



System Design Document

INDITEX **ReSkin**

Riferimento	INDITEXReSkin_SDD_2.0
Versione	2.0
Data	22/11/2023
Destinatario	C. Gravino
Presentato da	A. Ruggiero, G. Gurràli



Team composition

Ruolo	Nome	Acronimo	Contatti
Top Management	Carmine Gravino	CG	gravino@unisa.it
Team Member	Andrea Ruggiero	AR	a.ruggiero150@studenti.unisa.it
Team Member	Giovanni Gurrallì	GG	g.gurrallì@studenti.unisa.it



Sommario

Team composition	2
Revision history	4
1. Introduzione	5
1.1. Scopo del sistema:	5
1.2. Design Goals	6
1.3. Trade-offs	8
1.4. Definizioni, acronimi e abbreviazioni	8
1.5. Riferimenti.....	9
1.6. Panoramica.....	9
2. Architettura del sistema corrente	10
3. Architettura del sistema proposto	12
3.1. Panoramica.....	12
3.2. Decomposizione in sottosistemi	13
3.3. Mapping Hardware/Software	17
3.4. Gestione dati persistenti	19
3.5. Controllo degli accessi e sicurezza	21
3.6. Controllo del flusso globale del sistema	22
3.7. Condizioni Boundary	23
4. Servizi dei sottosistemi	24
5. Glossario	28



Revision history

Data	Versione	Descrizione	Autori
22/11/2023	0.1	Stesura struttura documento	[TEAM MEMBER]
24/11/2023	0.2	Scrittura dei riferimenti, panoramica ed introduzione della sezione "Design Goals"	[TEAM MEMBER]
25/11/2023	0.3	Definizione completa dei "Design Goals" e dei "Trade-off"	[TEAM MEMBER]
27/11/2023	0.4	Architettura del sistema proposto: Panoramica	[TEAM MEMBER]
29/11/2023	0.5	Decomposizione in sottosistemi: introduzione	GG
30/11/2023	0.6	Component diagram	GG
1/12/2023	0.6	Diagramma architetturale	GG
1/12/2023	0.7	Controllo degli accessi e sicurezza-Controllo del flusso globale del sistema	AR
3/12/2023	0.8	Condizione Boundary: Errore nel recupero dei dati	GG
3/12/2023	0.9	Condizione Boundary:	AR
5/12/2023	0.10	Entity class diagram	AR
7/12/2023	0.11	DBMS_SDD	AR
12/12/2023	0.12	Deployment diagram	[TEAM MEMBER]
12/12/2023	0.13	Servizi dei sottosistemi	GG
12/12/2023	0.14	Glossario	AR
17/12/2023	1.0	Revisione generale	[TEAM MEMBER]
11/02/2024	1.1	Revisione e modifiche post feedback	[TEAM MEMBER]
14/02/2024	2.0	Revisione finale	[TEAM MEMBER]

1. Introduzione

1.1. Scopo del sistema:

Nel corso dell'ultimo decennio, nonostante gli sforzi considerevoli compiuti da agenzie e governi per promuovere una vera e propria transizione ecologica attraverso ampi programmi di sviluppo, diversi settori continuano a lottare per adottare le pratiche comuni necessarie per lo smaltimento adeguato dei rifiuti. Uno dei settori più inquinanti che ha scalato posizioni, raggiungendo il secondo posto nella graduatoria tra quelli più dannosi per l'ambiente, è senz'altro quello della moda, in particolare il segmento noto come fast fashion, che si concentra sulla produzione in massa di capi d'abbigliamento. Diversi report hanno evidenziato la presenza di metano e petrolio, tra altre sostanze chimiche, in materiali ampiamente utilizzati dalle grandi aziende, come cotone, seta e fibre sintetiche. Queste sostanze, se non smaltite correttamente, possono contaminare le acque sotterranee o persistere nell'ambiente senza degradarsi, causando un inquinamento irreversibile del nostro pianeta. Inoltre, oggi, le aziende producono all'incirca 92 milioni di tonnellate di rifiuti tessili ogni anno, per ogni chilo di cotone consumano più di 20.000 litri d'acqua e quasi il 10% delle microplastiche disperse nell'oceano provengono dai tessuti non adeguatamente smaltiti. Problemi singoli ma che nell'insieme macchiano tutti i buoni propositi che da anni cerchiamo di portare avanti; propositi quali la sostenibilità ed il riciclo consapevole. Risolvere tutto e subito è irrealistico ma è bene iniziare a pensare a soluzioni alternative che possano modificare questa tendenza negativa. L'azienda INDITEX propone quindi un sistema innovativo chiamato INDITEX ReSkin. L'obiettivo principale di questo sistema è creare un canale di comunicazione che consenta a piccoli artigiani e innovative start-up italiane di acquistare a prezzi vantaggiosi il materiale di produzione in eccesso. L'azienda potrà dunque mettere in vendita sottoprodotti (ovvero uno scarto di lavorazione derivante da processi industriali, il quale viene riutilizzato in un altro processo produttivo come materia prima non vergine) derivanti dalla produzione di capi per i suoi principali brand di Fast Fashion. Questo approccio permette all'azienda di recuperare una parte delle spese sostenute per l'acquisto dei materiali, riducendo al contempo gli sprechi e promuovendo il riutilizzo delle stoffe e di altri materiali altrimenti destinati allo spreco. In particolare, l'azienda ha deciso di puntare sul mercato italiano proprio per la grande tradizione che l'Italia ha e ancora oggi mette in campo quando si parla di artigianato e innovazione nell'ambito della moda, puntando quindi a valorizzare questa eredità culturale ma al tempo stesso valorizzando i giovani talenti del territorio, che si mettono in gioco creando start-up innovative e all'insegna del green.



1.2. Design Goals

Attraverso i design goals (o obiettivi di design) andiamo ad identificare quelle che sono le qualità su cui il sistema deve essere focalizzato, andando quindi a specificare quali caratteristiche dovrebbero essere ottimizzate e per loro definire priorità diverse. Gli obiettivi di design possono essere selezionati da una lista di qualità:

- **Performance:** Assicurarsi che il software raggiunga i livelli desiderati di velocità ed efficienza durante l'esecuzione
- **Dependability:** Garantire la fiducia e l'affidabilità del software attraverso la gestione degli errori e la disponibilità continua
- **Cost:** Ottimizzare l'uso delle risorse finanziarie, assicurando che lo sviluppo e il mantenimento del software siano economicamente sostenibili
- **Maintenance:** Facilitare la manutenzione del software nel tempo, rendendolo adattabile a cambiamenti e aggiornamenti senza eccessivo sforzo
- **End user:** Concentrarsi sulle esigenze e le aspettative degli utenti finali, garantendo un'esperienza d'uso soddisfacente e intuitiva

Andremo quindi a definire i nostri obiettivi elencandoli in una tabella, definita dai seguenti campi, uno per ogni colonna:

- **Priorità:** Indica la priorità rispetto agli altri design goal
- **ID:** Identificatore per il design goal
- **Descrizione:** Descrive le caratteristiche del design goal
- **Categoria:** Inquadra il design goal in una delle qualità indicate in precedenza
- **Origine:** Indica il requisito non funzionale dal quale è stato ricavato
- **Trade-off:** Scelta di due parametri in cui la diminuzione di uno indica l'aumento dell'altro



