



Error__418

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

error418swe@gmail.com

Specifica Tecnica

Informazioni

Versione	1.1.1
Uso	Esterno
Stato	Approvato
Responsabile	Zaccone Rosario
Redattore	Todesco Mattia
Verificatore	Banzato Alessio
Destinatari	Gruppo Error__418 Vardanega Tullio Cardin Riccardo

Registro delle modifiche

Ver.	Data	PR	Titolo	Redattore	Verificatore
1.1.1	16-03-2024	370	DOC-598 Modifiche a sezione database	Todesco Mattia	Banzato Alessio
1.1.0	11-03-2024	360	DOC-563 Redigere sezione Tecnologie	Todesco Mattia	Banzato Alessio

Indice dei contenuti

1 Introduzione	1
1.1 Scopo del documento	1
1.2 Approccio al documento	1
1.3 Scopo del prodotto	1
1.4 Glossario	1
1.5 Riferimenti	1
1.5.1 Riferimenti a documentazione interna	1
1.5.2 Riferimenti normativi	1
1.5.3 Riferimenti informativi	2
2 Tecnologie	3
2.1 Introduzione	3
2.2 Elenco delle tecnologie	3
2.2.1 Tecnologie per lo sviluppo dell'applicazione	3
2.2.2 Tecnologie per il testing	3
3 Architettura del prodotto	4
3.1 Descrizione generale	4
3.2 Diagramma delle classi	4
3.3 Database	4
3.3.1 Entità	4
3.3.2 Relazioni	5
3.3.3 Interrogazione del database	5
3.4 Struttura (vari layer)	6
3.5 Design pattern utilizzati	6
4 Requisiti soddisfatti (aggiungere tabella requisiti soddisfatti)	6

Indice delle immagini

Figura 1: Schema ER del Database. 4

Indice delle tabelle

Tabella 1: Tecnologie utilizzate per lo sviluppo. 3

Tabella 2: Tecnologie utilizzate per il testing. 4

1 Introduzione

1.1 Scopo del documento

Il presente documento ha lo scopo di illustrare e motivare le scelte architetturelle effettuate dal gruppo durante le fasi di progettazione e codifica del prodotto *WMS3*.

Questo documento include i diagrammi delle classi per illustrare l'architettura e le funzionalità del prodotto, insieme alle tecnologie selezionate dal team per la realizzazione del progetto.

1.2 Approccio al documento

Il presente documento viene redatto in modo incrementale in modo da assicurare la coerenza delle informazioni al suo interno con gli sviluppi in corso e le esigenze evolutive del progetto.

1.3 Scopo del prodotto

Il seguente documento tratta del programma denominato *WMS3: Warehouse Management 3D*, avente come obiettivo la realizzazione di un sistema di gestione di magazzino_G in tre dimensioni.

Il prodotto offre le seguenti funzionalità principali:

- possibilità di creazione di un magazzino_G e delle sue componenti;
- visualizzazione tridimensionale del magazzino_G, con possibilità di muovere la vista;
- visualizzazione delle informazioni della merce_G presente in magazzino_G;
- caricamento dei dati relativi alle merci da un database_G SQL_G;
- emissione di richieste di spostamento della merce_G all'interno del magazzino_G;
- filtraggio e ricerca delle merci con rappresentazione grafica dei risultati;
- importazione di planimetrie in formato SVG_G.

1.4 Glossario

Al fine di agevolare la comprensione del presente documento, viene fornito un glossario che espliciti il significato dei termini di dominio specifici del progetto. I termini di glossario sono evidenziati nel testo mediante l'aggiunta di una "G" a pedice degli stessi:

Termine di glossario_G

Le definizioni sono disponibili nel documento *Glossario v1.3.0*.

