Project Name	...
Online team meeting	https://tu-berlin.zoom-x.de/j/64327398600?pwd=iKF2e9D7PEM8Q552cgabakS0O6L9xa.1
Production system (if any)	http://amos.b-iq.net/
Test system (if any)	...
GitHub repository	https://github.com/amosproj/amos2025ss02-building-documentation-management-system
GitHub feature board	https://github.com/orgs/amosproj/projects/80/views/2
GitHub imp-squared backlog	https://github.com/orgs/amosproj/projects/84
Team T-shirt (white)	https://www.shirtinator.de/loadBasket/vNV5XYUVMtq
Team T-shirt (black)	https://www.shirtinator.de/loadBasket/nQ9JYKa8F3W
Additional materials	...
Team maling list	oss-amos-proj2@lists.fau.de

Last Name	First Name	GitHub User Name	Email Address
Kurtz	Daniel	daku-de	daniel.kurtz@fau.de
Deli	Deniz	ddeli	d.deli@campus.tu-berlin.de
Egge	Carl	carl-egge	c.egge@campus.tu-berlin.de
Halder	Arpita	arpita739	arpita.halder@fau.de
Kazi	Masudur Rahaman	kamsur	masudur.r.kazi@fau.de
~~Almalla~~	~~Nirmin~~	~~Nirmin96~~	~~nirmin.almalla@campus.tu-berlin.de~~
Ouabo	Samuel	so77nava	samuel.ouabo@fau.de
Billah	Masum	masum-abid	masum.billah@fau.de
Hussain	Syed Affan	syedaffan10	affan.hussain@fau.de
Djamen	Bastiane	bastydjamen	eugenie.djamen@fau.de

		Product Owner					
#	Meeting Day	Review	Planning	Software Developer	Release Manager	Scrum Master	Comment
1	2025-04-16		Daniel	Everyone else	N/A	Deniz	
2	2025-04-23	Daniel	Carl	Everyone else	Arpita	Deniz	
3	2025-04-30	Carl	Daniel	Everyone else	Masudur Rahaman	Deniz	
4	2025-05-07	Daniel	Carl	Everyone else	Samuel	Deniz	
5	2025-05-14	Carl	Daniel	Everyone else	Masum	Deniz	
6	2025-05-21	Daniel	Carl	Everyone else	Syed Affan	Deniz	
7	2025-05-28	Carl	Daniel	Everyone else	Bastiane	Deniz	Mid-term due
8	2025-06-04	Daniel	Carl	Everyone else	Arpita	Deniz	
9	2025-06-11	Carl	Daniel	Everyone else	Masudur Rahaman	Deniz	
10	2025-06-18	Daniel	Carl	Everyone else	Samuel	Deniz	
11	2025-06-25	Carl	Daniel	Everyone else	Masum	Deniz	
12	2025-07-02	Daniel	Carl	Everyone else	Syed Affan	Deniz	
13	2025-07-09	Carl	Daniel	Everyone else	Bastiane	Deniz	
14	2025-07-16	Daniel	Carl	Everyone else	Arpita	Deniz	Demo day!
15	2025-07-23	Carl	Daniel	Everyone else	Masudur Rahaman	Deniz	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	Deliver a product that meets the stakeholder requirements.
	Everybody in the team wants to gain some new knowledge.
	Every team member should have fun during the project.
	Coming up with innovative solutions using modern technology.
	Develop our skills in teamwork and communication.
Meeting norms	We meet every Wednesday in the team meeting.
	POs will have weekly meetings with the stakeholder which are optional for the SDs.
	Everybody should be on time for the team meeting. If you are late or unavailable please inform the group beforehand through the defined communication channels.
	We practice a respectful and non-disciminative communication with all team members.
	Everyone should do their best to participate with Camera and Microphone turned on in the team meetings.
	Meetings will follow a clear structure that is quickly presented at the beginning of the meeting.
Working norms	We communicate or problems with the group and help colleagues that are stuck.
	We will follow a common Git Flow that will be defined by the SDs and follows industry best-practices.
	Commit often and use clear commit messages to avoid unnecessary conflicts.
	We will always use pull requests and code reviews instead of force pushes to improve coding quality.
Coordination norms	The POs are responsible for allocating assignments.
	The POs will alternatingly moderate the team meetings. The SM is responsible for keeping the team meetings on track.
Communication norms	We will use Discord as technical channel for programming topics.
	The Email list and Stand Up Email will be for organizational communication and announcements.
	Everybody should check these channels daily.
	We use english as a working language.
	The WhatsApp group is an additional optional channel for group conversations.
Consideration norms	No side conversation and technical conversations during the team meeting.
	For team decisions and disagreements an anonymous vote will be held and the majority decides.
Cont. improvement norms	Stand Up Emails and Hapiness Index will be used to track team work.
	No pull requests should be accepted without feedback from at least one other team member (Positive feedback is welcome).
	We practice open feedback culture and continously improve our working environment.
Rewards	By delivering on time everyone can reward themself with a free weekend.
Sanctions	We define no sanctions.