Priorità	ID	Descrizione	Categoria	Origine	Trade-off
1	DG_1 Persistenza dei dati	Il sistema deve poter immagazzinare ed interrogare i dati su un database SQL	Dependability	RNF_I.4	Tempi di risposta VS Sicurezza
2	DG_2 Riservatezza	Il sistema garantisce la riservatezza dei dati sensibili personali, criptandoli in fase di acquisizione	Dependability	RNF_A.1	Tempi di risposta VS Sicurezza
3	DG_3 Reperibilità	Il sistema deve essere usabile in ogni momento, 24 ore su 24	Dependability	RNF_A.2	Costi VS Affidabilità
4	DG_4 Tempi di risposta	Il sistema deve rispondere alle azioni dell'utente entro e non oltre i 3 secondi.	Performance	RNF_P.1	Tempi di risposta VS Sicurezza
5	DG_5 User Friendly	Il sistema deve essere di facile utilizzo per tutti gli utenti. Ogni utente deve poter navigare facilmente al sito, compiendo ogni azione senza ambiguità. Ogni aspetto grafico del sito deve favorire la user experience	End-user	RNF_U.1	Tempo di rilascio VS Qualità
6	DG_6 Responsive	Il sito, siccome sarà reperibile su diversi browser eseguibili su diverse piattaforme, dovrà adattarsi a tutti i diversi tipi di schermi.	End-user	RNF_U2	Prestazioni VS Compatibilità
7	DG_7 Evoluzione	Il sistema potrà essere facilmente ampliato con nuove funzionalità	Maintenance	RNF_S.1	Tempo di rilascio VS Qualità
8	DG_8 Tutela della Privacy	Tutti i dati sensibili gestiti dal sistema sono tutelati ai sensi delle legge del regolamento UE	Dependability	RNF_L.1	Tempi di risposta VS Sicurezza

1.3. Trade-offs

Trade-off	Descrizione
Tempo di rilascio VS Qualità	Al fine di rispettare la deadline del progetto, i tempi di consegna avranno priorità rispetto alla qualità del software; pur garantendo che la qualità rimarrà al di sopra degli standard minimi pattuiti.
Tempo di risposta VS Sicurezza	Il sistema dovendo gestire dati personali ed effettuare operazioni su database si è deciso di dilatare i tempi di risposta al fine di garantire un livello di sicurezza adeguato. I tempi di risposta per accedere ed interagire con il database rimarranno comunque al di sotto dei tre secondi
Costi VS Affidabilità	Al fine di rendere il sistema reperibile 24 ore su 24 è necessario l'utilizzo di server ad elevate prestazioni, i quali però aumentano il costo del sistema
Prestazioni VS Compatibilità	Abbiamo deciso di privilegiare la compatibilità rispetto alle prestazioni al fine raggiungere un pubblico più ampio, garantire la sostenibilità a lungo termine, adattarsi a evoluzioni del mercato e rispettare standard

1.4. Definizioni, acronimi e abbreviazioni

Acronimi:

- **DG:** Design Goal
- **RNF:** Requisito non funzionale
- **AGU:** Autenticazione e Gestione Utenti
- **GN:** Gestione delle Notifiche
- **GO:** Gestione degli Ordini
- **GC:** Gestione del Carrello
- **GI:** Gestione delle Inserzioni
- **RVP:** Ricerca e Visualizzazione dei Prodotti
- **PO:** Piazzamento degli Ordini

Definizioni:

- **Requisito non funzionale:** Specifica che descrive attributi di sistema, come prestazioni, sicurezza o usabilità, che influenzano l'efficacia del software senza definire comportamenti specifici. Contrariamente ai requisiti funzionali, che delineano le funzioni del sistema, i requisiti non funzionali riguardano le qualità globali del sistema.
- **Autenticazione e Gestione Utenti:** Sottosistema che definisce che gestisce l'accesso al sito da parte degli utenti
- **Gestione delle Notifiche:** Sottosistema che gestisce le notifiche POP inviate e ricevute
- **Gestione degli Ordini:** Sottosistema che gestisce gli ordini
- **Gestione del Carrello:** Sottosistema che gestisce il carrello
- **Gestione delle Inserzioni:** sottosistema che gestisce l'inserimento, la modifica e l'eliminazione di un'inserzione
- **Ricerca e Visualizzazione dei Prodotti:** Sottosistema che gestisce la ricerca e la visualizzazione dei prodotti
- **Piazzamento degli Ordini:** Sottosistema che gestisce il piazzamento degli ordini

1.5. Riferimenti

- **Libro:** "Object Oriented Software Engineering using UML, Patterns and Java". Edizione: 3rd Edition Anno: 2014. Autori: Bernd Bruegge, Allen H. Dutoit
- **INDITEXReskin_SOW**
- **INDITEXReskin_RAD**
- [INDITEX](#)

1.6. Panoramica

Il documento è stato strutturato in maniera tale da poter essere diviso in quattro macroaree:

- **Introduzione:** Fornisce una panoramica generale del documento, delineando lo scopo, il contesto e gli obiettivi del progetto. Includere anche una breve descrizione dei problemi affrontati e degli obiettivi da raggiungere
- **Architettura del sistema corrente:** Descrive l'architettura competitor dell'attuale sistema software in uso. Include dettagli sulle componenti principali e le relazioni tra di esse
- **Architettura del sistema proposto:** Presenta la progettazione proposta per il nuovo sistema software. Questa sezione includere diagrammi, modelli e spiegazioni dettagliate delle scelte architettureali.
- **Servizi dei sottosistemi:** Espone i servizi offerti da ciascun sottosistema all'interno del sistema proposto. Questa sezione offre una visione più dettagliata delle funzionalità e delle responsabilità di ogni componente del sistema
- **Glossario:** Fornisce una lista di definizioni per i termini tecnici o specifici del dominio utilizzati nel documento. Questo aiuta a garantire una comprensione comune tra tutti gli stakeholder del progetto e riduce la possibilità di interpretazioni errate dei concetti chiave

2. Architettura del sistema corrente

- **Sistema simile - FABSCARP.ORG:**

- 1. Architettura Generale:**

Il sistema adotta un'architettura a microservizi per garantire una maggiore flessibilità e scalabilità. I microservizi sono distribuiti su server cloud per garantire prestazioni ottimali e una gestione efficiente delle risorse.

- 2. Tecnologie Utilizzate:**

Il frontend del sito è sviluppato utilizzando le ultime tecnologie web, come React.js per la creazione di interfacce utente dinamiche e veloci. Il backend è implementato in Node.js per garantire una gestione asincrona delle richieste. La gestione dei dati è affidata a un database NoSQL, come MongoDB, per una rapida scalabilità orizzontale.

- 3. Struttura delle Pagine:**

Il design delle pagine segue le ultime tendenze nel settore dell'e-commerce, con un layout responsive per adattarsi a dispositivi di diversi formati. Le pagine includono un header con un menu di navigazione intuitivo, una sezione di anteprima prodotto dinamica e un processo di checkout semplificato.

- 4. Gestione dei Dati:**

I dati dei prodotti, degli ordini e degli utenti sono archiviati in un database NoSQL altamente performante. L'accesso ai dati è gestito attraverso API RESTful, permettendo una facile integrazione con servizi esterni e applicazioni client.

