasir	Everyone else	Patrick Schmidt	Cecilia Betancourt Barrita	
10	2024-06-19	Lama Rajjo, Shahraz Nasir	Everyone else	Ingo Sternberg	Cecilia Betancourt Barrita	
11	2024-06-26	Lama Rajjo, Shahraz Nasir	Everyone else	Saša Bojanić	Cecilia Betancourt Barrita	
12	2024-07-03	Lama Rajjo, Shahraz Nasir	Everyone else	Jonas Heisterberg	Cecilia Betancourt Barrita	
13	2024-07-10	Lama Rajjo, Shahraz Nasir	Everyone else	Maximilian Krug	Cecilia Betancourt Barrita	
14	2024-07-17	Lama Rajjo, Shahraz Nasir	Everyone else	Alexander Lorenz	Cecilia Betancourt Barrita	Demo day!
15	2024-07-24	Lama Rajjo, Shahraz Nasir	Everyone else	David Schmidt	Cecilia Betancourt Barrita	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	
	Fulfilling the requirements from the client
	Deliver a quality demo
	SMART principles
	Have fun
	Improve coding skills
Meeting norms	
	Punctuality
	Inform others of your absence or if you are late (WhatsApp)
	Openness
	Give feedback
	Active participation
	Meetings should have agendas
	Best practice is keeping the camera on; do what you want in call but do not interrupt the workflow
Working norms	
	Respect deadline
	Inform early to the team if you are struggling to complete a task
	Seek help or clarification
	Be efficient
	When doing flow charts use draw.io
Coordination norms	
	We use the StandUp-Emails
	Interact actively with the Industry partner during meetings
Communication norms	
	Honesty
	If there is any problem, communicate it on time
	Check discord at least Bi-daily
	Whatsapp: Used for small messages, stuff that does not need repleis
	Discord for general communication: pair coding
	Zoom: team-Meetings and Partner Meetings
	Avoid side conversations & interruptions
Consideration norms	
	Patience
	Accept some level of knowledge gap
	Be aware of the differences
	Add roles - "Most kills" - most bugfixes, "Most assists" - most supportive for others, "Headhunter" - Solving most tickets
Cont. improvement norms	

	Be available to take up various kinds of tasks
	Learn from each other
	Keep in touch with the industry partner(We collect Questions via discord and ask them via E-Mail)
	After commit ad at least 2 developers for code review
Rewards	
	Recognition
	Achievements/Titles:
	Most-Assists: Most-Helpfull TeamMember
	Most-Kills: Most-Bug fixes
	HeadHunter: Most-Tasks-Cleared
Sanctions	
	After breach of contract you have to wear a silly tie to the next team-meeting
Signatures	
Scrum Master	Cecilia Betancourt Barrita
Product owner	Lama Rajjo
Product owner	Shahraz Nasir
Software developer	Alexander Lorenz
Software developer	Ingo Sternberg
Software developer	Jonas Heisterberg
Software developer	David Schmidt
Software developer	Patrick Schmidt
Software developer	Saša Bojanić
Software developer	Maximilian Krug

Product Vision	Project Mission
Due to their positive alignment with global sustainability objectives, IoT technologies have been adopted in several industries. One such example is the asset resource management area, where the use of energy systems (and the included devices) needs to be closely monitored to assure safety and efficiency. Since temporary failures of IoT devices can have substantial consequences, allowing facility managers to quickly ascertain and react to the status of their assets is pivotal. Our application intends to address this issue by leveraging the Siemens Xcelerator ecosystem, which offers reliability and intelligence in facility management. Our solution will provide a centralized platform for tracking the status of all facilities, ensuring that critical issues are prioritized and resolved promptly. As a result, operational costs will encounter a reduction, while the lifespan of facility assets will extend. Moreover, it will be an example of what can be achieved with the Siemens Xcelerator platform, serving as an educational tool, and creating a robust foundation for developers to build upon.	The mission of this project is to develop an application streamlining the monitoring and management of facilities, allowing managers to rapidly identify and address issues such as faulty pumps or tanks. Moreover, they will have access to comprehensive data on their assets, enabling them to make data-driven decisions about repairs, replacements, and upgrades. In addition, we aim to showcase how to employ the Siemens Xcelerator ecosystem to monitor and manage multiple facilities, highlighting key features such as real-time data integration, predictive maintenance, and streamlined operations.

