



UNIVERSITÀ DEGLI STUDI DI SALERNO

System Design Document



Sommario

| | | |
|-------|--|----|
| 1 | Introduzione | 3 |
| 1.1 | <i>Proposta di Sistema</i> | 3 |
| 1.2 | Obiettivi di Design | 3 |
| 1.3 | Criteri di Affidabilità | 3 |
| 1.4 | Criteri di Performance | 4 |
| 1.5 | Criteri di Mantenimento..... | 4 |
| 1.6 | Criteri di End User..... | 5 |
| 1.7 | Definizione, Acronimi e abbreviazioni..... | 5 |
| 2 | Current Software architecture | 6 |
| 3 | Proposed software architecture..... | 6 |
| 3.1 | Overview..... | 6 |
| 3.2 | Subsystem decomposition..... | 7 |
| 3.3 | Schema Generale..... | 8 |
| 3.3.1 | Gestione Utente | 9 |
| 3.3.2 | Gestione Carrello | 11 |
| 3.4 | DataStorage: | 12 |
| 3.4.1 | Gestione Prodotti | 12 |
| 3.5 | Hardware/Software mapping..... | 13 |
| 3.6 | Persistent data management | 14 |
| 3.7 | Access control and security | 14 |
| 3.8 | Global Software control..... | 14 |
| 3.9 | Boundary conditions..... | 15 |
| 3.9.1 | Start-up | 15 |
| 3.10 | Terminazione | 15 |
| 3.11 | Fallimento..... | 15 |
| 4 | Servizi dei sottosistemi | 16 |

| Partecipanti | | |
|--------------|----------|-----------|
| Nome | Cognome | Matricola |
| Giulio | Di Maria | 03518 |
| Matteo | Volpe | 03656 |

1 Introduzione

1.1 Proposta di Sistema

L'obiettivo proposto da **Drive-In** è di creare un sito web nel quale qualunque tipo di persona può acquistare prodotti che vanno dalle scarpe alle borse. Il Sito è volte ad entrambi i sessi con scelta di prodotti che ricoprono tutti le attività giornaliere che possono andare dalla scelta di prodotti per il lavoro, per lo sport e per l'eleganza. Inoltre il sito **Drive-In** nasce con l'obiettivo di cercar di soddisfare ogni richiesta del cliente con prodotti delle migliori marche e con un occhio sul rispetto della natura. Il Sito prevede tra le sue funzionalità:

- Visualizzare i prodotti presenti nel catalogo andando a selezionare il prodotto che più rispecchia le caratteristiche che il cliente desidera utilizzando un menù veloce.
- Permettere di inserire elementi all'interno del carrello in modo tale da tener presente i prodotti che si intende acquistare.
- Un'interfaccia semplice e intuitiva adatto a persone che si interfacciano per la prima volta su un negozio online.

1.2 Obiettivi di Design

Il Sistema fornirà una struttura chiara, completa e semplice. Il Cliente non dovrà necessariamente effettuare operazioni che richiedono una conoscenza dell'applicazione, poiché l'utilizzo del sistema da parte del cliente sarà guidato dall'interfaccia semplice ed intuitiva.

L'interfaccia grafica sarà costruita in maniera molto semplice in modo tale da essere più comprensibile anche per gli utenti "alle prime armi", essa sarà curata nei minimi dettagli, mediante l'utilizzo di bottoni, label comprensibili e icone semplici e dettagliate, offrendo così al cliente un'esperienza di utilizzo rapida e semplificata.

Il Sistema proposto rispetterà i criteri di design di seguito elencati.

1.3 Criteri di Affidabilità

La Web Application garantirà il corretto funzionamento, gestendo vari tipi di errori che potranno verificarsi durante l'utilizzo di input non corretti o

violazione delle regole imposte sul sistema. Quindi, **Drive-In** rispetterà i seguenti requisiti di qualità, relativi all'affidabilità:

- **Robustezza:** Misura il grado con il quale il sistema riesce a reagire alle eccezioni o più in generali, alle situazioni di errore. Nel caso in cui l'utente inserisca dati errati nel sistema, quest'ultimo farà visualizzare dei messaggi di errore avvisando il Cliente che i dati non sono validi. (Requisito non Funzionale: Robustezza).
- **Affidabilità:** Il sistema garantisce il corretto svolgimento delle proprie funzionalità, producendo sempre l'output desiderato. (Requisito non Funzionale :Affidabilità).
- **Sicurezza:** L'accesso al sistema sarà controllato da un apposito sistema di autenticazione, che permetterà ad ogni utente di accedere alla proprio account senza modificare le operazioni altrui, garantendo la privacy. In questo modo, nessuno utente potrà visualizzare i dati sensibili e accedere alle funzionalità di altri utenti. (Requisito non Funzionale: Sicurezza)

