Università degli Studi di Salerno
Corso di Ingegneria del Software

Sandwich on web

ODD
Versione 1.0

Data: 06/02/2016

Partecipanti:

Nome	**Matricola**
Sara Volpe	0512102434
Egidio Giacoia	0512102376
Nunzia Esposito	0512102328

Scritto da:	Team members

Revision History

Data	**Versione**	**Descrizione**	**Autore**
18/12/2015	1	Prima stesura del documento	Team members
19/12/2015	1	Aggiunta di: • Capitolo 1: Introduzione	Team members
20/12/2015	1	Aggiunta di: • Capitolo 2: Linee guida per l'implementazione	Sara Volpe
21/12/2015	1	Aggiunta di: • Capitolo 3: Packages	Sara Volpe
22/12/2015	1	Aggiunta di: • Capitolo 4: Class interfaces	Sara Volpe
18/01/2016	2	Modifica di: • Capitolo 3: Packages	Sara Volpe Nunzia Esposito
18/01/2016	2	Modifica di: • Capitolo 4: Class interfaces	Sara Volpe
20/01/2016		Revisione documento	Nunzia Esposito
05/02/2016	1	Aggiunta di: • Capitolo 5: Documentazione delle classi	Egidio Giacoia
06/02/2016	1	Modifica di :	Nunzia Esposito

		• Capitolo 5: Documentazione delle classi	

INDICE

1. Introduzione

Il corrente documento, usato per supportare la fase di implementazione, ha lo scopo di elaborare un modello che integra in modo lineare e preciso tutti i servizi determinati nelle fasi precedenti. In particolare definisce le interfacce delle classi, le operazioni, i tipi, gli argomenti e la signature dei sottosistemi definiti nel SDD. Inoltre successivamente nei paragrafi vengono descritti i trade-off, le linee guida e i design pattern per l'implementazione.

1.1 Object Design trade-offs

- **Comprensibilità vs Costi**
  La comprensibilità è un aspetto molto importante nella fase di testing. Ogni classe e metodo deve essere facilmente interpretabile anche da persone che non hanno partecipato al progetto. Nel codice verranno utilizzati i commenti standard e PHPdoc per incrementare la leggibilità del codice sorgente. Naturalmente questa peculiarità aggiungerà dei costi allo sviluppo del nostro progetto.

- **Prestazioni vs Costi**
  Non disponendo di alcun budget, utilizzeremo prodotti open source per la realizzazione del software Sandwich on web avendo lo scopo di rendere il sistema performante ed efficiente.

- **Interfaccia vs Easy-use**
  L'interfaccia mediante l'utilizzo di form si mostra facilmente comprensibile e intuitiva, rendendo semplice la gestione del DB(Easy-use).

- **Memoria vs efficienza**
  Si è cercato di ristrutturare al meglio il modello relazionale per eliminare ridondanze all'interno del database al fine di ridurre il più possibile gli accessi allo stesso trovando quindi un compromesso tra memoria ed efficienza.

- **Sicurezza vs efficienza**
  Nel nostro sistema ogni richiesta del client viene validata mediante l'utilizzo di cookie e il controllo del livello di utenza. Questa caratteristica potrebbe far aumentare il tempo di risposta del sistema maggiormente se il carico è elevato, il sistema necessita di tali controlli per rispettare i requisiti iniziali del sistema.

- **Tempo di risposta vs Spazio di memoria**
  La scelta di utilizzare un DB relazionale è derivato dai diversi vantaggi che ne derivano:
    -Gestione consistente dei dati;
    - Tempo di risposta basso rispetto all'utilizzo del file system;
    - Accesso veloce e concorrente ai dati;
  Gli svantaggi nell'utilizzo di un DB relazionale sta nel fatto che richiede il triplo della memoria rispetto a un file system e si trovano difficoltà nel momento in cui si vuole gestire enormi quantità di dati.

- **Tempo di rilascio vs qualità**
  Nel caso in cui si presentino problemi durante l'implementazione del software o siano rilevati errori nel codice durante l'attività di testing, si è deciso di dare maggiore priorità alla qualità del software, piuttosto che al tempo di rilascio: nel caso in cui il software presenti dei bug, questi saranno corretti prima della data di rilascio.

- **Tempo di rilascio vs funzionalità**
  Nel caso si presentino problemi durante l'implementazione del software, o vengano rilevati errori durante la fase di testing, il software potrebbe essere rilasciato con meno funzionalità di quelle previste. Tali funzionalità saranno aggiunte dopo la data di rilascio.
  E' stata implementata completamente solo l'interfaccia utente, mentre tutte le funzionalità riguardanti il gestore degli ordini, dei menù e il fattorino sono state implementate solo a livello database.

2. Linee guida per l'implementazione

2.1 Pagine HTML

Le pagine HTML sono dei tipi di documenti digitali tramite i quali sono rese disponibili all'utente finale le informazioni del World Wide Web tramite un web browser. L'insieme di pagine contenute all'interno del sito Sandwich on web dovranno essere tra loro relazionate secondo una gerarchica e una struttura ipertestuale e saranno riferibili al web server locale.
Inoltre, le pagine del sito possono essere statiche o dinamico a seconda della presenza del database, le pagine statiche dovranno utilizzare l'indentazione, per facilitare la lettura, secondo le seguenti regole:

- Un'indentazione consiste in 4 spazi;
- Ogni tag deve avere un'indentazione maggiore o uguale del tag che lo contiene;
  - Non si dovrebbero superare gli 8 livelli di indentazione;
  - Qualora si superassero gli 8 livelli, è possibile azzerare il livello di indentazione, portandolo a 0 anziché incrementarlo;
- Sebbene l'uso dell'indentazione sia arbitrario, ogni tag di chiusura deve avere lo stesso livello di indentazione del corrispondente tag di apertura;

Accettabile

```
<div><span><nl>

<li>Uno</li>

<li>

Due

</li>

</nl></span></div>
```

<!Non

accettabile >

```
<div><span>

<nl>

<li>Uno</li>

<li>Due</li>

</nl></span>

</div>
```

- I tag di commento devono seguire le stesse regole che si applicano ai tag normali;
- Gli script e i fogli di stile incorporati, se non si completano in una sola riga di testo, devono partire dal livello 0 di indentazione.
  I fogli di stile e gli script incorporati nelle pagine devono rispettare le stesse linee guida che si applicano ai documenti dedicati. Le sezioni dinamiche (prodotte tramite l'oggetto out) dei documenti HTML generati da pagine JSP dovrebbero rispettare le stesse convenzioni dello HTML statico.

2.2 Script Javascript

Gli script che svolgono funzioni distinte dal mero rendering della pagina dovrebbero essere collocati in file dedicati.
Il codice Javascript deve seguire le stesse convenzioni per il layout e i nomi del codice PHP.
I documenti Javascript devono essere iniziati da un commento analogo a quello presente nei file PHP.
Le funzioni Javascript devono essere documentate in modo analogo ai metodi PHP, scritte possibilmente un file esterno, per consentire maggiore leggibilità al codice.
Gli oggetti Javascript devono essere preceduti da un commento in stile PHPdoc, che segue il seguente formato:

```
/**
*                   Descrizione breve
*                   Eventuale ulteriore descrizione
```

```
*                       Specifica degli argomenti del costruttore (@param)
*
*                       Metodo nomeMetodo1
*                       Descrizione breve
*                       Eventuale ulteriore descrizione
*                       Specifica degli argomenti (@param)
*                       Specifica dei risultati (@return)
*
*                       Metodo nomeMetodo2
*                       Descrizione breve
*                       Eventuale ulteriore descrizione
*                       Specifica degli argomenti (@param)
*                       Specifica dei risultati (@return)
*
*                       …
*/
function ClasseX(a, b, c) {
*
*
*
}
```

2.3 Fogli di stile CSS

Tutti gli stili non inline devono essere collocati in fogli di stile separati.
Ogni regola CSS deve essere formattata come segue:

- I selettori della regola si trovano a livello 0 di indentazione, uno per riga;
- L'ultimo selettore della regola è seguito da parentesi graffa aperta ({);
- Le proprietà che costituiscono la regola sono listate una per riga e sono indentate rispetto ai selettori, e per identificare la fine di un'istruzione si usa il punto e virgola;
- La regola è terminata da una parentesi graffa chiusa (}), collocata da sola su una riga;

Le proprietà e le regole poco chiare dovrebbero essere precedute da un commento esplicativo.
Le regole possono essere divise in blocchi concettualmente legati, preceduti da commenti in stile PHPdoc, che ne spiegano lo scopo, e seguiti da 2 righe bianche.

2.4 Database SQL

Le tabelle del database dovrebbero essere in terza forma normale di Codd (3NF). Qualora non lo fossero, la cosa deve essere documentata e motivata nello script di creazione del database.
I nomi delle tabelle devono seguire le seguenti regole:

- Sono costituiti da sole lettere maiuscole;
- Se il nome è costituito da più parole, queste sono separate da un underscore (_);
- Il nome deve essere un sostantivo singolare tratto dal dominio del problema ed esplicativo del contenuto.

I nomi dei campi devono seguire le seguenti regole:

- Sono costituiti da sole lettere minuscole;
- Se il nome è costituito da più parole, queste sono separate da un underscore (_);
- Il nome deve essere un sostantivo singolare tratto dal dominio del problema ed esplicativo del contenuto.

2.5 Istruzioni

2.5.1 Istruzioni semplici

Ogni linea deve contenere al massimo un'istruzione.

```
i++;          //Corretto
j++;          //Corretto
i++; j++;     //NO!
```

2.5.2 Istruzioni composte

Le istruzioni racchiuse dovrebbero essere indentate ad un ulteriore livello rispetto all'istruzione composta.

La parentesi graffa aperta dovrebbe stare alla fine della linea che inizia l'istruzione composta, mentre la parentesi graffa chiusa dovrebbe iniziare una linea ed essere indentata verticalmente con l'inizio dell'istruzione composta.
Le parentesi graffe vanno usate per tutte le istruzioni, anche quelle singole, quando sono parte di una struttura di controllo come nelle istruzioni if-else o for.

2.5.3 Istruzioni return

Un'istruzione return con un valore non dovrebbe usare parentesi, a meno che queste rendano in qualche modo il valore ritornato più ovvio.

