Università degli Studi di Salerno
Corso di Ingegneria del Software

Sandwich on web

SDD
Versione 1.0

Data: 23/12/2015

Partecipanti:

Nome	Matricola
Sara Volpe	0512102434
Egidio Giacoia	0512102376
Nunzia Esposito	0512102328

Scritto da:	Team members

Revision History

Data	Versione	Descrizione	Autore
07/12/2015	1	Prima stesura del documento	Team members
09/12/2015	1	Aggiunta di: • Capitolo 1: Introduzione	Team members
10/12/2015	1	Aggiunta di: • Capitolo 2: Architettura del sistema corrente • Capitolo 3: Architettura del sistema proposto	Sara Volpe
12/10/2015	1	Aggiunta di: • Paragrafo 3.2: Decomposizione in sottosistemi • Paragrafo 3.3: Hardware/Software Mapping • Paragrafo 3.5: Controllo degli accessi	Sara Volpe
13/11/2015	1	Aggiunta di: • Paragrafo 3.6: Flusso di controllo globale • Paragrafo 3.7: Boundary Conditions	Sara Volpe
15/11/2015	1	Aggiunta di: • Capitolo 4: Glossario	Sara Volpe
22/12/2015	1	Revisione documento	Nunzia Esposito

INDICE

1.Introduzione

Questo documento descrive gli obiettivi di design del progetto. Illustra la scomposizione del sistema in sottosistemi e definisce le strategie adottate per il loro sviluppo.

1.1 Scopo del sistema

L'obiettivo del progetto è quello di realizzare un sistema che permetta ai clienti di ordinare panini attraverso la creazione di un portale web. Lo sviluppo di un sito web di questo tipo permetterebbe di ottimizzare i tempi di servizio,velocizzare le ordinazioni effettuandole con un semplice click attraverso un'interfaccia grafica dei vari menù e tenere informati i clienti sullo stato dell'ordine ogniqualvolta viene effettuato.
Tale sito web dovrebbe consentire una comunicazione più immediata ed efficace tra gli altri attori del sistema (utente, gestore degli ordini, gestore dei menù, fattorino).
In definitiva Sandwich on web nasce per ottimizzare il lavoro del personale e velocizzare le preferenze dei clienti attraverso un portale web efficace ed interattivo.

Il sistema deve permettere:

1. la gestione degli account
2. la gestione dei menù
3. la gestione degli ordini
4. la gestione di un carrello
5. la gestione delle recensioni

6. la gestione del Quick menù

1.2 Design goals

Il sistema si avvale di una struttura grafica chiara e completa, con bottoni, finestre di dialogo, icone, form per l'immissione dei dati e finestre scorrevoli. Le informazioni presentate sullo schermo saranno in grado di indirizzare l'utente verso le funzionalità a cui desidera accedere, cercando di volta in volta di isolare soltanto le informazioni necessarie per l'esecuzione della funzione richiesta. L'utente non dovrà necessariamente effettuare operazioni che richiedono una discreta conoscenza dell'applicazione, quindi l'utilizzo del sistema sarà guidato dall'interfaccia semplice e intuitiva. Il sistema Sandwich on web ha come struttura centrale un database, il quale sarà periodicamente aggiornato per garantire il corretto funzionamento del sistema stesso.
Il sistema proposto cerca di rispettare tutti i criteri di design sotto elencati.

1.2.1DG_1 – Performance criteria

Il sistema sarà usabile e leggero, cosicché, nel caso di utilizzo in contemporanea da parte di più utenti, il sistema non sia rallentato. In definitiva il sistema dovrà garantire che le varie operazioni siano svolte in un intervallo di tempo accettabile.
Quindi Sandwich on web si propone di rispettare i seguenti requisiti di qualità (rispetto alle prestazioni):
• ***DG_1.1 - Tempo di risposta***: le risposte verranno date in un tempo accettabile a seguito dell'elaborazione dell'input.
• ***DG_1.2 - Throughput***: il sistema dovrà completare il maggior numero di operazioni nel minor tempo possibile, per garantire una maggiore interattività con i vari utenti connessi.
• ***DG_1.3 - Memoria***: il sistema necessita di una quantità di memoria dipendente da tutti i dati che saranno memorizzati all'interno della Web-Application realizzata.

1.2.2 DG_2 – Dependability criteria

Sandwich on web garantirà il corretto svolgimento delle proprie funzioni, gestendo i vari errori logici (quelli derivanti da una negligenza da parte dell'utente), che potranno verificarsi durante l'utilizzo, ed eventuali attacchi alla sicurezza. Sandwich on web si propone, quindi, di rispettare i seguenti requisiti di qualità, relativi all'affidabilità:

- ***DG_2.1 - Robustezza***: Sandwich on web dovrà offrire un buon grado di robustezza agli input invalidi forniti dagli utenti. Non verranno alterati i dati contenuti nel database: nel caso in cui l'utente sottometta dati errati al sistema, questo lancerà un messaggio d'errore per avvisare lo stesso utente che i dati inseriti sono invalidi;
- ***DG_2.2 - Affidabilità***: Sandwich on web dovrà garantire il corretto svolgimento delle proprie funzionalità, producendo unicamente l'output atteso;
- ***DG_2.3 - Disponibilità***: il software sarà sempre disponibile e funzionante, tranne in eventuali periodi di manutenzione;
- ***DG_2.4 - Sicurezza***: Ogni utente potrà accedere con una login e password personale; l'accesso al sistema sarà controllato da un apposito sistema di autenticazione, che permetterà ad ogni categoria di utente di eseguire il proprio lavoro senza intaccare o modificare quello altrui.

1.2.3 DG_3 - Maintenance criteria

Il sistema garantirà un'alta manutenibilità. Sandwich on web dovrà quindi rispettare i seguenti requisiti di qualità:
• ***DG_3.1 - Estendibilità***: il sistema sarà realizzato in modo da poter garantire l'inserimento di nuove funzionalità in maniera semplice, senza doverne modificare altre.
• ***DG_3.2 - Modificabilità***: il sistema dovrà essere realizzato in modo da poter garantire la modifica di funzionalità già presenti all'interno del sistema, senza doverne apportare altre, quindi il grado di entropia del sistema deve essere mantenuto il più basso possibile.
• ***DG_3.3 - Tracciabilità dei requisiti***: tramite una buona documentazione sarà possibile risalire ai corrispettivi requisiti funzionali, cui faranno riferimento le varie classi e i metodi.

1.2.4 DG_4 – End-user criteria

Per quanto riguarda gli utenti Sandwich on web dovrà garantire i seguenti requisiti di qualità:
• ***DG_4.1 - Utilità***: grazie ai requisiti funzionali raccolti, Sandwich on web supporterà in pieno le esigenze delle varie tipologie di utenti.
• ***DG_4.2 - Usabilità***: il sistema dovrà essere semplice ed intuitivo e dopo un breve utilizzo dovrà consentire all'utente di compiere le operazioni nel minor tempo possibile. Inoltre come descritto nel documento Sandwich on web_RAD dovranno essere rispettati i requisiti non funzionali di usabilità del sistema.

Per maggiore informazioni si faccia riferimento al paragrafo 4 (requisito non funzionale NRF4.1) del documento di analisi dei requisiti.

1.2.5 DG_5-Trade-offs

TO_1 - Spazio vs velocità
Poiché si vuole ottimizzare il tempo di risposta del sistema, si è deciso di concedere più spazio sia in termini di caching che di ridondanza dei dati.
TO_2 - 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.
TO_3 - 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.

1.3 Definizioni, acronimi e abbreviazioni

Vengono di seguito esplicitati definizioni, acronimi e abbreviazioni che verrano incontrati all'interno del documento.

ACRONIMO	DESCRIZIONE
RAD	Requirement Analysis Document
GUI	Graphical User Interface
SW	Software
HW	Hardware
SQL	Structured Query Language
SDD	System Design Document

DBMS	Database Management System

1.4-Riferimenti

• *Sandwich _on _web _RAD_2.0*
• *Bernd Bruegge e Allen H. Dutoit - Object-Oriented Software Engineering (using UML, Patterns and JavaTM) – Prentice Hall.*

1.5-Overview

Il nostro documento è stato diviso in quattro parti:

1. ***Introduzione***: viene riporta una descrizione del sistema specificando il motivo per cui è stato sviluppato, le sue caratteristiche e un accenno sull'utilizzo delle sue funzionalità.
2. ***Architettura del sistema proposto***: viene descritta l'architettura usata nel sistema, ed in particolare: la divisione in sottosistemi, il mapping hardware-software, la gestione dei dati persistenti, il controllo degli accessi di sicurezza, flusso di controllo globale e le condizioni limite.

- Decomposizione in sottosistemi, nella quale è descritta la suddivisione del sistema in vari sottosistemi.

- Hardware/software mapping, in cui sono prese alcune decisioni per quanto riguarda le piattaforme hardware e software su cui il sistema dovrà girare e una volta decise le piattaforme è necessario mappare le componenti su di esse.
  Questa operazione potrebbe portare all'introduzione di nuove componenti per interfacciare i sottosistemi su diverse piattaforme.

