



>Three Way Milkshake_

Verbale esterno 7 9 Aprile 2021

Three Way Milkshake - Progetto "PORTACS"

threewaymilkshake@gmail.com

Versione	1.0.0
Stato	Approvato
Uso	Esterno
Approvazione	Zuccolo Giada
Redazione	De Renzis Simone
Verifica	Greggio Nicolò
Destinatari	Three Way Milkshake Prof. Vardanega Tullio Prof. Cardin Riccardo

Descrizione

Verbale del meeting del 2021-04-09 del gruppo Three Way Milkshake con il
proponente Sanmarco Informatica.



Registro delle modifiche

Vers.	Descrizione	Data appr.	Approvazione
1.0.0	Approvazione del verbale	2021-04-15	Zuccolo Giada

Vers.	Descrizione	Redazione	Data red.	Verifica	Data ver.
0.1.0	Stesura del verbale	De Renzis Simone	2021-04-15	Greggio Nicolò	2021-04-15



Indice

1	Informazioni generali	3
1.1	Dettagli sull'incontro	3
1.2	Ordine del giorno	3
2	Verbale della riunione	4
2.1	Esposizione dell'architettura del software	4
2.2	Breve dimostrazione dell'eseguibile fin'ora prodotto	4
2.3	Chiarimenti su Docker	4
2.4	Modalità di parcheggio dei muletti	5
2.5	Proposta di file <i>json</i> per la persistenza	5
2.6	Chiusura del server	5
3	Tracciamento delle decisioni	6



1 Informazioni generali

1.1 Dettagli sull'incontro

- **Luogo:** Incontro telematico tramite piattaforma Google Meet;
- **Data:** 2021-04-09;
- **Ora di inizio:** 14:30;
- **Ora di fine:** 16:15;
- **Partecipanti interni:** (6/6)
 - Chiarello Sofia;
 - Crivellari Alberto;
 - De Renzis Simone;
 - Greggio Nicolò;
 - Tessari Andrea;
 - Zuccolo Giada.
- **Partecipanti esterni:** (2)
 - Beggiato Alex (Sanmarco Informatica);
 - Piva Alessandra (Sanmarco Informatica).

1.2 Ordine del giorno

La riunione prevede la discussione con il proponente dei seguenti punti:

- esposizione dell'architettura del software;
- breve dimostrazione dell'eseguibile fin'ora prodotto;
- chiarimenti su Docker;
- modalità di parcheggio dei muletti;
- chiusura del server.

2 Verbale della riunione

2.1 Esposizione dell'architettura del software

Il meeting è iniziato con l'esposizione dell'architettura del software modellata dal gruppo nelle ultime settimane. Sono stati quindi mostrati i diagrammi UML relativi alle componenti **client**: l'esposizione è stata motivo di conferma sulla bontà dell'applicazione del pattern architetturale *MVC* nel contesto dei client. È stato spiegato il funzionamento di ognuno dei moduli.

Il proponente ha inoltre chiarito come sia preferibile predisporre un modulo unico (lato client) dedicato al login, piuttosto che averne due separati (uno per l'Amministratore e uno per il Responsabile).

Si è proceduto poi a mostrare il diagramma UML del **server** e a spiegarne velocemente il funzionamento. La discussione si è focalizzata sui design pattern applicati (soprattutto Singleton): il proponente condivide le scelte fatte, e chiede dei chiarimenti in merito al modulo di gestione delle *Task*. L'esposizione non può raggiungere una profondità troppo elevata, anche a causa dei limiti temporali della riunione.

2.2 Breve dimostrazione dell'eseguibile fin'ora prodotto

Approfitando della presenza concordata di Alessandra Piva, Responsabile della logistica presso l'azienda proponente Sanmarco Informatica, il gruppo ha predisposto una breve dimostrazione dell'eseguibile fin'ora realizzato. Sono stati mostrati il monitor di visualizzazione real-time dei muletti, testando i movimenti di 3 muletti al raggiungimento delle task assegnate. È stato messo alla prova il sistema di gestione delle collisioni e di calcolo del percorso; per poi passare alla guida manuale di un muletto.

