



>Three Way Milkshake_

Verbale esterno 1

22 Dicembre 2020

Three Way Milkshake - Progetto "PORTACS"

threewaymilkshake@gmail.com

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

Descrizione

Verbale del meeting del 2020-12-22 del gruppo Three Way Milkshake con il
proponente Sanmarco Informatica



Registro delle modifiche

Versione	Descrizione	Data	Nominativo	Ruolo
1.0.0	Approvazione del verbale	2021-01-10	Greggio Nicolò	Responsabile
0.2.0	Verifica del verbale	2021-01-08	Simone De Renzis	Verificatore
0.1.0	Stesura del verbale	2020-12-22	Zuccolo Giada	Redattore



Indice

1	Informazioni generali	3
1.1	Dettagli sull'incontro	3
1.2	Ordine del giorno	3
2	Verbale della riunione	4
2.1	Fornitura API	4
2.2	Funzioni del server centrale e di ogni unità	4
2.3	Sviluppo delle UI	5
2.4	Presenza di server di backup per aggiornamenti	5
2.5	Tecnologie da usare	5
2.6	Geolocalizzazione simulata	5
2.7	Richiesta della lista dei bug	5
2.8	Definizione POI	6
2.9	Strumenti da usare per la comunicazione	6
3	Sommario	6



1 Informazioni generali

1.1 Dettagli sull'incontro

- **Luogo:** Incontro telematico tramite piattaforma Google Meet;
- **Data:** 2020-12-22;
- **Ora di inizio:** 11:30;
- **Ora di fine:** 12:30;
- **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 punti sotto elencati:

- Fornitura API;
- Funzioni del server centrale e di ogni unità;
- Sviluppo delle UI;
- Presenza di server di backup per aggiornamenti;
- Tecnologie da usare;
- Geolocalizzazione simulata;
- Richiesta della lista dei bug;
- Definizione POI;
- Strumenti da usare per la comunicazione.



2 Verbale della riunione

In occasione del meeting virtuale, il gruppo ha raccolto i dubbi sorti dopo una prima analisi del capitolato e li ha rivolti al proponente Beggiato Alex di Sanmarco Informatica sotto forma di domande. La discussione è proceduta per punti, intavolando un dialogo con il proponente che ha permesso di chiarire gli aspetti che risultavano oscuri.

2.1 Fornitura API

In primo luogo è stata affrontata la questione legata alla fornitura delle API, richiesta dal capitolato ma non del tutto chiara al gruppo nel significato e nelle modalità. È emersa la necessità di fornire due API:

- una per la gestione dell'invio dei dati da parte delle unità al sistema centrale;
- l'altra utilizzata dal monitor grafico per interrogare le unità sui loro dati.

Nel caso si decidesse di utilizzare API REST, il proponente suggerisce l'utilizzo del linguaggio Curl e del software Swagger per la loro documentazione.

2.2 Funzioni del server centrale e di ogni unità

Successivamente è stata affrontata una discussione generale sul progetto e sulle entità che ne fanno parte. I dubbi del gruppo vertevano sul rapporto tra il server e le unità: il proponente ha chiarito come i mezzi in movimento debbano essere pilotabili sia dall'utente che dal server, con possibilità di scegliere per ogni unità una modalità di guida manuale o automatica.

- Nel caso di guida manuale, l'utente ha controllo completo dell'unità, ma visualizza i suggerimenti del server sulle direzioni da percorrere;
- nel caso di guida automatica, il server deve farsi carico di guidare le unità nell'adempimento dei propri compiti, evitando le collisioni con altre unità o con elementi strutturali dell'ambiente in cui si muovono.

Il sistema centrale raccoglie i dati relativi alla posizione delle unità e fornisce ad ogni mezzo una risposta sulla prossima azione da intraprendere.

Il proponente ha poi chiarito che l'applicativo da realizzare debba riguardare un preciso contesto a nostra scelta. Da qui la richiesta di scegliere tra tre possibilità:

- Robot camerieri in un ristorante;
- Muletti all'interno di una fabbrica;
- Auto a guida autonoma.

La documentazione del capitolato fornisce indicazioni generali per queste tre categorie: definire uno contesto più ristretto porterà ad una selezione dei vincoli, dei requisiti e dei casi d'uso.

2.3 Sviluppo delle UI

Lo sviluppo delle UI necessarie dipende strettamente dalla decisione che verrà presa riguardo la definizione del contesto di applicazione del prodotto. Sono state delineate due idee, valide indipendentemente dal dominio scelto. La prima proposta è quella di sviluppare un'interfaccia per ogni unità, con i controlli necessari al suo funzionamento. La seconda proposta vede affiancato anche lo sviluppo di un'interfaccia unica per il coordinamento di tutte le unità: in entrambi i casi, la guida di un mezzo può limitarsi a quattro frecce direzionali e un pulsante di start e stop.

