Corso di Ingegneria del Software

Restaurant Management
System Design
Versione 1.1

Data: 10/12/2018

Partecipanti al progetto

Nome	Matricola
Andrea Cipriano	512104874
Gianmarco Cringoli	512104778
Manuel Flora	512104628

Scritto da:	Andrea Cipriano, Gianmarco Cringoli, Manuel Flora

Revision History

Data	Versione	Descrizione	Autore
10/12/18	1.1	System decomposition	Andrea Cipriano
10/12/18	1.1	System decomposition	Gianmarco Cringoli
10/12/18	1.1	System decomposition	Manuel Flora

Indice

1. Introduction

Con l'obiettivo di ottimizzare la gestione delle attività ristorative di piccola-media dimensione, si vuole sviluppare un progetto che miri a semplificare le funzioni che il gestore di un ristorante e i camerieri svolgono quotidianamente. Il software, inoltre, è rivolto anche ai clienti dei ristoranti, i quali possono interagire con esso attraverso il web.

1.1. Purpose of the system

L'obiettivo del sistema "Restaurant Management" è quello di creare una piattaforma per la gestione delle attività di ristorazione di media-piccola dimensione. In particolare "Restaurant Management" nasce dall'idea di:

- fornire un servizio per i clienti del ristorante che gli permette di prenotare un tavolo attraverso il web.
- fornire la possibilità al gestore di creare menù e profili cameriere per migliorare la gestione delle ordinazioni.
- offrire un sistema di gestione comande che permette ai camerieri di comunicare con la cucina.

1.2. Design goals

Il sistema sarà strutturato in maniera chiara e completa. Le operazioni che l'utente effettuerà non richiedono particolare conoscenza tecnologica poichè l'utilizzo sarà guidato dalla semplicità ed intuitività dell'interfaccia.
L'interfaccia grafica sarà curata nei minimi dettagli, mediante l'utilizzo di bottoni, finestre di dialogo non invasive, label semplici e icone dettagliate che offrono all'utente un'esperienza di utilizzo rapida ed esaustiva.

1.2.1. Dependability criteria

La web application garantirà il corretto funzionamento, gestendo i vari tipi di errori. Quindi Restaurant Management rispetterà i seguenti requisiti di qualità, relativi all'affidabilità:

- ***Robustezza (priorità alta):*** nel caso i cui l'utente inserisca dati errati nel sistema, quest'ultimo farà visualizzare dei messaggi di errore avvisando l'utente che i dati non sono validi. [Requisito non funzionale: Reability]
- ***Affidabilità (priorità alta):*** il sistema garantisce il corretto svolgimento delle proprie funzionalità, producendo sempre l'output desiderato [Requisito non funzionale: Reability]
- ***Fault tolerance (priorità alta):*** il sistema garantisce una tolleranza media agli errori, quando si verificassero esso dovrà essere in grado di gestirli e risolverli nel minor tempo possibile.

1.2.2. Performance criteria

Il sistema garantirà buone performance gestendo adeguamente tutti gli utenti senza rallentamenti.

1.2.3. End user criteria

Dal punto di vista dell'utente, il sistema dovrà garantire i seguenti requisiti di qualità:

- ***Utilità (priorità alta):*** attraverso l'attività di raccolta dei requisiti, il sistema sarà in grado di soddisfare le esigenze degli utenti.
- ***Usabilità (priorità alta):*** il sistema dovrà essere intuitivo e di semplice utilizzo, e sarà progettato tenendo conto di quella che è l'user experience. Non sarà necessario l'uso di un manuale utente per compiere le azioni. [Requisito non funzionale: Usability]

1.3. Definitions, acronyms, and abbreviations

Acronimi	Descrizione
RAD	Requirement Analysis Document
SDD	System Design Document
HW	Hardware
SW	Software
SQL	Structure Query Language
DBMS	Database Management System
GUI	Graphical User Interface

1.4. References

- Problem Statement v.3.0
- RAD_V_3.0
- B.Brugge, A.H. Dutoit, Object Oriented Software Engineering- Using UML, Patterns and Java, Prentice Hall.
- https://www.bruegge.in.tum.de

1.5. Overview

Il seguente documento sarà così strutturato:
Introduction*:* riporta una descrizione del sistema specificando le ragioni del suo sviluppo, le caratteristiche e un accenno sull'utilizzo delle funzionalità.
Proposed software architecture: fornisce una panoramica sull'architettura usata per il sistema. Tratta della suddivisione in sottosistemi, del mapping software-hardware, della gestione dei dati persistenti, del controllo degli accessi in sicurezza, del flusso di controllo globale e delle condizioni limite.
Subsystem services*:* espone una descrizione dei sottosistemi identificati e, per ognuno, i servizi offerti.
Glossary: è una raccolta di termini nell'ambito specifico

