



IL SITO E-COMMERCE PIÙ INNOVATIVO DI SEMPRE



## I membri del gruppo

|                        |            |
|------------------------|------------|
| D'Avanzo Giuseppe      | 0512102892 |
| Cerreto Luigi          | 0512102940 |
| Manganiello Alessandro | 0512103246 |
| Barone Gerardo         | 0512103264 |



Giuseppe D'Avanzo

Responsabile della documentazione

# REQUIREMENT ANALYSIS DOCUMENT

Il RAD ha l'obiettivo di fornire una prima modellazione del sistema descrivendolo attraverso strumenti offerti dall'ingegneria del software e serve come base contrattuale tra il cliente e lo sviluppatore.

Questo documento è formato da due tipi di requisiti:

- Requisiti funzionali
- Requisiti non funzionali



## Descrizione del problema:

*Il nostro progetto nasce dall'idea di fornire agli utenti un modo più rapido e sicuro di effettuare acquisti direttamente da casa: effettuare acquisti online rappresenta una valida alternativa al doversi recare personalmente in negozio.*

*Questo sistema permette ad utenti che per motivi di lavoro o di studio hanno a disposizione poco tempo libero, o ad utenti affetti da disabilità fisiche, di non rinunciare al piacere degli acquisti, in quanto sulla nostra piattaforma di e-commerce è possibile scegliere tra una vasta gamma di prodotti quelli di cui più abbiamo bisogno e farseli spedire direttamente a casa propria, eliminando così la necessità o in caso di disabilità fisiche, addirittura il disturbo di uscire di casa anche per gli acquisti più banali.*

## Scopo del sistema

Lo scopo del sistema è quello di mettere in vendita a tutti gli utenti oggetti di uso comune di diversi negozi su un unico portale.

L'utente non dovrà più acquistare gli oggetti presso più negozi online ma, attraverso un unico sito.

Questo comporta un risparmio di tempo da parte dell'utente e anche un risparmio da parte del venditore, il quale non avrà la necessità di creare un proprio sito e-commerce ma dovrà inserire solo i propri prodotti all'interno di questa grande piattaforma.

Inoltre l'utilizzo di questa piattaforma da parte di più venditori creerà una vera e propria competizione tra di loro, facendo una vera e propria gara al prezzo più basso.

La piattaforma, quindi, avrà sempre prezzi più bassi rispetto a tutti gli altri siti e-commerce in circolazione, permettendo un risparmio notevole all'utente medio.

# SCENARI

## LOGIN

### Utente

Effettua l'accesso nel pannello di login utilizzando le credenziali immesse durante la fase di registrazione.

### Amministratore

l'accesso al pannello amministratore utilizzando le credenziali di admin della piattaforma.

Entry condition : L'utente o gli admin immettono le credenziali (nome e password) e premono il bottone di log in.

Exit condition On success: Viene fornito un bottone di log out il quale termina la sessione dell'utente.

Exit condition On failure: Viene visualizzato un errore in caso il nome utente non è registrato nel database e nel caso in cui la password inserita, relativa al nome utente con cui si tenta di accedere, sia errata.

## FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO

- 1 Attore: Preme sul pulsante di log in.
- 2 Sistema: Controlla le credenziali fornite dall'utente.
- 3 Sistema: Notifica la possibilità di loggarsi come utente o come amministratore.
- 4 Attore: Interagisce con la tastiera, inserisce i parametri e preme invio.
- 5 Sistema: Verifica che il formato dei parametri sia Stringa ed esegue la richiesta.
- 6 Sistema: Utente: mostra l'homepage ad esso dedicata.  
Admin: mostra l'homepage ad esso dedicata.

CONTINUA

- I) Scenario/Flusso di eventi Alternativo: non è stato compilato alcun campo del form.
  - Sistema: Visualizzazione di un errore che notifica l'errore chiedere di inserire almeno un campo.
  - Sistema: Resta in attesa di una nuova compilazione del form da parte dell'utente.
- I) Scenario/Flusso di eventi di ERRORE: Credenziali non presenti all'interno del database
  - Sistema: Mostra un messaggio di errore, avvertendo l'utente di non poter effettuare il login.
  - Sistema: Suggerisce di cliccare sulla scritta “Registrati” per reindirizzare l'utente al form di registrazione.

# ACQUISTO PRODOTTI

Descrizione: L'utente dopo aver aggiunto un prodotto al carrello vuole effettuare il pagamento e terminare l'acquisto.

## Utente

Vuole acquistare un prodotto dalla piattaforma.

Entry Condition: L'utente preme il pulsante di acquisto, identificato da un bottone “Paga”, compare una schermata dove è richiesto di inserire i dati della carta di credito, e infine per completare si preme sul bottone di acquisto.

Exit condition On success : Le credenziali della carta di credito sono giuste.

Exit condition On failure: Viene visualizzato un messaggio di errore nel caso in cui le credenziali siano sbagliate.

## FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO

- 1 Attore: Preme sul pulsante di acquisto.
- 2 Sistema: Mostra il form di inserimento delle credenziali della carta di credito.
- 3 Attore: Interagisce con la tastiera, inserisce i dati della carta.
- 4 Sistema: Verifica che le credenziali siano giuste.
- 5 Sistema: Mostra un messaggio di pagamento effettuato.

I) Scenario/Flusso di eventi Alternativo: non è stato compilato il campo del form

- Sistema: Visualizzazione di un messaggio di errore che richiede di inserire i dati nei campi.
- Sistema: Resta in attesa di una nuova compilazione del form da parte dell'utente

II) Scenario/Flusso di eventi Alternativo: il parametro inserito non è conforme.

- Sistema: Notifica l'errore con un messaggio di errore.
- Sistema: Resta in attesa di una nuova compilazione del form da parte dell'utente.

# CASI D'USO

- **Nome:** Area utente, Registrazione e Login.

- **Attori:** Utenti.

## Flusso di eventi

Luca è un ragazzo di 22 anni che effettua spesso acquisti online presso il noto sito Amazon. Egli è venuto a conoscenza di questo nuovo sito e-commerce con prezzi vantaggiosi rispetto agli altri siti. Luca accede alla home e visualizza i prodotti, successivamente tramite il pulsante registrati verrà portato nella pagina di registrazione dove ci sarà un form da compilare. Effettuata la registrazione potrà fare il login al sito per accedere alle funzioni di acquisto.

- L'utente interessato all'acquisto di prodotti in vendita sulla piattaforma dovrà registrarsi inserendo un username e una password e tutti gli altri dati richiesti dal form. Una volta completata la registrazione l'utente, sarà dotato di un'area personale nella quale potrà modificare il profilo, accedere ai dati riservati e monitorare lo stato degli ordini.

- **Nome:** Acquisto dei prodotti.
- **Attori:** Clienti.

### Flusso di eventi

L'utente si è registrato al sito e ha effettuato il login tramite il pannello di login che si trova nella home page del sito. Sceglie i prodotti che gli interessano e li aggiunge al carrello tramite il bottone “Aggiungi al carrello”. Nella pagina dedicata al carrello sceglie la quantità tramite i bottoni “+” e “-“ e aggiorna il contenuto tramite il pulsante “aggiorna”, infine conferma l'acquisto tramite il bottone “Conferma ordine”. Immette i dati della carta di credito tramite un form di compilazione. Confermato l'ordine tramite il pulsante “Paga” e aspetta comodamente a casa l'arrivo del suo pacco.

