

2019

PharmaÉlite



Sdd - System Design Document

Versione 3.0

Corso di Ingegneria del Software

Università degli Studi di Salerno

**Partecipanti:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Bozzoli Luigi | 0512105477 |
| Martucci Antonio | 0512105612 |
| Squitieri Lucio | 0512105180 |

**Indice**

1. **Introduction**
   1. Purpose of the system
   2. Design goals
   3. Definitions, acronyms, and abbreviations
   4. References
   5. Overview
2. **Proposed software architecture**
   1. Overview
   2. Subsystem decomposition
   3. Hardware/software mapping
   4. Persistent data management
   5. Access control and security
   6. Global software control
   7. Boundary conditions
3. **Subsytem services**

System Design Document

* 1. Introduction
  2. Purpose of the system

Il sistema ha lo scopo di rendere accessibili prodotti farmaceutici ai clienti, anche coloro che non possono recarsi in una farmacia fisica. L’obbiettivo è realizzare un sistema che permetta una semplice visualizzazione dei prodotti e il loro semplice acquisto. Per raggiungere questo scopo i prodotti verranno divisi in quattro categorie: erboristeria, integratori, farmaci da banco e igiene orale. Inoltre, sarà possibile ricercare un prodotto attraverso il suo nome. Il gestore del catalogo potrà facilmente inserire, modificare ed eliminare prodotti mantenendo il catalogo completo e aggiornato.

1.2 Design goals

I design goals identificati sono i seguenti:

CRITERI DI PERFORMANCE

-**Tempo di risposta:** Il tempo di risposta per il sito potrà essere al massimo di un secondo.

-**Memoria:** La dimensione del sistema dipenderà dalla dimensione del database.

CRITERI DI AFFIDABILITA’

**Robustezza:** Gli input non validi immessi dall’utente saranno segnalati attraverso messaggi di errore.

**Affidabilità:** I risultati visualizzati sono attendibili. Quando l’utente visualizzerà un prodotto questo sarà sempre aggiornato.

**Disponibilità:** Una volta realizzato il sistema sarà disponibile 24h/24h e 7 giorni su 7

**Security:** L’accesso al sistema sarà effettuato attraverso un’email e una password. La sicurezza del database è garantita dal fatto che esso è in locale e accessibile solo ai suoi gestori.

CRITERI DI COSTO

**Costi di sviluppo:** Il costo complessivo è stimato a 125 ore per ogni Team member. Per un totale di 375 ore.

CRITERI DI MANUTENZIONE

**Adattabilità:** Il sistema sarà funzionabile per qualsiasi rivenditore farmaceutico.

**Portabilità:** In quanto l’interazione con il sistema avviene attraverso un browser, c’è indipendenza dal sistema operativo e quindi sarà portabile.

CRITERI UTENTI FINALI

**Usabilità:** Il sistema sarà molto semplice da apprendere senza la consultazione di una documentazione in quanto esso sarà prevedibile, cioè la risposta del sistema ad un’azione utente sarà corrispondente alle aspettative.

1.3 Definition, acronyms and abbreviations

**RAD:** Requirements Analysis Document

**SDD:** System Design Document

**DB:** Database

**DBMS:** Database Management System

1.4 References

* <https://www.epharma.com>
* <https://www.lloydsfarmacia.it>

1.5 Overview

In quanto non vi è un sistema già esistente il secondo punto di questo documento (Current software architecture) non sarà presente. Al terzo punto sarà presentata l’architettura del sistema proposto. Qui gestiremo la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi, il controllo del sistema globale e le condizioni limite. Al quarto punto saranno presentati i servizi del sottosistema.

3. Proposed Software Architecture

3.1 Overview

L’architettura del sistema da noi proposto è basata su un’applicazione web in locale, per motivi di sicurezza.

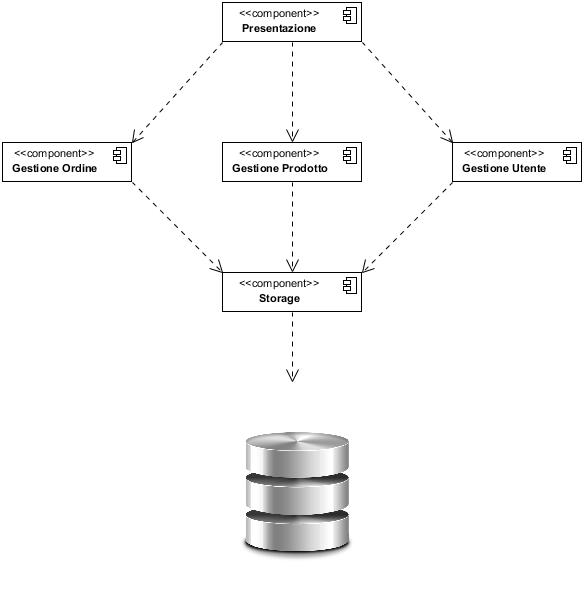
Il nostro server web riceve le richieste dal client e risponde in tempo utile, tale architettura fornisce l’affidabilità del sistema in particolare la robustezza e la sicurezza, facilitandone anche la sua progettazione e successiva implementazione. Le funzionalità saranno divise nei seguenti sottoinsiemi: Presentazione, gestione utente, gestione prodotto, gestione ordine, storage.

