

Progetto IS – D&D

SDD – System Design Document



TEAM – DAVIDE ALBANESE, DAVIDE SCUTIERO

INDICE

- 1. Introduzione
 - 1.1.Scopo del Sistema
 - 1.2.Obiettivi di progettazione
- 2. Architettura software attuale
- 3. Architettura software proposta
 - 3.1.Overview
 - 3.2.Decomposizione in sottosistemi
 - 3.3.Mapping hardware/software
 - 3.4.Gestione dei dati persistenti
 - 3.5.Controllo e sicurezza degli accessi
 - 3.6.Global Control Flow
 - 3.7.Boundary Conditions

1. Introduzione

1.1 Scopo del sistema

D&D è un sito web e-commerce dove è possibile acquistare prodotti elettronici, ricambi elettronici, smartphone, tablet e tanto altro. Il suo scopo è di essere accessibile a chiunque, in qualsiasi momento, garantendo una navigazione intuitiva, sicura ed efficiente ma soprattutto veloce (N.d.R.: come già ampiamente descritto nel Problem Statement), infine, lo scopo reale è quello di soddisfare in tutto e per tutto quelle che sono le necessità primarie degli utenti, garantendo assistenza e supporto tecnico per qualsiasi tipo di problema dovesse presentarsi (area utente, riepilogo ordine, fatturazione, ecc..). I prodotti sono facilmente acquistabili da un

catalogo dagli appositi bottoni, ma tra le cose fondamentali del web developing troviamo anche la possibilità di aggiungere al carrello dei prodotti anche non essendo registrati, ciò da avere una miglior visione della situazione al utente.

1.2 Obiettivi di progettazione

Gli obiettivi di progettazione li classifichiamo da una lista di criteri, suddivisi in cinque gruppi:

- 1. Criteri Performance**
- 2. Affidabilità**
- 3. Costi**
- 4. Mantenimento**
- 5. Criteri End User**

Performance

Tempo di risposta

Il sistema risponderà alle richieste impartite dai componenti in tempo reale. Ogni pagina può impiegare un tempo di risposta diverso, che dipende sostanzialmente dal peso della pagina. Se la pagina dei prodotti ricercati da un client è di grandi dimensioni, il sistema potrebbe visualizzarlo con leggero ritardo (in quantità di tempo brevi secondi) rispetto ad una pagina di peso più leggero.

Per quanto riguarda l'iscrizione dell'utente, dopo la sottomissione dei dati data dalla richiesta del cliente venga data una risposta (utente registrato con successo o meno) in non più di 24 ore.

Throughput

L'utente effettuerà l'acquisto in breve tempo, semplicemente selezionando il prodotto richiesto dal catalogo concorrente, accedendo al proprio account fino a procedere il checkout.

Memoria

La dimensione della memoria dipende dalla grandezza del Database, quindi è dinamica.

Affidabilità

Robustezza

L'iscrizione dell'utente sarà attuata una attenta verifica di attendibilità dei dati immessi dall'utente come controllo del codice fiscale per verificare la reale esistenza della persona fisica.

Tolleranza ai fault

I risultati dati dalle pagine dinamiche, Servlet, riguardo la disponibilità o meno dei prodotti presenti in quell'istante nel magazzino. Se un prodotto non è disponibile non verrà visualizzato come tale ma bensì "Non disponibile al momento, prenotalo!".

Disponibilità

Una volta che il sistema è stato realizzato sarà disponibile ogni qualvolta che un utente ne richiede l'utilizzo.

Sicurezza

La sicurezza è garantita nei controlli gestiti da username e di password.

Mantenimento

Estendibilità

Al sistema è concessa la possibilità di subire un Upgrade, o meglio aggiungere in futuro nuove funzionalità oppure creare nuove classi con la capacità di estendere quelle già esistenti.

Modificabilità

In caso di crescita esponenziale del sistema per quanto riguarda le utenze, se ad esempio il numero di utenze rispetto a quello iniziale è elevato ci sarà un ampliamento.

Il sistema è gestito da un server 24h su 24h.

Adattabilità:

Il sistema può essere facilmente portato su diversi domini di applicazione.

Portabilità

Il sistema è gestito da un software che sarà scritto usando il linguaggio di programmazione JSP per conformarsi alle richieste date. Non ci saranno vincoli sull'hardware. Il sistema è portabile su diverse piattaforme.

Leggibilità

Attraverso la lettura del codice il sistema è comprensibile. Le istruzioni sono semplici e consentono di comprendere le funzionalità del sistema.