2. Current software architecture

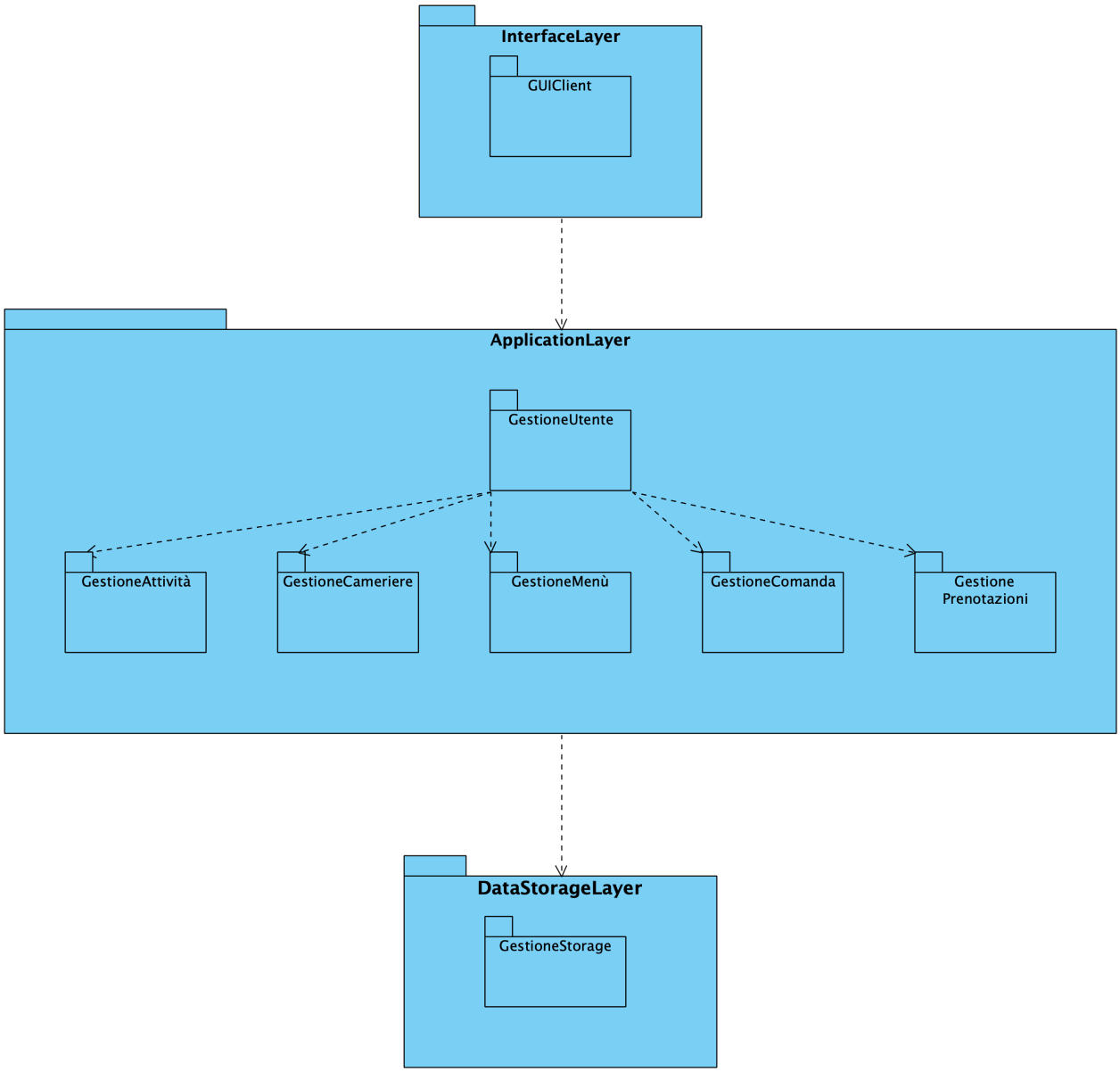
L'applicazione proposta non andrà a sostituire nessun sistema già esistente. La progettazione e lo sviluppo del software partirà da zero e seguirà i criteri della Greenfield Engineering.

3. Proposed software architecture

3.1. Overview

L'architettura scelta per il sistema da realizzare sarà quella Three-layer. L'utente potrà interagire con l'application layer mediante l'interface layer che offrirà diverse interfacce di base alle necessità dell'utente. L'application layer dovrà poi comunicare con i database per la memorizzare dei dati persistenti. Sul server, risiede un DBMS che si occupa di recuperare, memorizzare ed interrogare i dati presenti nel database, elaborando, quindi, la richiesta dell'utente. l'aspetto della concorrenza di accessi multipli al database, sarà pertanto gestito dal DBMS stesso che dovrà evitare eventuali colli di bottiglia. 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.

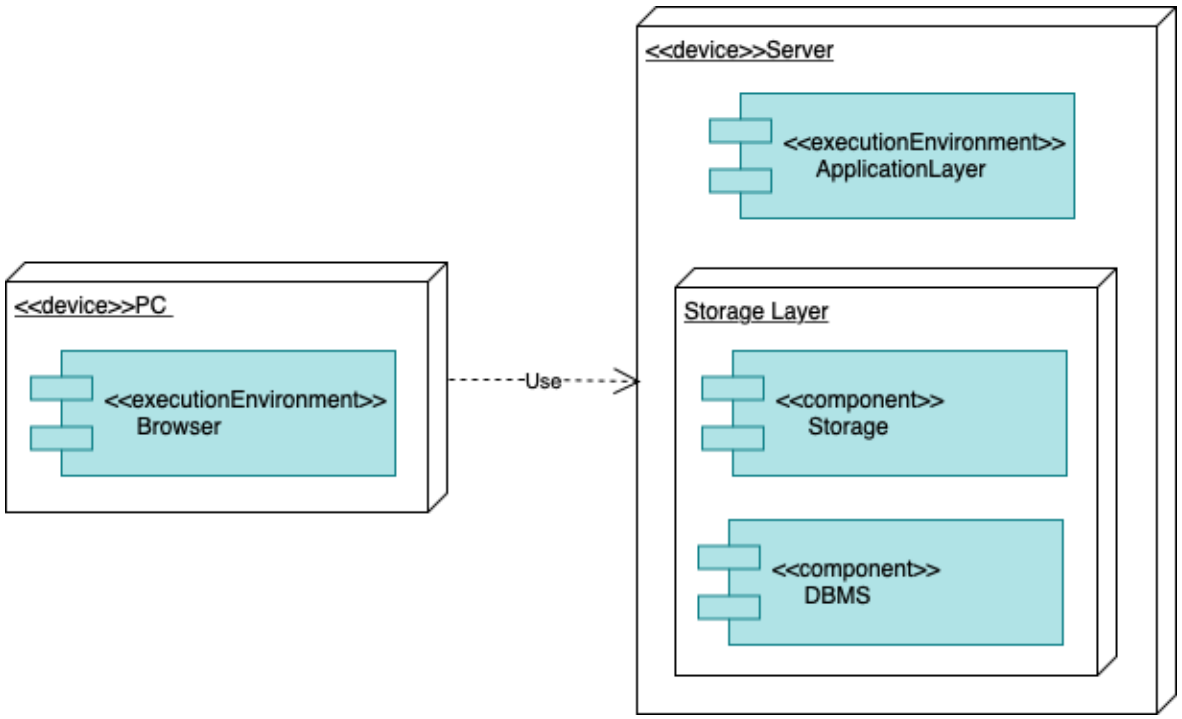
3.2. Subsystem decomposition



3.2.1. Schema generale

- ***InterfaceLayer***: GUIClient è l'interfaccia col la quale gli utenti interagiscono con il sistema.
- ***ApplicationLogicLayer***: contiene i seguenti sottosistemi: gestione utente, gestione cameriere, gestione attività, gestione menu, gestione prenotazioni
- ***GestoreStorage***: sottosistema che ha il compito di effettuare operazioni sul database.

3.3. Hardware/software mapping



Il sistema sviluppato sarà installato su un solo server e utilizzerà un DBMS MySQL stanziato sullo stesso per la gestione dei dati persistenti. Il sistema sarà diviso in un'architettura client e server.

Il Deployment Diagram mostra le componenti che utilizzerà il nostro sistema; questo diagramma aiuta gli sviluppatori a comprendere le relazioni tra le componenti software e i nodi hardware. Al lato client, l'interfaccia utente verrà mostrata sul browser web ed interagirà con l'application layer, che a sua volta memorizza e interroga e dati presenti nel database.