- 5. Sicurezza:**

La sicurezza del sito è una priorità assoluta. L'autenticazione degli utenti avviene attraverso protocolli moderni come OAuth2. Tutte le comunicazioni sono crittografate con HTTPS per proteggere i dati in transito. Le misure di sicurezza includono la gestione accurata delle sessioni, la prevenzione di attacchi DDoS e l'implementazione di controlli di sicurezza avanzati per proteggere i dati sensibili.

- 6. Scelte di Progettazione:**

La progettazione del sito è incentrata sull'esperienza utente, con un'interfaccia pulita e intuitiva. L'uso di tecnologie moderne consente una risposta rapida alle richieste degli utenti. I microservizi facilitano la manutenibilità del sistema e permettono l'implementazione di nuove funzionalità senza influire sulle altre parti del sistema.

- **Ispirazione modello metodo di pagamento - [PENS.COM](https://pens.com):**

- 1. Architettura Generale:**

I sistema segue un'architettura client-server, dove il client è rappresentato dal browser dell'utente e il server ospita l'applicazione web. L'interazione avviene mediante richieste HTTP tra il client e il server

- 2. Tecnologie Utilizzate:**

Il frontend del sito è implementato utilizzando HTML5, CSS3 e JavaScript per la creazione di interfacce dinamiche e reattive. Sul lato server, il backend è sviluppato in PHP, e fa uso di un database MongoDB

- 3. Struttura delle Pagine:**

Il layout delle pagine segue un design responsive, adattandosi a diverse dimensioni di schermo. Le pagine sono strutturate con un header che contiene il logo e il menu di navigazione, un corpo centrale con contenuto dinamico, e un footer con informazioni aggiuntive e link utili.

- 4. Gestione dei Dati:**

I dati vengono gestiti attraverso un database relazionale per la memorizzazione persistente. Il backend comunica con il database tramite query SQL.

- 5. Sicurezza:**

La sicurezza del sito è garantita mediante l'implementazione di HTTPS per la crittografia dei dati in transito. L'autenticazione degli utenti è gestita mediante OAuth. Vengono adottate pratiche di sicurezza per prevenire attacchi comuni attraverso XSS.

- 6. Scelte di Progettazione:**

La progettazione del sito si basa sui principi di usabilità e accessibilità. L'interfaccia utente è stata progettata in modo intuitivo per fornire una buona esperienza all'utente. La separazione chiara tra frontend e backend favorisce la manutenibilità e la scalabilità del sistema.

3. Architettura del sistema proposto

3.1. Panoramica

Lo sviluppo del sistema che stiamo proponendo segue lo stile architetturale “Three-tier”. Andiamo difatti a suddividere il nostro sistema in sottosistemi organizzati su tre livelli hardware differenti:

- **Interface layer**
- **Application logic layer**
- **Storage layer**

Questo tipo di approccio ci permetterà di dividere l'interfaccia dalla logica applicativa, consentendoci così di sviluppare e contestualmente di modificare diverse interfacce utente seguendo però la stessa logica applicativa.

Dal punto di vista delle tecnologie adottate per lo sviluppo abbiamo usato:

1. Lato Server (Back-end):

- **Java:** Utilizzato come linguaggio principale per lo sviluppo del back-end, gestendo la logica di business e le operazioni lato server

2. Interfaccia Utente (Front-end):

- **HTML e CSS:** Impiegati per definire la struttura e lo stile delle pagine web.
- **JavaScript (JS):** Utilizzato per aggiungere interattività e dinamicità alle pagine web.
- **Bootstrap 5.3:** Framework utilizzato per facilitare la progettazione di un'interfaccia utente responsive e migliorare la presentazione grafica.

3. Gestione delle Interazioni Client-Server:

- **JQuery:** Libreria JavaScript utilizzata per semplificare l'interazione con il DOM e fornire funzionalità aggiuntive per migliorare la user experience.

4. Gestione delle richieste asincrone:

- **AJAX:** Tecnologia impiegata per effettuare richieste asincrone al server senza dover ricaricare completamente la pagina, migliorando la reattività dell'applicazione.

5. Persistenza dei Dati (Database):

- **MySQL:** Sistema di gestione di database relazionali utilizzato per memorizzare e recuperare dati.

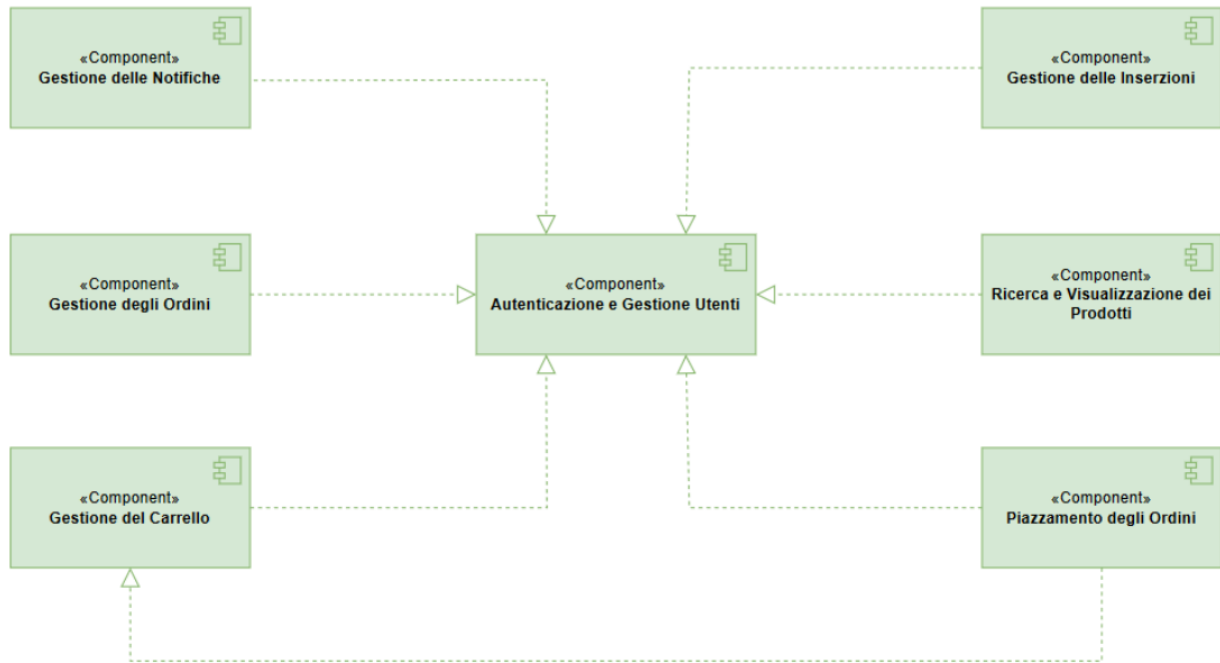
3.2. Decomposizione in sottosistemi

Al fine di semplificarne lo sviluppo, l'applicazione web è stata suddivisa nei seguenti sottosistemi:

- **Autenticazione e Gestione Utenti:** Questo sottosistema gestisce le operazioni legate all'autenticazione degli utenti e alla gestione dei loro account. Include la registrazione degli utenti admin attraverso nome-cognome-email-password e con la verifica attraverso un codice OTP, la registrazione degli utenti clienti attraverso nome-cognome-email-password, l'autenticazione tramite login e una funzione di logout.
- **Gestione delle Inserzioni:** Questo sottosistema è responsabile della gestione delle inserzioni nel sistema. Gli amministratori possono visualizzare gli annunci attivi, inserire nuovi annunci, modificarli o eliminarli dal sistema.
- **Gestione degli Ordini:** Questo sottosistema si occupa della gestione degli ordini nel sistema. Gli amministratori possono accedere a una lista di ordini visualizzarne i dettagli e modificarne lo stato. I clienti possono fare lo stesso per i propri acquisti (ottenere una lista e visualizzare i dettagli dei singoli acquisti)
- **Ricerca e Visualizzazione dei Prodotti:** Questo sottosistema fornisce funzionalità di ricerca e visualizzazione dei sottoprodotto presenti nel database. Gli utenti possono effettuare ricerche generiche, filtrate o basate su parole chiave e della lista ottenuta andare alla pagina specifica del prodotto.
- **Gestione del Carrello:** Questo sottosistema gestisce le operazioni legate al carrello degli acquisti. I clienti possono aggiungere prodotti al carrello, rimuoverli, modificare le quantità e visualizzare il contenuto del carrello.
- **Gestione delle Notifiche:** Questo sottosistema gestisce le notifiche all'interno del sistema. Gli utenti admin possono inviare notifiche POP (anche in modalità broadcast) e visualizzare le notifiche generali inviate. Gli utenti clienti possono visualizzare le notifiche ricevute in un'apposita sezione.
- **Piazzamento degli Ordini:** Questo sottosistema gestisce le fasi finali del processo di acquisto. I clienti possono aggiungere un indirizzo per la consegna e piazzare effettivamente l'ordine.

Component diagram UML

Attraverso il seguente diagramma andiamo a definire le relazioni tra i sottosistemi definiti nella fase di decomposizione



Abbiamo suddiviso il sistema secondo questa struttura per le seguenti motivazioni:

1. Sottosistema di Autenticazione e Gestione Utenti:

- **Descrizione:** Questo sottosistema gestisce le operazioni di autenticazione degli utenti, inclusa la registrazione e il login, e gestisce le informazioni utente.
- **Motivazione:** L'autenticazione è fondamentale per garantire l'accesso sicuro al sistema. La gestione utenti consente il monitoraggio e la personalizzazione dell'esperienza utente.

2. Sottosistema di Gestione delle Inserzioni:

- **Descrizione:** Questo sottosistema consente agli admin di gestire le inserzioni, inclusa la pubblicazione, la modifica e la rimozione di annunci.
- **Motivazione:** La gestione delle inserzioni è cruciale per mantenere aggiornati e pertinenti i prodotti o servizi offerti nel sistema.

3. Sottosistema di Gestione degli Ordini:

- **Descrizione:** Questo sottosistema gestisce il ciclo di vita degli ordini, consentendo agli admin di monitorare e gestire gli ordini e ai clienti di visualizzare e gestire i propri acquisti.
- **Motivazione:** La gestione degli ordini è essenziale per monitorare le transazioni, fornire assistenza e garantire una corretta esecuzione del processo di vendita.

4. Sottosistema di Ricerca e Visualizzazione dei Prodotti:

- **Descrizione:** Questo sottosistema offre funzionalità di ricerca e visualizzazione dei prodotti, consentendo agli utenti di cercare, filtrare i prodotti in base alle proprie esigenze e visualizzare la pagina specifica del prodotto richiesto
- **Motivazione:** La facilità di ricerca è cruciale per migliorare l'usabilità del sistema, aiutando gli utenti a individuare rapidamente i prodotti desiderati.

5. Sottosistema di Gestione del Carrello:

- **Descrizione:** Questo sottosistema consente ai clienti di gestire i prodotti nel proprio carrello, inclusa l'aggiunta, la rimozione e la modifica delle quantità.
- **Motivazione:** La gestione del carrello è fondamentale per facilitare l'esperienza di acquisto online, consentendo agli utenti di rivedere e modificare i prodotti che intendono acquistare.

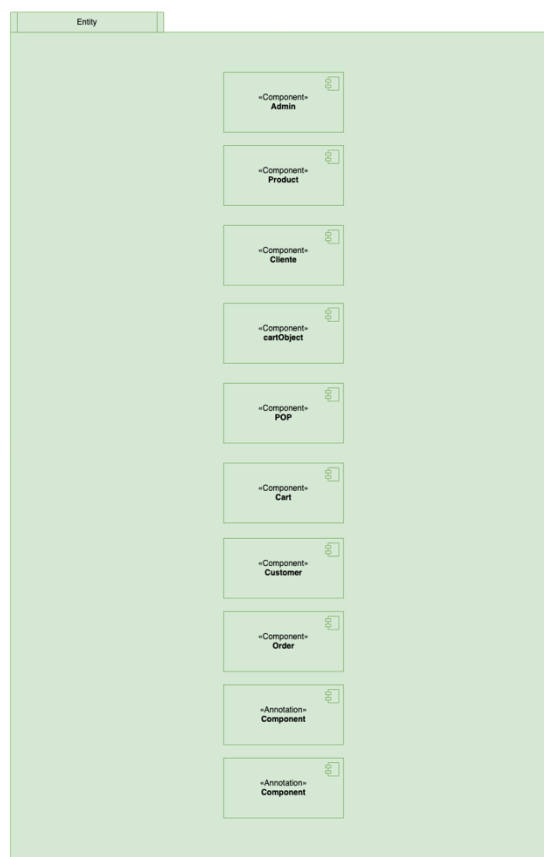
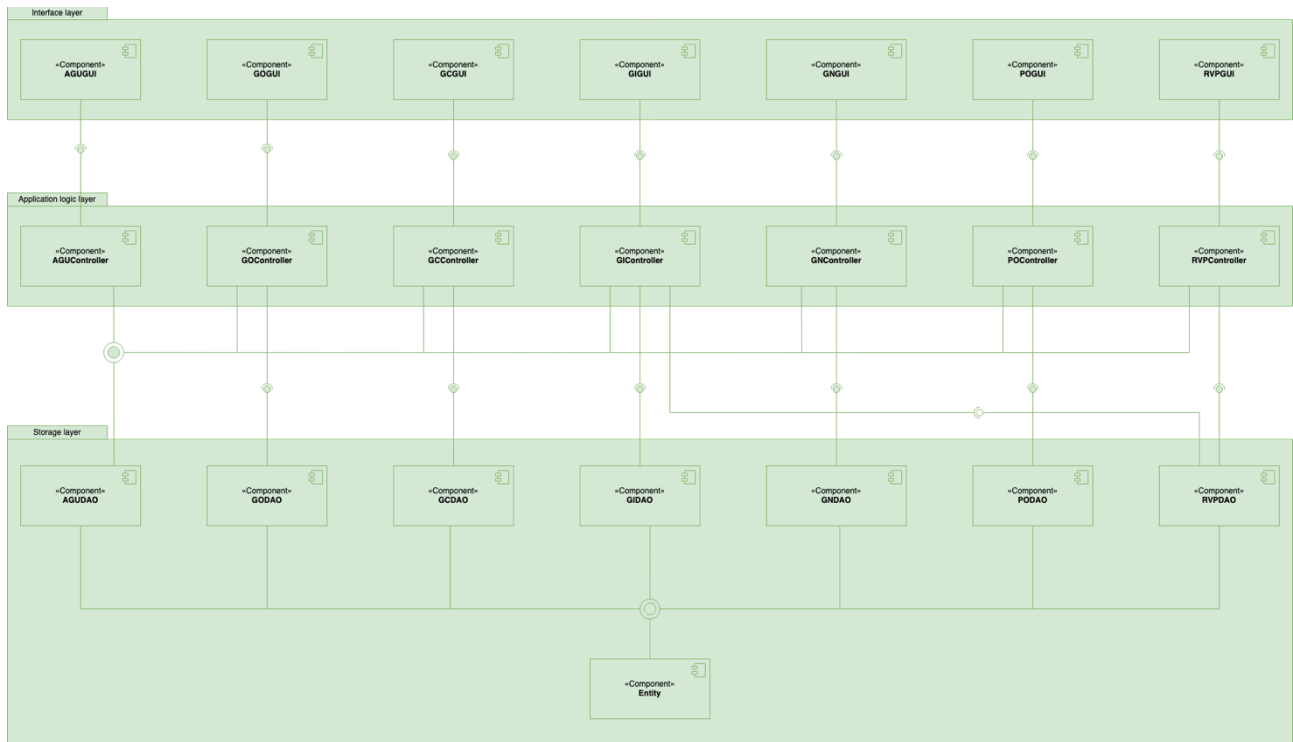
6. Sottosistema di Gestione delle Notifiche:

- **Descrizione:** Questo sottosistema gestisce l'invio e la ricezione di notifiche, inclusa la possibilità di inviare notifiche broadcast.
- **Motivazione:** Le notifiche migliorano l'interazione e l'immediatezza delle comunicazioni tra il sistema e gli utenti, fornendo informazioni cruciali o aggiornamenti.

7. Sottosistema di Piazzamento degli Ordini:

- **Descrizione:** Questo sottosistema gestisce le fasi finali del processo di acquisto, consentendo ai clienti di inserire un indirizzo e completare l'ordine.
- **Motivazione:** Il piazzamento degli ordini è il passo finale nell'esperienza di acquisto e richiede una gestione accurata per garantire la corretta esecuzione dell'ordine.

Diagramma architetturale

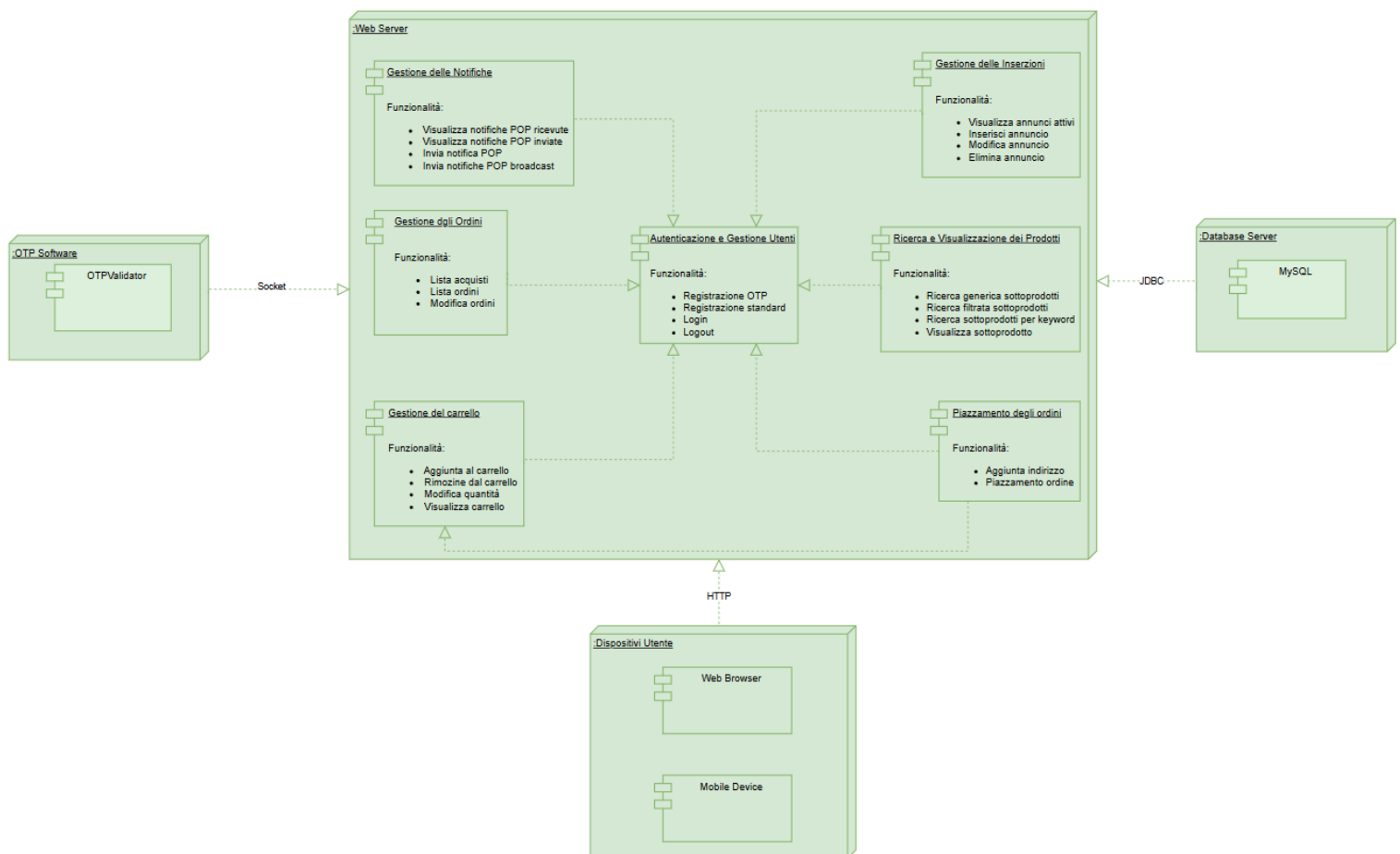


3.3. Mapping Hardware/Software

Il sistema proposto verrà sviluppato come una web-application cosicché qualsiasi utente che abbia a disposizione un dispositivo mobile o possa utilizzare un web browser, avrà accesso al nostro sito.

Il seguente deployment diagram descriverà meglio il modo in cui verrà mappato il sistema

Deployment Diagram



Si è scelto di strutturare nel seguente modo il Deployment Diagram per i seguenti motivi:

1. Web Server:

- **Descrizione:** Il Web Server ospita l'applicazione web che gestisce l'interfaccia utente e le funzionalità del sistema
- **Motivazione:** Un Web Server ci consente di accedere alle funzionalità del sistema attraverso una varietà di dispositivi utente tramite una connessione HTTP.