1.4 Criteri di Performance

Il Sistema garantisce buone performance gestendo tutti gli utenti senza rallentamenti. **Drive-In** rispetterà i seguenti requisiti di qualità:

- **Tempo di risposta:** Il Sistema deve garantire, per le operazioni più frequenti(Login)un tempo di risposta di massimo 3 secondi (Requisito non Funzionale: Prestazione).
- Il Sistema dovrà garantire di poter acquistare un prodotto con al massimo 10 click (Requisito non Funzionale: Prestazione).

1.5 Criteri di Mantenimento

Il sistema sarà facilmente mantenibile e rispetterà i seguenti requisiti:

- **Tracciabilità dei requisiti:** grazie ad una buona e coerente documentazione, sarà facile risalire ai rispettivi requisiti funzionali,

use case e altri artefatti, a cui faranno riferimento le varie classi e metodi del sistema.

1.6 Criteri di End User

Dal punto di vista dell'utente, il sistema dovrà garantire i seguenti requisiti di qualità:

- **Utilità:** Attraverso la raccolta dei requisiti, il sistema sarà in grado di soddisfare le esigenze degli utenti.
- **Usabilità:** Il sistema dovrà essere intuitivo e di semplice utilizzo. Non sarà necessario l'uso di un manuale utente per compilare le azioni. (Requisito non Funzionale: Usabilità).

1.7 Definizione, Acronimi e abbreviazioni

| Acronimi | Descrizioni |
|----------|-------------------------------|
| Rad | Requirement Analysis Document |
| SDD | System Design Document |
| GUI | Graphical User Interface |
| DBMS | Database Management System |
| SQL | Structured Query Language |
| HW | Hardware |
| SW | Software |

1.1. Riferimenti

- Drive-In_RAD

1.2. Overview

Il Seguento documento è strutturato in quattro parti:

- Introduzione: riporta una descrizione del sistema specificando la ragione del suo sviluppo, le caratteristiche del sistema e un accenno sull'utilizzo delle funzionalità offerte.
- Proposta Architettura Software: fornisce una panoramica sull'architettura utilizzata per il sottosistema. I punti riportati per l'architettura del sistema proposto sono: Decomposizione del sistema in sottosistemi, il mapping Hardware e Software, la gestione dei dati persistenti, gestione del controllo del flusso globale, il controllo degli accessi di sicurezza e le condizioni limite.
- Servizi dei sottosistemi: espone una descrizione di un sottosistema identificato esplicitando i servizi offerti da ogni sottosistema.
- Glossario: è una raccolta di termini utilizzati nella stesura dei documenti citati in questo documento.

2 Current Software architecture

L'architettura proposta di seguito non andrà a sostituire nessun sistema preesistente. La progettazione e lo sviluppo del software inizierà da zero e seguirà i criteri della Greenfield Engineering.

3 Proposed software architecture

3.1 Overview

L'architettura scelta per il sistema da realizzare sarà quella MVC. L'utente generico potrà interagire con il Controll layer mediante View layer che offrirà diverse interfacce in base alle necessità dell'utente. Il Controll

layer dovrà comunicare con il database per la memorizzazione dei dati persistenti. Sul server, risiede un DBMS che si occupa di recuperare, memorizzare ed interrogare i dati presenti nel database, elaborando, quindi, la richiesta dell'utente. L'aspetto della concorrenza di accessi multipli al database, sarà gestito dal DBMS stesso che dovrà evitare eventuali “colli di bottiglia”. Tale architettura conferisce all'intero sistema una maggiore manutenibilità e permette di gestire il problema della concorrenza degli accessi ai dati in maniera semplice ed efficace.

3.2 Subsystem decomposition

I tre livelli relativi all'architettura adottata sono:

- View layer
- Controll layer
- Model layer

❖ View layer:

- Si occupa di gestire l'interfaccia utente; include gli oggetti boundary con cui l'utente interagisce (finestre, form, pagina web, ecc...)