- L'utente nella pagina contenente tutti i prodotti in vendita, sceglierà un prodotto che offrirà lo store. I prodotti saranno presentati con un'immagine e il prezzo. Se l'utente sarà interessato ad un prodotto, potrà cliccare sull'oggetto e accedere alla pagina dedicata interamente al prodotto. Il cliente avrà accesso alle informazioni aggiuntive quali la descrizione dettagliata e ulteriori immagini; da questa pagina potrà anche scegliere se aggiungere al carrello il prodotto che si sta visualizzando e procedere con la finalizzazione dell'ordine.



Alcune funzionalità della nostra piattaforma..



# Registrazione Utente

La prima e fondamentale azione che un utente possa compiere una volta visitata la nostra piattaforma, è quella di registrarsi ad essa.

Questa azione permetterà all'utente di acquistare e/o vendere prodotti presenti sulla piattaforma; nel caso un utente fosse interessato anche alla vendita di prodotti potrà registrare un proprio negozio sulla piattaforma.

La registrazione richiede pochi minuti e assicura la privacy degli utenti, in quanto non saranno visibili in nessun modo dati sensibili agli altri utenti.



# Registrazione Utente

The diagram illustrates a user registration process. On the left, a screenshot of a web browser shows the homepage of "SafariTecnology". The page includes a navigation bar with "File", "Edit", "View", and "Help" menus. Below the menu is a search bar and a user authentication section with "Username" and "Password" fields, "Accedi" and "Registrati" buttons. The main content area is titled "WhatsShop" and contains links for "HOME", "NEGOZI", "CONTATTI", and "FAQ". A shopping cart icon with a plus sign is visible. At the bottom, there are six product thumbnails labeled "prodotto1" through "prodotto6". A large red arrow points from the "Registrati" button on the homepage to the registration form on the right. On the right, a screenshot of a web browser shows the "SafariTecnology" registration page. It features a series of input fields: "NOME", "COGNOME", "CITTA'", "CAP", "INDIRIZZO", "DATA DI NASCITA", "E-MAIL", "USERNAME", "PASSWORD", and a final "REGISTRATI" button at the bottom.



# Inserisci il Tuo Negozio

WhatsShop

Welcome User

Logout Amministra Negozio

HOME NEGOZI CONTATTI FAQ

prodotto1 prodotto2 prodotto3 prodotto4 prodotto5 prodotto6

double-click to edit

Registrazione Nuovo Negozio

WhatsShop

NOME NEGOZIO

DESCRIZIONE

PARTITA IVA

VIA

CITTA'

CAP

E-MAIL



Una volta creato il negozio è possibile inserire i prodotti

The screenshot shows a web-based application window titled "Aggiungi Prodotto" (Add Product). At the top, there is a navigation bar with links: HOME, NEGOZI, CONTATTI, FAQ, and ADMIN. Below the navigation bar, there are two sections: "Profilo Utente" (User Profile) and "Amministra Negozio" (Manage Store). The main form starts with a label "Nome Prodotto" (Product Name) followed by an empty input field. Next is a label "Prezzo" (Price) with an empty input field. Then comes a label "Descrizione Prodotto" (Product Description) with an empty input field. Below that is a label "Quantità" (Quantity) with an empty input field. The next section is "Categoria" (Category), which is a dropdown menu currently set to "Elettronica". Following this is a label "Immagine" (Image) with a button labeled "Seleziona File" (Select File). At the bottom of the form is a button labeled "Aggiungi" (Add). A green tooltip at the bottom right of the form area says "double-click, ENTER or F2 to edit".



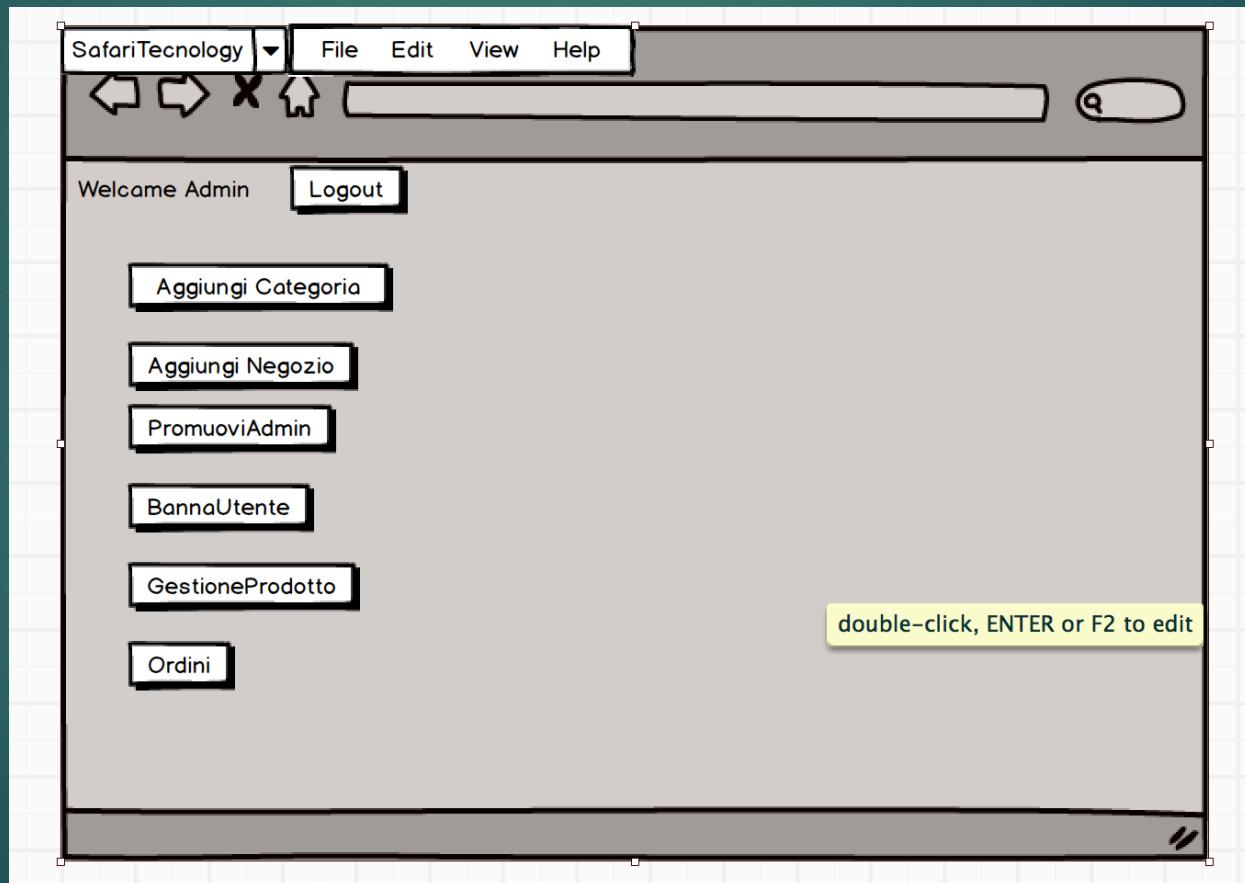
# Admin

Gli admin del sito hanno il compito di controllare e garantire il corretto funzionamento della piattaforma.

Possono accedere alla cronologia degli ordini effettuati da un utente o una lista di ordini effettuati in un determinato periodo, bannare utenti in caso di anomalie e controllare i prodotti messi in vendita.



