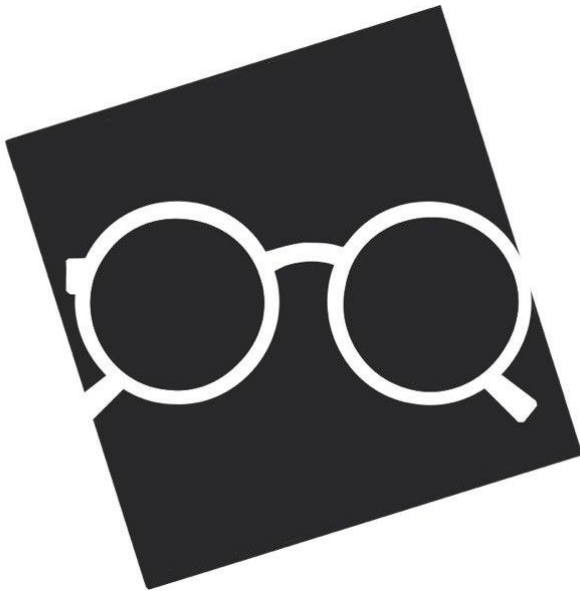


**Università degli Studi di Salerno**  
Corso di Ingegneria del Software

**The Spectacles**  
**System Design Document**  
**Versione 0.6**



**The**  
**Spectacles**

Data: 21/12/2022

Progetto: The Spectacles	Versione: 0.1
Documento: System Design Document	Data: 15/12/2022

### Coordinatore del progetto:

Nome	Matricola

### Partecipanti:

Nome	Matricola
Roberto Piscopo	0512109906
Alessandro Satta	0512110929
Mario Ranieri	0512110017
Luca Di Meglio	0512110767

<b>Scritto da:</b>	Alessandro Satta, Roberto Piscopo, Mario Ranieri, Luca Di Meglio
--------------------	--

## Revision History

Data	Versione	Descrizione	Autore
30/11/2022	0.1	Inizio documentazione del progetto	Alessandro Satta e Luca Di Meglio
14/12/2022	0.2	Aggiunte informazioni riguardo ai punti: 1.4,1.5,1.6,2,3,3.1,3.2,3.3,3.3.1,3.4,3.4.1,3.4.2	Alessandro Satta
15/12/2022	0.3	Aggiunte informazioni riguardanti ai punti: 3.5	Alessandro Satta
21/12/2022	0.4	Aggiunta informazioni 3.2	Alessandro Satta, Mario Ranieri
27/12/2022	0.5	Aggiunta informazioni riguardo i punti: 3.6, 3.7	Luca Di Meglio
13/01/2023	0.6	Aggiunta parziale informazioni punto 4	Luca Di Meglio

# Indice

1.	INTRODUZIONE.....	
1.1	Obiettivo Progetto .....	
1.2	Scopo Sistema .....	
1.3	Obiettivi e criteri di successo del progetto .....	
1.3.1	Tempo di rilascio vs Funzionalità .....	
1.3.3	Prestazioni vs Affidabilità .....	
1.4	Definizioni, acronimi e abbreviazioni.....	
1.5	Riferimenti .....	
1.6	Panoramica .....	
2.	SISTEMA CORRENTE .....	
3.	SISTEMA PROPOSTO .....	
3.1	Panoramica .....	
3.2	Decomposizione Sottosistemi .....	
3.3	mappatura Hardware/Software .....	
3.3.1	Deploy .....	
3.4	gestione dei dati persistenti .....	
3.4.1	Schema concettuale .....	
3.4.2	Schema logico .....	
3.5	Controllo degli accessi e sicurezza.....	
3.6	Controllo flusso globale del software.....	
3.7	Condizioni Limite.....	
3.7.1	Start-up .....	
3.7.2	Terminazione .....	
3.7.3	Fallimento.....	
4.	Servizi dei Sottosistemi.....	
5	Glossario .....	

# 1. INTRODUZIONE

## 1.1. *Scopo Sistema*

The Spectacles è un e-commerce finalizzato alla vendita di occhiali da sole e da vista.

L'obiettivo è quello di soddisfare i gusti più vari, permettendo di adattare qualsiasi montatura in base ad ogni esigenza.

## 1.2. *Ambito del Sistema*

Il sistema nasce per entrare a far parte del mondo degli E-commerce in ambito wearable.