2. Dispositivi Utente:

- **Descrizione:** Rappresentano i dispositivi attraverso i quali gli utenti interagiscono con l'applicazione web, inclusi Web Browser e Mobile Devices
- **Motivazione:** Questa rappresentazione riflette la varietà di dispositivi che possono accedere all'applicazione web, consentendo un'interazione flessibile e accessibile

3. Database Server (MySQL):

- **Descrizione:** Il Database Server gestisce la persistenza dei dati del sistema utilizzando il sistema di gestione di database relazionali MySQL.
- **Motivazione:** L'uso di un database relazionale come MySQL offre una solida base per la memorizzazione e la gestione dei dati del sistema.

4. OTP Software:

- **Descrizione:** L'OTP Software gestisce la creazione e l'invio delle password OTP agli admin
- **Motivazione:** L'uso di un software di questo tipo ha come fine quello di poter accertare l'identità degli admin in fase di registrazione

5. Comunicazione HTTP tra Dispositivi Utente e Web Server:

- **Descrizione:** Rappresenta la comunicazione diretta tra i Dispositivi Utente e il Web Server attraverso il protocollo HTTP.
- **Motivazione:** L'HTTP è utilizzato per consentire la trasmissione di richieste e risposte tra i dispositivi utente e il web server, supportando l'interazione utente.

6. Connessione JDBC tra Web Server e Database Server:

- **Descrizione:** Indica la connessione tra il Web Server e il Database Server utilizzando il protocollo JDBC.
- **Motivazione:** La connessione JDBC facilita l'accesso ai dati nel database MySQL, consentendo al Web Server di eseguire operazioni di lettura e scrittura.

7. Connessione socket tra OTP Software e Web Server:

- **Descrizione:** Indica il tipo di connessione tra l'OTP Software ed il Web Server utilizzando una socket connection di tipo client-server
- **Motivazione:** Consente lo scambio affidabile di dati tra client(Web Server) e server(OTP Software) attraverso un'interfaccia standardizzata

3.4. Gestione dati persistenti

Per la memorizzazione dei dati il sistema utilizza un database MySQL in quanto, nonostante imponga una rigidità maggiore nella struttura, ci permette a differenza di un database NoSQL che è principalmente adatto per scenari in cui la flessibilità dello schema è prioritaria o dove è richiesta una scalabilità orizzontale, di porre il focus su struttura e coerenza dei dati che in questo contesto è essenziale. Inoltre, l'uso di un database offre un livello di astrazione e gestione dei dati che i file non forniscono. MySQL semplifica la gestione, la query e la protezione dei dati rispetto alla memorizzazione diretta su file. In particolare, i motivi che ci hanno spinto all'uso di un database MySQL sono:

1. Struttura Relazionale:

- **Motivazione:** MySQL offre una struttura relazionale che permette di organizzare i dati in tabelle e definire relazioni tra di esse. Questa struttura è ideale per applicazioni complesse in cui le relazioni tra i dati sono fondamentali.

2. Flessibilità del Linguaggio SQL:

- **Motivazione:** L'uso di SQL consente una vasta gamma di operazioni, da query di base a operazioni più complesse come join e transazioni. Ciò offre flessibilità nello sviluppo e nella gestione dei dati.

3. Affidabilità e Robustezza:

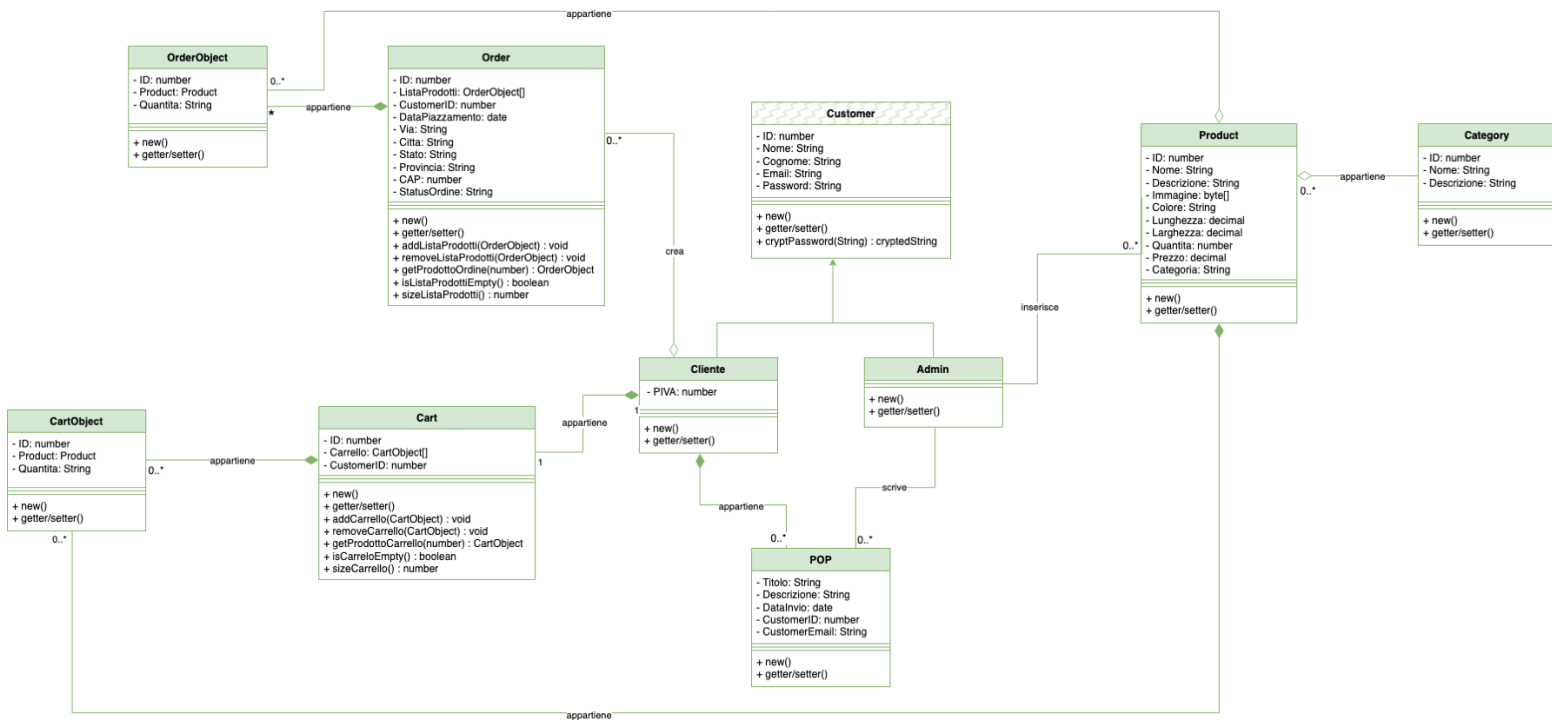
- **Motivazione:** MySQL è noto per la sua affidabilità e robustezza. È stato ampiamente testato e utilizzato in molte applicazioni crittografiche e ad alte prestazioni.

4. Sicurezza:

- **Motivazione:** MySQL fornisce funzionalità di sicurezza avanzate, compresa la crittografia dei dati e i meccanismi di autenticazione. Ciò è cruciale per proteggere i dati sensibili.

