Università degli Studi di Salerno
Corso di Ingegneria del Software

Object Design Document
Versione 2.0

Paninoteca 80Fame

Partecipanti:

Nome	Matricola
Andreana Balbi	0512103732
Gianluca Longodardi	0512103612

Revision History

Data	Versione	Descrizione	Autore
22/12/2017	1.0	Prima Stesura del Documento	Longobardi/Balbi
27/12/2017	1.1	Revisione Completa del Documento	Longobardi/Balbi
31/12/2017	2.0	Seconda Revisione	Longobardi/Balbi

Indice

1. INTRODUZIONE

1.1 Object Design Trade-offs

Dopo il documento di Requirements Analysis e il documento di System Design in cui vi è una descrizione di ciò che sarà il nostro sistema, definendo i nostri obiettivi ma tralasciando gli aspetti implementativi, andiamo ora a formare il documento di Object Design che ha come obiettivo quello di produrre un modello che sia in grado di integrare tutte le funzionalità individuate nelle fasi precedenti.
In particolar modo, in questo documento si definiscono le interfacce delle classi, le operazioni, i tipi, gli argomenti e la signature dei sottosistemi definiti nel System Design. Inoltre sono specificati i trade-off e le linee guida.

Comprensibilità vs Tempo:
Il codice del sistema deve essere comprensibile il più possibile, in modo da facilitare la fase di testing ed eventuali future modifiche da apportare. Per rispettare queste linee guida il codice sarà accompagnato da commenti che serviranno a semplificarne la comprensione.
Prestazioni vs Costi:
Dato che il nostro progetto è sprovvisto di budget, per poter mantenere prestazioni elevate, in determinate funzionalità verranno utilizzati dei template open source esterni.
Interfaccia vs Usabilità:
L'interfaccia grafica è stata realizzata in maniera molto semplice, chiara e concisa, vengono utilizzati i form e pulsanti con lo scopo di rendere semplice l'utilizzo del sistema da parte dell'utente finale.
Sicurezza vs Efficienza:
La sicurezza, come descritto nei requisiti non funzionali del Requirements Analysis, rappresenta uno degli aspetti importanti del sistema. Ci limiteremo ad implementare sistemi di sicurezza basati su username e password degli utenti.

1.2 Linee Guida per la Documentazione delle Interfacce

Bisogna seguire delle linee guida per la stesura del codice:
Variabili:
- I nomi delle variabili devono iniziare con la lettera minuscola, e le parole successive con la lettera maiuscola. La dichiarazione delle variabili deve essere effettuata ad inizio blocco; in ogni riga vi deve essere una sola dichiarazione di variabile e va effettuato l'allineamento per migliorare la leggibilità.

Metodi:
- I nomi dei metodi devono iniziare con la lettera minuscola, e le parole successive con la lettera maiuscola. Di solito il nome del metodo è costituito da un verbo che identifica un'azione, seguito dal nome di un oggetto.

- Ai metodi va aggiunta una descrizione, la quale deve essere posizionata prima della dichiarazione del metodo, e deve descriverne lo scopo. La descrizione del metodo deve includere anche

informazioni riguardanti gli argomenti, il valore di ritorno, le eccezioni. I metodi devono essere raggruppati in base alla loro funzionalità.

Classi e pagine:
- I nomi delle classi e delle pagine devono iniziare con la lettera maiuscola, e anche le parole successive all'interno del nome devono iniziare con la lettera maiuscola.

1.3 Definizioni, acronimi e abbreviazioni

Acronimi:
- RAD: Requirements Analysis Document
- SDD: System Design Document
- ODD: Object Design Document

Abbreviazioni:
- DB: DataBase

1.4 Riferimenti

- Documento RAD_80_Fame.pdf del progetto
- Documento DatiPersistenti_80_Fame.PDF

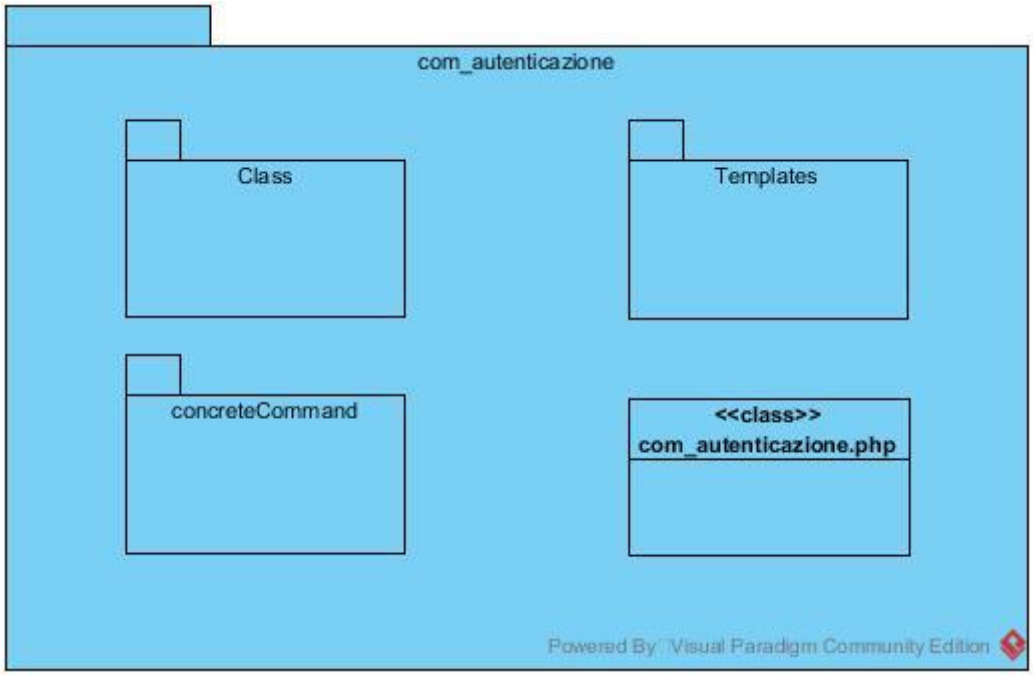
2. DESIGN PATTERN

2.1 Design pattern globali

Command Design Pattern: 80_Fame fa uso del Command Pattern che è uno dei pattern fondamentali, che permette di isolare la porzione di codice che effettua un'azione dal codice che ne richiede l'esecuzione; l'azione è incapsulata nell'oggetto Command. L'obiettivo è rendere variabile l'azione del client senza però conoscere i dettagli dell'operazione stessa.

