



# System design document

## The Spoon



Versione	1.0
Data	13/12/2023
Destinatario	Esame di Ingegneria del Software 2023/24
Presentato da	Alessandro Pascarella (0512109149) Vincenzo Catone (0512106704) Jacopo Gennaro Esposito (0512121285)



#### Revision History

Data	Versione	Descrizione	Autori
21/11/2023	0.1	Prima stesura	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
23/11/2023	0.2	Aggiunta sezione introduttiva e dei design goals	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
26/11/2023	0.3	Aggiunta Trade Off e architettura del sistema corrente/proposto	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
30/11/2023	0.4	Inserimento diagrammi delle componenti e architetturali	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
04/12/2023	0.5	Aggiunta mapping Hardware/Software e gestione dati persistenti	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
07/12/2023	0.6	Aggiunta schema ER e dizionario	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
08/12/2023	0.7	Aggiunta controllo degli accessi e condizioni limite	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
10/12/2023	0.8	Aggiunta servizi sottosistemi	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
11/12/2023	0.9	Stesura glossario	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
13/12/2023	1.0	Revisione	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito



### Team members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Alessandro Pascarella	Team Member	AP	a.pascarella9@studenti.unisa.it
Vincenzo Catone	Team Member	VC	v.catone@studenti.unisa.it
Jacopo Gennaro Esposito	Team Member	JE	j.esposito9@studenti.unisa.it



## Sommario

<b>1. Introduzione.....</b>	<b>4</b>
1.1 Scopo del sistema.....	4
1.2 Obiettivi di design (Design Goal).....	4
1.3 Definizioni, acronimi e abbreviazioni.....	6
1.4 Riferimenti.....	6
1.5 Organizzazione del documento.....	6
<b>2. Architettura del sistema corrente.....</b>	<b>7</b>
<b>3 Architettura del sistema proposto.....</b>	<b>7</b>
3.1 Panoramica.....	7
3.2 Decomposizione in sottosistemi.....	8
3.3 Mapping Hardware/Software.....	21
3.4 Gestione dei dati persistenti.....	22
3.5 Controllo degli accessi e sicurezza.....	27
3.6 Controllo globale del software.....	30
3.7 Condizioni limite.....	30
<b>4. Servizi dei sottosistemi.....</b>	<b>33</b>
<b>5. Glossario.....</b>	<b>38</b>



# 1. Introduzione

## 1.1 Scopo del sistema

**The Spoon**, con un approccio tecnologico e intuitivo, aspira a diventare la principale piattaforma per la prenotazione di tavoli e la gestione degli ordini nei ristoranti della penisola.

The Spoon consente agli utenti di effettuare comodamente ordini online, riducendo i tempi di attesa, e di prenotare un tavolo senza alcuna pressione, sia per i clienti che per i gestori dei ristoranti.

L'obiettivo principale è quello di offrire ai clienti un'esperienza intuitiva e user-friendly, mentre fornisce ai ristoratori uno strumento efficiente per la gestione delle prenotazioni e degli ordini sia d'asporto che direttamente al tavolo, infatti è possibile:

- Prenotare un nuovo tavolo;
- Ordinare direttamente a tavola ciò che si vuole mangiare;
- Ordinare da asporto ciò che si vuole e ritirarlo ad un orario prestabilito.

## 1.2 Obiettivi di design (Design Goal)

Nel seguente paragrafo si presentano i Design Goal, nei quali si procede a identificare le caratteristiche su cui il sistema deve concentrarsi, formalizzandole in modo esplicito. Ciò è essenziale perché ogni decisione importante di progettazione deve essere guidata dal medesimo insieme di criteri. I **Design Goal** individuati sono stati suddivisi nelle seguenti categorie:

- **Performance:** si definiscono i requisiti di spazio e velocità che il sistema deve rispettare;
- **Reliability:** si stabilisce l'entità dello sforzo necessario per minimizzare i crash del sistema e le loro conseguenze.
- **Maintainability:** si valuta la difficoltà associata alle modifiche al sistema dopo il suo rilascio.
- **Dependability:** Determina lo sforzo speso per minimizzare i fallimenti del sistema (crash e falle di sicurezza) e le loro conseguenze.
- **Final user:** includono qualità desiderabili dal punto di vista dell'utente, ma che non sono state contemplate dai criteri di prestazioni e affidabilità.

Ogni Design Goal è stato organizzato secondo:

- **Rank:** viene specificata la priorità secondo un valore compreso tra 1 (massima) e 16 (minima);
- **ID Design Goal:** identificativo e nome assegnato al Design Goal;
- **Descrizione:** descrizione del Design Goal;
- **Categoria:** categoria del Design Goal;
- **FNF di origine:** il requisito non funzionale d'origine, che ha generato il Design Goal.



Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
2	DG_1 Tempi di risposta	Il sistema deve garantire dei tempi di risposta inferiori a 10 secondi.	Performance	
7	DG_2 Manutenibilità	Il sistema deve essere in grado di garantire una facile manutenibilità ed estensione.	Maintainability	
6	DG_3 Estendibilità	Il sistema deve garantire l'aggiunta di nuove funzionalità, essendo che il sistema potrebbe prevedere nuove caratteristiche sia per gli utenti che per i ristoratori.	Maintainability	
8	DG_4 Interfaccia minimale	Il sistema deve prevedere un'interfaccia intuitiva, affinché l'utente possa usufruire di tutte le funzionalità che la piattaforma offre.	Final User	
1	DG_5 Trattamento dei dati	Il sistema deve garantire la massima protezione dei dati sensibili degli utenti, procedendo al salvataggio in maniera protetta e sicura.	Reliability	
4	DG_6 Attività del sistema	Il sistema deve garantire un'attività continua di tutte le sue funzioni	Reliability.	
5	DG_7 Sicurezza	Il sistema deve garantire la sicurezza dei dati conservati, utilizzando protocolli di comunicazione sicuri, proteggendo i dati salvati in modo persistente, e permettendone la visualizzazione solo ai legittimi proprietari	Dependability	



## Trade Off

Trade-Off	Descrizione
<b><u>Tempi di risposta vs Memoria/Costi</u></b>	Per garantire tempi di risposta più veloci, entro un massimo di 10 secondi, il sistema richiederà l'utilizzo di memorie capienti e veloci, comportando un aumento dei costi.
<b><u>Tempi di rilascio vs Staffing</u></b>	Un eventuale aumento delle risorse assegnate al progetto in una fase di sviluppo avanzata potrebbe risultare in una diminuzione della produttività, causando conseguentemente un prolungamento dei tempi di rilascio. Pertanto, si preferisce, una accurata pianificazione iniziale delle risorse necessarie.

## 1.3 Definizioni, acronimi e abbreviazioni

Vengono riportati di seguito alcune definizioni presenti in questo documento:

- **Sottosistema:** sottoinsieme di servizi del dominio applicativo, composto unicamente da servizi caratterizzati da una relazione funzionale.
- **Design Goal:** qualità sulle quali il sistema si deve focalizzare
- **Dati persistenti:** insieme di dati che permangono all'esecuzione del programma che li ha generati e che pertanto vengono salvati
- **Mapping Hardware/Software:** analisi della connessione fra le parti fisiche e logiche che compongono il sistema.
- **SDD:** System Design Document
- **RAD:** Requirements Analysis Document

## 1.4 Riferimenti

Di seguito una lista dei documenti utili durante la lettura:

- [Statement of Work](#)
- [Requirements Analysis Document](#)
- [System Design Document](#)

## 1.5 Organizzazione del documento

Il presente documento di System Design è composto da quattro sezioni:

- **Introduzione:** Si descrive in generale lo scopo del sistema e gli obiettivi di design che il sistema si propone di raggiungere.
- **Architettura Software corrente:** Si descrive lo stato corrente dell'architettura software presente



- **Architettura Software proposta:** Si descrive come il sistema software verrà definito e partizionato in sottosistemi, il mapping Hardware/Software, la gestione dei dati persistenti. Vengono poi descritte le strutture dei singoli sottosistemi e le boundary conditions dell'intero sistema.
- **Glossario:** Lista dei termini usati nel documento con annessa spiegazione.

## 2. Architettura del sistema corrente

Risultano esserci attualmente delle piattaforme molto simili a The spoon, che offrono semplicemente la possibilità della prenotazione ai tavoli e di tenere traccia dei ristoranti disponibili sulla piattaforma, oltre a prevedere la possibilità di aggiunta manuale di eventuali ristoranti all'interno dell'archivio di sistema. Le principali app come The fork, Restopolis o Yelp sono definite come applicazioni mobili e web-based. The spoon si pone unicamente come piattaforma web-based, ma differenza delle app nominate precedentemente, The spoon offre la possibilità oltre che di poter prenotare ai tavoli, l'opportunità di poter ordinare, sia d'asporto che direttamente al tavolo, favorendo una maggiore facilità d'uso sia per il lato cliente che per il lato ristoratore. Invece i sistemi attuali di ordinazione non prevedono alcuna forma di integrazione del POS, nonostante così facendo sia possibile snellire e rendere più efficiente il processo stesso di ordinazione.

## 3 Architettura del sistema proposto

### 3.1 Panoramica

The Spoon mira a semplificare la prenotazione e la gestione degli ordini nei ristoranti in Italia. Il suo obiettivo principale è migliorare l'esperienza sia dei clienti che dei ristoratori, riducendo le attese e semplificando i processi. Utilizza notifiche tramite app di messaggistica come Telegram per migliorare la comunicazione tra clienti e ristoratori, consentendo una gestione efficiente delle prenotazioni e degli ordini. Inoltre, offre un'integrazione con il punto vendita (POS) dei ristoranti per ottimizzare le operazioni. Gli utenti possono prenotare tavoli, visualizzare i menù e effettuare ordini direttamente attraverso la piattaforma, inoltre fornisce ai ristoratori un sistema di controllo, consentendo loro di tenere traccia delle prenotazioni e del numero di clienti attesi. The Spoon invia notifiche ai clienti e ai ristoratori per confermare, modificare o annullare le prenotazioni e gli ordini. L'obiettivo è migliorare l'efficienza operativa e rendere l'esperienza più piacevole per tutti i partecipanti.