❖ Control layer:

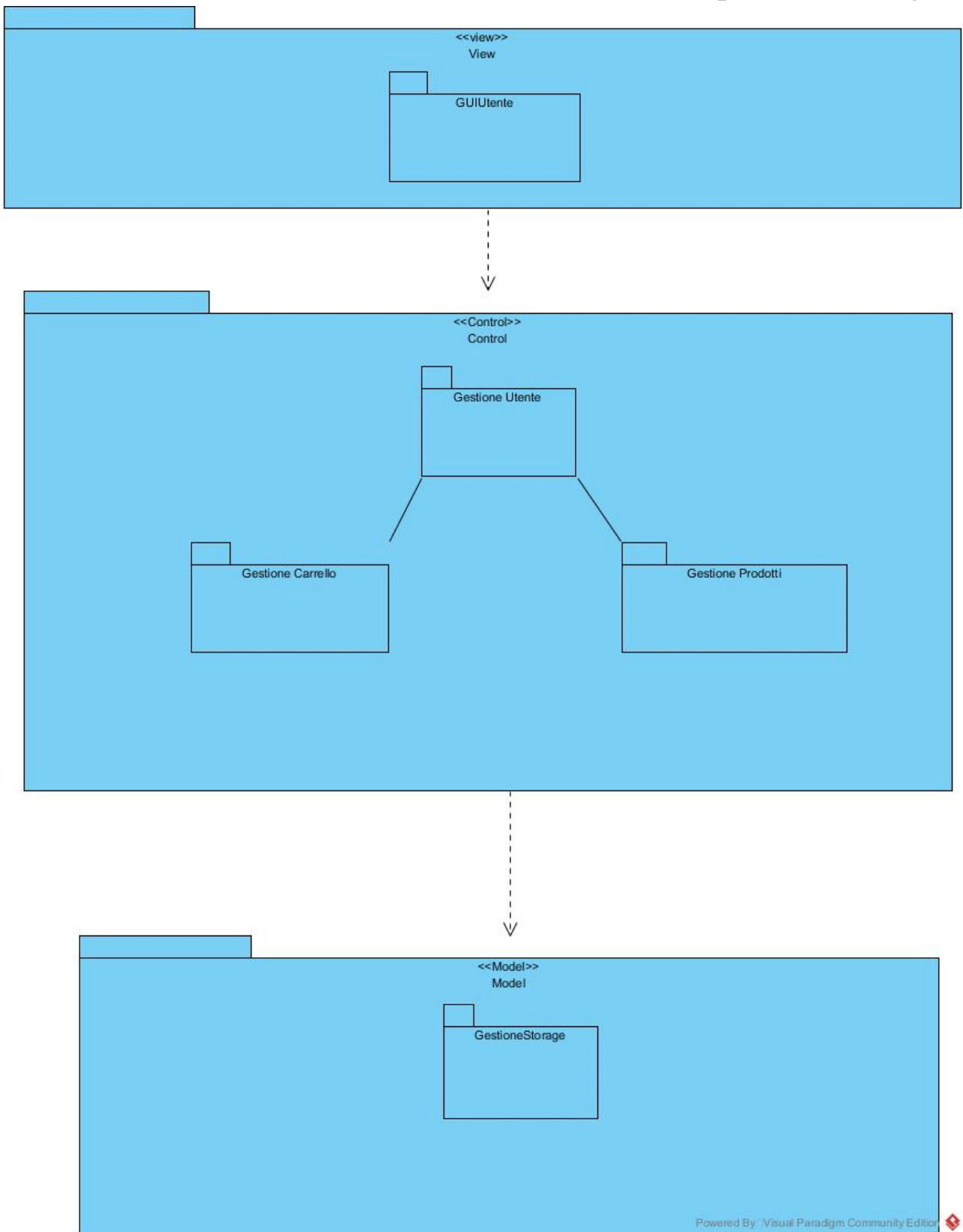
- Comprende le componenti logiche (oggetti control e entity), responsabile del corretto funzionamento del sistema, e utilizza il database per effettuare operazione sui dati.

❖ Model layer:

- Comprende un gestore che si occupa di effettuare operazioni di inserimento, cancellazione e aggiornamento e si occupa di rendere disponibili i dati presenti all'interno del database.

3.3 Schema Generale

Di seguito è riportato lo schema generale della suddivisione. Successivamente verranno descritti i sottosistemi più nel dettaglio.

