



>Three Way Milkshake_

Verbale esterno 3

17 Febbraio 2021

Three Way Milkshake - Progetto "PORTACS"

threewaymilkshake@gmail.com

Versione	0.1.0
Stato	Approvato
Uso	Esterno
Approvazione	
Redazione	Greggio Nicolò
Verifica	Zuccolo Giada
Destinatari	Three Way Milkshake Prof. Vardanega Tullio Prof. Cardin Riccardo

Descrizione

Verbale del meeting del 2021-02-17 del gruppo Three Way Milkshake con il proponente Sanmarco Informatica.



>Three Way
Milkshake_

Verbale esterno 3

Registro delle modifiche

Vers.	Descrizione	Redazione	Data red.	Verifica	Data ver.
0.1.0	Stesura e verifica del verbale	Greggio Nicolò	2021-02-21	Zuccolo Giada	–



Indice

1	Informazioni generali	3
1.1	Dettagli sull'incontro	3
1.2	Ordine del giorno	3
2	Verbale della riunione	4
2.1	Meccanismi di login	4
2.2	Interfaccia di guida	4
2.3	Riclassificazione dei Requisiti	4
2.3.1	Sui vincoli	5
2.4	Discussione bozza operativa PoC	6
2.4.1	Fonti suggerite dal proponente riguardo thread, timer e socket	6
3	Tracciamento temi affrontati	8



1 Informazioni generali

1.1 Dettagli sull'incontro

- **Luogo:** Incontro telematico tramite piattaforma Google Meet;
- **Data:** 2021-02-17;
- **Ora di inizio:** 17:05;
- **Ora di fine:** 18:20;
- **Partecipanti interni:** (6/6)
 - Chiarello Sofia;
 - Crivellari Alberto;
 - De Renzis Simone;
 - Greggio Nicolò;
 - Tessari Andrea;
 - Zuccolo Giada.
- **Partecipanti esterni:** (1)
 - Beggiato Alex (Sanmarco Informatica).

1.2 Ordine del giorno

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

- rivisitazione meccanismo login;
- interfaccia di guida;
- riclassificazione dei requisiti;
- discussione bozza operativa per PoC.



2 Verbale della riunione

2.1 Meccanismi di login

- adottare login con user e password per amministratori e responsabili;
- muletti come entità diventano nuovi attori per le operazioni che si possono svolgere a bordo
 - non serve login operatore;
 - al momento della connessione scambio token con server per identificazione;
 - utente a bordo è informazione in più
 - * fa parte della sezione facoltativa "pedoni" solo quando scende dal mezzo;
 - * il muletto comunica con il server indipendentemente;
- non serve appoggiarsi a servizi esterni, può essere realizzato internamente dal gruppo, seguendo buone pratiche di sicurezza:
 - comunicazioni con protocollo https
 - * certificato autogenerato "self signed" va bene;
 - * non serve acquistare/generare;
 - salvare hash delle password
 - * potenzialmente anche salted hash;
 - se si adottano sessioni queste devono avere scadenza.

2.2 Interfaccia di guida

- guida manuale rimane obbligatoria
 - unico modo di testare capacità di adattamento del sistema;
- il controllo delle unità non deve per forza avvenire su dispositivi/schermi/interfacce dedicate ma può essere centralizzato in unico pannello tramite il quale l'amministratore può far intraprendere qualunque azione ad ogni unità

2.3 Riclassificazione dei Requisiti

Durante il confronto sono emersi i seguenti punti notabili riguardo le differenze tra requisiti di vincolo e funzionali:

- **vincolo:**
 - tutto ciò che a sistema tratto come condizione iniziale;
 - sistema in cui lavoro è rigido, non c'è possibilità di ridiscutere; questi punti fermi
 - eg: tutto ciò che riguarda mappa e spazi;
- **funzionale:**
 - comportamento del sistema;
 - ciò che il software fa, su cui posso quindi lavorare e migliorare.



I requisiti di vincolo sono dunque stati rivisti e classificati come segue (**F**: funzionale, **V**: vincolo):

1. F;
2. F;
3. F;
4. tutto quello che si muove all'interno è censito dal sistema
 - (a) non esiste unità non riconosciuta e controllata dal sistema;
 - (b) non c'è nulla che il sistema non conosca e da cui non riceva dati;
5. F;
6. V;
7. V;
8. F, compito del nostro lavoro, derivato da 6 e 7, renderlo valido;
9. F
 - sparisce se 10, 11 e 12 vengono realizzati;
 - 10, 11 e 12 sono sotto funzionalità del 9.

I requisiti di vincolo dal 10 al 35 diventano tutti funzionali.

2.3.1 Sui vincoli

- se si impongono vincoli su browser, dare 1 o più versioni "secche" e non intervalli
 - altrimenti si parla di requisiti tecnici;
 - lo può imporre il gruppo;
- si possono introdurre altri tipi di vincoli come:
 - SO del server;
 - versioni docker, kubernetes se usati;
 - per garantire certezza nel funzionamento;
 - possono essere definiti alla fine dello sviluppo come requisiti tecnici.

2.4 Discussione bozza operativa PoC

Il gruppo ha delineato la seguente bozza operativa:

Java	Nodejs	Angular	JSON
utilizzo thread	comunicazione con socket server (java)	mostrare creazione interfacce utili	gestione salvataggio utenti fissi
utilizzo socket	invio periodico di dati (riguardo unità)	interfaccia gui- da/suggerimenti basic probabilmente automatica/simulata	gestione salvataggio planimetria inizialmente finta e statica
liste di task hard-coded		potenzialmente abbozzare modifica planimetria	

la quale è stata confermata dal proponente, con le seguenti note:

- si possono adottare 3 strategie per la gestione della concorrenza nel server centrale:
 1. completamente real time;
 2. completamente temporizzata: il sistema rielabora ad ogni intervallo specifico;
 3. ibrida: rielaboro ad intervalli, a meno che non ci siano dei segnali importanti che richiedono una gestione immediata;
- si consiglia di adottare approccio temporizzato tramite timer task, almeno per il PoC
 - gestire casi particolari (non ho ricevuto segnale da unità...);
- per quanto riguarda i socket, montare sopra qualche web server per gestire chiamate con rest (eg: tomcat)
 - per aprire chiamate verso l'esterno;
 - ne esistono anche di piccoli;
 - per ridurre i tempi via api rest;
- la gestione dei dati via JSON va bene, non è obbligatorio avere in seguito un db;
- per scritture concorrenti è sufficiente adottare timestamps e metodi synchronized.

Riguardo agli sviluppi successivi sono emersi i seguenti commenti:

- si possono adottare librerie grafiche (js puro + wrapper) per il monitor real time;
- l'utilizzo di java anche lato client potrebbe rendere più comoda la comunicazione con il server.

2.4.1 Fonti suggerite dal proponente riguardo thread, timer e socket

- [Java and timer task](#);
- [scheduled executor service](#);



- [guide to java sockets](#);
- [simple http server in java](#);
- [corso java html.it](#)
 - capitoli: 33, 34, 35, 36, 79, 80, 81, 82.



3 Tracciamento temi affrontati

Codice	Domanda	Risposta
VE_3.1	user e password per login	solo per amministratore e responsabile, dettagli in 2.1
VE_3.2	interfaccia guida PoC	va bene anche controllo centralizzato
VE_3.3	riclassificazione requisiti	vedi 2.3
VE_3.4	vincoli su browser	solo versioni secche, altrimenti diventano requisiti tecnici
VE_3.5	bozza operativa poc	va bene come definita dal gruppo, ulteriori commenti in 2.4