# Pannello di controllo degli admin



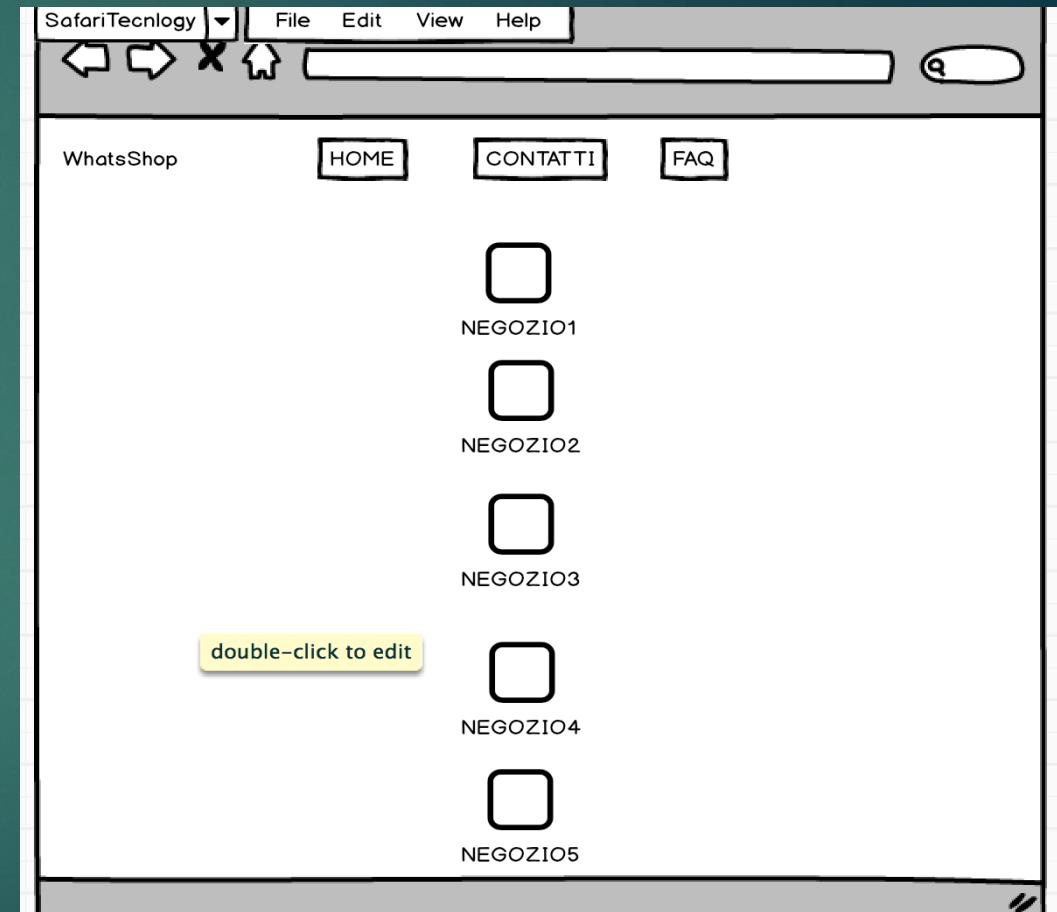
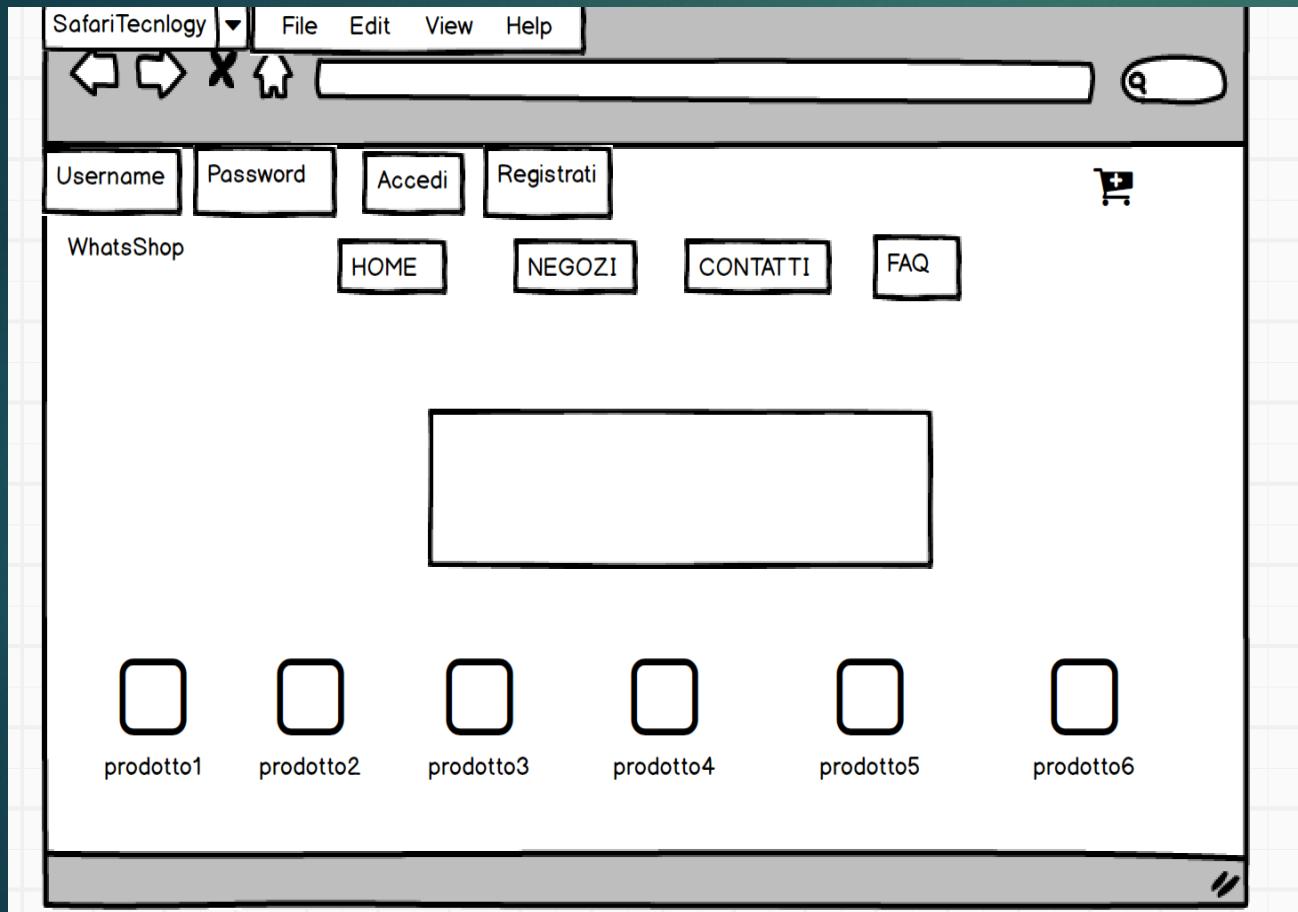


# Navigare nel sito

Navigare nel sito è facile e veloce anche per gli utenti meno esperti, poiché abbiamo cercato di rendere l'interfaccia quanto più elementare e comprensibile possibile.

