

# Specifica tecnica

v0.4.0



<🔗> Farmacode

[farmacode.swe.unipd@gmail.com](mailto:farmacode.swe.unipd@gmail.com)

## Registro delle modifiche

Versione	Data	Scrittori	Revisori	Descrizione
0.4.0	2024-03-23	Pandolfo Mattia		Prima stesura architettura Back-end
0.3.0	2024-03-22	Bomben Filippo		Architettura Front-end
0.2.0	2024-03-20	Bomben Filippo		Tecnologie
0.1.0	2024-03-01	Favaron Riccardo	Bomben Filippo	Struttura iniziale del documento

# Indice

1) Introduzione .....	4
1.1) Scopo del documento .....	4
1.1.1) Struttura logica casi d'uso .....	4
1.2) Scopo del prodotto .....	4
1.3) Glossario .....	5
1.4) Maturità e miglioramenti .....	5
1.5) Riferimenti .....	5
1.5.1) Riferimenti normativi .....	5
1.5.2) Riferimenti informativi .....	5
2) Tecnologie .....	6
2.1) Tecnologie per la codifica .....	6
2.1.1) Linguaggi .....	6
2.1.2) Librerie e framework .....	6
2.1.3) Strumenti e servizi .....	7
2.2) Tecnologie per l'analisi del codice .....	8
2.2.1) Analisi statica .....	8
2.2.2) Analisi dinamica .....	8
3) Architettura .....	9
3.1) Architettura Front-end .....	9
3.1.1) Diagramma delle classi .....	9
3.1.2) Pagine .....	9
3.1.2.1) Login .....	9
3.1.2.2) Ricerca .....	10
3.1.2.3) Clienti .....	10
3.1.2.4) Prodotti .....	10
3.1.2.5) Profilo .....	11
3.1.3) Componenti .....	11
3.1.3.1) Filtri .....	11
3.1.3.2) Header .....	11
3.1.3.3) Results .....	12
3.2) Architettura Back-end .....	12
3.2.1) Introduzione .....	12
3.2.2) Schema base di dati (AGGIUNGERE TABELLA FEEDBACK) .....	12
3.2.3) Design Pattern utilizzati .....	13
3.2.4) Documentazione API .....	13
3.2.4.1) Chiamate GET (CONTROLLARE SE CI SONO NUOVE	
CHIAMATE) .....	13
3.2.4.2) Chiamate PUT .....	14
4) Stato requisiti funzionali .....	15
4.1) Tabella requisiti funzionali .....	15
4.2) Grafico requisiti funzionali .....	18
5) Elenco delle immagini .....	19
6) Elenco delle tabelle .....	20

# 1) Introduzione

## 1.1) Scopo del documento

Il documento riguardante l'analisi dei requisiti è un elemento di fondamentale importanza per i progetti di sviluppo software che voglio rispettare i massimi standard di qualità definiti dall'insegnamento dell'ingegneria del software.

Il presente documento ha lo scopo di fornire una descrizione dettagliata e più precisa possibile riguardanti le linee di massima del prodotto, che comprende i requisiti, così detti, obbligatori, desiderati e opzionali che vanno a rispondere alle necessità del proponente.

Si specializza sull'analisi dei bisogni dell'utente utilizzatore esaminati dallo studio del capitolato e durante i vari incontri con l'azienda proponente volti a tale scopo.

Le richieste del proponente sono, dunque, raccolte e ben identificate nel seguente documento; inoltre, sono classificate secondo le categorie standard di requisiti funzionali, di qualità e di vincolo.

L'analisi dei requisiti compone la pietra portante della progettazione di un sistema software, in quanto esplicita le funzionalità che il prodotto finale deve offrire. È essenziale per i programmatori usufruire di tale documento per assimilare a pieno le necessità dei proponenti di progetto per poi trovare la soluzione che più si sposa a soddisfare le esigenze proposte.

Il documento seguente deve essere il più completo e specifico possibile così da garantire requisiti corretti e che riscoprano tutti gli scenari plausibili per limitare i rischi di progetto ed evitare di inciampare in errori e ritardi che si traducono in costi maggiori.

È utile definire una precisa e formale rappresentazione grafica dei requisiti e degli attori in gioco grazie ai diagrammi dei casi d'uso, così da facilitare la comprensione a tutti.

### 1.1.1) Struttura logica casi d'uso

I casi d'uso descritti in questo documento hanno una precisa struttura logica descritta dal seguente modello:

- Titolo: Titolo del caso d'uso;
- Figura;
- Attori coinvolti: Il soggetto che esegue una determinata azione;
- Precondizioni: Lo stato del sistema prima del caso d'uso;
- Postcondizioni: Lo stato del sistema dopo l'esecuzione dello scenario descritto dal caso d'uso;
- Scenario principale: Descrizione dettagliata delle azioni svolte dall'attore durante il caso d'uso, intermedio tra le ipotesi e i risultati;
- Estensioni (se presenti): Possibili estensioni derivanti dal caso d'uso;
- Generalizzazioni (se presenti): Generalizzazioni di attori e casi d'uso.