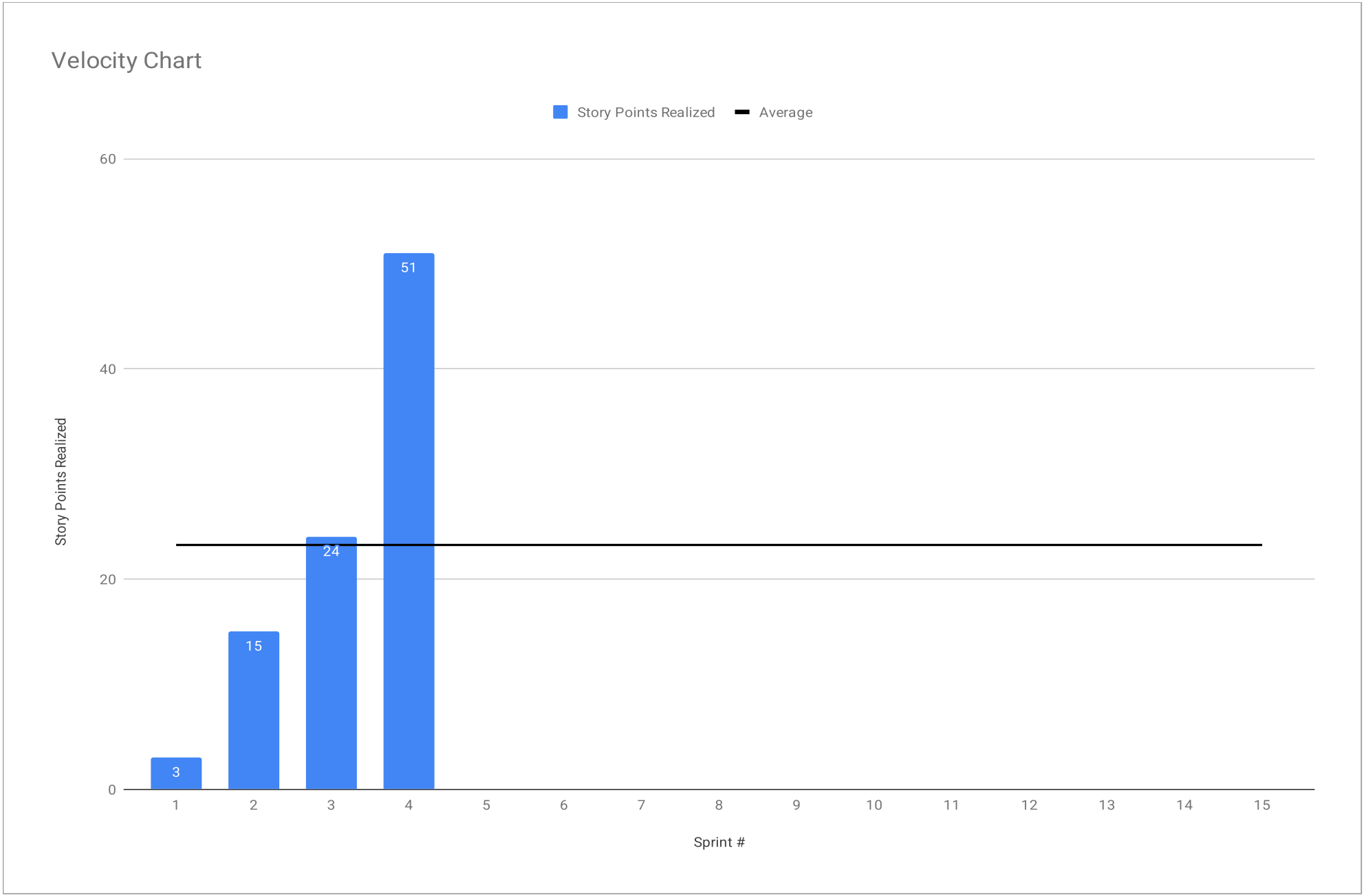
Signatures	
Scrum Master	Deniz Deli
Product owner	Daniel Kurtz
Product owner	Carl Egge
Software developer	Masudur Rahaman Kazi
Software developer	Bastiane Djamen
Software developer	Nirmin Almalla
Software developer	Arpita Halder
Software developer	Syed Affan Hussain
Software developer	Masum Billah
Software developer	Samuel Ouabo