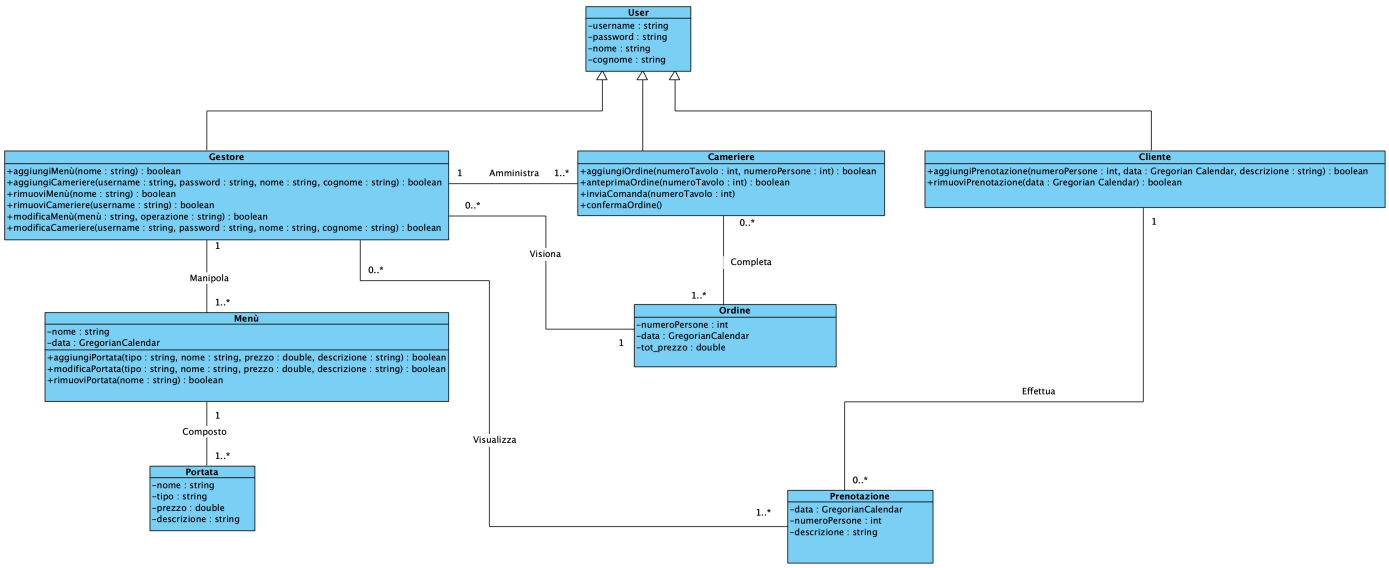
3.4. Persistent data management

Il sistema userà sia un servizio di storage su file, che servizi di storage su Database. Le risorse saranno opportunamente salvate in cartelle sul server.

La scelta del DBMS è ricaduta su MySQL viste le conoscenze del team. La scelta di un database relazionale, rispetto a un database object-oriented, è motivata dalle alte prestazioni offerte dai primi.

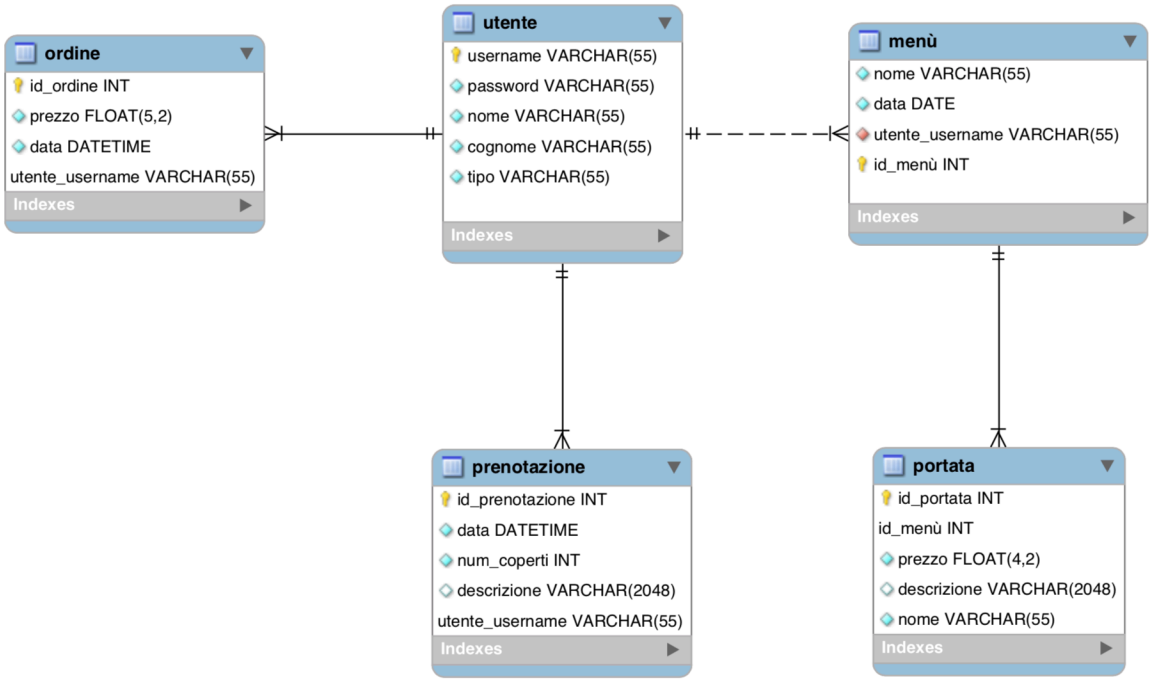
3.4.1. Class diagram

Il class diagram qui mostrato descrive le entità che saranno salvate nel database. A differenza del class diagram presente nel RAD_V3, abbiamo effettuato delle modifiche. Le modifiche riguardo le entità: cucina, tavolo, portata comanda, ristorante; le quali possono essere salvate direttamente in un file anziché nel database.



3.4.2. Modello logico

Il modello logico rappresenta la struttura del database.



