

Università degli Studi di Salerno

Corso di Ingegneria Del Software

BEerHAPPY

System Design Document

Docente:

Prof. Andrea De Lucia

Studenti:

Giovanni Forlenza

Gaetana Galdi

Sommario

1 Introduzione

- 1.1 Scopo del sistema
- 1.2 Design goals
- 1.3 Definizioni, acronimi e abbreviazioni
- 1.4 Riferimenti
- 1.5 Panoramica

2 Architettura del sistema

- 2.1 Panoramica
- 2.2 Decomposizione del sistema
- 2.3 Suddivisione dei sottosistemi
- 2.4 Mapping hardware/software
- 2.5 Gestione dei dati persistenti
- 2.6 Tabelle del database
- 2.7 Controllo accessi e sicurezza
- 2.8 Controllo globale software
- 2.9 Boundary condition

3 Servizi dei sottosistemi

Introduzione

Il seguente documento riguarda la progettazione del sistema BEerHAAPY che fa riferimento al documento RAD per la trasformazione del modello di analisi nel modello del sistema.

In questa fase vengono definiti gli obiettivi di progettazione del sistema, il sistema viene decomposto in sottosistemi più piccoli e vengono discusse le strategie da adottare per quanto riguarda le scelte hardware e software, la gestione dei dati persistenti, il flusso di controllo globale, le politiche di controllo degli accessi e la gestione delle condizioni bonari (startup, shutdown, eccezioni).

Scopo del sistema

Il negozio fisico BEerHAPPY vuole concorrere nel mercato globale attraverso una piattaforma semplice ed intuitiva che permetta agli utenti che accedono al sito di comprare birra artigianale e che offra anche una soluzione efficace per quanto riguarda la gestione dei prodotti, degli ordini e degli utenti.

Design goals

Il sistema proposto rispetterà i seguenti Criteri di Design:

Usabilità:

- Semplicità: Il sistema offrirà una interfaccia semplice, intuitiva e minimale, permettendo all'utente di cercare ed acquistare i prodotti che più gradiscono.

Affidabilità:

- Robustezza: Il sistema sarà in grado di gestire input errati senza interrompere il funzionamento del sistema. L'utente verrà allertato mediante eventuali messaggi di errore.
- Sicurezza: Il sistema garantisce agli utenti il corretto funzionamento delle operazioni in base alla tipologia di utente che lo utilizza.

Mantenimento

- Modularità: Il sistema permette di effettuare modifiche alle funzionalità già implementate senza la necessità di modificare i sottosistemi.

Performance

- Tempo di risposta: Il sistema garantisce una rapida risposta agli input da parte degli utenti, i quali saranno gestiti nell'arco di 10 secondi. Ovviamente tale tempo può dipendere dalla velocità della connessione.
- Il sistema dovrà essere in grado di supportare fino a 1000 richieste in modo simultaneo.

Definizioni acronimi e abbreviazioni

BEerHAPPY: nome del sito

UML: Unified Modelling Language

DBMS: Database Management System

SQL: Structured Query Language

SC: Scenario

RF: Requisito Funzionale

UC: Use Case

Architettura del sistema

Panoramica

L'architettura utilizzata per la realizzazione del sistema è un'architettura ibrida client - server combinata con il pattern architetturale Model – View – Controller (MVC) che offre un approccio flessibile per la progettazione e lo sviluppo.

L'architettura client – server è un modello in cui le responsabilità dell'applicazione sono suddivise tra il client (che è l'interfaccia utente) e il server (che gestisce la logica di business e l'accesso ai dati).

Il pattern MVC viene usato per organizzare il codice e separare le diverse responsabilità dell'applicazione. Si compone di tre componenti principali:

- **Model:** Rappresenta i dati dell'applicazione e gestisce la logica di business. Il Model è responsabile per l'elaborazione dei dati, il recupero e l'aggiornamento e l'aggiornamento dello stato di persistenza.
- **View:** Gestisce l'interfaccia utente e la presentazione dei dati. La View è responsabile per la visualizzazione dei dati provenienti dal modello in un formato compatibile all'utente.
- **Controller:** Gestisce la logica di controllo dell'applicazione e coordina l'interazione tra il modello e la vista. Riceve gli input dall'utente della vista, richiede le operazioni appropriate al modello e aggiorna la vista con i risultati.

