Project Name	Route planning App
Online team meeting	https://tu-berlin.zoom-x.de/j/63201367987?pwd=pwLP7eO90NCqkHyHE0cssfqZA0PwDl.1
Production system (if any)	
Test system (if any)	...
GitHub repository	https://github.com/amosproj/amos2025ss03-route-planning-app
GitHub feature board	https://github.com/orgs/amosproj/projects/81
GitHub imp-squared backlog	https://github.com/orgs/amosproj/projects/85
Team T-shirt (white)	https://www.shirtinator.de/loadBasket/PNnhXl2COPF
Team T-shirt (black)	...
Additional materials	...
Team maling list	oss-amos-proj3@lists.fau.de

Last Name	First Name	GitHub User Name	Email Address
Sandt	Eloi	eloinoel	eloi.sandt@campus.tu-berlin.de
Justus	Kleinau	jkleinau	j.kleinau@campus.tu-berlin.de
Harms	Finn	innif	
Karkani	Despoina	dkarkani	despoina.karkani@campus.tu-berlin.de
Karmokar	Subroto	SK-Subroto	subroto.karmokar@fau.de
Dutta	Arkadeep	**arkadeepberlin**	**arkadeep.dutta@campus.tu-berlin.de**
Uddin	Md Fahim	fahimu10	fahim.uddin@fau.de
Khanam	Faria Rahman	fariarahman15	faria.khanam@fau.de
Summerer	Felix	SUFelix	felix.summerer@fau.de

		Product Owner					
#	Meeting Day	Review	Planning	Software Developer	Release Manager	Scrum Master	Comment
1	2025-04-16		Despina	Everyone else	N/A	Eloi	
2	2025-04-23	Despina	Arkadeep	Everyone else	Finn	Eloi	
3	2025-04-30	Arkadeep	Despina	Everyone else	Justus	Eloi	
4	2025-05-07	Despina	Arkadeep	Everyone else	Faria	Eloi	
5	2025-05-14	Arkadeep	Despina	Everyone else	Subroto	Eloi	
6	2025-05-21	Despina	Arkadeep		Felix	Eloi	
7	2025-05-28	Arkadeep	Despina		Fahim	Eloi	Mid-term due
8	2025-06-04	Despina	Arkadeep		Subroto	Eloi	
9	2025-06-11	Arkadeep	Despina		Faria	Eloi	
10	2025-06-11	Despina	Arkadeep		Justus	Eloi	
11	2025-06-18	Arkadeep	Despina		Felix	Eloi	
12	2025-06-25	Despina	Arkadeep		Fahim	Eloi	
13	2025-07-02	Arkadeep	Despina		Subroto	Eloi	
14	2025-07-09	Despina	Arkadeep		Faria	Eloi	Demo day!
15	2025-07-16	Arkadeep				Eloi	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	Have an enjoyable experience in working together
	learn from each other
	Our teams tries to create tangible progress with the application every week
	Complete ECTS and get good grade
	Provide a usefull solution to meisterwerk
Meeting norms	All team-meetings are mandatory unless a valid reason is communicated in advance
	be on time to value each other's time
	Ask questions if your are not clear on something.
	The meeting starts on time, even if not all members are present yet
	Camera On :)
Working norms	Team members should be open to feedback and view it as a growth opportunity
	We use feature, development and fix-branches in git
	When we face problems or fail to reach our goals, we should communicate that as soon as possible
	Always help each other!
	agree on who does what and keep track of it
	Follow coding conventions
	Openly communicate working capacity, so work can be attributed accordingly
	Keep each other updated about progress
Coordination norms	The Product owners for Review and Planning respectively switch every week
	The PO's (at least, devs are always welcome to join) meet with the industry partner at regular intervals and communicate feature requests to the rest of the team
	When facing decisions that cant be resolved in a discussion, we put it to vote
	keep essential information and links to tools centralised in one place (seperate discord channel)
Communication norms	Use the communication tools that we have. In our case a Discord Server
	We should answer within one day
	There are no stupid questions!
	Communicate respectfully and let everybody speak out
	Respect each others opinions, we are all here to learn :)
	constructive criticism, talk about solutions
	communicate illnesses and expected work downtime and what it means for others
	Smaller problems with persons in thew group should be voiced in meetings, larger concerns should be discussed in a one-on-one call first
Consideration norms	Each task will have a primary owner(s) and a reviewer to ensure accountability and quality.
	Talk about the problem, talk with the roles, get the context. Communication is very important to understand everything
	prefer to speak in public over speaking privately?
Cont. improvement norms	Keep 15 minutes atleast for Sprint retrospective

