

3GTech  
Object Design Document  
Versione 1



Data: 20/12/2022

# 1. Introduzione

## 1.1.Object Design Trade-Offs

### Interfaccia VS. Usabilità

Lo sviluppo delle interfacce vuole puntare ad un risultato che prediliga l'usabilità. Le interfacce sono quindi intuitive e facilmente utilizzabili per la totalità dei consumatori. Per realizzare questo aspetto, si è voluta utilizzare una grafica semplice, a discapito di un layout appariscente ma più complesso per alcune fasce di utenti.

### Interfaccia VS. Tempi di risposta

Si vuole puntare ad un software che offra tempi di risposta rapidi, pertanto si è deciso di realizzare un'interfaccia che non includa molte animazioni ed altri elementi grafici che appesantiscono la pagina.

### Sicurezza VS. Costi

Il sistema di accesso al sito si basa su inserimento di e-mail e password, senza l'impiego di sistemi di protezione più evoluti a causa di tempi e costi di sviluppo. Il meccanismo di protezione adottato si basa quindi sul controllo di ogni input inserito dagli utenti, in modo da evitare problematiche come ad esempio l'SQL Injection. Il controllo verrà sviluppato principalmente in front-end, mentre per quanto riguarda le credenziali degli utenti verrà svolto in back-end.

### Efficienza VS. Costi

Siccome il progetto non presenta alcun budget, si è deciso di affidarsi alle prestazioni offerte da componenti e librerie open-source. Pur ponendo un limite a questo aspetto, il sistema risulta essere efficiente e rispetta i requisiti imposti.

## 1.2.Interface Documentation Guidelines

- i) I metodi sono definiti con l'utilizzo di verbi, mentre variabili e nomi con sostantivi.
- ii) I nomi utilizzati per classi, variabili e parametri devono essere descrittivi in modo da non creare perplessità.
- iii) I nomi dei package devono essere intuitivi, utilizzando termini che descrivano le classi contenute. Ogni parola dovrà essere separata dal punto '.'.
- iv) I file riguardanti una determinata interfaccia (.css, .js, .jsp, ...) dovranno essere contenuti in una cartella avente il nome dell'interfaccia in questione. I file contenuti avranno tutti lo stesso nome della cartella, seguito dall'estensione del file.

## 2. Packages

3GTech

src/main/java/Control	In questo percorso si trovano tutti i package che si occupa della parte logica del sito 3GTech	Ajax	AjaxAggiungiAlCarrelloServlet	Gestisce il pulsante aggiungi al carrello che si trova sulla pagina prodotto.jsp
		Carrello	GestioneCarrello	Contiene tutte le funzioni per la gestione del carrello
		Database	Connettore	Si occupa di gestire la connessione con il database
		Query	Esegue delle query standard senza risultato	
		Dispatcher	ComputerServlet	Reindirizza alla pagina scelta
			ConsoleServlet	Reindirizza alla pagina scelta
			DispatcherAcquistaCarrello	Reindirizza alla pagina scelta
			DispatcherCarrelloServlet	Reindirizza alla pagina scelta
			DispatcherConfermaPagamento	Reindirizza alla pagina scelta
			DispatcherElenco	Reindirizza alla pagina scelta
			DispatcherGestisciCatalogo	Reindirizza alla pagina scelta
			DispatcherImpostazioni	Reindirizza alla pagina scelta
			DispatcherIndexAltroServlet	Reindirizza alla pagina scelta
			DispatcherIndexServlet	Reindirizza alla pagina scelta
			DispatcherLoginServlet	Reindirizza alla pagina scelta
			DispatcherPagamento	Reindirizza alla pagina scelta
			DispatcherProdottoSearchBar	Reindirizza alla pagina scelta
			DispatcherProdottoServlet	Reindirizza alla pagina scelta
			DispatcherRegistrazioneServlet	Reindirizza alla pagina scelta
			DispatcherStoricoAcquistiFattura	Reindirizza alla pagina scelta
			DispatcherSvuotaCarrello	Reindirizza alla pagina scelta
			DispatcherWhereServlet	Reindirizza alla pagina scelta
			ElettrodomesticiServlet	Reindirizza alla pagina scelta
			LogoutServlet	Reindirizza alla pagina scelta
			OrdiniData	Reindirizza alla pagina scelta
			OrdiniUtente	Reindirizza alla pagina scelta
			VideogameServlet	Reindirizza alla pagina scelta
		Servlet	InsensiciProdotto	Esegue il comando scelto dall'utente
			LoginServlet	Esegue il comando scelto dall'utente
			ModificaEmail	Esegue il comando scelto dall'utente
			ModificaIndirizzo	Esegue il comando scelto dall'utente
			ModificaPassword	Esegue il comando scelto dall'utente
			ModificaProdotto	Esegue il comando scelto dall'utente
			RecensioneServlet	Esegue il comando scelto dall'utente
			RegistrazioneServlet	Esegue il comando scelto dall'utente
		Sessione	RimuoviProdotto	Esegue il comando scelto dall'utente
			Session	Gestisce la sessione con l'utente, salvando alcuni dati all'interno