2.5.4 Commenti

I programmi PHP possono avere due tipi di commenti: commenti d'implementazione e commenti di documentazione.
I commenti d'implementazione sono quelli classici del C++, che sono delimitati da /*...*/ e //.
I commenti di documentazione (noti anche come doc comments) sono esclusivi del PHP, e sono delimitati da /**...*/.
I doc comments possono essere estratti in file HTML utilizzando lo strumento PHPdoc.
I commenti di implementazione sono dei mezzi per commentare il codice o per commentare una particolare implementazione.
I doc comments vengono utilizzati per descrivere la specifica del codice da una prospettiva non implementativa, per essere letti da sviluppatori che non devono necessariamente avere il codice in mano.
I commenti dovrebbero essere usati per dare una panoramica del codice e per fornire informazioni aggiuntive che non sono prontamente disponibili nel codice stesso.
I commenti devono contenere solo informazioni rilevanti per leggere e comprendere il programma.
Ad esempio, informazioni su come il package corrispondente è costruito o in quale directory risiede non dovrebbero essere incluse in un commento.
La discussione sulle decisioni non banali o non ovvie è adatta, ma bisogna evitare di duplicare le informazioni che sono presenti in maniera chiara nel codice.
E' molto facile che commenti ridondanti diventino obsoleti; in generale, si dovrebbe evitare di inserire commenti suscettibili di diventare obsoleti con l'evoluzione del software.
La frequenza dei commenti talvolta riflette una povera qualità del codice.
Quando ci si sente obbligati ad aggiungere un commento, considerare il caso di riscrivere il codice per renderlo più chiaro.

I commenti non dovrebbero essere inclusi in grandi riquadri tracciati con asterischi o altri caratteri, né dovrebbero includere caratteri speciali come backspace.

2.6 Spazi bianchi

2.6.1 Linee bianche

Due linee bianche dovrebbero essere sempre usate nelle seguenti circostanze:

- Fra sezioni di un file sorgente;
- Fra definizioni di classe e interfaccia;

Una linea bianca dovrebbe essere sempre usata nelle seguenti circostanze:

- Fra metodi;
- Fra le variabili locali in un metodo e la sua prima istruzione;
- Prima di un commento di blocco o a singola linea;
- Fra sezioni logiche all'interno di un metodo.

2.6.2 Spazi bianchi

- Spazi bianchi dovrebbero essere usati nelle seguenti circostanze:
- Una parola chiave seguita da una parentesi dovrebbe essere separata da uno spazio;
- Uno spazio bianco non dovrebbe essere usato fra il nome di un metodo e le sue parentesi d'apertura;
- Uno spazio bianco dovrebbe essere interposto dopo le virgole nelle liste di argomenti;

- Tutti gli operatori binari eccetto l'operatore punto dovrebbero essere separati dai loro operandi tramite spazi. Gli spazi bianchi non dovrebbero mai separare gli operatori unari come l'operatore meno, l'incremento e il decremento.

2.7 Convenzioni di nomi

2.7.1 Metodi

I nomi dei metodi devono essere verbi con iniziale minuscola seguendo lo stile del camel case.
Cercare di rendere i nomi dei metodi semplici, descrittivi e che rispettino il dominio applicativo. I nomi dei metodi non devono iniziare con caratteri di underscore o di dollaro.
Usare parole intere evitando acronimi e abbreviazioni (a meno che l'abbreviazione sia più usata della forma lunga, come URL o HTML). Non dovrebbero essere usati underscore per legare nomi.

2.7.2 Variabili

Tutte le variabili e le istanze di classe devono essere scritte con iniziale minuscola e a gobba di cammello. I nomi delle variabili devono essere in inglese o in italiano. Non devono iniziare con caratteri di underscore o dollaro. La scelta di un nome deve essere mnemonica e deve rispettare il dominio applicativo.
I nomi di variabili di un solo carattere dovrebbero essere evitati.

2.7.3 Costanti

I nomi delle variabili dichiarate come costanti di classe devono essere scritte in lettere tutte maiuscole con le parole separate da underscore. I nomi delle costanti devono essere in inglese. La scelta di un nome deve essere mnemonica e deve rispettare il dominio applicativo.

2.8 Consuetudini di programmazione

2.8.1 Fornire accesso a variabili di istanza o di classe

Non rendere pubblica una variabile di istanza o di classe senza una buona ragione. Le variabili di istanza devono essere scritte o lette attraverso delle chiamate a metodi.

2.8.2 Riferire variabili e metodi di classe

Evitare di usare un oggetto per accedere a variabili o metodi di classe static. Usare invece il nome della classe.

2.8.3 Assegnamento di variabili

Evitare di assegnare a più variabili lo stesso valore in una sola istruzione.
Non usare l'operatore di assegnamento (=) in un punto in cui può essere facilmente confuso con l'operatore di uguaglianza (==).
Non usare assegnamenti innestati nel tentativo di migliorare le prestazioni a tempo di esecuzione.

2.8.4 Parentesi

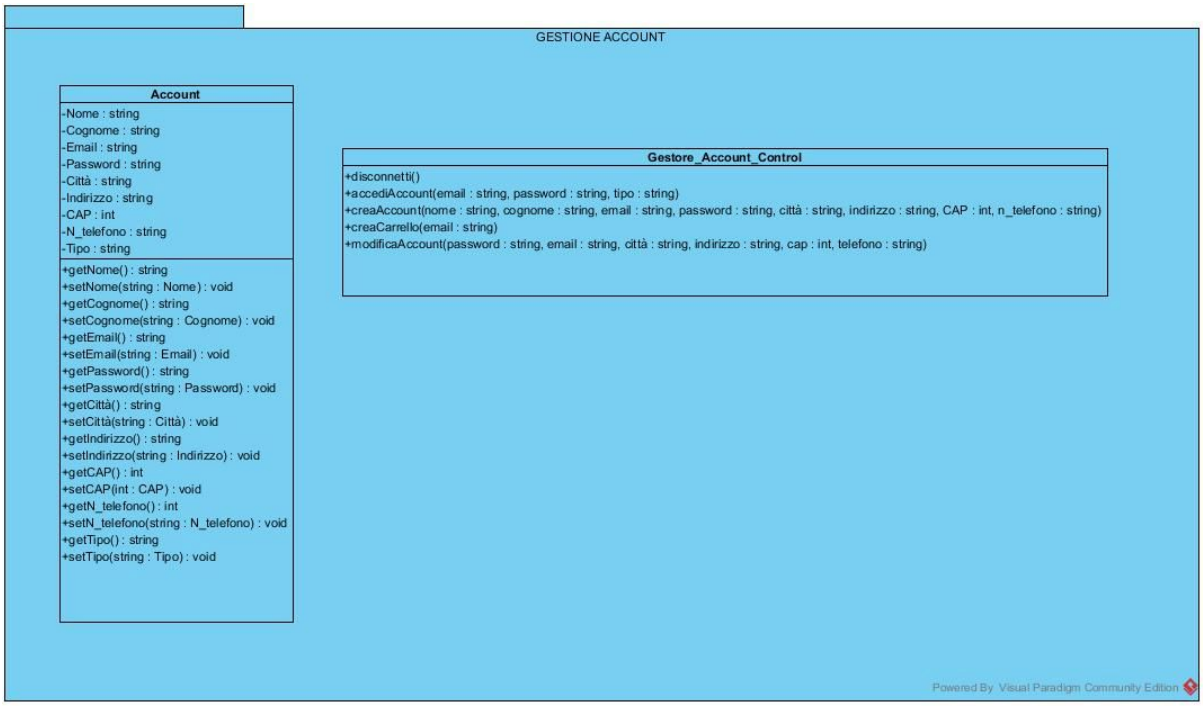
E' generalmente una buona idea usare le parentesi liberamente in espressioni che coinvolgono operatori misti per evitare problemi di precedenza degli operatori.

2.8.5 Valori ritornati

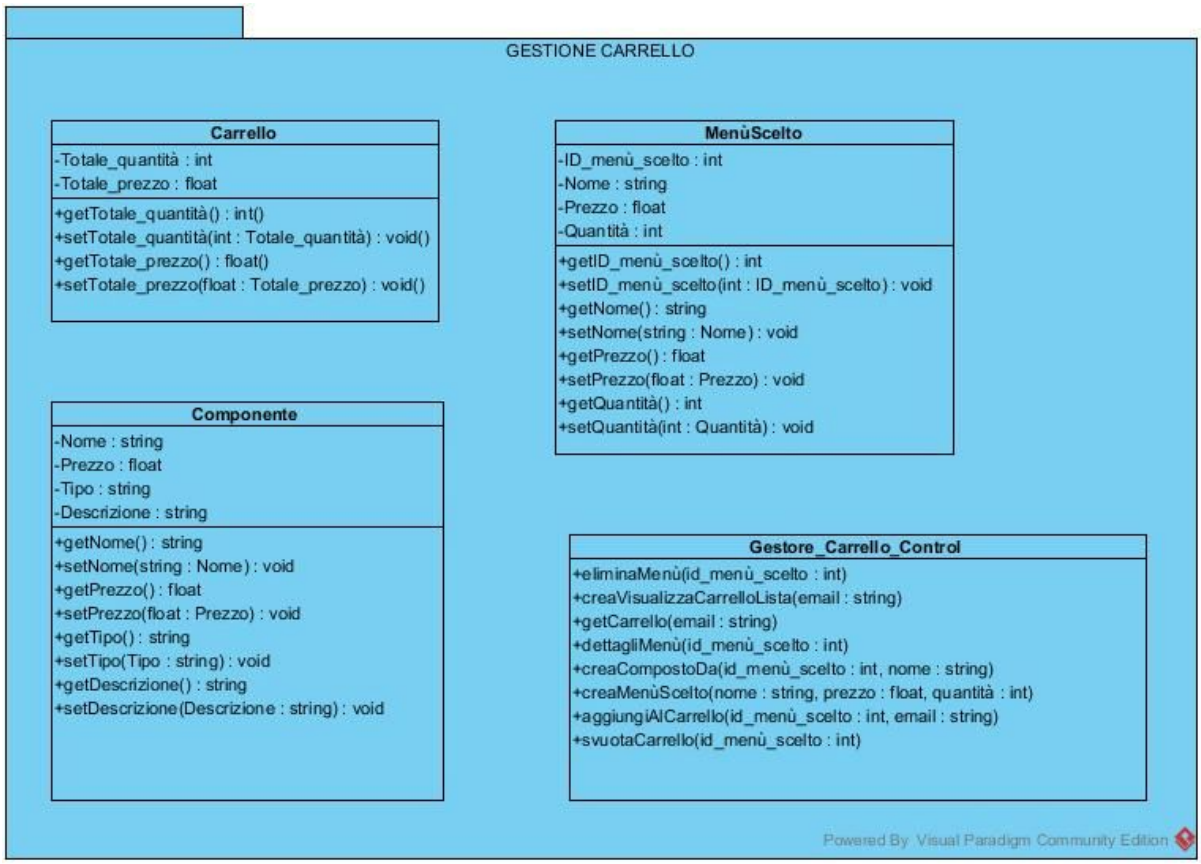
Provare a rendere la struttura del programma aderente alle proprie intenzioni.