Per usufruire dei servizi offerti da The Spoon per quanto riguarda la prenotazione dei tavoli non bisogna essere registrati in quanto al momento della prenotazione vanno inseriti i dati per confermare quest'ultima. Per quanto riguarda il lato ristoratore invece c'è bisogno di registrarsi alla piattaforma, per garantire la riservatezza e la sicurezza della gestione del ristorante per la gestione degli ordini e della prenotazione. Il ristoratore rispetto a un utente non registrato può:

- Aggiungere i suoi ristoranti con relativi menù e tavoli disponibili
- Gestione degli ordini ricevuti





Il nostro sistema sarà basato sullo stile architetturale **Model View Controller** misto al **Three Tier**, è verrà implementato mediante Spring MVC.

La motivazione dietro questa scelta è che tale architettura è particolarmente adatta per lo sviluppo di web application quali il nostro sistema, visto che la separazione fra la logica di presentazione e quella di elaborazione, porta una serie di miglioramenti:

- Manutenibilità
- Riutilizzabilità
- Scalabilità
- Leggibilità

Nello sviluppare il sistema verranno utilizzati **HTML5, CSS3 e Javascript** per la parte di front-end.

Per la logica di business ovvero il back-end verrà utilizzato **Java SPRING**.

Per gestire il database verranno usati:

- **Storage JPA** per connettersi al DB
- **MySQL 8.0.35** di Oracle Corporation come database.

Il sistema verrà istanziato sulla **piattaforma cloud** di Oracle.

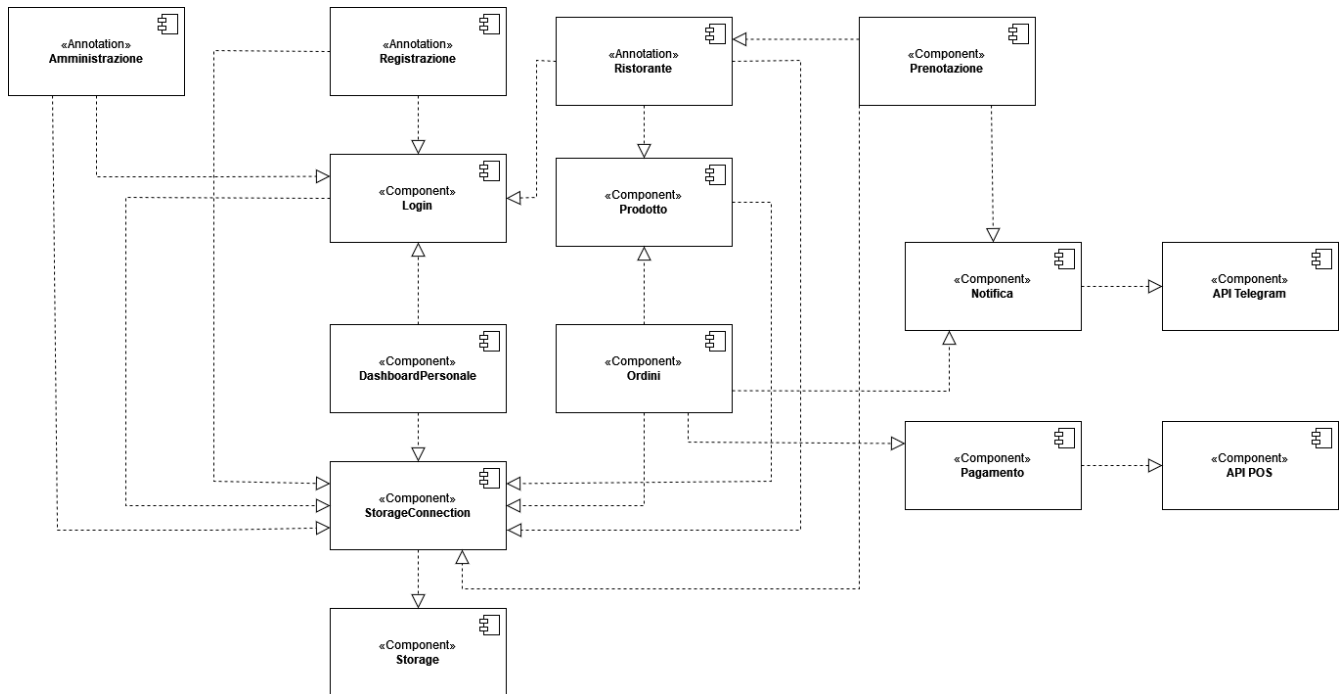
### 3.2 Decomposizione in sottosistemi

I sottosistemi individuati sono i seguenti:

Registrazione: Si occupa di definire la registrazione del ristorante

- **Login**: Si occupa di gestire l'accesso alla dashboard del ristorante/admin
- **Dashboard**: definisce la disconnessione dalla dashboard ristorante, la visualizzazione della dashboard e la modifica dei dati
- **Ordini**: Si occupa della gestione degli ordini, ricezione degli ordini, visualizzazione degli ordini, aggiunta e rimozione dei prodotti degli ordini
- **Prenotazioni**: Si occupa della gestione delle prenotazioni, ricezione delle prenotazioni, visualizzazione degli ordini, annullamento o modifica della prenotazione.
- **Ristorante**: Si occupa della funzionalità di creazione di un menù del ristorante, visualizzazione dei prodotti, aggiunta o rimozione dei prodotti al menu
- **Prodotto**: Si occupa della gestione delle informazioni del singolo prodotto, come la loro visualizzazione, l'aggiunta di nuovi prodotti e l'eventuale modifica degli stessi
- **Storage Connection**: Si va a frapponere fra i sottosistemi e il sottosistema di storage
- **Notifica**: Si occupa della gestione delle notifiche relative agli ordini e le prenotazioni (Si dovrà interfacciare con le API di Telegram)
- **Storage**: Si occupa della gestione di tutti i dati persistenti mediante un Database
- **Pagamenti**: Si occupa della gestione dei pagamenti degli ordini (Dovrà interfacciarsi con le API dei POS)
- **Amministrazione**: Gestisce le operazioni amministrative della piattaforma

Di seguito vengono mostrate le dipendenze fra i sottosistemi mediante un component diagram UML.



Alcuni dei sottosistemi faranno uso di componenti esterni, a seguire un elenco:

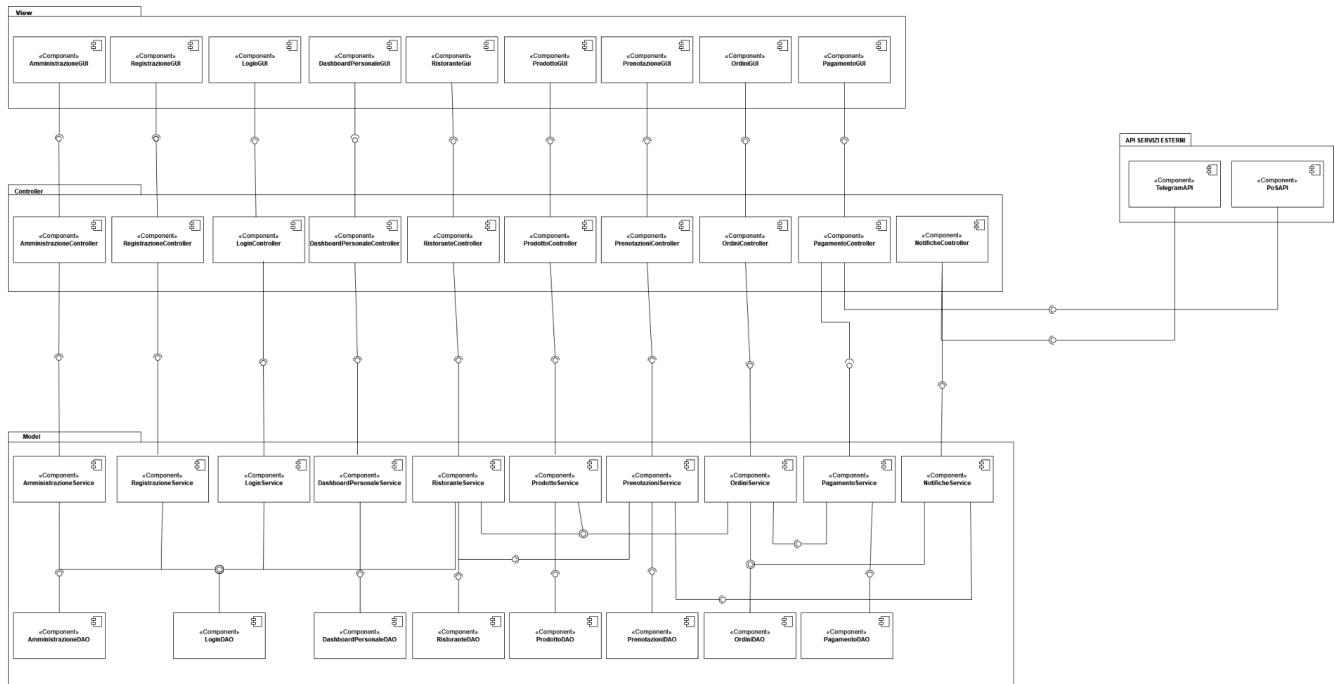
- **Pagamento**, che fa uso delle API POS
- **Notifica**, che utilizzerà le API di Telegram

In seguito una rappresentazione dettagliata di ogni sottosistema, andando ad evidenziare le componenti principali:

- **GUI**: Graphic User Interface, gestisce le varie view che devono essere renderizzate per creare le pagine view da mostrare al client.
- **Controller**: gestisce la logica per il controllo del sistema
- **Model**: si occupa della logica di business
- **DAO**: Data Access Object, si occupa di fornire accesso ai dati persistenti



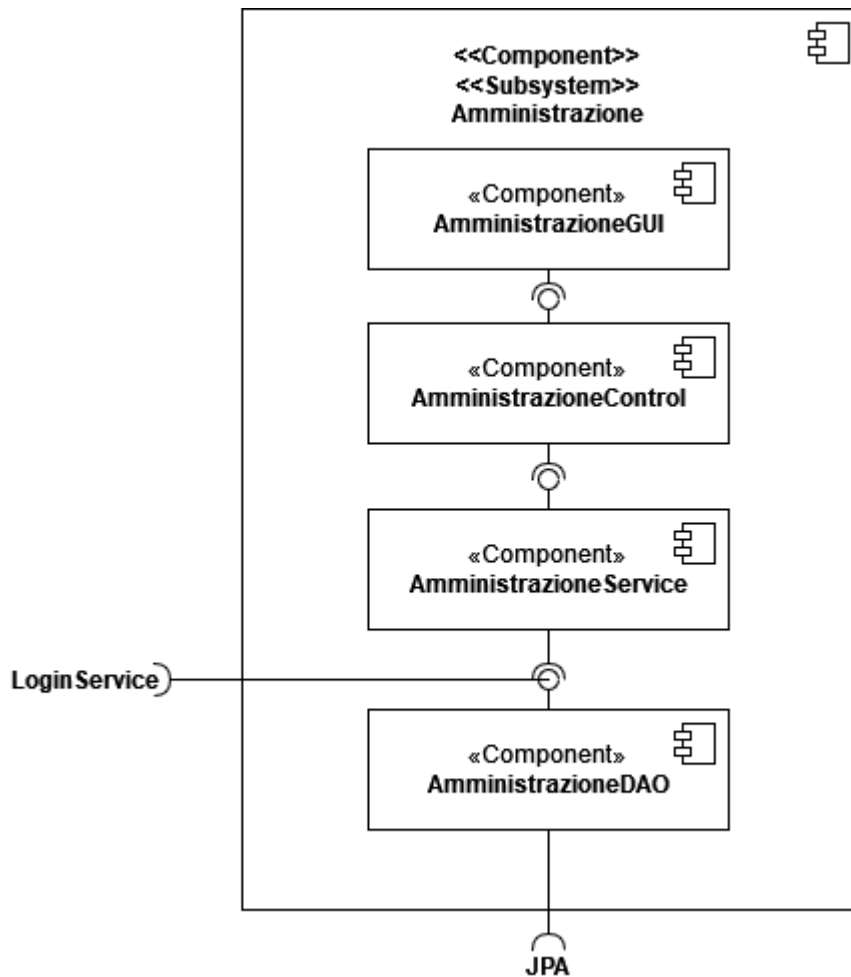
## Diagramma architetturale



[Link immagine ad alta risoluzione](#)