Criteri End User

Utilità

Il sistema seguirà passo-passo le procedure per supportare e aiutare l'utente nelle funzioni al momento disponibili. L'utente non avrà problema nel proseguimento delle funzionalità.

Usabilità

Il cliente può accedere al sistema facilmente, tutte le funzionalità come iscrizione, login, visualizzazione di dati sarà altrettanto facile.

Quando vuole iscriversi il client, gli viene richiesto di inserire i suoi dati personali all'interno di una form. Un utente dopo aver inserito il suo username e la sua

password, può svolgere tutte le attività del sistema come proseguire nella conferma di un ordine.

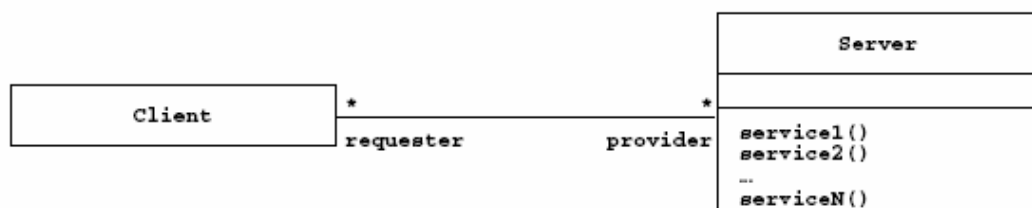
2. Architettura software attuale

Il sistema è implementato dal nulla in quanto nessun sistema esiste in precedenza, per cui questa fase è di “Greenfield Engineering”. La raccolta dei requisiti viene fatta interagendo con il cliente e cercando di estrarre i requisiti dalle sue richieste.

3. Architettura software proposta

3.1 Overview

Il sistema offerto è un sito e-commerce contenente un database contenente l’archivio di tutti i prodotti e utenti. Gli utenti interagiscono con il sistema centrale con la possibilità di effettuare acquisti dalle loro postazioni client; I client non possono interagire tra di loro, quindi le informazioni al database o dei documenti riguardante l’utente dovranno sempre essere fatte al server principale. Ciò comporta nella composizione del sistema offerto, e l’architettura più indicata ovvero del tipo Client-Server.



3.2 Decomposizione in sottosistemi

L'architettura proposta è possibile suddividerla grazie alla decomposizione a strati, quindi nel nostro caso utilizzando 3 strati:

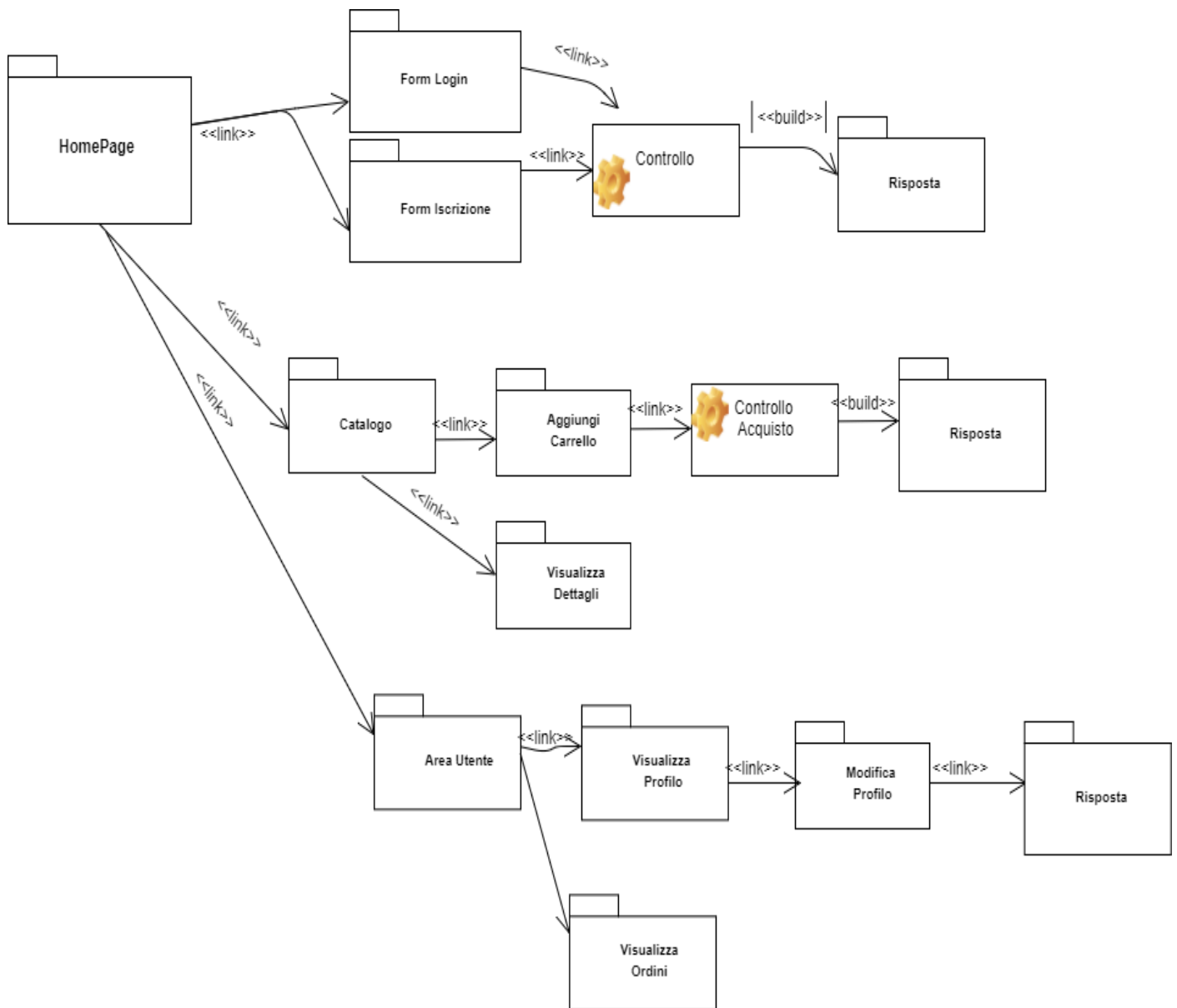
1. **InterfaceLayer**
2. **LogicLayer**
3. **StorageLayer**