2.4 Presenza di server di backup per aggiornamenti

Il seminario di approfondimento tenuto dal proponente esplorava l'architettura di un sistema in grado di effettuare aggiornamenti su se stesso mantenendo operativa l'infrastruttura. Il gruppo si è quindi chiesto se un approccio simile dovesse essere applicato anche per il progetto. Anche in questa occasione è stato sottolineato come la scelta di un contesto preciso definisce aspetti aggiuntivi come la presenza di un server di ridondanza. Per esempio, nel dominio delle auto a guida autonoma, il server non può prevedere periodi di down, necessitando le unità di un controllo costante. Invece nel contesto dei muletti all'interno di una fabbrica, si può pianificare un intervallo di tempo accettabile durante il quale il server perfeziona l'aggiornamento: questo può avvenire ad esempio nei giorni in cui la fabbrica è chiusa.

2.5 Tecnologie da usare

Essendo la documentazione del capitolato povera di indicazioni sulle tecnologie da preferire, il meeting con il proponente è stata occasione per esplorare alcune alternative. Lo sviluppo del server prevede la realizzazione di algoritmi complessi per la gestione dei messaggi scambiati tra le componenti del sistema: è stato suggerito lo sviluppo in Java con l'ausilio di server web come Tomcat, a fronte di linguaggi come Node.js, meno pratici nella gestione del multithreading.

Per quanto riguarda lo sviluppo delle singole unità le alternative si fanno più ampie: Node.js è una buona soluzione anche nell'ambito di comunicazione con il server centrale. Analogamente per lo sviluppo delle UI la scelta rimane aperta ai linguaggi web come Angular.js o React.

2.6 Geolocalizzazione simulata

Un vincolo del capitolato prevede la geolocalizzazione delle unità. L'implementazione di questa pratica può limitarsi ad una simulazione: i dati da inviare al sistema centrale per l'elaborazione si possono proporre come coordinate di una matrice (derivante dalla mappatura dell'ambiente reale nella sua rappresentazione nel programma), dichiarando se l'unità in quel preciso punto è ferma o aggiungendo la direzione nella quale sta andando. Ulteriore variabile, però facoltativa, è la velocità con la quale l'unità si sta spostando.

2.7 Richiesta della lista dei bug

Tra le richieste del capitolato compariva la presentazione di una lista dei bug: il gruppo ha voluto approfondire questo vincolo. Il proponente richiede infatti un elenco dei bug incontrati nello sviluppo e la loro risoluzione (se attuata): l'applicativo Jira proposto e utilizzato



dal gruppo è stato indicato come servizio utilizzabile per generare ed esportare i report delle problematiche incontrate.

2.8 Definizione POI

La mappatura dell'ambiente reale nell'applicativo dev'essere arricchito con i punti di interesse che le unità devono raggiungere. I POI sono definiti globalmente e dovranno essere disponibili a tutti; ogni singola unità avrà la sua personale lista, che sarà un sottoinsieme della lista completa di tutti i possibili punti. Ancora una volta, risulta necessaria la definizione di un contesto preciso per poter ragionare e successivamente sviluppare un modello che possa agire in modo adeguato e coerente. Ad esempio nel caso dei robot camerieri, dev'essere prevista la possibilità di variare dinamicamente l'ordine dei POI durante l'esecuzione: si pensi ad esempio ad una nuova richiesta proveniente da un cliente, che dovrà accodarsi alle richieste da eseguire se necessario variando anche l'ordine con cui i compiti sono evasi. In un magazzino invece, i muletti sono caricati dalla merce che dovranno smistare in un gruppo di POI definito a priori e che non cambierà prima dell'evasione di tutte le consegne.

2.9 Strumenti da usare per la comunicazione

Nell'ultima parte della riunione ci si è accordati sugli strumenti da utilizzare per la comunicazione con l'azienda.

- Google Meet per le riunioni;
- Servizio di posta elettronica e una stanza virtuale su Google Chat per la messaggistica.

3 Sommario

Nel corso di questa riunione sono stati discussi dei dubbi che il gruppo aveva raccolto riguardo il progetto proposto. Il gruppo si riunirà al più presto per concretizzare le proposte avanzate durante questo incontro in decisioni, vincoli e requisiti riguardanti il dominio scelto. Le decisioni prese verranno tracciate nel verbale relativo alla riunione.