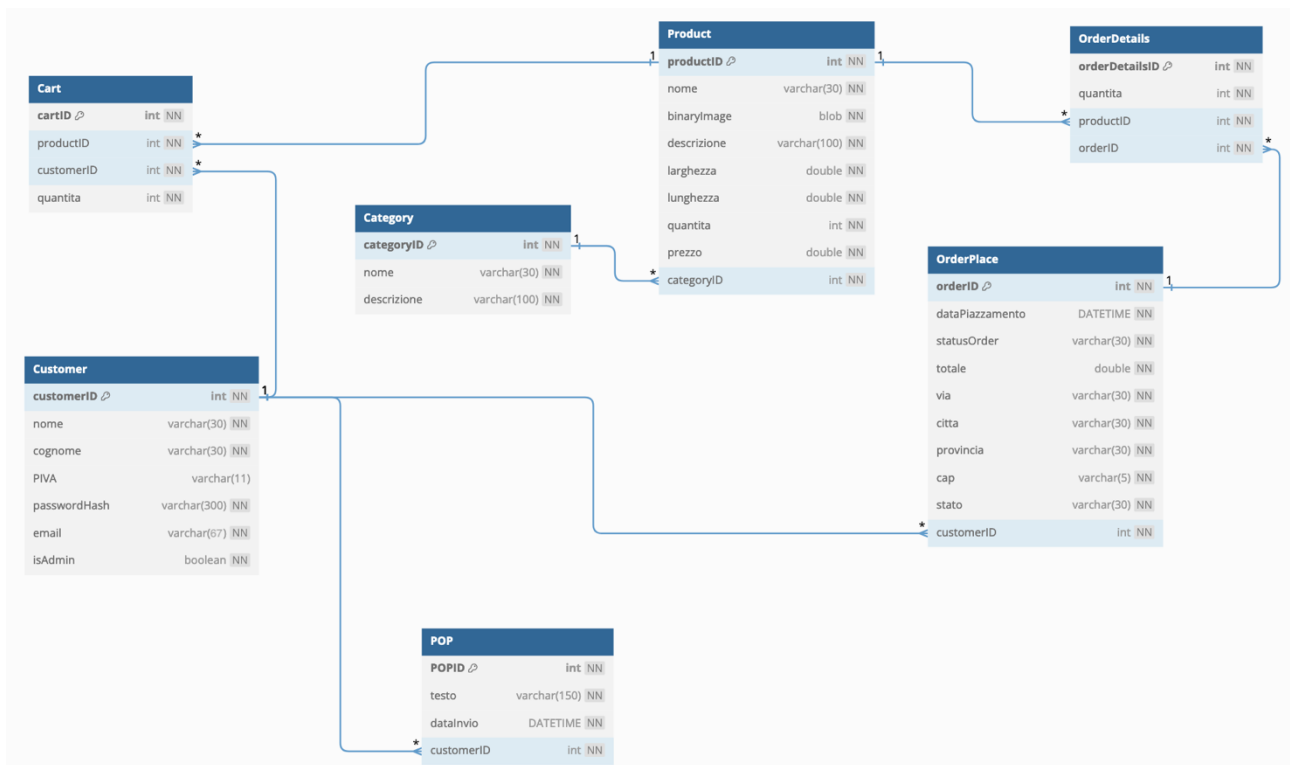
CD_SDD: Entity Class Diagram

L'entity class diagram non presenta variazioni rispetto a quello presentato nei precedenti documenti



DBMS_SDD: Database

Il Database Diagram è stato generato tramite un tool di conversione del linguaggio MySQL
Documentazione: <https://dbml.dbdiagram.io/docs/>





3.5. Controllo degli accessi e sicurezza

Oggetti \ Attori	Cliente	Admin
Autenticazione e Gestione Utenti	Registrazione standard Login Logout	Registrazione OTP Login Logout
Gestione delle Notifiche	Visualizza notifiche POP ricevute	Invia notifica POP Visualizza notifiche POP inviate Invia notifiche POP broadcast
Gestione degli Ordini	Lista acquisti	Lista ordini Modifica ordini
Gestione del Carrello	Aggiunta al carrello Rimozione dal carrello Modifica quantità Visualizza carrello	
Gestione delle Inserzioni		Visualizza annunci attivi Inserisci annuncio Modifica annuncio Elimina annuncio
Ricerca e Visualizzazione dei Prodotti	Ricerca generica sottoprodotti Ricerca filtrata sottoprodotti Ricerca sottoprodotti per Key Visualizza sottoprodotto	Ricerca generica sottoprodotti Ricerca filtrata sottoprodotti Ricerca sottoprodotti per Key Visualizza sottoprodotto
Piazzamento degli Ordini	Aggiunta indirizzo Piazzamento ordine	



3.6. Controllo del flusso globale del sistema

In un sistema di questo tipo la gestione del flusso di controllo e quindi della sequenza di azioni che avvengono all'interno del sistema è fondamentale in quanto è fondamentale definire il modo in cui il sistema prende decisioni su quali operazioni compiere e in che ordine. Nel caso specifico del nostro sistema, ogni decisione deve essere presa in seguito all'interazione dell'utente (cliente o Admin) con il sistema tramite interfaccia grafica. Difatti specifiche interazioni avviano specifiche funzionalità. Il sistema in particolare segue il modello di controllo del flusso "Procedure-driven control", in cui ogni operazione rimane in attesa di un input da parte dell'utente.

Quando un utente interagisce con il sistema, il sistema risponde selezionando il controllo corrispondente. Questo processo di selezione del controllo è il punto in cui il flusso del sistema si sposta verso lo strato logico contenente i moduli software. Ogni modulo fornisce servizi specifici, come l'autenticazione utente, la gestione degli ordini, e la gestione delle inserzioni. All'interno di questo strato logico, si attuano procedure di business specifiche in risposta ai comandi utente. Ad esempio, se un utente aggiunge un prodotto al carrello, si attiva la procedura di gestione del carrello. Queste procedure sono cruciali per mantenere un flusso coerente e strutturato nel sistema. Per accedere e manipolare i dati, il sistema si collega allo strato di storage che rappresentato dal database MySQL. Le procedure di business eseguono operazioni come l'aggiornamento delle informazioni del carrello o la registrazione di nuovi ordini, garantendo che i dati siano gestiti in modo efficiente e accurato. Infine, il sistema ritorna al livello di interazione utente, restituendo risposte attraverso l'interfaccia grafica. Ciò può includere l'aggiornamento della visualizzazione dei prodotti nel carrello o la conferma dell'avvenuto piazzamento di un ordine.

3.7. Condizioni Boundary

Definiamo ora il modo in cui il sistema deve comportarsi nel caso in cui si verifichino degli errori, in particolare errori nel tentativo di accedere a dati persistenti nel tentativo di comunicare con il software OTP esterno.