InterfaceLayer si occupa di gestire l'interfaccia utente e racchiude gli oggetti boundary che interagiscono con l'utente (button, form, etc.).

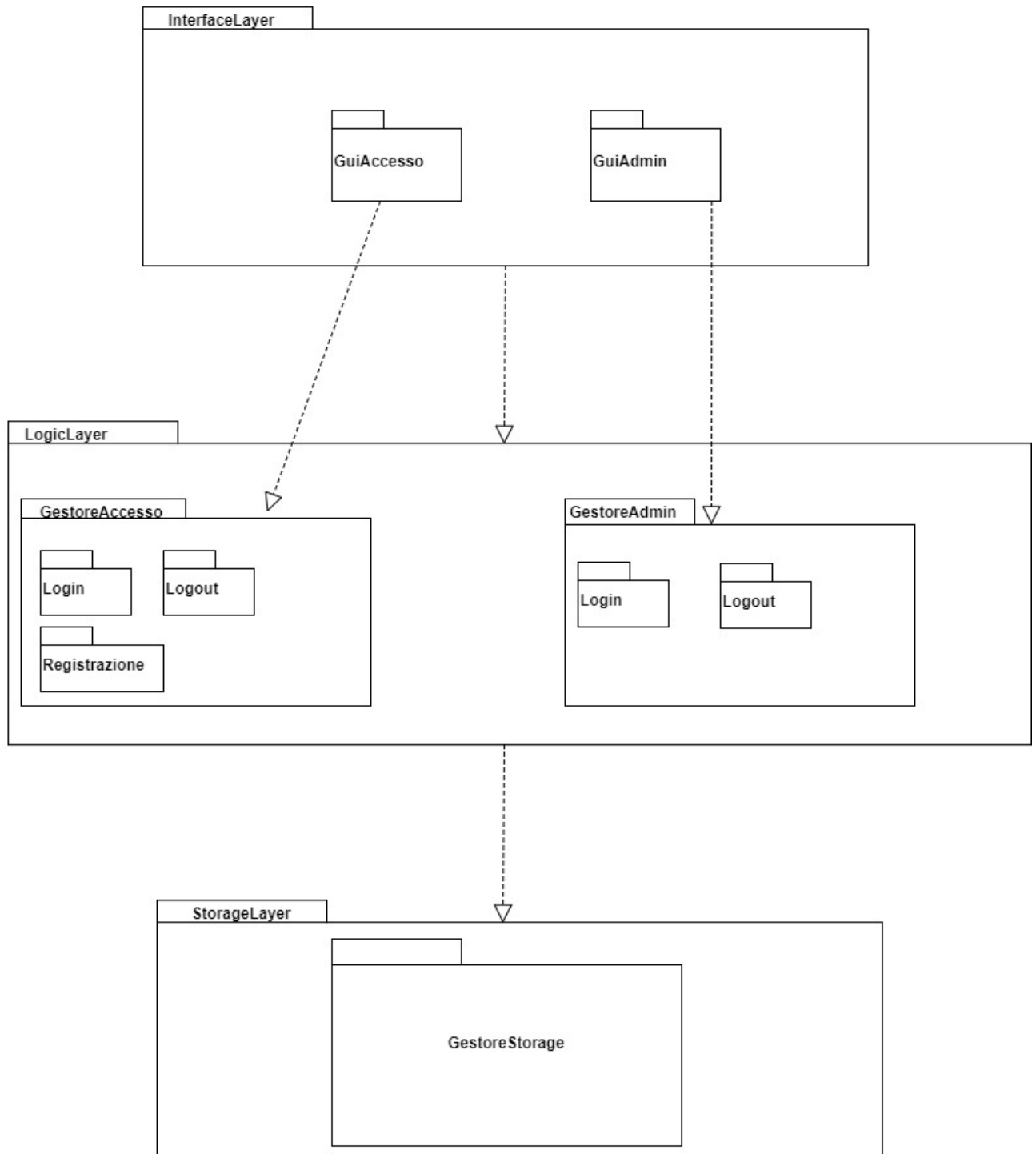
LogicLayer comprende le varie componenti logiche (control, entity) , che hanno il compito di far funzionare il sistema al meglio , utilizza il database per le operazioni da effettuare.