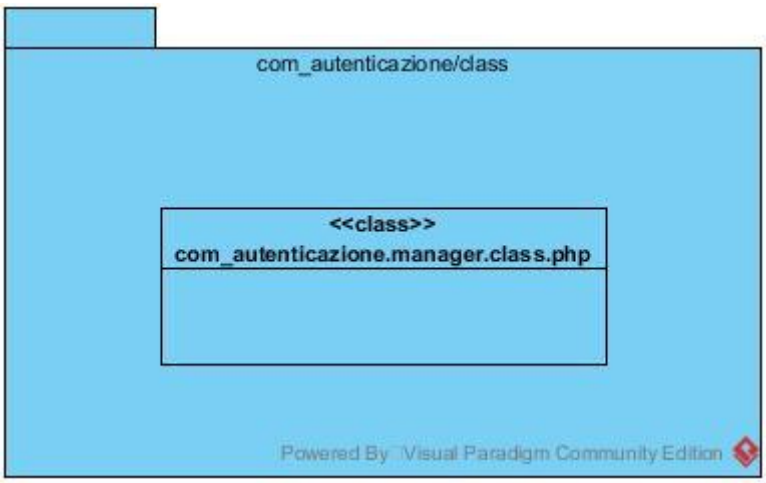
3.PACKAGE COMPONENTS

3.1 Package com_autenticazione



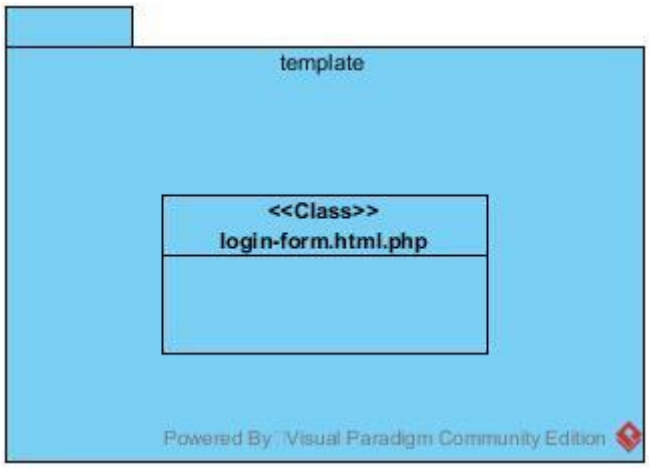
Classe	Descrizione
com_autenticazione.php	È il client del Command pattern. Questa classe in base al comando ricevuto reindirizza alla view corrispondente.

3.1.1 Package class



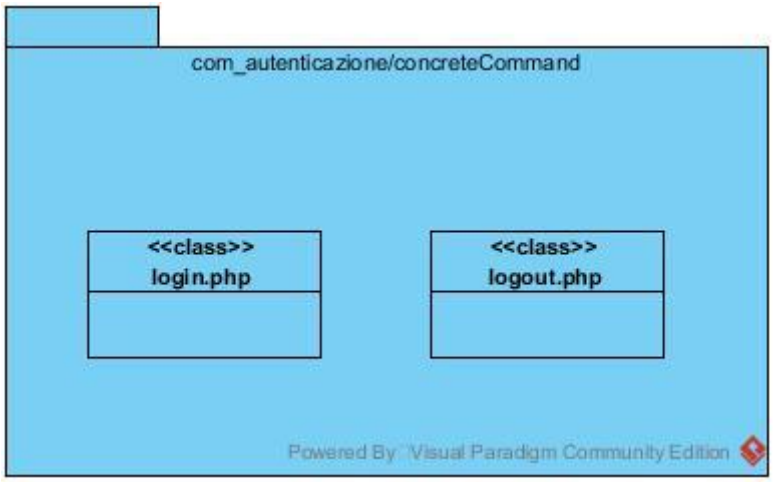
Classe	Descrizione
com_autenticazione.manager.class.php	Classe che ha al suo interno le regole business.

3.1.2 Package templates



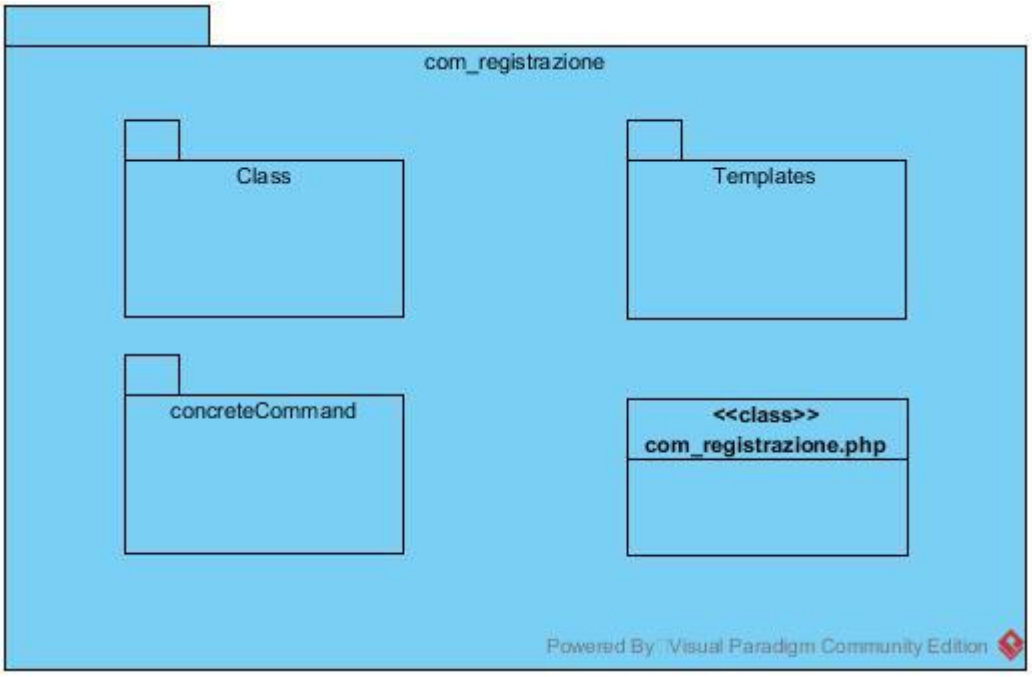
Classe	Descrizione
login-form.html.php	La view che consente al cliente di loggarsi al sistema