- Gestione dei dati persistenti, in cui sono identificati gli oggetti persistenti e è scelto il tipo di infrastruttura da usare per memorizzarli.

- Controllo degli accessi e sicurezza, che descrive il modello utente del sistema in termini di matrici di accesso e i problemi di sicurezza, come la scelta di un meccanismo di autenticazione.
- Flusso di controllo globale, esterno (tra client e server) ed interno, che descrive come il controllo globale del software è implementato, come le

procedure di richiesta sono avviate e come si sincronizzano i sottosistemi.
- Boundary Condition, in cui sono descritte le condizioni limite del sistema.

3. ***Servizi dei sottosistemi***: viene riportata una descrizione dei vari sottosistemi identificati precedentemente ed i servizi offerti per ognuno di essi.
4. ***Glossario****:* vengono definiti i termini più significativi ed utilizzati in maniera da rendere più immediata la lettura.

2-Architettura software corrente

L'architettura proposta nel seguito non andrà a rimpiazzare nessuna struttura pre-esistente.

3-Architettura software proposto

L'architettura scelta per il sistema Sandwich on web è quella Three-Tier.
L'espressione architettura Three-Tier ("a tre strati") indica una particolare architettura software di tipo multi-tier per l'esecuzione di un'applicazione web. Essa prevede la suddivisione dell'applicazione in tre diversi moduli o strati dedicati rispettivamente all'interfaccia utente, alla logica funzionale e alla gestione dei dati persistenti. Tale architettura va tipicamente a mappare a livello fisico-infrastrutturale quella del sistema informatico ospitante l'applicazione da eseguire.
Tali moduli interagiscono fra loro secondo le linee generali del paradigma client-server e utilizzando interfacce ben definite. In questo modo, ciascuno dei tre moduli può essere modificato o sostituito indipendentemente dagli altri, conferendo scalabilità e manutenibilità all'applicazione.

3.1 Overview

Il sistema che s'intende realizzare è, quindi, un sistema distribuito e gli utenti di Sandwich on web potranno interagire con esso tramite i propri terminali (client).
I client, per realizzare le funzionalità richieste, dovranno comunicare con l'application server (che si occupa della logica di controllo), il quale, a sua volta, comunica con il database server, dove sono contenute tutte le informazioni.
Sul database server, risiede un DBMS che si occupa di recuperare, memorizzare ed interrogare i dati presenti nel database, elaborando, quindi, la richiesta degli utenti, inviata sottoforma di query da parte dell'application server. L'aspetto della concorrenza di accessi multipli al database sarà pertanto gestito dal DBMS stesso.
Tale architettura conferisce all'intero sistema una maggiore manutenibilità e permette di gestire il problema della concorrenza degli accessi ai dati in maniera semplice ed efficace. Inoltre, grazie alla natura atomica delle transazioni col database, non si possono verificare problemi d'incoerenza nei dati (a meno di guasti ai supporti di memorizzazione secondari).

3.2 Decomposizione in sottosistemi

Per rendere il sistema più facile da progettare e aumentare il requisito di manutenibilità è stato deciso di decomporlo in sottosistemi, al fine di rendere la progettazione il più semplice possibile. L'architettura Three-Tier organizza i sottosistemi in tre strati: Interface Layer, ovvero tutti gli oggetti Boundary che interagiscono con l'utente. Application Logic Layer, ovvero tutti gli oggetti che riguardano il controllo e le entità del sistema, e Storage Layer, la quale funzione è quella di memorizzare, recuperare ed interrogare gli oggetti persistenti (i dati che sono presenti nei DB).
I tre livelli relativi all'architettura adottata per Sandwich on web, quindi, saranno:
- SandwichOnWebPresentationLayer
- SandwichOnWebApplicationLayer
- SandwichOnWebDataLayer

3.2.1 Servizi sottosistemi

Di seguito sono riportate le relazioni tra i vari layer per ogni singolo sottosistema.

3.2.1.1 SS_0 sottosistema - Gestione account

Application Layer	Presentation Layer	Data Layer
Registra Utente	GUIRegistra Utente	//
Effettua accesso	GUIEffettua accesso	//
Effettua logout	GUIEffettua logout	//
ModificaDatiUtente	GUIModificaDatiUtente	//

3.2.1.2 SS_1 sottosistema - Gestione menù

Application Layer	Presentation Layer	Data Layer
VisualizzaListaMenù	GUIVisualizzaListaMenù	//
AggiungiAQuickMenù	GUIAggiungiAQuickMenù→ GUIVisualizzaListaMenù	//
AcquistaMenù	GUIAcquistaMenù→GUIVisualizzaListaMenù	//
ModificaMenù	GUIModificaMenù→GUIVisualizzaListaMenù	//
RimuoviMenù	GUIRimuoviMenù→GUIVisualizzaListaMenù	//
InserisciMenù	GUIInserisciMenù→GUIVisualizzaListaMenù	//

3.2.1.3 SS_2 sottosistema - Gestione carrello

Application Layer	Presentation Layer	Data Layer
CompletaOrdine	GUICompletaOrdine→GUIVisualizzaCarrello	//

VisualizzaCarrello	GUIVisualizzaCarrello	//
RimuoviMenùDalCarrello	GUIRimuoviMenùDalCarrello→GUIVisualizzaCarrello	//
SvuotaCarrello	GUISvuotaCarrello→GUIVisualizzaCarrello	//

3.2.1.4 SS_3 sottosistema - Gestione ordini

Application Layer	Presentation Layer	Data Layer
VisualizzaOrdine	GUIVisualizzaOrdine	//
CancellaOrdine	GUICancellaOrdine→GUIVisualizzaOrdine	//
ModificaStatoOrdini	GUIModificaStatoOrdini→GUIVisualizzaOrdine	//
ModificaTempoDiAttesa	GUIModificaTempoDiAttesa→GUIVisualizzaOrdine	//

3.2.1.5 SS_4 sottosistema - Gestione recensioni

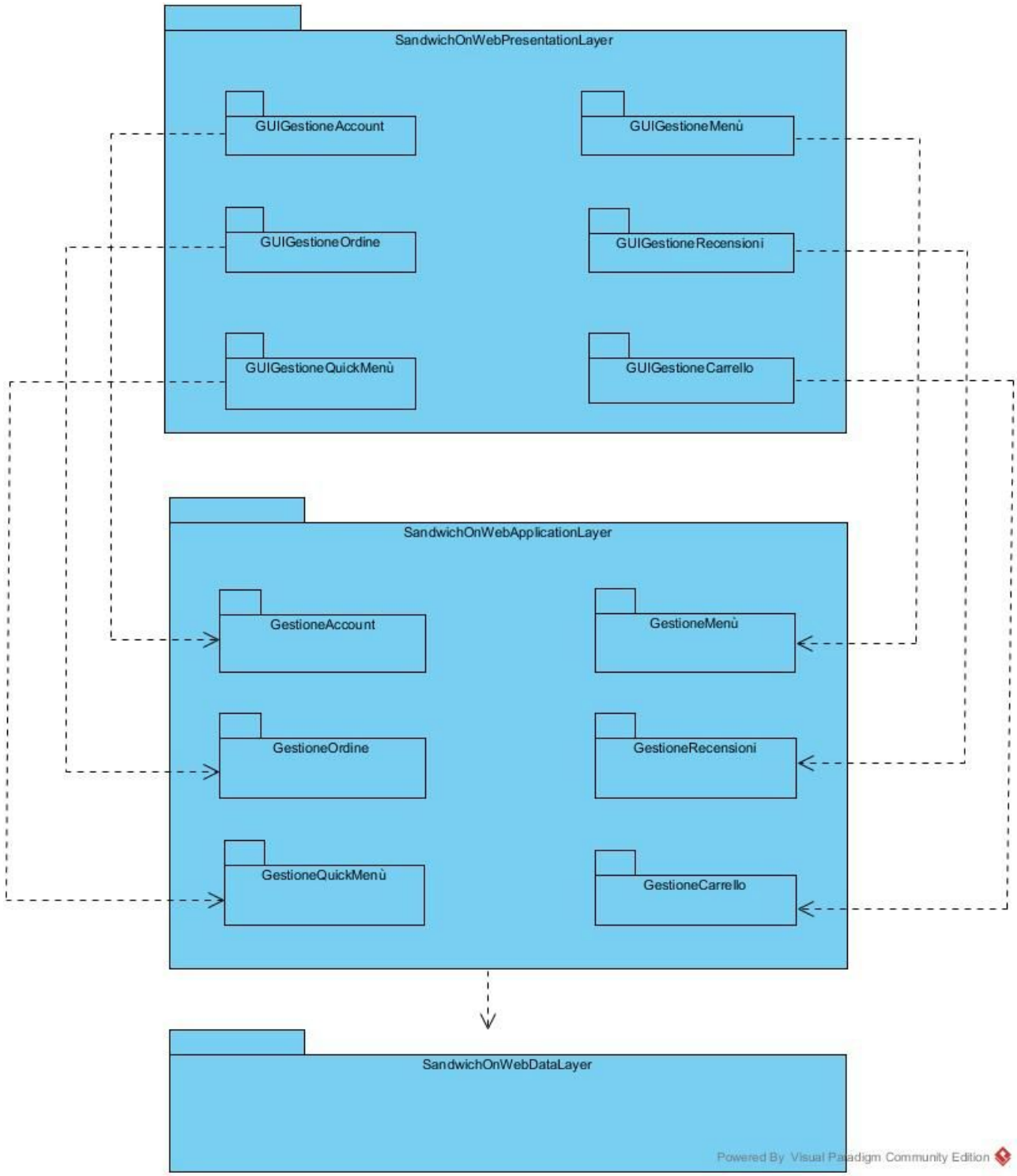
Application Layer	Presentation Layer	Data Layer
VisualizzaRecensioni	GUIVisualizzaRecensioni	//
AggiungiRecensioni	GUIAggiungiRecensioni→GUIVisualizzaRecensioni	//