StorageLayer all'interno di questo layer troviamo un gestore che si occupa di effettuare le varie operazioni come inserimento, modifica e cancellazione ma soprattutto ha il compito di rendere disponibili i dati persistenti.

Schema generale rappresentante il lavoro del sistema con sottosistemi.



3.2.1 Sottosistema Gestione Autenticazione



InterfaceLayer comprende tutte le interfacce grafiche utili al cliente per l'accesso al sistema.

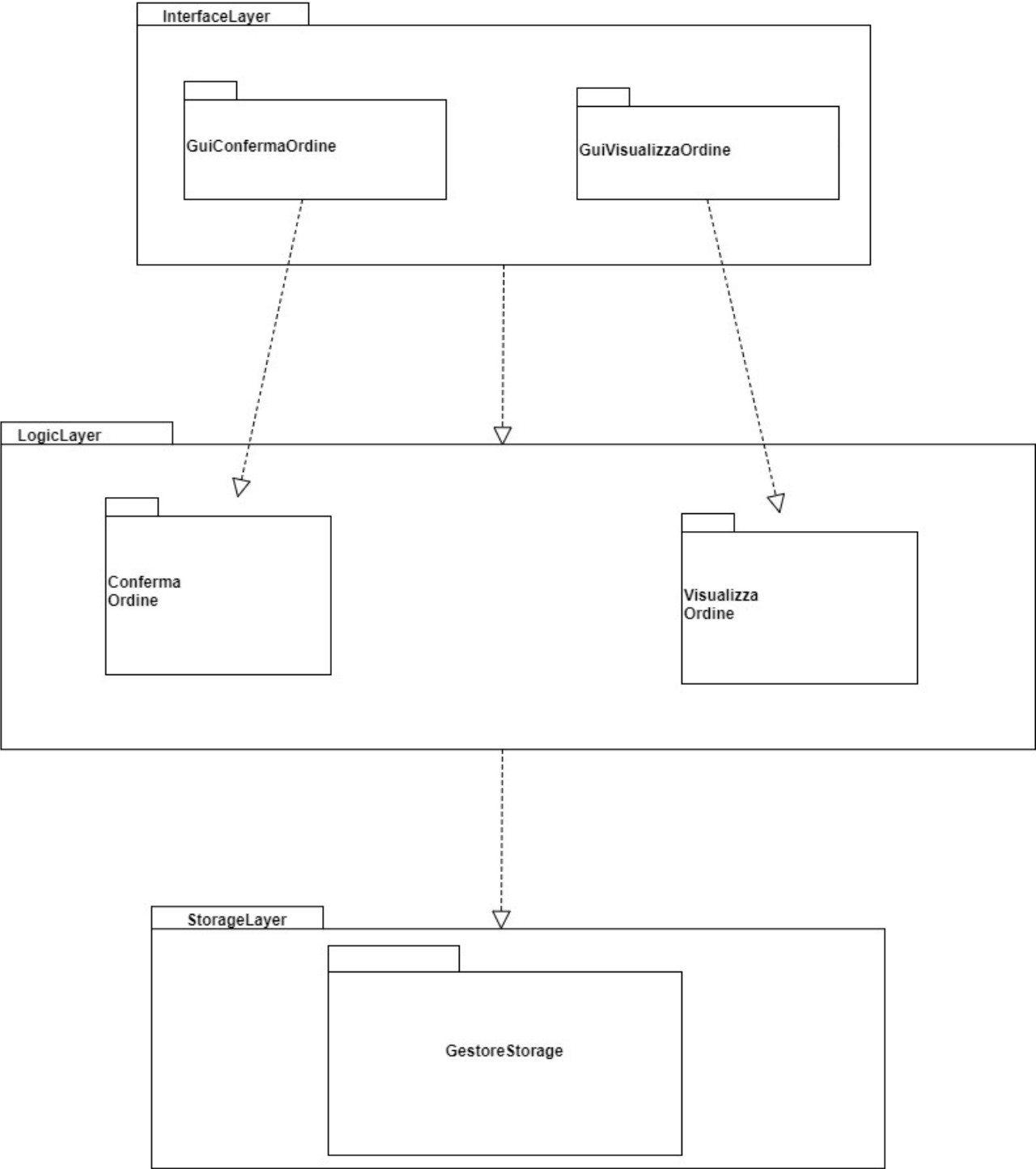
- **GuiAccesso** comprende l'interfaccia grafica per autenticarsi al sito.
- **GuiAdmin** tramite quest'interfaccia l'admin ha la possibilità di accedere al sito.

