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

***System Design Document 1.0***

***05/12/2022***

****

0512107476 **– Salvatore Santoriello**0512107395 **– Bruno Farano**0512106126 **– Emanuele Milito**0512107080 **– Gianmichele Cancellaro**

***Revision History***

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 05/12/2022 | 1.0 | Prima stesura del documento | Membri del team |

***Indice***

**1.** Introduzione  
 **1.1***Scopo del sistema* **1.2***Design goals*

**2.** Architettura del sistema corrente (Competitor)

**3.** Architettura del sistema proposto  
  **3.1***Decomposizione in sottoinsiemi* **3.1.1***Layering e Partitioning***3.2***Hardware/Software Mapping***3.3** *Persistent and Data Management***3.4***Access Control and Security***3.5** *Global Software Control***3.6** *Boundary conditions*

**1.** Introduzione  
 **1.1** *Scopo del sistema*

L’e-commerce è uno strumento che si sta espandendo in maniera vistosa negli ultimi anni, poiché utlilissimo per visionare ed acquistare prodotti che normalmente acquisteresti recandoti fisicamente in un negozio; infatti, il principale vantaggio dell’e-commerce è la possibilità di acquistare articoli ovunque ti trovi ed in qualsiasi circostanza.   
Se da una parte abbiamo il successo dell’e-commerce, dall’altra abbiamo lo sport più praticato e seguito al mondo: il calcio. Miliardi di persone attualmente hanno un notevole interesse per questo sport, praticandolo a livello amatoriale, professionale o semplicemente guardando la propria squadra del cuore alla TV.   
eSoccerce è un progetto che combina questi due fenomeni, mettendo a disposizione dell’utenza una piattaforma Web su cui acquistare articoli ed accessori di qualsiasi tipo riguardanti il calcio.

**1.2** *Design goals*

**DG (1) | Affidabilità**

* **Motivazione errori**  - il sistema deve specificare il motivo per il quale si è verificato un errore durante la navigazione. ***PRIORITÀ: ALTA***
* **Gestione input** - la piattaforma deve essere pronta a gestire formati errati di input e gestirli al meglio. ***PRIORITÀ: ALTA***
* **Criptazione password** - il sistema deve garantire che le password siano criptate e non visibili agli altri utenti. ***PRIORITÀ: ALTA***
* **Respinta dati malevoli** - la piattaforma non deve fornire la possibilità di inserire dati malevoli all’intero di sé stessa. ***PRIORITÀ: ALTA***

**DG (2) | Performance**

* **Accesso contemporaneo**  - il sistema deve garantire la navigazione contemporanea ad almeno 30 utenti, attraverso un sistema che sia scalabile. ***PRIORITÀ: BASSA***
* **Tempestività pagamento** - la piattaforma deve garantire che la procedura di pagamento sia tempestiva in modo da concludere l’ordine. ***PRIORITÀ: BASSA***

**DG (3) | Supportabilità**

* **Architettura 3-tier**  - il sistema deve prevedere un’architettura a 3 livelli, basandoci su un modello MVC (model-view-control), attraverso i DAO, i DTO, servlet e JSP. ***PRIORITÀ: ALTA***