1.5 Riferimenti

1.5.1 Riferimenti a documentazione_G interna

- Documento *Glossario v1.3.0*:
https://github.com/Error-418-SWE_G/Documents/blob/main/3%20-%20PB/Glossario_v1.3.0.pdf (ultimo accesso 25/02/2024)

1.5.2 Riferimenti normativi

- Capitolato_G "Warehouse Management 3D" (C5) di *Sanmarco Informatica S.p.A.*:
https://www.math.unipd.it/~tullio/IS_G-1/2023/Progetto/C5.pdf (ultimo accesso 13/02/2024)

1.5.3 Riferimenti informativi

2 Tecnologie

2.1 Introduzione

In questa sezione vengono esposte ed esaminate le tecnologie utilizzate nel processo di sviluppo e testing_G del prodotto WMS3.

Ciò avviene prestando attenzione alle tecnologie impiegate sia nel front-end che nel back-end, compresa la gestione del database_G e l'integrazione con i servizi previsti.

2.2 Elenco delle tecnologie

2.2.1 Tecnologie per lo sviluppo dell'applicazione

Tecnologia	Descrizione	Versione _G
Docker _G	Software utilizzato per il processo di deployment di applicazioni software. Permette di eseguire processi informatici in ambienti isolati chiamati container.	24.0.7
Docker _G -compose	Strumento per la definizione e l'esecuzione di applicazioni multi-container.	2.23.3
Drei	Libreria che fornisce componenti e utilità per semplificare lo sviluppo di applicazioni in 3D utilizzando React _G e Three.js _G .	9.97.6
Fiber	Libreria open-source che facilita l'integrazione di Three.js _G all'interno di applicazioni React _G .	8.15.16
HTML	Linguaggio di markup standard utilizzato per la creazione di pagine web.	5.0
Next.js _G	Framework di sviluppo web front-end basato su React _G e utilizzato per la creazione di applicazioni web.	14.1.0
Node.js _G	Runtime system orientato agli eventi per l'esecuzione di codice JavaScript _G estendibile tramite moduli.	20.11.0
PostgreSQL _G	Sistema di gestione di database _G relazionali.	16.2
React _G	Libreria JavaScript _G utilizzata per la creazione di interfacce utente _G dinamiche e reattive. Si basa sul concetto di "components", ovvero blocchi di codice autonomi che gestiscono la propria logica e rendering.	18.0.0
Tailwind CSS	Framework CSS utilizzato per lo sviluppo di interfacce utente _G web.	3.4.1
Three.js _G	Libreria JavaScript _G utilizzata per creare e visualizzare grafica computerizzata 3D animata in un browser _G Web utilizzando WebGL _G .	0.161.2
Typescript	Superset di JavaScript _G che aggiunge tipizzazione, offrendo maggiore struttura al codice.	5.3.3

Tabella 1: Tecnologie utilizzate per lo sviluppo.

2.2.2 Tecnologie per il testing_G

Tecnologia	Descrizione	Versione _G
Jest _G	Framework di testing _G per JavaScript _G . Utilizzato principalmente per lo unit testing _G , offre la parallelizzazione dei test e il mocking delle dipendenze.	29.7.0

Tabella 2: Tecnologie utilizzate per il testing_G.

