



>Three Way Milkshake_

Verbale esterno # 1

22 DICEMBRE 2020

Three Way Milkshake - Progetto "PORTACS"

threewaymilkshake@gmail.com

Versione	0.0.1
Stato	Non approvato
Uso	Esterno
Approvazione	Nome approvatore
Redazione	Zuccolo Giada
Verifica	De Renzis Simone
Destinatari	San Marco Informatica 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 San Marco Informatica



Registro delle modifiche

Versione	Descrizione	Data	Nominativo	Ruolo
1.0.0	Approvazione del documento	2021-01-10	NomeResponsabile	Responsabile
0.2.0	Verifica del documento	2021-01-08	Simone De Renzis	Verificatore
0.1.1	Aggiunta sezione condizionale	2021-01-07	Tessari Andrea	Redattore
0.1.0	Stesura del documento	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	4
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	5
2.9	Strumenti da usare per la comunicazione	6
3	Sommario	7



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 (San Marco 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 San Marco 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.

Per il loro sviluppo è stato consigliato l'uso dei linguaggi *Swagger* e *Curl*.

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. E? –TODO–

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.

–CONTINUARE DA QUI–

2.3 Sviluppo delle UI

La direttiva generale per la UI per la gestione di un'unità, presenterebbe 4 frecce direzionali e un pulsante start/stop. Lo sviluppo delle UI necessarie dipende strettamente dalla decisione che verrà presa riguardo la definizione del preciso contesto di sviluppo. Sono state quindi delineate due idee. 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 per un ipotetico gestore del coordinamento delle unità, con i controlli necessari per poter gestire ciascuna unità singolarmente. In ogni caso il consiglio è stato di non lasciare il controllo totale al sistema e di fornire un'implementazione migliore, sviluppando al meglio le UI, quindi affiancando al pilota automatico per la gestione di ogni unità, anche dei controlli manuali che un operatore può scegliere di utilizzare.

2.4 Presenza di server di backup per aggiornamenti

La discussione poi si è orientata sulle specifiche del server, in particolare sulla presenza o meno di un server di backup. Anche in questa occasione è stato sottolineato come la scelta di un contesto preciso definirebbe tutti i vari aspetti aggiuntivi come quello di un server di backup. Per esempio per quanto riguarda il contesto delle auto a guida autonoma, il server non può assolutamente andare in down, in quanto provocherebbe danni anche irreparabili. Invece nel contesto dei muletti all'interno di una fabbrica, si può pianificare un tempo accettabile in cui il server si può aggiornare e questo può avvenire nei giorni in cui la fabbrica è chiusa.

2.5 Tecnologie da usare

Successivamente, è stato chiesto un consiglio sulle tecnologie da utilizzare per lo sviluppo delle varie parti dell'applicativo. Non è stato dato nessun vincolo sulla scelta delle tecnologie, ma solo alcuni utili consigli. Per lo sviluppo della parte server, vista la necessità di fare algoritmi complessi per via dei messaggi in parallelo che dovranno essere elaborati abbastanza velocemente, viene sconsigliato il Node, in quanto troppo restrittivo per l'implementazione di algoritmi che potrebbero articolarsi con sviluppi nel multithreading. E' stato quindi suggerito lo sviluppo in Java (o di una qualche web application in Java) con l'ausilio di server web come Tomcat. Per quanto riguarda lo sviluppo delle singole unità, vista la non presenza di design pattern molto strutturati, è stato invece proposto il Node.js, sia per la web app che lo sviluppo del motorino di comunicazione con il server centrale. Al contempo è comunque stata data anche la possibilità di scegliere liberamente quale tecnologia utilizzare. Analogamente per lo sviluppo delle UI è stato indicato come qualsiasi linguaggio web possa essere utilizzato senza problemi, come ad esempio Angular.js o React, permettendo quindi una velocità di sviluppo maggiore.

2.6 Geolocalizzazione simulata

Discutendo riguardo la geolocalizzazione simulata è emerso che tutto dipende da come si implementa il sistema. I dati da mandare al sistema centrale per la sua elaborazione si possono comunque proporre come coordinate di una matrice, e dichiarando se l'unità in quel preciso punto è ferma o aggiungendo la direzione nella quale sta andando. Come variabile facoltativa c'è la velocità con la quale l'unità si sta spostando.

2.7 Richiesta della lista dei bug

Visto che nelle richieste era citata la presentazione di una lista dei bug, il gruppo ha voluto chiedere spiegazioni a riguardo. A questo proposito è stato spiegato che sarà necessario presentare la lista di bug incontrati e la loro risoluzione (qualora ci fosse stata). L'applicativo Jira proposto dal gruppo è stato indicato come servizio utilizzabile per generare ed esportare il report con la lista dei bug.

2.8 Definizione POI

Successivamente è stato affrontato l'argomento dei POI, in particolare i vincoli riguardanti l'ordine di una lista di POI assegnati ad un'unità. Ad esempio è stato posto l'esempio di un'unità potesse rimandare il raggiungimento di un POI occupato o se dovesse rimanere in



attesa e per quanto tempo. Questo vale anche per la gestione dei POI dal punto di vista dell'unità. Ad esempio i nuovi POI assegnati ad un'unità possono essere posti in coda alla lista, oppure l'unità, una volta arrivata nel POI in cui si stava dirigendo, decidere il successivo POI da raggiungere facendo una valutazione su quale sarebbe il più veloce da raggiungere. Risulta quindi necessaria la definizione di un contesto preciso per poter ragionare e successivamente sviluppare un modello che possa agire in modo adeguato e coerente con il contesto stesso. Inoltre, è stato dichiarato che i POI dovranno essere definiti globalmente e dovranno essere disponibili a tutti; ogni singola unità avrà la sua personale lista di interesse che dovrà raggiungere, che sarà un sottoinsieme della lista completa di tutti i possibili punti.

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. Innanzitutto è chiaro come le funzionalità delle varie entità che costituiscono il progetto possono essere determinate solo dopo aver definito il contesto specifico dell'applicativo (robot camerieri, muletti, auto a guida autonoma). Da questa scelta, seguiranno quelle riguardanti le funzionalità del server e delle singole unità e delle UI e riguardo la gestione degli aggiornamenti del server e della lista di POI da soddisfare. Una volta deciso come implementare il sistema ci si può concentrare sulla gestione della geolocalizzazione simulata. Sono state date informazioni riguardo le API, dove è stato specificato che ne saranno necessarie due e che cosa ci può avvalere per poterle sviluppare, e sono stati dati dei consigli riguardo quali tecnologie utilizzare. Infine è stato chiarito il dubbio riguardo la lista di bug da presentare e sono stati accordati gli strumenti da utilizzare per la comunicazione con l'azienda.