3. PACKAGES

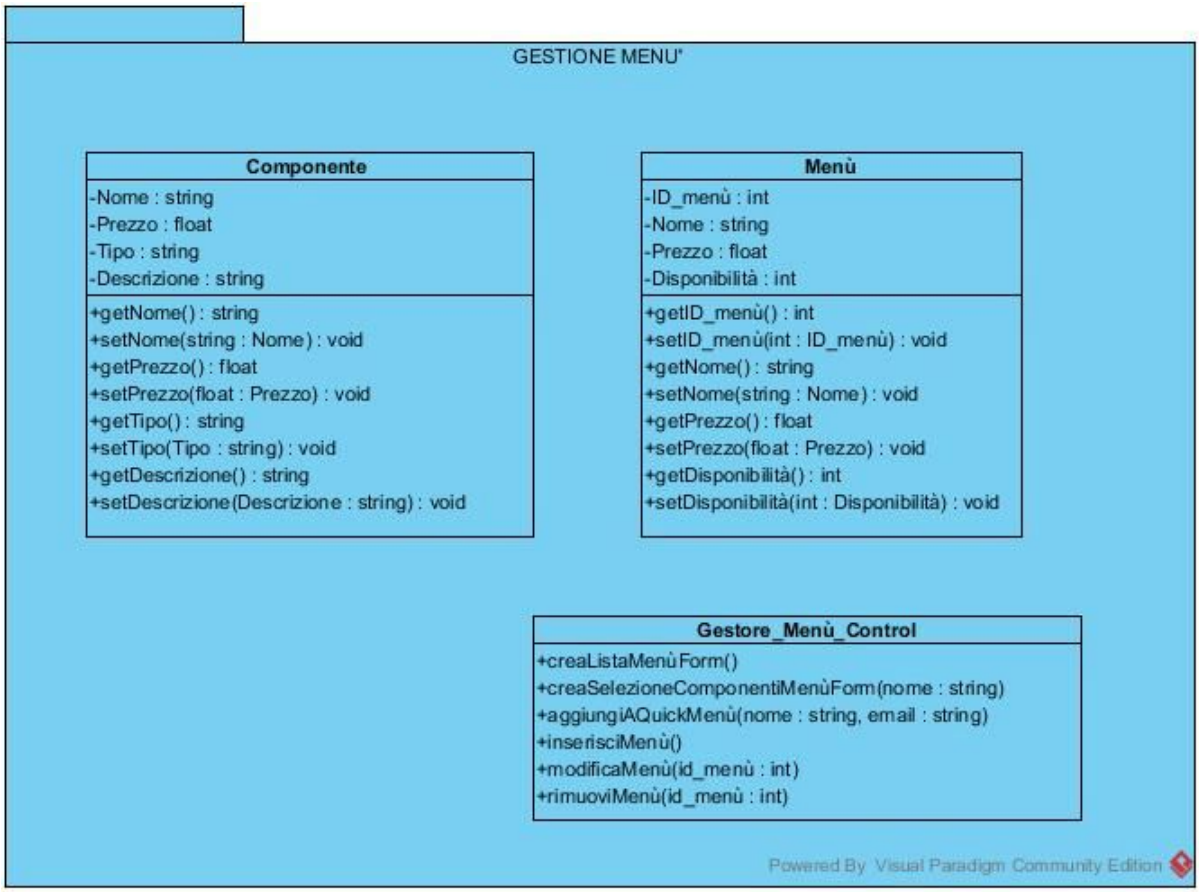
3.1 PK_1-Gestione Account



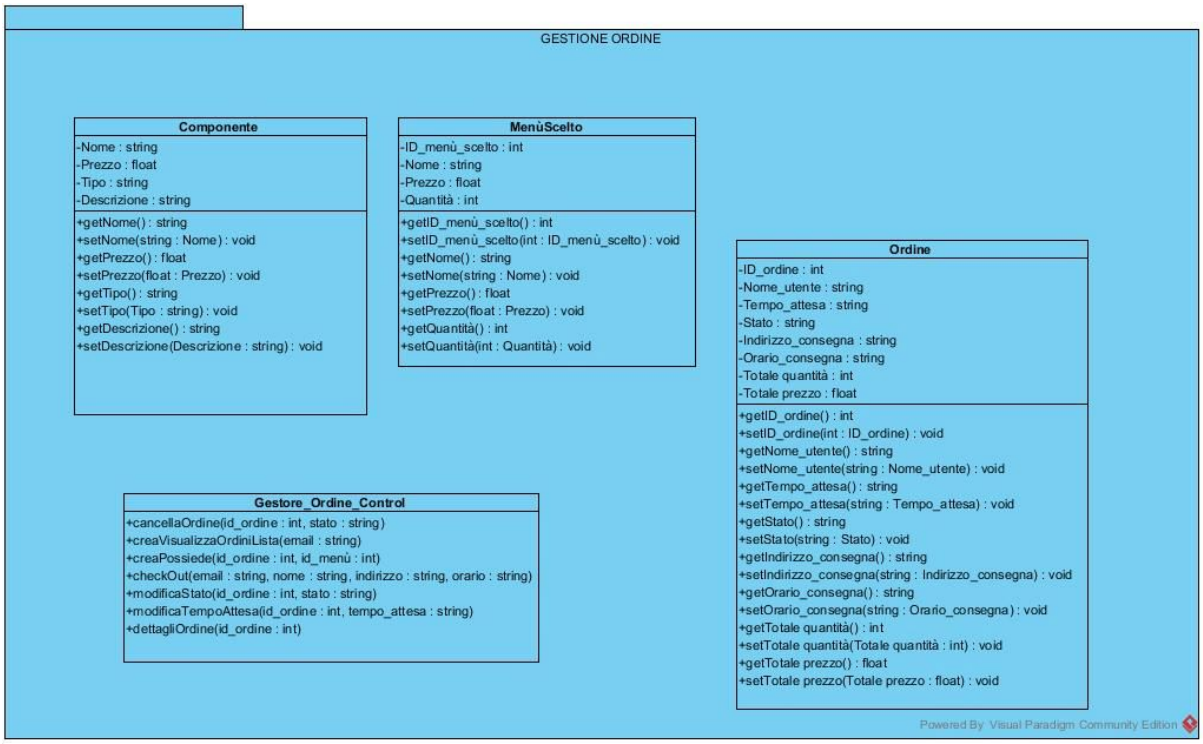
3.2 PK_2-Gestione Carrello



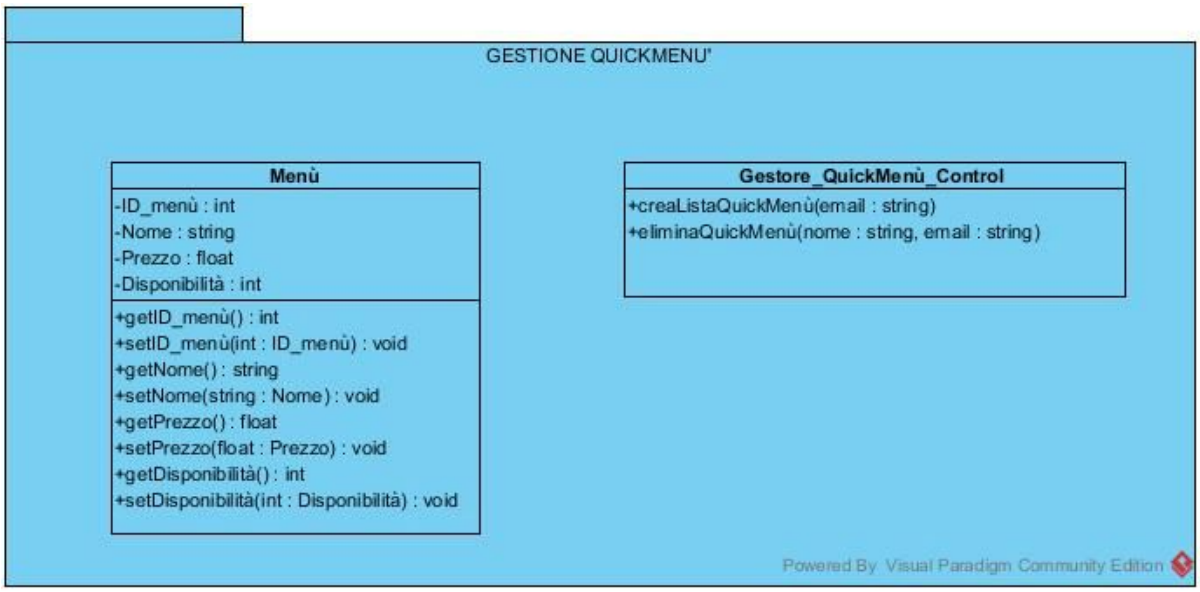
3.3 PK_3-Gestione Menù



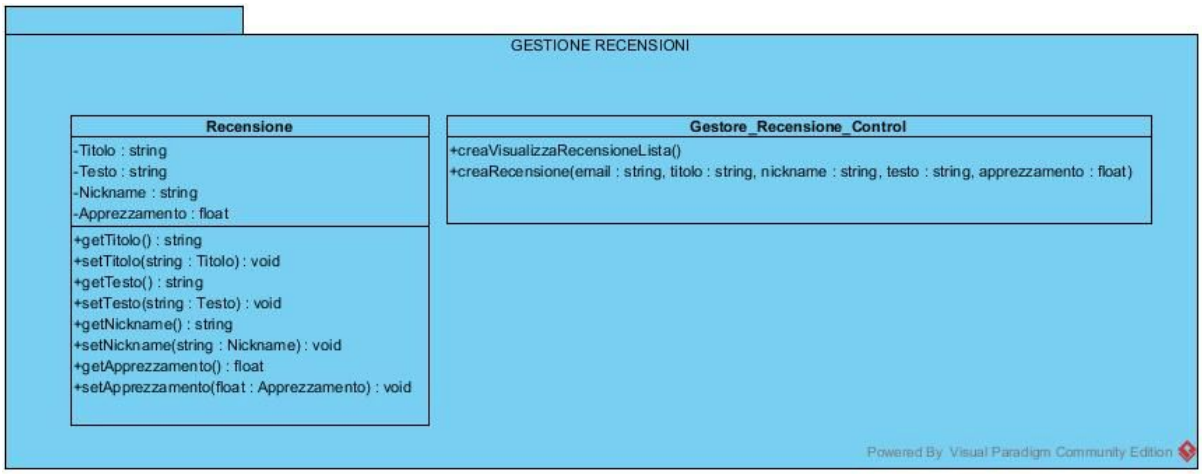
3.4 PK_4-Gestione Ordine



3.5 PK_5-Gestione QuickMenù



3.6 PK_6-Gestione Recensioni



4. CLASS INTERFACES

4.1-Classe Menù

Nome classe	Menù
Descrizione	Questa classe rappresenta l'astrazione di un menù
Pre-condizioni	**context** Menù : : getNome() **pre:** Nome != null **context** Menù : : setNome(Nome) **pre:** Nome != null **context** Menù : :getPrezzo() **pre:** Prezzo > 0 **context** Menù : : setPrezzo(Prezzo) **pre:** Prezzo > 0 **context** Menù : : getDisponibilità() **pre:** Disponibilità > 0 **context** Menù : : setDisponibilità(Disponibilità) **pre:** Disponibilità > 0
	context Menù : : setNome() **post:** Nome != null

Post-condizioni	**context** Menù : : setPrezzo() **post:** Prezzo > 0 **context** Menù : : setDisponibilità() **post:** Disponibilità > 0
Invarianti	**context** Menù **inv:** Nome != null and Prezzo > 0 and Disponibilità > 0

4.2-Classe Ordine

Nome classe	Ordine
Descrizione	Questa classe rappresenta l'astrazione di un ordine
Pre-condizioni	**context** Ordine : : getID_ordine() **pre:** ID_ordine > 0 **context** Ordine : : setID_ordine(ID_ordine) **pre:** ID_ordine > 0 **context** Ordine : : getNome_utente() **pre:** Nome_utente != null **context** Ordine : : setNome_utente(Nome_utente) **pre:** Nome_utente != null **context** Ordine : : getTempo_attesa() **pre:** Tempo_attesa != null **context** Ordine **: :** setTempo_attesa(Tempo_attesa) **pre:** Tempo_attesa != null **context** Ordine : : getStato() **pre:** Stato != null

	context Ordine : : setStato(Stato) **pre:** Stato != null **context** Ordine : : getIndirizzo_consegna() **pre:** Indirizzo_consegna != null **context** Ordine : : setIndirizzo_consegna(Indirizzo_consegna) **pre:** Indirizzo_consegna != null **context** Ordine : :getOrario_consegna() **pre:** Orario_consegna != null **context** Ordine : :setOrario_consegna (Orario_consegna) **pre:** Orario_consegna != null **context** Ordine : :getTotale_quantità() **pre:** Totale_quantità > 0 **context** Ordine : :setTotale_quantità (Totale_quantità) **pre:** Totale_quantità > 0 **context** Ordine : :getTotale_prezzo() **pre:** Totale_prezzo > 0 **context** Ordine : :setTotale_prezzo (Totale_prezzo) **pre:** Totale_prezzo > 0
Post-condizioni	**context** Ordine : : setNome_utente(Nome_utente) **post:** Nome_utente != null **context** Ordine : : setTempo_attesa(Tempo_attesa) **post:** Tempo_attesa != null **context** Ordine : :setStato(Stato) **post:** Stato != null

	context Ordine : : setIndirizzo_consegna(Indirizzo_consegna) **post:** Indirizzo_consegna != null **context** Ordine : : setOrario_consegna(Orario_consegna) **post:** Orario_consegna != null
Invarianti	**context** Ordine **inv:** ID_ordine > 0 and Nome_utente != null and Tempo_attesa != null and Stato != null and Indirizzo_consegna != null and Orario_consegna != null

4.3-Classe Recensione

Nome classe	Recensione
Descrizione	Questa classe rappresenta l'astrazione di una recensione
Pre-condizioni	**context** Recensione : : getTitolo() **pre:** Titolo != null **context** Recensione : : setTitolo(Titolo) **pre:** Titolo != null **context** Recensione : : getTesto() **pre:** Testo != null **context** Recensione : : setTesto(Testo) **pre:** Testo != null **context** Recensione : : getNickname() **pre:** Nickname != null

	context Recensione : : setNickname(Nickname) **pre:** Nickname != null **context** Recensione : : getApprezzamento() **pre:** Apprezzamento > 0 **context** Recensione : : setApprezzamento(Apprezzamento) **pre**: Apprezzamento > 0
Post-condizioni	**context** Recensione : : setTitolo() **post:** Titolo != null **context** Recensione : : setTesto() **post:** Testo != null **context** Recensione : : setNickname() **post:** Nickname != null **context** Recensione : : setApprezzamento() **post:** Apprezzamento > 0