3.4.2.1. Dizionario dei dati

Tabella	
Utente	**Attributi**
	• ***username***: chiave primaria utente • ***password***: attributo • ***nome***: attributo per il nome dell'utente • ***cognome***: attributo per il cognome dell'utente • ***tipo:*** attributo per distinguere i vari utenti (gestore, cameriere, cliente)
Tabella	
Menù	**Attributi**
	• ***id_menù:*** chiave primaria menù • ***nome:*** attributo per il nome del menù • **data**: attributo per la data di aggiunta
Tabella	
Portata	**Attributi**
	• ***id_portata:*** chiave primaria portata • ***nome***: attributo per il nome della portata • ***prezzo***: attributo per il prezzo della portata • ***descrizione***: attributo per la descrizione della portata
Tabella	
Prenotazione	**Attributi**
	• ***id_prenotazione***: chiave primaria prenotazione • ***data***: attributo per la data della prenotazione • ***numeroPersone***: attributo per il numero delle persone • ***descrizione***: attributo per la descrizione di una prenotazione • ***utente_username:*** vincolo di integrità referenziale
Tabella	
Ordine	**Attributi**
	• ***id_ordine***: chiave primaria ordine • ***totale***: attributo per l'ammontare dell'ordine • ***data***: attributo per la data dell'ordine • ***utente_username***: vincolo di integrità referenziale con l'utente (cameriere)

3.5. Access control and security

Il controllo degli accessi sarà alla base della sicurezza del nostro sistema, permettendo ad ogni utente di collegarsi al sistema tramite l'utilizzo di username e password, che verranno richieste ad ogni singolo accesso. La sessione termina quando l'utente effettua il logout.

Nel casi l'accesso al sistema non abbia successo, verrà inviata una notifica di fallimento indicando che è avvenuto un errore nell'inserimento di username o password e consentendo all'utente di effettuare un nuovo tentativo.

L'utente "cliente" potrà usufruire dei sevizi che non utilizzano la registrazione, ma nel caso cui lui vorrebbe effettuare una prenotazione verrà reindirizzato alla pagina di login o registrazione.

Il sistema fornirà più visite (interfacce grafiche) a seconda dell'attore che ci interagirà, in modo che ognuno posso accedere solo alle rispettive funzionalità previste.

“”tabella””???

3.6. Global software control

Per quanto riguarda il flusso di controllo esterno fra sottoinsiemi, il server sarà sempre in funzione in attesa di eventuali richieste da parte dell'utenza; il sistema non avrà problemi a gestire più utenti contemporaneamente.

3.7. Boundary conditions

Il server sarà sempre attivo permettendo l'utilizzo del servizio in qualsiasi orario e in presenza di un malfunzionamento, il sistema mostrerà una pagina di manutenzione.

3.7.1. Start-up

Per il primo start-up del sistema "RestaurantManagement", è necessario l'avvio di un web server che fornisca il servizio di un Database MySQL per la gestione dei dati persistenti e l'interpretazione ed esecuzione del codice lato server, assumendo che, prima dello start-up iniziale, il database ha un gestore che, una volta loggato, può gestire il proprio ristorante. In seguito, tramite la homepage, qualsiasi utente potrà usufruire di alcuni servizi, ma solo dopo essersi registrato ed effettuato il login può accedere a tutte le funzionalità a lui dedicate.

3.7.2. Terminazione

La chiusura della web page non comporta un logout dal sistema. Per effettuare il logout bisogna cliccare sul pulsante specifico

3.7.3. Fallimento

Un esempio di fallimento possibile generato dalla parte del sistema:

- malfunzionamento del server dovuto ad un improvviso guasto al disco. La priorità del sistema nell'affrontare questo tipo di problema è quella di perdere meno dati possibili. A questo scopo potrebbe essere usata questa strategia:
  - salvare su GitHub, e sul personal computer dell'amministratore, la logica di business dell'applicazione;

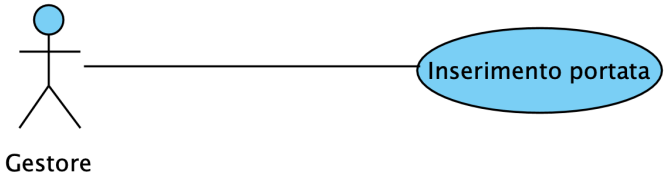
Un esempio di fallimento possibile generato dalla parte Client:

- nell'inserimento dei dati, per accedere al sistema oppure in fase di creazione (di una qualsiasi cosa, menù, cameriere ecc.), o modifica, se i campi non verranno completati

- correttamente e/o non avranno riscontro dal database, l'accesso, l'inserimento o modifica fallirà, mostrando un messaggio d'errore del mancato accesso/ inserimento/modifica.

3.7.4. Use case StartServer

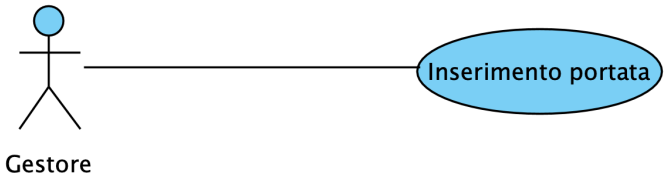
Nome caso d'uso: ***StartServer***	



Attori partecipanti:	Amministratore.
Flusso d'eventi:	1. L'amministratore avvia il server. 2. Il server segnala che l'avvio è avvenuto con successo e i suoi servizi vengono messi a disposizione degli utenti.
Entry condition:	• Il server è spento e funzionante.
Exit condition:	• Il server viene avviato con successo.
Exceptions:	Il sistema segnala che non è possibile avviare il server e che, quindi, non viene avviato con successo.