	Give constructive feedback on Pull Requests
Rewards	
	One evening/afternoon/whatever of fun online games
Sanctions	
	10 minutes of mandatory guided meditation… with a goat yoga video soundtrack
	A silly video-filter has to be activated for the next meeting
Signatures	
Scrum Master	Eloi Sandt
Product owner	Arkadeep Dutta
Product owner	Despina Karkani
Software developer	
Software developer	Subroto Karmokar
Software developer	Faria Rahman Khanam
Software developer	Md Fahim Uddin
Software developer	Finn Harms
Software developer	Justus Kleinau
Software developer	Felix Summerer

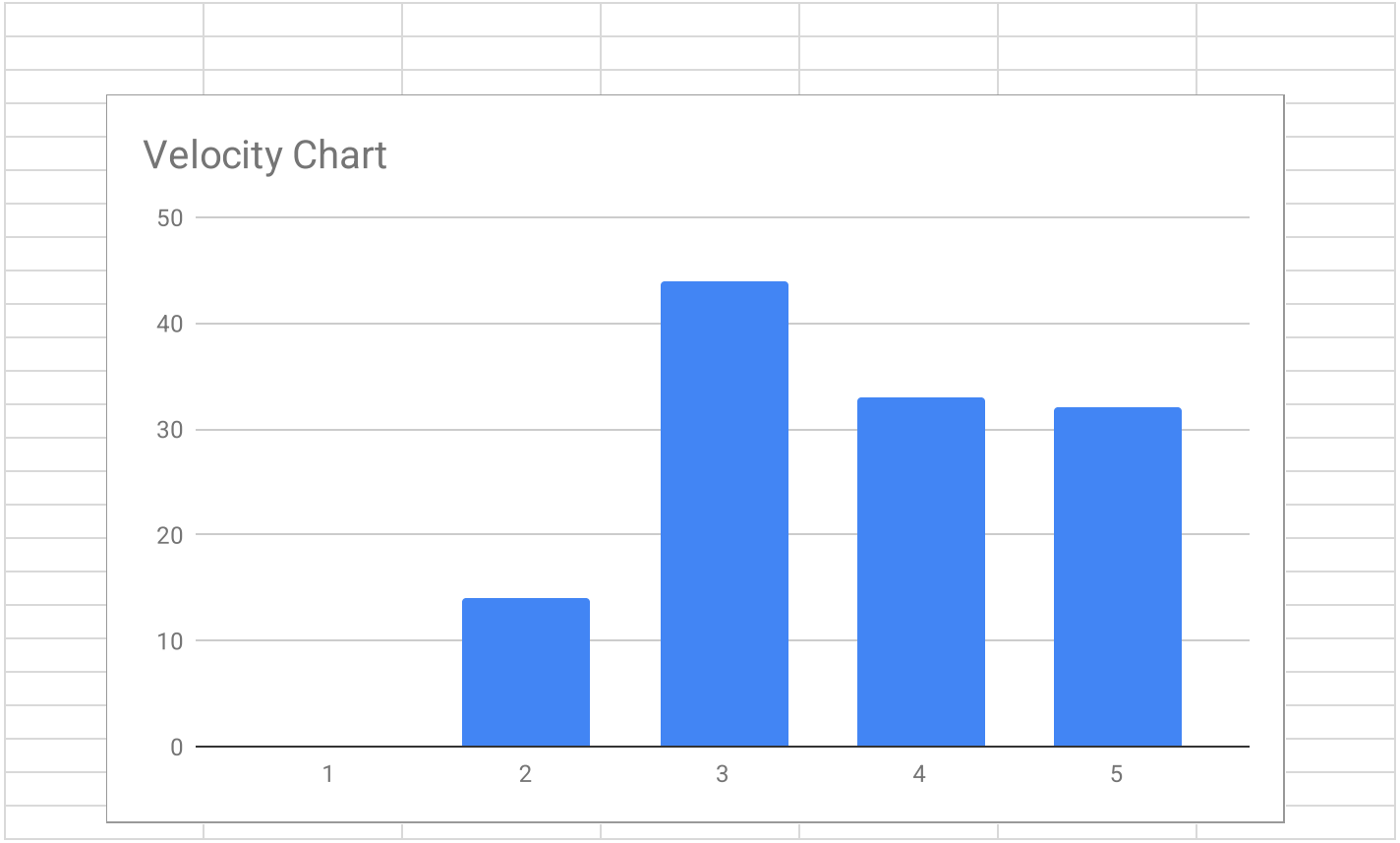
Product Vision	Project Mission
To build an intelligent, automated route optimization tool that empowers organizations to efficiently assign appointments to field workers—optimizing for time, cost, and resource usage—while maximizing overall operational efficiency and service quality	The mission of this project is to create an MVP for the AMOS Route Optimizer. Core functionality will include uploading appointment data, generating an optimized route plan based on efficiency criteria (e.g., time, cost, resource usage), and displaying the resulting plan on an interactive map interface.