Invarianti	**context** Recensione **inv:** and Titolo != null and Testo != null and Nickname != null and Apprezzamento > 0

4.4-Classe Carrello

Nome classe	Carrello
Descrizione	Questa classe rappresenta l'astrazione di un carrello
Pre-condizioni	**context** Carrello: :getTotale_quantità() **pre:**

	Totale_quantità > 0 **context** Carrello: : setTotale_quantità(Totale_quantità) **pre:** Totale_quantità > 0 **context** Carrello : :getTotale_prezzo() **pre:** Totale_prezzo > 0 **context** Carrello : : setTotale_prezzo(Totale_prezzo) **pre:** Totale_prezzo > 0
Post-condizioni	**context** Carrello: : setTotale_quantità() **post:** Totale_quantità > 0 **context** Carrello : : setTotale_prezzo() **post:** Totale_prezzo > 0
Invarianti	**context** Carrello **inv:** Totale_prezzo > 0 and Totale_quantità > 0

4.5-Classe Menù Scelto

Nome classe	Menù scelto
Descrizione	Questa classe rappresenta l'astrazione di un menù scelto
Pre-condizioni	**context** MenùScelto : : getID_menù_scelto() **pre:** ID_menù_scelto > 0 **context** MenùScelto : : setID_menù_scelto(ID_menù_scelto) **pre:** ID_menù_scelto > 0 **context** MenùScelto : : getNome() **pre:** Nome != null **context** MenùScelto : : setNome(Nome) **pre:** Nome != null **context** MenùScelto : :getPrezzo() **pre:** Prezzo > 0 **context** MenùScelto : : setPrezzo(Prezzo) **pre:** Prezzo > 0 **context** MenùScelto : : getQuantità() **pre:** Quantità > 0 **context** MenùScelto : :setQuantità(Quantità) **pre:** Quantità> 0
Post-condizion	**context** MenùScelto : : setID_menù_scelto() **post:** ID_menù_scelto > 0

i	**context** MenùScelto : : setNome() **post:** Nome != null **context** MenùScelto : : setPrezzo() **post:** Prezzo > 0 **context** MenùScelto : : setQuantità() **post:** Quantità> 0
Invarianti	**context** MenùScelto **inv:** ID_menù_scelto > 0 and Nome != null and Prezzo > 0 and Quantità > 0

4.6-Classe Account

Nome classe	Account
Descrizione	Questa classe rappresenta l'astrazione di un account
Pre-condizioni	**context** Account : :getNome() **pre:** Nome != null **context** Account : :setNome(Nome) **pre:** Nome != null **context** Account : :getCognome() **pre:** Cognome != null **context** Account : :setCognome(Cognome) **pre:** Cognome != null **context** Account : :getEmail() **pre:** Email != null **context** Account : : setEmail(Email) **pre:** Email !=

	null **context** Account : : getPassword() **pre:** Password != null **context** Account : : setPassword(Password) **pre:** Password != null **context** Account : :getCittà() **pre:** Città != null **context** Account : : setCittà(Città) **pre:** Città != null **context** Account : : getIndirizzo() **pre:** Indirizzo != null **context** Account : : setIndirizzo(Indirizzo) **pre:** Indirizzo != null **context** Account : : getCAP() **pre:** CAP > 0 **context** Account : : setCAP(CAP) **pre:** CAP > 0 **context** Account : : getN_telefono() **pre:** N_telefono != null **context** Account : : setN_telefono(N_telefono) **pre:** N_telefono != null **context** Account : : getTipo() **pre:** Tipo != null **context** Account : : setTipo(Tipo) **pre:** Tipo != null
	context Account : :setNome() **post:**Nome != null

Post-condizioni	**context** Account : :setCognome() **post:** Cognome != null **context** Account : : setEmail() **post:** Email != null **context** Account : : setPassword() **post:** Password != null **context** Account : : setCittà() **post:** Città != null **context** Account : : setIndirizzo() **post:** Indirizzo != null **context** Account : : setCAP() **post:** CAP > 0 **context** Account : : setN_telefono() **post:** N_telefono!= null **context** Account : : setTipo() **post:** Tipo != null
Invarianti	**context** Account **inv:** ID_account > 0 and Nome != null and Cognome != null and Email != null and Password !=null and Città !=null and Indirizzo != null and CAP > 0 and N_telefono!= null and Tipo != null