**DG (4) | Usabilità**

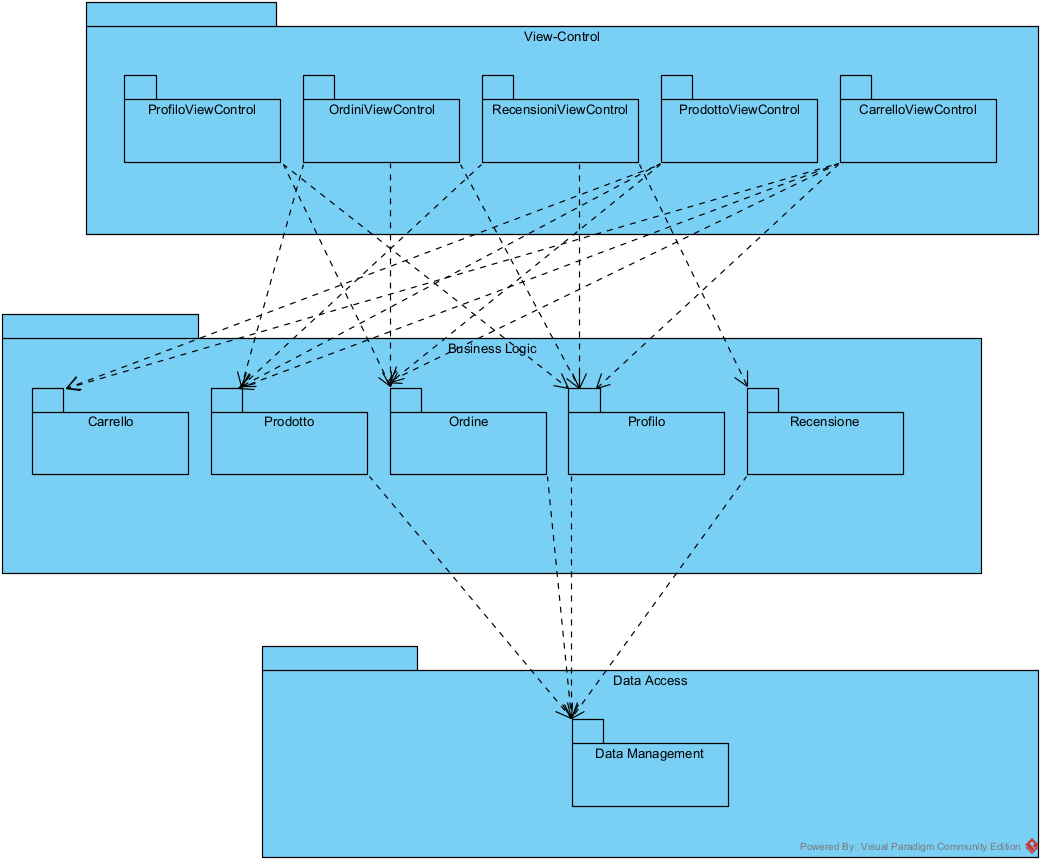
* **Interfaccia grafica**  - il sistema deve fornire un’interfaccia grafica semplice ed intuitiva. ***PRIORITÀ: ALTA***
* **Barra di navigazione** - il sistema deve fornire una barra di navigazione per accedere alle varie sezioni. ***PRIORITÀ: ALTA***
* **Adattamento ai dispositivi** - il sistema deve essere responsive, cioè adattarsi a tutti i dispositivi cui vi accedono. ***PRIORITÀ: ALTA***
* **Menù contestuale** - il sistema è fornito di un menù contestuale a tendina sulla sinistra dell’interfaccia, cosicché l’utente, ovunque egli si trovi, può decidere di accedere a tutte le categorie presenti nel sistema. ***PRIORITÀ: ALTA***

**2.** Architettura del sistema corrente (Competitor)

La piattaforma, attualmente, non esiste sul Web ma ci sono migliaia di competitors che mettono a disposizione la vendita di articoli sportivi e, nel nostro caso, calcistici e possiedono tutti un’architettura basata sul modello MVC (model-view-control).

**3.** Architettura del sistema proposto  
 **3.1** *Decomposizione in sottoinsiemi* **3.1.1** *Layering e Partitioning*

Il nostro sistema è suddiviso in 3 layers, i quali si occupano di gestire aspetti e funzionalità differenti. La suddivisione è basata su un particolare pattern per l’architettura software chiamato MVC (model-view-control).



***View-Control***

**ProfiloViewControl** - permette l’accesso alla visualizzazione della pagina del profilo e alla visione di tutti i suoi dati personali attuali nonché la modifica degli stessi, infatti indirizza l’utente a tutte le pagine di modifica dei dati personali.  
**OrdiniViewControl** - permette la visualizzazione di tutti gli ordini effettuati dall’utente, nonché, per ogni ordine, la visualizzazione della pagina apposita per i dettagli dello stesso.  
**RecensioniViewControl** - permette la visualizzazione della pagina relativa alle recensioni per ogni singolo prodotto, inoltre, permette di rimuovere una recensione propria nel caso in cui l’utente avesse un ripensamento per la pubblicazione.  
**ProdottoViewControl** - permette la visualizzazione della lista dei prodotti disponibili alla vendita, nonché la possibilità di aggiungerli al carrello. La vendita è suddivisa in categorie, ciascuna delle quali contiene la lista dei prodotti in base alla categoria di appartenenza.  
**CarrelloViewControl** - consente l’accesso al carrello per visionare i prodotti aggiunti e per modificare la quantità da voler acquistare dall’utente, nonché l’eliminazione del prodotto dal carrello stesso.