src/main/java/Model	In questo percorso si trovano tutti i package che forniscono i metodi per accedere ai dati del sito 3GTech	Bean	CartaBean	Oggetto carta
			OrdineBean	Oggetto ordine
			PossiedeBean	Oggetto possiede relazione con carta e utente
			ProdottoBean	Oggetto prodotto
			RecensioneBean	Oggetto recensione
			TelefonoBean	Oggetto telefono
			UtenteBean	Oggetto utente
		Dao	CartaDao	Interfaccia con tutte le funzioni che devono essere implementate
			OrdineDao	Interfaccia con tutte le funzioni che devono essere implementate
			PossiedeDao	Interfaccia con tutte le funzioni che devono essere implementate
			ProdottoDao	Interfaccia con tutte le funzioni che devono essere implementate
			RecensioneDao	Interfaccia con tutte le funzioni che devono essere implementate
			TelefonoDao	Interfaccia con tutte le funzioni che devono essere implementate
			UtenteDao	Interfaccia con tutte le funzioni che devono essere implementate
		Implementazione	CartaImplementazione	Implementa le funzioni del dao interfacciandosi con il database
			OrdineImplementazione	Implementa le funzioni del dao interfacciandosi con il database
			PossiedeImplementazione	Implementa le funzioni del dao interfacciandosi con il database
			ProdottoImplementazione	Implementa le funzioni del dao interfacciandosi con il database
			RecensioneImplementazione	Implementa le funzioni del dao interfacciandosi con il database
			TelefonoImplementazione	
			Implementa le funzioni del dao interfacciandosi con il database	
			UtenteImplementazione	Implementa le funzioni del dao interfacciandosi con il database

src\WebContent - view	In questo percorso si trovano tutti i package che si occupa della grafica del sito 3GTech	carrello	Contiene il css della pagina
		Gestione	Contiene il css della pagina
		Header_footer	Contiene il css della pagina
		ImageProdotto	Contiene tutte le immagini del prodotto. In questa cartella ci sono altre cartelle con numeri incrementali che corrispondono all'id del prodotto sul db
		Images	Contiene le immagini della pagina
		Impostazioni	Contiene il css della pagina
		Index	Contiene il css della pagina
		Login	Contiene il css della pagina
		Meta-inf	Contiene il manifesto
		Prodotto	Contiene il css della pagina
		Registrazione	Contiene il css della pagina
		Web-inf	lib
			Carrello.jsp
			Elenco.jsp
			ElencoOrdiniAdmin.jsp
			Errori.jsp
			Fattura.jsp
			Fattura2.jsp
			Footer.jsp
			GestisciCatalogo.jsp
			Header.jsp
			Impostazioni.jsp
			Index.jsp
			Login.jsp
			Navbar.jsp
			Pagamento.jsp
			Prodotto.jsp
			Registrazione.jsp
			Where.jsp

### 3. Class Interfaces

#### 3.1.Package control.carrello

GestioneCarrello	
Metodo	+aggiungiElemento(HttpServletRequest r, short codice, int quantità) : boolean
Descrizione	Aggiunge un elemento al carrello di un utente.
Pre-Conditions	null
Post-Conditions	leggi(request)!=null
Metodo	+incrementaQuantitaDiUno(HttpServletRequest r, short codice) : boolean
Descrizione	Incrementa di uno la quantità di un prodotto già presente nel carrello.
Pre-Conditions	null
Post-Conditions	leggi(request)!=null
Metodo	+leggi(HttpServletRequest r) : JSONArray
Descrizione	Restituisce un oggetto contenente tutti i prodotti presenti nel carrello.
Pre-Conditions	request.getSession() != null
Post-Conditions	null

