



UNIVERSITÀ DEGLI STUDI DI SALERNO

Corso di Laurea in Informatica, prof A.De Lucia,

a.a 2022/2023

Progetto di Ingegneria del software

Object Design Document

Partecipanti	Matricola
Antonio Michele Russo	0512109078

Revision History

Data	Versione	Descrizione	Autore
10/03/2023	1.0	Prima stesura del documento	Antonio Michele Russo
15/03/2023	1.1	Varie Modifiche	Antonio Michele Russo

Indice dei contrnuti

1.Introduzione.....	3
2.Packages.....	5
3.Class Interfaces.....	10

1. Introduzione

Dopo la realizzazione del RAD e del System Design Document, abbiamo descritto i nostri obiettivi tralasciando gli aspetti implementativi.

Questo documento in particolare ha come obiettivo quello di definire le interfacce delle classi, le operazioni, i tipi, gli argomenti, e tutto ciò che è stato definito nel System Design Document.

Vengono specificati anche i trade-off e le linee guida.

1.1 Object Design Trade-offs

- **Comprensibilità e Tempo:**

codice deve essere il più possibile comprensibile per facilitare le fasi successive, per effettuare questo step, verranno inseriti commenti per semplificare la comprensione, questo però porterà ad un aumento di tempo nello sviluppo del progetto.

- **Interfaccia ed Usabilità:**

L'interfaccia grafica è stata realizzata in modo semplice, fornendo menù chiari, e concisi, attraverso l'uso di pulsanti e form che consentono operazioni comuni come il login e la registrazione, in questo modo si favorisce l'usabilità per tutti i tipi di utenti.

- **Sicurezza ed Efficienza:**

La sicurezza viene trattata come elemento importante, il sistema implementa un sistema di crittografia di tipo sha1 per le password degli utenti.

- **Response Time e Hardware**

Sistema garantisce reattività alle richieste e contemporaneamente riesce a gestire una moltitudine di servizi agli utenti.

Questa caratteristica sarà limitata dall'hardware del sistema.

- **Prestazioni e Costi**

Sistema prevede l'uso di stili semplici, senza usare librerie esterne a pagamento data l'indisponibilità di Budget, si mantengono comunque sempre prestazioni elevate.

1.2 Linee guida per la documentazione delle interfacce

Sviluppatori dovranno seguire le seguenti convenzioni:

NAMING CONVENTION

Buona norma utilizzare nomenclature che rispettino le seguenti caratteristiche:

- Descrittivi
- Uso Comune
- Lunghezza Medio-Corta

VARIABILI

Nomi delle variabili possono iniziare con una lettera minuscola o maiuscola, se è costituito da più parole, solo l'iniziale dell'altra parola dovrà avere lettera maiuscola, per gli spazi è possibile usare il trattino basso.

METODI

I nomi dei metodi devono iniziare sempre con la lettera maiuscola, quelle dopo con la lettera minuscola.

I nomi dei metodi per l'accesso e la modifica dovranno essere `getNomeVariabile()` e `setNomeVariabile()`.

CLASSI E PAGINE

I nomi delle classi e delle pagine devono iniziare con una lettera maiuscola, il nome deve contenere informazioni relative allo scopo.

Ogni file sorgente contiene una singola classe e deve essere strutturato in un determinato modo:

- Descrizione della Classe
- Codice effettivo del file

1.3 Definizioni, Acronimi, e abbreviazioni

2.Packages

Il sistema presenta una suddivisione basata su tre livelli(three-tier):

- Presentation Layer
- Application Logic Layer
- Storage Layer

Presentation Layer	Rappresenta l'interfaccia del sistema, offre all'utente la possibilità di interagire con quest'ultimo dando la possibilità sia di inviare, in input, che di visualizzare in output i dati
Application Logic Layer	Ha il compito di elaborare i dati da inviare al client, e grazie a delle richieste fatte al database, tramite Storage Layer, accede ai dati persistenti. Si occupa di varie gestioni quali: <ul style="list-style-type: none">• Gestione Account• Gestione Acquisti• Gestione Carrello• Gestione Prodotti• Utils
Storage Layer	Ha il compito di memorizzare i dati del sistema, utilizzando un DBMS, inoltre riceve le varie richieste dell'Application Logic Layer inoltrandole al DBMS e restituendo i dati richiesti.