Product Vision	Project Mission
Managing building portfolios often means navigating scattered, inconsistent document repositories. Our system enables facility managers to easily upload files, automatically extract and validate critical metadata, classify documents by type and building, and find exactly what they need through intuitive search and natural-language queries — reducing search times from hours to seconds. By integrating into broader building document management platforms, our solution enhances existing workflows and ensures seamless access to critical information across systems.	We aim to build a secure, multi-tenant backend that stores uploaded permits, certificates, and maintenance reports; applies AI-driven OCR and metadata extraction to classify and validate each document; and provides a web UI where users can query and filter their building records using plain language. By implementing and demonstrating a full workflow — from document upload to automatic processing and natural-language search on sample data — we will create the technical foundation for a production-ready system that significantly reduces the time and effort required to manage building documents.

Term	Definition

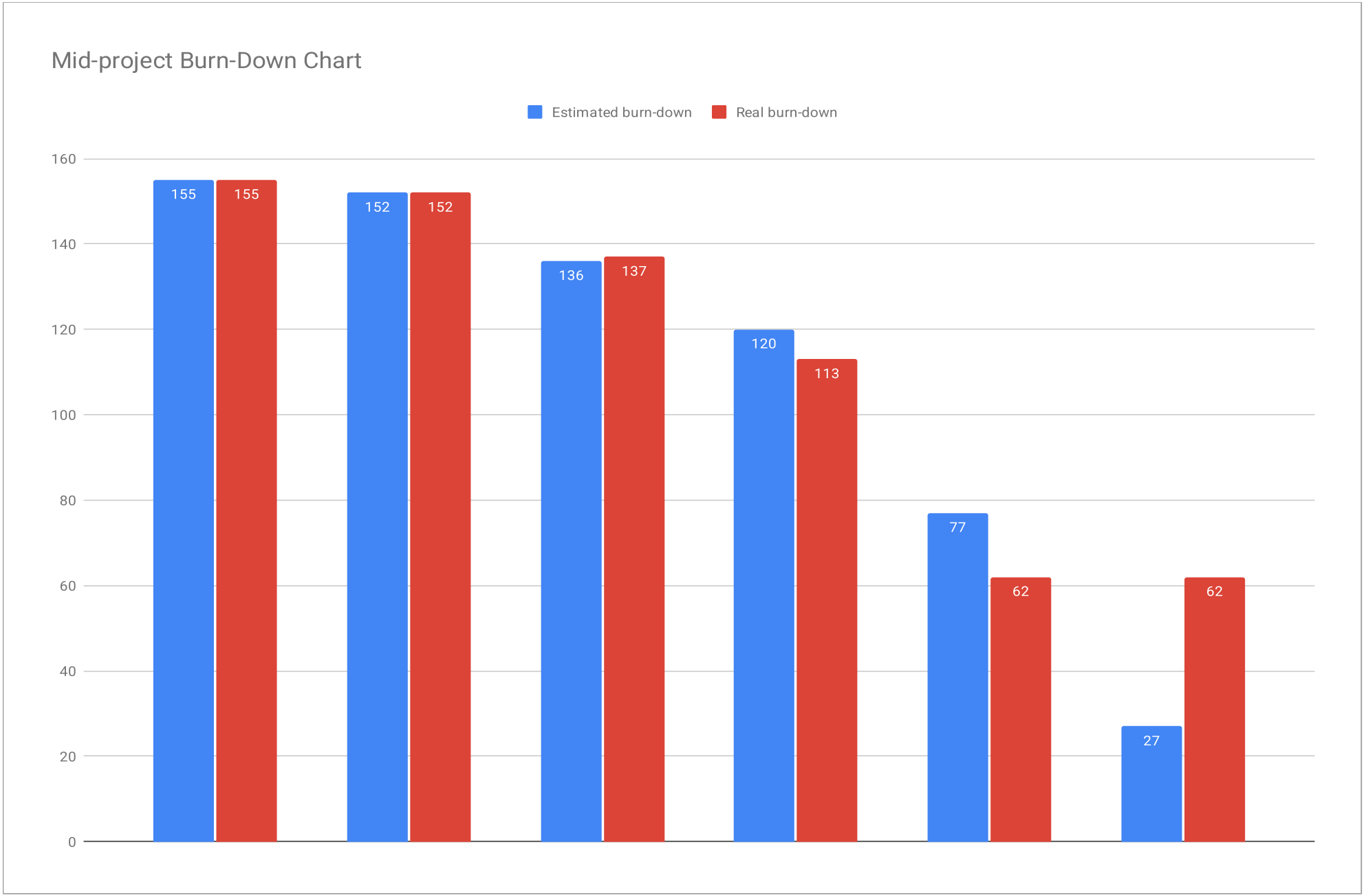
Sprint #	Sprint goal
1	Get to know the team and understand the stakeholder's requirements
2	Start scaffolding basic backend and frontend infrastructure and research domain-specific knowledge
3	Set up unified development and production environment through containerization and tooling
4	Fix the issue of sensitive documents in the repository and enhance backend with endpoints and database connection
5	Extend and prepare components so they can be easily connected
6	Prepare mid-project release by binding frontend and backend together and finalizing container orchestration
7	
8	
9	
10	
11	
12	
13	
14	
15	

Sprint #	Story Points Realized	Average
1	3	23.25
2	15	23.25
3	24	23.25
4	51	23.25
5		23.25
6		23.25
7		23.25
8		23.25
9		23.25
10		23.25
11		23.25
12		23.25
13		23.25
14		23.25
15		23.25


Velocity Chart
Story Points Realized
Average
60
40
20
0
Story Points Realized
3
15
24
51
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Sprint #

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
Release						