### 3.2. Package control.database

Connettore	
<i>-con: Connection</i>	
Metodo	+openConnection(String email, String password) : boolean
Descrizione	Instaura una connessione con il Database.
Pre-Conditions	email e password corrette
Post-Conditions	getConnection()!=null
Metodo	+getConnection() : Connection
Descrizione	Ritorna la connessione (con)
Pre-Conditions	null
Post-Conditions	null
Metodo	+isConnected() : boolean
Descrizione	Verifica se è stata instaurata una connessione col database.
Pre-Conditions	null
Post-Conditions	null
Metodo	+closeConnection()
Descrizione	Termina la connessione, se è stata instaurata.
Pre-Conditions	null
Post-Conditions	getConnection()==null
Query	
Metodo	+updateQuery(String query, String username, String password) : boolean
Descrizione	Esegue la query passata come parametro.
Pre-Conditions	Username e password corretti.
Post-Conditions	Query eseguita
Metodo	+controllaAndUpdateQueryProdotto(Short codice, String query, String email, String password) : Connection
Descrizione	Verifica che un oggetto Prodotto esista nel database ed esegue una query su di esso.
Pre-Conditions	Email e password sono corretti.
Post-Conditions	null
Metodo	+controllaAndUpdateQueryProdotto(String emailUtente, String query, String email, String password) : Connection
Descrizione	Verifica che un oggetto Utente esista nel database ed esegue una query su di esso.
Pre-Conditions	Email e password sono corretti.
Post-Conditions	null
Metodo	+controllaAndUpdateQueryOrdine(int id, String query, String email, String password)
Descrizione	Verifica che un oggetto Ordine esista nel database ed esegue una query su di esso.
Pre-Conditions	Email e password corretti.

Post-Conditions	null
-----------------	------

### 3.3.Package control.sessione

Session	
<i>-session: HttpSession</i>	
Metodo	+ getUtenteSessionLogin(HttpServletRequest request) : String
Descrizione	Recupera l'email dell'utente dalla sessione
Pre-Conditions	L'utente ha effettuato la login precedentemente
Post-Conditions	null
Metodo	+ setUtenteSessionLogin(HttpServletRequest request, UtenteBean utente): boolean
Descrizione	Viene usata per effettuare la login, impostando il tempo di connessione per 1h e registrando quale tipologia di utente è tra amministratore, gestore o cliente.
Pre-Conditions	request != null && i parametri dell'utente sono corretti
Post-Conditions	Utente loggato
Metodo	+ azzeraCarrello(HttpServletRequest request) : boolean
Descrizione	Rimuove la variabile "carrello" dalla sessione in modo tale da resettare il carrello
Pre-Conditions	request.getSession() != null
Post-Conditions	Variabile "carrello" non esiste
Metodo	+ inserisciCarrello(HttpServletRequest request, String carrello) : boolean
Descrizione	Inserisce la variabile "carrello" nella sessione
Pre-Conditions	request.getSession() != null
Post-Conditions	Variabile "carrello" esiste
Metodo	+ leggiCarrello(HttpServletRequest request) : String
Descrizione	Legge la variabile "carrello" dalla sessione recuperando tutti gli
Pre-Conditions	Request.getSession().getAttribute("carrello") != null
Post-Conditions	null
Metodo	+ isInizializzatoCarrello(HttpServletRequest request) : boolean
Descrizione	Ritorna un valore per indicare se il carrello è inizializzato oppure no
Pre-Conditions	Request.getSession() != null
Post-Conditions	null
Metodo	+ destroySession(HttpServletRequest request) : boolean
Descrizione	Elimina la sessione facendo il logout dell'utente
Pre-Conditions	Request.getSession() != null
Post-Conditions	Request.getSession() == null
Metodo	+ getEmail(HttpServletRequest request) : String
Descrizione	Recupera l'email dell'utente dalla sessione
Pre-Conditions	Request.getSession() != null
Post-Conditions	null