3.1.3 Package concreteCommand



Classe	Descrizione
Login.php	Controller che permette la gestione del login
Logout.php	Controller che permette la gestione del logout

3.2 Package com_registrazione

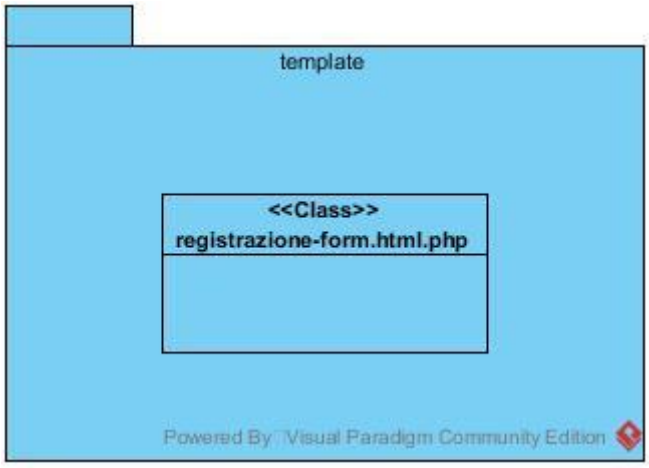


Classe	Descrizione
com_registrazione.php	È il client del Command pattern. Questa classe in base al comando ricevuto reindirizza alla view corrispondente.

3.2.1 Package class

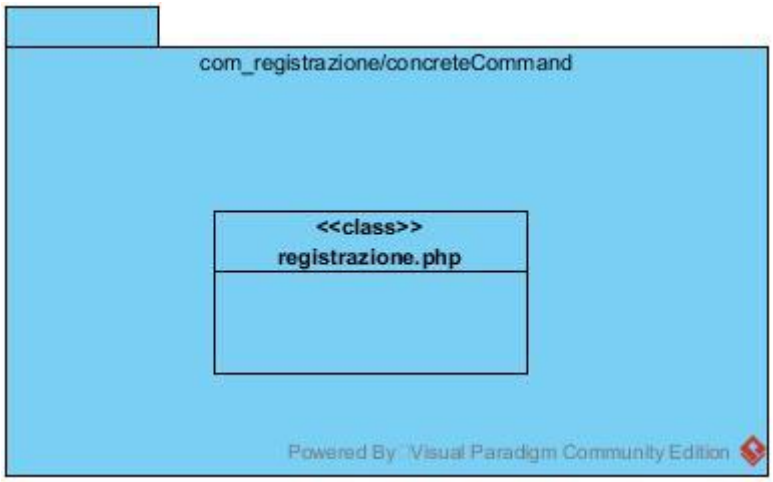
Classe	Descrizione
com_registrazione.manager.class.php	Classe che ha al suo interno le regole business.

3.2.2 Package template



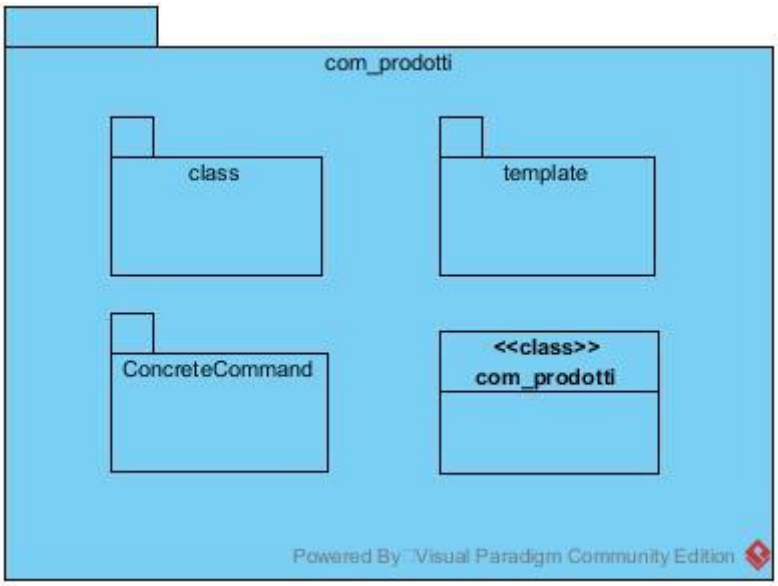
Classe	Descrizione
registrazione.-form..html.php	La view che consente all'utente per registrarsi al sistema

3.2.3 Package concreteCommand



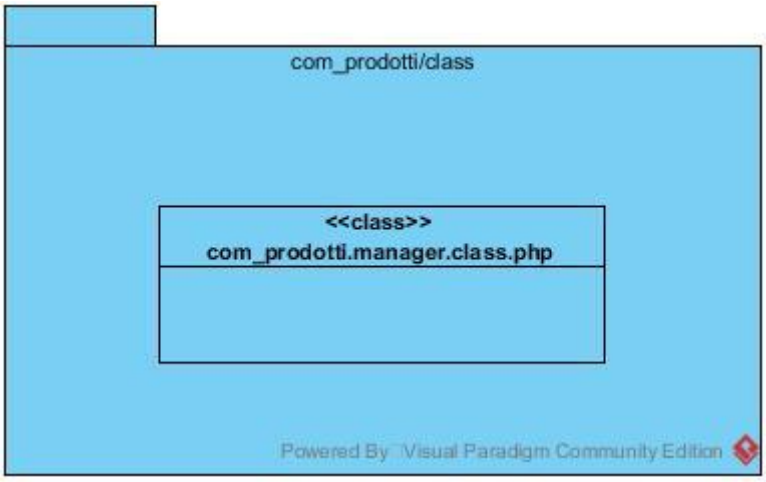
Classe	Descrizione
Registrazione.php	Controller che permette la gestione della registrazione