## 1.2) Scopo del prodotto

Il progetto ha lo scopo di realizzare un *sistema di raccomandazione* con relativa interfaccia web che guidi le attività dell'azienda utilizzatrice del prodotto finale; suggerendo a quali clienti rivolgere le singole attività di marketing e commerciali.

Dall'interfaccia utente del sistema software sarà possibile selezionare uno specifico cliente e visualizzare i prodotti da lui acquistati e quelli che il sistema ha individuato come raccomandati. Inoltre selezionato un articolo il sistema suggerirà a quali clienti proporli, selezionandoli in base a quanto probabile siano interessati a quel determinato prodotto. I vari prodotti possono essere filtrati per categoria così da facilitarne la ricerca e restringere il campo di soluzione.

Ogni risultato restituito dal sistema di raccomandazione è classificabile tramite un feedback

così da poter eventualmente correggere il tiro dell'algoritmo che ha fornito l'esito della suggerimento.

L'utente amministratore avrà poi la possibilità di usufruire di altre funzionalità dedicate, come ad esempio visualizzare la cronologia delle ricerche.

### 1.3) Glossario

Al fine di evitare eventuali equivoci o incomprensioni riguardo la terminologia utilizzata all'interno di questo documento, si è deciso di adottare un Glossario, con file apposito, in cui vengono riportate tutte le definizioni rigogliose delle parole ambigue utilizzate in ambito di questo progetto. Nel documento appena descritto verranno riportati tutti i termini definiti nel loro ambiente di utilizzo con annessa descrizione del loro significato.

La presenza di un termine all'interno del Glossario è evidenziata dal *colore blu*.

### 1.4) Maturità e miglioramenti

Questo documento è stato realizzato utilizzando un approccio incrementale, con lo scopo di semplificare i cambiamenti nel tempo in base alle reciproche esigenze decise da entrambi le parti, ovvero membri del gruppo di progetto e azienda proponente. Pertanto non può essere considerato esaustivo e completo, ma in costante miglioramento.

### 1.5) Riferimenti

#### 1.5.1) Riferimenti normativi

- Norme di Progetto v.2.0.0;
- Capitolato C2: Sistemi di raccomandazione  
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C2.pdf>;
- Regolamento progetto didattico  
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/PD2.pdf>.

#### 1.5.2) Riferimenti informativi

- I diagrammi dei casi d'uso (UML) (slide del corso di Ingegneria del Software)  
<https://www.math.unipd.it/~rcardin/swea/2022/Diagrammi%20Use%20Case.pdf>.
- Progettazione: I pattern architetturali (slide del corso di Ingegneria del Software)  
<https://www.math.unipd.it/~rcardin/swea/2022/Software%20Architecture%20Patterns.pdf>
- Verifica e validazione: analisi statica (T10) (slide del corso di Ingegneria del Software)  
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T10.pdf>
- Verifica e validazione: analisi dinamica aka testing (T11) (slide del corso di Ingegneria del Software)  
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T11.pdf>
- Programmazione: SOLID programming (slide del corso di Ingegneria del software)  
[https://www.math.unipd.it/~rcardin/swea/2021/SOLID%20Principles%20of%20Object-Oriented%20Design\\_4x4.pdf](https://www.math.unipd.it/~rcardin/swea/2021/SOLID%20Principles%20of%20Object-Oriented%20Design_4x4.pdf)

## 2) Tecnologie

Questa sezione serve a fornire una panoramica generale sulle tecnologie adottate per il progetto. Vengono riportate sottoforma di tabelle le diverse tecnologie, sia per la codifica che per l'analisi e il test del codice. Ogni tabella è formata da tre colonne che riportano:

- La tecnologia utilizzata (il nome del linguaggio/framework/strumento);
- La descrizione del ruolo che la tecnologia ha avuto all'interno del progetto;
- La versione della tecnologia usata.

### 2.1) Tecnologie per la codifica

Le tecnologie per la codifica del progetto riguardano i vari linguaggi utilizzati per la scrittura del codice, le librerie e framework adottate per facilitare l'implementazione delle funzionalità e gli strumenti utilizzati per la gestione della codifica del progetto.

La scelta di determinate tecnologie è il risultato di ricerche nelle quali abbiamo cercato di capire i vantaggi che avrebbero potuto portare al progetto.