Il sistema deve supportare:

- Acquisto prodotto
- Lasciare una recensione
- Aggiunta al carrello dei prodotti
- Visualizzare tipologia e categorie prodotti
- Pagare attraverso carta o alla consegna
- Visualizzare ordini

## 1.3. *Obiettivi e criteri di successo del Progetto (Design Goals)*

### 1.3.1 *Criteri di performance*

- *Tempo di risposta:* Il sito web deve fornire un tempo di risposta abbastanza basso da poter permettere una fluida navigazione a tutti gli utenti.
- *Memoria:* La memoria fornita dal DB dovrà essere scalabile per accomodare influssi di nuovi utenti e creazione di un numero qualsiasi di nuove entry.

### 1.3.2 *Criteri di affidabilità*

- *Disponibilità*

## 1.4 *Tempo di rilascio VS Funzionalità*

Al fine di avere un impatto positivo con i primi clienti è necessario che il sistema implementi tutte le funzionalità necessarie, per cui si preferisce mettere in secondo piano i tempi di rilascio per avere un e-commerce quanto più possibile funzionale e privo di bug.

## 1.5 *Prestazioni VS Affidabilità*

Elenco obiettivi:

- Dare la possibilità di consultare il sito a tutti gli utenti (ospiti compresi)
- Pagamenti rapidi e sicuri
- Selezionare i marchi con più gradimento
- Criteri di successo

## 1.6 *Definizioni, acronimi e abbreviazioni*

**RAD:** Requirements Analysis Document.

**SDD:** System Design Document.

**ODD:** Object Design Document.

**UC:** Use Case.

**RF:** Requisito funzionale.

**NRF:** Requisito non funzionale

**Admin:** Soggetto che si occupa di eliminare/aggiungere/modificare prodotti all'interno della piattaforma, può conoscere gli ordini effettuati dagli utenti e conoscere gli ordini effettuati in un intervallo di tempo.

**Utente:** Un utilizzatore della piattaforma che abbia fatto il login.

**Ospite:** Un qualsiasi utilizzatore della piattaforma che non ha effettuato il login.

### ***1.7 Riferimenti***

- Bernd Bruegge & Allen H. Dutoit, *Object-Oriented Software Engineering: Using UML, Patterns and Java*, (2nd edition), Prentice-Hall, 2003.
- Ian Sommerville, *Software Engineering*, Addison Wesley.