3.3 Package com_prodotti



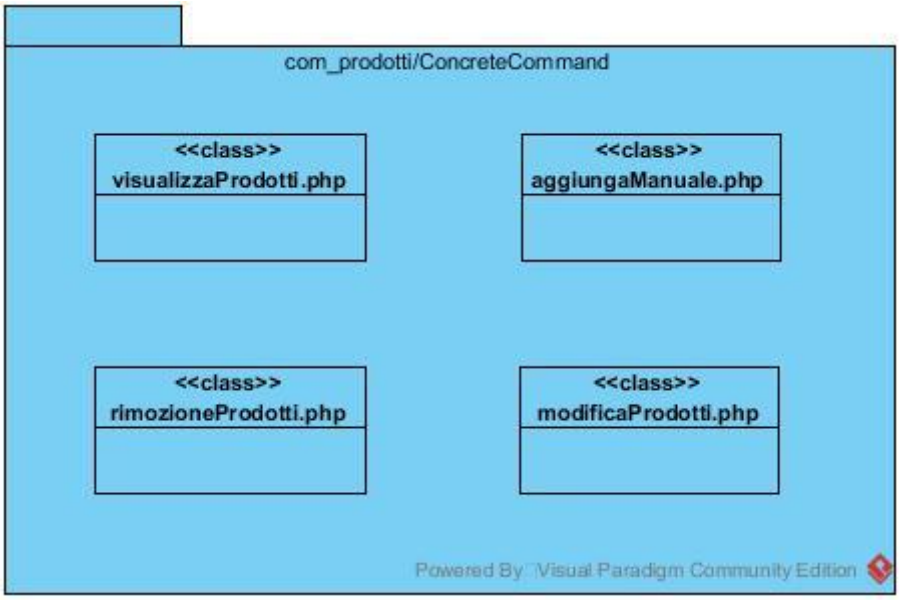
Classe	Descrizione
Com_prodotti.php	È il client del Command pattern.Questa classe in base al comando ricevuto reindirizza alla view corrispondente

3.3.1 Package class



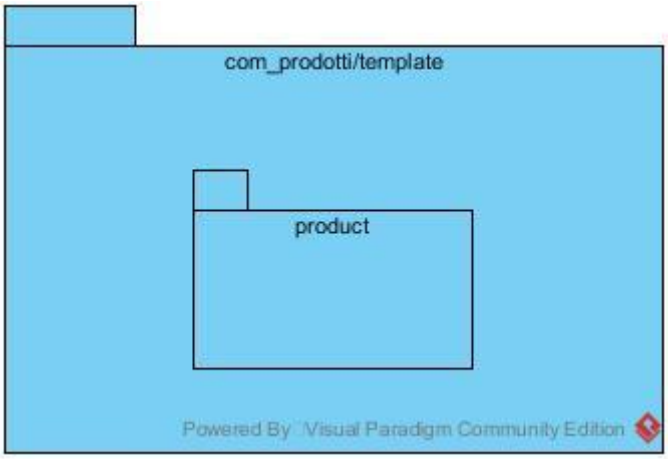
Classe	Descrizione
Com_prodotti.manager.class.php	Classe che ha al suo interno le regole business.

3.3.2 Package ConcreteCommand

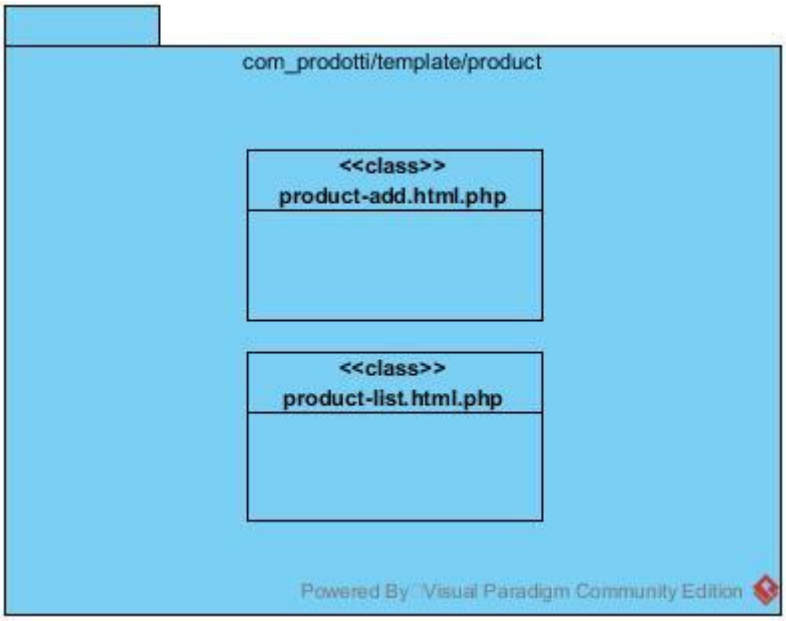


Classe	Descrizione
visualizzaProdotti.class.php	Controller che consente la visualizzazione dei prodotti
aggiuntaManuale.class.php	Controller che permette l'aggiunta di un prodotto manualmente
rimozioneProdotti.class.php	Controller che permette la rimozione di uno o più prodotti
modificaProdotti.class.php	Controller che permette la modifica di uno o più prodotti