L'esposizione ha avuto riscontro positivo da parte del proponente, che ha evidenziato come il "core" dell'applicazione sia stato realizzato coerentemente ai requisiti prefissati. È stato suggerito di distinguere i muletti in circolazione colorando in modo diverso o comunque differenziando in qualche modo le icone dei muletti.

2.3 Chiarimenti su Docker

Si sono discusse le difficoltà finora riscontrate con Docker ed i dubbi sulla configurazione con i seguenti riscontri:

- esisteranno tre dockerfile:
 1. macchina server, il quale ip dev'essere statico e conosciuto di modo da aggiungerlo nelle configurazioni dei client;
 2. client per muletti, da far girare su un dispositivo associato a questi, che avrà le caratteristiche necessarie (eseguire node/angular, browser per accedere all'interfaccia);
 3. client per utenti (responsabili e admin), da far girare su dispositivi con caratteristiche analoghe a quelli per i client muletti;
- per le configurazioni di parametri variabili ci sono due possibilità:
 1. creare una cartella condivisa tra il file system della macchina host ed il container che gira su questa, e qui mettere un file (e.g.: *config.cfg* con il contenuto necessario);



2. passare parametri necessari alla configurazione tramite ambiente;
- in ogni caso, si dovrà scegliere una strada e mantenere la consistenza.

2.4 Modalità di parcheggio dei muletti

É stata richiesta al proponente conferma sulla modalità di gestione dei muletti quando essi dovessero rimanere in uno stato di "pausa" perché non presenti nuove *Task* assegnabili: essi possono dirigersi in una località esterna alla mappa, tramite un passaggio nella stessa, nella quale sia previsto il "parcheggio" delle unità. Esse rimangono a disposizione per nuovi compiti da svolgere qualora questi venissero aggiunti dal Responsabile.

2.5 Proposta di file *.json* per la persistenza

É stata presentata una proposta di struttura per i file *.json* da utilizzare per la persistenza dei dati di autenticazione di muletti e utenti. La struttura si è rivelata coerente con quanto atteso, il proponente ha segnalato come non sia necessario applicare *hashing* anche ai token dei muletti.

2.6 Chiusura del server

L'applicativo non dispone di un modo per interrompere la componente server: per ora è prevista la chiusura tramite interruzione del processo (*ctrl-c* da terminale). Il proponente non pone vincoli su questo aspetto, ma sottolinea l'importanza di mantenere uno stato stabile delle risorse dopo la chiusura per quanto riguarda i dati salvati. Dato che l'accesso alle risorse di persistenza avviene tramite cicli completi di *apertura – lettura/scrittura – chiusura* delle risorse, l'interruzione dell'applicazione server non dovrebbe mettere a rischio la corretta conservazione dei dati.

3 Tracciamento delle decisioni

Codice	Decisione
VE_7.1	Unico modulo per l'autenticazione (lato client).
VE_7.2	Architettura dei client secondo il pattern MVC è corretta.
VE_7.3	Architettura del server risulta sostanzialmente corretta.
VE_7.4	Il core dell'applicazione è realizzato coerentemente con i requisiti prefissati.
VE_7.5	Il parcheggio delle unità può avvenire in un'area esterna alla mappa accessibile tramite un passaggio presente sulla stessa.
VE_7.6	L'indirizzo ip del server può essere statico e conosciuto.
VE_7.7	Le configurazioni necessarie per l'avvio dei container docker si possono passare tramite file di configurazione da condividere host-container o tramite ambiente.
VE_7.8	I muletti non necessitano di password per l'autenticazione.
VE_7.9	Importante mantenere coerente il salvataggio dei dati in caso di chiusura del server.