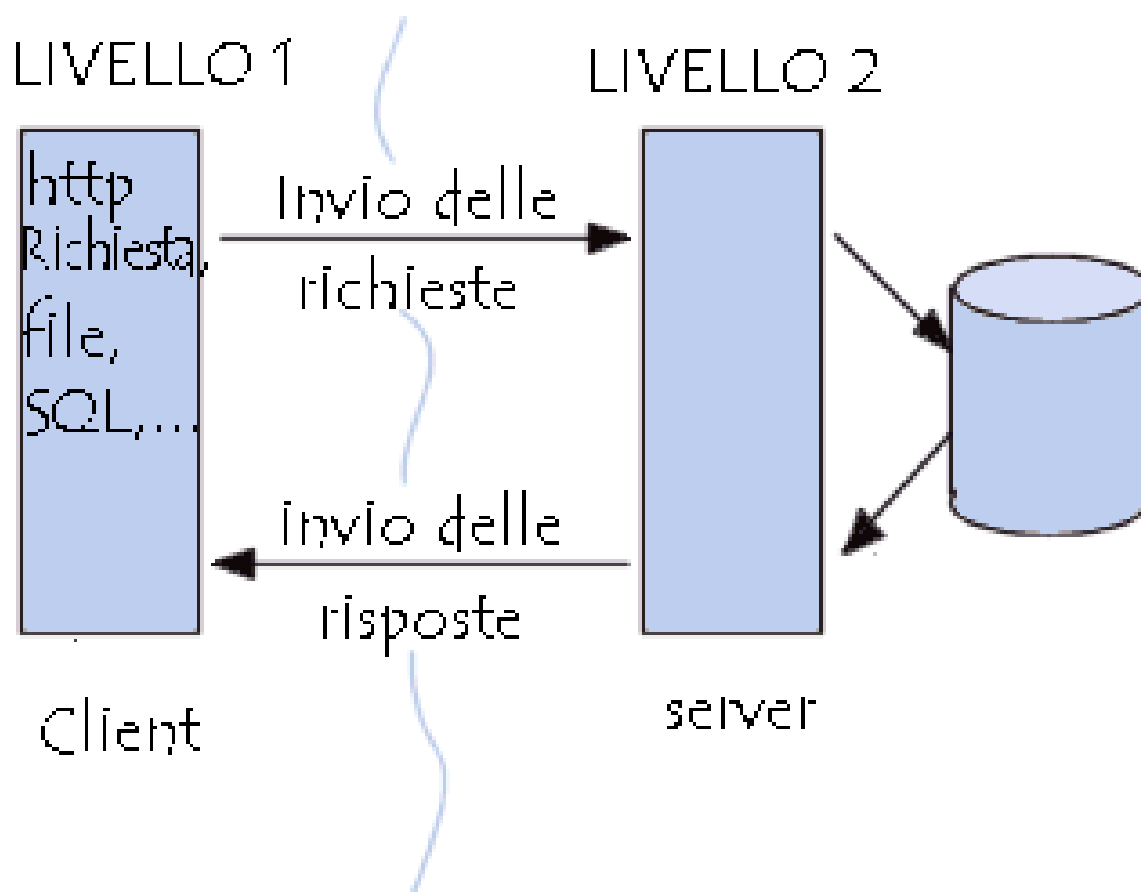
### ***1.8 Panoramica***

Il seguente documento SDD è diviso nelle seguenti sezioni:

- Al secondo punto del documento verrà presentato il sistema corrente.
- Al terzo punto verrà presentata l'architettura del sistema proposto in cui gestiremo la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema, le condizioni limite.
- Al quarto punto verranno presentati i servizi dei sottosistemi.

## **2. ARCHITETTURA DI SISTEMI SIMILI**

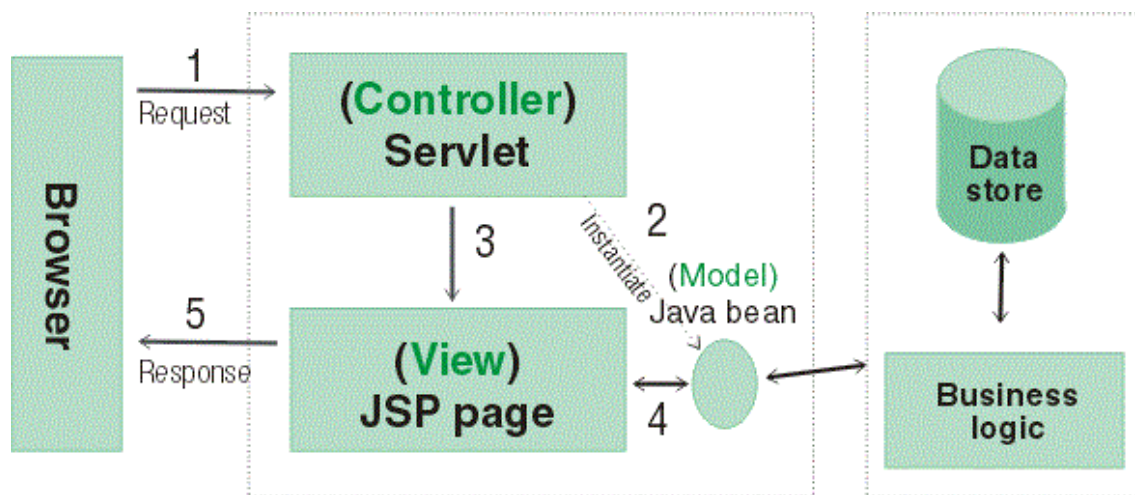
L'architettura del sistema corrente è costituita da sistemi client e server. Il client richiede servizi e inoltra tale richiesta al server, quest'ultimo risponde utilizzando le proprie risorse.



### 3. SISTEMA PROPOSTO

#### 3.1 *Panoramica*

Il pattern architetturale è MVC ( Model, View, Control).