Total			155	155	93	93
Sprints						
1	**Get to know the team and understand the stakeholder's requirements**		3	155	3	155
2	**Start scaffolding basic backend and frontend infrastructure and research domain-specific knowledge**		16	152	15	152
3	**Set up unified development and production environment through containerization and tooling**		16	136	24	137
4	**Fix the issue of sensitive documents in the repository and enhance backend with endpoints and database connection**		43	120	51	113
5	**Extend and prepare components so they can be easily connected**		50	77	0	62
6	**Prepare mid-project release by binding frontend and backend together and finalizing container orchestration**		27	27	0	62
Features						
1	**Get to know the team and understand the stakeholder's requirements**					
		Align on Development Guidelines	1		1	
		Meet the industry partner	1		1	
		Create a team logo	1		1	
2	**Start scaffolding basic backend and frontend infrastructure and research domain-specific knowledge**					
		Create a shirt design	1		1	
		Initialize SW Bill of Materials	2		2	
		Create SW Architecture Description	3		3	
		Scaffold Basic C# Web API	3		2	
		Scaffold Basic Angular Application	3		3	
		Research Local RAG-Based LLM Solutions	2		2	
		Research typical Building Documents	2		2	
3	**Set up unified development and production environment through containerization and tooling**					
		Dockerize Front-End	2		3	
		Dockerize Back-End	2		2	
		Create DB Scheme Diagram	2		2	
		Create README	1		1	
		Consolidate Project Structure	2		2	
		Create Docker-Compose File	2		3	
		Linting and Formatting Frontend	1		1	
		Linting and Formatting Backend	1		2	
		Set Up GitHub Actions Pipeline	3		8	
4	**Fix the issue of sensitive documents in the repository and enhance backend with endpoints and database connection**					
		Fix: Remove Sensitive Building Documents	5		5	
		Implement Backend CRUD for Document	8		8	
		Implement Backend CRUD for Building	5		8	
		Integrate Linting into Pipeline	5		8	
		Add Authentication to Frontend	3		5	
		Design Frontend Mockup	5		8	
		Set Up Ollama	3		3	
		Implement Basic DB models in C#	8		5	
		Reorganize DB Scheme	1		1	
5	**Extend and prepare components so they can be easily connected**					
		Document Build Process in Video Format	2			