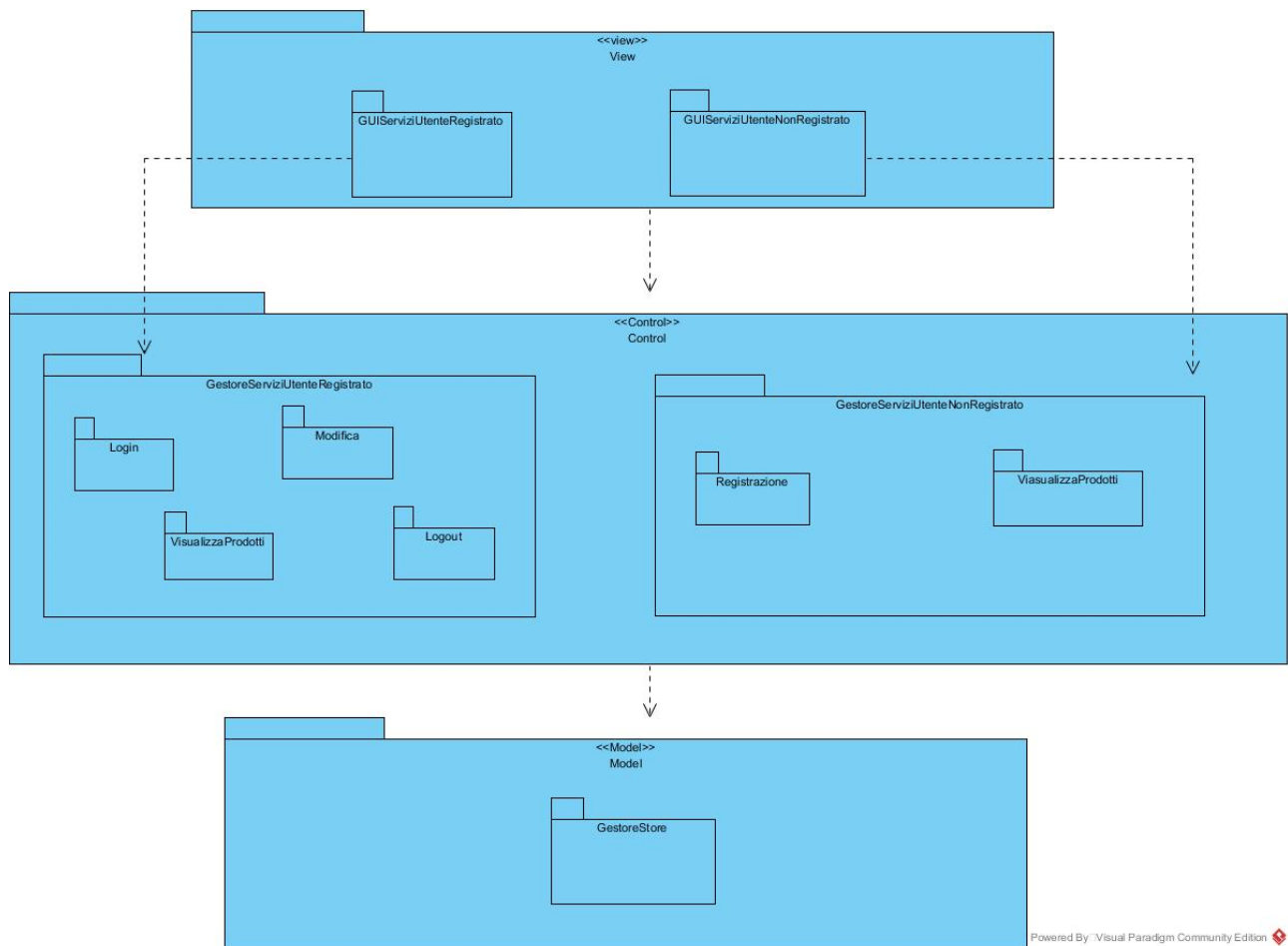


View: GUIUtente, indicano i sottosistemi che contengono tutti gli oggetti Boundary.

Control: Contiene i tre sottosistemi individuati (GestioneUtente, GestioneProdotti, GestioneCarrello).

Model: sottosistema che ha il compito di effettuare operazioni verso il database.

3.3.1 Gestione Utente



View:

Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per l'accesso a Drive-In accessibile da parte di uno o più Utenti Generici.

❖ **GUIServiziUtenteRegistrato:** comprende tutte le interfacce dei servizi offerti all'UtenteRegistrato.

❖ **GUIServiziUtenteNonRegistrato**: comprende tutte le interfacce dei servizi offerti all'UtenteNonRegistrato.

Control:

Il sottosistema è decomposto ulteriormente in altri due sottosistemi:

❖ **GestioneServiziUtenteRegistrato**: sottosistema che contiene le funzionalità di base offerte dal sistema come:

- *Login*: Operazione che permette l'autenticazione all'Utente_Registrato.
- *Logout*: Operazione che permette all'Utente_Registrato di scollegarsi dal sistema.
- *Visualizza Prodotti*: Operazione che permette all'Utente_Registrato di visualizzare i prodotti presenti nel sito.
- *Modifica*: Operazione che permette di modificare i dati personali dell'Utente_Registrato.