#### 2.1.1) Linguaggi

Tecnologia	Descrizione	Versione
Linguaggi		
HTML	Linguaggio di markup utilizzato per la creazione e gestione della struttura delle pagine web. La sua funzione è quella di “scheletro” delle pagine e del contenuto in esse.	5
CSS	Linguaggio per la formattazione dei documenti HTML, il suo scopo è di gestire lo stile e il design del sito.	3
JavaScript	Linguaggio di programmazione per la gestione degli eventi dell'utente e per la comunicazione con l'API.	TD
Python	Linguaggio di programmazione usato per la creazione del sistema di raccomandazione.	3.11.5
SQL	Linguaggio di interrogazione per la creazione e gestione del database.	TD

Tabella 1: Linguaggi

#### 2.1.2) Librerie e framework

Tecnologia	Descrizione	Versione
Librerie e framework		
Pandas	Libreria per Python utilizzata per la manipolazione e l'analisi dei dati	2.1.1

Surprise	Libreria per Python utilizzata per semplificare lo sviluppo di sistemi di raccomandazione e valutare le prestazioni di algoritmi di filtraggio collaborativo	1.1.3
React.js	Libreria JavaScript utilizzata per semplificare lo sviluppo front-end, consentendo una gestione modulare delle componenti grafiche.	18.2.0
PrimeReact	Suite per l'User Interface per React.js che utilizza componenti già definiti e ben strutturati.	10.5.1
Express	Libreria di JavaScript utilizzata per lo sviluppo back-end del sito	4.18.2
Flask	VIENE EFFETTIVAMENTE USATO?	3.0.x
NumPy	Libreria per Python utilizzata per la manipolazione di array e matrici multidimensionali.	1.26.0
PyTorch	Framework per l'apprendimento automatico basato su Python che offre tensori potenti, grafi computazionali dinamici e autograd.	2.2.2
Tailwind CSS	Framework per css utilizzato per lo sviluppo di interfacce utente.	3.4.1
Axios	Libreria JavaScript utilizzata per effettuare richieste HTTP sia lato client che lato server	1.6.8

Tabella 2: Librerie e framework

### 2.1.3) Strumenti e servizi

Tecnologia	Descrizione	Versione
<b>Strumenti e servizi</b>		
MySQL	RDBMS per la creazione e gestione dei database in SQL.	2.18.1
Node.js	Ambiente di runtime open-source per l'esecuzione di codice JavaScript lato server tramite appositi script.	18.16.1
NPM	Gestore di pacchetti (Node Package Manager) per JavaScript all'interno di Node.js.	9.5.1
VS Code	IDE di programmazione gratuito ricco estensioni esterne.	TD

Docker	Creatore di ambienti di sviluppo tramite container per la gestione delle dipendenze.	TD
Git	Sistema di controllo e versionamento utilizzato per la gestione del codice.	TD
Anaconda	Gestore e distributore per Python dei pacchetti per la gestione delle versioni.	TD

Tabella 3: Strumenti e servizi

## 2.2) Tecnologie per l'analisi del codice

### 2.2.1) Analisi statica

Tecnologia	Descrizione	Versione
<b>Analisi statica</b>		
Ruff	Strumento per l'analisi statica del codice Python, individua errori, violazioni delle convenzioni di codifica e altri problemi nel codice sorgente.	0.3.3
ESLint	Strumento utilizzato per l'analisi statica del codice JavaScript e TypeScript, che aiuta a individuare gli errori di codice e le pratiche non ottimali.	8.57.0

Tabella 4: Analisi statica

### 2.2.2) Analisi dinamica

Tecnologia	Descrizione	Versione
<b>Analisi dinamica</b>		
Pytest	Framework di test open-source per Python. Offre un'ampia gamma di funzionalità per la scrittura e l'esecuzione di test unitari, di integrazione funzionali	8.0.x
Jest	Framework di test basato su JavaScript con funzionalità di creazione di mock e il testing del codice in modo asincrono.	27.5.1
GitHub Action	Servizio di CI/CD per automatizzare il processo di build, test e deploy del progetto software.	/

Tabella 5: Analisi dinamica



### 3) Architettura

Il gruppo, durante la fase di progettazione, ha deciso di adottare un'architettura a microservizi. La scelta di questa precisa architettura è ricaduta per la natura ben separata dei ruoli delle varie componenti del progetto. Per questo motivo la comunicazione tra i vari servizi avviene tramite API Rest, sviluppate sia in Python con Flask che in JavaScript attraverso la libreria Express. Abbiamo quindi deciso di dividere nel seguente modo il sistema:

- Front-end, la parte di presentazione del sistema, che rappresenta l'interfaccia utente a cui l'utente può accedere attraverso il browser. La parte client è stata sviluppata tramite HTML, CSS e JavaScript con la libreria React;
- Back-end, la parte logica del progetto che sfruttando API Rest, crea interazione tra il Database e l'algoritmo in maniera che comunichino correttamente e riescano a recuperare tutti i dati necessari.

#### 3.1) Architettura Front-end

L'architettura front-end del prodotto sfrutta alcuni dei design pattern più comuni della libreria React, rimodellati in base alle esigenze della specifica situazione e del progetto.

Abbiamo cercato, per quanto possibile, di separare il più possibile i compiti tra le varie componenti, per semplificare e gestire al meglio i vari stati dell'applicazione.