LogicLayer qui il sistema viene decomposti in altri due sottosistemi:

- **GestoreAccesso** sottosistema che contiene le funzionalità per l'accesso al sistema come:
 - *Registrazione*: operazione che permette all'utente di registrarsi
 - *Login*: operazione che permette l'autenticazione al sistema
 - *Logout*: operazione che permette all'utente di uscire e scollegarsi dal sistema
- **GestoreAdmin** sottosistema che comprende l'accesso al sistema dell'amministratore:
 - *Login*: operazione che permette l'autenticazione dell'amministratore al sistema
 - *Logout*: operazione che permette all'amministratore di uscire e scollegarsi dal sistema

StorageLayer comprende il **GestoreStorage** che permette di effettuare operazioni sul database.

3.2.2 Sottosistema Gestione Acquisto



InterfaceLayer comprende tutte le interfacce grafiche utili all'utente per il completamento di un acquisto.

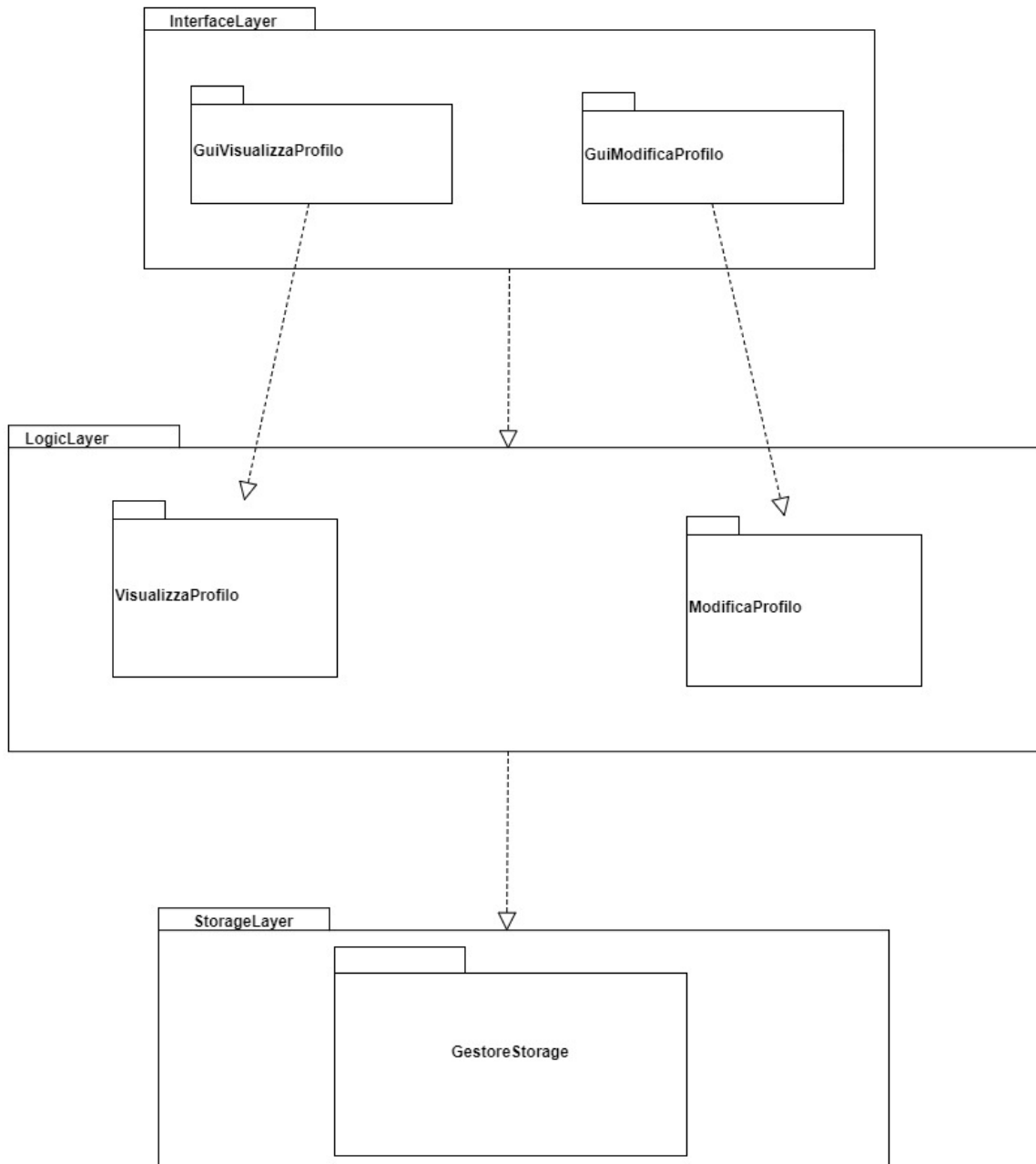
- **GuiConfermaOrdine** comprende l'interfaccia grafica per la conferma dell'ordine.
- **GuiVisualizzaOrdine** interfaccia grafica che permette all'utente di visualizzare gli eventuali ordini completati.