### 3.4. Package model.implementazione

CartaImplementazioneDAO	
Metodo	+ aggiungiCarta(CartaBEAN carta, UtenteBean utente) : boolean
Descrizione	Inserisce una nuova carta nel database collegandola con un determinato utente
Pre-Conditions	I parametri della carta sono corretti && i parametri dell'utente sono corretti
Post-Conditions	Carta salvata per i futuri acquisti
Metodo	+ rimuoviCarta(CartaBEAN carta) : boolean
Descrizione	Rimuove la carta dal database
Pre-Conditions	Carta.getNumero != null
Post-Conditions	Carta eliminata
Metodo	+ exists(CartaBEAN carta) : boolean
Descrizione	Controlla se il numero della carta esiste all'interno del database
Pre-Conditions	Carta.getNumero != null
Post-Conditions	null
Metodo	+ pagamento(CartaBEAN carta) : boolean
Descrizione	Controlla tutti i dati della carta inseriti dal cliente con i dati salvati nel database, in modo tale che se coincidono fa procedere il cliente all'acquisto
Pre-Conditions	Carta != null
Post-Conditions	null

UtenteImplementazioneDAO	
Metodo	+ inserisciUtente(UtenteBean utente) : boolean
Descrizione	Inserisce l'utente all'interno del database.
Pre-Conditions	I parametri dell'utente sono corretti.
Post-Conditions	Utente inserito.
Metodo	+ rimuoviUtente(UtenteBean utente) : boolean
Descrizione	Rimuove l'utente dal database
Pre-Conditions	I parametri dell'utente sono corretti
Post-Conditions	Utente rimosso
Metodo	+ modificaPassword(UtenteBean utente) : boolean
Descrizione	Modifica il parametro Password dell'utente
Pre-Conditions	I parametri dell'utente sono corretti
Post-Conditions	Password modificata
Metodo	+ modificaIndirizzo(UtenteBean utente) : boolean
Descrizione	Modifica il parametro Indirizzo dell'utente
Pre-Conditions	I parametri dell'utente sono corretti

<u>Post-Conditions</u>	<u>Indirizzo modificato</u>
------------------------	-----------------------------

  

<u>Metodo</u>	<u>+ modificaEmail(UtenteBean utente) : boolean</u>
<u>Descrizione</u>	<u>Modifica il parametro Email dell'utente</u>
<u>Pre-Conditions</u>	<u>I parametri dell'utente sono corretti</u>
<u>Post-Conditions</u>	<u>Email modificata</u>

  

<u>Metodo</u>	<u>+ isAdmin(UtenteBean utente) : boolean</u>
<u>Descrizione</u>	<u>Verifica che un utente sia di tipo Gestore</u>
<u>Pre-Conditions</u>	<u>I parametri dell'utente sono corretti</u>
<u>Post-Conditions</u>	<u>null</u>

  

<u>Metodo</u>	<u>+ isCliente(UtenteBean utente) : boolean</u>
<u>Descrizione</u>	<u>Verifica che un utente sia di tipo Cliente</u>
<u>Pre-Conditions</u>	<u>I parametri dell'utente sono corretti</u>
<u>Post-Conditions</u>	<u>null</u>

  

<u>Metodo</u>	<u>+ login(UtenteBean utente) : boolean</u>
<u>Descrizione</u>	<u>Effettua il login dell'utente.</u>
<u>Pre-Conditions</u>	<u>I parametri dell'utente sono corretti.</u>
<u>Post-Conditions</u>	<u>L'utente è loggato.</u>

TelefonImplementazioneDAO	
---------------------------	--

  

<u>Metodo</u>	<u>+ aggiungiTelefono (TelefonoBEAN telefono) : boolean</u>
<u>Descrizione</u>	<u>Inserisce un nuovo telefono nel database collegandolo con un determinato utente</u>
<u>Pre-Conditions</u>	<u>telefono.getNumero != null &amp;&amp; telefono.getUtente != null</u>
<u>Post-Conditions</u>	<u>Telefono inserito nel database</u>

  

<u>Metodo</u>	<u>+ rimuoviTelefono(TelefonoBEAN telefono) : boolean</u>
<u>Descrizione</u>	<u>Rimuove il telefono dal database</u>
<u>Pre-Conditions</u>	<u>Telefono.getNumero != null</u>
<u>Post-Conditions</u>	<u>Telefon cancellato dal database</u>

  

<u>Metodo</u>	<u>+ leggiTelefono(UtenteBean utente) : JSONArray</u>
<u>Descrizione</u>	<u>Legge tutti i numeri di telefono di un determinato utente</u>
<u>Pre-Conditions</u>	<u>Utente != null</u>
<u>Post-Conditions</u>	<u>null</u>



### ProdottoImplementazioneDAO

<b>Metodo</b>	+ inserisciProdotto(ProdottoBean p) : boolean
<b>Descrizione</b>	Inserisce un prodotto all'interno del database.
<b>Pre-Conditions</b>	I parametri di p sono corretti.
<b>Post-Conditions</b>	Prodotto inserito

<b>Metodo</b>	+ rimuoviProdotto(ProdottoBean p) : boolean
<b>Descrizione</b>	Rimuove un prodotto all'interno del database.
<b>Pre-Conditions</b>	I parametri di p sono corretti.
<b>Post-Conditions</b>	Prodotto rimosso.