3.7.5. Use case ShutdownServer

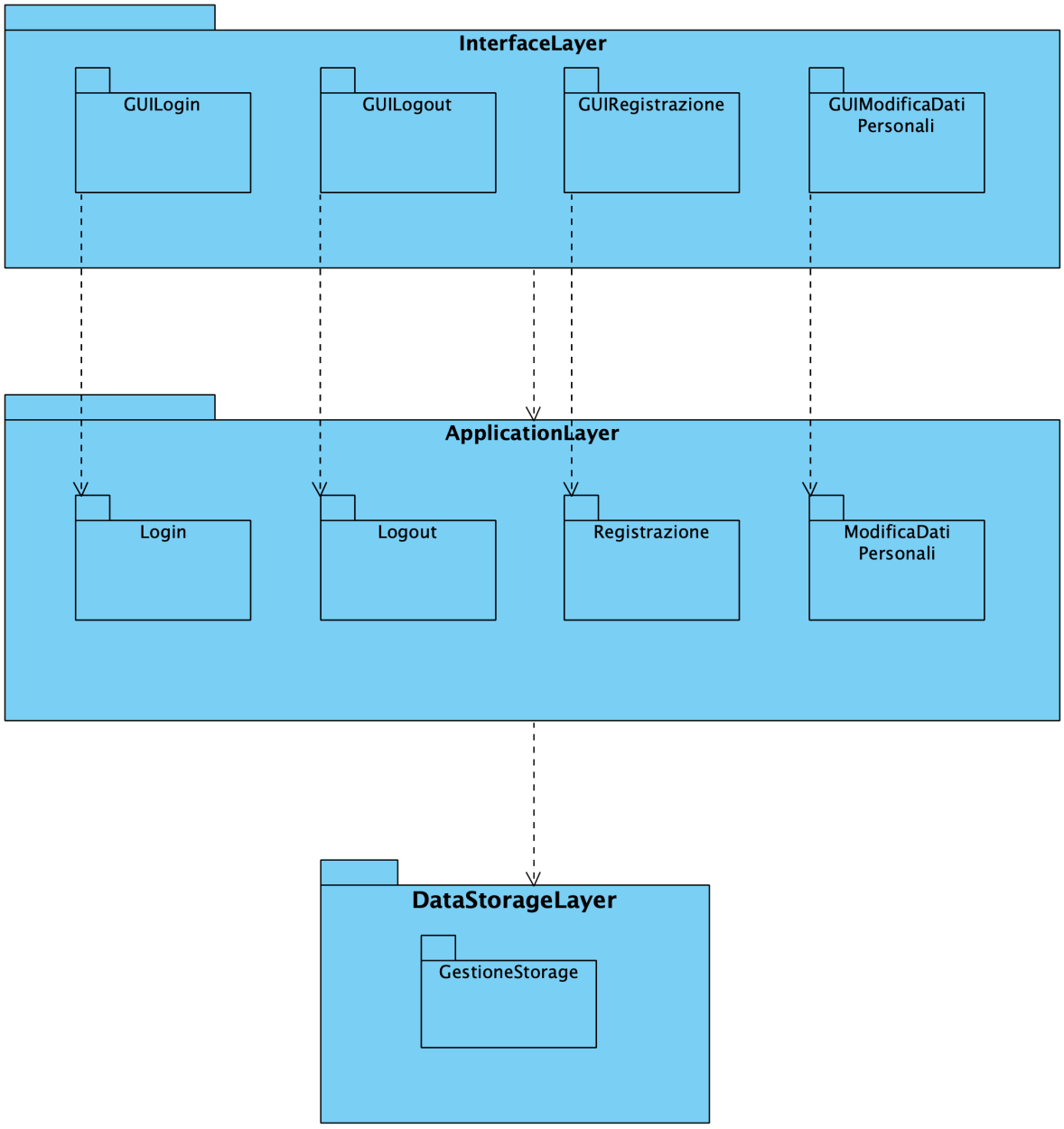
Nome caso d'uso: ***ShutdownServer***	



Attori partecipanti:	Amministratore.
Flusso d'eventi:	1. L'amministratore spegne il server.. 2. Il server si spegne
Entry condition:	• Il server è acceso.
Exit condition:	• Il server viene arrestato.

4. Subsystem services

4.1. Gestione utente



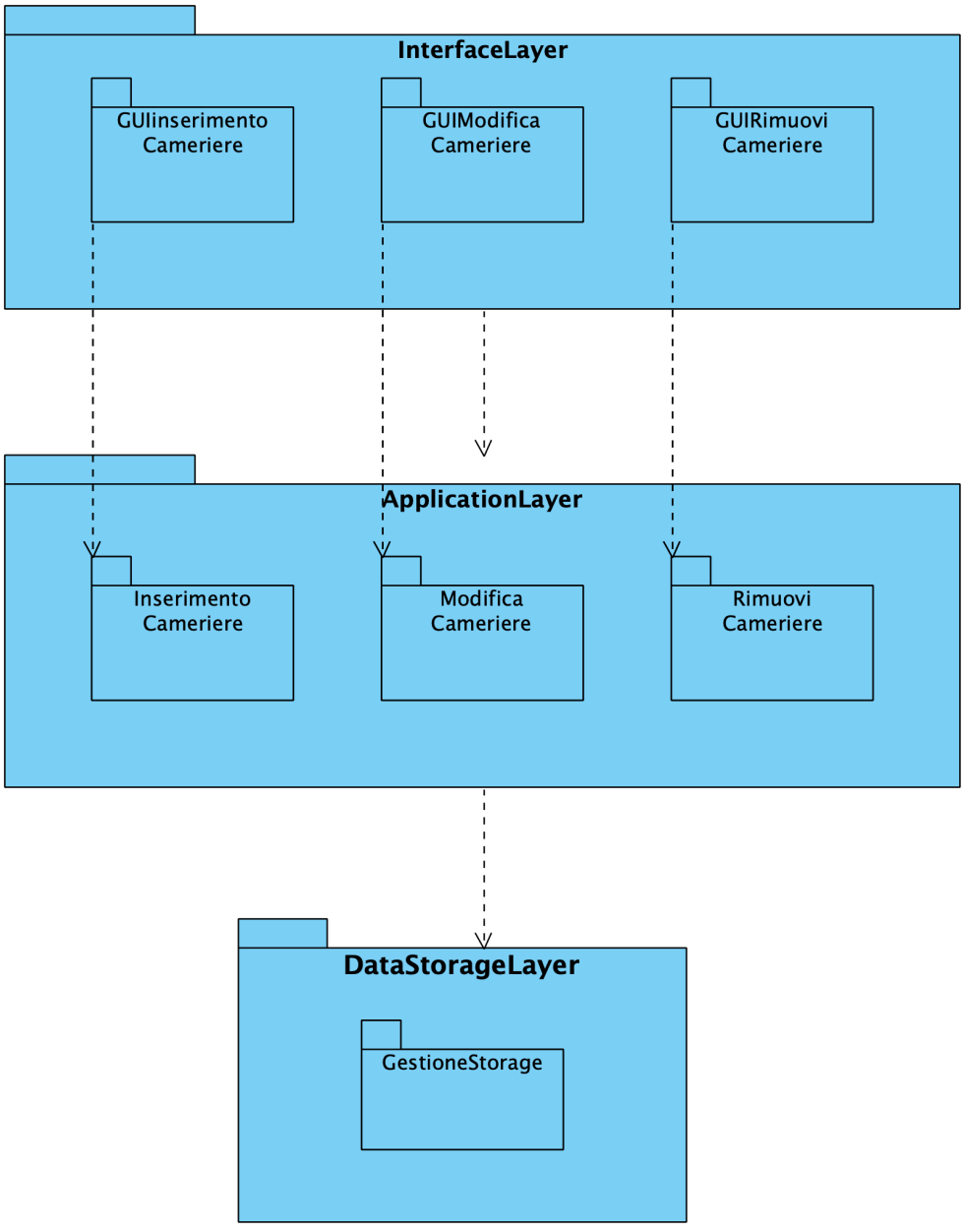
- ***InterfaceLayer***
  Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per l'accesso a RestaurantManagement accessibili da parte degli utenti.

  GUILogin: comprende l'interfaccia per effettuare il login.
  GUILogout: comprende l'interfaccia per effettuare il logout dal sistema.