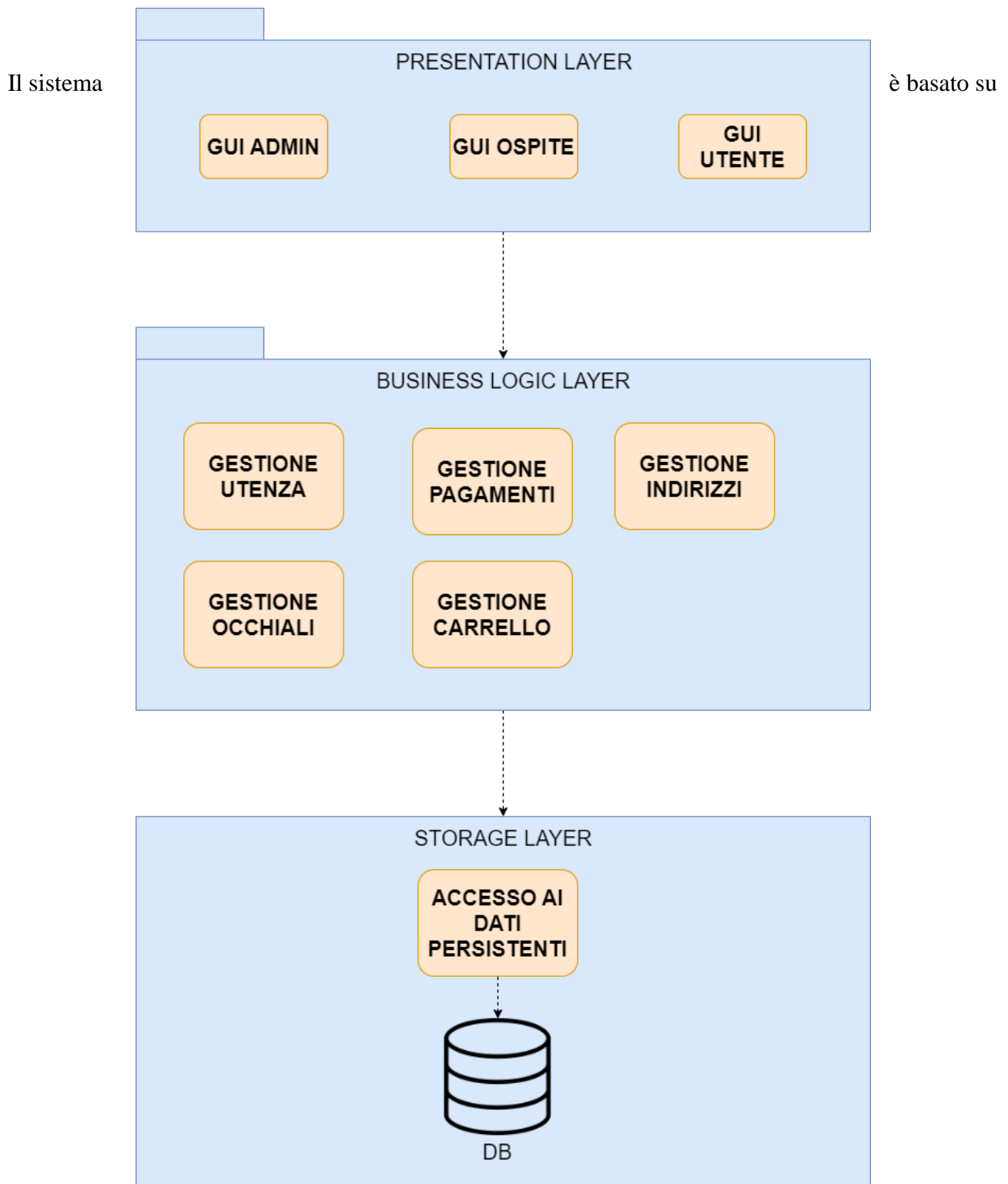


**Model:** contiene i metodi di accesso ai dati.

**View:** si occupa di visualizzare i dati all'utente e gestisce l'interazione fra quest'ultimo e l'infrastruttura sottostante.

**Controller:** riceve i comandi dell'utente attraverso il View e reagisce eseguendo delle operazioni che possono interessare il Model e che portano generalmente ad un cambiamento di stato del View.

### 3.2 Decomposizione in Sottosistemi



un'architettura three-tier. Grazie a questa divisione è infatti possibile immaginare di sviluppare, in futuro, un'applicazione che permetterà un più semplice utilizzo del sistema il quale, per natura, necessita di essere accessibile dai consumatori in qualsiasi momento.