Sprint	Goal	Feature Name	Est. size	Est. remaining	Real size	Real remaining
		Update Bill of Materials	1			
		Support OpenAPI Client Generation	8			
		Persist Data Using Docker Volumes	8			
		Enhance Angular Frontend based on Mock Up	5			
		Persist Building Data in DB	5			
		Query Ollama via Backend	5			
		Agree on DoD	3			
		Set Up Apache Tika	5			
		Add Organization Entity to DB	8			
6	**Prepare mid-project release by binding frontend and backend together and finalizing container orchestration**					
		Replace Default Web Services with Angular Frontend	1			
		Upload and Store Document via Frontend	3			
		Document View Page displays Actual Documents	5			
		Display Stored Documents in the Document Explorer	5			
		Connect Frontend to LLM	5			
		Apache Tika Integration during Document Upload	5			
		Enforce HTTPS on Production Server	3			


Mid-project Burn-Down Chart
Estimated burn-down
Real burn-down
160
140
120
100
80
60
40
20
0
155
155
152
152
136
137
120
113
77
62
27
62

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
1	Documentation is updated to include the new feature	Atleast one error-free functionality (which has passed feature DoD), more than previous release.	Stakeholders have reviewed and signed off the release
2	Another SD, not the implementer, has reviewed the feature using the documentation.	Entire project builds and deploys to production.	Documentation (user + technical) is complete and comprehensive
3	GitHub actions workflows for linting pass	Non-UI logic (e.g., DB transactions) verified by an SD or RM.	No critical open issues remain in backlog
4	Feature branch has been merged and deleted	Release manager writes release notes (& Change log)	
5		All GitHub actions workflows pass	
6		Bill of Materials is up to date	

Type	Link / reference

#	Context	Name	Version	License	Comment
1	UI	Angular	19	MIT License	Needed for developing UI as per client requirement
2	UI	Node.js	22.15.0	Custom (open-source)	Needed for Angular
3	UI	npm	10.9.2	Artistic License 2.0 with additions	Package manager, needed for Angular
4	Backend (Framework)	Microsoft.NETCore.App	Implicit	MIT License	.NET Core base runtime (System., Microsoft.)
5	Backend (Framework)	Microsoft.ASPNetCore.App	Implicit	MIT License	ASP.NET Core web framework (Controllers, Routing, Authorization)
6	Backend (Library)	Swashbuckle.AspNetCore	6.5.0	MIT License	Provides Swagger/OpenAPI documentation and UI
7	Backend (Library)	AspNetCore.HealthChecks.UI	6.0.4	Apache 2.0 License	Adds health check endpoints (/healthz) and optional health UI
8	Backend (Framework)	Entity Framework Core .NET Command-Line Tools	9.0.4	Apache 2.0 License	Used to manage EF Core database tasks from the terminal

Last Name	First Name	Value					
Halder	**Arpita**	1		**1.00**	**OK**		
Kazi	**Masudur Rahaman**	1					
Ouabo	**Samuel**	1					
Billah	**Masum**	1					
Hussain	**Syed Affan**	1		0	No size		
Djamen	**Bastiane**	1		1	Trivial size		
				2	Small size		
				3	Medium size		
				5	Large size		
				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							