4.7-Classe Componente

Nome classe	Componente

Descrizione	Questa classe rappresenta l'astrazione di un componente
Pre-condizioni	**context** Componente : : getNome() **pre:** Nome != null **context** Componente : : setNome(Nome) **pre:** Nome != null **context** Componente : : getTipo() **pre:** Tipo != null **context** Componente : :setTipo(Tipo) **pre:**Tipo != null **context** Componente : : getDescrizione() **pre:** Descrizione != null **context** Componente : :setDescrizione(Descrizione) **pre:** Descrizione != null **context** Componente : :getPrezzo() **pre:** Prezzo > 0 **context** Componente : : setPrezzo(Prezzo) **pre:** Prezzo > 0
Post-condizioni	**context** Componente : :setNome() **post:** Nome != null **context** Componente : : setTipo() **post:** Tipo != null **context** Componente : : setDescrizione() **post:** Descrizione != null **context** Componente : :setPrezzo() **post:** Prezzo > 0
Invarianti	**context** Componente **inv:** Nome != null and Tipo != null and Descrizione != null and Prezzo > 0

4.8 Classe Gestore_Account_Control

Nome classe	Gestore_Account_Control
Descrizione	Questa classe rappresenta l'astrazione di un generico account.
Pre-condizioni	**context** Gestore_Account_Control: : disconnetti() **pre:** disconnetti != null **context** Gestore_Account_Control: : accediAccount (Email,Password,Tipo) **pre:** Email != null and Password != null and Tipo != null **context** Gestore_Account_Control: : creaAccount(Nome,Cognome,Email,Password,Città, Indirizzo,CAP,N_Telefono) **pre:** Nome != null and Cognome != null and Email != null and Password != null and Città != null and Indirizzo != null and CAP > 0 and N_Telefono != null **context** Gestore_Account_Control: : creaCarrello(Email) **pre:** Email != null **context** Gestore_Account_Control: : modificaAccount(Email,Password,Città,Indirizzo,CAP, N_Telefono) **pre:** Email != null and Password != null and Città != null and Indirizzo != null and CAP > 0 and N_Telefono != null
	context Gestore_Account_Control: : disconnetti **post:**

Post-condizioni	disconnetti != null **context** Gestore_Account_Control: : accediAccount (Email,Password,Tipo) **post:** Email != null and Password != null and Tipo != null **context** Gestore_Account_Control: : creaAccount(Nome,Cognome,Email,Password,Città, Indirizzo,CAP,N_Telefono) **post:** Nome != null and Cognome != null and Email != null and Password != null and Città != null and Indirizzo != null and CAP > 0 and N_Telefono != null **context** Gestore_Account_Control: : creaCarrello(Email) **post:** Email != null **context** Gestore_Account_Control: : modificaAccount(Email,Password,Città,Indirizzo,CAP, N_Telefono) **post:** Email!= null and Password != null and Città != null and Indirizzo != null and CAP > 0 and N_Telefono != null
Invarianti	**context** Gestore_Account_Control **inv:** Nome != null and Cognome != null and Email != null and Password != null and Città != null and Indirizzo != null and CAP > 0 and N_Telefono != null and Tipo != null

4.9 Classe Gestore_Carrello_Control

Nome classe	Gestore_Carrello_Control

Descrizione	Questa classe rappresenta l'astrazione di un generico carrello.
Pre-condizioni	**context** Gestore_Carrello_Control: : eliminaMenù (ID_menù) **pre:** id_menù > 0 **context** Gestore_Carrello_Control: : svuotaCarrello(ID_menù_scelto) **pre:** ID_menù_scelto > 0 **context** Gestore_Carrello_Control: : creaVisualizzaCarrelloLista (Email) **pre:** Email != null **context** Gestore_Carrello_Control: : getCarrello(Email) **pre:** Email != null **context** Gestore_Carrello_Control: : creaCarrello(Email) **pre:** Email != null **context** Gestore_Carrello_Control: : creaCompostaDa(ID_menù_scelto, Nome) **pre:** ID_menù_scelto > 0 and Nome != null **context** Gestore_Carrello_Control: : creaMenùScelto(Nome,Prezzo,Quantità) **pre:** Nome != null and Prezzo > 0 and Quantità > 0 **context** Gestore_Carrello_Control: : aggiungiAlCarrello(ID_menù_scelto, Email) **pre:** ID_menù_scelto > 0 and Email != null **context** Gestore_Carrello_Control: : dettagliMenù(ID_menù_scelto) **pre:** ID_menù_scelto > 0

Post-condizioni	**context** Gestore_Carrello_Control: : eliminaMenù (ID_menù) **post:** id_menù > 0 **context** Gestore_Carrello_Control: : creaVisualizzaCarrelloLista (Email) **post:** Email != null **context** Gestore_Carrello_Control: : getCarrello(Email) **post:** Email != null **context** Gestore_Carrello_Control: : creaCarrello(Email) **post:** Email != null **context** Gestore_Carrello_Control: : creaCompostaDa(ID_menù_scelto, Nome) **post:** ID_menù_scelto > 0 and Nome != null **context** Gestore_Carrello_Control: : creaMenùScelto(Nome,Prezzo,Quantità) **post:** Nome != null and Prezzo > 0 and Quantità > 0 **context** Gestore_Carrello_Control: : aggiungiAlCarrello(ID_menù_scelto, Email) **post:** ID_menù_scelto > 0 and Email != null **context** Gestore_Carrello_Control: : svuotaCarrello(ID_menù_scelto) **post:** ID_menù_scelto > 0 **context** Gestore_Carrello_Control: : dettagliMenù(ID_menù_scelto) **post:** ID_menù_scelto > 0
Invarianti	**context** Gestore_Carrello_Control **inv:** ID_menù > 0

	and Email != null and Nome != null and Prezzo > 0 and Quantità > 0 and ID_menù_scelto > 0

4.10 Classe Gestore_Menù_Control

Nome classe	Gestore_Menù_Control
Descrizione	Questa classe rappresenta l'astrazione di un generico menù.
Pre-condizioni	**context** Gestore_Menù_Control: : creaListaMenùForm() **pre:** creaListaMenùForm != null **context** Gestore_Menù_Control: creaSelezioneComponentiMenùForm (Nome) **pre:** Nome != null **context** Gestore_Menù_Control: : aggiungiAQuickMenù(Nome,Email) **pre:** Nome != null and Email != null **context** Gestore_Menù_Control: : inserisciMenù() **pre:** inserisciMenù != null **context** Gestore_Menù_Control: : modificaMenù(ID_menù) **pre:** ID_menù > 0 **context** Gestore_Menù_Control: : rimuoviMenù(ID_menù) **pre:** ID_menù > 0

Post-condizioni	**context** Gestore_Menù_Control: : creaListaMenùForm() **post:** creaListaMenùForm != null **context** Gestore_Menù_Control: creaSelezioneComponentiMenùForm (Nome) **post:** Nome != null **context** Gestore_Menù_Control: : aggiungiAQuickMenù(Nome,Email) **post:** Nome != null and Email != null **context** Gestore_Menù_Control: : inserisciMenù() **post:** inserisciMenù != null **context** Gestore_Menù_Control: : modificaMenù(ID_menù) **post:** ID_menù > 0 **context** Gestore_Menù_Control: : rimuoviMenù(ID_menù) **post:** ID_menù > 0
Invarianti	**context** Gestore_Menù_Control **inv:** ID_menù > 0 and Email != null and Nome != null

4.11 Classe Gestore_Ordine_Control

Nome classe	Gestore_Ordine_Control
Descrizione	Questa classe rappresenta l'astrazione di un generico ordine.

Pre-condizioni	**context** Gestore_Ordine_Control: : cancellaOrdine(ID_ordine,Stato) **pre:** ID_ordine > 0 and Stato != null **context** Gestore_Ordine_Control: creaVisualizzaOrdiniLista(Email) **pre:** Email != null **context** Gestore_Ordine_Control: : creaPossiede(ID_ordine,ID_menù) **pre:** ID_ordine > 0 and ID_menù > 0 **context** Gestore_Ordine_Control: : checkOut(Email,Nome,Indirizzo,Orario) **pre:** Email != null and Nome != null and Indirizzo != null and Orario != null **context** Gestore_Ordine_Control: : modificaStato(ID_ordine,Stato) **pre:** ID_ordine > 0 and Stato != null **context** Gestore_Ordine_Control: : modificaTempoAttesa(ID_ordine,TempoAttesa) **pre:** ID_ordine > 0 and TempoAttesa != null **context** Gestore_Ordine_Control: : dettagliOrdine(ID_ordine) **pre:** ID_ordine > 0
Post-condizioni	**context** Gestore_Ordine_Control: : cancellaOrdine(ID_ordine,Stato) **post:** ID_ordine > 0 and Stato != null **context** Gestore_Ordine_Control: creaVisualizzaOrdiniLista(Email) **post:** Email != null **context** Gestore_Ordine_Control: :

	creaPossiede(ID_ordine,ID_menù) **post:** ID_ordine > 0 and ID_menù > 0 **context** Gestore_Ordine_Control: : checkOut(Email,Nome,Indirizzo,Orario) **post:** Email != null and Nome != null and Indirizzo != null and Orario != null **context** Gestore_Ordine_Control: : modificaStato(ID_ordine,Stato) **post:** ID_ordine > 0 and Stato != null **context** Gestore_Ordine_Control: : modificaTempoAttesa(ID_ordine,TempoAttesa) **post:** ID_ordine > 0 and TempoAttesa != null **context** Gestore_Ordine_Control: : dettagliOrdine(ID_ordine) **post:** ID_ordine > 0
Invarianti	**context** Gestore_Ordine_Control **inv:** ID_ordine > 0 and Email != null and ID_menù > 0 and Nome != null and Indirizzo != null and Orario != null and Stato != null and TempoAttesa != null

4.12 Classe Gestore_QuickMenù_Control

Nome classe	Gestore_QuickMenù_Control
Descrizione	Questa classe rappresenta l'astrazione di un generico quick menù.