LogicLayer comprende le componenti logiche associate al completamento di un ordine:

- *ConfermaOrdine*: operazione che permette di confermare un ordine
- *VisualizzaOrdine*: operazione che permette di mostrare gli ordini

StorageLayer comprende il **GestoreStorage** che permette di effettuare operazioni sul database.

3.2.3 Sottosistema Gestione Profilo



InterfaceLayer comprende tutte le interfacce grafiche utili all'utente per i vari servizi della propria area utente.

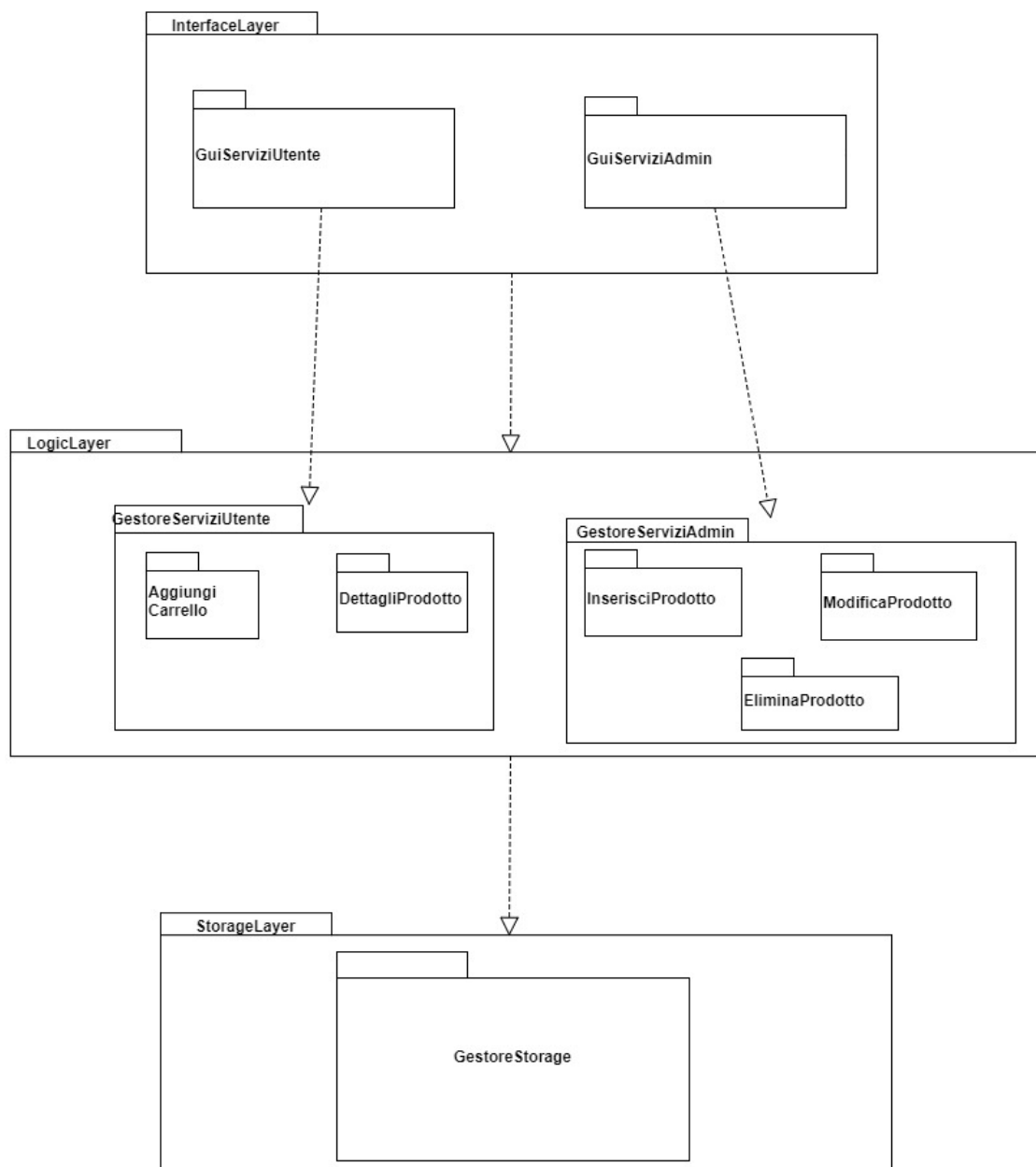
- **GuiVisualizzaProfilo** comprende l'interfaccia grafica per visualizzare l'area dell'utente stesso.
- **GuiModificaProfilo** interfaccia grafica che permette la modifica dei dati personali all'utente.