Il sistema si compone di tre layers:

- **Il Presentation Layer** si occupa di presentare le informazioni ai diversi utenti e consente agli stessi di interagire con il sistema in modo semplice e intuitivo.
- **Il Business logic Layer** definisce la logica applicativa.
- **Lo Storage Layer** si occupa della gestione dei dati persistenti.

Il layer di presentazione presenta tre sottosistemi:

- **Gui Utente:** rappresenta l'insieme delle pagine web con le quali l'utente "Utente" interagisce per poter usufruire dell'ecommerce.
- **Gui Ospite:** rappresenta l'insieme delle pagine web con le quali un Ospite può interagire.
- **Gui Admin:** rappresenta l'insieme delle pagine web con le quali l'Admin del sistema interagisce per poter effettuare operazioni che riguardano la gestione degli occhiali o la ricerca degli ordini effettuati dagli utenti del sistema.

Il layer di business logic è diviso in cinque sottosistemi:

- **Gestione Indirizzi** fornisce le funzionalità per permettere all'utente di gestire i propri indirizzi.
- **Gestione Pagamenti:** fornisce le funzionalità per inserire i dati della carta di credito.
- **Gestione Occhiali:** fornisce le funzionalità per inserire o cancellare o modificare un occhiale nel database.
- **Gestione dell'utenza:** fornisce le funzionalità per effettuare l'autenticazione e, in generale, ciò che concerne la definizione e la gestione dei permessi per poter visualizzare determinate aree e funzionalità del sistema;
- **Gestione Carrello:** fornisce le funzionalità per modificare la quantità di un prodotto nel carrello o eliminarlo dallo stesso.

Il layer di persistenza è costituito da un unico sottosistema:

**Accesso ai dati persistenti:** si occupa di leggere e modificare i dati memorizzati sul Database.

### ***3.3 Mappatura Hardware/Software***

Per il sistema, basato su un'architettura MVC, sono state scelti questi tipi di configurazioni:

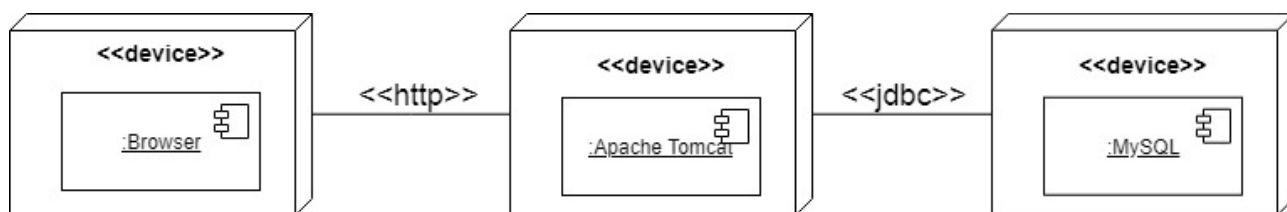
Per la parte di Interface (View) usiamo le JSP, HTML, CSS, JavaScript.

Per la parte di business logic (Control) usiamo Apache Tomcat e Java.

Per la parte di Data access (Model) usiamo MySQL con JDBC.

Come Protocollo di comunicazione usiamo HTTP, TCP/IP.

#### ***3.2.1 Deploy***



### 3.3 Gestione dei dati persistenti

È stato preferito l'utilizzo di un DBMS relazionale, in specifico MySQL per la gestione dei dati.

In più i DBMS garantiscono una gestione concorrente, l'accesso al DBMS è trasparente allo sviluppatore ed anche al cliente.

### 3.4 Schema concettuale

Per lo schema concettuale si rimanda al Class Diagram presente nel documento RAD\_THE\_SPECTACLES.

### 3.5 Schema logico

#### *Tabella utente*

<b>ATTRIBUTO</b>	<b>TIPO</b>	<b>VINCOLI</b>
nome	VARCHAR(20)	Not null
cognome	VARCHAR(20)	Not null
data di nascita	DATE	Not null
email	VARCHAR(45)	Primary key, Not null
password	VARCHAR(50)	Not null
ruolo	INT	Not null

La tabella utente contiene le informazioni riguardanti gli utenti presenti nel sistema. Un utente è identificato univocamente all'interno del sistema dalla sua e-mail (non possono esistere due utenti con la stessa username). Un utente nel database può essere sia un utente registrato al sistema, un gestore oppure un amministratore.