3.2.1.6 SS_5 sottosistema - Gestione quick menù

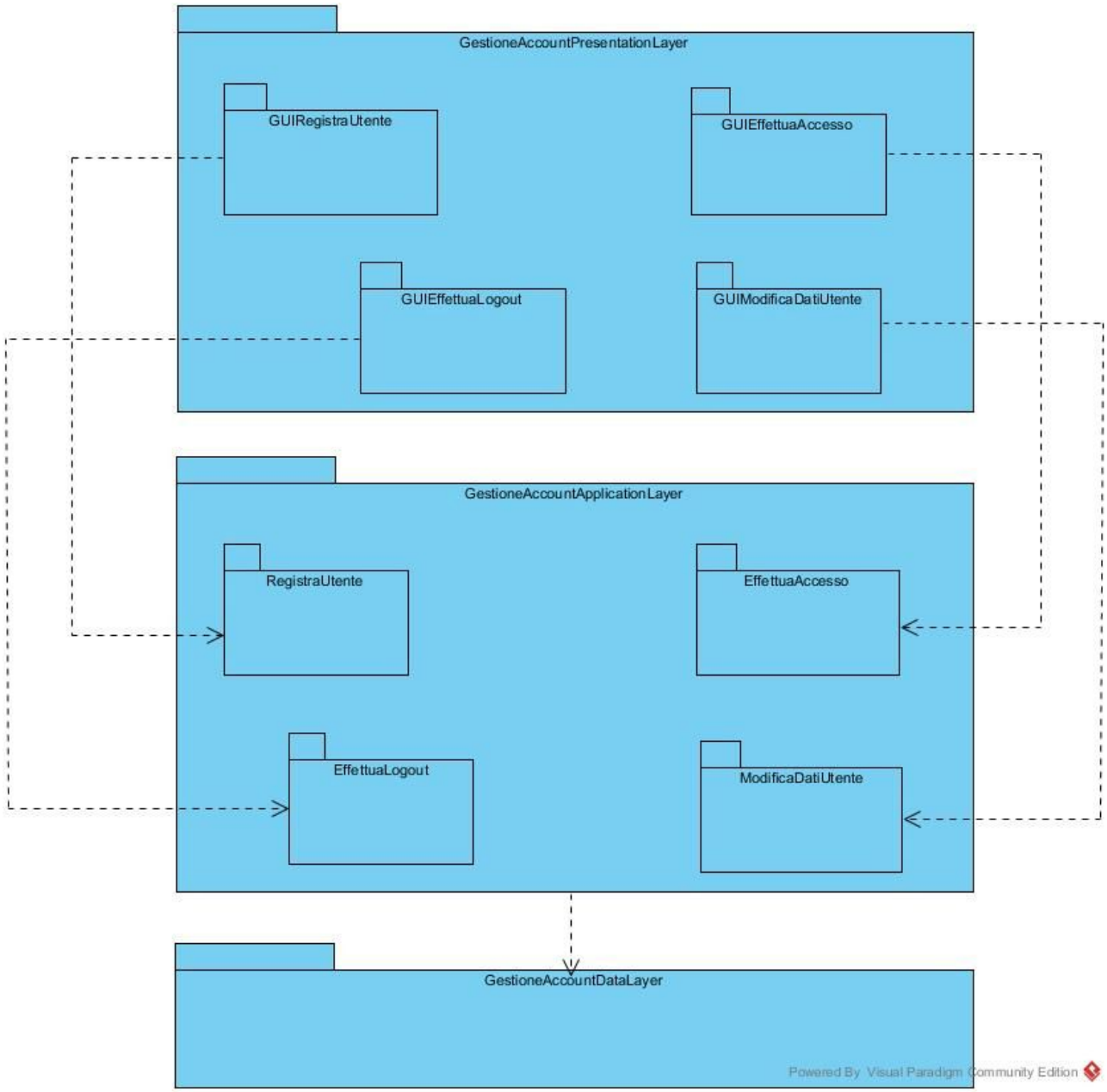
Application Layer	Presentation Layer	Data Layer
VisualizzaQuickMenù	GUIVisualizzaQuickMenù	//
RimuoviDaQuickMe	GUIRimuoviDaQuickMenù→GUIVisualizzaQ	//

nù	uickMenù	

Di seguito è riportato lo schema generale della suddivisione. Successivamente descriveremo tutti i sottosistemi più nel dettaglio.



3.2.2 SubS_0 - Gestione Account



GestioneAccountPresentationLayer

Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per la gestione dell'account, accessibile da tutti gli utenti del sistema.
È riportata di seguito una breve descrizione di ognuna di esse:
• GUIRegistraUtente: è un'interfaccia tramite la quale l'utente si registra nel sistema.
• GUIEffettuaAccesso: è un'interfaccia tramite la quale l'utente e gli admin si autenticano al sistema.
• GUIEffettuaLogout: è un'interfaccia tramite la quale l'utente e gli admin escono dal sistema.
• GUIModificaDatiUtente: è un'interfaccia tramite la quale l'utente modifica i suoi dati nel sistema.

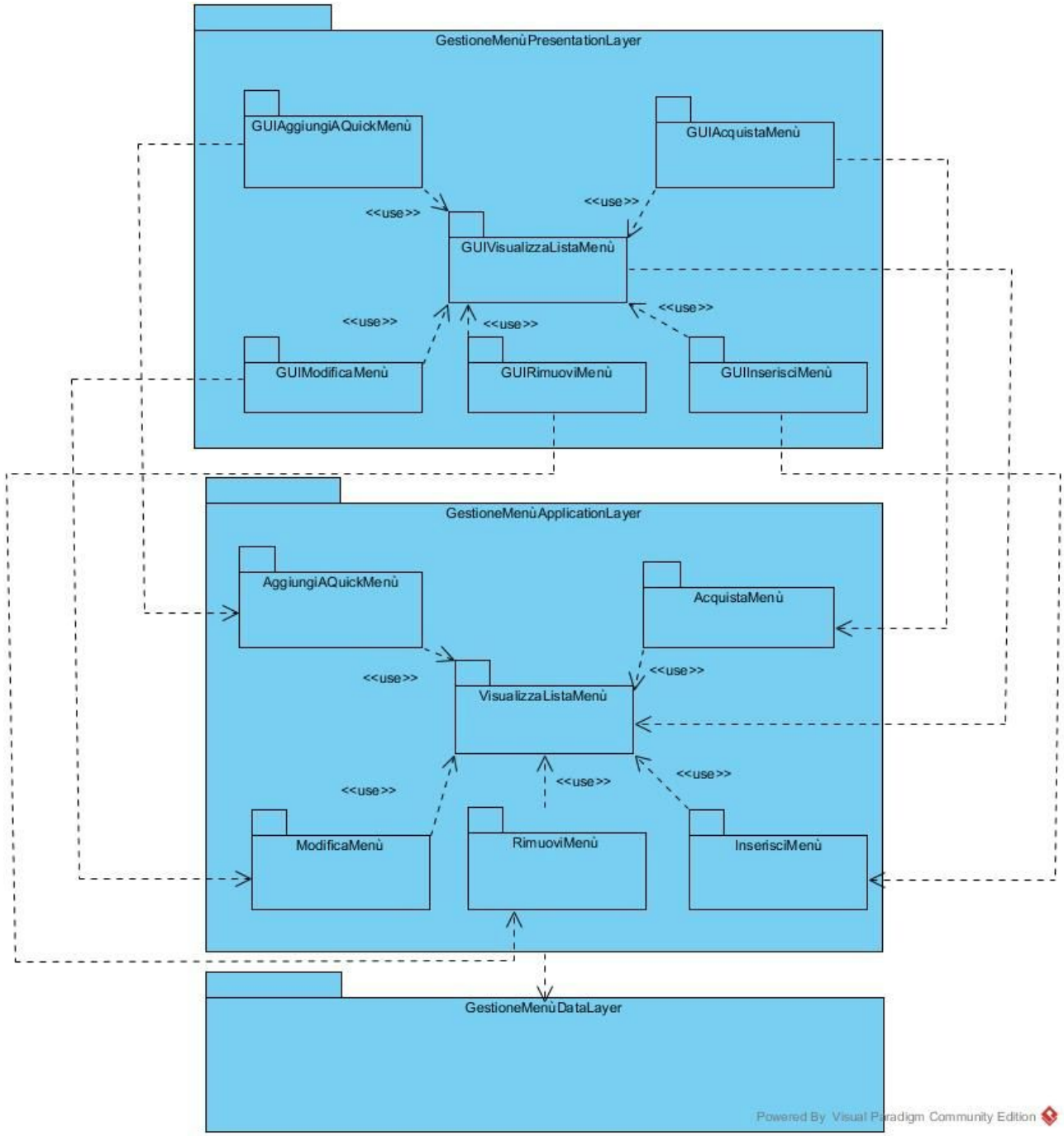
GestioneAccountApplicationLayer

Comprende tutte le componenti logiche, responsabili del corretto funzionamento del sistema, e fa da tramite con il database per aggiornare i dati relativi alla gestione account.
È riportata di seguito una breve descrizione di ognuno di esse:
• RegistraUtente: si occupa di eseguire le operazioni necessarie alla registrazione di un utente del sistema.
• EffettuaAccesso: si occupa di eseguire le operazione necessarie all'autenticazione di un utente e degli admin del sistema.
• EffettuaLogout: si occupa di eseguire le operazione necessarie al logout di un utente e degli admin del sistema.
• ModificaDatiUtente: si occupa di eseguire le operazione necessarie alla modifica dei dati di un utente del sistema.