2.1 Package src

2.1.1 Package Gestione Account	
Classe	Descrizione
AdminPage	Servlet che gestisce la pagina home relativa all'accesso da admin
Amministratore	Modello che delinea le caratteristiche dell'amministratore
AmministratoreServlet	Servlet che gestisce l'accesso al sistema come Amministratore
AmmModel	Interfaccia per AmmModelDS
AmmModelDS	Classe che contiene i metodi che permettono di effettuare il retrieve, l'inserimento e l'update di un amministratore
CancellaUtente	Servlet che consente la cancellazione di un utente
LoginModelDS	Classe che contiene i metodi che permettono l'inserimento, i retrieve per nome e email e l'update di un utente
loginServlet	Servlet che permette il login di un utente
logout	Servlet che permette il logout di un utente
ModificaProfilo	Servlet che permette di modificare il profilo di un utente
passwordpersa	Servlet che consente di reimpostare la password
PwdModelDS	Classe che contiene i metodi che permettono l'update della password, e il retrieve attraverso l'email dell'utente
registrazione	Servlet che consente la registrazione
Utente	Modello che delinea le caratteristiche dell'utente
utenteModel	Interfaccia per LoginModelDS

2.1.2 Package Gestione Acquisti

Classe	Descrizione
Ordine	Modello che delinea le caratteristiche di un ordine
OrdineModel	Classe che contiene i metodi per il retrieve, inserimento e update dello stato di un ordine
OrdineModelDs	Interfaccia di supporto a OrdineModel
Ordiniclienti	Servlet che consente di recuperare gli ordini dei clienti
Pagamento	Modello che delinea le caratteristiche di un pagamento
PagamentoModel	Classe che contiene i metodi per il retrieve e l'inserimento di un pagamento
PagamentoModelDS	Interfaccia a supporto di PagamentoModel
Spedizione	Modello che delinea le caratteristiche di una spedizione
SpedizioneModel	Classe che contiene i metodi per il retrieve e l'inserimento di una spedizione
SpedizioneModelDS	Interfaccia a supporto di SpedizioneModel
totaleordine	Servlet per il totale dell'ordine

2.1.3 Package Gestione Carrello

Classe	Descrizione
Carrello	Modello che delinea le caratteristiche di un carrello
CarrelloServlet	Servlet che consente di gestire operazioni nel carrello
Composto	Modello che delinea le caratteristiche di ciò che contiene un ordine
compostoModel	Classe che contiene i metodi per retrieve e inserimento
compostoModelDs	Interfaccia di supporto per composto Model

2.1.4 Package Gestione Prodotti

Classe	Descrizione
CancellaProdotto	Servlet che consente la cancellazione di un prodotto
Home	Servlet che visualizza la homepage
Offerte	Servlet che visualizza prodotti in evidenza

prodotto	Modello che delinea le caratteristiche di un prodotto
ProductModel	
ProductModelDS	Classe che contiene metodi per il retrieve, l'inserimento e l'update di un prodotto
Ricerca	Servlet che consente la ricerca nel sito
Scegliprodotto	Servlet che consente la scelta di un prodotto
Visualizzaprodotto	Servlet che consente la visualizzazione di un prodotto in dettaglio

2.2 Front-End Packages

Utils	
Classe	Descrizione
MainContext	Gestisce il contesto principale
Sha1	Gestisce la crittografia dei dati

/WebContent/	
JSP	Descrizione
Carrello.jsp	View che mostra componente carrello
Chisiamo.jsp	View che mostra componente pagina "ChiSiamo"
Confermaordine.jsp	View che mostra componente della conferma ordine
Contatti.jsp	View che mostra componente pagina Contatti
ErroreAggProdotti.jsp	View che mostra pagina di Errore di aggiunta prodotti
Footer.jsp	View che mostra componente footer
Header.jsp	View che mostra componente Header
Homepage.jsp	View che mostra componente Homepage
Login.jsp	View che mostra componente Login
Ordini.jsp	View che mostra componente Ordini
Prodotti.jsp	View che mostra componente Prodotti