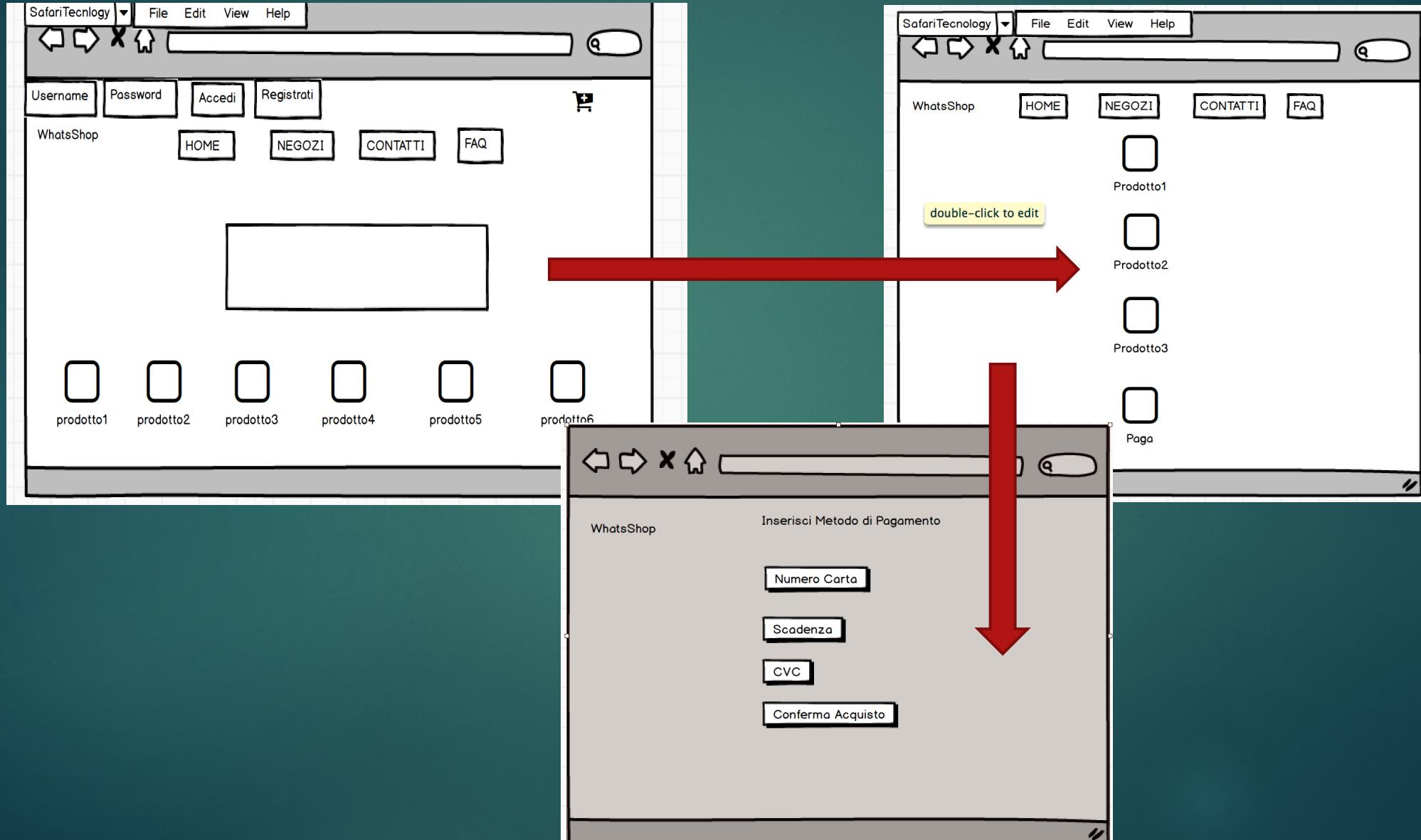
Un utente interessato ad una determinata tipologia di prodotto o ad un negozio in particolare, potrà agilmente trovare i prodotti di interesse; una volta selezionati tali prodotti, l'utente registrato potrà inserirli nel carrello e procedere al pagamento.

## Ricerca e selezione dei prodotti





# Pagamento



# SYSTEM DESIGN DOCUMENT



Gerardo Barone

Database Analyst

## Architettura di sistema

Il nostro sistema è strutturato con un architettura Model-Wiew-Controller

- Il **model** si occupa della gestione dei dati quindi dell'interazione con il database e dell'astrazione degli stessi.
- la **view** si occupa dell'interazione con l'utente e quindi della visualizzazione dei dati con una certa formattazione
- il **controller** riceve i comandi inviati dall'utente alla visualizzazione si occupa dell'eventuale elaborazione del passaggio al modello e di nuovo della risposta alla visualizzazione.

# MODEL-VIEW-CONTROL

## View

### **Utente Amministratore**

Modulo che si occupa di gestire le funzionalità del sistema riservate all'amministratore, come la gestione degli utenti e dei prodotti come il ban di un utente o l'aggiunta, la modifica o la rimozione di un prodotto dalla piattaforma

### **Utente Visitatore del sito**

Modulo che si occupa di gestire le funzionalità del sistema riservate all'utente, come la visualizzazione dei prodotti che offre la piattaforma e l'aggiunta di un prodotto al carrello per poi passare all'acquisto.

# MODEL-VIEW-CONTROL

## Model

### Gestione Utenti

Modulo che permette la registrazione di un nuovo utente al sistema, gestisce l'accesso degli utenti, garantendo l'adeguata assegnazione dei privilegi dovuti (possibilità di acquistare prodotti) e le relative funzionalità.

### Gestione Prodotti

Modulo che permette di visualizzare, creare, eliminare o modificare prodotti sulla piattaforma.

### Gestione Carrello

Modulo che permette di visualizzare, eliminare, modificare o aumentare la quantità degli oggetti nel carrello elettronico

### Gestione Login

Modulo che si occupa della gestione del login degli utenti, controlla se username e password coincidono con quelle esistenti nel database e successivamente permette il login ad un utente.