3.3.3 Package template

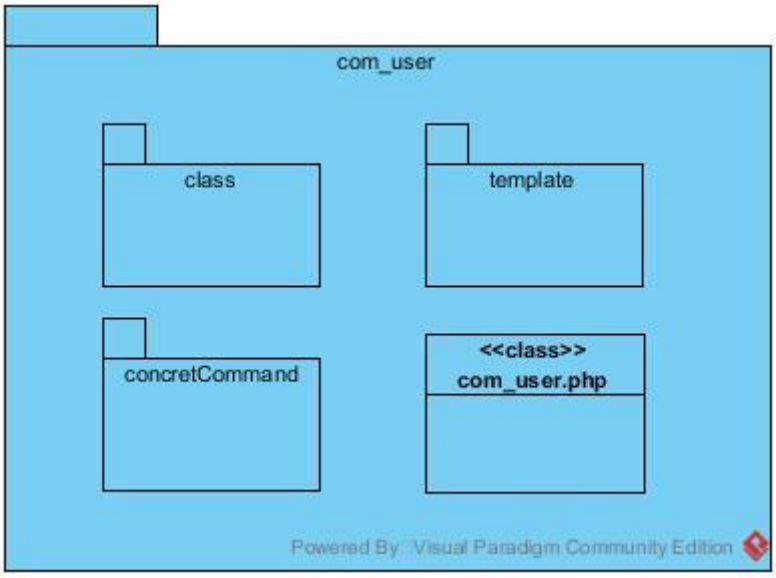


3.3.3.1 Package product



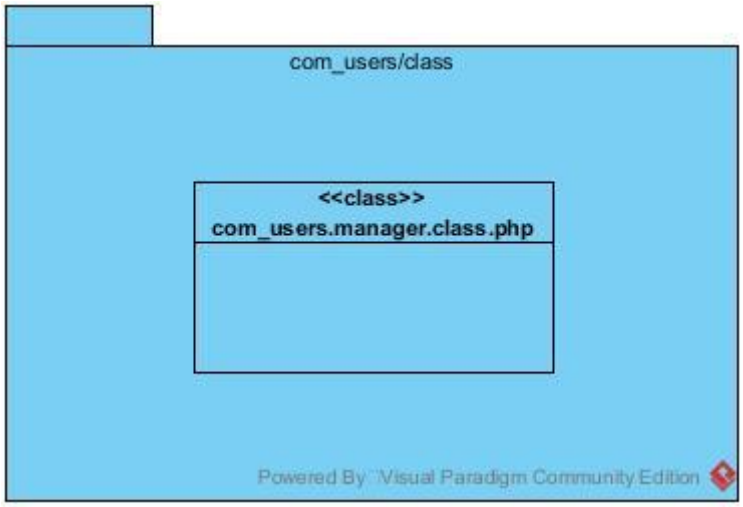
Classe	Descrizione
Product-add.html.php	Questa view permette l'aggiunta dei prodotti
Product-list.html.php	Questa view permette di generare la lista dei prodotti

3.4 Package com_utenti



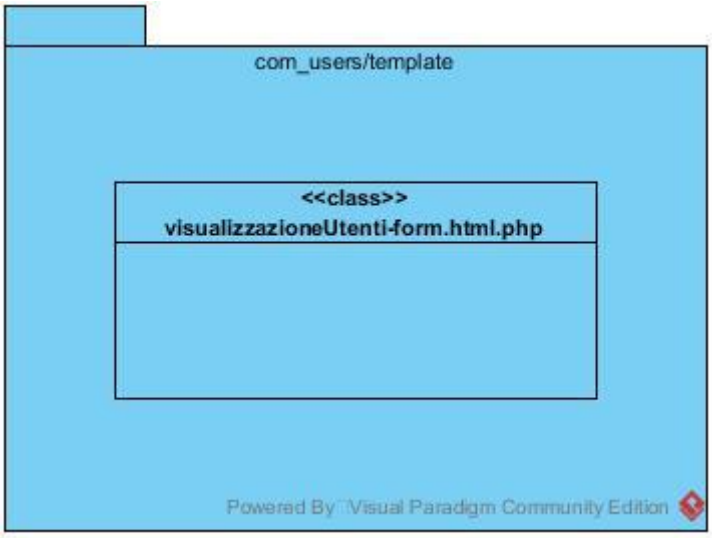
Classe	Descrizione
Com_user.php	È il client del Command Pattern. Questa classe in base al comando ricevuto reindirizza alla view corrispondente

3.4.1 Package class



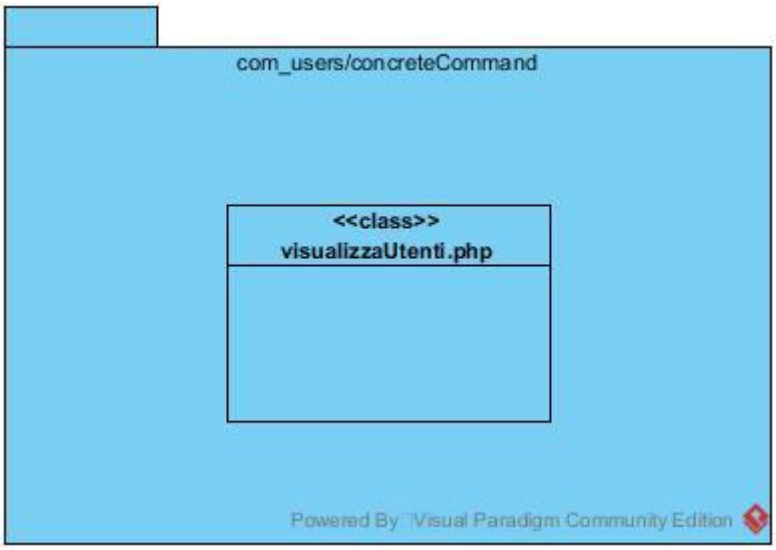
Classe	Descrizione
Com_users.manager.class.php	Classe che ha al suo interno le regole business.

3.4.2 Package template



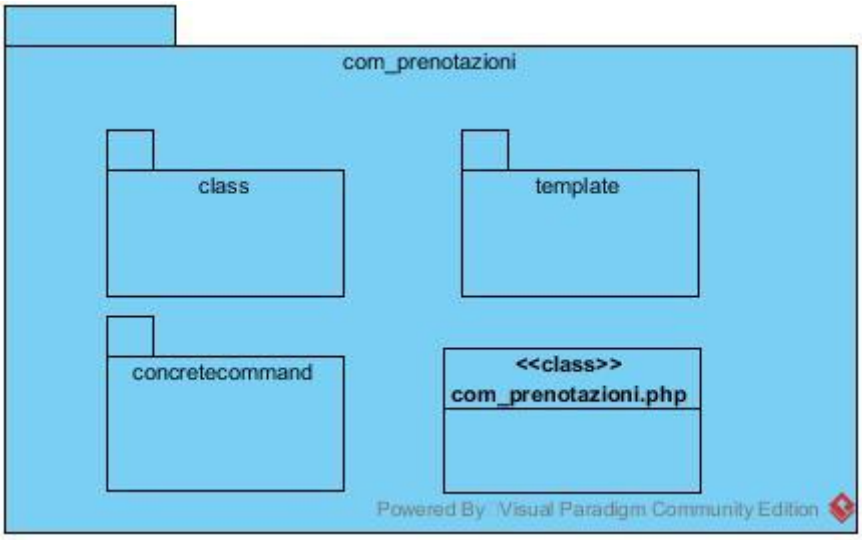
Classe	Descrizione
visualizzazioneUtentiAdmin-form.html.php	La view che consente all'amministratore di visualizzare tutti gli utenti del sistema