Pre-condizioni	**context** Gestore_QuickMenù_Control: : creaListaQuickMenù(Email) **pre:** Email != null **context** Gestore_QuickMenù_Control: eliminaQuickMenù(Nome,Email) **pre:** Nome != null and Email != null
Post-condizioni	**context** Gestore_QuickMenù_Control: : creaListaQuickMenù(Email) **post:** Email != null **context** Gestore_QuickMenù_Control: eliminaQuickMenù(Nome,Email) **post:** Nome != null and Email != null
Invarianti	**context** Gestore_QuickMenù_Control **inv:** Email != null and Nome != null

4.13 Classe Gestore_Recensione_Control

Nome classe	Gestore_Recensione_Control
Descrizione	Questa classe rappresenta l'astrazione di un generico quick menù.
Pre-condizioni	**context** Gestore_Recensione_Control: : creaVisualizzaRecensioneLista() **pre:** creaVisualizzaRecensioneLista != null **context** Gestore_Recensione_Control: creaRecensione(Email,Titolo,Nickname,Testo,Apprezzamento) **pre:** Email != null and Titolo != null and Nickname != null and Testo != null and

	Apprezzamento > 0
Post-condizioni	**context** Gestore_Recensione_Control: : creaVisualizzaRecensioneLista() **post:** creaVisualizzaRecensioneLista != null **context** Gestore_Recensione_Control: creaRecensione(Email,Titolo,Nickname,Testo,Apprezzamento) **post:** Email != null and Titolo != null and Nickname != null and Testo != null and Apprezzamento> 0
Invarianti	**context** Gestore_Recensione_Control **inv:** Email != null and Titolo != null and Nickname != null and Testo != null and Apprezzamento> 0

5. DOCUMENTAZIONE DELLE CLASSI

Sandwich On Web - Documentazione

Account	Questa classe rappresenta un'astrazione di un account.
Account_Test	Classe Test in riferimento alla classe Account.
Carrello	Questa classe rappresenta un'astrazione di un carrello.
Carrello_Test	Classe Test in riferimento alla classe Carrello.

Componente	Questa classe rappresenta un'astrazione di una componente.
Componente_Test	Classe Test in riferimento alla classe Componente.
Connection	Classe che contiene i dati necessari per effettuare la connessione con il database.
Gestore_Account_Control	Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di un account.
Gestore_Account_Control_Test	Classe Test in riferimento alla classe Gestore_Account_Control.
Gestore_Carrello_Control	Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di un carrello.
Gestore_Carrello_Control_Test	Classe Test in riferimento alla classe Gestore_Carrello_Control.
Gestore_Menu_Control	Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di un menù.
Gestore_Menu_Control_Test	Classe Test in riferimento alla classe Gestore_Account_Control.

Gestore_Ordine_Control	Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di un ordine.
Gestore_Ordine_Control_Test	Classe Test in riferimento alla classe Gestore_Ordine_Control.
Gestore_QuickMenu_Control	Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di un menù tra i preferiti.
Gestore_QuickMenu_Control_Test	Classe Test in riferimento alla classe Gestore_QuickMenu_Control.
Gestore_Recensione_Control	Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di una recensione.
Gestore_Recensione_Control_Test	Classe Test in riferimento alla classe Gestore_Recensione_Control.
Menu	Questa classe rappresenta un'astrazione di un menù.
Menu_Scelto	Questa classe rappresenta un'astrazione di un menù che è stato scelto.
Menu_Scelto_Test	Classe Test in riferimento alla classe Menu_Scelto.

Menu_Test	Classe Test in riferimento alla classe Menù.
Ordine	Questa classe rappresenta un'astrazione di un ordine.
Ordine_Test	Classe Test in riferimento alla classe Ordine.
Recensione	Questa classe rappresenta un'astrazione di una recensione.
Recensione_Test	Classe Test in riferimento alla classe Recensione.

Di seguito è riportata la documentazione relativa alle classi implementate, creata usando PHPdoc, e le classi di test associate.

5.1 Class Account

Questa classe rappresenta un'astrazione di un account.

Located at File PHP/Gestione Account/Account.php

public	**__construct(type $nome, type $cognome, type $email, type $pass, type $citta, type $indirizzo, type $cap, type $telefono, type $tipo)** **Metodo costruttore per istanziare un oggetto di tipo Account** Parameters **$nome** **nome dell'utente.** **$cognome** **cognome dell'utente.**

	$email email univoca dell'utente. **$pass** password dell'utente. **$citta** città dell'utente. **$indirizzo** indirizzo dell'utente. **$cap** cap associato alla città dell'utente. **$telefono** telefono dell'utente **$tipo** tipo di utente.
public type	**__get(type $property)** Metodo che ritorna il valore di una determinata proprietà. Parameters **$property** nome proprietà Returns **type** valore della proprietà
public type	**__set(type $property, type $value)** Metodo che cambia il valore di una determinata proprietà, ritornando il valore precedente di essa. Parameters **$property** nome della proprietà da cambiare **$value**

	nuovo valore della proprietà Returns type valore precedente della proprietà
public string	__toString( ) Metodo che ritorna il valore di tutte le proprietà dell'oggetto Account in un formato stabilito. Returns string ritorna il valore di tutte le proprietà dell'oggetto

5.1.1 Class Account_Test

Classe Test in riferimento alla classe Account.

PHPUnit_Framework_TestCase

AccountTest

See: Account

Located at test/File PHP/Gestione Account/AccountTest.php

protected	setUp( ) Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.
protected	tearDown( ) Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.

public	test__get( ) Covers Account::__get
public	test__set( ) Covers Account::__set
public	test__toString( ) Covers Account::__toString

protected Account	$object Riferimento globale all'oggetto di tipo Account

5.2 Class Carrello

Questa classe rappresenta un'astrazione di un carello.

Located at File PHP/Gestione Carrello/Carrello.php

public	__construct(type $totale_prezzo, type $totale_quantità) Metodo costruttore per istanziare un oggetto di tipo Carrello Parameters $totale_prezzo totale prezzo del carrello $totale_quantità totale quantità carrello
public type	__get(type $property) Metodo che ritorna il valore di un determinato proprietà. Parameters $property nome proprietà Returns type valore della proprietà
public type	__set(type $property, type $value) Metodo che cambia il valore di un determinata proprietà, ritornando il valore precedente di essa.

	Parameters **$property** **nome della proprietà da cambiare** **$value** **nuovo valore della proprietà** Returns **type** **valore precedente della proprietà**
public string	__toString( ) **Metodo che ritorna il valore di tutte le proprietà dell'oggetto Carrello in un formato stabilito.** Returns **string** **ritorna il valore di tutte le proprietà dell'oggetto**

5.2.1 Class CarrelloTest

Classe Test in riferimento alla classe Carrello.

PHPUnit_Framework_TestCase

CarrelloTest

See: Carrello

Located at test/File PHP/Gestione Carrello/CarrelloTest.php

protected	**setUp( )** **Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.**
protected	**tearDown( )** **Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.**
public	**test__get( )** Covers **Carrello::__get**
public	**test__set( )** Covers **Carrello::__set**
public	**test__toString( )** Covers **Carrello::__toString**

protected Carrello	$object

5.3 Class Componente

Questa classe rappresenta un'astrazione di una componente.

Located at File PHP/Gestione Menu/Componente.php

public	__construct(type $nome, type $tipo, type $descrizione, type $prezzo) Metodo costruttore per istanziare un oggetto di tipo Componente. Parameters $nome nome della componente $tipo tipo della componente $descrizione descrizione della componente $prezzo

	prezzo associato alla componente
public type	__get(type $property) Metodo che ritorna il valore di un determinato proprietà. Parameters $property nome proprietà Returns type valore della proprietà
public type	__set(type $property, type $value) Metodo che cambia il valore di un determinata proprietà, ritornando il valore precedente di essa. Parameters $property nome della proprietà da cambiare $value nuovo valore della proprietà Returns

	type **valore precedente della proprietà**
public string	**__toString( )** **Metodo che ritorna il valore di tutte le proprietà dell'oggetto Componente in un formato stabilito.** Returns **string** **ritorna il valore di tutte le proprietà dell'oggetto**

5.3.1 Class Componente_Test

Classe Test in riferimento alla classe Componente.

PHPUnit_Framework_TestCase

ComponenteTest

See: Componente

Located at test/File PHP/Gestione Menu/ComponenteTest.php

protected	**setUp( )** **Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.**
protected	**tearDown( )**

	Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.
public	**test__get( )** Covers **Componente::__get**
public	**test__set( )** Covers **Componente::__set**
public	**test__toString( )** Covers **Componente::__toString**

protected Componente	**$object**

5.4 Class Connection

Classe che contiene i dati necessari per effettuare la connessione con il database.

Located at File PHP/connect_db.php

public static string	$SERVER	"localhost:3306"
public static string	$USERNAME	"root"
public static string	$PASSWORD	"floccari"

5.5 Class Gestore_Account_Control

Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di un account.

See: Account

Located at File PHP/Gestione Account/Gestore_Account_Control.php

public	__construct( ) Metodo costruttore per istanziare un oggetto di tipo Gestore_Account_Control
public	disconnetti( ) Metodo che effettua il logout di un utente autenticato.

public Account	**accediAccount**(type $email, type $password, type $type) Metodo che effettua il login. Verifica che un utente sia registrato nel sistema. Parameters $email email univoca dell'utente. $password password associata all'email dell'utente. $type tipo di account con cui si vuole effettuare l'accesso. Returns Account Ritorna l'account associato ai parametri inseriti se presenti nel db. null Ritorna un valore null se il db non contiene un account associato con tali parametri.
public type	**creaAccount**(type $nome, type $cognome, type $email, type $pass, type $citta, type $indirizzo, type $cap, type $telefono) Metodo che crea un account registrandolo nel sistema.

Parameters

$nome

nome dell'utente.

$cognome

cognome dell'utente.

$email

email univoca dell'utente.

$pass

password dell'utente.

$citta

città dell'utente.

$indirizzo

indirizzo dell'utente.

$cap

cap associato alla città dell'utente.

$telefono

dell'utente.

	Returns type null Ritorna un valore null se l'inserimento è andata a buon fine. string Ritorna una stringa contenente la causa di errore.
public type	creaCarrello(type $email) Metodo che associa un carrello ad un utente. Parameters $email email univoca dell'utente. Returns type null Ritorna un valore null se l'inserimento è andata a buon fine. string Ritorna una stringa contenente la causa di errore.
public type	modificaAccount(type $email, type $password, type $citta, type $indirizzo, type $cap, type $telefono)

Metodo che modifica i dati di un account registrato nel sistema.