GUIRegistrazione: comprende l' interfaccia attraverso la quale il cliente può registrarsi.
GUIModificaDati: comprende l'interfaccia per la modifica dei dati personali.

- ***ApplicationLogicLayer***
  Comprende tutte le componenti logiche associate all'accesso al sistema
  Registrazione: operazione che permette al cliente di registrarsi al sistema
  Login: permette agli utenti di autenticarsi al sistema.
  Logout: permette agli utenti di scollegarsi dal sistema.
  ModificaDatiPersonali: permette di modificare i dati personali

- ***DataStorageLayer***
  Comprende il GestoreStorage che permette di effettuare operazioni sul database

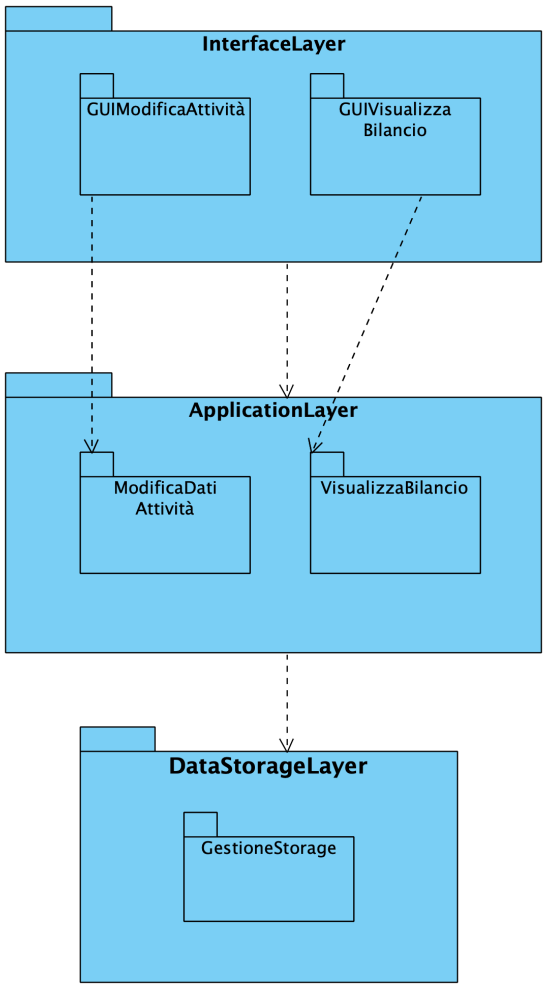


4.2. Gestione cameriere

- ***InterfaceLayer***
  Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per la gestione dei profili "cameriere":
  GUIInserimentoCameriere: interfaccia che permette al gestore di creare i profili "cameriere".
  GUIIModificaCameriere: interfaccia che permette al gestore di modificare i profili "cameriere".
  GUIIRimuoviCameriere: interfaccia che permette al gestore di rimuovere i profili "cameriere".

- ***ApplicationLogicLayer***
  Comprende tutte le componenti logiche associate alla gestione di un profilo "cameriere
  InserimentoCameriere: permette di inserire un account "cameriere" attraverso il quale i camerieri possono accedere al sistema.
  ModificaCameriere: permette di apportare modifiche ad un account "cameriere" creato in precedenza.
  RimuoviCameriere: permette di eliminare un account "cameriere" creato in precedenza.

- ***DataStorageLayer***
  Comprende il GestoreStorage che permette di effettuare operazioni sul database



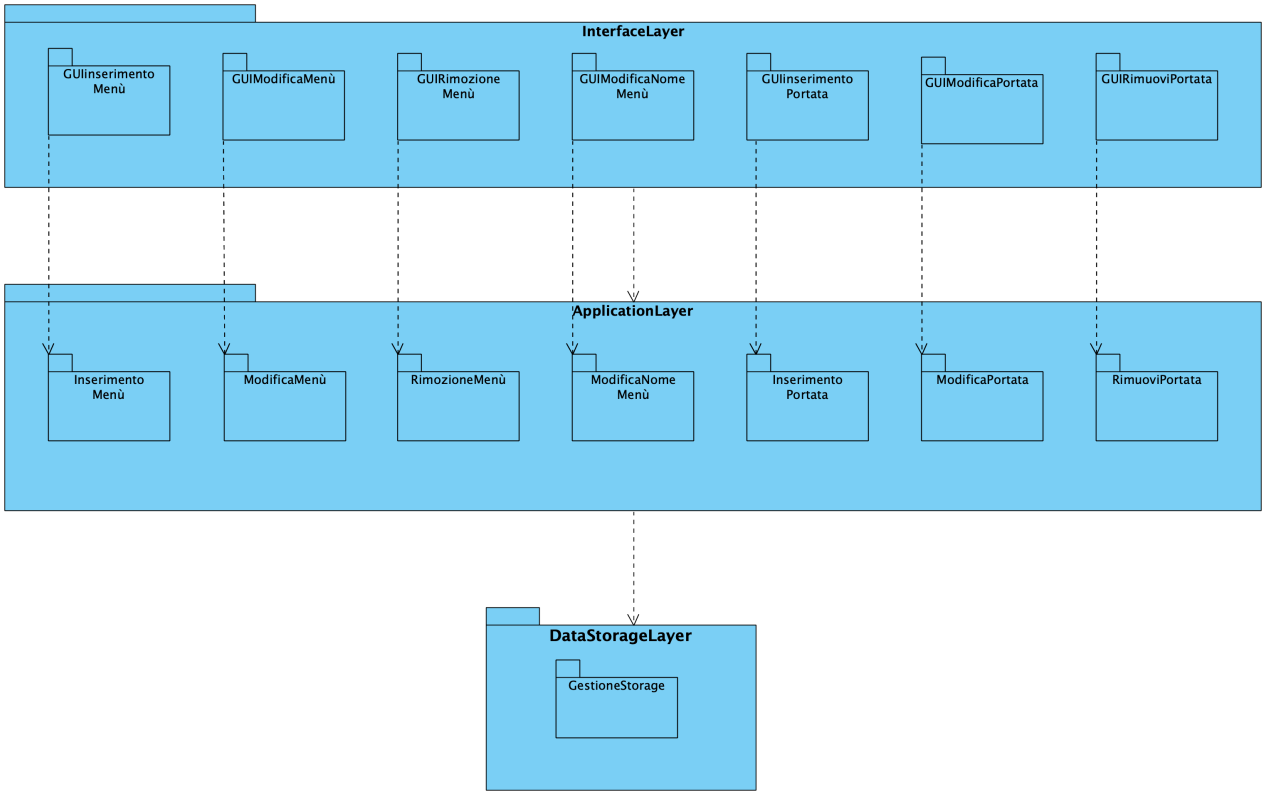
4.3. Gestione attività

- ***InterfaceLayer***
  Include tutte le componenti dell'interfaccia grafica che permettono al gestore di accedere a funzionalità legate alla propria attività ristorativa.
  GUIModificaDatiAttività: interfaccia che permette al gestore di modificare i dati dell'attività.
  GUIVisualizzaBilancio: interfaccia attraverso la quale il gestore consulta il bilancio