Errore nel recupero dei dati

Identificativo UCBC.1	Errore nel recupero dei dati	<i>Data</i>	03/11/2023
		<i>Vers.</i>	1.0
		<i>Autore</i>	Gurrali Giovanni
Descrizione	Caso che descrive il particolare evento in cui il sistema non riesce a richiamare i dati dal database		
Attore Principale	Sistema		
Attori secondari	NA		
Entry Condition	Il sistema non riesce a recuperare i dati immagazzinati nel database		
Exit condition On success	Il sistema visualizza correttamente i dati		
Exit condition On failure	Il sistema non riesce a riprendere il corretto funzionamento		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Sistema:	Il sistema reindirizza l'utente alla home page del sito	
2	Sistema:	Il sistema mostra all'utente una notifica contenente l'errore riscontrato	

Errore nel raggiungimento Software Esterno

Identificativo UCBC.2	Errore nel raggiungimento del Software Esterno	<i>Data</i>	03/11/2023
		<i>Vers.</i>	1.0
		<i>Autore</i>	Ruggiero Andrea
Descrizione	Caso che descrive il particolare evento in cui nel tentativo di stabilire una connessione con il software esterno questo risulta irraggiungibile		
Attore Principale	Sistema		
Attori secondari	NA		
Entry Condition	Il sistema tenta di raggiungere il software esterno per la verifica dell'OTP inserito dall'utente nella pratica di registrazione OTP		
Exit condition On success	Il sistema riesce entro i termini di timeout a comunicare con il software esterno		
Exit condition On failure	Il sistema non riesce a comunicarvi entro i tempi limite si verifica un timeout		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Sistema:	Il sistema reindirizza l'utente alla homepage del sito	
2	Sistema:	Il sistema mostra all'utente una notifica contenente l'errore riscontrato	

4. Servizi dei sottosistemi

Descriviamo in questa sezione ogni servizio offerto dai sottosistemi definiti in precedenza

Autenticazione e Gestione Utenti

Servizio	Descrizione	Interfaccia
Registrazione OTP	Permette ad un admin di registrarsi al sistema mediante codice randomico OTP (generato da software esterno) e credenziali personali	AGUDAO
Registrazione standard	Permette ad un cliente di registrarsi alla piattaforma mediante credenziali personali	AGUDAO
Login	Permette ad un cliente/admin di accedere alla piattaforma mediante proprie credenziali	AGUDAO
Logout	Permette ad un admin o ad un cliente di effettuare il logout dal sistema	AGUDAO

Gestione delle Notifiche

Servizio	Descrizione	Interfaccia
Invia notifica POP	Questa funzione permette ad un admin di inviare ad un determinato cliente, tramite email, una notifica POP	GNDAO
Visualizza notifiche POP ricevute	Questa funzione permetta ad un cliente di visionare le notifiche POP ricevute	GNDAO
Visualizza notifiche POP inviate	Questa funzione permetta ad un admin di visionare tutte le notifiche POP inviate dagli admin	GNDAO
Invia notifica POP broadcast	Questa funzione permette ad un admin di inviare a tutti i clienti del sito una notifica POP	GNDAO



Gestione degli Ordini

Servizio	Descrizione	Interfaccia
Lista ordini	Permette ad un admin di visualizzare in ordine di tempo, dal più recente al meno recente, tutti gli ordini	GODAO
Modifica ordini	Permetta ad un admin di modificare lo stato degli ordini che ricerca	GODAO
Lista acquisti	Permette ad un cliente di visualizzare in ordine di tempo, dal più recente al meno recente, tutti gli acquisti da esso effettuati	GODAO

Gestione del Carrello

Servizio	Descrizione	Interfaccia
Aggiunta al carrello	Questa funzione permette ad un cliente di inserire una data quantità di un prodotto all'interno del proprio carrello	GCD AO
Rimozione dal carrello	Questa funzione permette ad un cliente di rimuovere totalmente un prodotto all'interno del proprio carrello	GCD AO
Modifica quantità	Questa funzione permette ad un cliente di diminuire la quantità di un prodotto all'interno del proprio carrello	GCD AO
Visualizza carrello	Questa funzione permette di visionare il personale carrello di ogni utente cliccando sull'apposito pulsante	GCD AO



Gestione delle Inserzioni

Servizio	Descrizione	Interfaccia
Visualizza annunci attivi	L'admin può accedere ad una pagina contenente tutti gli annunci attualmente attivi sul sito	GIDAO
Inserisci annuncio	L'admin, dalla pagina contenente tutti gli annunci attivi sul sito, deve poter inserire un nuovo sottoprodotto da vendere utilizzando una sezione ad esso dedicata	GIDAO
Modifica annuncio	L'admin, dalla pagina contenente tutti gli annunci attivi sul sito, deve poter modificare un'inserzione precedentemente inserita, cambiandone i parametri di vendita	GIDAO
Elimina annuncio	L'admin, dalla pagina contenente tutti gli annunci attivi sul sito, per qualsiasi motivo deve poter scegliere di eliminare dal sistema l'inserzione relativa ad un particolare sottoprodotto	GIDAO



Ricerca e Visualizzazione dei Prodotti

Servizio	Descrizione	Interfaccia
Ricerca generica sottoprodotti	Questa funzione permette ad un cliente e ad admin di visualizzare una lista contenente tutti i sottoprodotti presenti nel database	RVPDAO
Ricerca filtrata sottoprodotti	Questa funzione permette ad un cliente e ad admin di visualizzare una lista contenente tutti i sottoprodotti presenti nel database, filtrata per specifici criteri di ricerca	RVPDAO
Ricerca sottoprodotti per keyword	Questa funzione permette ad un cliente e ad admin di visualizzare una lista contenente tutti i sottoprodotti presenti nel database, in base ad a specifica keyword inserita nella barra di ricerca	RVPDAO
Visualizza sottoprodotto	Questa funzione permette ad un cliente e ad un admin di visualizzare nello specifico una pagina dedicata ad un sottoprodotto	RVPDAO

Piazzamento degli ordini

Servizio	Descrizione	Interfaccia
Aggiunta indirizzo	Questa funzione permette ad un cliente di inserire il proprio indirizzo di spedizione	PODAO
Piazzamento ordine	Questa funzione permette al cliente di concludere l'acquisto	PODAO



5. Glossario

Sigla/Termine	Definizione
Design goal	Dichiarazioni chiare e misurabili che delineano ciò che un sistema software dovrebbe raggiungere. Sono le linee guida che guidano il processo di progettazione e contribuiscono a definire i requisiti del sistema.
Trade-off	Bilanciamento di vari fattori come prestazioni, costo, facilità di manutenzione e altri, la cui ottimizzazione di uno potrebbe influenzare gli altri
Interface layer	Lo strato di interfaccia è la parte di un sistema software che gestisce le interazioni con altri sistemi o componenti. Include le interfacce utente, i servizi web o qualsiasi meccanismo attraverso il quale il sistema comunica con il mondo esterno
Application logic layer	Questo strato rappresenta la logica principale del software, dove vengono gestiti i processi e le funzioni specifiche dell'applicazione. Include algoritmi, regole di business e altre funzionalità centrali del sistema.
Storage layer	Questo strato si occupa della gestione dei dati e della memorizzazione. Include database, file system o altri meccanismi utilizzati per conservare e recuperare informazioni.
Component diagram	Un diagramma che mostra l'architettura di un sistema suddividendo il software in componenti distinti e mostrando le relazioni e le dipendenze tra di essi.
Diagramma architetturale	Un diagramma che rappresenta l'architettura complessiva di un sistema software, evidenziando le relazioni tra i vari componenti e strati.
Deployment Diagram	Questo diagramma illustra come i componenti del sistema sono distribuiti fisicamente o logicamente in un ambiente di esecuzione. Mostra i nodi hardware o software e le connessioni tra di essi
DBMS_SDD: Database	Descrive come il sistema interagisce con il database. Include la struttura del database.