Parameters

$email

email dell'utente

$password

password associata all'utente

$citta

città dell'utente

$indirizzo

indirizzo dell'utente

$cap

cap associtao alla città dell'utente

$telefono

telefono dell'utente

Returns

type

null Ritorna un valore null se la modifica è andata a buon fine.

	string **Ritorna una stringa contenente la causa di errore.**

5.5.1 Class Gestore_Account_Control_Test

Classe Test in riferimento alla classe Gestore_Account_Control

PHPUnit_Framework_TestCase

Gestore_Account_ControlTest

See: Gestore_Account_Control

See: Account

Located at test/File PHP/Gestione Account/Gestore_Account_ControlTest.php

protected	**setUp( )** **Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.**
protected	**tearDown( )** **Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.**
public	**testCreaAccount( )** Covers **Gestore_Account_Control::creaAccount**

public	testAccediAccount( ) Covers Gestore_Account_Control::accediAccount
public	testCreaCarrello( ) Covers Gestore_Account_Control::creaCarrello
public	testModificaAccount( ) Covers Gestore_Account_Control::modificaAccount

protected Gestore_Account_Control	$object Riferimento globale all'oggetto di tipo Gestore_Account_Control
protected Account	$account Riferimento globale all'oggetto di tipo Account

5.6 Class Gestore_Carrello_Control

Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di un carrello.

See: Carrello

See: Componente

See: Menu_scelto

Located at File PHP/Gestione Carrello/Gestore_Carrello_Control.php

public	**__construct( )** **Metodo costruttore per istanziare un oggetto di tipo Gestore_Carrello_Control**
public type	**eliminaMenù(type $id_menu)** **Metodo che elimina un menù scelto dal db.** Parameters **$id_menu** **identificativo del menù scelto da eliminare.** Returns **type** **null Ritorna un valore null se l'eliminazione è andata a buon fine.** **string**

	Ritorna una stringa contenente la causa di errore.
public	svuotaCarrello(array $menu_codici) Metodo che elimina più menù scelti. Parameters $menu_codici identificativi dei menù scelti.
public	creaVisualizzaCarrelloLista(type $email) Metodo che fa l'eco dei menù scelti di un utente contenuti nel carrello. Parameters $email email univoca dell'utente.
public	getCarrello(type $email) Metodo che ritorna i dati di un carrello di un utente. Parameters $email email univoca dell'utente.
public type	creaCompostoDa(type $id_menu_scelto, type $componente)

	Metodo che associa una componente ad un menù scelto. Parameters **$id_menu_scelto** **identificativo del menù scelto** **$componente** **nome della componente.** Returns **type** **null Ritorna un valore null se l'inserimento è andato a buon fine.** **string** **Ritorna una stringa contenente la causa di errore.**
public type	**creaMenuScelto(type $nome, type $prezzo, type $quantita)** **Metodo che inserisce un menù scelto nel db.** Parameters **$nome** **nome del menù scelto** **$prezzo**

	prezzo associato al menù scelto **$quantita** quantità associato al menù scelto Returns type null Ritorna un valore null se l'inserimento è andato a buon fine. string Ritorna una stringa contenente la causa di errore.
public type	**aggiungiAlCarrello**(type **$id_menu_scelto**, type **$email**) Metodo che aggiunge un menù scelto nel carrello di un utente. Parameters **$id_menu_scelto** identificativo del menù scelto **$email** email univoca dell'utente Returns type

	null Ritorna un valore null se l'inserimento è andato a buon fine. **string** **Ritorna una stringa contenente la causa di errore.**
public	**dettagliMenù(type $id_menu)** **Metodo che fa l'eco delle componenti associate a un menù scelto.** Parameters **$id_menu** **identificativo del menù scelto**

5.6.1 Class Gestore_Carrello_Control_Test

Classe Test in riferimento alla classe Gestore_Carrello_Control

PHPUnit_Framework_TestCase

Gestore_Carrello_ControlTest

See: Gestore_Carrello_Control

See: Carrello

Located at test/File PHP/Gestione Carrello/Gestore_Carrello_ControlTest.php

protected	**setUp( )**

	Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.
protected	**tearDown( )** **Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.**
public	**testCreaMenuScelto( )** Covers **Gestore_Carrello_Control::creaMenuScelto**
public	**testCreaCompostoDa( )** Covers **Gestore_Carrello_Control::creaCompostoDa**
public	**testAggiungiAlCarrello( )** Covers **Gestore_Carrello_Control::aggiungiAlCarrello**
public	**testGetCarrello( )** Covers **Gestore_Carrello_Control::getCarrello**
public	**testCreaVisualizzaCarrelloLista( )** Covers

	Gestore_Carrello_Control::creaVisualizzaCarrelloLista
public	testDettagliMenù( ) Covers Gestore_Carrello_Control::dettagliMenù
public	testEliminaMenù( ) Covers Gestore_Carrello_Control::eliminaMenù
public	testSvuotaCarrello( ) Covers Gestore_Carrello_Control::svuotaCarrello

protected Gestore_Carrello_Control	$object Riferimento globale all'oggetto di tipo Gestore_Carrello_Control
protected Carrello	$carrello Riferimento globale all'oggetto di tipo Carrello

5.7 Class Gestore_Menu_Control

Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di un menù.

See: Menu

See: Componente

Located at File PHP/Gestione Menu/Gestore_Menu_Control.php

public	**__construct( )** **Metodo costruttore per istanziare un oggetto di tipo Gestore_Account_Control**
public	**creaListaMenùForm( )** **Metodo che fa l'eco dei menù presenti nel db.**
public	**creaSelezioneComponentiMenùForm(type $nome_menu)** **Metodo che fa l'eco delle componenti associate ad un menù** Parameters **$nome_menu** **nome del menù**
public type	**aggiungiAQuickMenù(type $nome_menu, type $email)** **Metodo che aggiunge un menù tra i preferiti dell'utente.** Parameters

	$nome_menu **nome del menù** **$email** **email univoca dell'utente** Returns **type** **null Ritorna un valore null se l'inserimento è andato a buon fine.** **string** **Ritorna una stringa contenente la causa di errore.**

5.7.1 Class Gestore_Menu_Control_Test

Classe Test in riferimento alla classe Gestore_Account_Control

PHPUnit_Framework_TestCase

Gestore_Menu_ControlTest

See: Gestore_Menu_Control

See: Menu

Located at test/File PHP/Gestione Menu/Gestore_Menu_ControlTest.php

protected	**setUp( )** **Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.**
protected	**tearDown( )** **Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.**
public	**testCreaListaMenùForm( )** Covers **Gestore_Menu_Control::creaListaMenùForm**
public	**testCreaSelezioneComponentiMenùForm( )** Covers **Gestore_Menu_Control::creaSelezioneComponentiMenùForm**
public	**testAggiungiAQuickMenù( )** Covers **Gestore_Menu_Control::aggiungiAQuickMenù**

protected Gestore_Menu_Control	**$object** **Riferimento globale all'oggetto di tipo Gestore_Menu_Control**
protected Menu	**$menu**

	Riferimento globale all'oggetto di tipo Menu

5.8 Class Gestore_Ordine_Control

Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di un ordine.

See: Ordine

See: Componente

See: Menu_scelto

Located at File PHP/Gestione Ordine/Gestore_Ordine_Control.php

public	**__construct( )** **Metodo costruttore per istanziare un oggetto di tipo Gestore_Ordine_Control**
public type	**cancellaOrdine(type $id_ordine, type $stato)** **Metodo che cancella un ordine nel db.** Parameters **$id_ordine** **identificativo dell'ordine** **$stato**

	stato dell'ordine Returns type null Ritorna un valore null se l' eliminazione è andata a buon fine. string Ritorna una stringa contenente la causa di errore.
public	creaVisualizzaOrdiniLista(type $email) Metodo che fa l'eco degli ordini associati ad un utente. Parameters $email email univoca dell'utente
public type	creaPossiede(type $id_ordine, type $id_menu) Metodo che associa un menù scelto ad un ordine. Parameters $id_ordine identificativo dell'ordine $id_menu

	identificativo del menù scelto Returns type null Ritorna un valore null se l'inserimento è andato a buon fine. string Ritorna una stringa contenente la causa di errore.
public type	checkOut(type $email, type $nome, type $indirizzo, type $orario) Metodo che crea un ordine nel db Parameters $email email univoca dell'utente $nome nome dell'utente $indirizzo indirizzo della consegna dell'ordine $orario orario desiderato dall'utente

	Returns type null Ritorna un valore null se l'inserimento è andato a buon fine. string Ritorna una stringa contenente la causa di errore.
public	dettagliOrdine(type $id_ordine) Metodo che fa l'eco dei menù scelti e delle componenti associato ad un ordine. Parameters $id_ordine identificativo dell'ordine

5.8.1 Class Gestore_Ordine_Control_Test

Classe Test in riferimento alla classe Gestore_Ordine_Control

PHPUnit_Framework_TestCase

Gestore_Ordine_ControlTest

See: Gestore_Ordine_Control

See: Ordine

Located at test/File PHP/Gestione Ordine/Gestore_Ordine_ControlTest.php

protected	**setUp( )** **Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.**
protected	**tearDown( )** **Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.**
public	**testCancellaOrdine( )** Covers **Gestore_Ordine_Control::cancellaOrdine**
public	**testCreaVisualizzaOrdiniLista( )** Covers **Gestore_Ordine_Control::creaVisualizzaOrdiniLista**
public	**testCreaPossiede( )** Covers **Gestore_Ordine_Control::creaPossiede**
public	**testCheckOut( )** Covers **Gestore_Ordine_Control::checkOut**
public	**testDettagliOrdine( )**

	Covers **Gestore_Ordine_Control::dettagliOrdine**

protected Gestore_Ordine_Control	**$object** **Riferimento globale all'oggetto di tipo Gestore_Ordine_Control**
protected Ordine	**$ordine** **Riferimento globale all'oggetto di tipo Ordine**

5.9 Class Gestore_QuickMenu_Control

Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di un menù tra i preferiti.