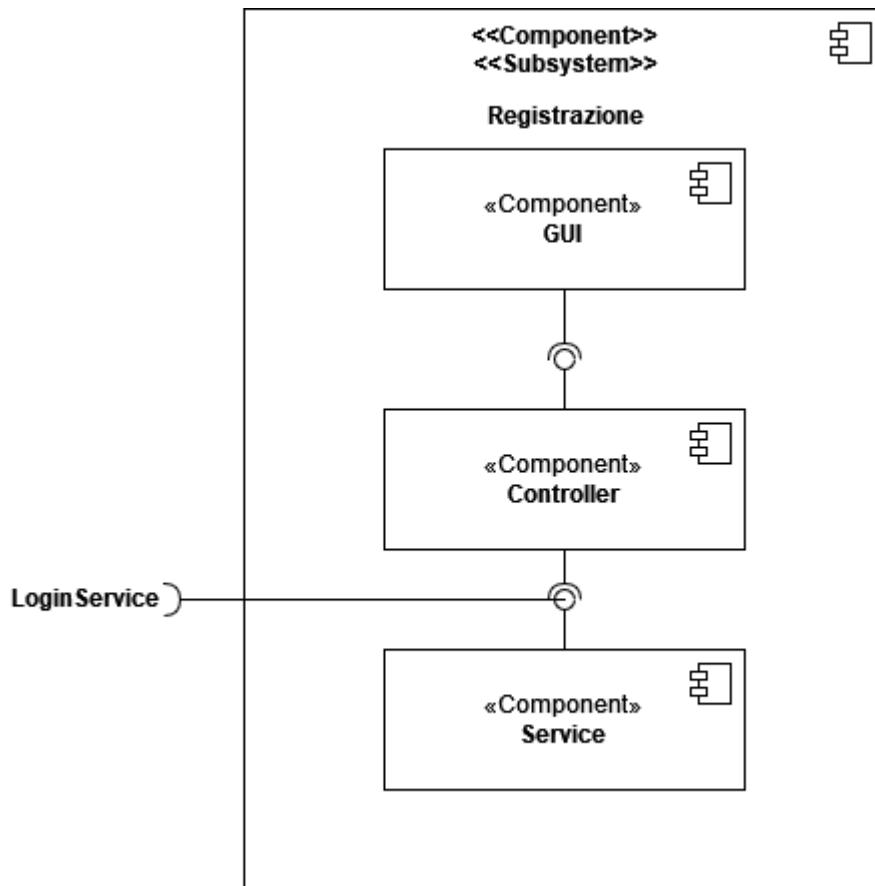


### Sottosistema amministrazione



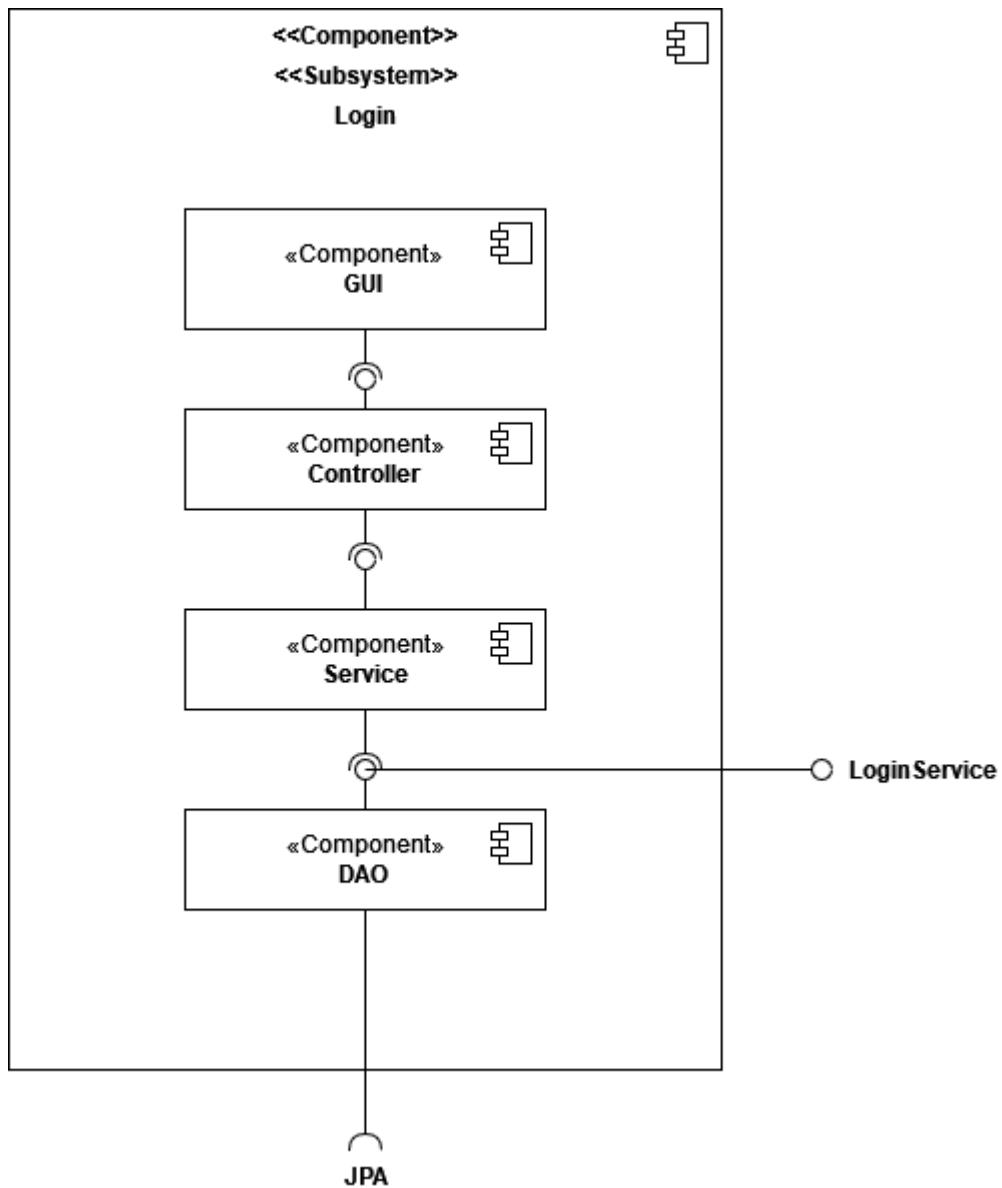


### Sottosistema Registrazione



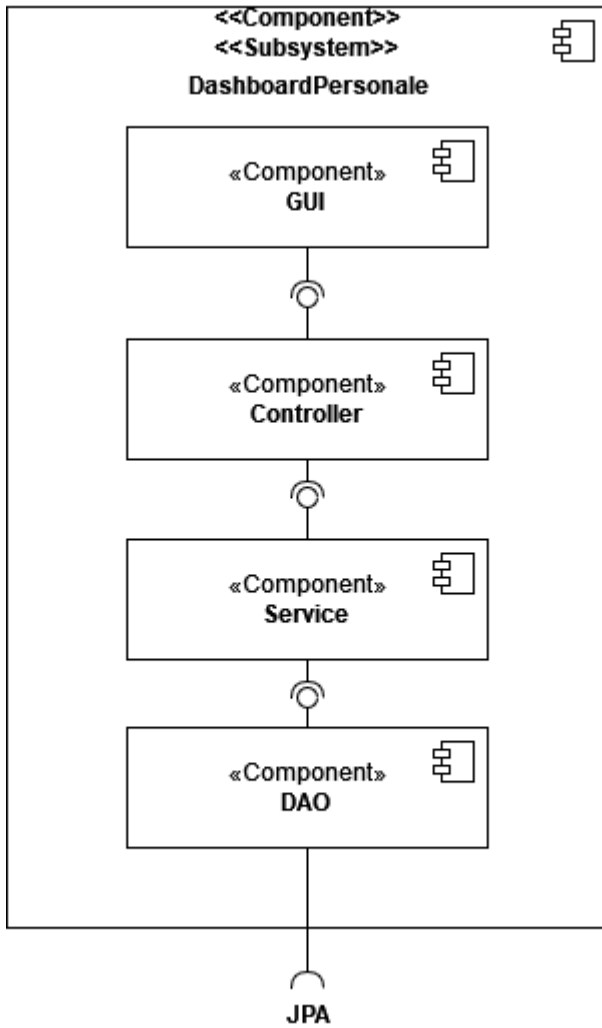


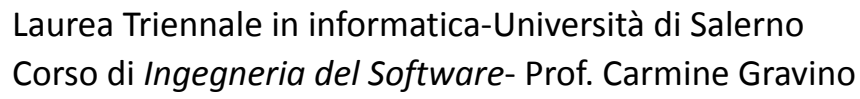
### Sottosistema Login



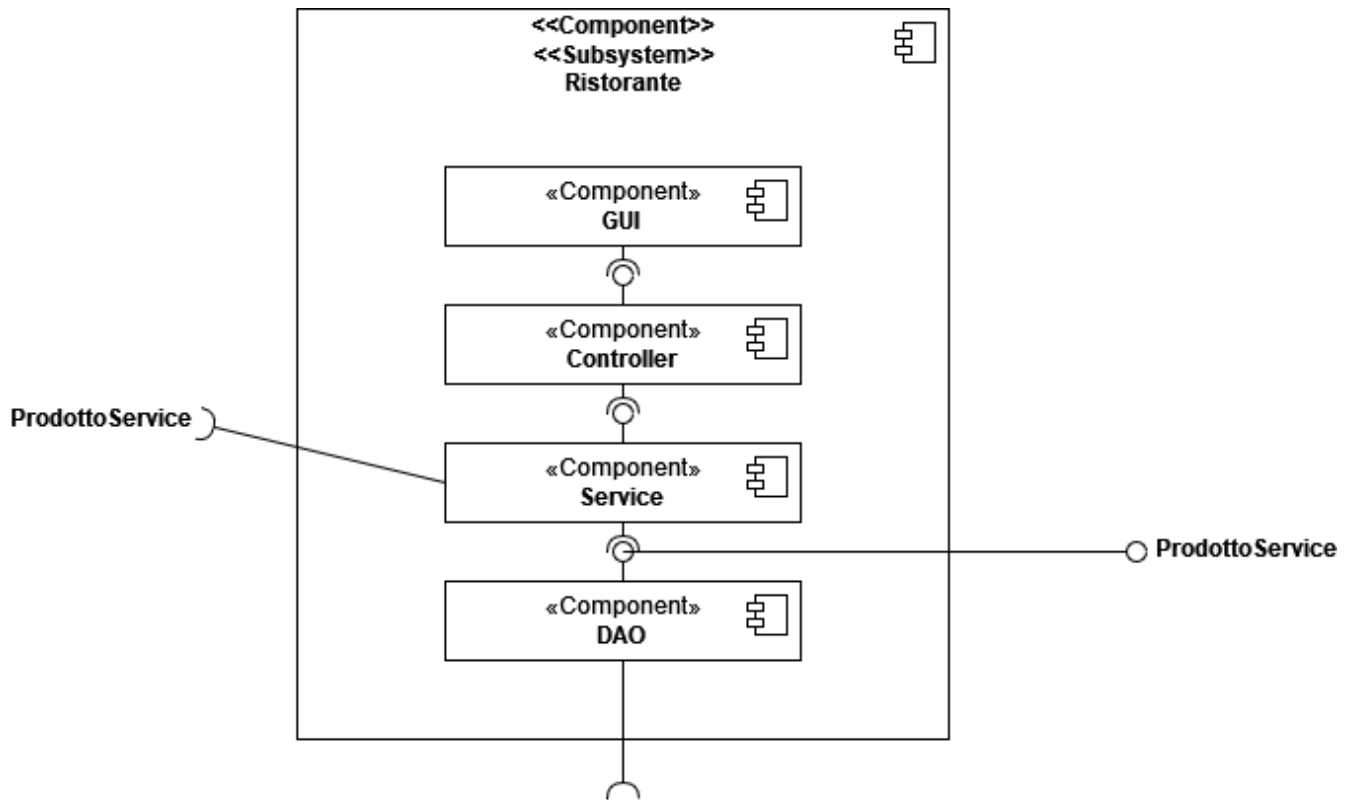


### Sottosistema DashboardPersonale





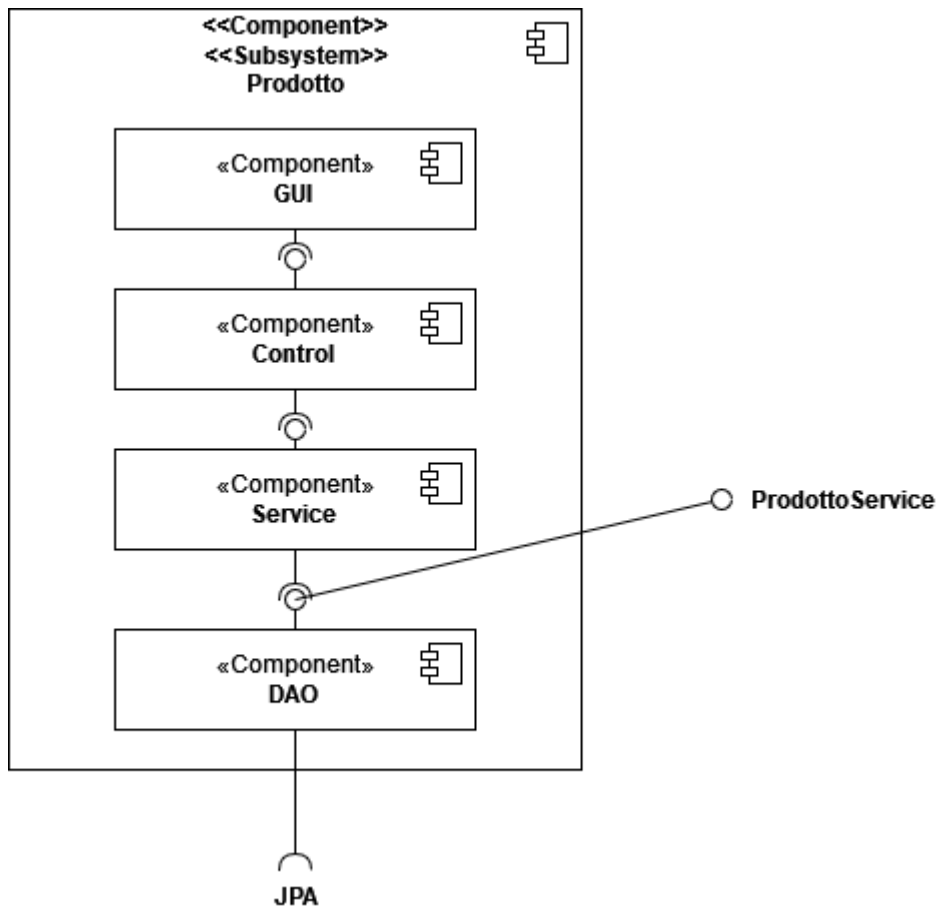
## Sottosistema Ristorante





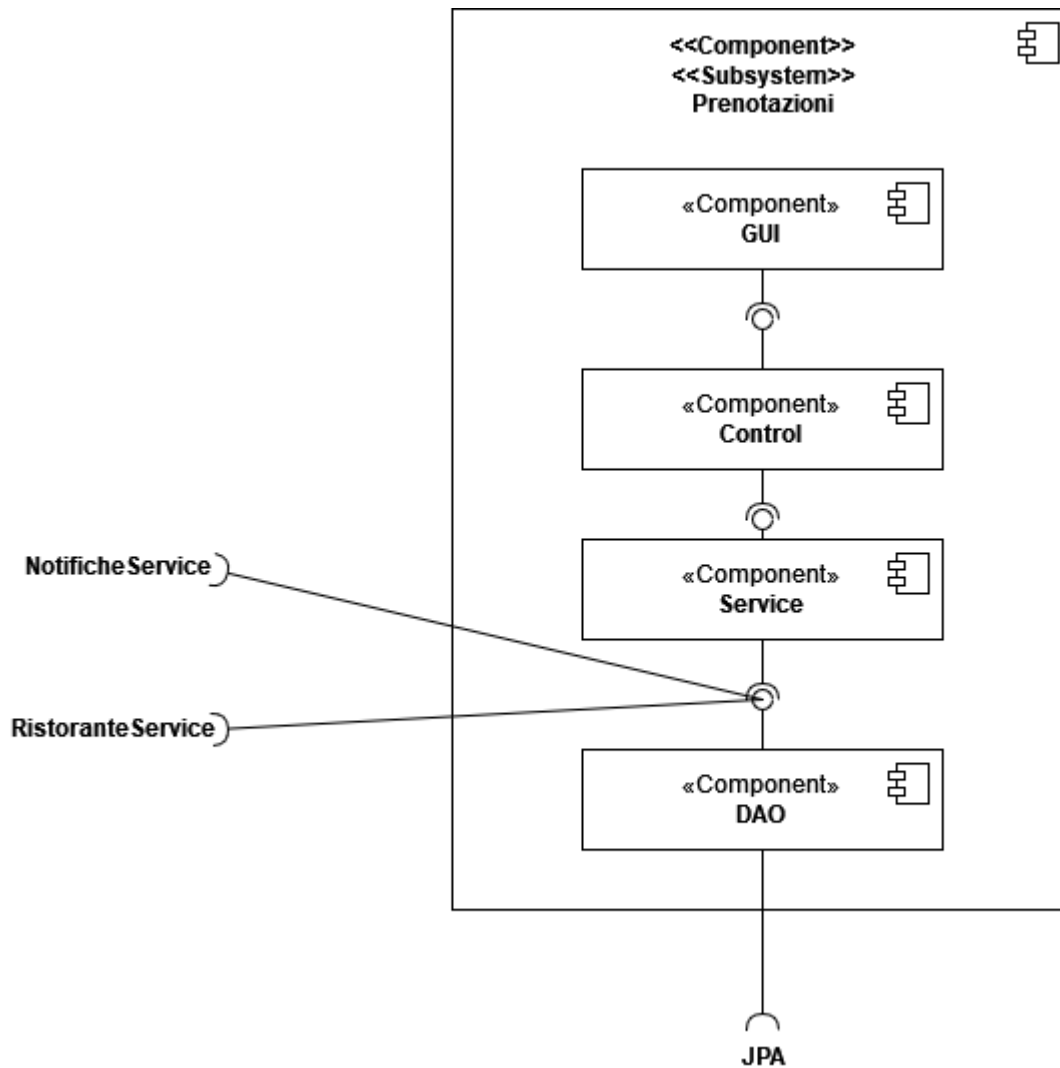


### Sottosistema Prodotto



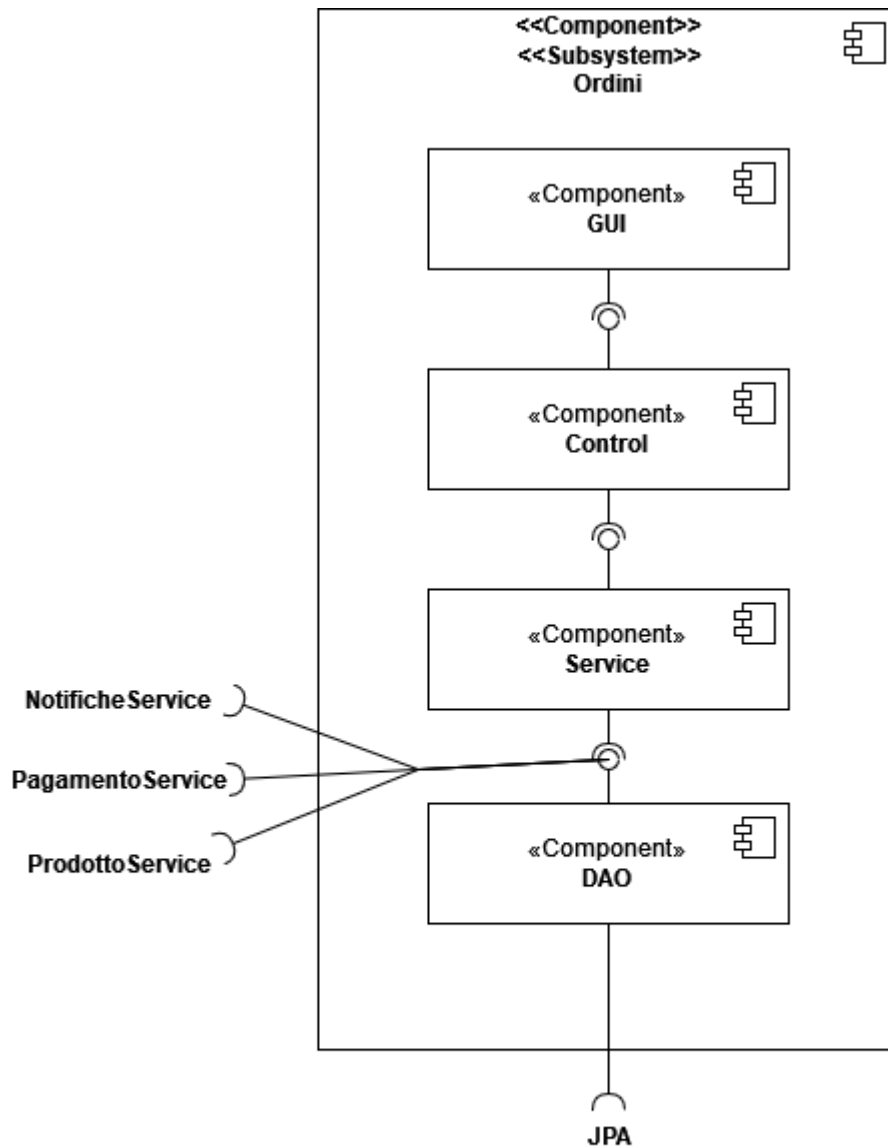


### Sottosistema Prenotazioni



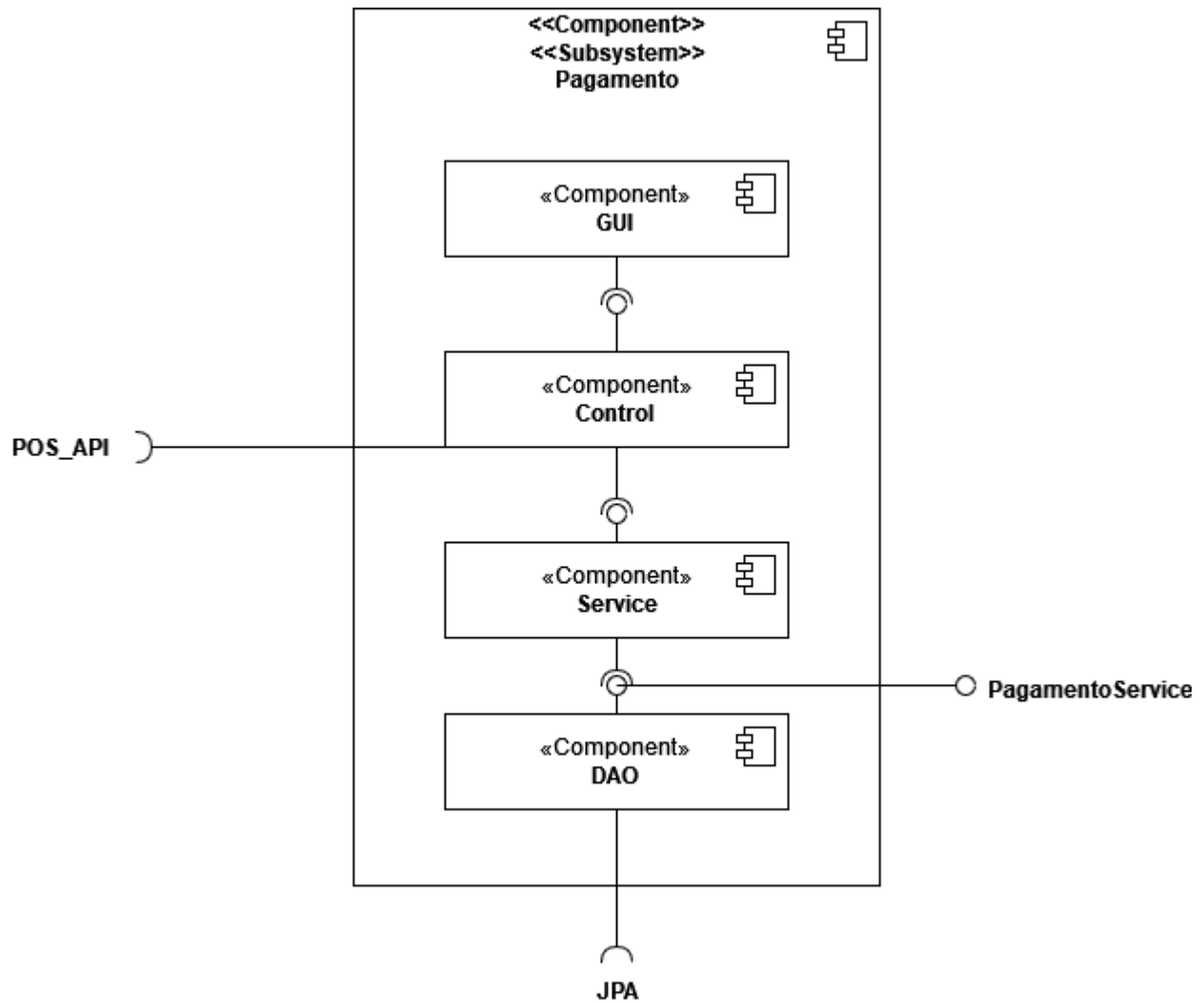


### Sottosistema Ordini



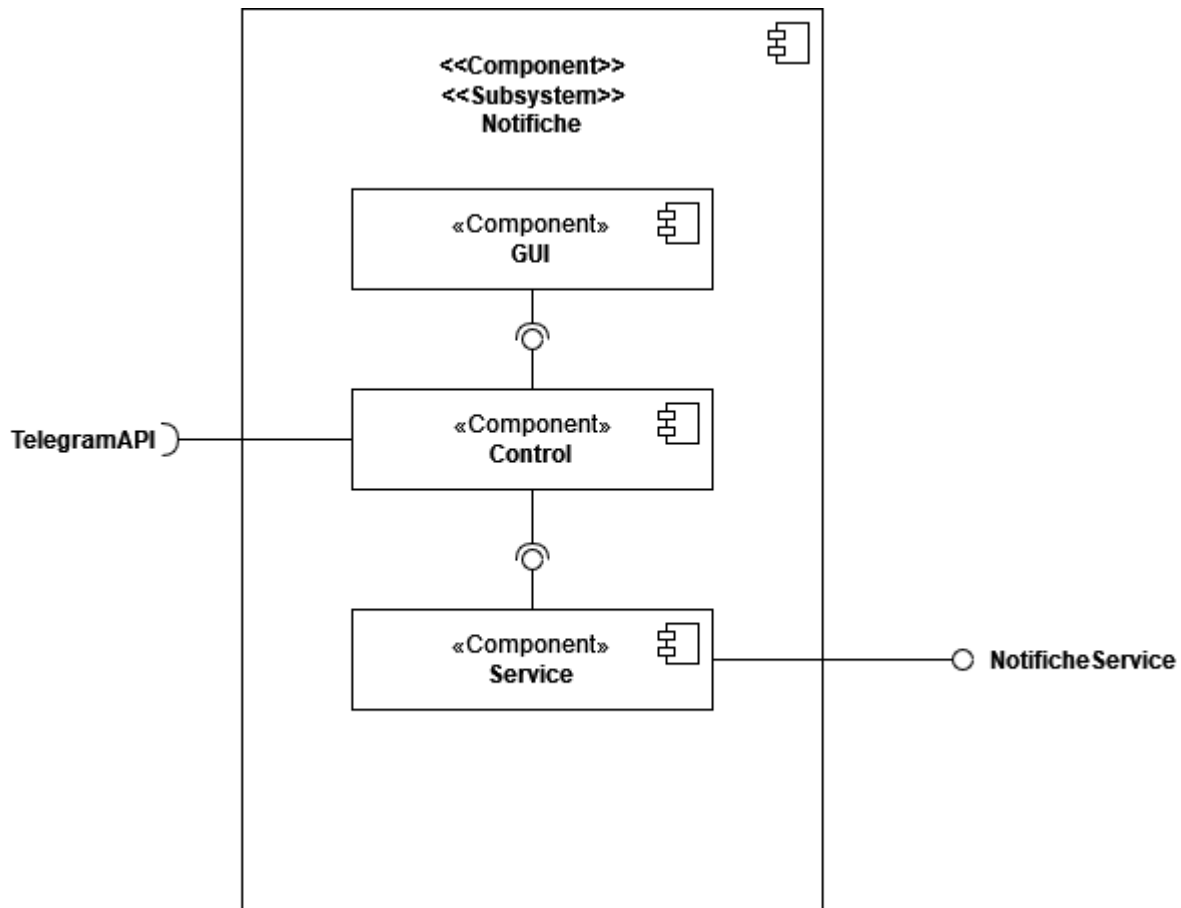


### Sottosistema Pagamento





### Sottosistema Notifica

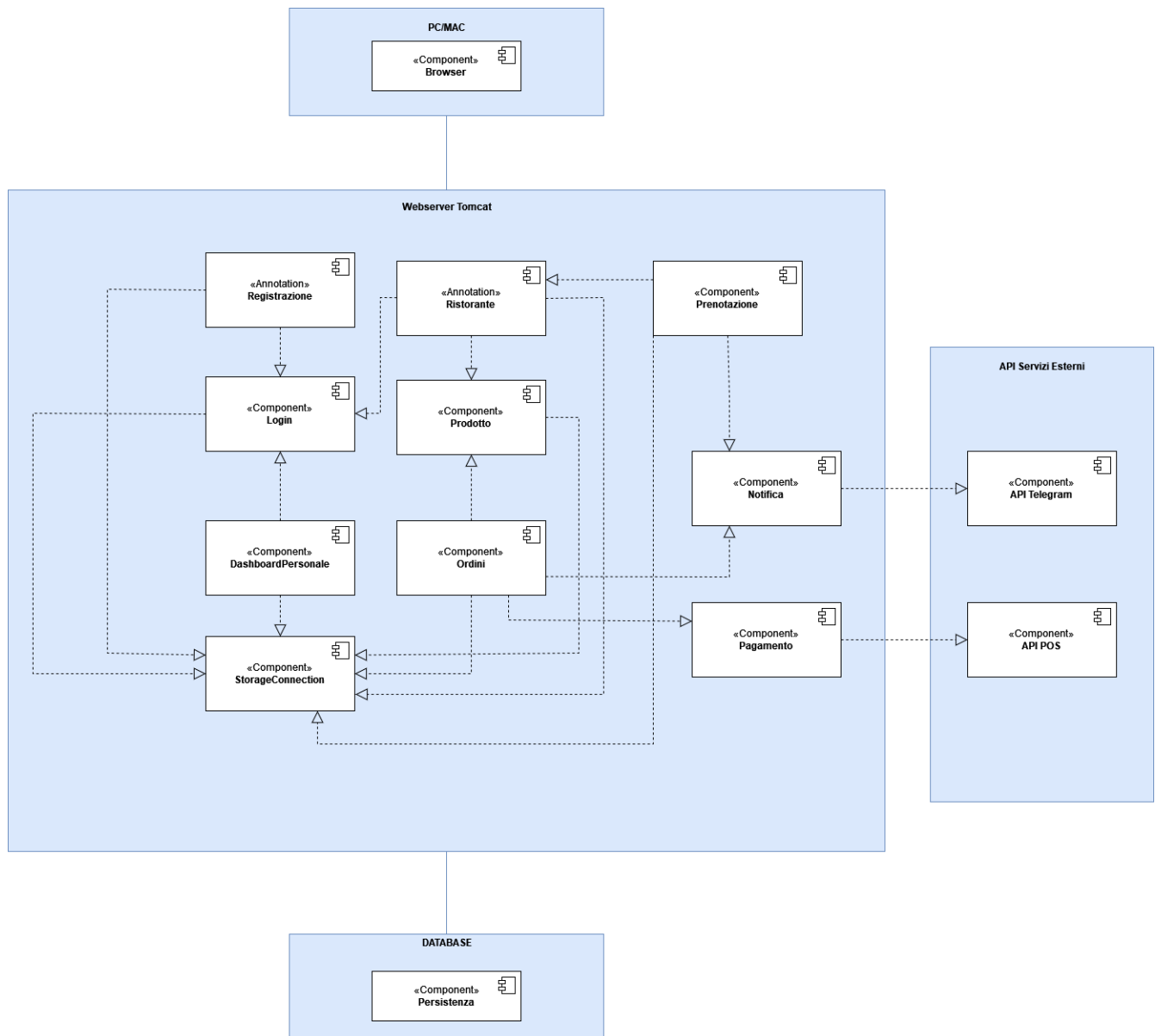




### 3.3 Mapping Hardware/Software

L'applicazione web da implementare si baserà su una piattaforma hardware composta da un **server** che risponde alle richieste effettuate dai **client**, ovvero un qualsiasi device dotato di un Browser e una connessione ad Internet.

A seguire un UML Deployment Diagram che descrive il mapping Hardware/Software





### 3.4 Gestione dei dati persistenti

#### Introduzione

Per gestire il salvataggio dei dati persistenti del nostro sistema software abbiamo deciso di affidarci ad un database relazionale, in modo tale da gestire facilmente l'accesso concorrente ai dati e garantire in contemporanea la consistenza dei dati mediante l'utilizzo di un DBMS.

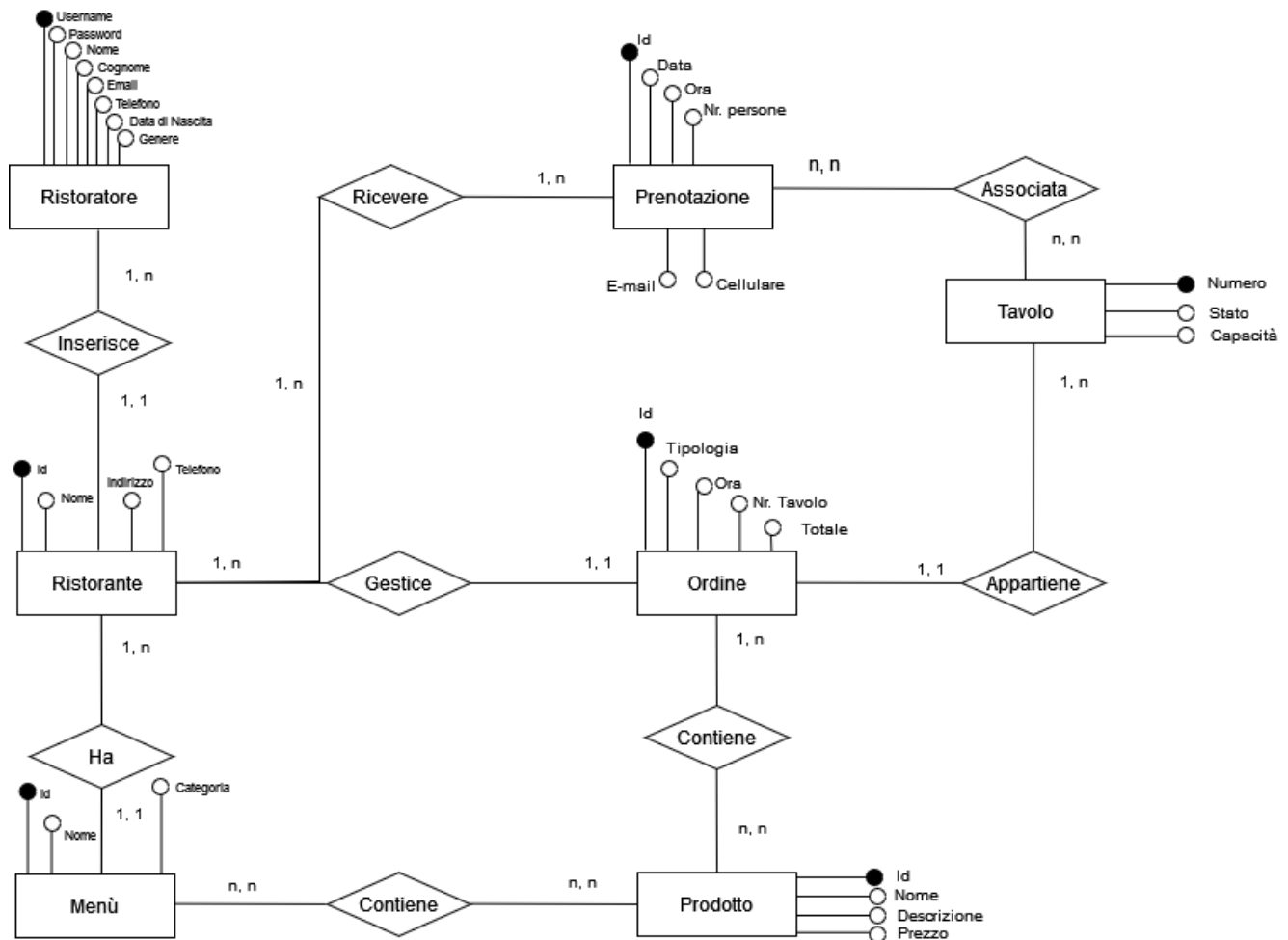
Difatti un DBMS ci permette di rimanere fedeli il più possibile ai design goal, potendo fare affidamento su:

- **Imposizioni di vincoli di integrità sui dati**, in quanto un DBMS dà la possibilità di definire diverse tipologie di vincoli al fine di mantenere l'integrità dei dati e verifica che tutti i vincoli siano soddisfatti quando la base di dati viene modificata.
- **Privatezza dei dati**, un DBMS garantisce un accesso protetto ai dati, potendo definire diversi Utenti con accesso a diverse zone della basi di dati e potendo scegliere che tipologia di operazioni possono effettuare.
- **Affidabilità dei dati**, un DBMS offre diversi metodi per salvare copie dei dati ed effettuare un ripristino dello stato della base di dati in caso di malfunzionamenti hardware e/o software.
- **Atomicità delle operazioni**, un DBMS consente di eseguire una serie di operazioni in maniera atomica. Dunque o si esegue l'intera sequenza di operazioni con successo o nessuna di queste comporterà una modifica dei dati della base. L'atomicità delle operazioni consente di mantenere uno stato consistente con la realtà modellata.

In particolare abbiamo deciso di affidarci al database messo a disposizione da **Oracle Corporation**. Il database in questione è **MySQL 8.0.35**.



## ER: Scheme ER del database



Ristoratore (**IdRistoratore**, Password, Nome, Cognome, Email, Telefono, Data di Nascita, Genere);

Ristorante (**IdRistorante**, Nome, Indirizzo, Telefono);

Prenotazione (**IdPrenotazione**, Data, Ora, Nr. Persone, E-mail, Cellulare);

Tavolo (**NumeroTavolo**, Stato, Capacità);

Ordine (**IdOrdine**, Tipologia, Ora, Nr. Tavolo, Totale,);

Menù (**IdMenu**, Nome, Categoria);

Prodotto (**IdProdotto**, Nome, Descrizione, Prezzo).





## Dizionario dei dati

Di seguito verranno mostrati gli attributi e le entità individuate nello schema ER.

Nome entità		Ristoratore	
Descrizione		Contiene i dati relativi ad un utente che utilizza la piattaforma	
Nome Campo	Tipo	Vincoli di chiave	Altri vincoli
IdRistoratore	int	PRIMARY KEY	NOT NULL AUTO_INCREMENT
Password	varchar(30)		NOT NULL
Nome	varchar(20)		NOT NULL
Cognome	varchar(20)		NOT NULL
Email	varchar(20)		NOT NULL
Telefono	varchar(11)		NOT NULL
Data di Nascita	date		NOT NULL
Genere	char(1)		

Nome entità		Ristorante	
Descrizione		Contiene i dati relativi ad un ristorante presente sulla piattaforma	
Nome Campo	Tipo	Vincoli di chiave	Altri vincoli
IdRistorante	int(10)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
Nome	varchar(50)		NOT NULL
N.Civico	varchar(3)		NOT NULL
CAP	int(5)		NOT NULL
Via	varchar(100)		NOT NULL
Provincia	varchar(2)		NOT NULL
Telefono	varchar(11)		NOT NULL



Nome entità		Prenotazione	
Descrizione		Contiene i dati relativi ad una prenotazione effettuata tramite la piattaforma	
Nome Campo	Tipo	Vincoli di chiave	Altri vincoli
IdPrenotazione	int(10)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
Data	date		NOT NULL
Ora	time		NOT NULL
Nr. persone	int(2)		NOT NULL
E-mail	varchar(100)		NOT NULL
Cellulare	int(11)		NOT NULL

Nome entità		Tavolo	
Descrizione		Contiene i dati relativi ai tavoli presenti nel ristorante	
Nome Campo	Tipo	Vincoli di chiave	Altri vincoli
NumeroTavolo	int(2)	PRIMARY KEY	NOT NULL
Stato	boolean		NOT NULL
Capacità	int(2)		NOT NULL

Nome entità		PrenotazioneTavolo	
Descrizione		Contiene i dati relativi ai tavoli presenti nel ristorante	
Nome Campo	Tipo	Vincoli di chiave	Altri vincoli
NumeroTavolo	int(2)	foreign key(NumeroTavolo) references Tavolo	NOT NULL



IdPrenotazione	int(10)	foreign key(IdPrenotazione) references Prenotazioe	NOT NULL
----------------	---------	---	----------

Nome entità		Ordine	
Descrizione		Contiene i dati relativi ad un ordine effettuato tramite la piattaforma	
Nome Campo	Tipo	Vincoli di chiave	Altri vincoli
IdOrdine	int(10)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
Tipologia	boolean		NOT NULL
Ora	time		
Nr. tavolo	int(2)		
Totale	decimal(10,2)		NOT NULL

Nome entità		Menù	
Descrizione		Contiene i dati relativi al menù del ristorante	
Nome Campo	Tipo	Vincoli di chiave	Altri vincoli
IdMenu	int(10)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
Nome	char(30)		NOT NULL
Categoria	char(30)		NOT NULL

Nome entità		MenuProdotto	
Descrizione		Contiene i prodotti del menù all'interno del ristorante	
Nome Campo	Tipo	Vincoli di chiave	Altri vincoli
IdProdotto	int	foreign key(Id_prodotto) references prodotto	NOT NULL



IdMenu	int	foreign key(Id_menu) references menu	NOT NULL
--------	-----	---	----------

Nome entità		Prodotto	
Descrizione		Contiene i dati relativi ad un prodotto presente nel menù del ristorante	
Nome Campo	Tipo	Vincoli di chiave	Altri vincoli
IdProdotto	int(10)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
Nome	char(50)		NOT NULL
Descrizione	varchar(2000)		
Prezzo	Decimal(10, 2)		NOT NULL

### 3.5 Controllo degli accessi e sicurezza

Nella seguente sezione si riporta la matrice degli accessi per poter tracciare degli attori, che possono accedere ai servizi offerti dal sistema.

Attori o Oggetti	Ristoratore
Registrazione	
Login	Login
Area Personale	Logout VisualizzaAreaPersonale ModificaAreaPersonale CancellazioneAccount
Ristorante	CreaRistorante VisualizzaRistorante ModificaRistorante



	EliminaRistorante InserisciRistorante
Prenotazione	VisualizzaPrenotazione ConfermaPrenotazione RifiutaPrenotazione
Tavolo	VisualizzaNumero VisualizzaStato ModificaStato VisualizzaCapacità ModificaCapacità
Ordine	VisualizzaOrdine ConfermaOrdine RifiutaOrdine ModificaOrdine
Menù	CreaMenu InserisciMenu ModificaMenu EliminaMenu
Prodotto	CreaProdotto InserisciProdotto ModificaProdotto EliminaProdotto VisualizzaProdotto



Attori o Oggetti	Amministratore	Ospite
Registrazione		RegistrazioneRistoratore
Login	Login	
Area Personale	Logout VisualizzaAreaPersonale ModificaAreaPersonale CancellazioneAccount	
Ristorante	CreaRistorante VisualizzaRistorante ModificaRistorante EliminaRistorante InserisciRistorante	VisualizzaRistorante
Prenotazione	VisualizzaPrenotazione ModificaPrenotazione ConfermaPrenotazione RifiutaPrenotazione	VisualizzaPrenotazione ModificaPrenotazione
Tavolo	VisualizzaNumero VisualizzaStato ModificaStato VisualizzaCapacità ModificaCapacità	
Ordine	VisualizzaOrdine ConfermaOrdine RifiutaOrdine ModificaOrdine	VisualizzaOrdine ModificaOrdine
Menù	CreaMenu InserisciMenu ModificaMenu EliminaMenu	



Prodotto	CreaProdotto InserisciProdotto ModificaProdotto EliminaProdotto VisualizzaProdotto	
----------	--	--

### 3.6 Controllo globale del software

The Spoon è un servizio dinamico ed interattivo, dove il ristoratore si interfaccia con la piattaforma e interagisce col sistema tramite l'utilizzo di interfacce grafiche. A seconda dell'azione utilizzata dall'utente, viene scelto il relativo controllo e la giusta gestione dell'evento.

The Spoon utilizza un controllo di flusso di tipo event-driven, essendo che viene implementato secondo una web-application.