Term	Definition
IoT devices	Technologies that behave and adjust their status in a predefined manner based on live data
Facility	A location, typically a large building, having IoT devices installed
Faulty	Adjective describing a facility suffering from one or more device failures
Work order	A request for support, repairment, or maintenance regarding one or more device failures in a facility
Facility Manager	The person who needs/wants/can inspect the status a certain facility, and create work orders for the ones with device failures.
Case	A work order submitted for a facility suffering from one or more device failures.

Sprint #	Sprint goal
1	None
2	None
3	None
4	Frame application screens refinement
5	Backend optimizations and authentication
6	Integrating Backend and Frontend
7	Frontend/Backend refactoring
8	Connecting to Insights Hub APIs
9	
10	
11	
12	
13	
14	
15	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			136	136		
Sprints						
1	**Research and team organization**		0	136	0	136
2	**Fundamental setup and definition of resources**		18	136	27	136
3	**Frame application screens skeleton**		25	118	30	109
4	**Frame application screens refinement**		22	93	24	79
5	**Backend optimizations and authentication**		29	71	0	55
6	**Integrating Backend and Frontend**		42	42	0	55
Features						
1	**Research and team organization**					
		Design and upload team logo	-		-	
		Create T-shirt mockup	-		-	
		Research of fundamentals	-		-	
2	**Fundamental setup and definition of resources**					
		Software architecture document	5		5	
		Fill in the bill od materials	2		2	
		Github Pipelines	2		5	
		Nx Setup	3		5	
		Config based service implementation in Angular	3		5	
		Config service for NEST	3		5	
3	**Frame application screens skeleton**					
		Build process review	3		3	
		Automate release process	5		5	
		Frame Application Screen: Homepage (Facilities with errors)	3		5	
		Frame Application Screen: Facilities Overview	3		2	
		Navigation/Routing	3		5	
		Set up Tailwind	2		2	
		Setting up Prisma	3		3	
		Frame Application Screen: Order Form	3		5	
4	**Frame application screens refinement**					
		Automate the bill of materials	3		3	
		Frame Application Screen: Order Form	3		5	
		Frame Application Screen: Implement Details Page	5		5	
		Authentication: Login Page	5		5	
		Add Favicon in App	1		1	
		Deployment of the Application to a free Infrastructure	3		3	
		Switch from card view to event list view in the List all facilities page	2		2	
5	**Backend optimizations and authentication**					
		Define product vision	-		-	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
		Define product mission	-		-	
		Create Build Process Video	2		0	
		Rename "work orders" to "cases" throughout the application	2		0	
		Show selected facility on the right side of the work orders form page	3		0	
		Implement Work Order view	3		0	
		Back button for headers from details to facility overview	2		0	
		Implement Case CRUD operations	8		0	
		Implement Mock Server for IOT Timeseries API in NestJS	8		0	
		Make charts bigger in details page	1		0	
6	**Integrating Backend and Frontend**					
		Connect Frontend to Backend	8		0	
		Error handling in backend	5		0	
		Case Crud operations frontend	8		0	
		Display information box for facilities in details page	3		0	
		Make charts show multiple data in Y axis	3		0	
		Refactor headlines of webpages	3		0	
		Connect IoT service to local service	8		0	
		Description of the product on homepage, Fill about and legal information	2		0	
		Write user, (technical) design, and build/deploy documentation	2		0	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
Release						
Total			147	147		
Sprints						
1	**Research and team organization**		0	147	0	147
2	**Fundamental setup and definition of resources**		18	147	27	147
3	**Frame application screens skeleton**		25	129	30	120
4	**Frame application screens refinement**		22	104	24	90
5	**Backend optimizations and authentication**		29	82	29	66
6	**Integrating Backend and Frontend**		30	53	28	37
7	**Frontend/Backend refactoring**		23	23	27	9
8	**Connecting to Insights Hub APIs**		36	0	53	-18
9	**Final UI optimizations**		12	-36	0	-71
10	**Pre-deployment testing**		13	-48	0	-71
11	**Deployment of the app on AWS**		8	-61	0	-71
12	**Preparations for Demo Day and documentation**		9	-69	0	-71
Features						
1	**Research and team organization**					
		Design and upload team logo	-		-	
		Create T-shirt mockup	-		-	
		Research of fundamentals	-		-	
2	**Fundamental setup and definition of resources**					
		Software architecture document	5		5	
		Fill in the bill od materials	2		2	
		Github Pipelines	2		5	
		Nx Setup	3		5	
		Config based service implementation in Angular	3		5	
		Config service for NEST	3		5	
3	**Frame application screens skeleton**					
		Build process review	3		3	
		Automate release process	5		5	
		Frame Application Screen: Homepage (Facilities with errors)	3		5	
		Frame Application Screen: Facilities Overview	3		2	
		Navigation/Routing	3		5	
		Set up Tailwind	2		2	
		Setting up Prisma	3		3	
		Frame Application Screen: Order Form	3		5	
4	**Frame application screens refinement**					
		Automate the bill of materials	3		3	
		Frame Application Screen: Order Form	3		5	
		Frame Application Screen: Implement Details Page	5		5	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
		Authentication: Login Page	5		5	
		Add Favicon in App	1		1	
		Deployment of the Application to a free Infrastructure	3		3	
		Switch from card view to event list view in the List all facilities page	2		2	
5	**Backend optimizations and authentication**					
		Define product vision	-		-	
		Define product mission	-		-	
		Create Build Process Video	2		2	
		Rename "work orders" to "cases" throughout the application	2		2	
		Show selected facility on the right side of the work orders form page	3		3	
		Implement Work Order view	3		3	
		Back button for headers from details to facility overview	2		1	
		Implement Case CRUD operations	8		8	
		Implement Mock Server for IOT Timeseries API in NestJS	8		8	
		Make charts bigger in details page	1		2	
6	**Integrating Backend and Frontend**					
		Connect Frontend to Backend	8		8	
		Display information box for facilities in details page	3		3	
		Make charts show multiple data in Y axis	3		3	
		Refactor headlines of webpages	3		3	
		Connect Siemens API to the IoT service	5		5	
		Maintenance of architecture overview	5		2	
		Signal handling workshop	1		1	
		Write user, (technical) design, and build/deploy documentation	2		3	
7	**Frontend/Backend refactoring**					
		Adjust data collected in the cases form	3		3	
		Support: Assist in BE and FE integration	1		1	
		Workshop: Database Seeding	1		2	
		Error handling in backend	5		3	
		Case CRUD Operations Frontend	8		13	
		Description of the product on homepage, Fill about and legal information	2		2	
		Cases Details Page	3		3	
8	**Connecting to Insights Hub APIs**					
		Workshop at siemens building	-		-	
		Prepare for Demo App presentation at siemens building	-		.	
		Connect IoT service to local service	8		13	
		Refactoring time series API	3		3	
		Update readme.md to procide entry point to the project	3		3	
		Show actual pump data on the charts in details page	3		3	
		Service-like feature for FE pages of facilities to BE	5		5	
		Content: Cases screen	3		8	
		Connect cases details page to backend	5		8	
		Validation of selection boxes in Orders form	3		5	
		Husky fix, Docker build	1		3	
		Integrate swagger open api	2		2	