***Business Logic***

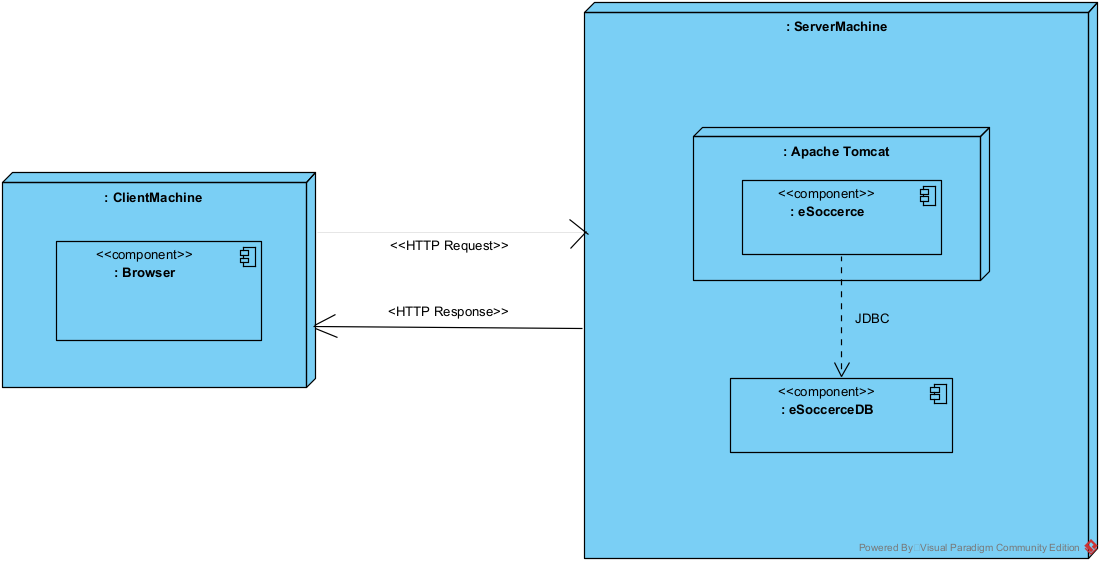
**Profilo** - offre numerosi servizi quali: visualizzazione dati attuali del profilo, accesso alla pagina degli ordini, accesso alle pagine apposite per la modifica di ogni singolo dato componente del profilo.  
**Ordine** - offre numerosi servizi quali: visualizzazione della data di effettuazione ordine e della spedizione, visualizzazione dell’importo totale, visualizzazione dei prodotti componenti acquistati, notifica della spedizione avvenuta, visualizzazione dei dettagli.  
**Prodotto** - offre la possibilità di: aggiungere prodotto al carrello, visualizzare attributi del prodotto (taglia, dimensione, numero ecc.), accesso alle recensioni e possibilità di aggiunta recensioni.  
**Recensione** - permette di accedere alle recensioni, visualizzarne i dettagli, l’autore, numero di stelle e descrizione.  
**Carrello** - permette di accedere alla pagina del carrello, visualizzarne i prodotti, modificarne le quantità e procedere al pagamento.

***Data Access***

**DataManagement** - si occupa della transazione dei dati, di garantire le proprietà ACID e per le operazioni dirette dal DBMS relazionale di MySQL.

**3.2** *Hardware/Software Mapping*

eSoccerce è una piattaforma che utilizza un’architettura Client-Server. Il Web Server è realizzato da Apache Tomcat ed è situato sulla stessa macchina su cui è implementato il database gestito da MySQL, eSoccerce. Il client è un Web Browser utilizzabile su qualsiasi dispositivo e la connessione tra client e server è stabilita mediante protocollo HTTP.



**3.3** *Persistent and Data Management*

La descrizione della gestione dei dati persistenti è presente nel documento “SDD DataManagement eSoccerce”.

**3.4** *Access Control and Security*

L’autenticazione al sistema viene effettuata attraverso l’inserimento di email e password, questo vale per ogni tipologia di utente presente all’interno della piattaforma. Gli utenti non registrati dispongono di limitate funzionalità, ad esempio essi sono impediti nella possibilità di acquistare un prodotto o di visualizzarne le recensioni. L’interfaccia per ogni tipologia utente è pressoché identica, fatta eccezione per l’amministratore, che, dovendo gestire la logistica dei prodotti e la loro categorizzazione, dispone di una categoria aggiuntiva apposita denominata “Amministratore” per l’aggiunta, rimozione, modifica scorte di un prodotto.  
La matrice degli accessi è situata all’interno del file “SDD AccessMatrix eSoccerce”.   
Per garantire la sicurezza dei dati dell’utente, la password è criptata mediante cifratura AES.

**3.5** *Global Software Control*

Il flusso di controllo globale del sistema che realizzeremo è un event-driven control. In questo controllo di flusso il sistema aspetta che si verifichi un evento esterno. Quando quest’ultimo si verifica, la richiesta viene inoltrata all’opportuno oggetto. Alla ricezione dei dati dal sistema, il flusso di controllo diventa di tipo procedure-driven nel quale la logica di business aspetta i dati richiesti per poterli elaborare.

**3.6** *Boundary conditions*

Non sono presenti boundary conditions di tipo rilevante nella nostra piattaforma poiché, l’accensione e lo spegnimento di quest’ultima viene effettuata mediante connessione del DB e successivamente il Web Container Tomcat.