Una architettura ibrida client – server con MVC combina i vantaggi di entrambi. Il client può includere la vista e il controller, mentre il server gestisce il modello e fornisce i servizi di back-end. Ciò consente una separazione chiara tra l'interfaccia utente e la logica di business, consentendo una maggiore modularità, riusabilità del codice e facilitando la manutenzione dell'applicazione.

Decomposizione del sistema in sottosistemi

L'architettura scelta per lo sviluppo del sistema è stata decomposta in diversi sottosistemi o componenti per gestire le diverse funzionalità.

Sottosistema Client: Il sottosistema Client rappresenta il lato client dell'architettura ibrida. Può essere suddiviso ulteriormente in sottosistemi:

- **View (Vista):** Questo componente gestisce l'interfaccia utente, può includere pagine HTML e i componenti dell'interfaccia grafica dell'applicazione. È responsabile per la visualizzazione dei dati provenienti dal Controller e per la gestione degli eventi dell'utente.
- **Controller:** Questa componente gestisce la logica di controllo dell'interfaccia utente. Riceve gli input dell'utente della Vista e coordina le richieste al Server tramite richieste HTTP. Si occupa anche della validazione dei dati e delle azioni da intraprendere in base alle interazioni dell'utente.

Sottosistema Server: Il sottosistema Server il lato server dell'architettura ibrida. Può essere suddiviso ulteriormente in sottocomponenti:

- **Model (Modello):** Questo componente rappresenta i dati e la logica di business dell'applicazione. Gestisce l'accesso ai dati persistenti nel Database. Fornisce i metodi per leggere, scrivere, aggiornare e cancellare i dati.
- **Controller:** Questo componente gestisce la logica di business dell'applicazione e coordina l'interazione tra la View e il Model. Riceve le richieste del Cliente, elabora i dati nel Modello e restituisce i risultati alla View. Si occupa della logica di elaborazione dei dati, dell'autenticazione e delle operazioni specifiche dell'applicazione.

Sottosistema Storage: Il sottosistema Storage gestisce l'archiviazione persistente dei dati dell'applicazione. Includere il Database. Può essere organizzato in diversi componenti

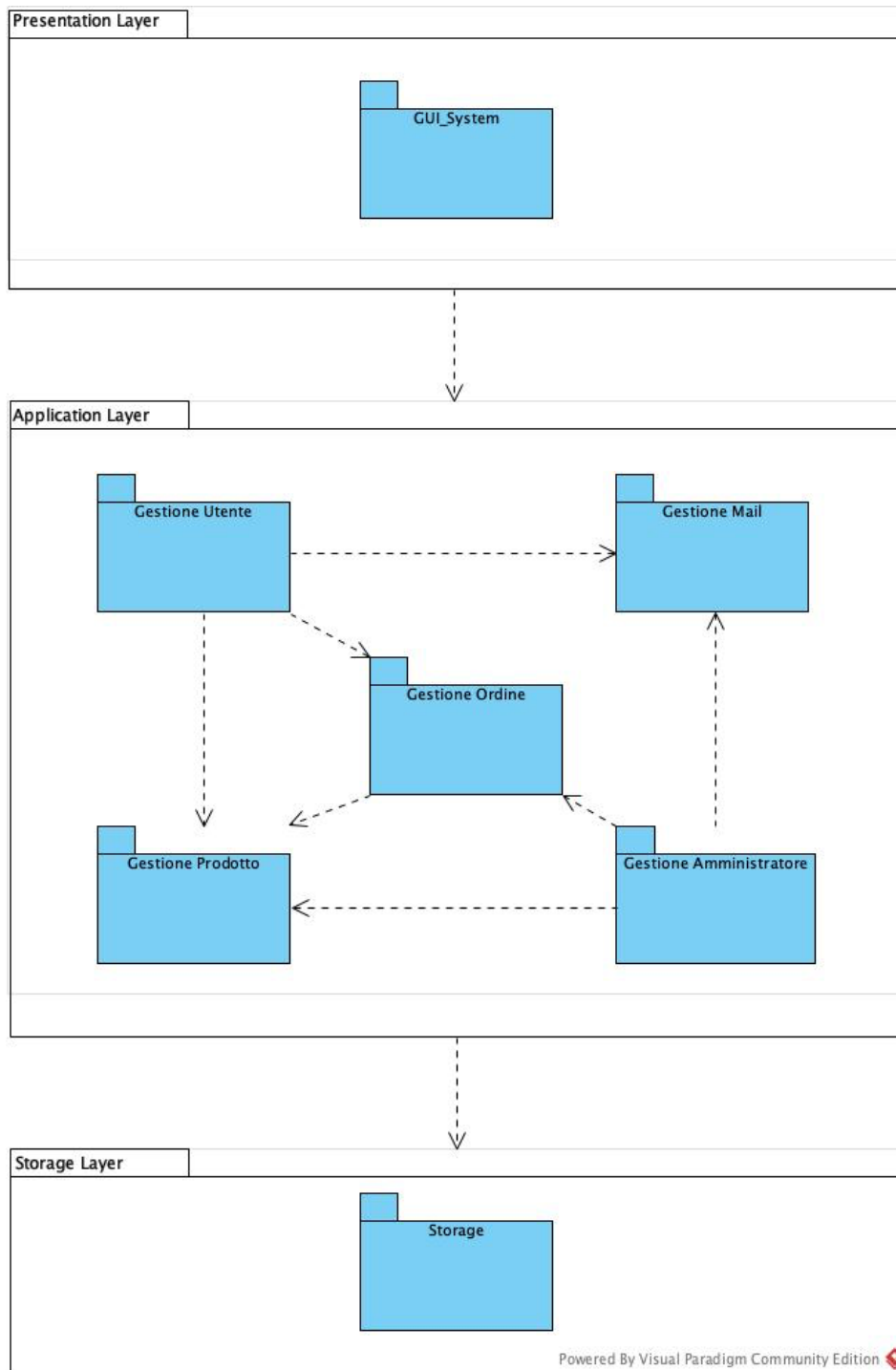
- **Database:** Questo componente gestisce la memorizzazione dei dati in modo strutturato. È implementato utilizzando un database relazionale (MySQL).
- **Cache:** Questo componente può essere utilizzato per memorizzare temporaneamente i dati che vengono richiesti in modo frequente per migliorare le prestazioni dell'applicazione.