Sprint	Goal	Feature Name	Est. Size	Est. Remaining	Real Size	Real Remaining
9	**Final UI optimizations**					
		Details page responsiveness	3		0	
		Facilities page responsiveness	3		0	
		Orders page responsiveness	3		0	
		Theme colors	3		0	
10	**Pre-deployment testing**					
		Write tests for backend	5		0	
		Improve error-handling for backend	5		0	
		Fallbacks for API calls	3		0	
11	**Deployment of the app on AWS**					
		Deploy on AWS (IP account)	5		0	
		Post-deployment Testing	3		0	
12	**Preparations for Demo Day and documentation**					
		Prepare demo day video	3		0	
		Prepare demo day slides	3		0	
		Finalize documentation	3		0	

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done
	A Feature is done when all acceptance criteria are met.	A Sprint Release is done when the Features awaiting review meet their individual DoD and AC.	The Project Release is done when there are no more fixes left to be done in the code.
	A Feature is done when its code has been reviewed and merged into the primary development branch.	A Sprint Release is done when all team members agree for its publishing.	The Project Release is done when it is building and running succesfully on all the desired environments and machines.
	A Feature is done when its unit/integration tests are passing.	A Sprint Release is done when the project is not outputting or displaying any errors.	The Project Release is done when the team members agree that the related documentation is comprehensive enough.
	A Feature is done when the related pipeline is building successfully.	A Sprint Release is done when the pipeline running after its last commit on the primary development branch completes without errors.	The Project Release is done when all stakeholders, including the Industry Partner, approve of the release state.