Term	Definition

Sprint #	Sprint goal
1	None
2	None
3	None
4	None
5	Finalize the optimization trigger flow, integrate distance calculations, and implement preprocessing to detect potential solver issues before execution.
6	Connect backend route data to the frontend, visualize the full daily and worker plans, and introduce servicetime customization in both backend and UI.
7	
8	
9	
10	
11	
12	
13	
14	
15	

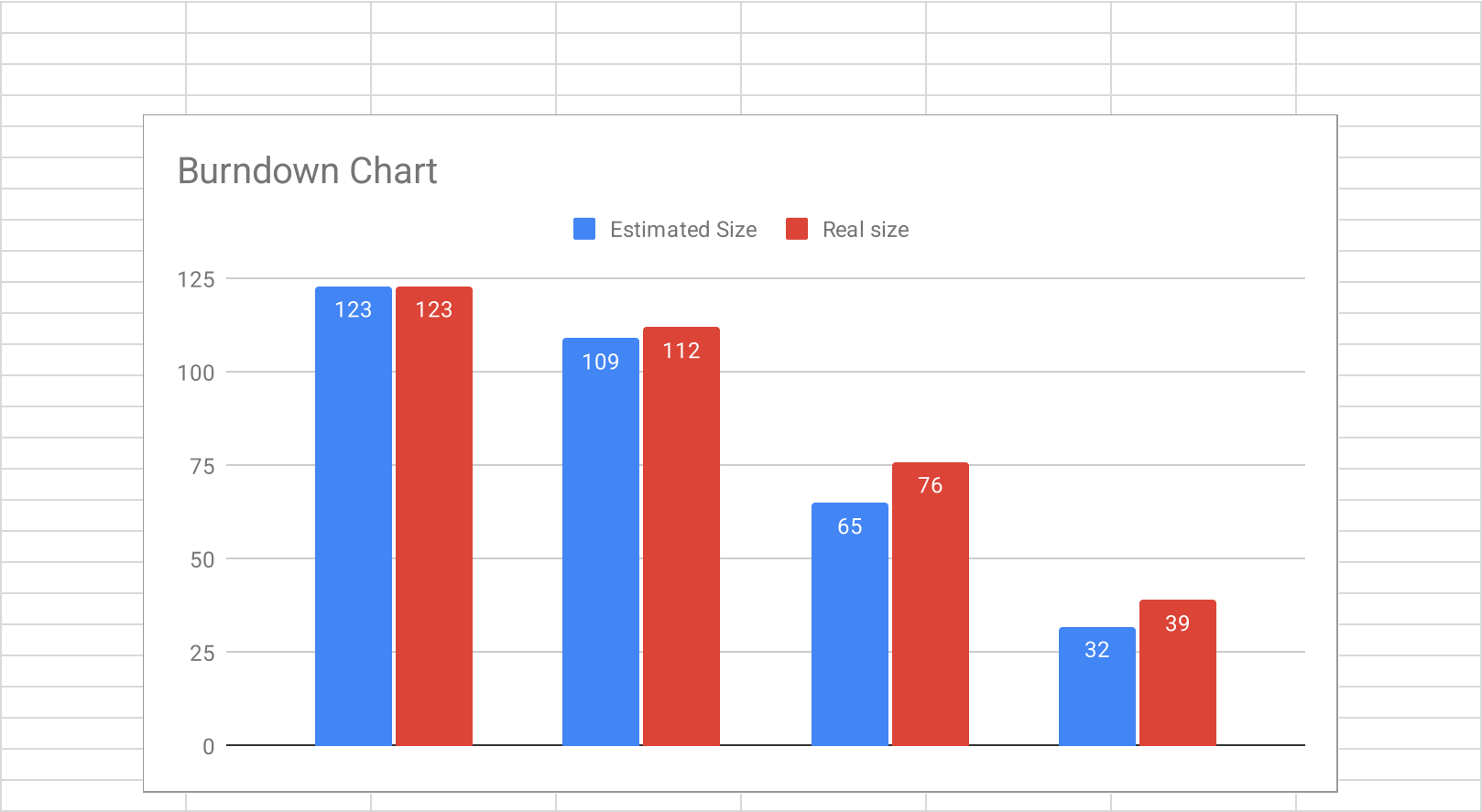
feature Specific	Sprint Specific	Project release
Frontend and backend components for the feature are implemented and integrated Feature is reviewed via code review (e.g., pull request approved) Feature is merged into the development branch without breaking other functionalities The feature is demo-ready and accepted by the Product Owner	The app builds and runs successfully in a clean environment All sprint backlog items marked as "Done" meet the Feature DoD Demo conducted with functioning features shown via "show and tell" in Sprint Planning session by the developers Release notes are created and shared (e.g., summary in planning document or GitHub) Velocity is tracked and recorded	All core MVP functionality is implemented and passes acceptance tests Optimized route output is visualized on the map using real user inputs Documentation is complete (README.md, user guide, build instructions) System passes testing across common browsers/environments Mid- and final release plans are fulfilled Build process video and required deliverables are uploaded

Sprint #	Story Points Realized
1	
2	14
3	44
4	33
5	32
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
	PLEASE CREATE THE VELOCITY CHART ON A NEW TAB USING THE DATA FROM THIS TAB


Velocity Chart
50
40
30
20
10
0
1
2
3
4
5

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
Release						
Total			123	123		
Sprints						
1	Basic formal documentation requirements		0	123	0	123
2	Set up backend, frontend, and upload infrastructure to support the route planning flow.		14	123	11	123
3	Complete the foundation for file upload, schema validation, and routing structure on both backend and frontend to prepare for full route processing.		44	109	36	112