GestioneAccountDataLayer

Sottosistema che si occupa di rendere reperibili i dati, presenti all'interno del database, relativi alla gestione dell'account. Questo livello consente all'utente e agli admin di effettuare l'autenticazione.

3.2.3 SubS_1 - Gestione Menù



GestioneMenùPresentationLayer

Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per la gestione dei menù, accessibile da tutti gli utenti del sistema e il gestore dei menù.
È riportata di seguito una breve descrizione di ognuna di esse:
• GUIVisualizzaListaMenù: è un'interfaccia tramite la quale l'utente registrato,non registrato e il gestore dei menù possono accedere a tutte le operazioni legate alla visualizzazione dei menù.

• GUIAggiungiAQuickMenù: è un'interfaccia tramite la quale l'utente del sistema può accedere a tutte le operazioni legate all'aggiunta di un menù al quick menù.
• GUIAcquistaMenù: è un'interfaccia tramite la quale l'utente del sistema può accedere a tutte le operazioni legate all'acquisto del menù.
• GUIModificaMenù: è un'interfaccia tramite la quale il gestore dei menù può accedere a tutte le operazioni legate alla modifica dei menù presenti nel sistema.
• GUIRimuoviMenù: è un'interfaccia tramite la quale il gestore dei menù può accedere a tutte le operazioni legate alla rimozione dei menù presenti nel sistema.
• GUIInserisciMenù: è un'interfaccia tramite la quale il gestore dei menù può accedere a tutte le operazioni legate all'inserimento dei menù.

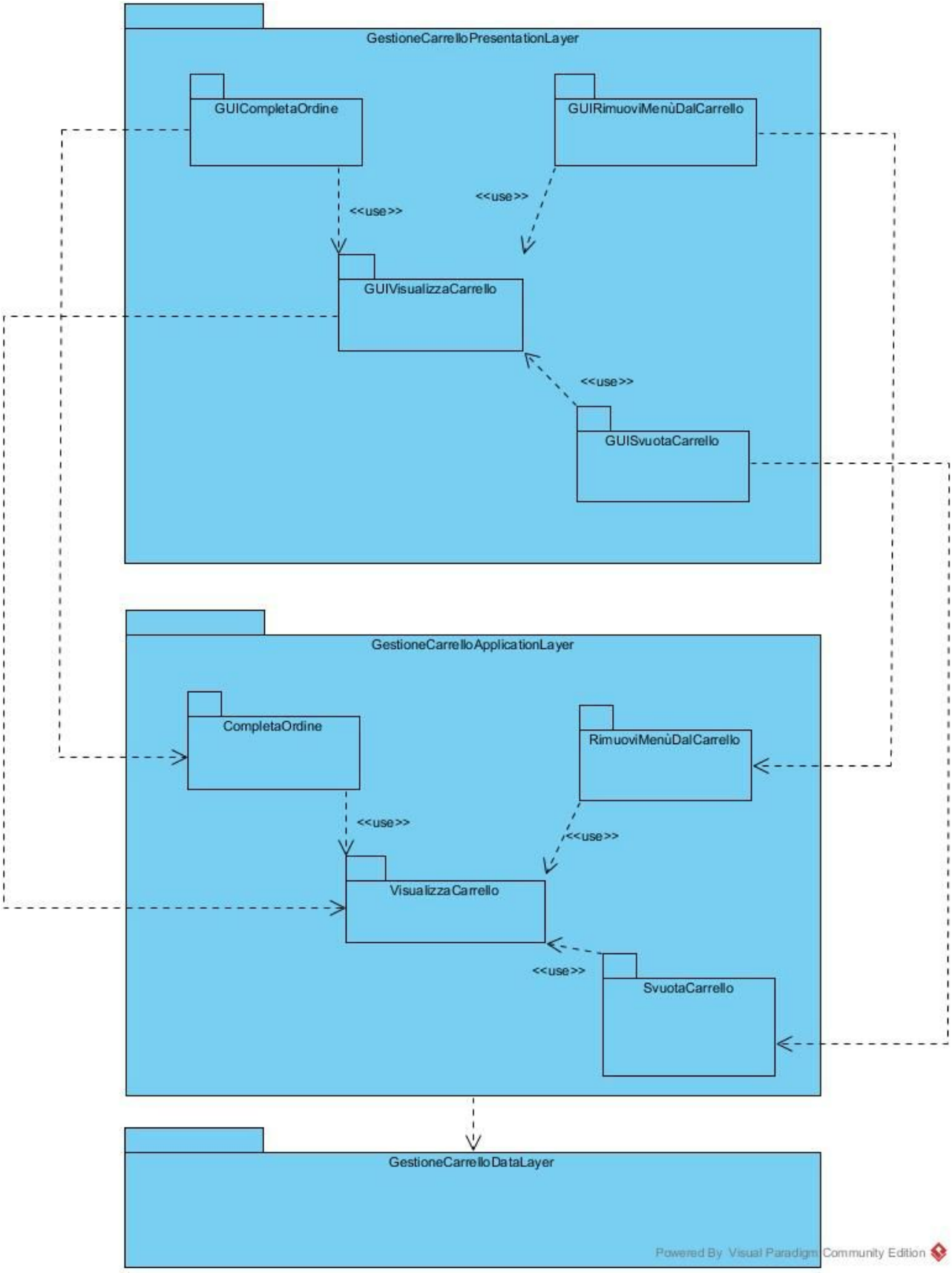
GestioneMenùApplicationLayer

Comprende tutte le componenti logiche, responsabili del corretto funzionamento del sistema, e fa da tramite con il database per aggiornare i dati relativi alla gestione dei menù.
È riportata di seguito una breve descrizione di ognuno di esse:
• VisualizzaListaMenù:si occupa di eseguire le operazioni necessarie alla visualizzazione dei menù presenti all'interno del database.
•AggiungiAQuickMenù : si occupa di eseguire le operazioni necessarie all'aggiunta di un menù nel quick menù e memorizzarlo all'interno del database.
•AcquistaMenù:si occupa di eseguire le operazioni necessarie all'acquisto di un menù e memorizzarlo all'interno del database.
• ModificaMenù: si occupa di eseguire le operazioni necessarie alla modifica di un menù già presente all'interno del database
• RimuoviMenù: si occupa di eseguire le operazioni necessarie alla rimozione di un menù presente all'interno del database
•InserisciMenù: si occupa di eseguire le operazione necessarie all'inserimento di un menù e memorizzarlo all'interno del database.

GestioneMenùDataLayer

Sottosistema che si occupa di rendere reperibili i dati, presenti all'interno del database, relativi alla gestione dei menù. Questo livello consente di inserire nuovi menù nel database, ma anche di cancellarli e di modificarli,di acquistarli,di aggiungerli al quick menù e di memorizzare nel database.

3.2.4 SubS_2 - Gestione carrello


GestioneCarrelloPresentationLayer
GUICompletaOrdine
GUIRimuoviMenùDalCarrello
<<use>>
<<use>>
GUIVisualizzaCarrello
<<use>>
GUISvuotaCarrello
GestioneCarrelloApplicationLayer
CompletaOrdine
RimuoviMenùDalCarrello
<<use>>
<<use>>
VisualizzaCarrello
<<use>>
SvuotaCarrello
GestioneCarrelloDataLayer
Powered By Visual Paradigm Community Edition

GestioneCarrelloPresentationLayer

Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per la gestione del carrello, accessibile da tutti gli utenti del sistema.
È riportata di seguito una breve descrizione di ognuna di esse:
• GUIVisualizzaCarrello: è un'interfaccia tramite la quale l'utente del sistema può accedere a tutte le operazioni legate alla visualizzazione del carrello.
• GUICompletaOrdine: è un'interfaccia tramite la quale l'utente del sistema può accedere a tutte le operazioni legate al completamento dell'ordine di un utente del sistema.
• GUIRimuoviMenùDalCarrello: è un'interfaccia tramite la quale l'utente del sistema può accedere a tutte le operazioni legate alla rimozione di un menù dal carrello.
• GUISvuotaCarrello: è un'interfaccia tramite la quale il gestore dei menù può accedere a tutte le operazioni legate allo svuotamento del carrello.