Type	Link / reference
Github-Wiki	https://github.com/amosproj/amos2024ss01-xcelerator-demo-app/wiki

#	Context	Name	Version	License	Comment
1	frontend	angular/animations	~17.3.0	MIT License	
2	frontend	angular/common	~17.3.0	MIT License	
3	frontend	angular/compiler	~17.3.0	MIT License	
4	frontend	angular/core	~17.3.0	MIT License	
5	frontend	angular/forms	~17.3.0	MIT License	
6	frontend	angular/platform-browser	~17.3.0	MIT License	
7	frontend	angular/platform-browser-dynamic	~17.3.0	MIT License	
8	frontend	angular/router	~17.3.0	MIT License	
9	frontend	faker-js/faker	^8.4.1	MIT-license	
10	frontend	nestjs/common	^10.0.2	MIT-license	
11	frontend	nestjs/config	^3.2.2	MIT	
12	frontend	nestjs/core	^10.0.3	MIT-license	
13	frontend	nestjs/platform-express	^10.0.4	MIT-license	
14	frontend	prisma/client	^5.13.0	Apache-2.0	
15	frontend	nx/angular	18.3.4	MIT-license	
16	frontend	siemens/ix	^2.2.1	MIT	
17	frontend	siemens/ix-angular	^2.2.1	MIT licensed	Copyright © 2022 Siemens AG
18	frontend	siemens/ix-echarts	^2.2.0	MIT	
19	frontend	siemens/ix-icons	^2.1.0	MIT	
20	frontend	axios	^1.6.8	MIT License	Copyright (c) 2014-present Matt Zabriskie & Collaborators
21	frontend	class-validator	^0.14.1	MIT License	
22	frontend	dotenv	^16.4.5	BSD 2-Clause "Simplified" License	Copyright (c) 2015, Scott Motte
23	frontend	echarts	^5.5.0	Apache-2.0	
24	frontend	lodash	^4.17.21	MIT License	Copyright JS Foundation and other contributors <https://js.foundation/>
25	frontend	ngx-echarts	^17.2.0	MIT	
26	frontend	postgres	^3.4.4	The PostgreSQL Licence	https://opensource.org/license/postgresql
27	frontend	prisma	^5.13.0	Apache License 2.0	
28	frontend	reflect-metadata	^0.1.13	Apache License 2.0	
29	frontend	rxjs	~7.8.0	Apache License 2.0	
30	frontend	ts-enum-util	4.1.0	MIT License	
31	frontend	tslib	^2.3.0	BSD Zero Clause License	https://github.com/microsoft/tslib/blob/main/LICENSE.txt
32	frontend	zone.js	~0.14.3	MIT License	
33	devDependencies	angular-devkit/build-angular	~17.3.0	MIT License	
34	devDependencies	angular-devkit/core	~17.3.0	MIT License	
35	devDependencies	angular-devkit/schematics	~17.3.0	MIT License	
36	devDependencies	angular-eslint/eslint-plugin	~17.3.0	MIT License	
37	devDependencies	angular-eslint/eslint-plugin-template	~17.3.0	MIT License	
38	devDependencies	angular-eslint/template-parser	~17.3.0	MIT License	
39	devDependencies	angular/cli	~17.3.0	MIT License	
40	devDependencies	angular/compiler-cli	~17.3.0	MIT License	
41	devDependencies	angular/language-service	~17.3.0	MIT License	
42	devDependencies	commitlint/cli	^19.3.0	MIT	
43	devDependencies	commitlint/config-conventional	^19.2.2	MIT	
44	devDependencies	commitlint/config-nx-scopes	^19.3.0	MIT	
45	devDependencies	nestjs/schematics	^10.0.1	MIT-license	
46	devDependencies	nestjs/testing	^10.3.8	MIT-license	
47	devDependencies	nx/devkit	18.3.4	MIT-license	
48	devDependencies	nx/eslint	18.3.4	MIT-license	
49	devDependencies	nx/eslint-plugin	18.3.4	MIT-license	
50	devDependencies	nx/jest	18.3.4	MIT-license	
51	devDependencies	nx/js	18.3.4	MIT-license	