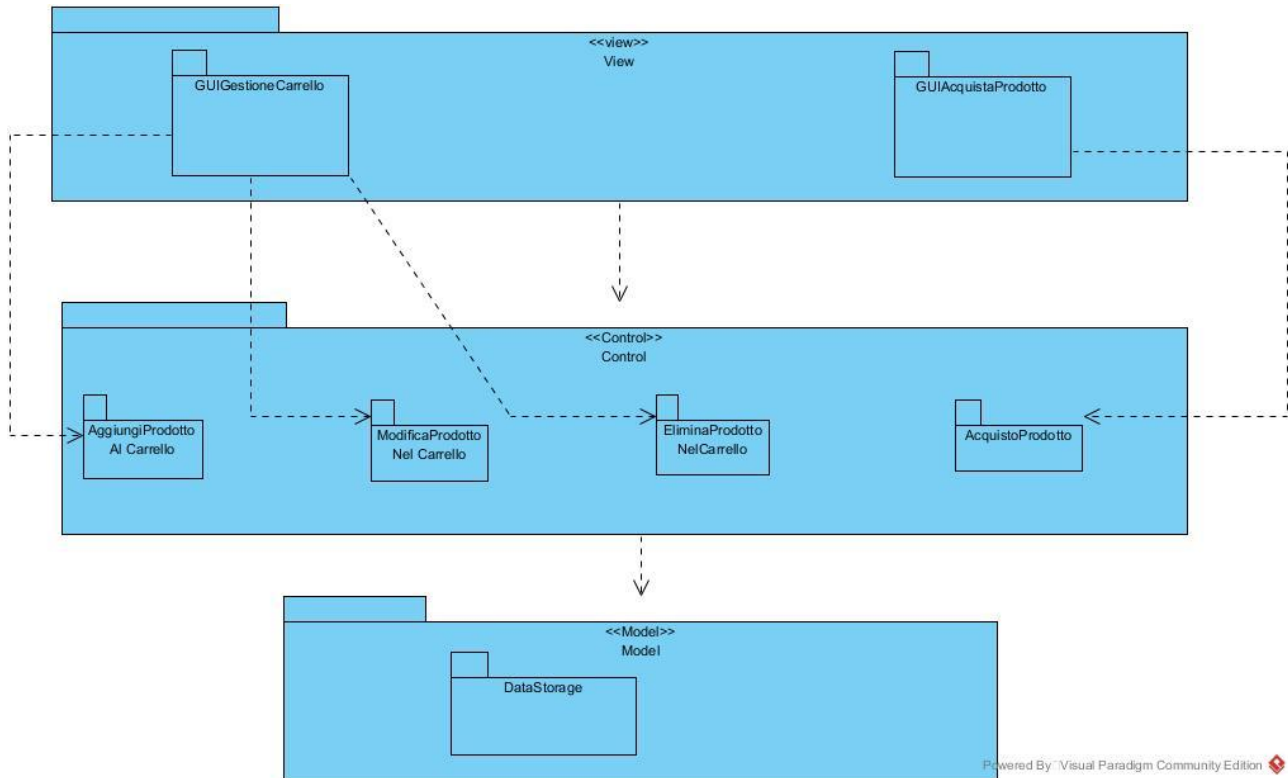
❖ **GestioneServiziUtenteNonRegistrato**: sottosistema che contiene le funzioni per far parte del sistema:

- *Registrazione*: Operazione che permette all'UtenteNonRegistrato di registrarsi.
- *Visualizza Prodotti*: Operazione che permette all'Utente_RegistratoNonRegistrato di visualizzare i prodotti presenti nel sito.

Model:

Comprende il **GestioneStorage** che permette di effettuare operazioni sul database.

3.3.2 Gestione Carrello



View: Include tutte le componenti dell'interfaccia grafica del sistema che offrono funzionalità per la gestione del prodotto e dell'acquisto.

- **GUIGestioneCarrello:** interfaccia che permette di aggiungere, modificare ed eliminare un prodotto all'interno del carrello.
- **GUIAcquistaProdotto:** interfaccia per l'acquisto di un prodotto

Control: comprende tutte le componenti logiche associate alle gestione di un prodotto.

