



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

GLITCHHub
TEAM

Verbale esterno 24/02/2026 (M31)

Ordine del giorno

1. Resoconto incontri con i professori Cardin e Vardanega
2. Pianificazione per i prossimi sprint

•
Versione **1.0.0**

Stato	Verificato
Partecipanti	Alessandro Dinato Riccardo Graziani Elia Ernesto Stellin Jaume Bernardi Michele Dioli Silvia Salvalaio
Distribuzione	GlitchHub Team M31 SRL Prof. Cardin Riccardo Prof. Vardanega Tullio

Indice

1. Introduzione	2
2. Resoconto	2
2.1. Resoconto incontri con i professori Cardin e Vardanega	2
2.2. Pianificazione per i prossimi sprint	2
2.2.1. Progettazione	2
2.2.2. Dev containers	2
2.2.3. Gestione dei secrets	2
2.2.4. Gestione repository	2
2.2.5. Priorità	3
3. Attività conseguenti	3

1. Introduzione

Il presente verbale attesta che in data 24 febbraio 2026 dalle 14:30 alle 15:20, si è svolto l'incontro con la proponente **M31 SRL**, in modalità remota.

Durante l'incontro il gruppo ha riportato l'esito degli incontri RTB con i professori, successivamente sono stati dati consigli pratici dalla proponente su come avanzare.

2. Resoconto

2.1. Resoconto incontri con i professori Cardin e Vardanega

Il gruppo ha riportato l'esito positivo (semaforo verde) dell'incontro con il professor Cardin, e ha riportato i dubbi espressi dal professor Vardanega riguardanti soprattutto la mancanza di ore di orologio per poter garantire la data di consegna del MVP prevista per il 27 marzo 2026.

Ci è stato chiesto se la data di consegna fosse ufficiale sia per i professori sia per l'azienda, ed è stato ragionevole pensare di stabilire che la consegna per la proponente sia alcuni giorni in anticipo.

2.2. Pianificazione per i prossimi sprint

2.2.1. Progettazione

Partire dalla logica ad alto livello (modelli C4, diagrammi delle classi), basandosi sul PoC e sui suggerimenti di M31, per poi scendere nel dettaglio dei singoli componenti. È importante organizzarsi in modo efficace, dato che il gruppo è numeroso. Le task devono essere assegnate in maniera ordinata e ciascun membro deve avere un ruolo ben definito. L'approccio consiste nel passare dall'architettura ad alto livello alla definizione dei singoli componenti, per poi ampliarli progressivamente. La fase di progettazione è bloccante e svolgerla in sette persone può risultare caotico. Per parallelizzare il lavoro, non tutti si dedicheranno alla progettazione: alcuni potranno occuparsi del setup e dello studio degli strumenti mancanti. È utile anche portarsi avanti con la documentazione, sia aggiornando i documenti esistenti sia producendo quelli nuovi (specifica tecnica, manuale utente).

2.2.2. Dev containers

C'è stato consigliato di sviluppare un ambiente di sviluppo in container con *hot reload* dalla cartella locale tramite *docker compose*. Utile per sincronizzare versioni e ambienti, da svolgere in parallelo alla progettazione.

2.2.3. Gestione dei secrets

Per l'MVP, utilizzare *docker secrets* iniettati tramite *compose*. I secrets non vanno pubblicati su GitHub. Si è pensato ad una possibile condivisione interna tramite Bitwarden, con file *.env* locali e script di setup automatico delle credenziali.

2.2.4. Gestione repository

- Adottare un approccio *multi-repository* con *Dockerfile* separati per servizio, evitando un'unica immagine troppo pesante.
- *Repository parent* con *git submodule* per orchestrare il tutto.
- Piccoli *compose* separati per consentire lo sviluppo indipendente dei singoli servizi.

Inoltre, è stato consigliato di modificare la gestione delle *repository*, anziché utilizzare un'unica repository per tutti i microservizi, creare una repository per ciascun microservizio. Il consiglio nasce

dall'esperienza della proponente, che ha trovato il lavoro in *multi-repository* più comodo per il singolo sviluppatore.

2.2.5. Priorità

Concentrarsi prima sui requisiti obbligatori, aggiungendo quelli desiderabili/opzionali in modo incrementale. Focus sulla parte core (comunicazione e scambio dati). Per la dashboard usare strumenti predefiniti (es. Angular Material).

3. Attività conseguenti

Task	Assegnatari	Issue
Riflettere modifiche a workflow issue, branch e PR	Riccardo Graziani	<u>Docs/#328</u>
Dividere servizio data consumer per funzionalità	Alessandro Dinato	<u>Docs/#329</u>
Dividere servizio gateway per funzionalità	Elia Ernesto Stellin	<u>Docs/#330</u>
Dividere servizio dashboard frontend per funzionalità	Jaume Bernardi	<u>Docs/#331</u>
Dividere servizio dashboard backend per funzionalità	Michele Dioli	<u>Docs/#332</u>

Jaume Bernardi



Firma del revisore interno

Cristian Pirlog



Firma del revisore esterno