# MODEL-VIEW-CONTROL

## Model

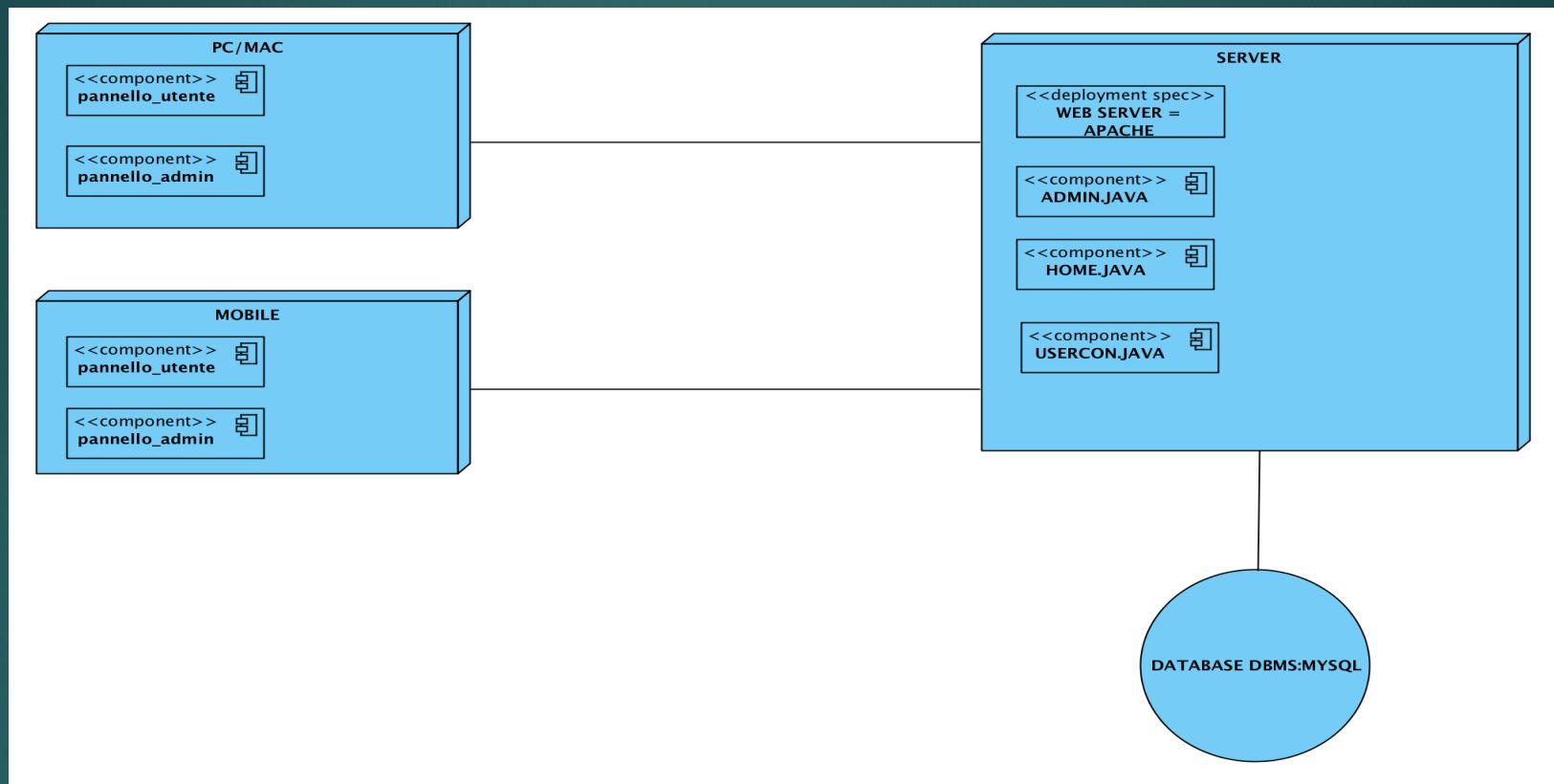
### Dati persistenti

Modulo che si occupa di memorizzare i dati fondamentali degli utenti e dei prodotti in memoria, in modo da poter essere prelevati e modificati in modo corretto.

Per memorizzare i dati persistenti abbiamo utilizzato un database relazionale il quale assicura:

- Privatezza dei dati
- Atomicità delle operazioni
- Affidabilità dei dati

# MAPPING H/S



L'utente utilizzerà il sistema mediante un Browser installato sulla propria macchina

**Web Server**

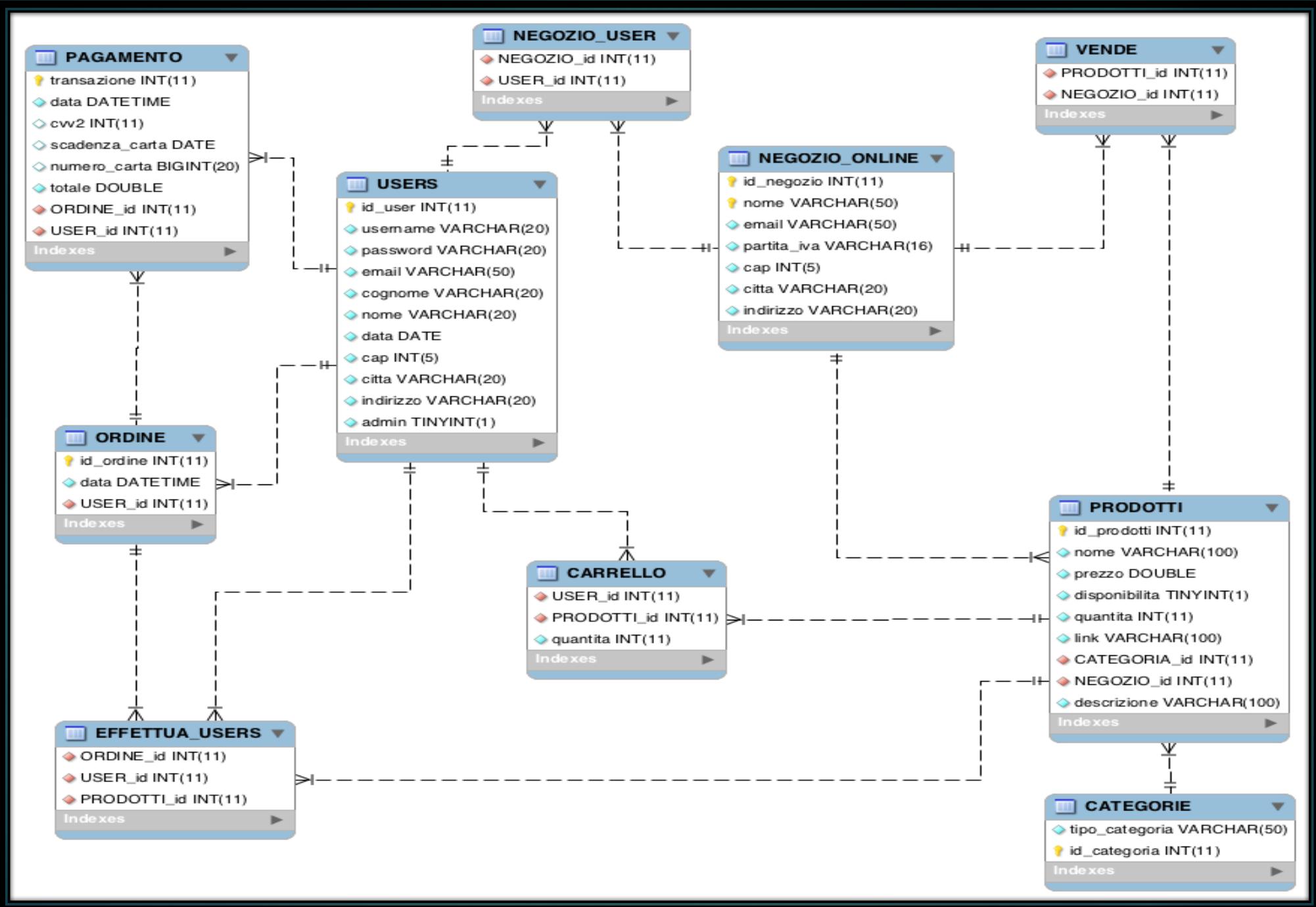
Il server utilizzato è Apache Tomcat

**Database Server**

Il DBMS usato è MySQL

# SCHEMA ER





# Object Design Document



Alessandro Manganiello

Supervisor and quality

Dopo la finalizzazione del documento RAD (Requirement Analysis Document) e SDD (System Design Document), abbiamo descritto quello che sarà il nostro sistema e dunque i nostri obiettivi, tralasciando gli aspetti di implementazione. Per quanto riguarda la realizzazione del sistema Whatshop, sono stati individuati i seguenti trade-off.

- **Prestazioni VS Costi**

Considerando il nostro sistema, possiamo dire che il budget non eccessivo a nostra disposizione ci ha permesso di realizzare il nostro prodotto utilizzando materiale open source per permetterci di minimizzare i costi e rendendo il sistema molto soddisfacente.