4	Integrate and run the initial optimization plan, connect validated input data, and prepare the frontend for dynamic interaction with editable and visual route elements.		33	65	37	76
5	Finalize the optimization trigger flow, integrate distance calculations, and implement preprocessing to detect potential solver issues before execution.		32	32	0	39
6	Connect backend route data to the frontend, visualize the full daily and worker plans, and introduce servicetime customization in both backend and UI.		0			
Features						
1	Basic formal documentation requirements	None				
2	Set up backend, frontend, and upload infrastructure to support the route planning flow.	Set up Backend	2		3	
		Save & Load Scenarios from JSON	5		3	
		Set Up CI/CD Pipeline	5		3	
		Set up Front End	2		2	
3	Complete the foundation for file upload, schema validation, and routing structure on both backend and frontend to prepare for full route processing.	Agreement on basic structure – Frontend Backend	8		8	
		Validate Addresses via Google Maps Geocoding API	8		5	
		Implement Start and Finish Address Fields (Business data)	1		1	
		Add Worker Count Slider Component	1		1	
		Create “Start Optimization” Button	1		1	
		Create a routing structure on the frontend	3		3	
		Build File Upload Component	3			
		Docker File Creation	3		3	
		Excel File Upload Functionality	3		3	
		Convert Uploaded Excel File to JSON	5		3	
		JSON Schema Validation for Uploaded File	3		3	
		Create distance matrix from the uploaded address	5		5	
4	Integrate and run the initial optimization plan, connect validated input data, and prepare the frontend for dynamic interaction with editable and visual route elements.	Migrate Frontend Components to ShadcnUI and TailwindCSS	2		3	
		Map Business Data Sheet and Appointment Data to the UI	3		5	
		Move Editable Inputs and Start Button to Map View	3		3	
		Implement Basic Optimization Plan: First Draft	8		8	
		Connecting Worker Address Validation and Distance Matrix	3		3	
		Implement Basic Map View with Location Points Only	8		8	
		Add Calendar View with Daily Appointment Overview	3		5	
		Explore Options for Handling Faulty Addresses from Uploaded Files in the UI	3		2	