Suddivisione dei sottosistemi

Per semplificarne la progettazione l'applicazione client server è stata suddivisa in tre strati distinti, ciascuno con responsabilità specifiche.

- **Livello di presentazione:** Questo è il livello più vicino all'utente finale e si occupa dell'interfaccia utente e della presentazione dei dati. Include componenti come le pagine web, le interfacce grafiche utente (GUI) o le applicazioni mobili.
- **Livello di logica applicativa:** Questo livello gestisce la logica di business dell'applicazione. È responsabile per il trattamento delle richieste dell'utente, l'elaborazione dei dati, la logica di validazione e la gestione delle regole di business. Questo livello può essere implementato tramite servizi web, API o componenti dell'applicazione.
- **Livello di dati:** Questo è il livello più basso dell'architettura three-tier e gestisce la persistenza dei dati. Include il database o altri sistemi di archiviazione dati. Il livello di dati si occupa dell'archiviazione, del recupero, dell'aggiornamento e della gestione dei dati necessari per l'applicazione.

Questa architettura separa le responsabilità e le funzioni dell'applicazione in questi tre livelli distinti, fornendo una struttura scalabile, modulare e flessibile. Ogni livello può essere sviluppato e gestito indipendentemente dagli altri, consentendo la riutilizzabilità del codice e una maggiore manutenibilità del sistema.



GUI_System: fornisce l'interfaccia utente per i client.

GestioneUtente: sottosistema che gestisce tutti i servizi di interesse di un utente acquirente.

Gestione Amministratore: sottosistema che gestisce i servizi legati alla gestione del sistema da parte degli amministratori.