- **Comprendibilità VS Tempo**

Parte fondamentale della progettazione è costituita da codice comprensibile anche per eventuali persone esterne al progetto. Per avere tale requisito è necessario prevedere l'utilizzo di classi con metodi ben identificabili e facilmente interpretabili, utilizzando l'indentazione e una documentazione appropriata del codice sorgente come i commenti. Ciò favorisce anche la comprensibilità, agevolando il processo di mantenimento e di modifica del progetto anche per futuri sviluppatori che non hanno lavorato dall'inizio al progetto stesso. Questo vantaggio tuttavia comporta un incremento del tempo per lo sviluppo e la realizzazione dell'intero sistema.

- **Interfaccia VS Tempo di risposta**

I tempi di risposta tra il nostro server e l'interfaccia non troppo minimale del sito web sono molto rapidi e quindi più che sufficienti per soddisfare le richieste dei vari utenti connessi alla piattaforma. I tempi di risposta potranno, però, variare a seconda della quantità dei negozi presenti sulla piattaforma e quindi nel database e a seconda della quantità di prodotti inseriti. Dunque maggiore sarà la grandezza del database e maggiori saranno i tempi di inoltro delle richieste nel database.

- **Interfaccia VS Easy-use**

La nostra interfaccia, grazie all'utilizzo delle form e grazie ad una impostazione molto semplice ed intuitiva, permette un facile utilizzo della gestione della piattaforma sia ad un rivenditore esperto che ad un neofita, fino ad arrivare ad un facile utilizzo anche per l'utente iscritto che andrà semplicemente ad acquistare prodotti sul sito web.

- Costi VS Mantenimento

I nostri costi sono stati ridotti al minimo grazie all'utilizzo di materiale open source e l'utilizzo di linguaggi Java. Grazie a determinate implementazioni, il sistema può essere facilmente modificato in futuro ed implementato di nuove funzioni, oppure corretto se in presenza di errori.

I costi di mantenimento aumenteranno in rapporto alla quantità di negozi aperti sul nostro sito web e al personale impiegato per la gestione dell'intera piattaforma.

La scelta di utilizzare un database è scaturita dai diversi vantaggi che se ne derivano:

- Gestione consistente dei dati;
- Tempo di risposta basso rispetto all'utilizzo di un file system;
- Accesso veloce e concorrente ai dati;

Ovviamente l'utilizzo di un database comporta l'utilizzo di una grande quantità di memoria nel momento in cui la mole di dati dovesse essere notevole.

# Design Pattern

Per la gestione della connessione al database si è deciso di utilizzare sulla classe connessione.java il design pattern **singleton**, ovvero fare in modo che della classe esista solo un'istanza e fornire un solo punto di accesso ad essa, impedendo che sia possibile istanziare diverse copie.

# COM'E' ORGANIZZATO ?

Il sistema è multi – utente (cioè può accedervi chiunque, sia un semplice cliente, un rivenditore con un negozio oppure un amministratore del sito web).

- Al cliente semplice ed al rivenditore, il sistema nasconde la logica delle operazioni e del codice, fornendogli solamente dei form base utili per effettuare funzioni nel sito web, come ad esempio ricercare prodotti, aggiungere prodotti ad un carrello o creare un nuovo negozio per vendere i propri oggetti.
- L'amministratore del sito deve invece avere accesso alle funzionalità avanzate del sito, come ai pannelli di gestione ed amministrazione della piattaforma, al database per verificare errori, ai log per verificare acquisti ed al pannello di gestione degli utenti iscritti, per gestire allontanamenti dei suddetti in casi specifici di utilizzo scorretto del sito web.

# Ebay come ispirazione...e rivale

La nostra piattaforma di e-commerce è stata fortemente ispirata ai grandi nomi del web, come eBay. A differenza di altri store online, eBay racchiude meglio la nostra idea di commercio, in quanto è nostro desiderio permettere agli utenti di essere parte integrante della piattaforma, permettendo loro di mettere in vendita i propri prodotti (gli utenti devono rispettare determinati requisiti, come essere forniti di partita IVA). Tuttavia la nostra piattaforma non consente ad un utente di mettere in vendita prodotti senza aver prima registrato un negozio all'interno del sito. E' una differenza importante rispetto ad Ebay(che consente ad ogni utente di mettere in vendita prodotti senza dover creare un negozio) che riteniamo garantisca ad un utente un maggior controllo e una più facile gestione dei prodotti messi in vendita.

# TESTING

Per effettuare il testing della piattaforma abbiamo utilizzato il framework JUnit, per verificare il corretto funzionamento del sistema.

Verranno testate varie funzionalità del sistema per capire dove si dovrà intervenire per correggere eventuali errori o apportare modifiche per il miglioramento dei vari sottosistemi.

# JUnit



Luigi Cerreto

Programmer/Tester

# Cosa andremo a testare?

Nella fase di testing verranno prese in considerazione le funzionalità fondamentali:

- Login: Sarà testato il corretto funzionamento dell'accesso al sito, quindi darà all'utente la possibilità di sfruttare al massimo la nostra piattaforma.
- Registrazione: Sarà testato il corretto funzionamento della registrazione al sito che poi permetterà a esso poi di accedere alla piattaforma.
- Aggiungi Negozio (Admin): Sarà testato l'aggiunta da parte di un amministratore di un negozio all'interno della piattaforma.
- Aggiungi Negozio (User): Sarà testato l'aggiunta da parte dell'utente di un negozio all'interno della piattaforma.
- Aggiungi Prodotto (Admin e User): Verranno testate le funzionalità sull'aggiunta di un prodotto nella piattaforma collegata a un negozio e una categoria.
- Aggiungi Categoria (Admin): Sarà testata la funzionalità da parte dell'amministratore che permette a egli di aggiungere una nuova categoria.
- Aggiungi a Carrello: Sarà testata la funzionalità da parte dell'utente che permette di aggiungere oggetti all'interno del proprio carrello.
- Pagamento: Sarà testato il metodo di pagamento che invia i dati riservati dell'utente alla piattaforma e completare il pagamento per l'acquisto dei prodotti esistenti nel carrello personale.

# TEST CASE ESEGUITI

RegisterTest2

Classi di equivalenza

| Classi                          | Descrizione Input                         |
|---------------------------------|---|
| Campi obbligatori non compilati | I campi obbligatori non vengono compilati |