##### 3.1.1) Diagramma delle classi

In questa sezione vengono descritte le varie pagine attraverso la convenzione UML per la rappresentazione delle classi.

Alcune di queste pagine avranno delle componenti usate all'interno di esse, per evitare ridondanza, la descrizione delle pagine e la descrizione delle componenti saranno separate, in questo modo il diagramma sarà più semplice da leggere evitando un eccessivo caos.

Allo scopo di rendere il tutto più chiaro possibile, a seguito di ogni diagramma ci sarà una breve spiegazione sulla funzionalità della pagina/componente.

##### 3.1.2) Pagine

###### 3.1.2.1) Login

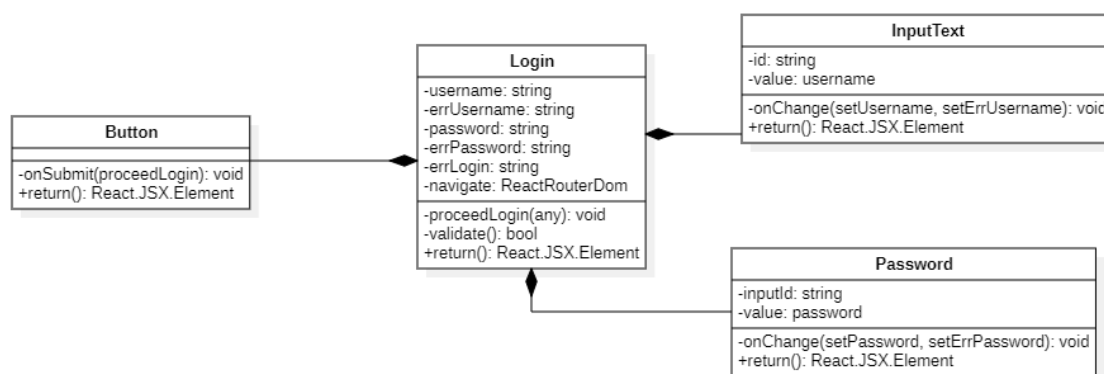


Figura 1: Login

La pagina Login è composta da tre diverse componenti, implementate sfruttando la suite PrimeReact.

E' composta da un *InputText*, usato per ricevere il dato username per effettuare il login, il componente *Password* per l'immissione della password ed infine *Button*, utilizzato per inviare tutte le informazioni compilate nel form e effettuare concretamente il login per accedere alla pagina principale.

### 3.1.2.2) Ricerca

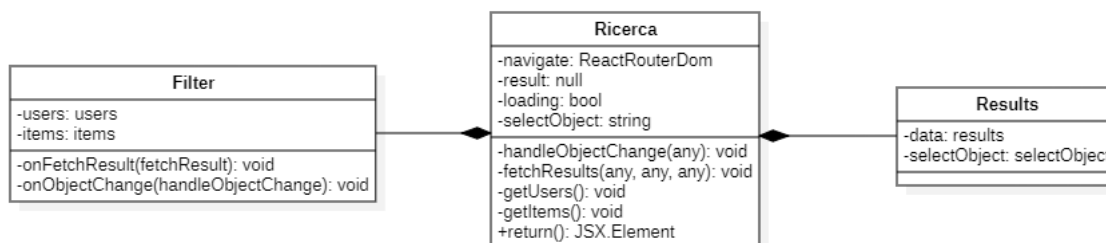


Figura 2: Ricerca

La pagina di ricerca è il core del progetto, in questa pagina si fanno le ricerche per le raccomandazioni in base ai criteri di scelta (prodotto per clienti o cliente per prodotti).

La componente *Filter* è quella che va a impostare la query per il recupero, una volta impostata correttamente, dei risultati di raccomandazione del modello. Quindi in base a come viene impostati i vari criteri del filtro, cambia anche la query al database.

*Results* invece si occupa di mostrare i dati recuperati, renderizzandoli a schermo all'interno di una tabella. La grandezza della tabella è direttamente correlata al numero di risultati che l'utente ha impostato in *Filters*.

### 3.1.2.3) Clienti

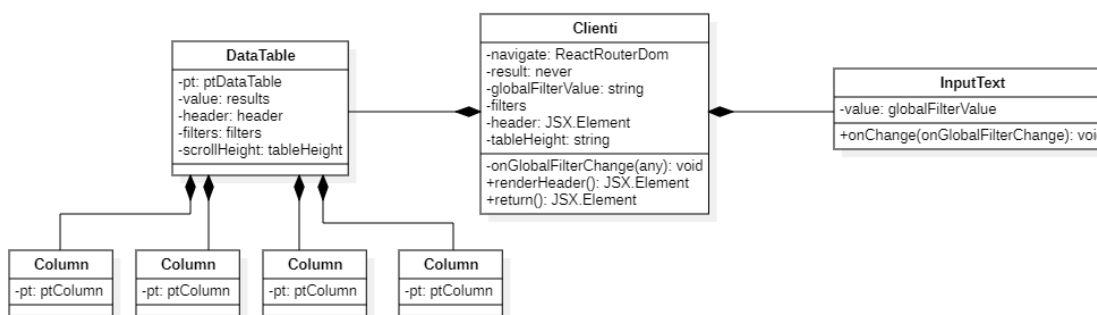


Figura 3: Clienti

Clienti è una pagina consultiva, dove vengono visualizzati tutti i clienti memorizzati nel database.