- ***ApplicationLogicLayer***
  Comprende tutte le componenti logiche associate alla gestione delle informazioni relative al ristorante.
  Modifica dati attività: permette di modificare i dati dell'attività ristorativa inseriti al primo accesso al sistema
  Visualizza bilancio: permette di visualizzare una panoramica sul bilancio dell'attività

- ***DataStorageLayer***
  Comprende il GestoreStorage che permette di effettuare operazioni sul database

4.4. Gestione menù



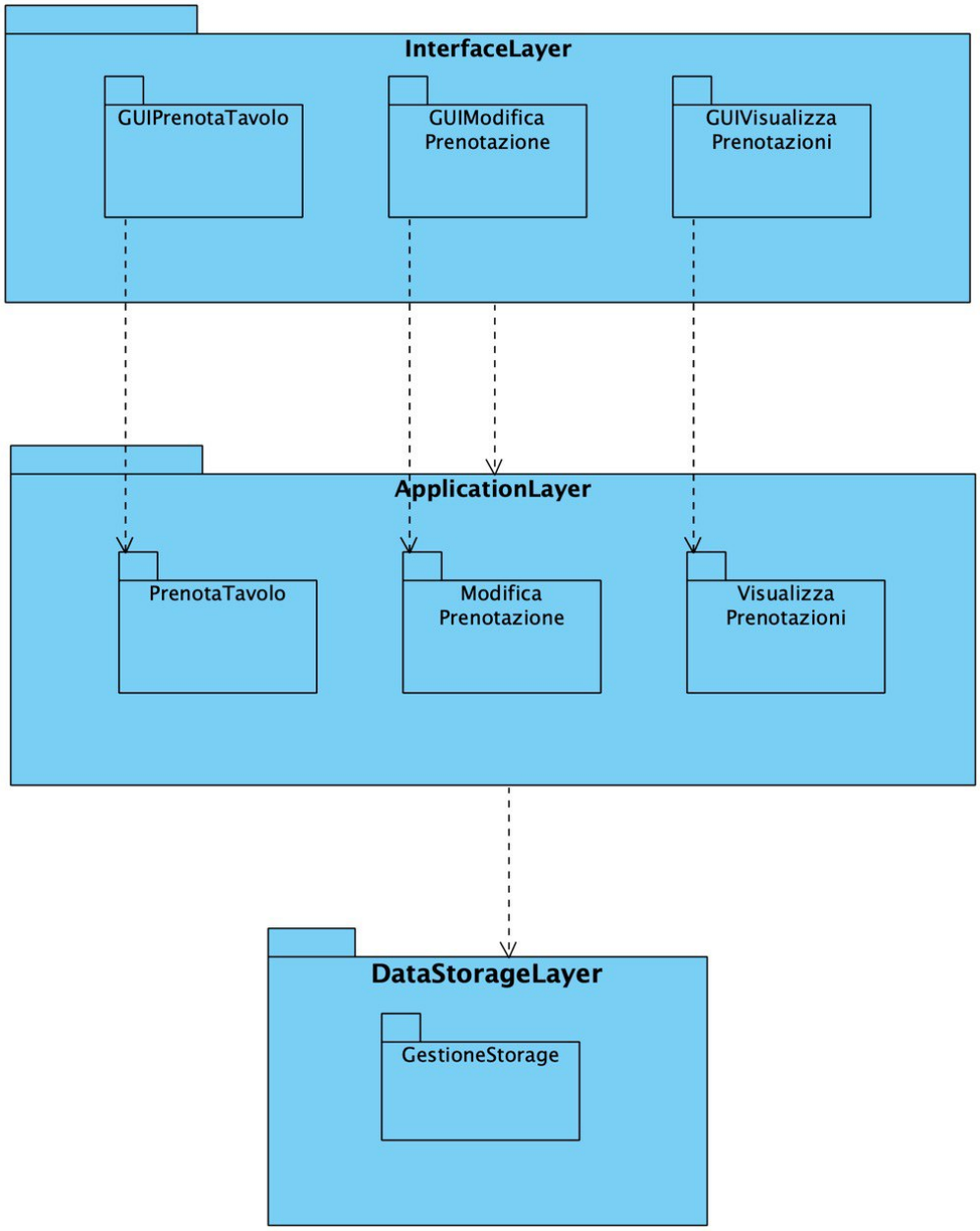
- ***InterfaceLayer***
  Include tutte le componenti dell'interfaccia grafica che permettono al gestore di accedere a funzionalità legate alla gestione dei menu

GUIInserimentoMenu: interfaccia attraverso la quale si può inserire un nuovo menu.
GUIModificaMenu: interfaccia attraverso la quale si può modificare un menu.
GUIRimozioneMenu: interfaccia attraverso la quale si può eliminare un menu.
GUIModificaNomeMenu: interfaccia attraverso la quale si può modificare il nome di un menu.
GUIInserimentoPortata: interfaccia attraverso la quale si può inserire una portata al menu.
GUIModificaPortata: interfaccia attraverso la quale si può modificare una portata del menu.
GUIRimuoviPortata: interfaccia attraverso la quale si può rimuovere una portata al menu.

- ***ApplicationLogicLayer***
  Comprende tutte le componenti logiche associate alla gestione dei menu e delle relative portate
  InserimentoMenu: permette di inserire un nuovo menu.
  ModificaMenu: permette di modificare uno dei menu inseriti.
  RimozioneMenu: permette di eliminare definitivamente un menu, con le rispettive portate.
  ModificaNomeMenu: permette di modificare il nome di un menu inserito.
  InserimentoPortata: permette di aggiungere una portata ad un menu.
  ModificaPortata: permette di modificare tutti gli attributi relativi ad una portata
  RimozionePortata: permette di rimuovere una portata da un menu.