3.4.3 Package concreteCommand



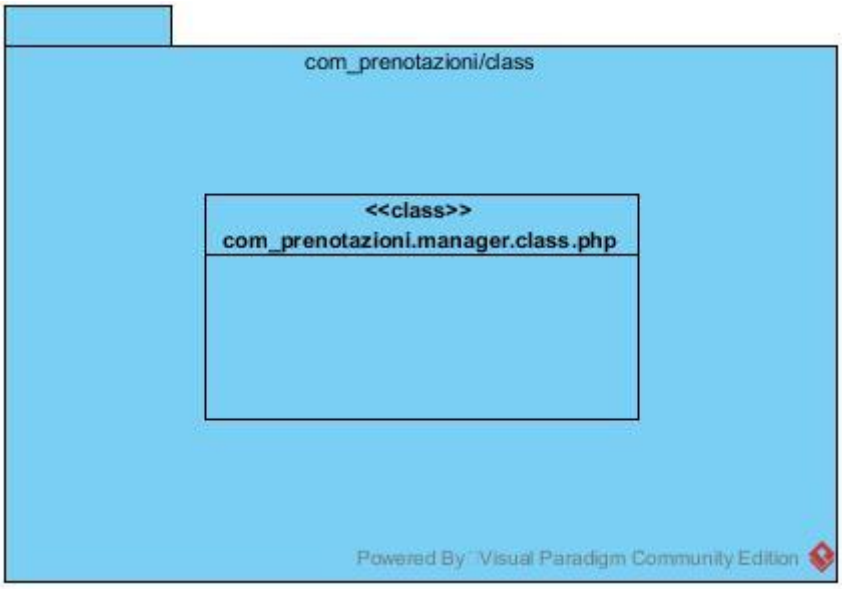
Classe	Descrizione
visualizzaUtenti.php	Controller che restituisce la lista degli utenti

3.5 Package com_prenotazioni



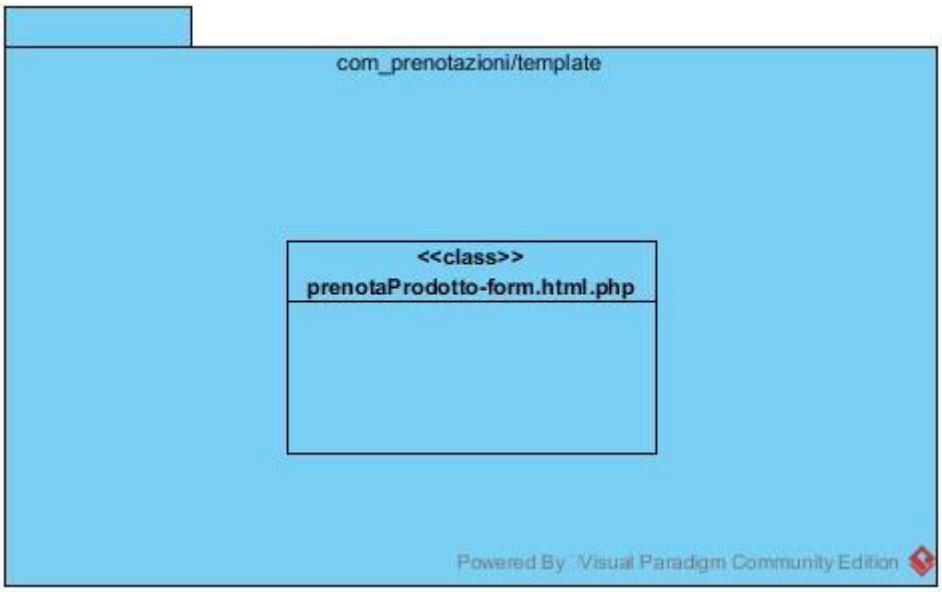
Classe	Descrizione
Com_prenotazioni.php	È il client del Command Pattern. Questa classe in base al comando ricevuto reindirizza alla view corrispondente

3.5.1 Package class



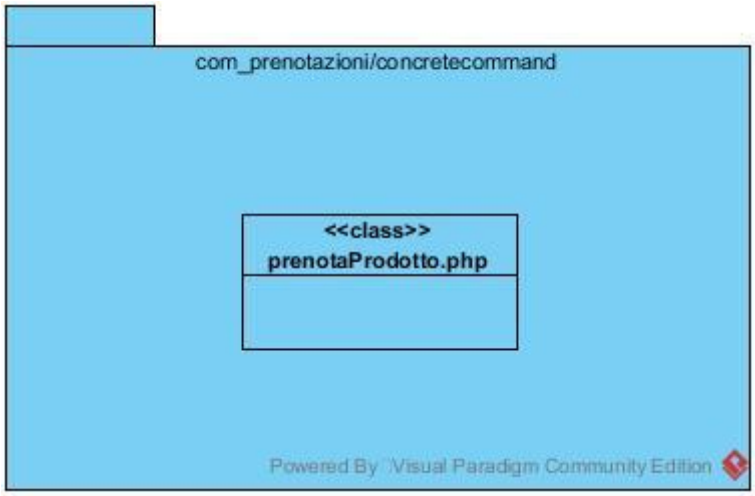
Classe	Descrizione
Com_prenotazioni.manager.class.php	Class che ha al suo interno le regole di business

3.5.2 Package template



Classe	Descrizione
visualizzaProdotto-form.html.php	La view che permette all'amministratore di visualizzare gli ordini effettuati
ordinaProdotto-form.html.php	La view che permette all'utente di effetture una prenotazione di un prodotto