E' divisa nella parte superiore da un insieme di filtri per cercare nello specifico un cliente per i diversi

### 3.1.2.4) Prodotti

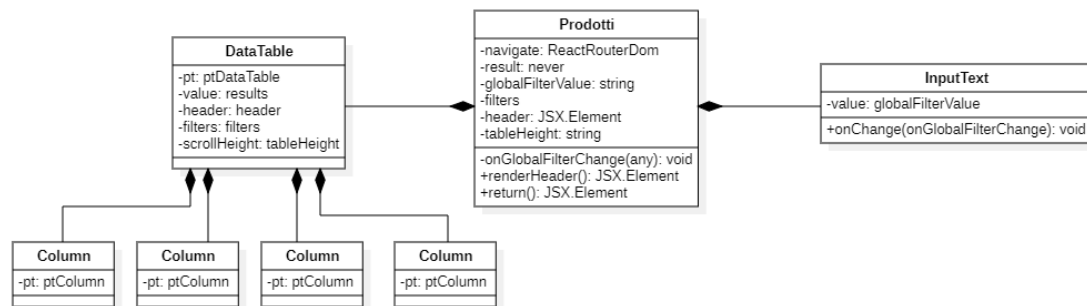


Figura 4: Prodotti

### 3.1.2.5) Profilo

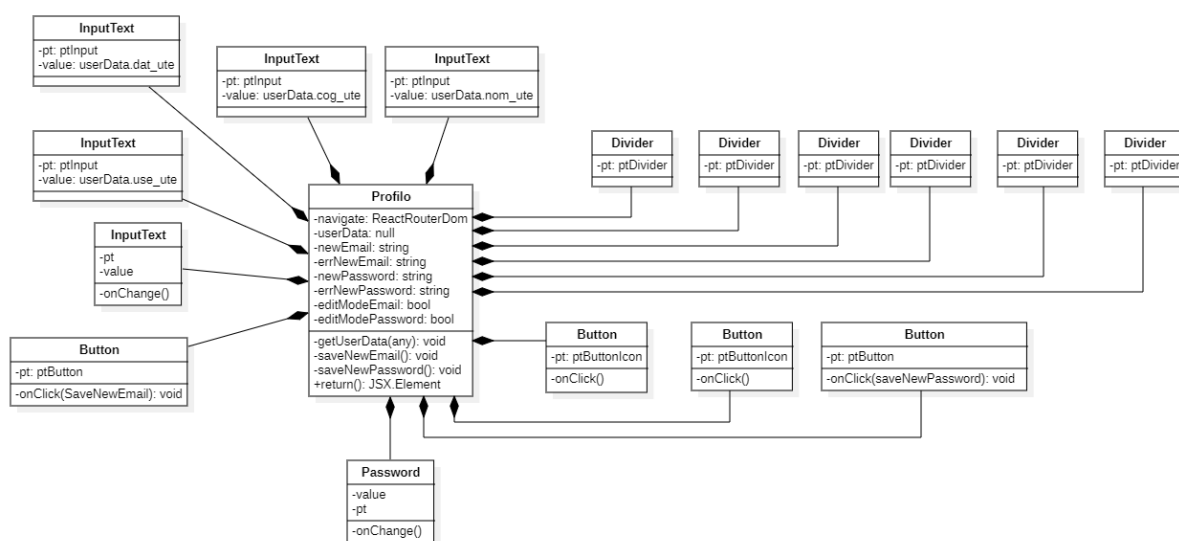


Figura 5: Profilo

### 3.1.3) Componenti

#### 3.1.3.1) Filtri

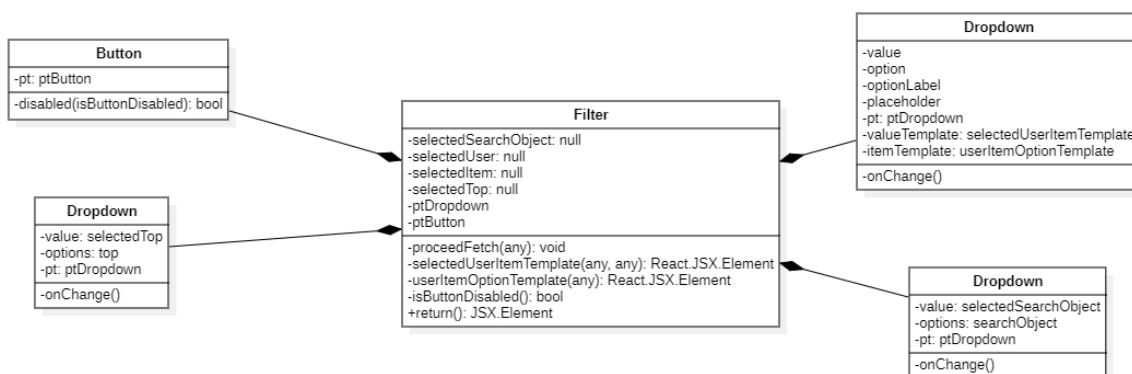


Figura 6: Filtri

#### 3.1.3.2) Header

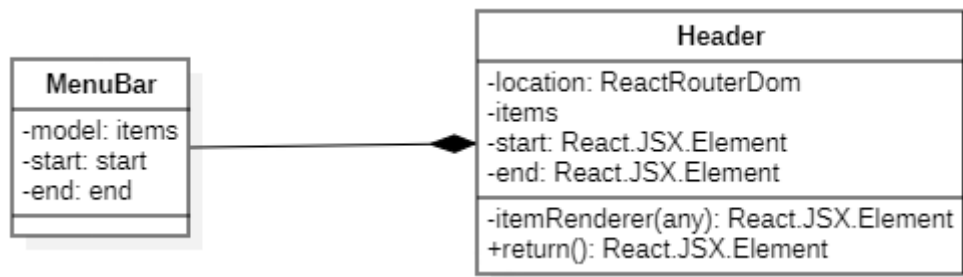


Figura 7: Header

### 3.1.3.3) Results

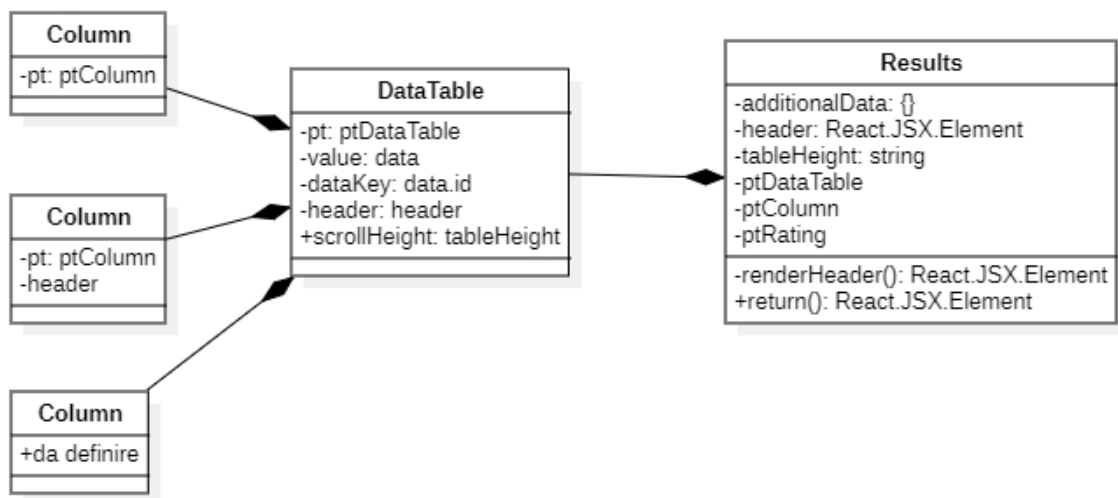


Figura 8: Results

## 3.2) Architettura Back-end

### 3.2.1) Introduzione

### 3.2.2) Schema base di dati (AGGIUNGERE TABELLA FEEDBACK)

In questa sezione, viene presentato lo schema di base di dati realizzato con MySQL, relativo all'architettura back-end del servizio descritto. Descriviamo più nel dettaglio questa composizione:

- **ute**, rappresentante i singoli utenti, comprensiva di:
  1. un username univoco;
  2. il nome dell'utente;
  3. il cognome;
  4. la data di nascita;
  5. una mail univoca;
  6. una password;
  7. l'informazione sull'essere un amministratore o meno.
- **prov**, rappresentante le provincie italiane, comprensiva di:
  1. il codice identificativo univoco della provincia;
  2. il nome della provincia.

- **anacli**, rappresentante i clienti, comprensiva di:
  1. un codice identificativo univoco;
  2. la ragione sociale;
  3. il codice della provincia di appartenenza (chiave esterna). ??????
- **linee\_comm**, rappresentante la linea dei prodotti, comprensiva di:
  1. un codice identificativo univoco;
  2. la linea del prodotto.
- **settori\_comm**, rappresentante il settore dei prodotti, comprensiva di:
  1. un codice identificativo univoco;
  2. il settore del prodotto.
- **famiglie\_comm**, rappresentante le famiglie dei prodotti, comprensiva di:
  1. un codice identificativo univoco;
  2. la famiglia del prodotto.
- **sottofamiglie\_comm**, rappresentante la sottofamiglia dei prodotti, comprensiva di:
  1. un codice identificativo univoco;
  2. la sottofamiglia del prodotto.
- **anaart**, rappresentante i singoli prodotti, comprensiva di:
  1. un codice identificativo univoco;
  2. la descrizione dell'articolo;
  3. la linea del prodotto (chiave esterna);
  4. il settore del prodotto (chiave esterna);
  5. la famiglia del prodotto (chiave esterna);
  6. la sottofamiglia del prodotto (chiave esterna).