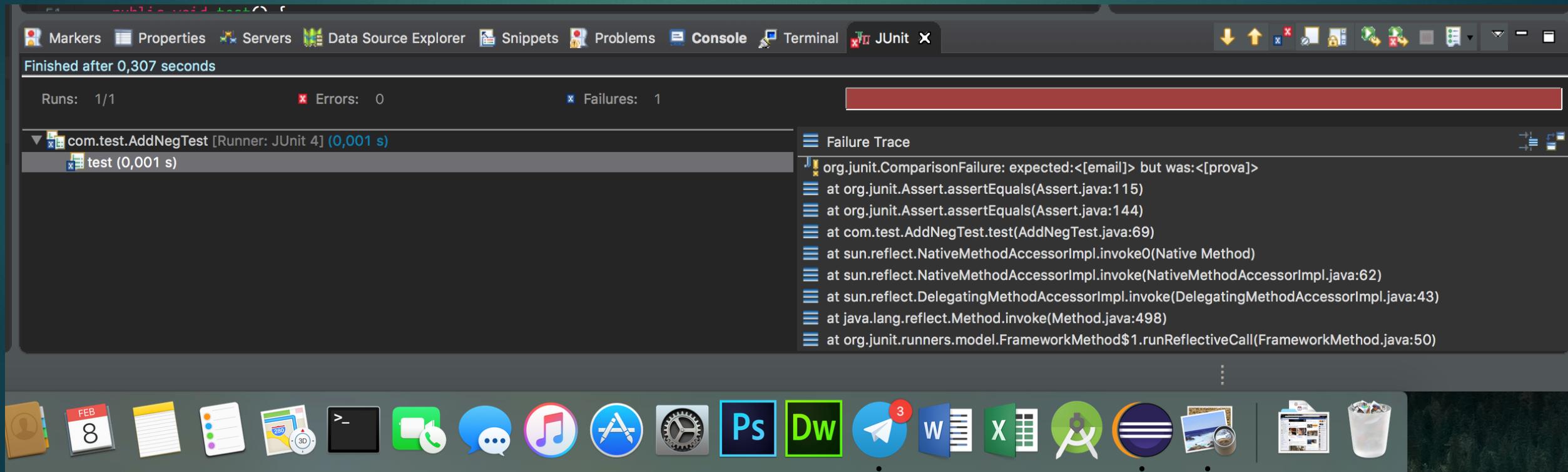
|                       |   |
|-----------------------|---|
| Test Case Name        | RegisterTest2   |
| Test Case ID          | TC_2.2.1  |
| Condizione di Entrata | L'utente si trova nella home e clicca su Registrati   |
| Flusso degli Eventi   | <ul style="list-style-type: none"><li>L'utente clicca su Registrati e viene indirizzato nella pagina di registrazione.</li><li>WS mostra i campi che devono essere compilati</li><li>L'utente non inserisce la password e l'indirizzo</li></ul> |
| Condizione di uscita  | L'utente visualizza i campi non compilati di colore rosso   |

Sono stati trascritti i casi di test con le relative classi di equivalenza. Tutti i test case sono reperibili nel documento WS\_TestCase.

|                              |   |
|------------------------------|---|
| <b>Test Case Name</b>        | AggiungiCarrelloTest1   |
| <b>Test Case ID</b>          | TC_5.1.1  |
| <b>Condizione di Entrata</b> | L'utente si trova nella home e sceglie i prodotti da comprare   |
| <b>Flusso degli Eventi</b>   | <ul style="list-style-type: none"> <li>• L'utente, dopo aver scelto il prodotto clicca su aggiungi</li> <li>• WS aggiunge il prodotto al carrello con un messaggio a comparsa</li> <li>• L'utente clicca sul carrello e trova il suo oggetto</li> </ul> |
| <b>Condizione di uscita</b>  | Il prodotto è stato aggiunto al carrello  |

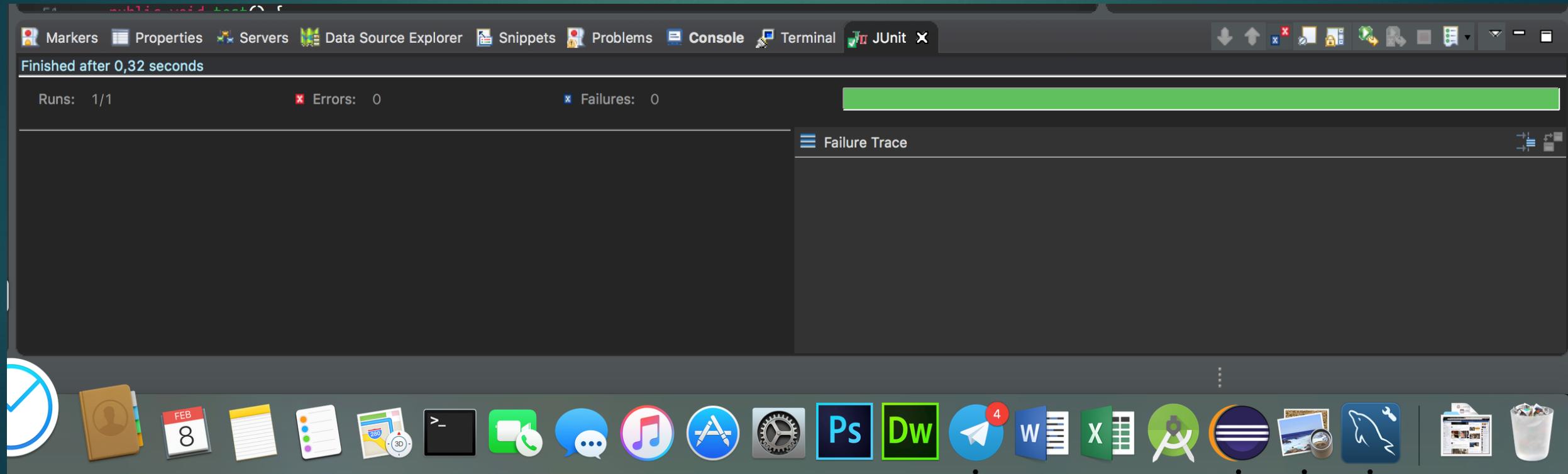
# Test Failed

## Aggiunta di un nuovo negozio



# Test Success

## Aggiunta di un nuovo negozio





# Relazione sull'Usabilità

*Le indagini vengono svolte dai designer con il supporto necessario dei valutatori.*

*Abbiamo inoltre selezionato eventuali utenti che hanno utilizzato la nostra piattaforma per il test di usabilità, abbiamo fatto provare loro la nostra applicazione e abbiamo esaminato le loro reazioni mentre la usavano per comprendere se la piattaforma a primo impatto sia stata intuitiva ed efficace e quindi se riuscivano a compiere tutte le operazioni disponibili senza difficoltà.*

*A fine indagini, gli utenti ci hanno riferito che la piattaforma risulta molto **semplice** e **veloce** da utilizzare. In particolare gli utenti sono rimasti estremamente soddisfatti per alcune funzionalità, come ad esempio la possibilità di aggiungere nuovi negozi , quindi, vendere oggetti personali del venditore che si è iscritto sulla piattaforma.*



# Relazione sulla tecnica di Usabilità

A lavoro terminato la nostra piattaforma è stata testata da utenti estranei al gruppo di progetto. Inizialmente la piattaforma era difficile da usare e da capire, ad esempio non permetteva la visualizzazione dei prodotti venduti da un singolo negozio e la visualizzazione della descrizione completa di un singolo prodotto.

Feedback positivi ne erano pochi e i tanti feedback negativi hanno scaturito in noi la voglia di migliorare il tutto per rendere la piattaforma facile e funzionale.

Il duro lavoro e la voglia di creare un servizio ottimo per tutti ci hanno permesso di creare una piattaforma di forte comprensibilità e usabilità oltre ad una semplicità d'uso elevata. Anche se la piattaforma ora riceve tanti feedback positivi, continueremo a modificarla e a migliorarla introducendo sempre di più nuovi servizi.



## Miglioramenti della piattaforma

La relazione sulle tecniche di usabilità ci ha portato a migliorare i seguenti punti della nostra Web Application:

- Visualizzazione prodotti di un singolo negozio.
- Visualizzazione completa descrizione di un prodotto.
- Migliore navigabilità tra le sezione.
- Ogni utente può creare un proprio negozio e aggiungere prodotti.
- Visualizzazione lista completa di tutti i negozi.