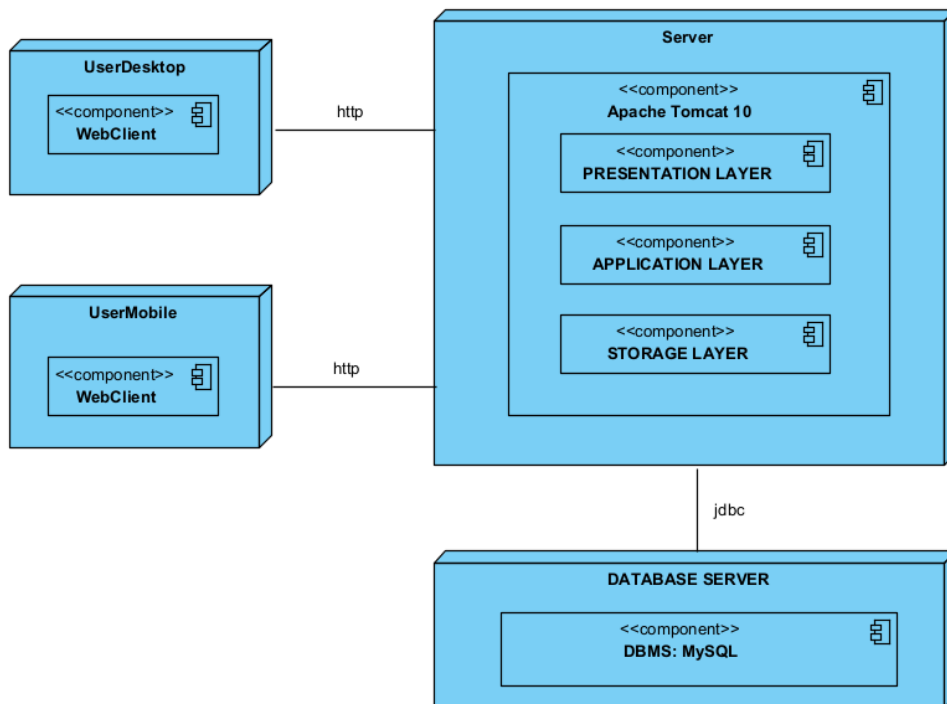
GestioneOrdine: sottosistema che gestisce tutti i servizi riguardanti la gestione degli ordini.

GestioneProdotto: sottosistema che gestisce i servizi legati alla gestione dei prodotti.

GestioneMail: sottosistema che si occupa di gestire l'invio di e-mail.

Storage: gestisce il salvataggio e il recupero dei dati dal database.

Mapping Hardware/Software



WebClient: Browser come Mozilla Firefox, Internet Explorer, Chrome, ...

Web Server: Tomcat 10

Presentation Layer: jsp

Application Layer: servlet

Storage Layer: DBMS MySQL

DataBase Server: Database relazionale MySQL

In un'architettura ibrida client-server con l'adozione del pattern architetturale MVC (Model-View-Controller), il mapping hardware/software coinvolge l'assegnazione dei componenti hardware e software ai vari strati del modello MVC.

Hardware: Il mapping hardware riguarda l'allocazione dei componenti hardware necessari per supportare l'architettura client-server. Questi componenti possono includere i server, che possono essere macchine fisiche o virtuali, che ospitano l'applicazione server e il database. Inoltre, i client possono essere dispositivi come computer, smartphone o tablet, che eseguono l'applicazione client e comunicano con il server.

Software: Il mapping software riguarda l'assegnazione dei componenti software alle varie parti dell'architettura ibrida client-server e al pattern MVC.

- **Model:** Il modello rappresenta la logica di business e i dati dell'applicazione. Il mapping software nel modello coinvolge lo sviluppo dei componenti software che gestiscono la logica di business e l'accesso ai dati. Questi componenti possono includere classi, librerie o servizi che gestiscono l'interazione con il database e forniscono funzionalità per la logica di business dell'applicazione.
- **View:** La vista rappresenta l'interfaccia utente dell'applicazione. Il mapping software nella vista coinvolge la creazione dei componenti software responsabili della presentazione dell'interfaccia utente al client. Questi componenti possono includere pagine HTML, fogli di stile CSS o altri elementi che definiscono l'aspetto e il layout dell'interfaccia utente.
- **Controller:** Il controller gestisce il flusso dell'applicazione e coordina le interazioni tra il modello e la vista. Il mapping software nel controller coinvolge la creazione di componenti software che gestiscono le richieste del client, invocano le azioni appropriate nel modello e determinano quale vista deve essere presentata al client. Questi componenti possono includere classi o servizi che gestiscono la logica di routing, l'autenticazione, la validazione dei dati e altre attività correlate.