LogicLayer comprende le componenti logiche associate all'area utente.

- *VisualizzaProfilo*: operazione che permette la visualizzazione del profilo
- *ModificaProfilo*: operazione che permette di modificare i propri dati.

StorageLayer comprende il **GestoreStorage** che permette di effettuare operazioni sul database.

3.2.4 Sottosistema Gestione Prodotto



InterfaceLayer comprende tutte le interfacce grafiche utili al cliente per la guida all'acquisto di un prodotto.

- **GuiServiziUtente** comprende le interfacce grafiche con le varie funzionalità sui prodotti per quanto riguarda l'utente.
- **GuiServiziAdmin** tramite quest'interfaccia l'admin accede a tutte le funzionalità disponibili ad esso.

LogicLayer qui il sistema viene decomposto in altri due sottosistemi:

- **GestoreServiziUtente** sottosistema che contiene le funzionalità per i prodotti.
 - *Aggiungi al carrello*: operazione che permette di aggiungere prodotti dal catalogo al carrello.
 - *Visualizza dettagli*: operazione che permette di aprire una pagina con i dettagli del prodotto selezionato.

- **GestoreServiziAdmin** sottosistema che comprende le funzionalità disponibili per i prodotti all'amministratore.
 - *Inserimento Prodotto*: operazione che consente di inserire nuovi prodotti o già esistenti.
 - *Modifica Prodotto*: operazione che permette la modifica di un prodotto già esistente.
 - *Cancellazione Prodotto*: operazione che permette la cancellazione di un prodotto già esistente.

StorageLayer comprende il **GestoreStorage** che permette di effettuare operazioni sul database.

3.3 Mapping hardware/software

Per il sistema, basato su un'architettura distribuita client/server, sono state scelte questi tipi

di configurazioni:

PIATTAFORMA: Qualsiasi che supporta un browser java2 compatibile (client)
Windows NT/2000/XP con almeno 128mb di ram (server)

COMPUTAZIONE: Monoprocessore (server) almeno un i586 (pentium) intel 100% compatibile intel con 128MB per WinNT/2000 e 256MB per WinXP

CONNETTIVITA: Tasso di trasmissione standard (minimo 56 kbps)

protocollo: HTTP, TCP/IP

Interazioni: Sincrone, eccetto per la conferma dell'iscrizione dell'utente (il cui tempo di risposta è previsto entro le 24 ore).

COMPONENTI:

Memorizzazione dati: DBMS Borland Interbase requisiti minimi: JSDK 1.1 e 10MB circa liberi sul disco

Interazione: CLIENT-SERVER/APACHE TOMCAT v.5 requisiti minimi:

processore i586 compatibile intel 100%, JSDK 1.4.x e 50MB circa liberi sul disco

Piattaforma virtuale: JSDK 1.4.2 requisiti minimi: Pentium 166MHz o

microprocessore più veloce con almeno 48MB di RAM, raccomandati per

applicazioni avviate all'interno di un browser che usa il Java Plug-in e con 84MB circa liberi sul disco.