### 3.7 Condizioni limite

Identificativo	UCBC_1 - Avvio del sistema	Data: 28/11/2023 Versione Autore: Alessandro Pascarella
Descrizione	Lo UC consente l'avvio del sistema	
Attore principale	Amministratore	
Attori secondari	NA	
Entry condition	L'Amministratore accede al server	
Exit condition On success	Il sistema viene avviato correttamente	
Exit condition On failure	Il sistema non viene avviato correttamente	
Flusso di eventi principali		
1	Amministratore	L'amministratore procede ad avviare il sistema mediante l'apposito comando



2	Sistema	Il sistema risponde alla richiesta, andando a rendere disponibili le funzionalità e tutti i servizi, dopo che le componenti necessarie siano state controllate
I flusso di Eventi Alternativo: le componenti necessarie non funzionano adeguatamente		
2.a1	Sistema	Il Sistema notifica all'amministratore l'impossibilità di avviare adeguatamente il sistema per una problematica relativamente alle componenti
2.a2	Amministratore	L'amministratore procede a correggere le problematiche notificate
2.a3	Amministratore	L'amministratore procede ad avviare il sistema
Scenario/Flusso di eventi di errore: il sistema non riesce a far funzionare correttamente le componenti necessarie		
2	Sistema	Il sistema comunica all'amministratore con una notifica l'errore.

Identificativo	UCDC_2 - Fallimento del sistema	Data: 30/11/2023 Versione Autore: Alessandro Pascarella
Descrizione	L'UC definisce il comportamento del Sistema quando si verifica un fallimento	
Attore principale	Amministratore	
Attori secondari	NA	
Entry condition	Il sistema arresta improvvisamente le proprie attività	
Exit condition On success	Il sistema viene riavviato correttamente	
Exit condition On failure	Il sistema non viene riavviato correttamente	
Flusso di eventi principali		
1	Amministratore	Include UCBC_1-Avvio del sistema





Identificativo	UCDC_3 - Errore di accesso ai dati persistenti	Data: 30/11/2023 Versione Autore: Alessandro Pascarella
Descrizione	L'UC definisce il comportamento del sistema quando non i dati persistenti risultano essere corrotti o non riesce ad accedere ai dati peristenti	
Attore principale	Amministratore	
Attori secondari	NA	
Entry condition	Il sistema non riesce ad accedere ai dati persistenti OR I dati persistenti sono corrotti	
Exit condition On success	Il sistema riesce a funzionare correttamente	
Exit condition On failure	Il sistema non riesce a funzionare correttamente	
Flusso di eventi principali		
1	Sistema	Comunica all'amministratore l'impossibilità di accedere ai dati persistenti
2	Sistema	Cessa di eseguire le richieste e termina con un errore
3	Sistema	Riprende la corretta esecuzione delle attività e delle funzionalità
I flusso di eventi alternativo: le funzionalità del sistema non vengono ripristinate al riavvio		
2.1	Amministratore	L'amministratore si fa carico della risoluzione della problematica



## 4. Servizi dei sottosistemi

In questa sezione vengono descritti i servizi di ogni sottosistema precedentemente elencati.

### Sottosistema Registrazione

Servizio	Descrizione	Interfaccia
<b>Registrazione utente</b>	La seguente funzionalità permette la registrazione dell'ospite alla piattaforma permettendogli, quindi, di ottenere il ruolo di ristoratore.	Registrazione service

### Sottosistema Login

Servizio	Descrizione	Interfaccia
<b>Login ristoratore</b>	La seguente funzionalità permette al ristoratore di effettuare l'accesso alla piattaforma.	LoginService

### Sottosistema DashboardPersonale

Servizio	Descrizione	Interfaccia
<b>Logout ristoratore</b>	La seguente funzionalità permette al ristoratore di effettuare la disconnessione dell'account dalla piattaforma.	DashboardPersonale service
<b>Visualizzazione dati personali</b>	La seguente funzionalità permette al ristoratore di visualizzare i dati personali inseriti nel sistema.	DashboardPersonale service
<b>Modifica dati personali</b>	La seguente funzionalità permette al ristoratore di modificare i dati personali inseriti nel sistema.	DashboardPersonale service
<b>Eliminazione account</b>	La seguente funzionalità permette al lettore di eliminare l'account personale dal sistema.	DashboardPersonale service



### Sottosistema ristorante

Servizio	Descrizione	Interfaccia
<b>Ricerca ristorante</b>	La seguente funzionalità permette all'utente la ricerca di ristoranti tramite la barra di ricerca e relativi filtri all'interno della piattaforma.	Ristorante service
<b>Visualizza ristorante</b>	La seguente funzionalità permette all'utente di visualizzare i dettagli di un ristorante all'interno della piattaforma.	Ristorante service
<b>Crea ristorante</b>	La seguente funzionalità permette al ristoratore la creazione di un ristorante all'interno della piattaforma.	Ristorante service
<b>Modifica ristorante</b>	La seguente funzionalità permette al ristorante la modifica di un suo ristorante (se già creato) all'interno della piattaforma.	Ristorante service
<b>Elimina ristorante</b>	La seguente funzionalità permette al ristoratore l'eliminazione di un suo ristorante (se già creato) all'interno della piattaforma.	Ristorante service

### Sottosistema prodotto

Servizio	Descrizione	Interfaccia
<b>Crea prodotto</b>	La seguente funzionalità permette al ristoratore di creare un prodotto ed inserirlo all'interno del menù del proprio ristorante.	Prodotto service
<b>Visualizza prodotto</b>	La seguente funzionalità permette ad un utente di visualizzare le specifiche di un prodotto all'interno del menù di un ristorante quali possono essere per esempio gli ingredienti o gli allergeni.	Prodotto service
<b>Modifica prodotto</b>	La seguente funzionalità permette ad un ristoratore di modificare un prodotto all'interno del menù del proprio ristorante.	Prodotto service



<b>Elimina prodotto</b>	La seguente funzionalità permette ad un ristoratore di eliminare un prodotto all'interno del menù del proprio ristorante	Prodotto service
-------------------------	--	------------------

#### Sottosistema ordine

Servizio	Descrizione	Interfaccia
<b>Crea ordine</b>	La seguente funzionalità permette ad un utente di poter creare (ma non di confermare) un ordine tramite il proprio dispositivo o in alternativa tramite un terminale all'interno del ristorante	Ordine service
<b>Modifica ordine</b>	La seguente funzionalità permette ad un utente di poter modificare(se già creato) un ordine tramite il proprio dispositivo o in alternativa tramite un terminale all'interno del ristorante	Ordine service
<b>Conferma ordine</b>	La seguente funzionalità permette ad un utente di poter confermare(se già creato) un ordine tramite il proprio dispositivo o in alternativa tramite un terminale all'interno del ristorante	Ordine service
<b>Annulla ordine</b>	La seguente funzionalità permette ad un utente di poter annullare(se già creato e non ancora confermato) un ordine tramite il proprio dispositivo o in alternativa tramite un terminale all'interno del ristorante	Ordine service
<b>Visualizza ordine</b>	La seguente funzionalità permette ad un utente di poter visualizzare (se già creato) un ordine tramite il proprio dispositivo o in alternativa tramite un terminale all'interno del ristorante	Ordine service



### Sottosistema prenotazione

Servizio	Descrizione	Interfaccia
<b>Nuova prenotazione</b>	La seguente funzionalità permette ad un utente di poter richiedere una prenotazione al tavolo di un ristorante tramite il proprio dispositivo.	Prenotazione service
<b>Modifica prenotazione</b>	La seguente funzionalità permette ad un utente di poter modificare le specifiche di una nuova prenotazione al tavolo di un ristorante tramite il proprio dispositivo.	Prenotazione service
<b>Conferma prenotazione</b>	La seguente funzionalità permette ad un utente di poter confermare una prenotazione al tavolo di un ristorante tramite il proprio dispositivo.	Prenotazione service
<b>Annulla prenotazione</b>	La seguente funzionalità permette ad un utente di poter annullare una prenotazione al tavolo di un ristorante tramite il proprio dispositivo.	Prenotazione service
<b>Visualizza prenotazione</b>	La seguente funzionalità permette ad un utente di poter visualizzare tramite un codice la propria prenotazione al tavolo di un ristorante tramite il proprio dispositivo.	Prenotazione service

### Sottosistema notifica

Servizio	Descrizione	Interfaccia
<b>Notifica conferma prenotazione</b>	La seguente funzionalità permette al sistema di inviare un'email o un messaggio(tramite app di messaggistica esterna) all'utente per confermare la prenotazione da lui effettuata.	Notifica service
<b>Notifica ordine pronto</b>	La seguente funzionalità permette al sistema di inviare un'email o un messaggio(tramite app di messaggistica esterna) per notificare all'utente l'arrivo del suo ordine d'asporto precedentemente effettuato	Notifica service



### Sottosistema pagamento

Servizio	Descrizione	Interfaccia
<b>Pagamento tramite POS</b>	La seguente funzionalità permette all'utente di poter effettuare il pagamento del suo ordine tramite pos.	Pagamento service
<b>Pagamento tramite carta/paypal</b>	La seguente funzionalità permette all'utente di poter effettuare il pagamento del suo ordine tramite carta di credito o in alternativa tramite paypal.	Pagamento service



## 5. Glossario

Sigla/Termine	Descrizione
<b>TheSpoon</b>	Nome dell'applicativo da realizzare.
<b>Piattaforma</b>	Base hardware o software sulla quale si sviluppano o vengono eseguite le applicazioni.
<b>POS</b>	Un pos (dall'inglese point of sale, lett. "punto di vendita") o terminale di pagamento è un dispositivo elettronico che consente di effettuare pagamenti mediante moneta elettronica, ovvero tramite carte di credito, di debito o prepagate.
<b>API</b>	Un'interfaccia di programmazione dell'applicazione (API) è un modo attraverso il quale due o più programmi comunicano tra loro. È un tipo di interfaccia software che offre un servizio ad altri componenti software.
<b>Telegram</b>	Servizio di messaggistica istantanea e broadcasting basato su cloud ed erogato senza fini di lucro dalla società Telegram LLC.