Gestione dei dati persistenti

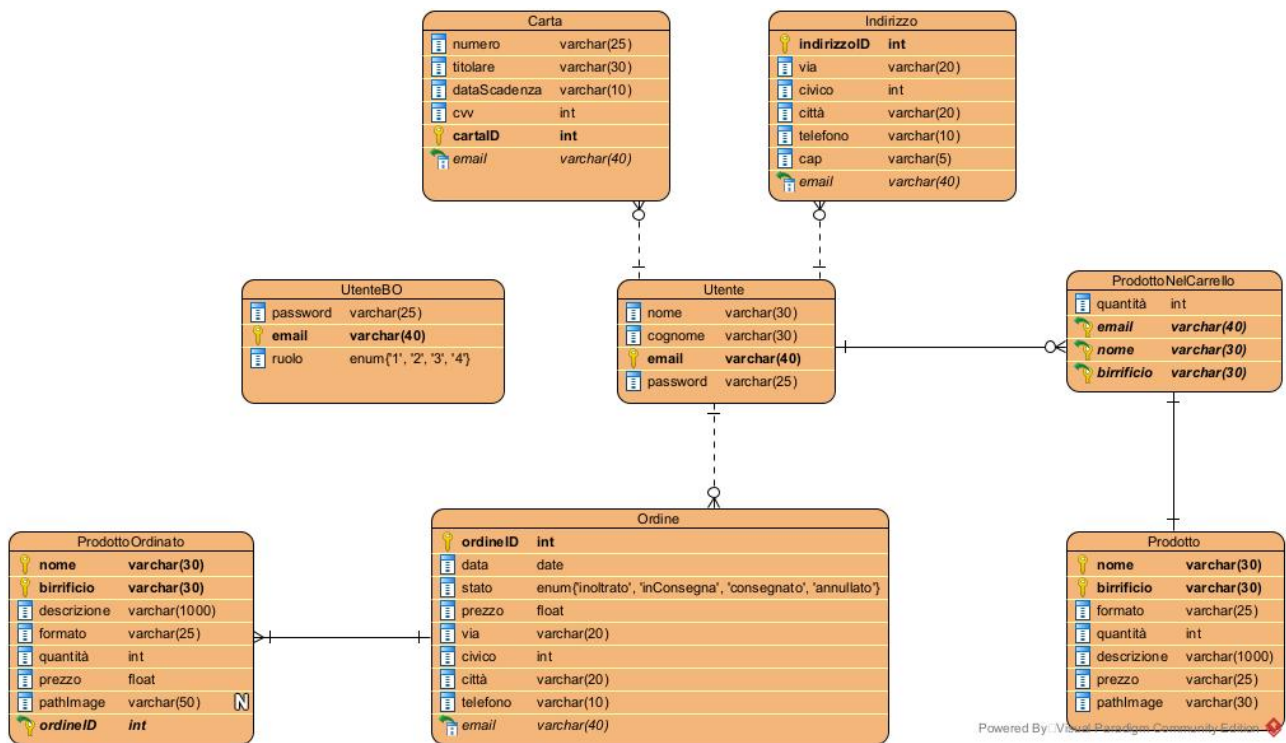


Tabelle del database

Utente		
nome	varchar(30)	not null
cognome	varchar(30)	not null
email	varchar(40)	primary key
password	varchar(25)	not null
primary key(email)		

Prodotto		
nome	varchar(30)	not null
birrifficio	varchar(30)	not null
descrizione	varchar(1000)	null
formato	varchar(25)	not null
quantità	int	not null
prezzo	varchar(25)	not null
pathImage	varchar(250)	null
primary key(nome, birrifficio)		

Carta		
cartalID	int	auto_increment
numero	varchar(25)	not null
email	varchar(40)	not null
cvv	int	not null
titolare	varchar(30)	not null
dataScadenza	varchar(10)	not null
primary key(cartalID)		
foreign key(email) references utente(email)		

UtenteBO		
email	varchar(40)	primary key
password	varchar(25)	not null
ruolo	enum {'1', '2', '3', '4'}	not null

Ordine		
OrdineID	int	auto_increment
email	varchar(40)	not null
data	varchar(25)	
prezzo	float	not null
via	varchar(25)	not null
civico	int	not null
città	varchar(20)	not null
telefono	varchar(10)	not null
stato	enum {'inoltrato', 'inConsegna', 'consegnato', 'annullato'}	
primary key (ordineID)		
foreign key (email) references utente(email)		

