



Error__418

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

error418swe@gmail.com

Specifica Tecnica

Informazioni

Versione	1.7.0
Uso	Esterno
Stato	Approvato
Responsabile	Zacone Rosario
Redattori	Nardo Silvio Todesco Mattia
Verificatore	Banzato Alessio
Destinatari	Gruppo Error__418 Vardanega Tullio Cardin Riccardo

Registro delle modifiche

Ver.	Data	PR	Titolo	Redattore	Verificatore
1.7.0	02-04-2024	406	DOC-654 Estendere sezione Tecnologie	Carraro Riccardo	Gardin Giovanni
1.6.0	27-03-2024	367	DOC-651 Redigere sezione Classi	Nardo Silvio	Gardin Giovanni
1.5.0	27-03-2024	367	DOC-650 Redigere sezione Design pattern utilizzati	Nardo Silvio	Gardin Giovanni
1.4.0	27-03-2024	367	DOC-649 Redigere sezione Struttura	Nardo Silvio	Gardin Giovanni
1.7.0	27-03-2024	367	DOC-652 Redigere sezione Componenti	Nardo Silvio	Gardin Giovanni
1.3.0	21-03-2024	385	DOC-595 Redigere sezione Requisiti soddisfatti	Todesco Mattia	Banzato Alessio
1.2.0	18-03-2024	376	DOC-606 Aggiungere sezione Requisiti	Todesco Mattia	Banzato Alessio
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 alla redazione	1
1.3	Scopo del prodotto	1
1.4	Glossario	1
1.5	Riferimenti	1
1.5.1	Riferimenti a documentazione interna	2
1.5.2	Riferimenti normativi	2
1.5.3	Riferimenti informativi	2
1.5.4	Riferimenti a documentazione tecnica	2
2	Requisiti	4
2.1	Requisiti di sistema minimi	4
2.2	Requisiti hardware	4
2.3	Browser	4
3	Installazione	5
3.1	Scaricare il progetto	5
3.2	Avviare la web app	5
3.3	Terminare l'esecuzione	5
4	Tecnologie	6
4.1	Introduzione	6
4.2	Tecnologie implementative	6
4.2.1	JSX	6
4.2.2	CSS	6
4.2.3	TypeScript	6
4.3	Tecnologie per la validazione dei dati	8
4.3.1	Zod	8
4.4	Tecnologie per la persistenza dei dati	8
4.4.1	SQL	8
4.4.2	JSON	8
4.4.3	PostgreSQL	9
4.5	Tecnologie per il testing	9
4.5.1	Jest	9
4.6	Tecnologie per il deployment	9
4.6.1	Docker	9
4.6.2	Docker-compose	10
5	Architettura di sistema	11
5.1	Architettura di implementazione	11
5.1.1	Vantaggi	11
5.1.2	Svantaggi	11
5.2	Design pattern utilizzati	12

5.2.1 Data Mapper	12
5.2.2 Repository	12
5.2.3 Provider	12
5.2.4 Strategy	12
5.2.5 Factory	12
5.3 Classi e Componenti	12
5.3.1 Persistence layer	12
5.3.2 Business layer	14
5.3.3 Presentation layer	21
5.4 Database	22
5.4.1 Entità	23
5.4.2 Relazioni	24
5.4.3 Interrogazione del database	24
6 Requisiti soddisfatti	25
6.1 Requisiti funzionali soddisfatti	25
6.2 Requisiti di qualità soddisfatti	33

Indice delle immagini

Figura 1: Layered architecture.	11
Figura 2: Diagramma delle classi del layer di persistenza	14
Figura 3: Diagramma delle classi del layer di business	18
Figura 4: Diagramma delle classi Floor del layer di business	19
Figura 5: Diagramma delle classi Search del layer di business	20
Figura 6: Schema ER del Database.	23

Indice delle tabelle

Tabella 1: Requisiti di sistema minimi	4
Tabella 2: Requisiti hardware	4
Tabella 3: Browser supportati	4
Tabella 4: Requisiti funzionali	25
Tabella 5: Requisiti di qualità	33

1 Introduzione

1.1 Scopo del documento

Il presente documento ha come obiettivo la descrizione dettagliata delle scelte progettuali effettuate, al fine di garantire una comprensione chiara e completa del software “WMS3: Warehouse Management 3D”, proposto da Sanmarco Informatica S.p.A.

Gli aspetti fondamentali riguardano l’architettura implementativa, analizzando tecnologie e design pattern adottati, e l’architettura di deployment del prodotto.

Mediante il documento si intende fornire le linee guida per lo sviluppo del software, garantendo la coerenza con i requisiti individuati nel documento di Analisi dei Requisiti_G e il loro soddisfacimento.

1.2 Approccio alla redazione

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

1.3 Scopo del prodotto

Il fine ultimo è lo sviluppo di un software, “WMS3: Warehouse Management 3D”, che rivoluziona la gestione di un magazzino_G, transcendendo la rappresentazione bidimensionale tradizionale a favore di un ambiente tridimensionale più informativo e intuitivo.

“WMS3” si distingue per le sue funzionalità avanzate, tra cui la creazione personalizzata di un magazzino_G e delle sue componenti, arricchita da una visualizzazione tridimensionale che offre una comprensione spaziale ottimale grazie alla possibilità di manipolare la vista.

Il software consente inoltre l’accesso e la visualizzazione delle informazioni relative alla merce_G e alla disposizione degli scaffali, sfruttando un database_G SQL_G per il caricamento di tali dati.

Un altro aspetto fondamentale di “WMS3” è la facilità con cui è possibile emettere richieste di spostamento della merce_G all’interno del magazzino_G, rendendo la gestione logistica un processo semplice e intuitivo. Il software integra funzionalità di filtraggio e ricerca delle merci, presentando i risultati in modo grafico per una interpretazione immediata.

Per concludere, “WMS3” supporta la personalizzazione dell’ambiente attraverso l’importazione di planimetrie in formato SVG_G, permettendo una configurazione avanzata del layout del magazzino_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/Documenti/blob/main/3%20-%20PB/Glossario_v1.3.0.pdf (ultimo accesso 25/02/2024)
- Documento *Analisi dei Requisiti_G v2.0.1*:
https://github.com/Error-418-SWE_G/Documenti/blob/main/3%20-%20PB/Documentazione_G%20esterna/Analisi%20dei%20Requisiti_v2.0.1.pdf (ultimo accesso 25/02/2024)

1.5.2 Riferimenti normativi

- Regolamento del progetto didattico_G:
<https://www.math.unipd.it/~tullio/IS_G-1/2023/Dispense/PD2.pdf> (ultimo accesso 20/03/2024)
- 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

- Verbali interni;
- Verbali esterni;
- Analisi dei requisiti:
<https://www.math.unipd.it/~tullio/IS_G-1/2023/Dispense/T5.pdf> (ultimo accesso 20/03/2024)
- Analisi e descrizione delle funzionalità, Use Case_G e relativi diagrammi (UML_G):
<https://www.math.unipd.it/~rcardin/swea/2022/Diagrammi%20Use%20Case.pdf> (ultimo accesso 20/03/2024)

1.5.4 Riferimenti a documentazione_G tecnica

- Docker_G:
<https://docs.docker.com/engine/> (ultimo accesso 28/03/2024)
- Docker Compose_G:
<https://docs.docker.com/compose/> (ultimo accesso 28/03/2024)
- Jest_G:
<https://jestjs.io/docs/getting-started> (ultimo accesso 28/03/2024)
- Next.js_G:
<https://nextjs.org/docs> (ultimo accesso 28/03/2024)
- PostgreSQL_G:
<https://www.postgresql.org/docs/16/index.html> (ultimo accesso 28/03/2024)
- React_G:
<https://react.dev/reference/react> (ultimo accesso 28/03/2024)
- @react_G-three/drei:
<https://github.com/pmndrs/drei?tab=readme-ov-file#index> (ultimo accesso 28/03/2024)
- @react_G-three/fiber:
<https://docs.pmnd.rs/react-three-fiber/> (ultimo accesso 28/03/2024)



- shadcn/ui:
<https://ui.shadcn.com/docs> (ultimo accesso 28/03/2024)
- Tailwind CSS:
<https://tailwindcss.com/docs/> (ultimo accesso 28/03/2024)
- Three.js:
<https://threejs.org/docs/> (ultimo accesso 28/03/2024)
- Zod:
<https://zod.dev/> (ultimo accesso 28/03/2024)

2 Requisiti

Di seguito sono elencati i requisiti minimi necessari per l'esecuzione dell'applicazione, comprese le caratteristiche necessarie per configurare l'ambiente di sviluppo del progetto.

2.1 Requisiti di sistema minimi

Componente	Versione _G	Riferimenti
Docker _G	≥ 24.0.7	https://docs.docker.com/
Docker _G -compose	≥ 2.23.3	https://docs.docker.com/compose/

Tabella 1: Requisiti di sistema minimi

2.2 Requisiti hardware

Componente	Requisito _G minimo
Processore	Processore a 64 bit con SLAT (Second Level Address Translation)
Memoria RAM	4GB DDR4
Spazio su disco	≥ 20 GB

Tabella 2: Requisiti hardware

2.3 Browser_G

Browser _G	Versione _G
Google Chrome	≥ 89
Microsoft Edge	≥ 89
Mozilla Firefox	≥ 16.4
Apple Safari	≥ 108
Opera Browser _G	≥ 76
Google Chrome per Android	≥ 89
Apple Safari per iOS	≥ 17.1
Samsung Internet	≥ 23

Tabella 3: Browser_G supportati

3 Installazione

3.1 Scaricare il progetto

Vengono fornite due modalità di download del prodotto WMS3: la prima (consigliata) è eseguire il download del prodotto in formato zip o tar.gz dalla pagina

<https://github.com/Error-418-SWE/WMS3/releases>

In alternativa, se nel dispositivo è presente Git, si può clonare il repository con il comando

```
git clone git@github.com:Error-418-SWE/WMS3.git
```

oppure

```
git clone https://github.com/Error-418-SWE/WMS3.git
```

3.2 Avviare la web app

Per avviare la web app è necessario collocarsi all'interno della cartella scaricata al passaggio *Scaricare il progetto* (Sezione 3.1) ed eseguire il comando

```
docker compose up -d
```

Questo avvierà i container Docker che formano il prodotto:

- Container PostgreSQL (database);
- Container Web (web app).

Completato l'avvio dei container, la web app sarà disponibile all'indirizzo

<http://localhost:3000/>

3.3 Terminare l'esecuzione

Per terminare l'esecuzione della web app è necessario collocarsi nella cartella scaricata al passaggio *Scaricare il progetto* (Sezione 3.1) ed eseguire il comando

```
docker compose down
```

4 Tecnologie

4.1 Introduzione

In questa sezione, viene presentata una panoramica completa degli strumenti e delle tecnologie utilizzati per lo sviluppo e l'implementazione del software “WMS3”. Questo include una descrizione dettagliata delle tecnologie, del linguaggio di programmazione adottato, delle librerie e dei framework necessari.

L'obiettivo principale è assicurare che il software sia sviluppato utilizzando le tecnologie adeguate in termini di efficacia ed efficienza e che soddisfi i requisiti individuati nel documento *Analisi dei Requisiti_G*.

4.2 Tecnologie implementative

4.2.1 JSX

JSX (JavaScript_G XML) è un'estensione di sintassi di JavaScript_G che consente di scrivere codice HTML all'interno di file JavaScript_G. Viene utilizzato per definire la struttura delle interfacce utente _G all'interno delle applicazioni React_G.

Versione_G: 18.0.0.

Contesto di utilizzo:

- Definizione della struttura dei componenti web.

4.2.2 CSS

CSS (Cascading Style Sheets) è un linguaggio utilizzato per definire lo stile e la presentazione delle pagine web. Viene utilizzato per definire la formattazione, il layout e il design delle pagine web.

Versione_G: 3.0.

Contesto di utilizzo:

- Definizione stile e layout componenti web.

Librerie e framework

- **Tailwind CSS**

- Framework CSS utilizzato per lo sviluppo di interfacce utente_G web. Offre una serie di classi predefinite per la definizione dello stile degli elementi.
- **Versione_G:** 3.4.1.
- **Link:** <https://tailwindcss.com/> (ultimo accesso 27/03/2024)

4.2.3 TypeScript_G

TypeScript_G è un superset di JavaScript_G che aggiunge tipizzazione statica al linguaggio, offrendo maggiore struttura al codice. Questo permette di rilevare errori di programmazione in fase di sviluppo, riducendo il rischio di bug_G e semplificando la manutenzione del codice.

Versione_G: 5.3.3.

Contesto di utilizzo:

- Definizione tipi e interfacce per i dati;
- Implementazione dei componenti React_G ;
- Codifica lato front-end e back-end;
- Implementazione ambiente 3D.

Librerie e framework

- **Next.js_G**
 - Framework di sviluppo web front-end basato su React_G e utilizzato per la creazione di applicazioni web. Offre funzionalità avanzate realizzazione di API_G , gestione del routing e Server Action.
 - **Versione_G:** 14.1.0.
 - **Link:** <https://nextjs.org/> (ultimo accesso 27/03/2024)
- **React_G**
 - Libreria JavaScript_G utilizzata per la creazione di interfacce utente_G dinamiche, reattive e stateful. Si basa sul concetto di “components”, ovvero blocchi di codice autonomi che gestiscono la propria logica e rendering.
 - **Versione_G:** 18.0.0.
 - **Link:** <https://reactjs.org/> (ultimo accesso 27/03/2024)
- **Node.js_G**
 - Runtime system orientato agli eventi per l'esecuzione di codice JavaScript_G estendibile tramite moduli. Viene utilizzato per eseguire il codice JavaScript_G lato server.
 - **Versione_G:** 20.11.0.
 - **Link:** <https://nodejs.org/> (ultimo accesso 27/03/2024)
- **Shadcn/ui**
 - Raccolta di componenti React_G personalizzati per la creazione di interfacce utente_G. Offre una serie di componenti pronti all'uso per la realizzazione di interfacce grafiche.
 - **Versione_G:** 0.8.0.
 - **Link:** <https://ui.shadcn.com/> (ultimo accesso 27/03/2024)

Librerie e framework ambiente 3D

- **Three.js_G**
 - Libreria JavaScript_G utilizzata per creare e visualizzare grafica computerizzata 3D animata in un browser_G Web utilizzando WebGL_G. Offre funzionalità avanzate per la creazione di ambienti 3D interattivi.
 - **Versione_G:** 0.161.2.
 - **Link:** <https://threejs.org/> (ultimo accesso 27/03/2024)
- **@react_G-three/fiber**

- Libreria open-source che facilita l'integrazione di Three.js_G all'interno di applicazioni React_G. Offre funzionalità avanzate per la creazione di grafica 3D animata.
 - **Versione_G**: 8.15.16.
 - **Link**: <https://docs.pmnd.rs/react-three-fiber/getting-started/introduction> (ultimo accesso 27/03/2024)
- **@react_G-three/drei**
 - Libreria che fornisce componenti e utilità per semplificare lo sviluppo di applicazioni in 3D utilizzando React_G e Three.js_G. Offre funzionalità avanzate per la creazione di ambienti 3D interattivi.
 - **Versione_G**: 9.97.6.
 - **Link**: <https://www.npmjs.com/package/@react-three/drei> (ultimo accesso 27/03/2024)

4.3 Tecnologie per la validazione dei dati

4.3.1 Zod

Zod è una libreria di validazione dei dati per TypeScript_G. Viene utilizzata per definire schemi di validazione dei dati e garantire che i dati ricevuti siano conformi a tali schemi.

Versione_G: 3.22.4.

Contesto di utilizzo:

- Validazione dei dati inseriti dall'utente_G.

4.4 Tecnologie per la persistenza dei dati

4.4.1 SQL_G

SQL_G (Structured Query Language_G) è un linguaggio di programmazione utilizzato per la gestione dei database_G relazionali. Viene utilizzato per la creazione, la modifica e la gestione dei dati all'interno del database_G. Nel contesto del capitolato_G, le operazioni svolte sul database_G sono esclusivamente di tipo interrogativo.

Contesto di utilizzo:

- Interrogazione del database_G relazionale.

4.4.2 JSON_G

JSON_G (JavaScript Object Notation_G) è un formato di scambio dati leggero e indipendente dal linguaggio. Viene utilizzato per la trasmissione e lo scambio di dati tra client e server. Utilizzato principalmente nelle comunicazioni nella rete, trova naturale utilizzo nello sviluppo di una web app.

Contesto di utilizzo:

- Risultato delle interrogazioni al database_G;
- Risultato delle chiamate API_G.

4.4.3 PostgreSQL_G

PostgreSQL_G è un sistema di gestione di database_G relazionali. Viene utilizzato per la memorizzazione e la gestione dei dati relativi al software “WMS3”.

Versione_G: 16.2.

Contesto di utilizzo:

- Memorizzazione e gestione dei dati relativi ai bin_G, ai prodotti e alle zone del magazzino_G.

4.5 Tecnologie per il testing_G

4.5.1 Jest_G

Jest_G è un framework di testing_G per JavaScript_G e TypeScript_G. Viene utilizzato principalmente per lo unit e l'integration testing_G, offrendo funzionalità avanzate come la parallelizzazione dei test e il mocking delle dipendenze.

Versione_G: 29.7.0.

Contesto di utilizzo:

- Implementazione della suite di unit testing_G;
- Implementazione della suite di integration testing_G.

4.6 Tecnologie per il deployment

4.6.1 Docker_G

Docker_G è un software utilizzato per il processo di deployment di applicazioni software. Permette di eseguire processi informatici in ambienti isolati chiamati container, garantendo la portabilità e la scalabilità delle applicazioni.

Versione_G: 24.0.7.

Contesto di utilizzo:

- Deployment del software “WMS3” mediante container Docker_G;
- Isolamento dell'ambiente di sviluppo.

Immagini Docker_G utilizzate

- **PostgreSQL_G:** container per il database_G relazionale;
 - Immagine: postgres_G:16.2.
- **Web:** container per l'applicazione web;
 - Immagine: node_G:20-alpine.



4.6.2 Docker_G-compose

Docker_G-compose è uno strumento per la definizione e l'esecuzione di applicazioni multi-container. Viene utilizzato per gestire l'orchestrazione dei container Docker_G e semplificare il processo di deployment.

Versione_G: 2.23.3.

Contesto di utilizzo:

- Gestione dell'orchestrazione dei container Docker_G utilizzati.

5 Architettura di sistema

5.1 Architettura di implementazione

Il software WMS3 al fine di perseguire manutenibilità, flessibilità e scalabilità, adotta ed implementa un'architettura “layered”, nota anche come “Multi-tier architecture”.

I layer definiti sono “closed”, ovvero una richiesta si sposta esclusivamente da un livello superiore a quello immediatamente adiacente.

Tale architettura permette di individuare e suddividere la logica del software in 3 principali aspetti, definiti tier (separation of concerns), quali:

- **Persistence layer:** gestisce l'accesso al database_G e fornisce gli strumenti dedicati alla lettura dei dati al suo interno. I dati letti vengono processati al fine di poter creare gli elementi del Business layer;
- **Business layer:** si occupa di elaborare i dati ricevuti dal layer di persistenza e applicare le regole di business definite. È responsabile di implementare la logica dell'applicazione in modo indipendente dalle tecnologie di persistenza e di presentazione utilizzate;
- **Presentation layer:** permette di trasformare i dati elaborati dal Business layer e le informazioni in una forma comprensibile e accessibile agli utenti finali. Questo include la creazione di interfacce utente_G grafiche e visualizzazioni 3D degli elementi di interesse.

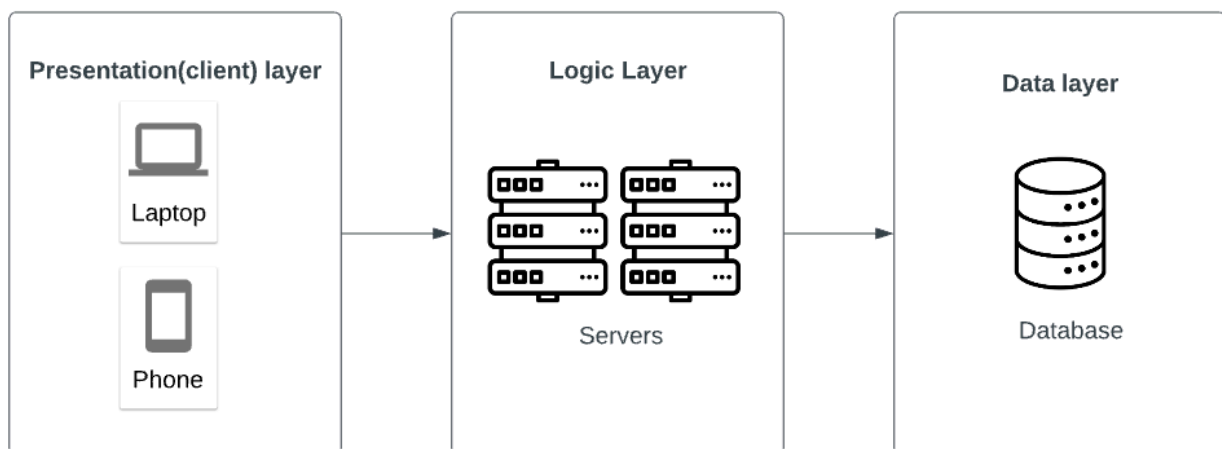


Figura 1: Layered architecture.

5.1.1 Vantaggi

- Ogni livello dell'architettura crea un livello di astrazione che permette di perseguire *separation of concerns* e di rendere il software più manutenibile e scalabile;
- Semplicità di implementazione in termini di costi e tempo;
- Facilità di test e debug;

5.1.2 Svantaggi

- Cambiamenti consistenti possono richiedere modifiche in layer diversi.

5.2 Design pattern utilizzati

5.2.1 Data Mapper

Il design pattern Data Mapper viene utilizzato per interpretare i dati letti del database_G mantenendo separate la logica di business dal layer di persistenza. Le classi relative a questo pattern fungono da intermediari tra l'applicazione e la sorgente dati e sono responsabili della conversione delle strutture dati atte alla persistenza (ottenute in formato JSON_G a seguito di query al database_G) in oggetti del dominio dell'applicazione.

5.2.2 Repository_G

Il design pattern Repository_G viene implementato per separare la logica di business dalla logica di accesso ai dati. Le classi relative a questo pattern eseguono operazioni di lettura, aumentando l'astrazione dei dettagli specifici della persistenza dei dati e permettendo all'applicazione di interagirvi in modo indipendente dal tipo di archivio sottostante.

5.2.3 Provider

Il design pattern Provider viene applicato nel contesto tecnologico del progetto, soprattutto rispetto all'utilizzo di React_G, vengono sfruttate delle Context API_G per permettere di gestire e trasferire i dati attraverso l'albero dei componenti in modo strutturato, evitando il "prop drilling", ovvero l'effetto che si verifica nei casi in cui è necessario trasportare i dati attraverso più livelli di componenti, anche se alcuni di essi non ne necessitano.

5.2.4 Strategy

Il design pattern Strategy consente di definire una famiglia di algoritmi, incapsularli in classi separate e renderli intercambiabili. In questo modo è possibile applicare l'algoritmo appropriato senza dover conoscere i dettagli implementativi.

È stato implementato nella gestione di diversi algoritmi di creazione del piano, degli algoritmi di ricerca dei prodotti e degli algoritmi di ricerca delle zone.

5.2.5 Factory

Il design pattern Factory permette, definendo un'interfaccia comune, la creazione di oggetti senza specificarne esplicitamente le classi esatte, lasciando alle sottoclassi la decisione su quale istanziare.

È stato adottato per separare l'implementazione dalla creazione degli oggetti relativi agli algoritmi di ricerca dei prodotti e delle zone.

5.3 Classi e Componenti

Ciascun layer possiede il suo indipendente sistema di classi e componenti e prevede metodi per comunicare con i layer adiacenti.

5.3.1 Persistence layer

Mediante le Server Action offerte da Next.js_G, vengono eseguite delle query SQL_G atte alla lettura dei dati utili all'applicazione da un database_G esterno. Esse sono implementate e rese disponibili in file separati, organizzati nell'omonima cartella "Server Action", contenente:

- **getAllBins**: ritorna le informazioni di tutti i bin_G lette dal database_G;

- **getBinById**: dato un codice identificativo univoco, ritorna le informazioni relative al `binG` corrispondente lette dal database_G;
- **getAllCategories**: ritorna le informazioni di tutte le categorie di prodotti lette dal database_G;
- **getAllProducts**: ritorna le informazioni di tutti i prodotti lette dal database_G;
- **getProductById**: dato un codice identificativo univoco, ritorna le informazioni relative al prodotto corrispondente lette dal database_G;
- **SVGSanitize**: dato un path ad un file SVG_G caricato, ritorna il contenuto del relativo file SVG_G sanificato, ovvero normalizzato e reso sicuro, prevenendo attacchi XSS;
- **readSavedSVG**: ritorna il contenuto del file SVG_G salvato su server;
- **saveSVG**: dato il contenuto di un file SVG_G, esso viene salvato come *saved.svg_G*;
- **getAllZones**: ritorna le informazioni di tutte le zone lette dal database_G;
- **getBinsByZoneId**: dato un codice identificativo univoco, ritorna le informazioni relative a tutti i `binG` contenuti nella zona corrispondente lette dal database_G;
- **getZoneById**: dato un codice identificativo univoco, ritorna le informazioni relative alla zona corrispondente lette dal database_G.

Al fine di agevolare la divisione tra il Persistence layer ed il Business layer, viene utilizzato il pattern Repository_G (Sezione 5.2.2) mediante classi che implementano l'interfaccia `dataRepositoryInterface`, quali:

- **binRepository**: è responsabile dell'ottenimento dei dati relativi agli oggetti `BinG`;
- **productRepository**: è responsabile dell'ottenimento dei dati relativi agli oggetti `Product`;
- **zoneRepository**: è responsabile dell'ottenimento dei dati relativi agli oggetti `Zone`.

Il pattern Repository_G impiega in maniera consequenziale le classi correlate del pattern Data Mapper (Sezione 5.2.1), le quali implementano l'interfaccia `DataMapperInterface`. Di seguito sono elencate le classi specifiche:

- **binMapper**: è responsabile della creazione di oggetti `BinG`;
- **productMapper**: è responsabile della creazione di oggetti `Product`;
- **zoneMapper**: è responsabile della creazione di oggetti `Zone`.

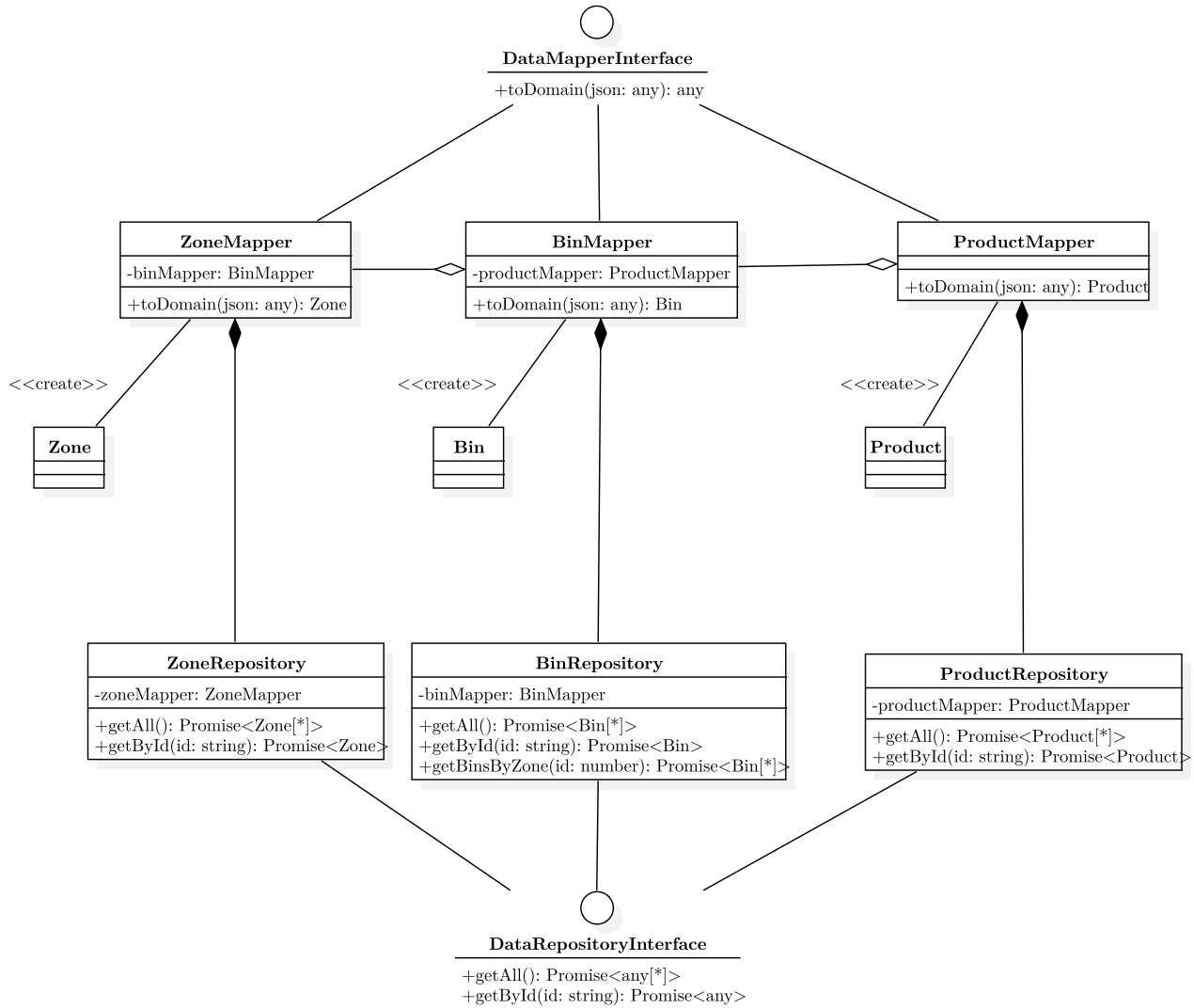


Figura 2: Diagramma delle classi del layer di persistenza

Nel diagramma delle classi del layer di persistenza fornito, le classi **Zone**, **Bin_G** e **Product** sono rappresentate senza gli attributi e i metodi per garantire una maggiore chiarezza grafica. Tali informazioni sono rappresentate dettagliatamente nel diagramma delle classi del layer di business.

5.3.2 Business layer

Le classi che vengono utilizzate per rappresentare il modello dell'applicativo sono:

- **Bin_G**:

rappresenta un elemento bin_G, ovvero uno spazio definito in grado di contenere un prodotto. I suoi attributi sono:

- **id**: stringa di massimo dieci caratteri che rappresenta il codice identificativo univoco del bin_G. Essa è composta da:

idZona_letteraColonna_numeroLivello

La lettera corrispondente alla colonna fa riferimento ad una mappatura per cui “A” equivale alla colonna zero e viene incrementata seguendo i caratteri dell’alfabeto inglese con l’aumentare del numero della colonna.

Dopo la lettera “Z” viene utilizzato “AA” proseguendo con la logica descritta;

- **level**: intero che rappresenta il numero del livello di appartenenza;
- **column**: intero che rappresenta il numero della colonna di appartenenza;
- **height**: numero in virgola mobile che rappresenta l’altezza del bin_G;
- **length**: numero in virgola mobile che rappresenta la profondità del bin_G;
- **width**: numero in virgola mobile che rappresenta la larghezza del bin_G;
- **product**: riferimento al prodotto contenuto nel bin_G. Può essere null.
- **state**: enum `BinState` che identifica lo stato di un bin_G contestualmente alla richiesta di spostamento dei prodotti.

Può assumere i valori:

- **Idle**: valore di default, dichiara che il bin_G non è coinvolto in richieste di spostamento di prodotti;
- **ProductIncoming**: dichiara che il bin_G è coinvolto in una richiesta di spostamento di un prodotto, il quale deve essere immesso al suo interno;
- **ProductOutgoing**: dichiara che il bin_G è coinvolto in una richiesta di spostamento del prodotto al suo interno, il quale deve essere prelevato.

Per ogni attributo è presente il corrispondente metodo `get`.

Sono presenti i metodi `set` per gli attributi `id`, `product` e `state`.

Inoltre è previsto il metodo `clearProduct` che permette di assegnare il valore `null` all’attributo che riferisce il prodotto contenuto nel bin_G.

• **Zone**:

rappresenta un elemento zona, può essere interpretata come uno scaffale_G oppure, nel caso abbia un solo livello, come una zona del piano definita per contenere bin_G. I suoi attributi sono:

- **id**: intero che rappresenta il codice identificativo univoco della zona;
- **xcoordinate**: numero in virgola mobile che rappresenta la coordinata X di posizione nel piano;
- **ycoordinate**: numero in virgola mobile che rappresenta la coordinata Y di posizione nel piano;
- **height**: numero in virgola mobile che rappresenta l’altezza della zona;
- **length**: numero in virgola mobile che rappresenta la profondità della zona;
- **width**: numero in virgola mobile che rappresenta la larghezza della zona;
- **bins**: lista di elementi `BinG` contenuti nella zona;
- **orientation**: booleano che identifica l’orientamento (perpendicolare o parallelo) della zona rispetto all’asse delle ascisse del piano.

Per ogni attributo è presente il corrispondente metodo `get`.

Sono disponibili i metodi `set` per gli attributi `xcoordinate` e `ycoordinate`.

Sono inoltre forniti i metodi:

- **getBin**: dato un codice identificativo univoco, ritorna l'elemento Bin_G corrispondente presente in bins , oppure null se non presente;
- **getLevels**: ritorna una lista contenente le liste di bin_G che rappresentano i livelli della zona;
- **getColumns**: ritorna una lista contenente le liste di bin_G che rappresentano le colonne della zona;
- **getMaxUsedLevel**: ritorna il numero dell'ultimo livello della zona con almeno un bin_G contenente un prodotto;
- **getMaxUsedColumn**: ritorna il numero dell'ultima colonna della zona con almeno un bin_G contenente un prodotto.

- **Product:**

rappresenta un prodotto, i suoi attributi sono:

- **id**: intero che rappresenta il codice identificativo univoco del prodotto;
- **name**: stringa che rappresenta il nome del prodotto;
- **weight**: numero in virgola mobile che rappresenta il peso del prodotto;
- **height**: numero in virgola mobile che rappresenta l'altezza del prodotto;
- **length**: numero in virgola mobile che rappresenta la profondità del prodotto;
- **width**: numero in virgola mobile che rappresenta la larghezza del prodotto;
- **categories**: lista di stringhe che rappresentano le categorie del prodotto.

Per ogni attributo è presente il corrispondente metodo get.

- **Order:**

rappresenta un ordine di spostamento di un prodotto tra due bin_G . I suoi attributi sono:

- **id**: intero che rappresenta il codice identificativo univoco della richiesta;
- **startPoint**: riferimento al bin_G iniziale;
- **endPoint**: riferimento al bin_G finale;
- **product**: riferimento al prodotto da spostare.

Per ogni attributo è presente il corrispondente metodo get.

- **SVG_G:**

rappresenta un file SVG_G utilizzato per la configurazione dell'ambiente di lavoro mediante file. I suoi attributi sono:

- **length**: numero in virgola mobile che rappresenta la lunghezza dell'immagine rappresentata dal file;

- **width**: numero in virgola mobile che rappresenta la larghezza dell'immagine rappresentata dal file;
- **svg_G**: stringa contenente il path del file.

Per ogni attributo è presente il corrispondente metodo get.

- **Floor**:

rappresenta il piano dell'ambiente 3D, i suoi attributi sono:

- **length**: numero in virgola mobile che rappresenta la profondità del piano;
- **width**: numero in virgola mobile che rappresenta la larghezza del piano;
- **SVG_G**: elemento **SVG_G** utilizzato per la configurazione dell'ambiente di lavoro. Nel caso in cui non sia stato identificato nessun file **SVG_G**, la variabile è **null**.

Per ogni attributo è presente il corrispondente metodo get e set.

Inoltre è presente il metodo **clone** utile per creare una copia dell'oggetto invocante.

Potendo generare l'oggetto **Floor** con modalità diverse a seconda della presenza del file **SVG_G**, la sua creazione è gestita tramite il design pattern Strategy (Sezione 5.2.4) e le relative classi che implementano l'interfaccia **FloorStrategy**:

- **StandardFloorStrategy**: rappresenta la creazione di un elemento **Floor** senza file **SVG_G**;
- **CustomFloorStrategy**: rappresenta la creazione di un elemento **Floor** con file **SVG_G**;

Inoltre è presente la classe:

- **FloorStrategyContext**: utilizza un **FloorStrategy** per generare un elemento **Floor**.

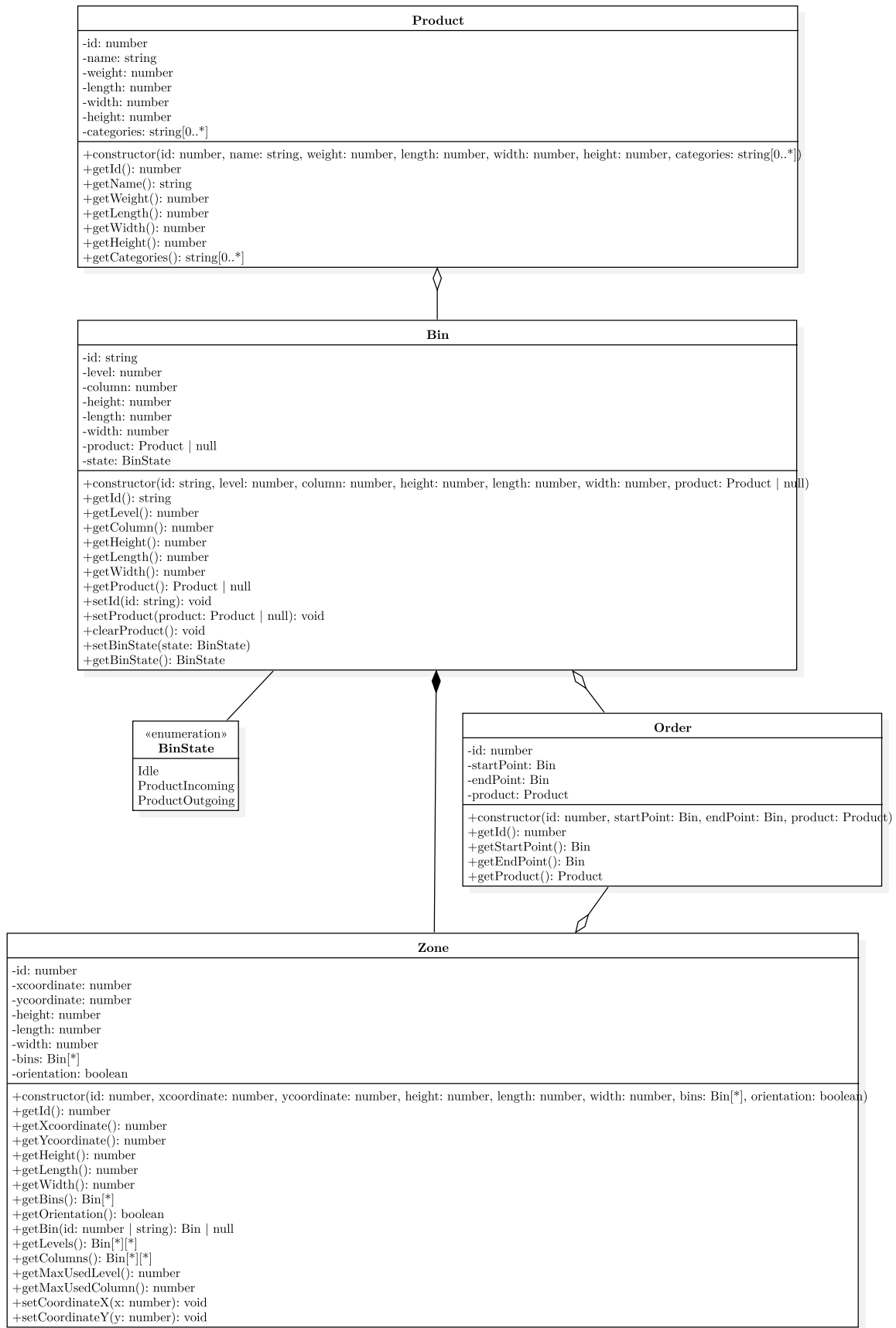


Figura 3: Diagramma delle classi del layer di business

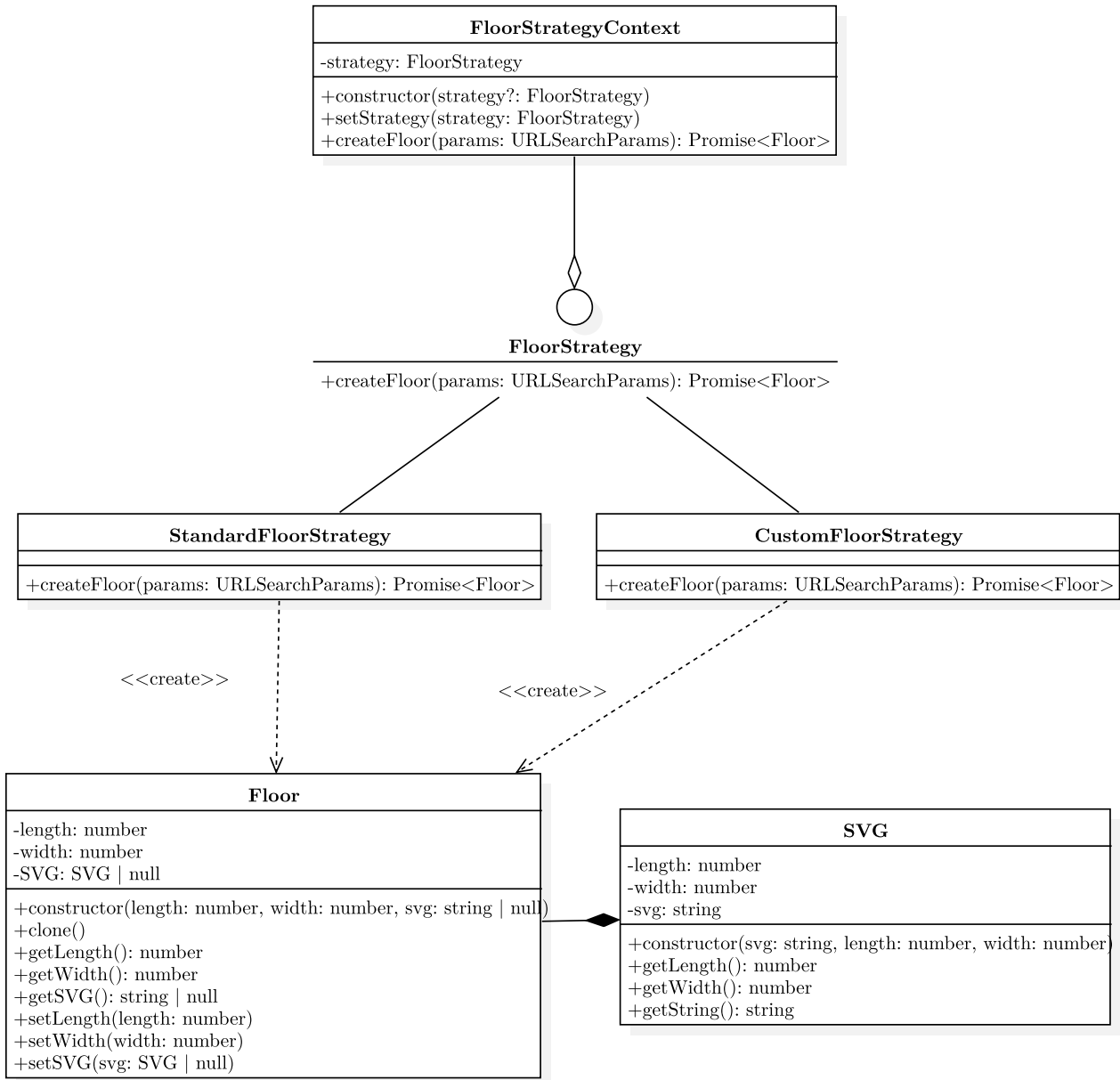


Figura 4: Diagramma delle classi Floor del layer di business

Per gestire la possibilità di cercare specifici prodotti e specifiche zone, sono state realizzate delle classi che permettono l'implementazione dei design pattern Strategy (Sezione 5.2.4) e Factory (Sezione 5.2.5).

Le classi relative al design pattern Strategy implementano l'interfaccia **SearchStrategy** e sono:

- **ProductSearchStrategy**: permette la ricerca di prodotti in base al loro codice identificativo, nome o categoria;
- **ZoneSearchStrategy**: permette la ricerca di zone in base al loro codice identificativo.

La classe relativa al design pattern Factory è:

- **SearchStrategyFactory**: permette l'adozione del corretto algoritmo di ricerca in funzione del tipo di oggetto fornito.

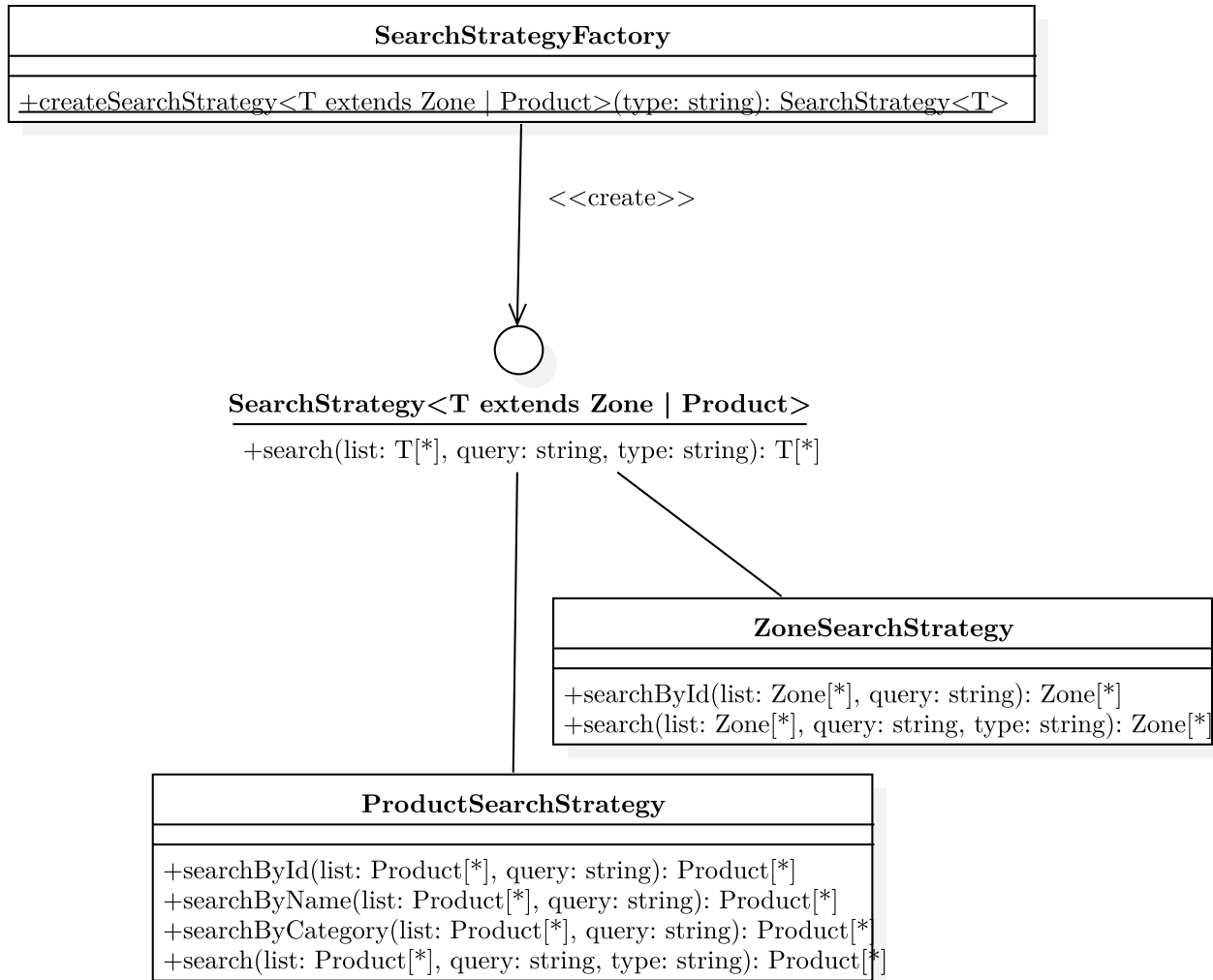


Figura 5: Diagramma delle classi Search del layer di business

In aggiunta alle classi, per aderire all'adozione del pattern Provider (Sezione 5.2.3), sono presenti i componenti:

- **warehouseProvider**: fornisce un provider per gestire dati relativi alle variabili di contesto dell'ambiente grafico;
- **ElementDetailsProvider**: fornisce un provider per gestire dati relativi ai componenti da visualizzare in un pannello dedicato;
- **formContextProvider**: fornisce un provider per gestire dati relativi allo stato di processing di un form;
- **binsProvider**: fornisce un provider per gestire dati relativi agli oggetti `Bin`;
- **floorProvider**: fornisce un provider per gestire dati relativi agli oggetti `Floor`;
- **ordersProvider**: fornisce un provider per gestire dati relativi agli oggetti `Order`;
- **productsProvider**: fornisce un provider per gestire dati relativi agli oggetti `Product`;

- **zonesProvider**: fornisce un provider per gestire dati relativi agli oggetti **Zone**.

5.3.3 Presentation layer

5.3.3.1 UI

L'interfaccia utente_G è realizzata mediante elementi importati da Shadcn-UI e componenti personalizzati.

I componenti realizzati sono i seguenti:

- **Form di creazione**:
 - **creationForm**: componente dedicato alla selezione della modalità di creazione dell'ambiente di lavoro;
 - **dropFileArea**: componente dedicato al caricamento di un file SVG_G;
 - **manualCreationFrame**: componente dedicato alla creazione manuale dell'ambiente di lavoro;
 - **svgCreationFrame**: componente dedicato alla definizione dei parametri di creazione dell'ambiente di lavoro mediante file SVG_G;
 - **zodScheme**: schema dedicato alla validazione dei dati di creazione dell'ambiente di lavoro.
- **Componenti relativi ai bin_G**:
 - **binItemDetails**: componente dedicato alla visualizzazione delle informazioni dettagliate di un bin_G.
- **Componenti relativi agli ordini**:
 - **orderItem**: componente dedicato alla visualizzazione di un ordine;
 - **ordersPanel**: componente dedicato alla visualizzazione ordinata di tutti gli ordini dall'ultimo effettuato al meno recente. Ogni ordine è rappresentato da un **OrderItem**.
- **Componenti relativi ai prodotti**:
 - **productItem**: componente dedicato alla visualizzazione di un prodotto;
 - **productItemDetails**: componente dedicato alla visualizzazione delle informazioni dettagliate di un prodotto;
 - **productsPanel**: componente dedicato alla visualizzazione di tutti i prodotti.
- **Componenti relativi alle impostazioni**:
 - **floorDimensionsItem**: componente dedicato alla visualizzazione delle dimensioni del piano;
 - **restoreItem**: componente dedicato al comando atto al ripristino o alla reimpostazione dell'ambiente di lavoro. Il ripristino permette di caricare nuovamente, con i parametri precedentemente specificati, l'ambiente di lavoro, mentre è possibile eseguirne una riconfigurazione mediante la reimpostazione;
 - **settingsPanel**: componente dedicato al pannello delle impostazioni contenente la versione_G del software e i componenti **floorDimensionsItem** e **restoreItem**;
 - **zodDimensionScheme**: schema dedicato alla validazione dei dati dimensionali per la modifica del piano.

- **Zone:**
 - **bin_G_columns:** componente dedicato alla visualizzazione delle colonne di bin_G di una zona (necessario al componente **ZoneItemDetails** utilizzato);
 - **levelItem:** componente dedicato alla visualizzazione di un livello della zona durante il processo di creazione/modifica dello stesso;
 - **zoneCreationFrame:** componente dedicato alla creazione di una zona;
 - **zoneItem:** componente dedicato alla visualizzazione di una zona;
 - **zoneItemDetails:** componente dedicato alla visualizzazione delle informazioni dettagliate di una zona;
 - **zonePanel:** componente dedicato alla visualizzazione di tutte le zone;
 - **zoneZodSchemes:** schemi dedicati alla validazione dei dati necessari alla creazione di una zona, sia durante la configurazione manuale, che mediante inserimento di un file SVG_G.
- **panel:** componente generico utilizzato per la visualizzazione e l'organizzazione dei componenti al suo interno.

5.3.3.2 Three.js_G

L'ambiente tridimensionale è realizzato mediante i componenti:

- **Floor:** elemento che rappresenta il piano dell'ambiente di lavoro;
- **Bin3D:** elemento che rappresenta un bin_G nell'ambiente di lavoro;
- **Zone3D:** elemento che rappresenta uno scaffale_G o un'area del piano dedicata al contenimento di bin_G nell'ambiente di lavoro;
- **Warehouse:** elemento che contiene la logica principale della visualizzazione dell'ambiente grafico ed il canvas con gli elementi 3D.

5.4 Database_G

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

Esso è così composto:

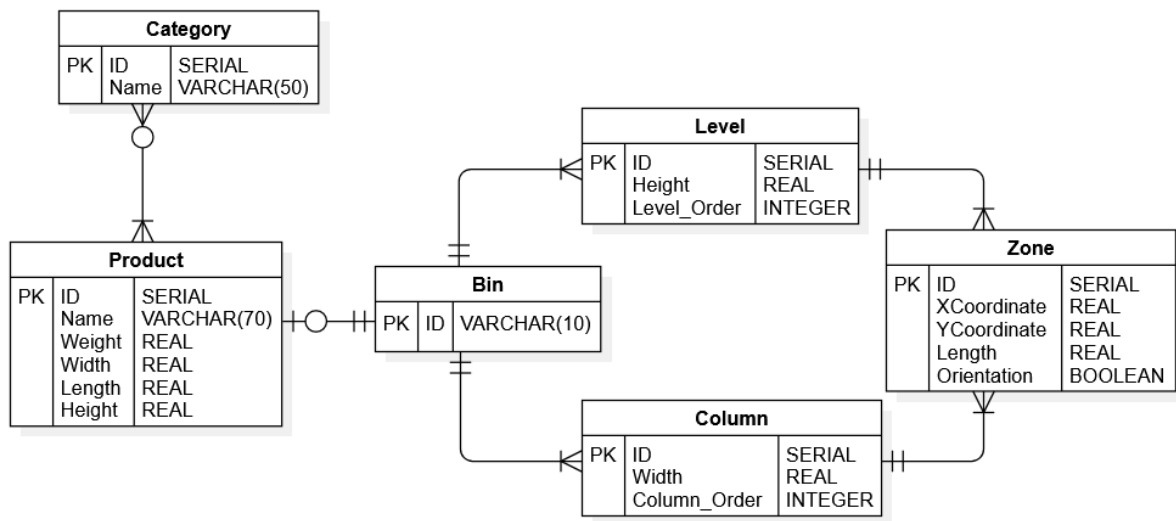


Figura 6: Schema ER del Database_G.

5.4.1 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:

$$\text{ID zona} + _ + \text{Column_Order} + _ + \text{Level_Order}$$
- **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.

5.4.2 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.

5.4.3 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.

6 Requisiti soddisfatti

Di seguito vengono riportati i requisiti funzionali e di qualità soddisfatti dall'applicazione.

Per una visione più completa sui requisiti si rimanda al documento *Analisi dei Requisiti_G v2.0.1*.

6.1 Requisiti funzionali soddisfatti

Codice	Descrizione	Stato
FM-1	L'utente _G deve poter configurare un ambiente 3D all'avvio della sessione d'uso	Soddisfatto
FD-2	L'utente _G deve avere la possibilità di scegliere tra diverse modalità di configurazione del magazzino _G	Soddisfatto
FM-3	Deve essere fornita una modalità di configurazione dell'ambiente 3D per la rappresentazione di un magazzino _G con pianta rettangolare	Soddisfatto
FD-4	Deve essere fornita una modalità di configurazione dell'ambiente 3D per la rappresentazione di un magazzino _G con planimetria importata da file SVG _G	Soddisfatto
FM-5	L'utente _G deve poter indicare la larghezza della planimetria rettangolare	Soddisfatto
FM-6	L'utente _G deve poter indicare la lunghezza della planimetria rettangolare	Soddisfatto
FM-7	L'utente _G deve visualizzare un errore se la larghezza indicata non è positiva (≤ 0)	Soddisfatto
FM-8	L'utente _G deve visualizzare un errore se la lunghezza indicata non è positiva (≤ 0)	Soddisfatto
FD-9	L'utente _G deve poter caricare un file SVG _G da usare come planimetria qualora abbia scelto di definire la planimetria a partire da un file SVG _G	Soddisfatto
FD-10	Il file SVG _G deve essere sanificato prima dell'importazione	Soddisfatto
FD-11	Il file SVG _G deve contenere almeno un elemento grafico tra path, rect, circle, ellipse, line, polyline, polygon, text, g per essere considerato valido	Non soddisfatto
FD-12	L'utente _G deve ricevere un messaggio di errore qualora avesse caricato un file SVG _G privo di elementi grafici (path, rect, circle, ellipse, line, polyline, polygon, text, g)	Non soddisfatto
FD-13	Il file SVG _G deve essere validato	Soddisfatto
FD-14	L'utente _G deve ricevere un messaggio di errore qualora avesse caricato un file SVG _G non valido o corrotto	Soddisfatto
FD-15	L'utente _G che abbia scelto la modalità di configurazione a partire da un file SVG _G , deve poter indicare il solo lato maggiore del magazzino _G per configurare la planimetria	Soddisfatto

FD-16	Il sistema deve determinare il valore del lato minore a partire dal rapporto di aspetto del file SVG _G e dai dati forniti dall'utente _G	Soddisfatto
FD-17	L'utente _G deve visualizzare un errore se il valore indicato come lato maggiore non è positivo (≤ 0)	Soddisfatto
FM-18	L'utente _G deve poter riconfigurare la planimetria dell'ambiente 3D corrente	Soddisfatto
FM-19	A seguito della riconfigurazione della planimetria, le modifiche a zone, bin _G e prodotti non devono subire variazioni	Soddisfatto
FO-20	L'utente _G deve poter visualizzare un'anteprima delle modifiche alla planimetria prima di confermare l'operazione	Soddisfatto
FM-21	L'utente _G deve poter ridefinire la larghezza dell'ambiente 3D corrente	Soddisfatto
FM-22	L'utente _G deve poter ridefinire la lunghezza dell'ambiente 3D corrente	Soddisfatto
FD-23	L'utente _G che abbia configurato un ambiente 3D a partire da file SVG _G non può definire un valore di lunghezza inferiore a quello corrente	Soddisfatto
FD-24	L'utente _G che abbia configurato un ambiente 3D a partire da file SVG _G non può definire un valore di larghezza inferiore a quello corrente	Soddisfatto
FM-25	L'utente _G deve visualizzare un errore se il nuovo valore di larghezza indicato non è positivo (≤ 0)	Soddisfatto
FM-26	L'utente _G deve visualizzare un errore se il nuovo valore di lunghezza indicato non è positivo (≤ 0)	Soddisfatto
FD-27	L'utente _G deve poter disporre di una griglia di aggancio come aiuto al posizionamento delle zone nell'ambiente 3D	Soddisfatto
FD-28	Il passo della griglia deve essere configurabile	Soddisfatto
FD-29	L'utente _G deve poter disattivare la griglia di posizionamento	Soddisfatto
FD-30	La griglia deve essere configurabile durante le normali operazioni sull'ambiente 3D, non esclusivamente durante la configurazione dell'ambiente	Soddisfatto
FD-31	Se il passo di griglia non è nullo, il collocamento delle zone deve agganciarsi ad essa	Soddisfatto
FD-32	L'utente _G deve poter importare le zone da un database _G	Soddisfatto
FD-33	L'utente _G deve poter importare le zone da un database _G durante la fase di configurazione dell'ambiente 3D	Soddisfatto
FD-34	Le zone importate devono essere collocate automaticamente nell'ambiente 3D, nella posizione descritta dal database _G	Soddisfatto

FD-35	I bin _G delle zone devono essere importati contestualmente all'importazione delle zone	Soddisfatto
FD-36	L'importazione delle zone può avvenire solo se l'utente _G ha configurato un ambiente 3D a partire da file SVG _G	Soddisfatto
FD-37	L'utente _G deve visualizzare un messaggio di errore nel caso l'importazione non dovesse andare a buon fine	Soddisfatto
FD-38	L'utente _G deve poter importare i prodotti da database _G	Soddisfatto
FD-39	Quando l'utente _G importa zone e prodotti da un database _G , i prodotti devono essere collocati nei rispettivi bin _G di appartenenza	Soddisfatto
FM-40	L'utente _G deve poter alterare il proprio POV sull'ambiente 3D	Soddisfatto
FM-41	L'utente _G deve poter ruotare il proprio POV attorno all'asse longitudinale	Soddisfatto
FM-42	L'utente _G deve poter traslare il proprio POV lungo l'asse orizzontale	Soddisfatto
FM-43	L'utente _G deve poter effettuare zoom _G -in	Soddisfatto
FM-44	L'utente _G deve poter effettuare zoom _G -out	Soddisfatto
FM-45	L'utente _G deve poter configurare un nuovo ambiente 3D	Soddisfatto
FM-46	La configurazione di un nuovo ambiente 3D deve cancellare tutti i dati della sessione corrente	Soddisfatto
FM-47	Il sistema non deve offrire la persistenza dei dati importati	Soddisfatto
FM-48	Il sistema non deve offrire la persistenza dei dati generati durante la sessione corrente	Soddisfatto
FM-49	La lista delle movimentazioni di prodotti richieste durante la sessione corrente deve essere scartata contestualmente alla riconfigurazione dell'ambiente 3D	Soddisfatto
FM-50	Le aggiunte alle zone devono essere scartate contestualmente alla riconfigurazione dell'ambiente 3D	Soddisfatto
FM-51	Le modifiche alle zone devono essere scartate contestualmente alla riconfigurazione dell'ambiente 3D	Soddisfatto
FM-52	Le cancellazioni delle zone devono essere scartate contestualmente alla riconfigurazione dell'ambiente 3D	Soddisfatto
FM-53	La configurazione della planimetria deve essere scartata contestualmente alla riconfigurazione dell'ambiente 3D	Soddisfatto
FM-54	Le informazioni sui prodotti devono essere scartate contestualmente alla riconfigurazione dell'ambiente 3D	Soddisfatto
FM-55	L'utente _G deve poter creare nuove zone	Soddisfatto
FD-56	L'utente _G deve poter indicare una sequenza numerica come codice identificativo delle nuove zone create	Soddisfatto

FD-57	L'utente _G deve visualizzare un errore qualora avesse indicato un codice identificativo già in uso	Soddisfatto
FM-58	L'utente _G deve indicare la lunghezza della nuova zona da creare	Soddisfatto
FM-59	L'utente _G deve visualizzare un errore se la lunghezza indicata non è positiva (≤ 0)	Soddisfatto
FD-60	L'utente _G deve poter scegliere tra "NS" e "WE" come orientamento della zona da creare	Soddisfatto
FM-61	L'utente _G deve indicare il numero di colonne della nuova zona	Soddisfatto
FM-62	Una zona deve contenere almeno 1 colonna	Soddisfatto
FD-63	L'identificazione delle colonne deve avvenire tramite lettere crescenti in senso lessicografico a partire da "A"	Soddisfatto
FM-64	L'utente _G deve visualizzare un errore se il numero di colonne della nuova zona non è almeno pari a 1	Soddisfatto
FD-65	L'utente _G deve poter personalizzare la larghezza delle colonne della nuova zona	Soddisfatto
FD-66	L'utente _G deve poter suddividere la larghezza della nuova zona in colonne di equa larghezza	Soddisfatto
FD-67	L'utente _G deve indicare la larghezza complessiva della nuova zona, qualora avesse richiesto la suddivisione della stessa in colonne di equa larghezza	Soddisfatto
FD-68	L'utente _G deve poter suddividere la larghezza della nuova zona in colonne di larghezza specifica	Soddisfatto
FD-69	L'utente _G deve poter indicare la larghezza di ciascuna colonna, qualora avesse richiesto la suddivisione della nuova zona in colonne di larghezza specifica	Soddisfatto
FD-70	Il sistema deve determinare il valore della larghezza della zona dalla somma delle larghezze delle singole colonne	Soddisfatto
FD-71	L'utente _G deve visualizzare un errore se la larghezza indicata per la singola colonna non è positiva (≤ 0)	Soddisfatto
FM-72	L'utente _G deve poter personalizzare il numero di livelli della nuova zona da creare	Soddisfatto
FM-73	L'utente _G deve visualizzare un errore se il numero di livelli della nuova zona non è almeno pari a 1	Soddisfatto
FM-74	L'utente _G deve poter personalizzare l'altezza dei singoli livelli della zona	Soddisfatto
FM-75	Una zona deve contenere almeno 1 livello	Soddisfatto
FM-76	La numerazione dei livelli deve partire da 0 ("piano terra")	Soddisfatto
FM-77	Il sistema deve determinare il valore dell'altezza della zona dalla somma delle altezze dei singoli livelli	Soddisfatto

FM-78	L'utente _G deve visualizzare un errore se l'altezza indicata per il singolo livello non è positiva (≤ 0)	Soddisfatto
FM-79	L'utente _G deve poter modificare una zona già creata	Soddisfatto
FM-80	L'utente _G deve poter modificare una zona importata da database _G	Soddisfatto
FM-81	L'utente _G deve poter rimuovere una singola colonna, purché l'operazione non elimini una colonna con almeno un bin _G occupato	Soddisfatto
FM-82	L'utente _G deve poter rimuovere una singola colonna, purché l'operazione non elimini una colonna con indice inferiore all'indice di una colonna con almeno un bin _G occupato	Soddisfatto
FM-83	L'utente _G deve visualizzare un errore se l'operazione di rimozione di una colonna è impossibile per i vincoli individuati	Soddisfatto
FM-84	L'utente _G deve poter rimuovere un singolo livello, purché l'operazione non elimini un livello con almeno un bin _G occupato	Soddisfatto
FM-85	L'utente _G deve poter rimuovere un singolo livello, purché l'operazione non elimini un livello con indice inferiore all'indice di un livello con almeno un bin _G occupato	Soddisfatto
FM-86	L'utente _G deve visualizzare un errore se l'operazione di rimozione di un livello è impossibile per i vincoli individuati	Soddisfatto
FM-87	L'operazione di creazione di una nuova zona è da ritenersi conclusa solo con il corretto collocamento della stessa nell'ambiente 3D	Soddisfatto
FM-88	L'operazione di modifica di una zona è da ritenersi conclusa solo con il corretto collocamento della stessa nell'ambiente 3D	Soddisfatto
FM-89	L'utente _G deve poter eliminare qualsiasi zona	Soddisfatto
FM-90	I prodotti collocati in una zona rimossa non devono essere cancellati	Soddisfatto
FM-91	L'utente _G deve visualizzare un messaggio di avviso prima di procedere con l'eliminazione di una zona	Soddisfatto
FM-92	L'utente _G deve poter ispezionare una zona a partire dall'ambiente 3D	Soddisfatto
FM-93	L'utente _G deve poter visualizzare l'ID della zona ispezionata	Soddisfatto
FM-94	L'utente _G deve poter visualizzare la larghezza della zona ispezionata	Soddisfatto
FM-95	L'utente _G deve poter visualizzare la lunghezza della zona ispezionata	Soddisfatto
FM-96	L'utente _G deve poter visualizzare l'altezza della zona ispezionata	Soddisfatto
FM-97	La zona ispezionata deve essere evidenziata graficamente nell'ambiente 3D	Soddisfatto

FM-98	L'utente _G deve poter visualizzare la lista dei bin _G inclusi nella zona ispezionata	Soddisfatto
FM-99	L'utente _G deve poter visualizzare l'ID dei bin _G inclusi nella zona ispezionata	Soddisfatto
FM-100	L'utente _G deve poter visualizzare lo stato di occupazione dei bin _G inclusi nella zona ispezionata	Soddisfatto
FM-101	L'utente _G deve poter collocare una zona creata nello spazio 3D	Soddisfatto
FM-102	L'utente _G deve poter collocare una zona modificata nello spazio 3D	Soddisfatto
FM-103	Il sistema deve evidenziare graficamente una zona in una posizione non occupabile	Soddisfatto
FM-104	Il sistema deve impedire il collocamento di una zona su una posizione non occupabile	Soddisfatto
FM-105	Il sistema deve impedire il collocamento di una zona su di un'altra, ovvero deve impedire la compenetrazione tra zone	Soddisfatto
FM-106	Il sistema deve impedire il collocamento di una zona al di fuori del perimetro dell'ambiente 3D	Soddisfatto
FM-107	L'utente _G deve poter visualizzare la lista delle zone contenute nell'ambiente 3D	Soddisfatto
FM-108	L'utente _G deve poter visualizzare l'ID delle zone incluse nella lista	Soddisfatto
FD-109	L'utente _G deve poter cercare le zone in base all'ID	Soddisfatto
FD-110	Le zone che rispondono ai criteri di ricerca devono essere evidenziate graficamente	Soddisfatto
FM-111	L'utente _G deve poter ispezionare un bin _G a partire dall'ambiente 3D	Soddisfatto
FM-112	L'utente _G deve poter visualizzare l'ID del bin _G ispezionato	Soddisfatto
FM-113	L'utente _G deve poter visualizzare la lunghezza del bin _G ispezionato	Soddisfatto
FM-114	L'utente _G deve poter visualizzare la larghezza del bin _G ispezionato	Soddisfatto
FM-115	L'utente _G deve poter visualizzare l'altezza del bin _G ispezionato	Soddisfatto
FM-116	Il bin _G ispezionato deve essere evidenziato graficamente	Soddisfatto
FM-117	L'utente _G deve poter visualizzare le informazioni associate al prodotto eventualmente contenuto nel bin _G	Soddisfatto
FM-118	Ogni bin _G può contenere al massimo 1 prodotto	Soddisfatto
FD-119	L'utente _G può richiedere lo spostamento del POV sulla zona ispezionata	Non soddisfatto
FD-120	L'utente _G può richiedere lo spostamento del POV sul bin _G ispezionato	Non soddisfatto

FD-121	L'utente _G deve poter visualizzare le informazioni associate ad un prodotto importato da database _G	Soddisfatto
FD-122	L'utente _G deve poter visualizzare l'ID del prodotto ispezionato	Soddisfatto
FD-123	L'utente _G deve poter visualizzare il nome del prodotto ispezionato	Soddisfatto
FD-124	L'utente _G deve poter visualizzare la categoria del prodotto ispezionato	Soddisfatto
FD-125	L'utente _G deve poter visualizzare la larghezza del prodotto ispezionato	Soddisfatto
FD-126	L'utente _G deve poter visualizzare la lunghezza del prodotto ispezionato	Soddisfatto
FD-127	L'utente _G deve poter visualizzare l'altezza del prodotto ispezionato	Soddisfatto
FD-128	L'utente _G deve poter visualizzare il peso del prodotto ispezionato	Soddisfatto
FD-129	L'utente _G deve poter visualizzare la lista dei prodotti importati da database _G	Soddisfatto
FD-130	L'utente _G deve poter distinguere tra prodotti collocati in un bin _G e non collocati	Soddisfatto
FD-131	L'utente _G deve poter visualizzare la lista dei prodotti collocati	Soddisfatto
FD-132	L'utente _G deve poter visualizzare la lista dei prodotti non collocati	Soddisfatto
FD-133	L'utente _G deve poter visualizzare il nome del prodotto nella lista dei prodotti (collocati e non)	Soddisfatto
FD-134	L'utente _G deve poter visualizzare l'ID del prodotto nella lista dei prodotti (collocati e non)	Soddisfatto
FD-135	L'utente _G deve poter visualizzare la categoria del prodotto nella lista dei prodotti (collocati e non)	Soddisfatto
FD-136	L'utente _G deve poter visualizzare l'ID della zona di appartenenza di un prodotto nella lista dei prodotti collocati	Non soddisfatto
FD-137	L'utente _G deve poter visualizzare l'ID del bin _G di appartenenza di un prodotto nella lista dei prodotti collocati	Non soddisfatto
FD-138	L'utente _G deve poter filtrare la lista dei prodotti collocati in base all'ID	Non soddisfatto
FD-139	L'utente _G deve poter filtrare la lista dei prodotti non collocati in base all'ID	Non soddisfatto
FD-140	L'utente _G deve poter filtrare la lista dei prodotti collocati in base al nome	Soddisfatto
FD-141	L'utente _G deve poter filtrare la lista dei prodotti non collocati in base al nome	Soddisfatto

FD-142	L'utente _G deve poter filtrare la lista dei prodotti collocati in base alla categoria	Soddisfatto
FD-143	L'utente _G deve poter filtrare la lista dei prodotti non collocati in base alla categoria	Soddisfatto
FD-144	I filtri di ricerca devono essere mutuamente esclusivi	Soddisfatto
FD-145	L'utente _G deve poter inserire un ordine di movimentazione di un prodotto dalla lista dei prodotti ad un bin _G	Non soddisfatto
FM-146	L'utente _G deve poter inserire un ordine di movimentazione di un prodotto da un bin _G ad un altro tramite drag and drop	Soddisfatto
FM-147	Ciascun ordine di movimentazione deve inviare una richiesta alla API _G per la convalida dell'operazione	Soddisfatto
FM-148	La API _G deve ricevere almeno l'ID del bin _G di destinazione	Soddisfatto
FM-149	La API _G deve rispondere con stato HTTP 200 se l'operazione è stata convalidata	Soddisfatto
FM-150	La API _G deve rispondere con stato HTTP 4XX se l'operazione è stata rifiutata	Soddisfatto
FM-151	La API _G convalida o rifiuta le operazioni in maniera casuale	Soddisfatto
FM-152	Il sistema deve impedire l'inserimento di un ordine di movimentazione verso un bin _G occupato	Soddisfatto
FM-153	L'utente _G deve visualizzare l'esito dell'operazione di convalida da parte dell'API _G	Soddisfatto
FM-154	L'utente _G deve visualizzare un errore di connessione se l'accesso all'API _G non è possibile	Soddisfatto
FD-155	Quando un ordine di movimentazione è convalidato, esso viene inserito in una cronologia delle operazioni accessibile dall'utente _G	Soddisfatto
FM-156	Quando un ordine di movimentazione è rifiutato, il prodotto oggetto dell'operazione ritorna nella posizione di partenza	Soddisfatto
FD-157	L'utente _G deve poter visualizzare la cronologia degli ordini di movimentazione convalidati	Soddisfatto
FD-158	L'utente _G deve poter visualizzare l'ID del bin _G di partenza degli ordini di movimentazione convalidati se l'operazione è partita da un bin _G	Soddisfatto
FD-159	L'utente _G deve poter visualizzare l'ID del bin _G di destinazione dell'ordine di movimentazione convalidato	Soddisfatto
FD-160	L'utente _G deve poter visualizzare l'ID del prodotto oggetto dell'ordine di movimentazione convalidato	Soddisfatto
FD-161	L'utente _G deve poter visualizzare il nome del prodotto oggetto dell'ordine di movimentazione convalidato	Soddisfatto

FD-162	L'utente _G deve poter visualizzare l'ID del bin _G di partenza degli ordini di movimentazione convalidati se l'operazione è partita da un bin _G	Soddisfatto
FD-163	L'utente _G può poter ispezionare un singolo ordine di movimentazione convalidato	Soddisfatto
FM-164	Il sistema deve evidenziare graficamente il bin _G di destinazione dell'ordine di movimentazione ispezionato	Soddisfatto
FM-165	Se l'ordine di movimentazione ispezionato si è originato da un bin _G , il sistema deve evidenziare graficamente il bin _G di partenza nell'ambiente 3D	Soddisfatto

Tabella 4: Requisiti funzionali

6.2 Requisiti di qualità soddisfatti

Codice	Descrizione	Stato
QM-1	Deve essere rispettato quanto previsto dal documento <i>Norme di Progetto_G v1.26.0</i>	Soddisfatto
QM-2	Deve essere rispettato quanto previsto dal documento <i>Piano di Qualifica_G v1.2.0</i>	Soddisfatto
QM-3	Il codice sorgente deve essere consegnato utilizzando un repository _G GitHub _G pubblico	Soddisfatto
QM-4	Devono essere consegnati i diagrammi UML _G degli UC _G	Soddisfatto
QM-5	Deve essere consegnata la lista dei bug _G risolti	Soddisfatto
QM-6	Deve essere fornito un manuale d'uso per l'utente _G	Soddisfatto
QO-7	Deve essere consegnato lo schema del DB _G	Soddisfatto
QO-8	Deve essere consegnata la documentazione _G delle API _G realizzate	Soddisfatto
QM-9	Deve essere fornita la documentazione _G dell'architettura del prodotto	Soddisfatto

Tabella 5: Requisiti di qualità