- ***DataStorageLayer***
  Comprende il GestoreStorage che permette di effettuare operazioni sul database

4.5. Gestione prenotazioni

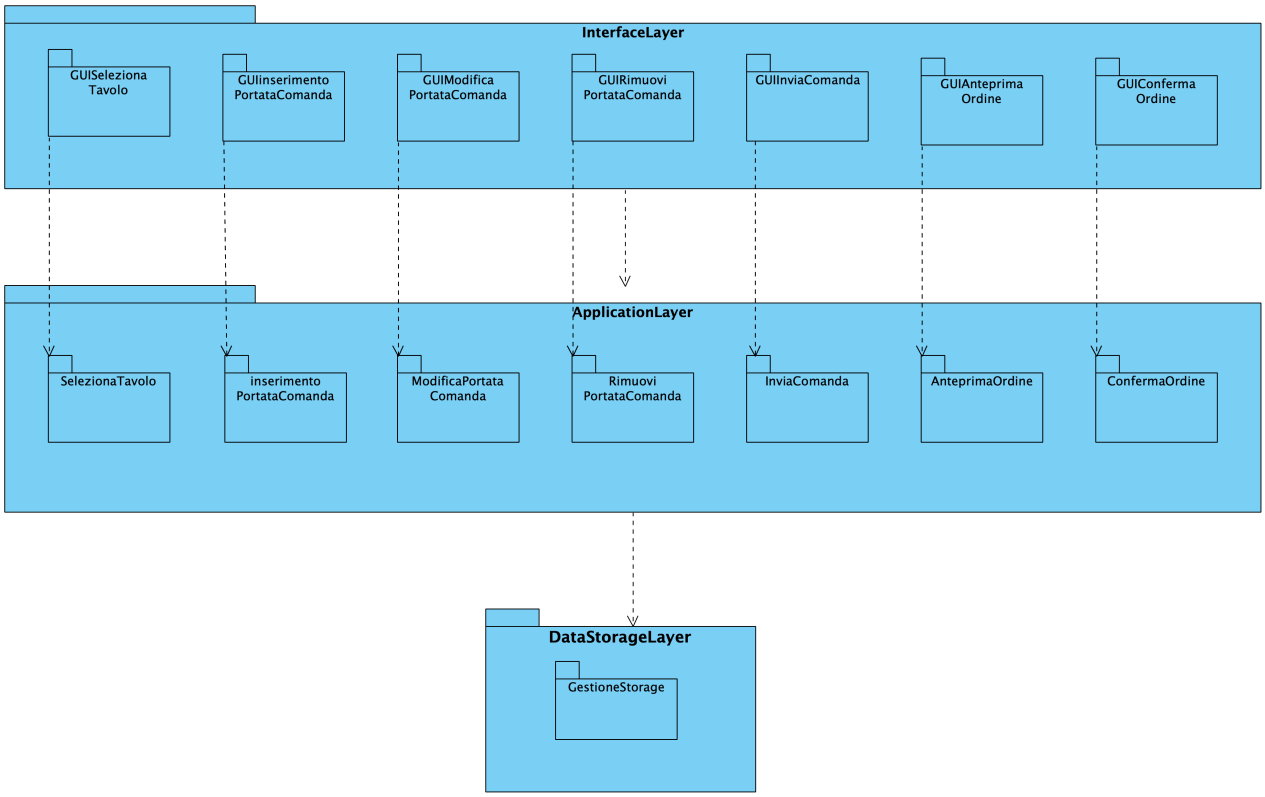


- ***InterfaceLayer***
  Include tutte le componenti dell'interfaccia grafica che permettono di gestire le prenotazioni.
  GUIPrenotaTavolo: interfaccia attraverso la quale il cliente può prenotare un tavolo
  GUIModificaPrenotazione: interfaccia attraverso la quale il cliente può modificare la prenotazione di un tavolo.
  GUIVisualizzaPrenotazioni: interfaccia attraverso la quale si possono visualizzare le prenotazioni effettuate.

- ***ApplicationLogicLayer***
  Comprende tutte le componenti logiche associate alla gestione delle prenotazioni
  PrenotaTavolo: permette di prenotare un tavolo presso il ristorante.
  VisualizzaPrenotazioni: permette di visualizzare lo storico delle prenotazioni effettuate
  RimozionePrenotazione: permette, laddove è possibile, di disdire una prenotazione

- ***DataStorageLayer***
  Comprende il GestoreStorage che permette di effettuare operazioni sul database

4.6. Gestione comanda



- ***InterfaceLayer***
  Include tutte le componenti dell'interfaccia grafica che permettono al gestore di accedere a funzionalità legate alla gestione dei menu
  GUISelezionaTavolo: interfaccia attraverso la quale si può inserire un nuovo menu.
  GUInserimentoPortataComanda: interfaccia attraverso la quale si può modificare un menu.
  GUModificaPortataComanda: interfaccia attraverso la quale si può eliminare un menu.
  GUIRimuoviPortataComanda: interfaccia attraverso la quale si può modificare il nome di un menu.
  GUIInviaComanda: interfaccia attraverso la quale si può inserire una portata al menu.
  GUIAnteprimaOrdine: interfaccia attraverso la quale si può modificare una portata del menu.

GUIConfermaOrdine: interfaccia attraverso la quale si può rimuovere una portata al menu.

- ***ApplicationLogicLayer***
  Comprende tutte le componenti logiche associate alla gestione dei menu e delle relative portate
  SelezionaTavolo: interfaccia attraverso la quale si può inserire un nuovo menu.
  InserimentoPortataComanda: interfaccia attraverso la quale si può modificare un menu.
  ModificaPortataComanda: interfaccia attraverso la quale si può eliminare un menu.
  RimuoviPortataComanda: interfaccia attraverso la quale si può modificare il nome di un menu.
  InviaComanda: interfaccia attraverso la quale si può inserire una portata al menu.
  AnteprimaOrdine: interfaccia attraverso la quale si può modificare una portata del menu.
  ConfermaOrdine: interfaccia attraverso la quale si può rimuovere una portata al menu.

- ***DataStorageLayer***
  Comprende il GestoreStorage che permette di effettuare operazioni sul database

5. Glossary

- ***Design goals***: obiettivi qualitativi del sistema, identificano le qualità su cui deve essere focalizzato il sistema.
- ***Dependability Criteria:*** quantificazione dello sforzo che deve essere speso per minimizzare i crash del sistema e delle loro conseguenze.
- ***Perfomance Criteria:*** requisiti imposti sul sistema in termini di spazio e velocità.
- ***End User Criteria:*** qualità non incluse nei criteri di performance e affidabilità che sono desiderabili dal punto di vista dell'utente.
- Interface layer
- Application logic layer
- Data storage:
- Persistent data management:
- ***Modello logico:*** il modello logico discende dal modello concettuale e disegna un 0architettura che tiene conto delle strutture proprie di quel particolare tipo di database.
- ***Query:*** interrogazione di un database per estrarre o aggiornare i dati che soddisfano un certo criterio di ricerca.
- ***Dizionario dei dati:*** permette di arricchire il modello logico con descrizioni in linguaggio naturale.
- ***Boundary conditions***: soluzioni a problemi fisici.