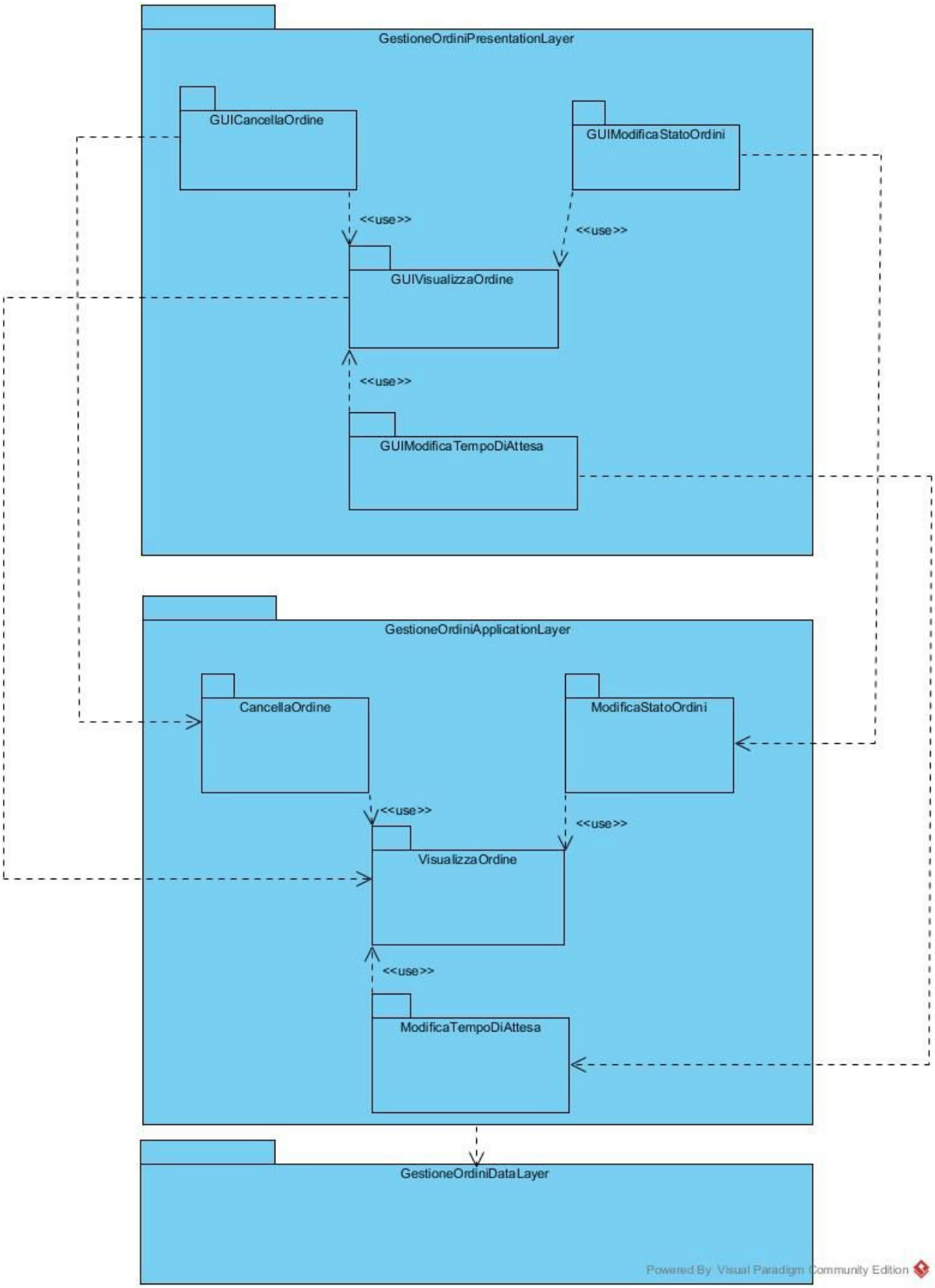
GestioneCarrelloApplicationLayer

Comprende tutte le componenti logiche, responsabili del corretto funzionamento del sistema, e fa da tramite con il database per aggiornare i dati relativi alla gestione del carrello.
È riportata di seguito una breve descrizione di ognuno di esse:
•VisualizzaCarrello:si occupa di eseguire le operazioni necessarie alla visualizzazione dei menù presenti all'interno del database.
•CompletaOrdine : si occupa di eseguire le operazioni necessarie all'aggiunta di un menù nel quick menù e memorizzarlo all'interno del database.
•RimuoviMenùDalCarrello:si occupa di eseguire le operazioni necessarie all'acquisto di un menù e memorizzarlo all'interno del database.
•SvuotaCarrello: si occupa di eseguire le operazioni necessarie alla modifica di un menù già presente all'interno del database

GestioneCarrelloDataLayer

Sottosistema che si occupa di rendere reperibili i dati, presenti all'interno del database, relativi alla gestione dei menù. Questo livello consente di completare l'ordine,di rimuovere i menù dal carrello e di eliminare tutti i menù dal carrello e memorizzare nel database.

3.2.5 SubS_3 - Gestione ordini



GestioneOrdiniPresentationLayer

Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per la gestione degli ordini, accessibile da tutti gli utenti del sistema, il gestore degli ordini e il fattorino.
È riportata di seguito una breve descrizione di ognuna di esse:
• GUIVisualizzaOrdine: è un'interfaccia tramite la quale l'utente registrato, il gestore degli ordini e il fattorino possono accedere a tutte le operazioni legate alla visualizzazione degli ordini.
• GUICancellaOrdine: è un'interfaccia tramite la quale l'utente del sistema può accedere a tutte le operazioni legate alla cancellazione dell'ordine.
• GUIModificaStatoOrdini: è un'interfaccia tramite la quale il gestore degli ordini e il fattorino può accedere a tutte le operazioni legate alla modifica dello stato degli ordini.
• GUIModificaTempoDiAttesa: è un'interfaccia tramite la quale il gestore degli ordini può accedere a tutte le operazioni legate alla modifica del tempo di attesa.