### 3.2.3) Design Pattern utilizzati

### 3.2.4) Documentazione API

La sezione seguente fornisce una panoramica delle API create dal team Farmacode per comunicare con l'applicazione, delineando brevemente le operazioni disponibili e i dati accessibili. Una descrizione dettagliata della loro struttura è disponibile nel documento “Manuale Sviluppatore v.1.0.0”; questo permette agli sviluppatori interessati di comprendere appieno il software e di implementare nuove funzionalità in modo automatizzato, senza dover interagire manualmente con l'interfaccia utente. Pertanto questa sezione fornisce un'illustrazione generale e indicativa delle API disponibili.

#### 3.2.4.1) Chiamate GET (CONTROLLARE SE CI SONO NUOVE CHIAMATE)

- `/login/:use` :  
Ritorna i dati di login di uno specifico utente.
- `/users` :  
Ritorna la lista completa dei clienti.
- `/users/:id` :  
Ritorna le informazioni di uno specifico cliente.
- `/items` :  
Ritorna la lista completa dei prodotti con codice articolo e descrizione.

- /items/:id :  
Ritorna le informazioni di uno specifico prodotto.
- /prodotti :  
Ritorna la lista completa dei prodotti con tutte le informazioni, ordinata in base al codice.
- /prodotti/lineecommerciali :  
Ritorna la lista delle linee commerciali.
- /prodotti/settoricommerciali :  
Ritorna la lista dei settori commerciali.
- /prodotti/famigliecommerciali :  
Ritorna la lista delle famiglie commerciali.
- /prodotti/sottofamigliecommerciali :  
Ritorna la lista delle sottofamiglie commerciali.
- /clienti :  
Ritorna la lista completa dei clienti ordinata in base al codice.
- /clienti/province :  
Ritorna la lista delle province.
- /userana/:use :  
Ritorna i dati anagrafici di uno specifico utente.

#### **3.2.4.2) Chiamate PUT**

- /userana/:use/email :  
Aggiorna la mail di uno specifico utente.
- /userana/:use/password :  
Aggiorna la password di uno specifico utente.

## 4) Stato requisiti funzionali

### 4.1) Tabella requisiti funzionali

Codice	Descrizione	Fonti
ROF 1	L'utente per potere accedere all'applicazione deve autenticarsi all'interno del sistema.	Soddisfatto
ROF 2	L'utente deve fornire la propria email personale, nel campo email, per procedere con l'autenticazione nella pagina di Login.	Soddisfatto
ROF 3	L'utente deve fornire la propria password, nel campo password, per procedere con l'autenticazione nella pagina di Login.	Soddisfatto
RDF 4	Nel caso il sito sia in manutenzione è necessario che l'utente riceva un messaggio che esplicita l'impossibilità di usare l'applicazione.	
RDF 5	Nel caso l'autenticazione fallisse, è necessario che l'utente riceva un messaggio con dettagli che ne indicano il motivo.	
ROF 6	L'utente, una volta autenticato, deve poter accedere alla funzione "Profilo Utente" nella pagina principale del sito.	Soddisfatto
ROF 7	L'utente, una volta entrato nella sezione "Profilo Utente", deve poter visualizzare i dati utente o modificarli.	Soddisfatto
ROF 8	L'utente che ha scelto di visualizzare i "dati utente" deve visualizzare, l'anagrafica, l'email, l'username e la password.	
ROF 9	L'utente che ha scelto di modificare i dati utente, deve poter modificare l'email e la password.	
ROF 10	L'utente, una volta autenticato, deve poter effettuare il Logout tramite il pulsante presente nella pagina principale del sito.	Soddisfatto
ROF 11	L'utente, una volta autenticato, deve poter accedere alla funzione "Ricerca" nella pagina principale del sito.	Soddisfatto
ROF 12	L'utente una volta entrato nella sezione "Ricerca", deve poter effettuare una ricerca e visualizzarne i risultati.	Soddisfatto
ROF 13	L'utente che ha scelto di effettuare una ricerca, deve compilare tutti i campi per effettuarla e poter visualizzare i risultati.	Soddisfatto
ROF 14	L'utente che compila la scelta del topic della ricerca, può scegliere tra prodotto per cliente o cliente per prodotto, per poi compilare i successivi campi.	Soddisfatto