#### *Tabella ordine*

<b>ATTRIBUTO</b>	<b>TIPO</b>	<b>VINCOLI</b>
idOrdine	VARCHAR(36)	Primary key

		,Not null
data	TIMESTAMP	Not null
email	VARCHAR(36)	Not null
stato	TEXT	Not null

La tabella ordine contiene tutte le informazioni riguardanti l'ordine, quest'ultimo viene identificato univocamente tramite idOrdine, viene memorizzato il timestamp cioè la data e l'orario in cui si effettua l'ordine, l'email per associare un utente all'ordine effettuato e lo stato in cui si trova l'ordine(in lavorazione, spedito).

### *Tabella occhiale\_ordine*

<b>ATTRIBUTO</b>	<b>TIPO</b>	<b>VINCOLI</b>
id	INT	Primary key ,Not null, Auto increment
idOcchiale	VARCHAR(45)	Not null
idOrdine	VARCHAR(45)	Not null
prezzo_reale	INT	Not null
quantità	INT	Not null
iva	FLOAT	Not null

La tabella Occhiale\_Ordine è generata da un relazione N-M tra la tabella ordine e la tabella occhiale. Si identifica univocamente tramite id.

### *Tabella occhiale*

<b>ATTRIBUTO</b>	<b>TIPO</b>	<b>VINCOLI</b>
idOcchiale	VARCHAR(45)	Primary key ,Not null
nomeOcchiale	VARCHAR(30)	Not null
marca	VARCHAR(30)	Not null
prezzo	INT	Not null
disponibilità	INT	Not null
tipo	VARCHAR(1)	Not null
colore	VARCHAR(30)	Not null
categoria	VARCHAR(30)	Not null
img	VARCHAR(45)	Not null

img	VARCHAR(45)	Not null
descrizione	VARCHAR(500)	Not null

La tabella occhiale contiene tutte le informazioni riguardanti gli occhiali, questi vengono identificati univocamente tramite un idOcchiale, inoltre ci sono tutti gli attributi che lo caratterizzano.

### *Tabella indirizzi*

<i>ATTRIBUTO</i>	<i>TIPO</i>	<i>VINCOLI</i>
idIndirizzi	INT	Primary key ,Not null, Auto increment
Indirizzo	VARCHAR(45)	Not null
Attivo	TINYINT(1)	Not null
città	VARCHAR(45)	Not null
Provincia	VARCHAR(45)	Not null
Cap	INT	Not null
Email	VARCHAR(45)	Not null
Telefono	VARCHAR(45)	Not null

La tabella indirizzi contiene tutte le informazioni riguardanti l'indirizzo, quest'ultimo è associato ad un utente, il quale può avere più indirizzi associati. Un indirizzo viene identificato univocamente tramite un idIndirizzo.

## **3.4 Controllo degli accessi e sicurezza**

L'accesso agli utenti è garantito dall'utilizzo di username e password, e verrà chiesto a ogni scadenza delle sessioni.

<b>Oggetti</b> <b>Attori</b>	<b>TABELLA UTENTE</b>	<b>TABELLA ORDINE</b>	<b>TABELLA OCCHIALE</b> <b>ORDINE</b>	<b>TABELLA OCCHIALE</b>	<b>TABELLA INDIRIZZI</b>
<b>UTENTE NON REGISTRATO</b>	<ul style="list-style-type: none"> <li>• Effettuare registrazione</li> <li>• Accedere alla piattaforma</li> </ul>			<ul style="list-style-type: none"> <li>• Visualizza catalogo</li> </ul>	
<b>UTENTE REGISTRATO</b>	<ul style="list-style-type: none"> <li>• Logout</li> <li>• Modificare dati</li> </ul>	<ul style="list-style-type: none"> <li>• Visualizza ordini effettuati</li> </ul>		<ul style="list-style-type: none"> <li>• Visualizza catalogo</li> </ul>	<ul style="list-style-type: none"> <li>• Visualizza indirizzi</li> </ul>
<b>ADMIN</b>	<ul style="list-style-type: none"> <li>• Logout</li> <li>• Visualizza ordini effettuati da un cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Visualizza ordini effettuati da un cliente</li> <li>• Visualizza ordini effettuati in un intervallo di tempo</li> </ul>		<ul style="list-style-type: none"> <li>• Aggiungere nuovi occhiali</li> <li>• Modificare occhiali esistenti</li> <li>• Eliminare occhiali esistenti</li> </ul>	

### **3.5 Controllo flusso globale del software**

Il flusso è Event-Driven in cui ad ogni azione dell'utente è associato un evento, questo perché si tratta di un sistema web fortemente guidato dall'interazione dell'utente con le pagine web.