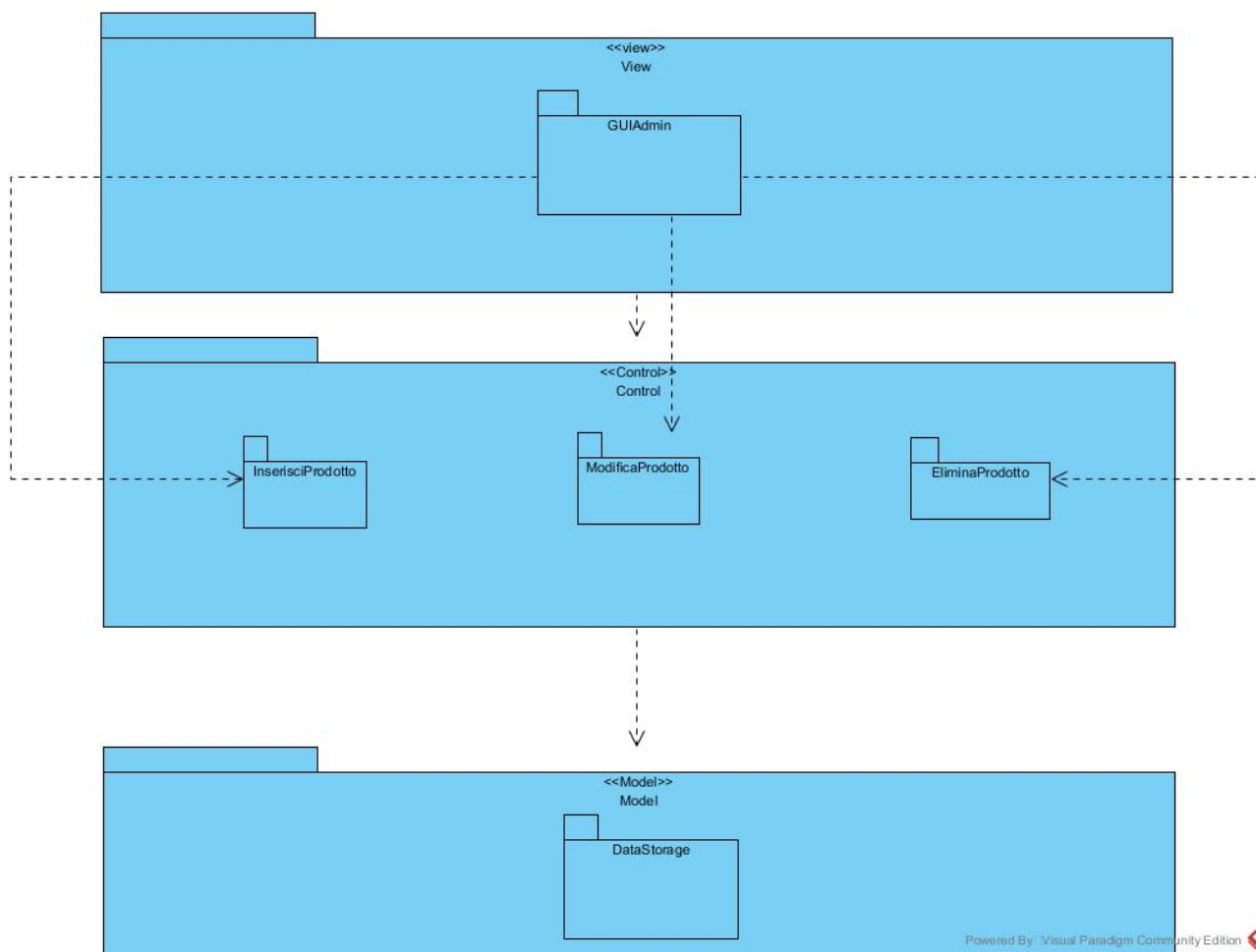
- *Aggiungi Prodotto Al Carrello:* operazione che permette l'inserimento di un prodotto all'interno del carrello.
- *Modifica Prodotto nel Carrello:* operazione che permette di modificare un prodotto all'interno del carrello.
- *Elimina Prodotto nel Carrello:* operazione che permette di eliminare un prodotto all'interno del carrello.

- *Acquisto Prodotto*: Permette di acquistare tutti i prodotti all'interno del carrello.

3.4 DataStorage:

Comprende il GestoreStorage che permette di effettuare operazioni sul database.

3.4.1 Gestione Prodotti



View: Include tutte le componenti dell'interfaccia messe a disposizione all'Admin-Prodotti.

- **GUIAdmin:** tramite questa interfaccia, l'Admin-prodotti può accedere alle funzionalità solo a lui disponibili.