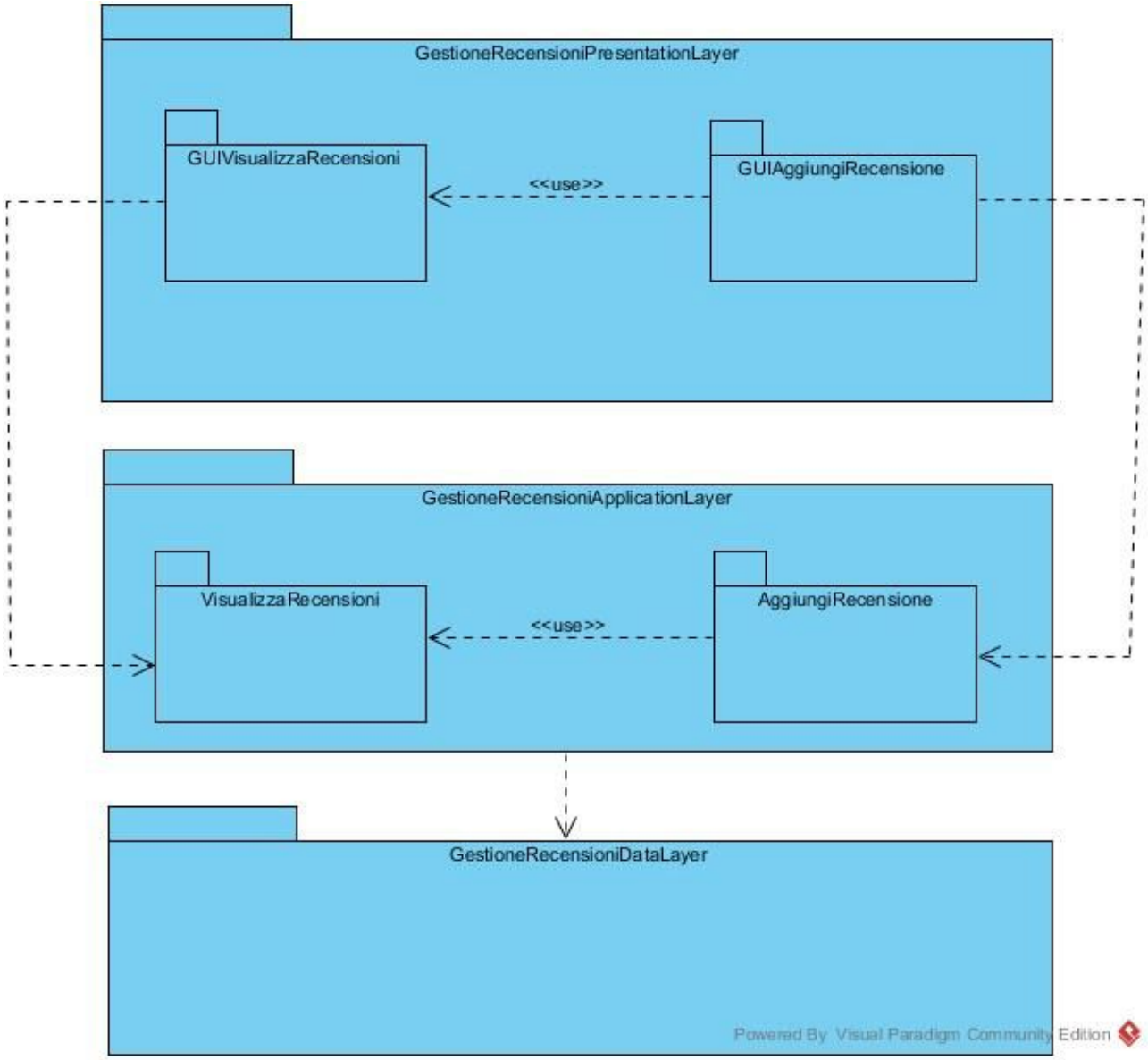
GestioneOrdiniApplicationLayer

Comprende tutte le componenti logiche, responsabili del corretto funzionamento del sistema, e fa da tramite con il database per aggiornare i dati relativi alla gestione degli ordini.
È riportata di seguito una breve descrizione di ognuno di esse:
•VisualizzaOrdine:si occupa di eseguire le operazioni necessarie alla visualizzazione degli ordini presenti all'interno del database.
•CancellaOrdine : si occupa di eseguire le operazioni necessarie alla cancellazione dell 'ordine e memorizzarla all'interno del database.
•ModificaStatoOrdini:si occupa di eseguire le operazioni necessarie alla modifica dello stato degli ordini e memorizzarla all'interno del database.
• ModificaTempoDiAttesa: si occupa di eseguire le operazioni necessarie alla modifica del tempo di attesa e memorizzarla nel database.

GestioneOrdiniDataLayer

Sottosistema che si occupa di rendere reperibili i dati, presenti all'interno del database, relativi alla gestione degli ordini. Questo livello consente di cancellare l'ordine, modificarne lo stato e il tempo di attesa nella coda e memorizzarle nel database.

3.2.6 SubS_4 - Gestione recensioni



GestioneRecensioniPresentationLayer

Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per la gestione delle recensioni, accessibile da tutti gli utenti del sistema.
È riportata di seguito una breve descrizione di ognuna di esse:
• GUIVisualizzaRecensioni: è un'interfaccia tramite la quale l'utente registrato può accedere a tutte le operazioni legate alla visualizzazione delle recensionii.
• GUIAggiungiRecensione: è un'interfaccia tramite la quale l'utente del sistema può accedere a tutte le operazioni legate all'aggiunta delle recensioni.

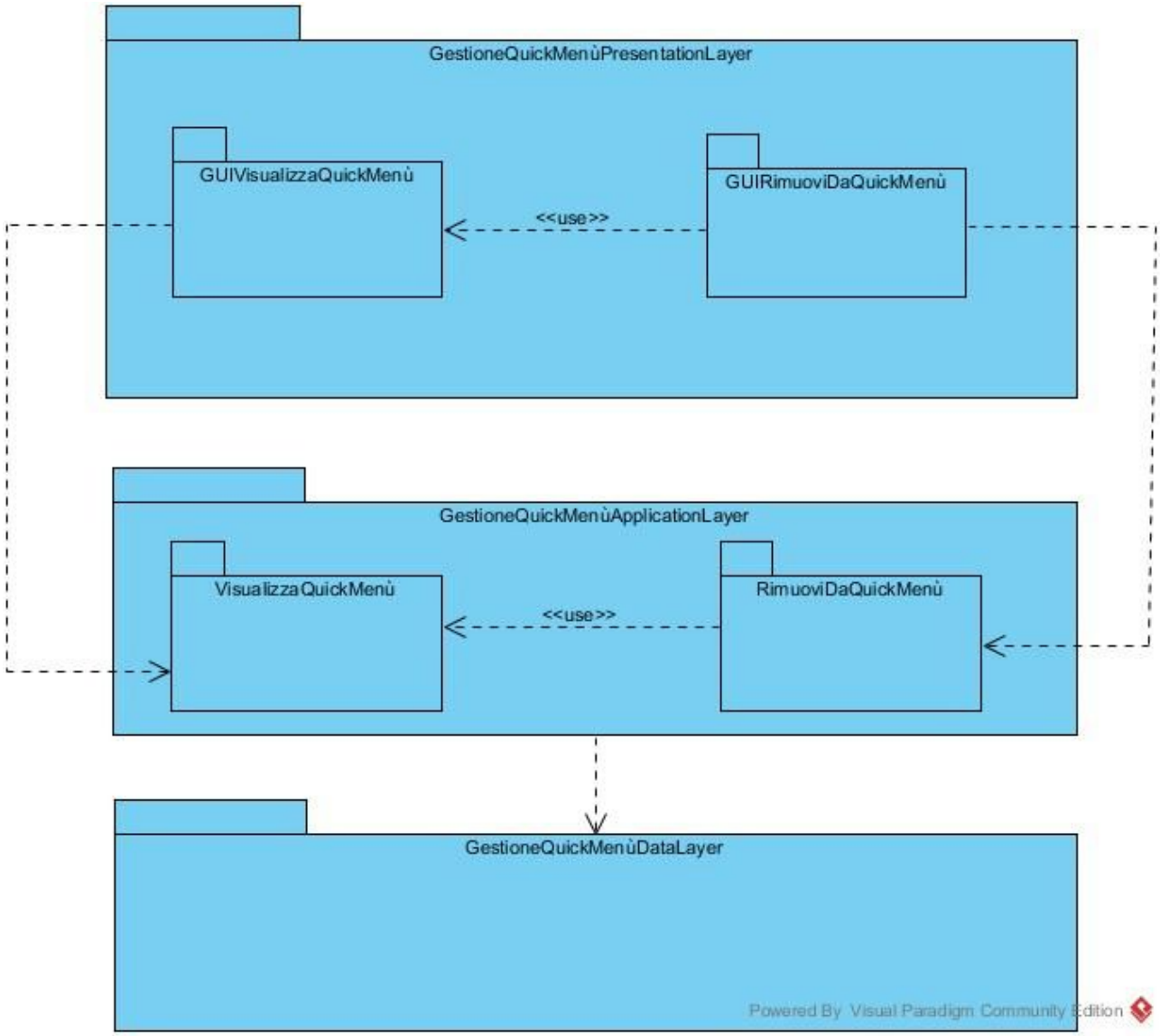
GestioneRecensioniApplicationLayer

Comprende tutte le componenti logiche, responsabili del corretto funzionamento del sistema, e fa da tramite con il database per aggiornare i dati relativi alla gestione delle recensioni.
È riportata di seguito una breve descrizione di ognuno di esse:
•VisualizzaRecensioni:si occupa di eseguire le operazioni necessarie alla visualizzazione degli ordini presenti all'interno del database.
•AggiungiRecensione : si occupa di eseguire le operazioni necessarie all'aggiunta della recensione e memorizzarla all'interno del database.

GestioneRecensioniDataLayer

Sottosistema che si occupa di rendere reperibili i dati, presenti all'interno del database, relativi alla gestione delle recensioni. Questo livello consente di aggiungere le recensioni e memorizzarle nel database.

3.2.7 SubS_5 - Gestione quick menù



GestioneQuickMenùPresentationLayer

Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per la gestione delle recensioni, accessibile da tutti gli utenti del sistema.
È riportata di seguito una breve descrizione di ognuna di esse:
• GUIVisualizzaQuickMenùi: è un'interfaccia tramite la quale l'utente registrato può accedere a tutte le operazioni legate alla visualizzazione del quick menù.
• GUIRimuoviDaQuickMenù: è un'interfaccia tramite la quale l'utente del sistema può accedere a tutte le operazioni legate alla rimozione del menù dal quick menù.