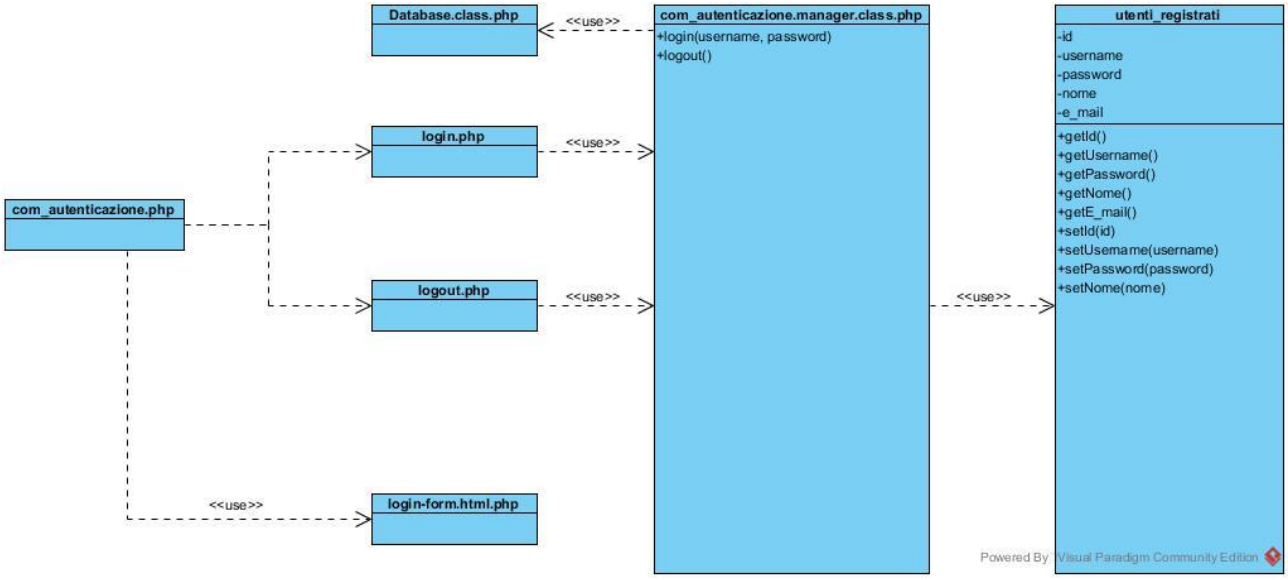
3.5.3 Package concreteCommand



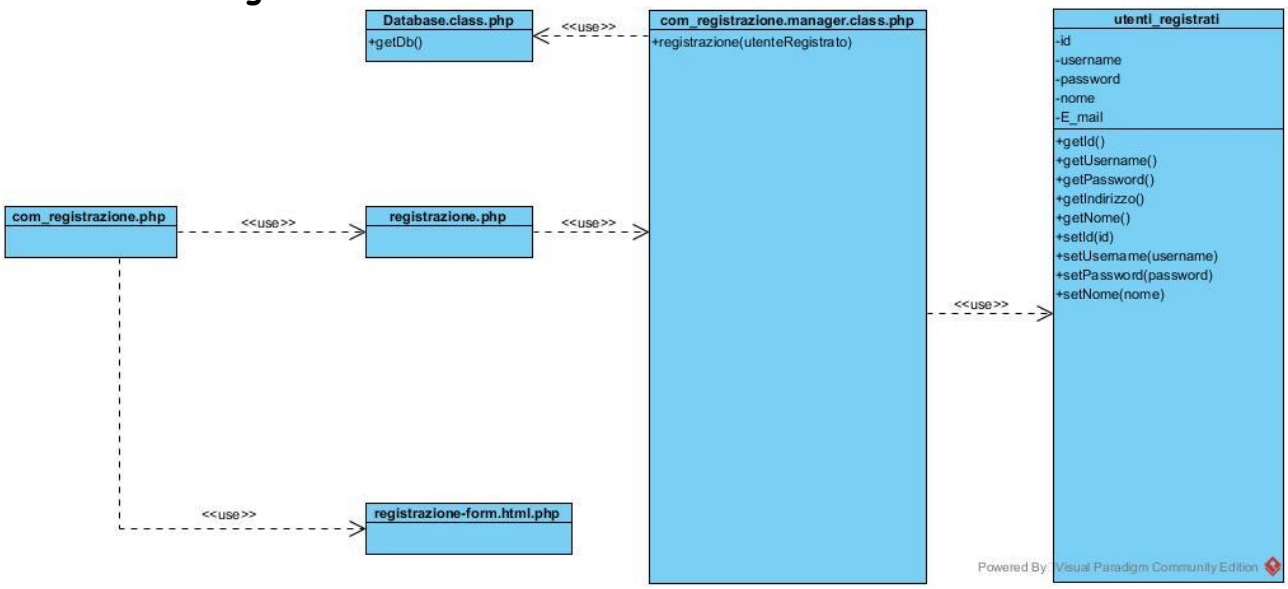
Classe	Descrizione
visualizzaProdotto.php	Controller che restituisce la lista dei prodotti
ordinaProdotto.php	Controller che consente l'acquisto di un prodotto