PwdDimenticata.jsp	View che mostra componente PasswordDimenticata
Registrazione.jsp	View che mostra componente Registrazione

/WebContent/prodotti

JSP	Descrizione
VisualizzaProdotto	View che mostra pagina Prodottto

/WebContent/utente

JSP	Descrizione
Profilo	View che mostra pagina profilo utente

3.Class Interface

3.1 Servlet

Nome Interfaccia	AmmModel
Descrizione	Interfaccia per AmmModelDS Usata per tutto quello che riguarda l'amministratore, la possibilità di trovarlo attraverso il nome, dall'email di inserirne uno, di aggiornare i dati e di cancellarlo
Signature Dei Metodi	+doRetrieveByNome(String rags):T +doRetrieveByKey(String email):Amministratore +doRetrieveAll(): Collection<T> +doSave(T item):boolean +doUpdate(T item):void +doDelete(String item):boolean
Pre-Condizioni	
Post-Condizioni	
Invariante	

Nome Interfaccia	utenteModel
Descrizione	Interfaccia per LoginModelDS, permette di ritrovare un utente attraverso l'email o attraverso il nome, inserirlo, effettuare un update o cancellarlo
Signature Dei Metodi	+doRetrieveByKey(String email):Utente +doRetrieveByNome(String rags):T +doRetrieveAll(): Collection<T> +doSave(T item):boolean +doUpdate(T item):void +doDelete(String item):boolean
Pre-Condizioni	
Post-Condizioni	
Invariante	

Nome Interfaccia	OrdineModelDS
Descrizione	Interfaccia per OrdineModel, serve per ritrovare gli ordini per utente, o per ritrovare

	tutti gli ordini, include la possibilità di salvare, aggiornare o eliminare un ordine
Signature Dei Metodi	+doRetrieveAll(): Collection<T> +doRetrieveByCodice_Fiscale(): Collection<T> +doSave(T item):boolean +doUpdate(T item):void +doDelete(int num):boolean
Pre-Condizioni	
Post-Condizioni	
Invariante	

Nome Interfaccia	PagamentoModelDS
Descrizione	Interfaccia per PagamentoModel, consente di inserire,ritrovareperid, aggiornare o cancellare un pagamento
Signature Dei Metodi	+doRetrieveById(String code):boolean; +doSave(T item):boolean +doUpdate(T item): void +doDelete(T item): void
Pre-Condizioni	
Post-Condizioni	
Invariante	

Nome Interfaccia	SpedizioneModelDS
Descrizione	Interfaccia a supporto di spedizioneModel, contiene riferimenti per l'inserimento, aggiornamento e cancellazione di un ordine
Signature Dei Metodi	+doRetrieveByNumero(String code):boolean;

	+doSave(T item):int +doUpdate(T item): void +doDelete(T item): boolean
Pre-Condizioni	
Post-Condizioni	
Invariante	

Nome Interfaccia	compostoModel
Descrizione	Interfaccia che gestisce il retrieve di cui è composto un ordine, o ne permette il salvataggio/update/delete
Signature Dei Metodi	+doRetrieveByP.Iva (int code):collection <T>; +doSave(T item):void +doUpdate(T item): void +doDelete(T item): boolean
Pre-Condizioni	
Post-Condizioni	
Invariante	

Nome Interfaccia	ProductModel
Descrizione	Interfaccia che gestisce la parte relativa ai Prodotti, inserimento, retrieveperKey, SSN , prodotto, Parametro, scelta, nome Cancellazione, update

Signature Dei Metodi	+doRetrieveByKey (int code):collection <T>; +doRetrieveBySSN (String code):T; +doRetrieveByProdotto (String code):T; +doRetrieveByParamentro (String code):Collection<T>; +doRetrieveByScelta (String code):T; +doRetrieveByNome (String code):T; +doRetrieveAll (String code):Collection<T>; +doSave(T item):void +doUpdate(T item): void +doDelete(T item): boolean
Pre-Condizioni	
Post-Condizioni	
Invariante	