3.4 Gestione dei dati persistenti

“D&D” sfrutta l’uso di un database di tipo relazionale, il quale interagisce tramite l’utilizzo di un DBMS. La scelta del DBMS è stata attuata secondo i criteri offerti ovvero:

- Sicurezza
- Affidabilità
- Facilità di gestione(multiutente)
- Velocità di accesso e trasmissione dati (query di ricerca)
- Garanzia di coerenza

Possiamo identificare tra i diversi oggetti Entità estratti dagli use-case, dati persistenti in diversi scope:

❖ **Dati persistenti a livello sessione** che si perdono una volta che il client chiude l’applicazione:

- AcquistaProdotto (il prodotto acquistato dal cliente)
- Login (oggetto relativo all’autenticazione nel sistema)
- RegistrazioneUtente(oggetto relativo alla registrazione del cliente)

❖ **Dati persistenti a livello applicazione** (dati presenti anche se il server non è disponibile):

- areaUtente (informazioni sull’utente)
- Product_Catalog (elenco prodotti) –
- Checkout_View (storico vendite degli utenti)

3.5 Controllo e sicurezza degli accessi

L’accesso sicuro viene garantito tramite l’utilizzo di username e password per ogni singolo utente, che tali dati verranno richiesti ad ogni singolo accesso al sistema. Per ogni singolo utente/amministratore si hanno a disposizione numerose interfacce grafiche per l’accesso in modo tale da dare ogni rispettiva interfaccia che rientra

nella sua categoria di utenza. L'utente può autenticarsi tramite una form apposita nella home page, dove inserisce i dati richiesti, e tramite la pressione di un bottone.

3.6 Global Control Flow

Il flusso di controllo esterno è gestito dal server che sarà sempre in funzione in attesa delle eventuali richieste di funzionalità da parte dei vari Client, nel caso di molteplici richieste contemporanee il server utilizzerà la tecnica "First in – First Out". Il sistema è caratterizzato da un portale accessibile da browser. Quando un utente si logga e sottomette i propri dati, vi è un accesso al database (query di interrogazione) che permette di controllare l'autenticazione del soggetto. Dopo la conferma, l'utente può accedere a diverse operazioni messe a disposizione dal sistema.

Flussi per USE CASE:

- ❖ **Visualizza_Dettagli** consiste nel mostrare i dettagli di un singolo prodotto o di una classe di prodotti simili nel tipo; poiché tale accesso è in sola lettura non c'è bisogno della gestione della concorrenza (infatti non vi sarà modifica di dati)
- ❖ **AcquistaProdotto** richiede un flusso di operazioni di complessità maggiore. Inizialmente vi è la chiamata allo UseCase Prodotto, in modo da ricercare il prodotto desiderato, prima di confermare l'acquisto si entra in una sezione dove il DBMS lato server gestisce la concorrenza di più utenti, per evitare che vi siano accavallamenti; per poi aggiungere al carrello i relativi prodotti.

3.7 Boundary Condition

Inizializzazione

Il server deve essere sempre acceso altrimenti i client non possono richiedere funzionalità. Il sistema lato server parte nel momento in cui è lanciato il web server dopo di che è sempre attivo 24h su 24 ed una volta attivato non sarà più stoppato; mentre per il sistema lato client è inizializzato ogni volta che un utente accede al portale inserendo l'URL www.D&D.it nel browser.

Terminazione

Il sistema lato server non può terminare (a meno di guasti). Il sistema lato client è terminato alla chiusura del browser ed in quell'istante viene chiusa la sessione e vengono rilasciate le informazioni sulle variabili di sessione temporanee (Prodotto, LoginUtente); in presenza di una chiusura o malf funzionamento del client, il sistema funziona correttamente ma non può accedere alla documentazione presente in quest'ultimo

Fallimento

Il sistema lato server può fallire solo a causa di condizioni eccezionali quali mancanza di elettricità o guasti all'hardware (hard disk danneggiato ecc., che danneggiano permanentemente il sistema) o in caso di crash di sistema (attacchi al server dall'esterno). Il sistema lato client può fallire a causa di guasti temporanei (hardware o software) ma ciò non influisce in alcun modo sul lato server (che resterà comunque stabile), neanche se ci fosse stata una connessione attiva al DB, visto che il DBMS gestisce le transazioni. Per recuperare da un fallimento basta riaccendere il web server, ma ovviamente, nel frattempo, i client non possono usufruire nel sistema visto che il server è down.

Eccezioni

Il web server può cadere e non essere agibile per un po' di tempo. Al sistema lato client apparirà il messaggio "Http 404", ossia che la pagina non è disponibile.