See: Menu

Located at File PHP/Gestione QuickMenu/Gestore_QuickMenu_Control.php

public	**__construct( )** **Metodo costruttore per istanziare un oggetto di tipo Gestore_QuickMenu_Control**
public	**creaListaQuckMenù(type $email)** **Metodo che fa l'eco dei menù associati ai preferiti di un utente.**

	Parameters **$email** email univoca dell'utente
public type	**eliminaQuickMenù**(type $nome_menu, type $email) Metodo che elimina un menù dai preferiti di un utente Parameters **$nome_menu** nome del menù **$email** email univoca dell'utente Returns type null Ritorna un valore null se l'eliminazione è andata a buon fine. string Ritorna una stringa contenente la causa di errore.

5.9.1 Class Gestore_QuickMenu_Control_Test

Classe Test in riferimento alla classe Gestore_QuickMenu_Control

PHPUnit_Framework_TestCase

Gestore_QuickMenu_ControlTest

See: Gestore_QuickMenu_Control

See: Menu

Located at test/File PHP/Gestione QuickMenu/Gestore_QuickMenu_ControlTest.php

protected	**setUp( )** **Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.**
protected	**tearDown( )** **Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.**
public	**testCreaListaQuckMenù( )** Covers **Gestore_QuickMenu_Control::creaListaQuckMenù**
public	**testEliminaQuickMenù( )** Covers **Gestore_QuickMenu_Control::eliminaQuickMenù**

protected Gestore_QuickMenu_Control	$object Riferimento globale all'oggetto di tipo Gestore_QuickMenu_Control
protected Menu	$menu Riferimento globale all'oggetto di tipo Menù

5.10 Class Gestore_Recensione_Control

Questa classe gestisce le interazioni tra utente e sistema, eseguendo query al database. Si occupa della gestione di una recensione.

See: Recensione

Located at File PHP/Gestione Recensione/Gestore_Recensione_Control.php

public	__construct( ) Metodo costruttore per istanziare un oggetto di tipo Gestore_Recensione_Control
public	creaVisualizzaRecensioneLista( ) Metodo che fa l'eco delle recensioni presenti nel db.
public type	creaRecensione(type $email, type $titolo, type $nickname, type $testo, type $valutazione) Metodo che inserisce una recensione nel db. Parameters

$email

email univoca dell'utente

$titolo

titolo della recensione.

$nickname

nickname da utilizzare per la recensione.

$testo

contenuto della recensione

$valutazione

valutazione della recensione.

Returns

type

null Ritorna un valore null se la modifica è andata a buon fine.

string

Ritorna una stringa contenente la causa di errore.

5.10.1 Class Gestore_Recensione_Control_Test

Classe Test in riferimento alla classe Gestore_Recensione_Control

PHPUnit_Framework_TestCase

Gestore_Recensione_ControlTest

See: Gestore_Recensione_Control

See: Recensione

Located at test/File PHP/Gestione Recensione/Gestore_Recensione_ControlTest.php

protected	**setUp( )** **Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.**
protected	**tearDown( )** **Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.**
public	**testCreaVisualizzaRecensioneLista( )** Covers **Gestore_Recensione_Control::creaVisualizzaRecensioneLista**
public	**testCreaRecensione( )** Covers **Gestore_Recensione_Control::creaRecensione**

protected Gestore_Recensione_Control	$object Riferimento globale all'oggetto di tipo Gestore_Recensione_Control
protected Recensione	$reecnsione Riferimento globale all'oggetto di tipo Recensione

5.11 Class Menu

Questa classe rappresenta un'astrazione di un menù.

Located at File PHP/Gestione Menu/Menu.php

public	__construct(type $nome, type $prezzo, type $disponibilita) Metodo costruttore per istanziare un oggetto di tipo Menù Parameters $nome nome del menù $prezzo prezzo associato al menù $disponibilita

	disponibilità associata al menù
public type	__get(type $property) Metodo che ritorna il valore di un determinato proprietà. Parameters $property nome proprietà Returns type valore della proprietà
public type	__set(type $property, type $value) Metodo che cambia il valore di un determinata proprietà, ritornando il valore precedente di essa. Parameters $property nome della proprietà da cambiare $value nuovo valore della proprietà Returns

	type valore precedente della proprietà
public string	__toString( ) Metodo che ritorna il valore di tutte le proprietà dell'oggetto Menu in un formato stabilito. Returns string ritorna il valore di tutte le proprietà dell'oggetto

5.11.1 Class Menu_Test

Classe Test in riferimento alla classe Menu.

PHPUnit_Framework_TestCase

MenuTest

See: Menu

Located at test/File PHP/Gestione Menu/MenuTest.php

protected	setUp( ) Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.
protected	tearDown( )

	Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.
public	**test__get( )** Covers **Menu::__get**
public	**test__set( )** Covers **Menu::__set**
public	**test__toString( )** Covers **Menu::__toString**

protected Menu	**$object**

5.12 Class Menu_Scelto

Questa classe rappresenta un'astrazione di un menù che è stato scelto.

Located at File PHP/Gestione Carrello/Menu_Scelto.php

public	**__construct(type $id, type $nome, type $prezzo, type $quantità)**

	Metodo costruttore per istanziare un oggetto di tipo Menu_Scelto Parameters **$id** **identificativo del menù scelto** **$nome** **nome del menù scelto** **$prezzo** **prezzo del menù scelto** **$quantità** **quantità del menù scelto**
public type	**__get(type $property)** **Metodo che ritorna il valore di un determinato proprietà.** Parameters **$property** **nome proprietà** Returns **type**

	valore della proprietà
public type	__set(type $property, type $value) Metodo che cambia il valore di un determinata proprietà, ritornando il valore precedente di essa. Parameters $property nome della proprietà da cambiare $value nuovo valore della proprietà Returns type valore precedente della proprietà
public string	__toString( ) Metodo che ritorna il valore di tutte le proprietà dell'oggetto Menu_Scelto in un formato stabilito. Returns string ritorna il valore di tutte le proprietà dell'oggetto

5.12.1 Class Menu_Scelto_Test

Classe Test in riferimento alla classe Menu_Scelto.

PHPUnit_Framework_TestCase

Menu_SceltoTest

See: Menu_Scelto

Located at test/File PHP/Gestione Carrello/Menu_SceltoTest.php

protected	**setUp( )** **Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.**
protected	**tearDown( )** **Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.**
public	**test__get( )** Covers **Menu_Scelto::__get**
public	**test__set( )** Covers **Menu_Scelto::__set**
public	**test__toString( )** Covers

	Menu_Scelto::__toString

protected Menu_Scelto	$object

5.13 Class Ordine

Questa classe rappresenta un'astrazione di un ordine.

Author: Egidio Giacoia

Located at File PHP/Gestione Ordine/Ordine.php

public	__construct(type $id, type $nome_utente, type $tempo_attesa, type $stato, type $indirizzo_consegna, type $orario_consegna, type $totale_quantita, type $totale_prezzo) Metodo costruttore per istanziare un oggetto di tipo Ordine. Parameters $id identificativo dell'ordine $nome_utente nome dell'utente $tempo_attesa

	tempo di attesa specificato per la consegna dell'ordine **$stato** stato dell'ordine **$indirizzo_consegna** indirizzo di consegna dell'ordine **$orario_consegna** orario di consegna desiderato dall'utente **$totale_quantita** quantità totale dell'ordine **$totale_prezzo** prezzo totale dell'ordine
public type	__get(type $property) Metodo che ritorna il valore di un determinato proprietà. Parameters **$property** nome proprietà Returns

	type valore della proprietà
public type	__set(type $property, type $value) Metodo che cambia il valore di un determinata proprietà, ritornando il valore precedente di essa. Parameters $property nome della proprietà da cambiare $value nuovo valore della proprità Returns type valore precedente della proprietà
public string	__toString( ) Metodo che ritorna il valore di tutte le proprietà dell'oggetto Ordine in un formato stabilito. Returns string ritorna il valore di tutte le proprietà dell'oggetto

5.13.1 Class Ordine_Test

Classe Test in riferimento alla classe Ordine.

PHPUnit_Framework_TestCase

OrdineTest

See: Ordine

Located at test/File PHP/Gestione Ordine/OrdineTest.php

protected	**setUp( )** **Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.**
protected	**tearDown( )** **Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.**
public	**test__get( )** Covers **Ordine::__get**
public	**test__set( )** Covers **Ordine::__set**

public	test__toString( ) Covers Ordine::__toString

protected Ordine	$object

5.14 Class Recensione

Questa classe rappresenta un'astrazione di una recensione.

Located at File PHP/Gestione Recensione/Recensione.php

public	__construct(type $titolo, type $testo, type $nickname, type $apprezzamento) Metodo costruttore per istanziare un oggetto di tipo Recensione Parameters $titolo titolo della recensione $testo contenuto della recensione $nickname

	nickname di chi ha scritto la recensione **$apprezzamento** giudizio del servizio offerto.
public type	**__get(type $property)** Metodo che ritorna il valore di un determinato proprietà. Parameters **$property** nome proprietà Returns type valore della proprietà
public type	**__set(type $property, type $value)** Metodo che cambia il valore di un determinata proprietà, ritornando il valore precedente di essa. Parameters **$property** nome della proprietà da cambiare **$value**

	nuovo valore della proprità Returns type valore precedente della proprietà
public string	__toString( ) Metodo che ritorna il valore di tutte le proprietà dell'oggetto Recensione in un formato stabilito. Returns string ritorna il valore di tutte le proprietà dell'oggetto

5.14.1 Class Recensione_Test

Classe Test in riferimento alla classe Recensione.

PHPUnit_Framework_TestCase

RecensioneTest

See: Recensione

Located at test/File PHP/Gestione Recensione/RecensioneTest.php

protected	setUp( )

	Sets up the fixture, for example, opens a network connection. This method is called before a test is executed.
protected	**tearDown( )** **Tears down the fixture, for example, closes a network connection. This method is called after a test is executed.**
public	**test__get( )** Covers **Recensione::__get**
public	**test__set( )** Covers **Recensione::__set**
public	**test__toString( )** Covers **Recensione::__toString**

protected Recensione	**$object** **Riferimento globale all'oggetto di tipo Recensione**