52	devDependencies	nx/nest	18.3.4	MIT-license	
53	devDependencies	nx/node	18.3.4	MIT-license	
54	devDependencies	nx/playwright	18.3.4	MIT-license	
55	devDependencies	nx/plugin	^18.3.4	MIT	
56	devDependencies	nx/web	18.3.4	MIT-license	
57	devDependencies	nx/webpack	18.3.4	MIT-license	
58	devDependencies	nx/workspace	18.3.4	MIT-license	
59	devDependencies	playwright/test	^1.43.1	Apache License 2.0	
60	devDependencies	schematics/angular	~17.3.6	MIT License	
61	devDependencies	stylistic/eslint-plugin-js	^2.0.0	MIT	
62	devDependencies	swc-node/register	~1.8.0	MIT License	
63	devDependencies	swc/core	~1.5.0	Apache License 2.0	
64	devDependencies	swc/helpers	~0.5.11	Apache License 2.0	
65	devDependencies	types/jest	^29.5.12	MIT license	
66	devDependencies	types/node	20.12.7	MIT license	
67	devDependencies	typescript-eslint/eslint-plugin	^7.7.1	MIT license	
68	devDependencies	typescript-eslint/parser	^7.7.1	TypeScript ESLint Parser	https://github.com/typescript-eslint/typescript-eslint/blob/main/LICENSE
69	devDependencies	eslint	8.56.0	MIT License	https://github.com/eslint/eslint/blob/main/LICENSE
70	devDependencies	eslint-config-stylelint	^21.0.0	MIT	
71	devDependencies	eslint-plugin-playwright	^1.6.0	MIT License	
72	devDependencies	eslint-config-prettier	9.1.0	MIT License	
73	devDependencies	eslint-plugin-simple-import-sort	^12.1.0	MIT	
74	devDependencies	eslint-plugin-unused-imports	^3.1.0	MIT	
75	devDependencies	husky	^9.0.11	MIT	
76	devDependencies	jest	^29.7.0	MIT License	
77	devDependencies	jest-environment-jsdom	^29.7.0	MIT License	
78	devDependencies	jest-environment-node	^29.7.0	MIT License	
79	devDependencies	jest-preset-angular	~14.0.3	MIT License	
80	devDependencies	lint-staged	^15.2.2	MIT	
81	devDependencies	nx	18.3.4	MIT License	
82	devDependencies	prettier	^3.2.5	MIT License	
83	devDependencies	prisma	^5.13.0	Apache-2.0	
84	devDependencies	stylelint	^16.4.0	MIT	
85	devDependencies	stylelint-config-recess-order	^5.0.1	ISC	https://en.wikipedia.org/wiki/ISC_license
86	devDependencies	stylelint-config-standard	^36.0.0	MIT	
87	devDependencies	stylelint-config-standard-scss	^13.1.0	MIT	
88	devDependencies	tailwindcss	^3.4.3	MIT	
89	devDependencies	ts-jest	^29.1.2	MIT License	
90	devDependencies	ts-node	10.9.2	MIT License	
91	devDependencies	typescript	~5.4.5	Apache License 2.0	
92	devDependencies	webpack-cli	^5.1.4	MIT License	
93	PackageManager	pnpm	8.15.1...	MIT	

Last Name	First Name	Value					
#REF!	#REF!			5.14	NOK		
#REF!	#REF!						
Lorenz	Alexander	3					
Rajjo	Lama						
Sternberg	Ingo	5		0	No size		
Betancourt Barrita	Cecilia			1	Trivial size		
Heisterberg	Jonas	8		2	Small size		
Krug	Maximilian	5		3	Medium size		
Nasir	Shahraz			5	Large size		
Schmidt	Patrick	5		8	Very large size		
Schmidt	David	5		13	Too large (size)		
Bojanić	Saša	5					
How to play planning poker							
1. Everyone type their number into their value field, don't hit return yet							
2. Someone, perhaps a product owner, count down 3.. 2.. 1..							
3. Then, everyone hit return to submit their value							