GestioneQuickMenùApplicationLayer

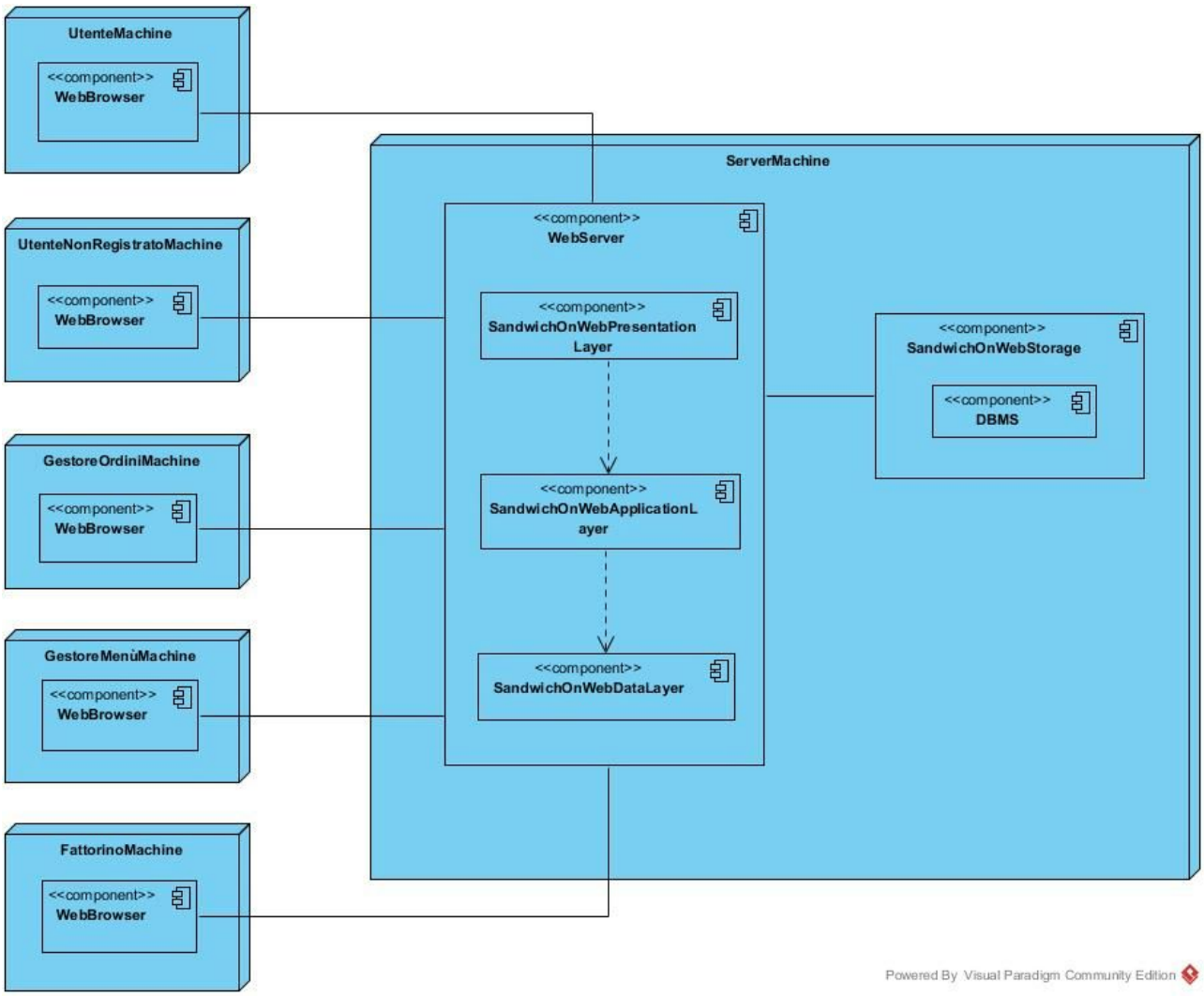
Comprende tutte le componenti logiche, responsabili del corretto funzionamento del sistema, e fa da tramite con il database per aggiornare i dati relativi alla gestione del quick menù.
È riportata di seguito una breve descrizione di ognuno di esse:
•VisualizzaQuickMenù:si occupa di eseguire le operazioni necessarie alla visualizzazione del quick menù.
•RimuoviDaQuickMenù : si occupa di eseguire le operazioni necessarie alla rimozione di un menù dal quick menù e memorizzarla all'interno del database.

GestioneQuickMenùDataLayer

Sottosistema che si occupa di rendere reperibili i dati, presenti all'interno del database, relativi alla gestione del quick menù. Questo livello consente di rimuovere i menù dal quick menù e memorizzarli nel database.

3.3-Hardware-software mapping

Per quanto riguarda il flusso d'informazioni, l'architettura è stata scomposta in due livelli, un livello CLIENT ed un livello SERVER. Rappresentiamo nel seguito la distribuzione delle componenti Hardware e Software sui due nodi.



3.4-Gestione dei dati persistenti

Per maggiori informazioni sulla gestione dei dati persistenti si faccia riferimento al documento Sandwich_on_web_DBD_1.0.

3.5-Controllo degli accessi e sicurezza

Sandwich on web sarà realizzato in modo da interagire con diverse tipologie di utenti, quali l'Utente non registrato, l'Utente registrato, il Gestore degli ordini, il Gestore dei menù, il Fattorino.
Ognuno di essi potrà accedere esclusivamente ad un determinato tipo di funzionalità e dati, ovviamente previo autenticazione. I dati per effettuare quest'ultima saranno concordati nella fase di creazione dell'account, per tutti gli utenti tranne che per gli admin (Gestore degli ordini,Gestore dei menù,

Fattorino), i cui dati saranno impostati di default all'interno del database, non appena creato il sistema.

- L'Utente non registrato può creare il suo account, visualizzare la lista dei menù e visualizzare le recensioni.
- L'Utente registrato può effettuare il login e il logout e modificare i dati del proprio account, visualizzare la lista dei menù,acquistare un menù, aggiungere un menù a quick menù, visualizzare il carrello, rimuovere un menù dal carrello, svuotare il carrello, visualizzare l'ordine e cancellare l'ordine, visualizzare le recensioni e aggiungere una recensione, visualizzare il quick menù e rimuovere un menù dal quick menù.
- Il Gestore degli ordini può effettuare il login e il logout del proprio account, visualizzare l'ordine e modificare lo stato dell'ordine in consegna o in preparazione, modificare il tempo di attesa dell'ordine nella coda.
- Il Gestore dei menù può effettuare il login e il logout del proprio account, visualizzare la lista dei menù e rimuovere un menù dalla lista, modificare i menù presenti nella lista e inserire un nuovo menù.
- Il Fattorino può effettuare il login e il logout del proprio account, visualizzare l'ordine e modificare lo stato dell'ordine in evaso o in ritardo.

Per rappresentare in modo schematico e permettere una lettura immediata delle operazioni consentite agli attori sulle diverse entità del nostro sistema, utilizzeremo la matrice degli accessi, qui di seguito presentata.

3.5.1-Matrice di accesso

Oggetto	Account	Menù	Carrello	Ordine	Recensione	Quick menù
Attore						
Utente non registrato	registraAccount()	visualizzaLista()			visualizzaRecensione()	
Utente registrato	modificaAccount()	visualizzaLista() acquistaMenù() aggiungeAQ	visualizzaCarrello() rimuoveMenù() svuotaCarrel	visualizzaOrdine() cancellaOrdine()	visualizzaRecensione() aggiungeRecensione()	visualizzaQuick() rimuoveMenù()

		uick()	lo()			
Gestore degli ordini				visualizzaOrdine() modificaStato() modificaTempoAttesa()		
Gestore dei menù		visualizzaLista() rimuoveMenù() modificaMenù() inserisceMenù()				
Fattorino				visualizzaOrdine() modificaStato()		

3.5.2 Sicurezza

La sicurezza del sistema è garantita dall'obbligo, di tutte le utenze che lo utilizzano, di autenticarsi, infatti Sandwich on web chiederà come forma di autenticazione l'immissione di un'e-mail e di una password. Tali credenziali dovranno esser inserite ogni volta che si desidera utilizzare il sistema; tale sessione terminerà quando l'utente richiederà esplicitamente di effettuare logout. Nel caso l'accesso al sistema non abbia successo, Sandwich on web mostrerà una schermata in cui sarà specificato se c'è stato un errore di inserimento della password, o se l'e-mail non esiste all'interno del database, consentendo, successivamente, all'utente di inserire le credenziali per effettuare un nuovo tentativo.

Nello specifico la password deve avere una lunghezza minima di 6 caratteri alfanumerici.
In seguito all'autenticazione Sandwich on web mostrerà diverse "viste" dello stesso sistema a seconda dell'utente, ovvero sarà mostrata una schermata contenente le sole funzionalità a cui una determinata tipologia di può accedere.

3.6-Flusso di controllo globale

3.6.1-Flusso di controllo esterno

Il sistema Sandwich on web è principalmente un sistema interattivo: tutte le funzionalità sono avviate in seguito ad un comando dell'utente. Per tale motivo, il sistema userà un meccanismo di controllo di flusso di tipo event-driven, grazie all'utilizzo di un linguaggio fortemente event-driven, qual è Java. Il sistema rimane, quindi, in attesa di una richiesta da parte dell'utente, il quale, tramite la pressione di un bottone, o l'apertura di un menù, scatena un evento, che sarà gestito correttamente dal giusto handler, il quale - a sua volta - indirizzerà il controllo del flusso del sistema alla classe corretta del sottosistema che si occupa della logica di controllo.