RDF 15	L'utente che compila la scelta del topic della ricerca, può scegliere la ricerca per cronologia, per poi compilare i successivi campi.	
RDF 16	L'utente che compila la scelta degli "N risultati", può scegliere tra i 5 migliori risultati (Top 5) o tra i migliori 10 (Top 10).	Soddisfatto
ROF 17	L'utente che ha effettuato una ricerca e ne visualizza i risultati, deve poter visualizzare, l'ID, il nome e lo score assegnato alla raccomandazione.	Soddisfatto
RDF 18	Nel caso la ricerca non andasse a buon fine, l'utente deve visualizzare un messaggio di errore che indica che la ricerca non è terminata correttamente. Il messaggio di errore deve essere mostrato in caso di errore anche per le ricerche di RDF 23, RDF 29, RDF 37, RDF 41.	Soddisfatto
ROF 19	L'utente che ha visualizzato i risultati della ricerca, deve poter inserire un feedback delle raccomandazioni mostrate.	
ROF 20	L'utente se decide di assegnare un feedback ad una raccomandazione, dovrà compilare i campi di "valutazione" e "commento".	
RDF 21	L'utente, una volta autenticato, deve poter accedere alla funzione "Catalogo Prodotti" nella pagina principale del sito.	Soddisfatto
RDF 22	L'utente una volta entrato nella sezione "Catalogo Prodotti", deve poter effettuare una ricerca e visualizzarne i risultati.	
RDF 23	L'utente che ha scelto di effettuare una ricerca deve compilare i campi Codice prodotto, Linea commerciale, Settore commerciale e Marca prodotto.	
RDF 24	L'utente che ha effettuato una ricerca e ne visualizza i risultati, deve poter visualizzare, l'immagine, l'ID e il nome del prodotto.	
RDF 25	L'utente che ha visualizzato i risultati della ricerca, può visualizzare i dettagli di un prodotto, cliccando sul bottone a fianco delle riga della ricerca.	
RDF 26	L'utente se decide di visualizzare i dettagli di un prodotto, deve poter vedere l'immagine, l'ID, il nome, la linea commerciale, il settore commerciale, la marca e la provenienza del prodotto.	
RDF 27	L'utente, una volta autenticato, deve poter accedere alla funzione "Lista clienti" nella pagina principale del sito.	Soddisfatto



RDF 28	L'utente una volta entrato nella sezione "Lista clienti", deve poter effettuare una ricerca e visualizzarne i risultati.	Soddisfatto
RDF 29	L'utente che ha scelto di effettuare una ricerca, deve compilare i campi Nome, Cognome e Provincia.	
RDF 30	L'utente che effettuatato una ricerca e ne visualizza i risultati, deve poter visualizzare l'ID, il nome, il cognome e la provincia del cliente	
RDF 31	L'utente, una volta autenticato, deve poter accedere alla funzione "Statistiche mensili" nella pagina principale del sito.	
RDF 32	L'utente una volta entrato nella sezione "Statistiche mensili", deve poter visualizzare il grafico e la lista di raccomandazioni utili.	
RDF 33	L'utente che ha scelto di visualizzare il grafico, visualizza sull'asse delle X i giorni e sull'asse delle Y le raccomandazioni utili.	
RDF 34	L'utente che ha scelto di visualizzare le raccomandazioni utili, deve poter visualizzare l'ID del prodotto, l'ID del cliente e lo score assegnato alla raccomandazione.	
RDF 35	L'utente, una volta autenticato, deve poter accedere alla funzione "Cronologia ricerche" nella pagina principale del sito.	
RDF 36	L'utente una volta entrato nella sezione "Cronologia ricerche", deve poter effettuare una ricerca e visualizzarne i risultati.	
RDF 37	L'utente che ha scelto di effettuare una ricerca, deve compilare i campi "Data" e "Username".	
RDF 38	L'utente che effettuatato una ricerca e ne visualizza i risultati, deve poter visualizzare la data, l'username e i criteri di ricerca riguardanti la cronologia della ricerca.	
RDF 39	L'utente, una volta autenticato, deve poter accedere alla funzione "Cronologia feedback" nella pagina principale del sito.	
RDF 40	L'utente una volta entrato nella sezione "Cronologia feedback", deve poter effettuare una ricerca e visualizzarne i risultati.	
RDF 41	L'utente che ha scelto di effettuare una ricerca, deve compilare i campi "Data" e "Username".	

RDF 42	L'utente che effettuatato una ricerca e ne visualizza i risultati, deve poter visualizzare la data, l'username e il contenuto del feedback riguardanti la cronologia del feedback.	
RDF 43	L'utente, una volta autenticato, deve poter accedere alla funzione "Carica dataset" e caricare un dataset esterno all'interno dell'applicazione.	
RDF 44	L'utente, se ha caricato un dataset esterno, deve poter avviare il training del dataset in maniera da poterlo usare per le raccomandazioni.	

Tabella 1: Requisiti funzionali

## 4.2) Grafico requisiti funzionali

## 5) Elenco delle immagini

## 6) Elenco delle tabelle