4.CLASS INTERFACES

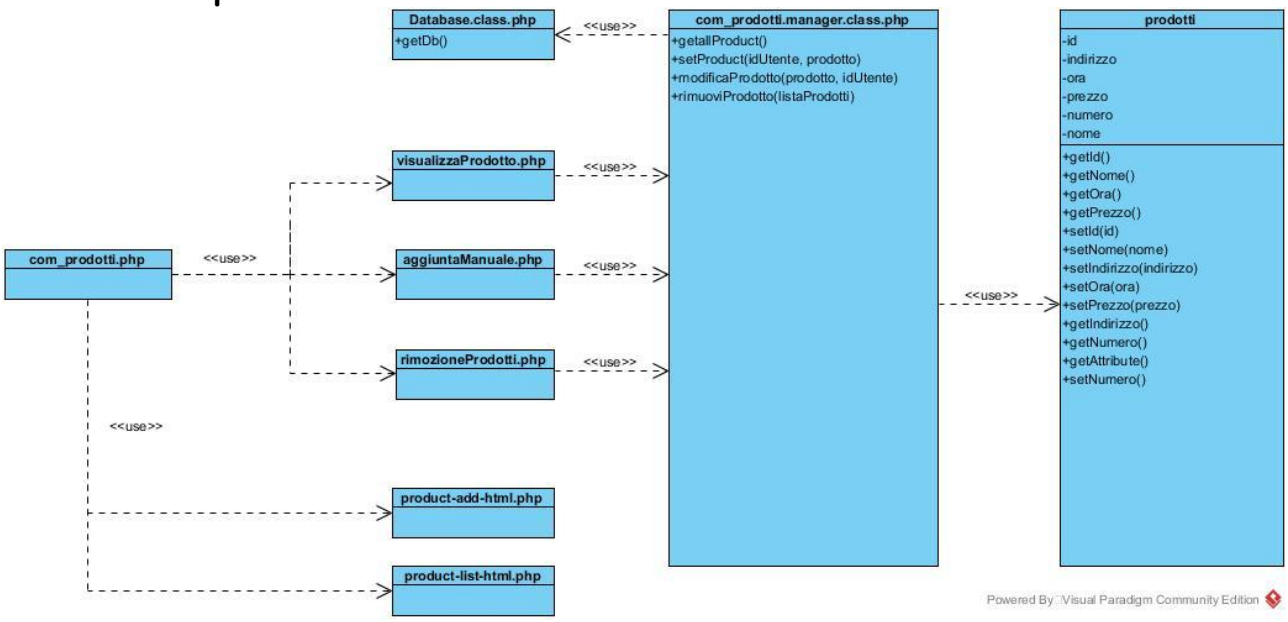
4.1 Gestione autenticazione



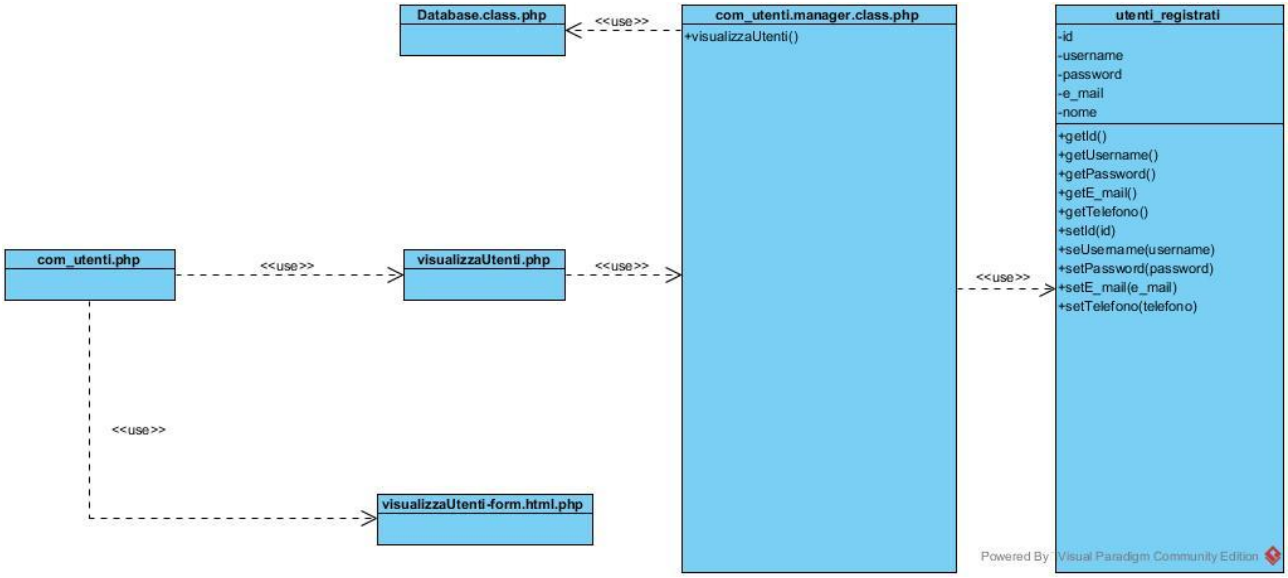
4.2 Gestione registrazione



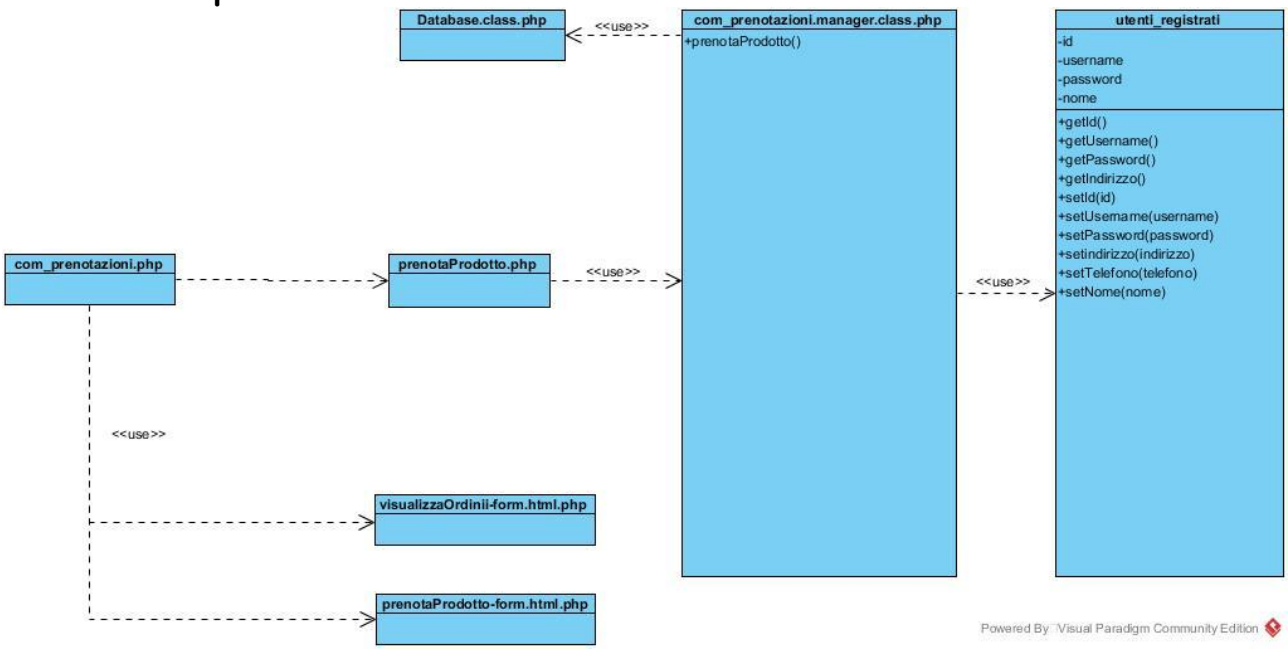
4.3 Gestione prodotti



4.4 Gestione utenti



4.5 Gestione prenotazioni



5.Glossario

Paninoteca_80Fame: nome del sistema che verrà sviluppato;
Utente: il termine identifica un attore del sistema che può usufruire dei servizi offerti
Utente loggato: il termine identifica un utente che ha eseguito il login correttamente;
Utente registrato: il termine identifica utente che ha effettuato la registrazione sul Sistema;
Amministratore: il termine identifica il creatore del sito che ha accesso al codice sorgente e a varie funzionalità;
Prodotti: il termine identifica un oggetto venduto dal sistema;
Aggiunta Manuale: il termine identifica una funzione del sistema che permette all'amministratore di inserire un prodotto non presente, fornendo tutti i parametri manualmente
Modifica Prodotto: il termine identifica una funzionalità del sistema consentita al solo amministratore che gli permette di modificare i prodotti;
Rimozione Prodotto: il termine identifica una funzionalità del sistema consentita al solo amministratore che gli permette di rimuovere i prodotti;
Prenotazione: il termine identifica una funzionalità del sistema che permette agli utenti di poter prenotare qualsiasi prodotto;
SDD: Documento di System Design
ODD: Documento di Object Design
RAD: Documento di Analisi dei Requisiti
DBMS: Sistema di gestione di basi di dati
Query: l'interrogazione da parte di un utente di un database
DataBase: Insieme organizzato di dati persistenti