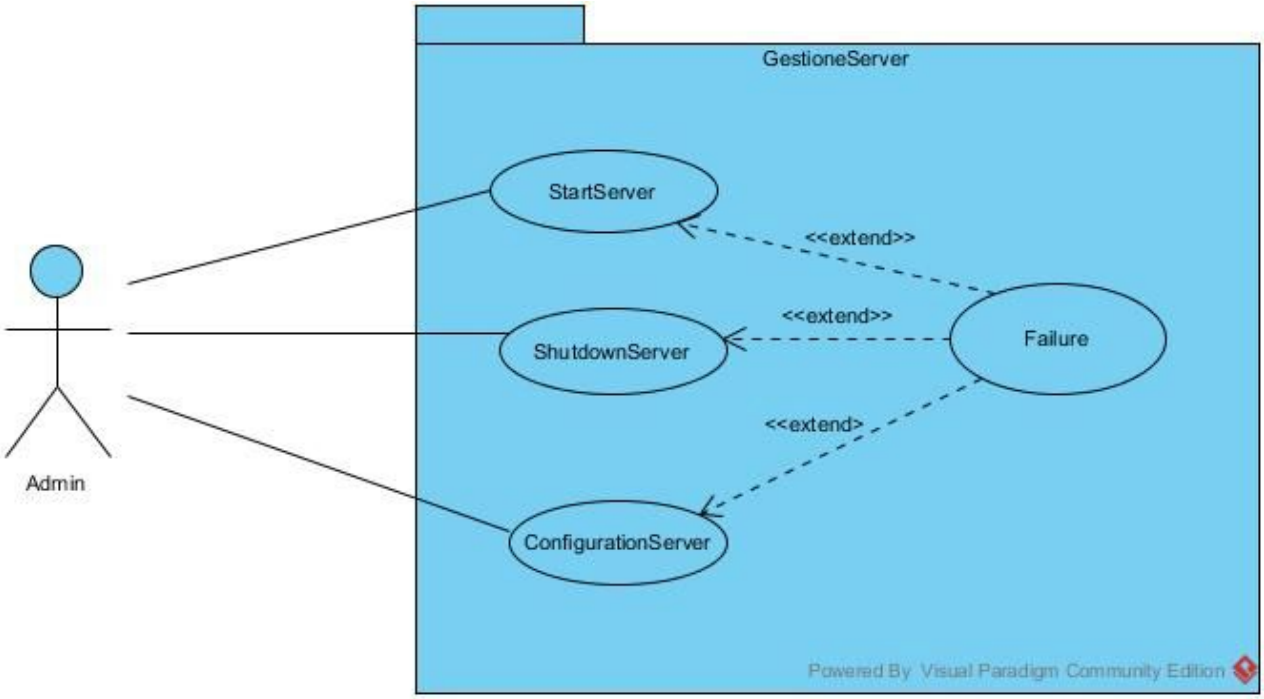
3.6.2-Controllo della concorrenza

Più utenti possono accedere contemporaneamente all'application server; è necessario (per il corretto funzionamento del sistema), quindi, che sia generato un nuovo thread, per ogni utente che richiede un servizio. Sarà poi compito del DBMS occuparsi della concorrenza degli accessi al database. E' necessario, quindi, che si valutino bene le condizioni di concorrenza e che siano prese opportune precauzioni.
Per facilitare la manutenzione del sistema e gestire nella maniera più efficace la coordinazione della concorrenza degli accessi al database, sarà utilizzato sul database server un DBMS: il sistema permette, quindi, l'accesso concorrente all'application server, ma la gestione della priorità d'accesso al database sarà gestita dal DBMS.

3.7-Boundary conditions

3.7.1 Configurations

3.7.1.1 UCD_1 – GestioneServer



3.7.1.2 UC_1.1 - StartServer

Nome caso d'uso	StartServer
Attori partecipanti	Admin
Flusso di eventi	1. L' Admin avvia il server attraverso la funzione "startServer". 2. Sandwich on web avvia il server, mettendo a disposizione dei client le varie funzionalità offerte.
Condizione di entrata	L'admin avvia il server.
Condizione di uscita	Il server è avviato con successo e le varie funzionalità sono messe a disposizione dei client in remoto
Eccezioni	Failure

3.7.1.3 UC_1.2 - ShutdownServer

Nome caso d'uso	ShutdownServer

Attori partecipanti	Admin
Flusso di eventi	1. L'Admin arresta il server attraverso la funzione "ShutdownServer." 2. Sandwich on web provvede ad arrestare il sistema, rimuovendo prima i servizi forniti dallo stesso server ai client.
Condizione di entrata	L'admin avvia il server.
Condizione di uscita	Il server è arrestato.
Eccezioni	Failure

3.7.1.4 UC_1.3 - Failure

Nome caso d'uso	Failure
Attori partecipanti	Admin
Flusso di eventi	1.Sandwich on web rileva un crash nel sistema e presenta una schermata dove è possibile ripristinare il sistema allo stato precendente
Condizione di entrata	Si verifica un crash nel sistema.
Condizione di uscita	L'Admin visualizza una notifica relativa al crash di sistema.
Eccezioni	Failure

3.7.1.4 UC_1.3 - ConfigureServer

Nome caso d'uso	ConfigureServer
Attori partecipanti	Admin
Flusso di eventi	1.L'Admin avvia la configurazione del server, attivando

	la funzionalità “ConfigureServer”. 2. Sandwich on web visualizza una schermata con il pannello di controllo del server. 3. L’Admin controlla se si sono verificati errori nel sistema.
Condizione di entrata	L’admin avvia il server.
Condizione di uscita	L’Admin ha effettuato la configurazione del server e la correzione di un eventuale errore.
Eccezioni	Failure

3.7.2 Start-up del sistema

Essendo Sandwich on web un sistema distribuito, l’avvio e l’esecuzione dei vari sottosistemi (con cui i vari utenti interagiscono attraverso le funzionalità offerte dagli stessi) può avvenire in maniera indipendente.
Dunque, allo start-up del sistema, i vari sottosistemi non hanno la necessità di accedere ad alcun dato: all’avvio i client sono indipendenti dal server, infatti è lanciata un’interfaccia intuitiva, per l’autenticazione dell’utente, che non ha bisogno di alcun dato per la sua creazione.
Tuttavia, per il riconoscimento dell’utente e l’esecuzione delle funzionalità di un sottosistema, è invece indispensabile che i client comunichino con il server, in quanto quest’ultimo si occupa di elaborare le richieste dei client e di inviare, quindi, la corretta query al Database, e, se necessario, di restituire il risultato della stessa query al client.
Inoltre, ad ogni utente saranno offerti servizi diversi in base alla tipologia a cui appartengono.

3.7.3 Fallimento del sistema

Nel caso in cui si verifichi un crash del sistema (dovuto ad un errore hardware o software) nel client o nel server, si tenta un ripristino della configurazione precedente allo stato d’errore, quindi, si forza il riavvio del sistema stesso.
Dato che i dati sono gestiti dal DBMS (che risiede sul database server, separato dai client e dall’application server), non c’è rischio di perdite di dati: le transazioni effettuate con il DBMS sono infatti atomiche.
Tuttavia, in caso di guasti al supporto di memorizzazione dei dati nel database server, si va incontro ad una perdita dei dati.

Per rendere minimo tale rischio, si eseguono periodicamente dei backup del database del sistema. Per evitare problemi, è comunque consigliabile effettuare dei controlli periodici sull'hardware.
Nel caso in cui, invece, le informazioni contenute nei supporti di memorizzazione secondaria (ai quali ha accesso il sistema per la scrittura dei dati persistenti) stiano per superare la capacità di memoria di quest'ultimi, Sandwich on web avvisa l'utente (l'amministratore), lasciandogli la possibilità di gestire tale situazione correttamente (espansione della memoria secondaria o cancellazione di eventuali dati superflui).
Nel caso in cui la memoria secondaria sia completamente esaurita, sono possibili solo operazioni di lettura o sovrascrittura dei dati. In caso di crash dovuto ad un bug nel codice del sistema, si distinguono tre casi:

- Il crash si è verificato nel client. Il sistema potrà, quindi, continuare a funzionare regolarmente, a meno ché non venga riutilizzata la funzionalità che ha causato il crash.
- Il crash si è verificato nell'application server. Il sistema risulta, quindi, inutilizzabile, in quanto non è possibile comunicare con il database server.
- Il crash si è verificato nel database server. Come detto precedentemente, il sistema risulta inutilizzabile, in quanto non è possibile accedere ai dati.

In tutti e tre i casi, Sandwich on web provvederà a segnalare nella maniera opportuna il tipo di problema riscontrato. Nel caso in cui uno dei sottosistemi risulti inaccessibile, a causa di problemi legati alla rete, Sandwich on web tenta nuovamente di stabilire la connessione.

3.7.4 Terminazione del sistema

La terminazione del sistema avviene solo nel caso in cui tutti i sottosistemi siano stati disattivati: nel caso in cui due sottosistemi siano ancora in esecuzione, il sistema rimane attivo. Per non incorrere in problemi, che possano scoraggiare il cliente nell'uso del sistema, prima di disattivare l'application server e il database server, è consigliabile disattivare prima tutti i client. In questo caso, non si ha la necessità di notificare la terminazione di un client al sistema.
Ogni sottosistema può essere disattivato tramite la funzione di logout, presente su tutte le postazioni, tramite le quali i vari utenti possono accedere al sistema. Questa funzionalità, sulla postazione dell'utente, ha come unico effetto la chiusura del suo terminale; sull'application server e sul database server, invece, ha l'effetto di disattivare tali sottosistemi.

In questo caso, se vi fosse ancora in rete qualche client attivo, questi verrebbero notificati, affinché possano disattivarsi. Si procede, quindi, con l'arresto dell'intero sistema.

4. Glossario

4.1 Definizioni acronimi e abbreviazioni

TERMINI	DESCRIZIONE
RF	Requisiti funzionali
RNF	Requisiti non funzionali
SC	Scenari
DG	Design goals
AM	Matrice di accesso
SS	Servizi sottosistema
TO	Trade-offs
UC	Casi d'uso
UCD	Diagrammi dei casi d'uso
DD	Dizionario dei dati
DB	Database
DBMS	Database Management System
SubS	Sub System