# DEVELOPMENT



JS



CSS







**Fifa 17**  
Gioco di calcio

€49.90 3 Aggiorna

Totale: 149.70

Totale da pagare = 149.70

Conferma Ordine

## Fifa 17



Negozi: EA Sports

Categoria: Videogiochi

ID Prodotto: #1

Prezzo: € 49.9

Disponibilità: Disponibile

Quantità: 10

Descrizione:

Gioco di calcio

Aggiungi

[Torna Indietro](#)

© Whatshop



## Aggiungi Prodotto

## Nome Prodotto

Prezzo

## Descrizione Prodotto

## Quantita Prodotto

Negozio

Apple

## Categorie

Elettronica

## Immagine

**Choose Files** no files selected

Aggiungi

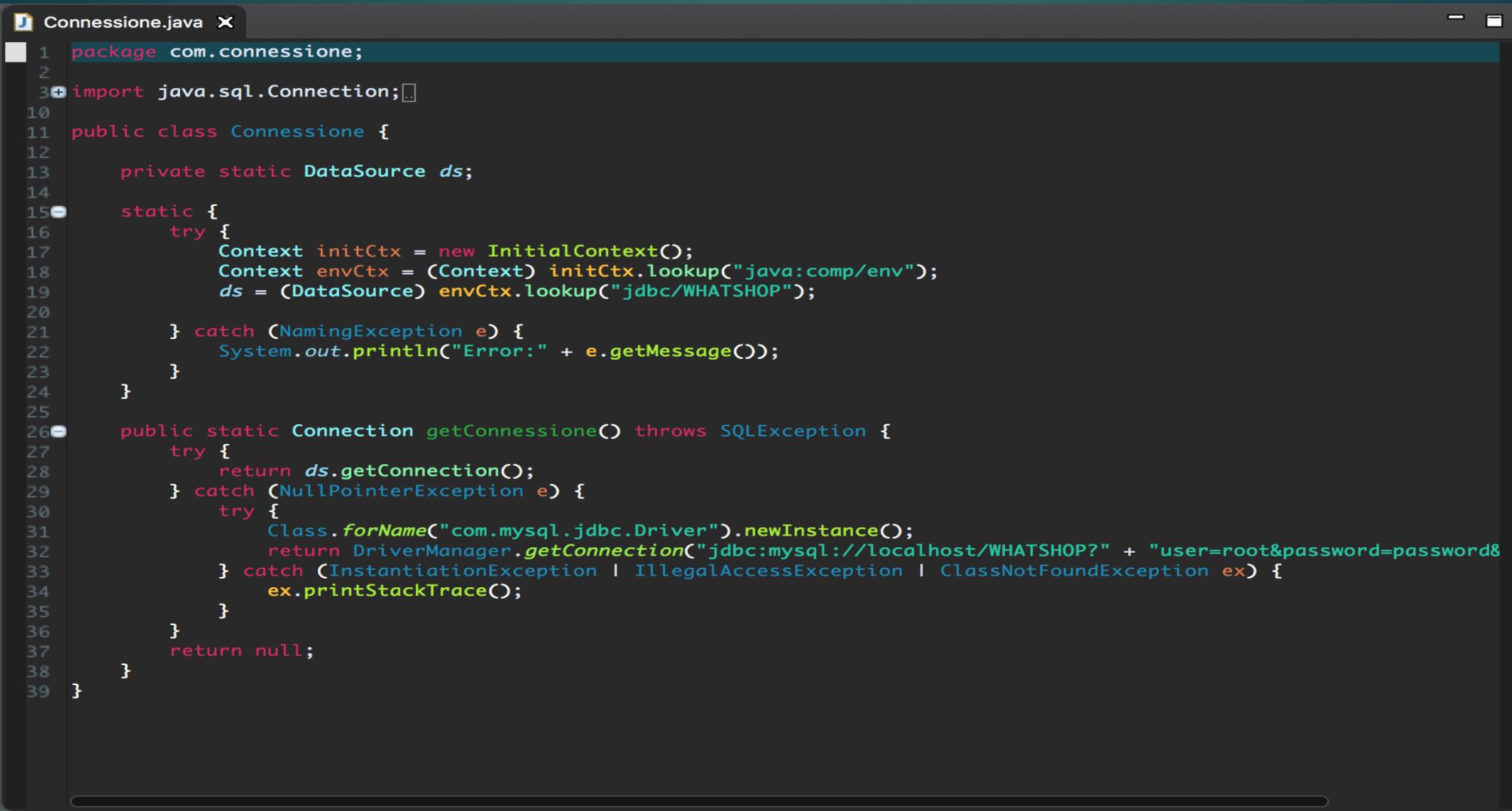
**Elimina Prodotto**

## Prodotti

Fifa 17

**Elimina**

# Connessione al database



The screenshot shows a Java code editor window with a dark theme. The title bar says "Connessione.java". The code is as follows:

```
1 package com.conessione;
2
3+import java.sql.Connection;
4
5 public class Connessione {
6
7     private static DataSource ds;
8
9     static {
10         try {
11             Context initCtx = new InitialContext();
12             Context envCtx = (Context) initCtx.lookup("java:comp/env");
13             ds = (DataSource) envCtx.lookup("jdbc/WHATSHOP");
14
15         } catch (NamingException e) {
16             System.out.println("Error:" + e.getMessage());
17         }
18     }
19
20     public static Connection getConnessione() throws SQLException {
21         try {
22             return ds.getConnection();
23         } catch (NullPointerException e) {
24             try {
25                 Class.forName("com.mysql.jdbc.Driver").newInstance();
26                 return DriverManager.getConnection("jdbc:mysql://localhost/WHATSHOP?" + "user=root&password=password8");
27             } catch (InstantiationException | IllegalAccessException | ClassNotFoundException ex) {
28                 ex.printStackTrace();
29             }
30         }
31         return null;
32     }
33 }
```

# Query Pagamento

```
1  package com.giulio.pizzaria;
2
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.SQLException;
6  import java.util.Date;
7
8  public class PagamentoModel {
9
10    public boolean registraPagamento(int prezzo, int user_id, int ordine_id, String data, String cvv2, String scadenzaCarta, long nCarta) {
11
12      String sql = null;
13
14      boolean bool = false;
15      Connection con = null;
16      try {
17          con = Connessione.getConnessione();
18
19          sql = "INSERT into PAGAMENTO(data,cvv2,scadenza_carta,numero_carta,totale,ORDINE_id,USER_id) value (?,?,?,?,?,?,?)";
20
21          PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(sql);
22
23          pstmt.setString(1, data);
24          pstmt.setInt(2, cvv2);
25          pstmt.setString(3, scadenzaCarta);
26          pstmt.setLong(4, nCarta);
27          pstmt.setDouble(5, prezzo);
28          pstmt.setInt(6, ordineId);
29          pstmt.setInt(7, userId);
30
31          if(pstmt.executeUpdate() > 0) bool = true;
32          else bool= false;
33          pstmt.close();
34
35
36          return bool;
37      } catch (SQLException e) {
38          // TODO Auto-generated catch block
39          e.printStackTrace();
40          return false;
41      }finally {
42          try {
43              con.close();
44          } catch (SQLException e) {
45              // TODO Auto-generated catch block
46              e.printStackTrace();
47          }
48      }
49  }
50 }
```



Grazie per l'attenzione