Control: comprende tutte le attività svolte dall'Admin-Prodotti:

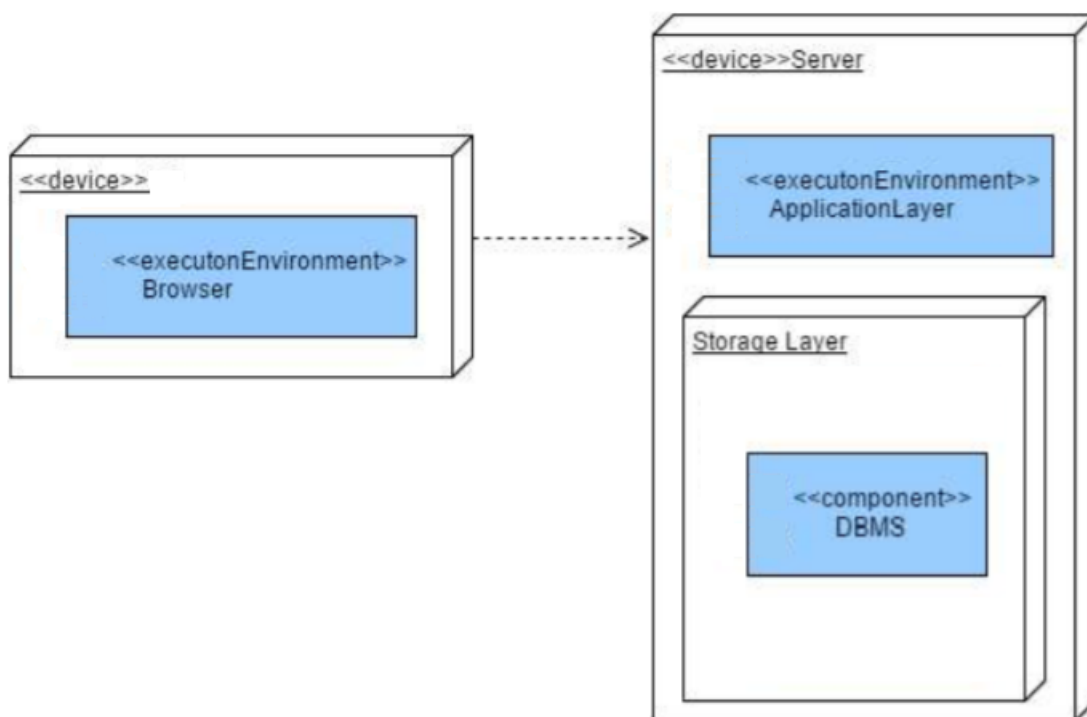
- *Inserisci Prodotto:* Permette di inserire un prodotto all'interno del sistema
- *Modifica Prodotto:* Permette di modificare le caratteristiche di un prodotto presente all'interno del sistema.
- *Elimina Prodotto:* Permette di modificare un prodotto da catalogo del sistema.

Model:

Comprende il **GestoreStorage** che permette di effettuare operazioni sul database.

3.5 Hardware/Software mapping

Il sistema sviluppato sarà installato su un solo server e utilizzerà un DBMS MySQL stanziato sullo stesso per la gestione dei dati persistenti. Il sistema sarà diviso in un'architettura client e server.



Il Deployment Diagram mostra le componenti che utilizzerà il nostro sistema; questo diagramma aiuta gli sviluppatori a comprendere le relazioni tra le componenti software e i nodi hardware. Al lato client, l'interfaccia utente verrà mostrata sul browser web ed interagirà con l'application layer, che a sua volta memorizza e interroga i dati presenti sul database.

3.6 Persistent data management

Il sistema userà un servizio di storage su Database. Le risorse saranno opportunamente salvate in cartelle sul server.

La scelta del DBMS è ricaduta su MySQL viste le conoscenze dei team member. La scelta di un database relazionale, rispetto ad un database object-oriented, è motivata dalle alte prestazioni offerte dai primi.

Le informazioni per tener traccia degli utenti registrati, dei prodotti inseriti nei carrelli, e dei prodotti acquistati con i relativi ordini sono salvati all'interno del database.

3.7 Access control and security

Il controllo degli accessi sarà alla base della sicurezza del nostro Sistema, permettendo ad ogni cliente o amministratore di collegarsi al sistema tramite l'utilizzo di username e password, che verranno richieste ad ogni singolo accesso. La sessione termina quando l'utente effettua il logout.

Nel caso l'accesso al sistema non abbia successo, verrà inviata una notifica di fallimento indicando che è avvenuto un errore nell'inserimento di username e password e consentendo all'utente di effettuare un nuovo tentativo.

Il visitatore (utente non registrato) potrà usufruire dei servizi che non utilizzano la registrazione visualizzando un invito a registrarsi o loggarsi per i servizi che la richiedano.