Indirizzo		
IndirizzoID	int	auto_increment
via	varchar(20)	not null
civico	int	not null
città	varchar(20)	not null
telefono	varchar(10)	not null
cap	varchar(5)	not null
email	varchar(40)	not null
primary key (indirizzoID)		
foreign key (email) references utente(email)		

ProdottoOrdinato		
nome	varchar(30)	not null
birrifficio	varchar(30)	not null
descrizione	varchar(1000)	null
formato	varchar(25)	not null
quantità	int	not null
prezzo	folat	not null
ordineID	int	not null
pathImage	varchar(50)	not null
primary key (ordineID, nome, birrifficio)		
foreign key (ordineID)		

ProdottoNelCarrello		
email	varchar(40)	not null
nome	varchar(30)	not null
foreign key(nome, birrifficio) references prodotto(nome, birrifficio)		
primary key(nome, birrifficio, email)		
foreign key(email) references utente(email)		
birrifficio	varchar(30)	not null
quantità	int	not null

Controllo accessi e sicurezza

BEerHAPPY è un sistema multi-user che consente ad utenti diversi di svolgere operazioni differenti sui vari oggetti.

Per rappresentare le operazioni che possono effettuare i diversi attori sugli oggetti è stata utilizzata una matrice degli accessi.

Matrice degli accessi

Oggetto/attore	Utente guest	Utente store	Utente back-office	Utente Manager
Utente	Registrazione Login Recupero password	Visualizza dati Modifica dati Modifica password Elimina account		
UtenteBO	Login Recupero password			Visualizza utente Aggiunge utente Modifica ruoli Rimuove utente
Carrello	Visualizza carrello Rimuove prodotto Aggiunge prodotto	Visualizza carrello Rimuove prodotto Aggiunge prodotto		
Prodotto	Visualizza prodotto	Visualizza prodotto	Aggiunge prodotto Rimuove prodotto Modifica prodotto	Aggiunge prodotto Rimuove prodotto Modifica prodotto
Ordine		Effettua ordine Visualizza ordine Annulla ordine	Visualizza ordine Modifica stato	Visualizza ordine Modifica stato
Carta		Visualizza carta Aggiunge carta Rimuove carta		
Indirizzo		Visualizza indirizzo Aggiunge indirizzo Rimuove indirizzo		

Controllo globale software

Il controllo del flusso del software viene gestito da classi servlet che interagiscono con il client svolgono le varie operazioni. Il server smista ogni nuova richiesta alla classe servlet adeguata, inoltrando poi la risposta al client.

Il sistema utilizzerà un controllo del flusso event-driven in quanto il flusso del programma è principalmente determinato dal verificarsi di eventi esterni.

Gli eventi esterni saranno causati dagli utenti collegati alla piattaforma tramite link e pulsanti sulle pagine.

Il programma risponde a eventi che si verificano nel sistema o nell'ambiente circostante, invece di eseguire un flusso sequenziale di istruzioni.

Boundary condition

Per poter usufruire del sistema l'utente necessita di una connessione alla rete e di un Browser.

First Start-Up

Per il primo start-up è necessario avviare il DBMS che fornisce servizi di un database MySQL.

Prima dell'avvio l'utente amministratore che ha accesso alla macchina deve configurare almeno un account amministratore per permettere la gestione dei servizi (gestione ordini, prodotti, utenti).

L'account verrà poi fornito all'utente back-office che si occupa della gestione del sito, successivamente l'amministratore avvierà la macchina necessaria per l'esecuzione della web app (Tomcat – MySQL).

Terminazione

Nel caso in cui si necessita terminare il servizio l'utente amministratore che ha accesso alla macchina dovrà terminare l'esecuzione del web server e del DBMS, assicurando la consistenza dei dati e annullando eventuali operazioni ancora in esecuzione.

Start-Up a seguito di un fallimento

Potrebbero verificarsi diversi casi di fallimento del sistema.

Nel caso in cui vi fosse l'interruzione della connessione alla rete, i servizi del sistema non saranno disponibili fino al ripristino della connessione.

Un altro fallimento potrebbe verificarsi a seguito di un'interruzione inaspettata dell'alimentazione.

In tal caso non vi sono metodi che possano ripristinare lo stato del sistema al momento dello spegnimento.

Un ulteriore caso di fallimento potrebbe essere di tipo hardware (hardware obsoleto o guasto). A tal proposito è previsto un backup periodico che aiuti a non perdere i dati.