### **3.6 Condizioni limite**

#### **SERVER**

Fase di Startup:

L'avvio del server Apache Tomcat richiede circa 5 secondi, nel server è contenuto il DBMS MySQL.

Viene effettuata la connessione al DBMS tramite il driver JDBC e poi viene avviata la Java Virtual Machine sul server e vengono letti i dati di configurazione del database.

La fase di avvio del server Apache Tomcat dura circa 5 secondi, dopo di che viene avviato il server dove è contenuto il DBMS MySQL. Viene effettuata la connessione al DBMS tramite il driver JDBC. Successivamente viene avviata la JVM sul server e vengono letti i dati di configurazione del database. Infine, vengono caricati tutti i dati in memoria volatile. Il sistema è pronto all'uso.

#### Fase di Terminazione:

Il server può essere spento nel caso si verifichi un guasto o sia necessario un intervento di manutenzione. Una volta conclusa la riparazione del guasto o l'intervento di manutenzione il server viene riavviato normalmente

#### Fallimento:

Il server può andare in crash quando viene sollevata un'eccezione che non è stata gestita.

### **CLIENT**

Fase di Terminazione:

L'utente chiude la pagina del sistema interrompendo la comunicazione e quindi la sessione TCP.

#### **Scenario “Startup”**

Un amministratore vuole avviare la piattaforma The Spectacles. Inizia avviando il server Apache Tomcat dopodiché configura il DBMS MySQL per essere raggiunto dal medesimo web server. A questo punto, carica su Tomcat il file .war della piattaforma The Spectacles. Dopo 5 secondi dall'avvio la piattaforma è operativa e raggiungibile dagli utenti.

#### **Scenario “Fallimento”**

Il server mostra al client una schermata di errore.

## **4. SERVIZI DEI SOTTOSISTEMI**

Gestione utenza:

Login: il sottosistema permette di effettuare l'autenticazione

Registrazione: il sottosistema permette di creare un nuovo account

Logout: il sottosistema permette di far disconnettere l'utente autenticato

#### Gestione occhiali:

Visualizza occhiali: il sottosistema permette di visualizzare le informazioni per un certo tipo di occhiali

Visualizza lista occhiali: il sottosistema permette di visualizzare tutti gli occhiali presenti nel catalogo

#### Gestione carrello:

Inserisci prodotto nel carrello: il sottosistema permette ad un utente di inserire un prodotto nel carrello

Modifica quantità prodotto nel carrello: il sottosistema permette ad un utente di modificare la quantità di un prodotto già presente nel carrello

Elimina prodotto dal carrello: il sottosistema permette ad un utente di eliminare un prodotto dal carrello

#### Gestione indirizzi:

Inserisci indirizzo: il sottosistema permette di aggiungere un indirizzo di consegna

Elimina indirizzo: il sottosistema permette di eliminare un indirizzo di consegna

Seleziona indirizzo di consegna: il sottosistema permette di selezionare un indirizzo di consegna

#### Gestione pagamenti:

##### Accesso ai dati persistenti:

Eliminazione occhiali: Il sottosistema permette l'eliminazione dal database degli occhiali individuandoli attraverso il loro ID.

Inserimento occhiali: il sottosistema permette di salvare sul database le informazioni che riguardano degli occhiali.

Modifica occhiali: il sottosistema permette di modificare le informazioni riguardanti degli occhiali nel database identificandoli tramite il loro ID.

Inserimento indirizzo: il sottosistema permette di associare un nuovo indirizzo nel database ad un utente registrato identificandolo tramite email.

Eliminazione indirizzo: il sottosistema permette di eliminare un indirizzo associato ad un utente identificandolo tramite email

Inserisci ordine: il sottosistema permette di associare un nuovo ordine ad un utente registrato identificandolo per email.

Ottieni ordini: il sottosistema permette di reperire gli ordini di utente identificandolo tramite email.

## **5. GLOSSARIO**