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
5	Finalize the optimization trigger flow, integrate distance calculations, and implement preprocessing to detect potential solver issues before execution.	Implement Preprocessing Logic to Detect Potential Solver Failures: First Draft	5			
		Implement Distance Calculation in the Route Optimization	3		0	
		Project Optimized Routes on the Map View	8		0	
		Prepare Application for Deployment in Industry Partner's AWS Environment (Backend)	8		0	
		Allow User to Uncheck Faulty Addresses and Proceed with Optimization	5		0	
		Connect "Start Optimization" Button to Backend Optimization Trigger	3		0	
6	Connect backend route data to the frontend, visualize the full daily and worker plans, and introduce servicetime customization in both backend and UI.	Implement Frontend Visualization for Errors Raised in Preprocessing	0		0	
		Display Daily Plan as a List/Table with Export Option	0		0	
		Display and Export Individual Worker Plans	0		0	
		Integrate Solver API Output to Frontend for Route Visualization	0		0	
		Allow User to Configure Service Time per Appointment (With Default Fallback)	0		0	
		Create Space and Distinguish between Service Time and Appointment Time in Backen	0		0	
		PLEASE CREATE THE BURNDOWN CHART ON A NEW TAB USING THE DATA FROM THIS TAB				


Burndown Chart
Estimated Size
Real size
125
100
75
50
25
0
123
123
109
112
65
76
32
39

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
Release						
Total			0	0		
Sprints						
1			0	0	0	0
2			0	0	0	0
3			0	0	0	0
...				0		0
Features						
1						
2						
3						
		PLEASE CREATE THE BURNDOWN CHART ON A NEW TAB USING THE DATA FROM THIS TAB				

#	Feature Definition of Done	Sprint Release Definition of Done	Project Release Definition of Done

Type	Link / reference

#	Context	Name	Version	License	Comment
	Programming Language for Backend	Python	3.11	Python Software Foundation (PSF) License	
	API Framework for Backend API	FastAPI	0.103.1	MIT License	
	Route Optimization in Backend	Google OR	9.7	Apache 2.0	
	Get Distance-Matrix between points for routing	Google Maps API	N/A	Commercial	
	Programming Language for Frontend	Typescript	5.7.2	Apache 2.0	
	Framework for Frontend	ReactJS	19.01	MIT License	
	UI-Components for Frontend	Shadcm UI	2.5.0	MIT License	
	CSS Framwork for Frontend	TailwindCSS	4	MIT License	

Last Name	First Name	Value					
Last Name	**#REF!**			#DIV/0!	#DIV/0!		
#REF!	**#REF!**						
Justus	**Kleinau**						
#REF!	**#REF!**						
Harms	**Finn**			0	No size		
Karkani	**Despoina**			1	Trivial size		
Karmokar	**Subroto**			2	Small size		
Uddin	**Md Fahim**			3	Medium size		
Khanam	**Faria Rahman**			5	Large size		
Summerer	**Felix**			8	Very large size		
				13	Too large (size)		
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							