Infine, a seguito di un fallimento del software dovuto ad errori commessi nella fase di implementazione non è prevista nessuna procedura che ristabilisca il corretto funzionamento del sistema. L'unica procedura da adottare in questo caso prevede la chiusura del sistema e successivamente il riavvio.

Servizi dei sottosistemi

Servizi di Gestione Amministratore

Sottosistema	Servizi
Login	Permette ad un amministratore di autenticarsi come amministratore.
Logout	Permette all'amministratore di uscire dal sistema.
Cambio ruolo	Consente il trasferimento di un amministratore alla pagina in cui può selezionare un ruolo.
Visualizza ordini	Permette ad un amministratore di accedere alle funzionalità di gestione degli ordini.
Cambia lo stato di un ordine	Consente di modificare lo stato di un ordine.
Visualizza prodotti	Permette ad un amministratore di accedere alle funzionalità di gestione dei prodotti.
Aggiunta prodotto	Permette ad un amministratore di aggiungere un prodotto.
Rimozione prodotto	Consente ad un amministratore di rimuovere un prodotto dal sistema.
Modifica prodotto	Permette ad un utente amministratore di modificare i dati di un prodotto.
Aggiunta utente back-office	Permette all'utente Manager di aggiungere un utente back-office.
Rimozione utente back-office	Consente all'utente Manager di rimuovere un utente back-office.
Modifica ruoli utente back-office	Permette all'utente Manager di modificare i ruoli assegnati ad un utente back-office.
Visualizza utenti	Permette all'utente Manager di visualizzare gli utenti back-office.

Servizi di GestioneUtente

Sottosistema	Servizi
Login	Consente ad un utente di autenticarsi al sistema.
Logout	Consente all'utente di uscire dal sistema.
Recupero password	Consente all'utente di ricevere una nuova password per accedere al sistema.
Registrazione	Permette all'utente di registrare i suoi dati nel sistema.
Visualizza dati	Permette all'utente di visualizzare i suoi dati all'interno del proprio profilo.
Modifica dati	Permette all'utente di modificare i propri dati personali.
Eliminazione account	Consente all'utente di rimuovere i propri dati dal sistema.
Modifica password	Consente all'utente di modificare la password.
Aggiunta carta	Permette all'utente di aggiungere una carta.
Rimozione carta	Permette all'utente di rimuovere una carta.
Aggiunta indirizzo	Permette all'utente di aggiungere un indirizzo.
Rimozione indirizzo	Permette all'utente di rimuovere un indirizzo.
Visualizza ordini	Permette ad un utente di visualizzare i propri ordini.
Annulla ordine	Permette ad un utente di annullare un ordine.
Visualizza carrello	Consente all'utente di visualizzare il carrello.
Aggiunta prodotto al carrello	Consente ad un utente di aggiungere un prodotto al carrello.
Rimozione prodotto dal carrello	Permette ad un utente di rimuovere un prodotto dal carrello.
Aggiunta indirizzo Checkout	Permette all'utente di aggiungere un indirizzo in fase di checkout.
Aggiunta carta Checkout	Permette all'utente di aggiungere un indirizzo in fase di checkout.
Selezione indirizzo	Permette all'utente di selezionare un indirizzo in fase di checkout.
Selezione carta	Permette all'utente di selezionare una carta in fase di checkout.

Effettua ordine	Permette all'utente di effettuare un ordine.
Cerca prodotto	Permette all'utente di cercare un prodotto.
Visualizza prodotto	Consente all'utente di visualizzare i dati di un prodotto.

Servizi di GestioneProdotto

Sottosistema	Servizi
Aggiunta prodotto	Consente di aggiungere un prodotto.
Modifica prodotto	Consente di modificare un prodotto.
Rimozione prodotto	Permette di rimuovere un prodotto.
Recupero dati	Permette di recuperare i dati di un prodotto

Servizi di GestioneOrdine

Sottosistema	Servizi
Aggiunta ordine	Consente di aggiungere un prodotto.
Modifica stato ordine	Consente di modificare un prodotto.
Recupero prodotti	Permette di recuperare i prodotti di un ordine.
Recupero dati	Permette di recuperare i dati di un ordine.

Servizi di GestioneMail

Sottosistema	Servizi
Invia e-mail	Permette di inviare un'e-mail.