3.8 Global Software control

Per quanto riguarda il controllo del flusso, utilizzeremo un controllo centralizzato di tipo Event Driver, in quanto il controllo risiede in un dispatcher che chiama le funzioni mediante callback.

3.9 Boundary conditions

3.9.1 Start-up

Per il primo start-up del sistema “Drive-In”, è necessario l’avvia di un web Server che fornisca il servizio di un Database MySQL per la gestione dei dati persistenti e l’interpretazione ed esecuzione del codice lato server, assumendo che prima dello start-up iniziale il database abbia almeno un amministratore che una volta loggato può accedere a tutte le funzionalità che gli offre il sistema.

3.10 Terminazione

Al momento della chiusura della pagina web l’utente rimane connesso e la sua sessione decade solamente al momento in cui deciderà di effettuare il logout.

3.11 Fallimento

Un esempio di un possibile fallimento generato da parte del sistema:

Malfunzionamento del server dovuto ad un improvviso guasto al database. La priorità del sistema nell’affrontare questo tipo di problema è quella di perdere meno dati possibili e segnalare immediatamente all’utente che sta facendo uso del sistema che vi sono delle anomalie, e nel caso di fare appello alla sezione contatti per segnalare il problema ai moderatori del sito.

Alcuni esempi di possibili fallimenti generati da parte del Cliente:

Il cliente potrebbe sbagliare nel riempimento dei campi per il login, in tal caso il sistema provvederà opportunamente a segnalare le informazioni che sono state mancate all’interno del form.

Il cliente potrebbe anche provare ad inserire un prodotto all’interno del carrello senza però aver effettuato il login, in tal caso il sistema comunicherà all’utente di effettuare il login o la registrazione per proseguire.

Alcuni esempi di possibili fallimenti generati da parte dell'Amministratore:

L'amministratore potrebbe provare ad aggiungere un prodotto che esiste già. In tal caso il sistema sarà pronto a segnalare questa eventualità, mostrando un alert al video che avvisa l'amministratore.

L'amministratore potrebbe provare a modificare un prodotto in modo che quest'ultimo si identifichi come un prodotto che è già presente all'interno del sistema. In tal caso il sistema sarà pronto a segnalare questa eventualità, mostrando un alert al video che avvisa l'amministratore.

4 Servizi dei sottosistemi

Il modulo di **Gestione Utente** offre i seguenti servizi:

| Nome servizio | Parametri | Descrizione |
|---------------------|--|--|
| Logout | IdSessione | Permette di invalidare la sessione |
| Login | Username,Password | Permette di accedere al sistema con le proprie credenziali |
| Modifica Profilo | Username,DatoDaModificare,DatoDaInserire | Permette di modificare un informazione personale. |
| Visualizza prodotti | Lista Prodotti | Permette la visualizzazione di un determinato gruppo di prodotti |
| Registrazione | Username,Password,Nome,Cognome, Carta | Permette di registrarsi al sistema come cliente |

Il modulo di **Gestione Carrello** offre i seguenti servizi:

| Nome servizio | Parametri | Descrizione |
|----------------------------------|--|--|
| Aggiungi Prodotto al Carrello | Codice, Colore,Taglia,Quantità, Username | Permette di inserire un prodotto all'interno del carrello |
| Modifica Prodotto dal carrello | Username,Codice,Colore,Taglia, NuovaQuantità | Permette di modificare la quantità di un prodotto che si desidera avere all'interno del carrello |
| Elimina il Prodotto dal carrello | Username,Codice,Colore,Taglia | Permette di eliminare un prodotto dal carrello che era stato precedentemente inserito. |
| Acquista prodotto | Numero ordine, lista prodotti (con i relativi identificativi e quantità) | Permette di effettuare un ordine al fine dell'acquisto |
| Registrazione | Username,Password,Nome,Cognome, Carta | Permette di registrarsi al sistema come cliente |

Il modulo di **Gestione Prodotto** offre i seguenti servizi:

| Nome servizio | Parametri | Descrizione |
|----------------------|---|--|
| Inserisci Prodotto | Codice, Colore,Taglia,Disponibilità, TipoProdotto, Sesso, Nome, NomeImmagine | Permette di inserire un prodotto all'interno del sistema |
| Modifica Prodotto | Codice, Colore,Taglia, DatoDaModificare, Valore da inserire | Permette di modificare un dato di un prodotto all'interno del sistema |
| Elimina il Prodotto | Codice,Colore,Taglia | Permette di eliminare un prodotto dal sistema. |