<b>Metodo</b>	+ modificaDescrizione(ProdottoBean p) : boolean
<b>Descrizione</b>	Modifica la descrizione di un prodotto all'interno del database.
<b>Pre-Conditions</b>	I parametri di p sono corretti.
<b>Post-Conditions</b>	Descrizione del prodotto modificata.

<b>Metodo</b>	+ modificaNome(ProdottoBean p) : boolean
<b>Descrizione</b>	Modifica il nome di un prodotto all'interno del database.
<b>Pre-Conditions</b>	I parametri di p sono corretti.
<b>Post-Conditions</b>	Nome del prodotto modificato.

<b>Metodo</b>	+ modificaPrezzo(ProdottoBean p) : boolean
<b>Descrizione</b>	Modifica il prezzo di un prodotto all'interno del database.
<b>Pre-Conditions</b>	I parametri di p sono corretti.
<b>Post-Conditions</b>	Prezzo del prodotto modificato.

<b>Metodo</b>	+ leggiProdotto() : JSONArray
<b>Descrizione</b>	Restituisce un array contenente tutti i prodotti del database.
<b>Pre-Conditions</b>	null
<b>Post-Conditions</b>	null

<b>Metodo</b>	+ ultimiInseriti() : JSONArray
<b>Descrizione</b>	Restituisce un JSONArray contenente gli ultimi 3 prodotti inseriti
<b>Pre-Conditions</b>	null
<b>Post-Conditions</b>	null

<b>Metodo</b>	+ random() : JSONArray
<b>Descrizione</b>	Restituisce un JSONArray contenente 3 prodotti casuali.
<b>Pre-Conditions</b>	null
<b>Post-Conditions</b>	null

<b>Metodo</b>	+ cerca(String s) : JSONArray
<b>Descrizione</b>	Restituisce un JSONArray contenente i prodotti che contengono la stringa passata come parametro.
<b>Pre-Conditions</b>	null

Post-Conditions	null
-----------------	------

  

Metodo	+ getProdottoByName(String name) : Short
Descrizione	Restituisce il codice univoco del prodotto con nome = name.
Pre-Conditions	Name deve essere valido nel database.
Post-Conditions	null

  

Metodo	+ exists(Short id) : boolean
Descrizione	Verifica se il prodotto avente il codice passato come parametro esiste.
Pre-Conditions	null
Post-Conditions	null

OrdineImplementazioneDAO	
-	

  

Metodo	+ inserisciOrdine(OrdineBean ordine) : int
Descrizione	Inserisce un ordine nel database
Pre-Conditions	Ordine != null
Post-Conditions	Ordine inserito nel database

  

Metodo	+ modificaDataConsegna(OrdineBean ordine) : boolean
Descrizione	Modifica la data di consegna di un singolo prodotto
Pre-Conditions	Prodotto.getId != null
Post-Conditions	Data consegna modificata

  

Metodo	+ modificaOrarioConsegna(OrdineBean ordine) : boolean
Descrizione	Modifica l'orario di consegna di un singolo prodotto
Pre-Conditions	Prodotto.getId != null
Post-Conditions	Orario consegna modificato

  

Metodo	+ leggiSingoloOrdine(OrdineBean ordine) : void
Descrizione	Legge tutti i valori di un singolo ordine
Pre-Conditions	Ordine.getId != null
Post-Conditions	null

  

Metodo	+ leggiOrdine(UtenteBean utente) : JSONArray
Descrizione	Legge tutti gli ordini effettuati da un dato utente
Pre-Conditions	Utente.getEmail != null
Post-Conditions	null

  

Metodo	+ selectOrdiniData(Date d1, Date d2) : JSONArray
Descrizione	Seleziona tutti gli ordini compresi in un determinato range
Pre-Conditions	D1 != null d2 != null
Post-Conditions	null