3 Architettura del prodotto

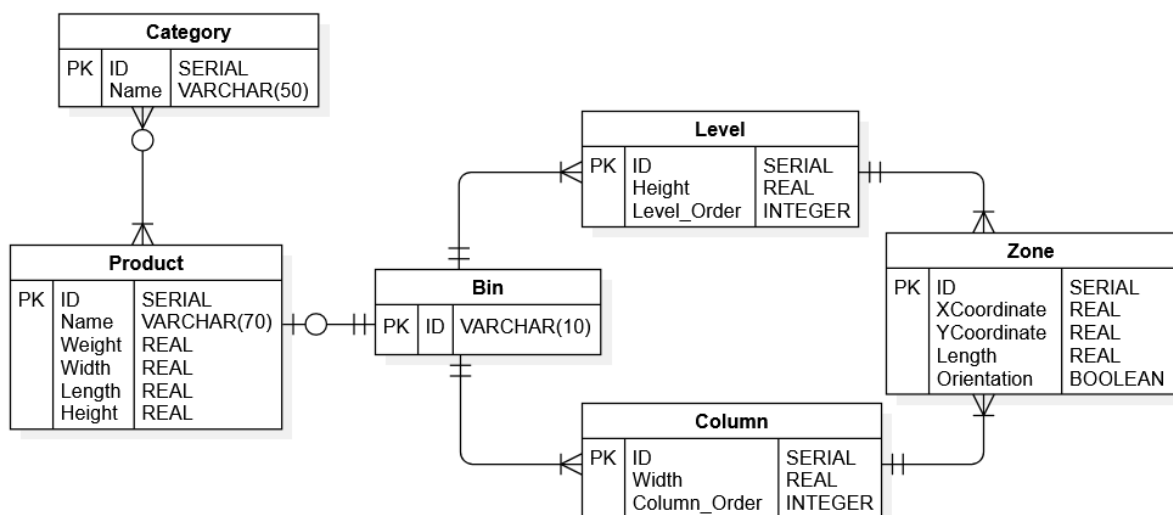
3.1 Descrizione generale

3.2 Diagramma delle classi

3.3 Database_G

In questa sezione viene presentato lo schema della base di dati realizzata con PostgreSQL_G.

Esso è così composto:

Figura 1: Schema ER del Database_G.

3.3.1 Entità

Il database_G è composto da 6 entità:

- **Product**: rappresenta un prodotto presente all'interno del magazzino_G. Composto da:
 - ID: identificativo univoco e seriale di un prodotto;
 - Name: nome del prodotto;
 - Weight: peso del prodotto;
 - Width: larghezza del prodotto;
 - Length: lunghezza del prodotto;
 - Height: altezza del prodotto.

- **Category:** rappresenta la categoria merceologica di appartenenza del prodotto. Composto da:
 - ID: identificativo univoco e seriale di una categoria;
 - Name: nome della categoria.
- **Bin_G:** rappresenta uno spazio del magazzino_G in cui è possibile inserire un prodotto. Composto da:
 - ID: identificativo univoco di un bin_G, esso è così composto:

ID zona + _ + Column_Order + _ + Level_Order"
- **Level:** rappresenta un ripiano_G dello scaffale_G. Composto da:
 - ID: identificativo univoco e seriale di un ripiano_G;
 - Height: altezza del ripiano_G;
 - Level_order: valore incrementale che rappresenta la posizione del ripiano_G all'interno di uno scaffale_G. Se il suo valore è 0 allora esso rappresenta una zona a terra.
- **Column:** rappresenta una colonna dello scaffale_G. Composto da:
 - ID: identificativo univoco e seriale di una colonna;
 - Width: larghezza della colonna;
 - Column_order: valore incrementale che rappresenta la posizione della colonna all'interno di uno scaffale_G.
- **Zone:** rappresenta una zona del piano del magazzino_G. Essa può essere sia uno scaffale_G che una zona a terra. Composto da:
 - ID: identificativo univoco e seriale di una zona;
 - XCoordinate: coordinata orizzontale della zona;
 - YCoordinate: coordinata verticale della zona;
 - Length: lunghezza della zona;
 - Orientation: orientamento della zona.

3.3.2 Relazioni

All'interno del database_G le relazioni fra le differenti entità sono del tipo:

- **Zero..One to One** per quanto riguarda le entità:
 - Product e Bin_G.
- **One to Many** per quanto riguarda le entità:
 - Bin_G e Level;
 - Bin_G e Column;
 - Level e Zone;
 - Column e Zone.
- **Many to Zero..Many** per quanto riguarda le entità:
 - Product e Category.

3.3.3 Interrogazione del database_G

Il database_G viene utilizzato dall'applicazione per il caricamento, il posizionamento e la visualizzazione dei prodotti all'interno del magazzino_G. In nessun caso il database_G verrà modificato dall'applicazione.

3.4 Struttura (vari layer)

3.5 Design pattern utilizzati

4 Requisiti soddisfatti (aggiungere tabella requisiti soddisfatti)