



Error__418

[GitHub/Error-418-SWE](https://github.com/Error-418-SWE)

error418swe@gmail.com

Verbale esterno 15/11/23

Avvio dell'Analisi dei Requisiti

Informazioni

Versione	1.0.0
Uso	Esterno
Stato	Approvato
Responsabile	Oseliero Antonio
Redattore	Carraro Riccardo
Verificatore	Banzato Alessio
Destinatari	Gruppo Error_418 Sanmarco Informatica Vardanega Tullio Cardin Riccardo
Referente	Bassani Matteo

1 Informazioni generali

- Luogo: Discord
- Data e ora: 15/11/23 @ 10:00 ~ 11:00
- Partecipanti (8):
 - Banzato Alessio
 - Bassani Matteo
 - Carraro Riccardo
 - Gardin Giovanni
 - Nardo Silvio
 - Oseliero Antonio
 - Todesco Mattia
 - Zaccone Rosario

2 Ordine del giorno

- Approfondimento specifiche del capitolato;
- Presentazione, supportata da una board Miro, delle domande raccolte dal team sostanziate con proposte e mock up. Le domande si articolano in:
 - identificazione della tipologia di utente a cui il prodotto è destinato;
 - prime proposte di user interface e user experience;
 - differenza di funzionalità tra applicativo desktop e mobile;
 - funzionalità desiderate;
 - API RESTful.
- Conferme riguardo lo stack tecnologico individuato;
- Scelta del canale di comunicazione preferenziale tra Proponente e Fornitore;
- Organizzazione di un prossimo meeting e programmazione regolare degli incontri.

2.1 Approfondimento specifiche del capitolato

Lo scopo del meeting è iniziare l'analisi dei requisiti, focalizzando l'attenzione sulla comprensione delle funzionalità richieste, la loro implementazione e il dominio tecnologico che il team utilizzerà. Le domande raccolte dal team sono state catalogate nei seguenti ambiti:

- use case;
- funzionalità;
- user interface e user experience;
- stack tecnologico.

2.2 Presentazione delle domande raccolte dal team in merito ai requisiti

2.2.1 Utente

L'utente tipico è stato individuato nella figura di un amministratore e/o impiegato lato back-office, dotato dunque di massimi privilegi durante l'uso dell'applicativo.

Le principali azioni che tale utente svolgerà sono:

- interrogazione del magazzino;
- richiesta di spostamento di un prodotto;
- zoom e spostamento all'interno dell'ambiente 3D.

2.2.2 User interface & user experience

Durante l'incontro si è presentata un'idea generale di possibile user interface. Sono state individuate alcune caratteristiche desiderabili, quali:

- esporre con chiarezza le operazioni disponibili all'utente (distinzione evidente tra elementi selezionabili e non);
- evitare eccessive animazioni o effetti *lampeggianti*;
- evidenziare in modo chiaro gli elementi ricercati (ad esempio riducendo l'opacità degli altri elementi), al fine di non disorientare l'utente;
- spostare la telecamera sull'oggetto ricercato (se singolo);
- essenziale che la ricerca non modifichi la struttura visualizzata del magazzino.

2.2.3 Differenze tra applicativo desktop e mobile

Durante la riunione è emerso che vi saranno delle differenze tra la versione desktop e la versione mobile. L'accesso completo alla funzionalità (esempio richiesta di spostamento di prodotti) sarà fornito per la versione desktop, mentre la versione mobile presenterà delle funzioni limitate (ad esempio solo visualizzazione e ricerca). In ogni caso, il gap funzionale tra dispositivi rimane ancora da definire nel dettaglio. L'applicativo sarà responsive.

2.2.4 Funzionalità desiderate

Da una prima analisi sono emerse le principali funzionalità desiderate, quali:

- creazione dell'ambiente 3D del magazzino, articolata in:
 - definizione dei parametri dimensionali del magazzino ;
 - creazione e posizionamento nello spazio degli scaffali;
 - definizione dei bin, ovvero le locazioni capaci di accogliere i prodotti;
- caricamento dei dati da un database per la popolazione del magazzino;
- interrogazione del magazzino mediante ricerche per ID, nome, o caratteristiche dei prodotti;
- evidenziazione dei prodotti ricercati mediante effetti visivi (ad esempio cambio del colore) e/o spostamento della telecamera;
- invio di notifiche mediante la chiamata a API RESTful per lo spostamento di prodotti nel magazzino.

Durante l'analisi del processo di configurazione si è chiarito che il primo passaggio di configurazione riguarda la definizione della geometria del magazzino. Per un magazzino a pianta rettangolare, la geometria può essere definita con tre parametri (lunghezza, larghezza, altezza); per scenari più complessi, il software deve essere in grado di definire la geometria del magazzino dai <path> contenuti in un file SVG caricato dall'utente. Successivamente, vengono caricati i dati presenti nel DB. Il database non impone vincoli strutturali, ma è pensato per interfacciarsi con la struttura determinata in fase di creazione.

2.2.5 API RESTful

Per l'invio delle notifiche di richiesta di spostamento, l'applicativo si interfacerà con delle API RESTful. È emerso che l'implementazione di tali API non è di interesse del capitolato, e per tale motivazione possiamo procedere ad una simulazione del comportamento delle stesse con assunzioni riguardo i parametri e le risposte. Ad esempio per lo spostamento di un elemento da un bin ad un altro:

- invio all'API delle coordinate di partenza (es: scaffale) e delle coordinate d'arrivo (un altro scaffale) e il prodotto interessato;
- l'API risponde:
 - stato 200 (contatto con l'API avvenuto con successo): spostamento richiesto con successo;
 - stato 200 (contatto con l'API avvenuto con successo): spostamento non possibile;
- risposte alle interrogazioni ottenuta in formato JSON;
- la logica di controllo della facoltà di spostare un prodotto è gestita interamente dall'endpoint REST.

2.3 Conferme riguardo lo stack tecnologico individuato

Lo stack tecnologico individuato dal team si articola in:

- DB SQL per la gestione del database relazionale;
- Three.js per l'implementazione della parte grafica (JavaScript o TypeScript);
- Node.js per la comunicazione tra applicativo e database;
- Docker per la containerizzazione del prodotto.

Lo stack tecnologico discusso rappresenta un punto di partenza, ma è soggetto a modifiche. Ad ogni modo, al momento, le tecnologie individuate e proposte hanno trovato l'approvazione del Proponente. In merito all'utilizzo di JavaScript o TypeScript ed eventuali framework per il lato front end (ad esempio React, Angular), il Proponente ha lasciato la massima libertà al Gruppo. L'azienda Proponente utilizza internamente TypeScript e Angular.

2.4 Stabilimento delle metodologie di contatto

Il Proponente ha confermato l'intenzione di proseguire i contatti mediante l'uso di posta elettronica.

2.5 Organizzazione di un prossimo meeting e programmazione regolare degli incontri

L'incontro si è concluso con la programmazione del prossimo meeting in data 23 novembre 2023, dalle ore 14:00 alle ore 15:00.

In questa prima fase di analisi dei requisiti, Proponente e Fornitore si sono trovati d'accordo circa la necessità di svolgere incontri con cadenza almeno settimanale.

3 Azioni da intraprendere

A seguito del meeting sono state individuate le seguenti operazioni da svolgere:

- raccolta e catalogazione di aspettative e bisogni;

-
- produzione della documentazione quale use cases, analisi dei requisiti da presentare al prossimo meeting;
 - esplorazione del dominio tecnologico.

A handwritten signature in black ink, appearing to read 'Mth Ami.', is positioned above a horizontal line.

Firma partecipanti esterni: