



Error__418

[GitHub/Error-418-SWE](#)

error418swe@gmail.com

Verbale interno 26/02/24

Meeting di Design Thinking

Informazioni

Versione	WIP
Uso	Interno
Stato	Approvato
Responsabile	Carraro Riccardo
Redattore	Banzato Alessio
Verificatore	Oseliero Antonio
Destinatari	Gruppo Error__418 Vardanega Tullio Cardin Riccardo

1 Informazioni generali

- Luogo: Discord_G
- Data e ora: 26/02/24 @ 15:10 ~ 16:50
- Partecipanti (7):
 - Banzato Alessio
 - Carraro Riccardo
 - Gardin Giovanni
 - Nardo Silvio
 - Oseliero Antonio
 - Todesco Mattia
 - Zaccone Rosario

2 Ordine del giorno

- Progettazione:
 - Design pattern;
 - Architettura;
 - Diagramma ER.
- Pianificazione.

2.1 Progettazione

2.1.1 Design pattern

Il gruppo ha cercato di determinare i design pattern da utilizzare nella progettazione del prodotto, e ha individuato le seguenti ipotesi:

- model-view pattern, in particolare il model-view-controller (MVC);
- repository_G pattern.

Il gruppo individua, inoltre, la possibilità di inserire il proxy pattern nel contesto della comunicazione con le API_G.

Sono sorti dei dubbi riguardo la possibilità che il model-view-controller sia già implementato nativamente nel framework Next.js_G. In questo caso la sua implementazione risulterebbe un lavoro che aggiungerebbe complessità non necessaria al prodotto. Il gruppo quindi si interroga su come comportarsi nel caso alcuni pattern siano già implementati all'interno di librerie o framework utilizzati.

Durante la discussione riguardo i pattern il gruppo considera la possibilità di non definire classi all'interno del codice. Questo nasce da due osservazioni:

- JavaScript_G non è un vero e proprio linguaggio ad oggetti;
- gli oggetti presenti nel prodotto, intesi come elementi grafici, non hanno comportamenti particolari che necessitano di essere raggruppati e organizzati.

In conseguenza di ciò, sorge la questione di come affrontare i design pattern in mancanza di vere e proprie classi, la quale verrà esposta tramite mail al Professor Cardin.

2.1.2 Architettura

Il gruppo discute sulla scelta dell'architettura più adatta al prodotto da sviluppare, individuando infine l'architettura layered come opzione principale, in quanto l'architettura esagonale, alternativa presa in considerazione, pone molta attenzione sulla business logic, la quale non è presente nel capitolato_G in quanto il prodotto richiesto è un visualizzatore, e di conseguenza non sono presenti dati su cui eseguire modifiche o validazioni. Inoltre, la comunicazione tra componenti è unidirezionale: da database_G verso la presentation logic.

Si è concordato l'utilizzo di un monolite come architettura di deploy, vista la sua semplicità e velocità di sviluppo. Inoltre, non è presente un numero di moduli distinti tra loro tale da giustificare l'utilizzo di un'architettura a microservizi.

2.1.3 ER

Il gruppo concorda con l'utilizzo del primo schema ER prodotto, applicando i seguenti aggiornamenti:

- presenza di coordinate di posizionamento degli scaffali (verbale_G esterno del 22/02/2024);
- considerazione dei bin_G esterni come appartenenti ad un particolare scaffale_G con un unico ripiano_G ad altezza 0 e dimensioni uguali a quelle del piano;
- modifica della relazione bin_G-prodotti: ora i prodotti possono essere presenti nel database_G ma non essere assegnati a nessun bin_G.

2.2 Pianificazione

Il gruppo contatterà via mail:

- il Professor Cardin, per esporre in forma scritta i dubbi sorti durante il meeting oggetto di questo verbale_G. Successivamente alla sua risposta si valuterà se sarà necessario chiedere anche un colloquio via Zoom_G;
- il Proponente, per:
 - confermare il meeting programmato per giovedì 29/02/2024, durante il quale saranno esposti i dubbi di progettazione sorti durante il meeting oggetto di questo verbale_G;
 - convalidare il verbale_G del meeting di giovedì 22/02/2024;
 - inviare lo schema ER del database_G e la bozza del diagramma delle classi, che saranno il punto di partenza per il prossimo meeting.