3.2 Subsystem decomposition

La decomposizione in sottoinsiemi permetterà di ridurre la complessita del dominio della soluzione.

Per realizzare il sistema è stato utilizzato lo stile architetturale “MVC”. L’architettura MVC indica una particolare architettura software di tipo multi-tier per l’esecuzione di un’applicazione web-based che comporta la suddivisione del sistema in tre moduli dedicati: all’interfaccia utente, alla logica applicativa e alla gestione dei dati persistenti. Descriviamo di seguito i tre strati:

* L’interfaccia View che rappresenta l’interfaccia grafica e in generale i boundary object come form e i link che permettono gli utenti di interagire con il sistema.
* Il control object che ha il compito di eseguire le elaborazioni generando le risposte da inviare verso il client. In molti casi bisogna interrogare il database, per questo si interfaccia con il model
* Il model si occupa della gestione dei dati. Riceve interrogazioni dal control attraverso JDBC e risponderà connettendosi al DBMS ed eseguendo le query.



3.3 Mapping Hardware/Software

Il sistema che si vuole sviluppare sarà installato su un solo computer e utilizzerà un DBMS MySQl installato sullo stesso. Il sistema quindi sarà formato solo dal server che in futuro potrà essere installato su un’altra macchina. Il sistema utilizzerà un DBMS per la registrazione di dati persistenti di nostro interesse.

**Protocollo richiesto:** HTTP

**Memorizzazione dati:** DBMS MySQL

**WebServer:** Apache

**Linguaggi di programmazione utilizzati:** PHP,Html, CSS, Javascript, Ajax

3.4 Gestione dei dati persistenti

Cliente (Email, nome, cognome, sesso, telefono, password, città)

IndirizzoDiSpedizione (Indirizzo, emailCliente)

MetodoDiPagamento (numeroCarta, tipoCarta, emailCliente)

Prodotto (Id, immagine, categoria, nome, prezzo, quantità, descrizione, IdProdottoOrdine)

Ordine (Id, dataOrdine, emailCliente)

ProdottoNell’ordine(Id, nome, idOrdine)

|  |  |  |  |
| --- | --- | --- | --- |
| **ENTITA** | **DESCRIZIONE** | **ATTRIBUTI** | **IDENTIFICATORE** |
| Cliente | Utente che naviga sul sito | Indirizzo e-mail, password, tipo, numero di telefono | Indirizzo E-mail |
| Prodotto | Prodotto presente nel catalogo del sito | Id prodotto, url immagine, categoria, nome, prezzo, quantità, descrizione | Id prodotto |
| Ordine | Acquisto effettuato dal cliente | Id ordine, e-mail cliente, data ordine, costo | Id ordine |
| Indirizzo di spedizione | Stringa che identifica l’indirizzo a cui saranno spediti i prodotti | Indirizzo, e-mail cliente | E-mail cliente |
| Metodo di pagamento | Stringa che identifica il metodo di pagamento utilizzato | Numero carta, tipo carta, e-mail cliente | E-mail cliente |
| Prodotto nell’ordine | Prodotto identificato nell’ordine effetuato | Id prodotto, nome, id ordine | Id ordine |

Il sistema effettua query complesse sui dati. Si utilizzerà MySql come database relazionale per la gestione dei dati persistenti e SQL come linguaggio di programmazione per la creazione di tabelle e l’esecuzione di query.

3.5 Access control and security

L’accesso del cliente e del gestore del catalogo verrà effettuato tramite l’inserimento di email e password. La sicurezza del catalogo è garantita dal fatto che il solo gestore di quest’ultimo potrà effettuare modifiche. Nella tabella sottostante sono riportate le operazioni che sono consentite agli utenti.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oggetti  Attori | Prodotto | Indirizzo di spedizione | Metodo di pagamento | Ordine | Carrello |
| Cliente | Aggiungi al carrello  Rimuovi dal carrello  Modifica quantità nel carrello | Aggiungi indirizzo di spedizione | Aggiungi metodo di  pagamento | Procedi all’ordine | Checkout |
| Gestore del catalogo | Aggiungi al carrello  Rimuovi dal carrello  Modifica quantità nel carrello  Aggungi prodotto al catalogo  Rimuovi prodotto dal catalogo  Modifica prodotto del catalogo | Aggiungi indirizzo di spedizione | Aggiungi metodo di  pagamento | Procedi all’ordine | Checkout |

3.6 Global software control

Il sistema PharmaÉlite fornisce funzionalità che richiedono la continua interazione da parte dell’utente, per questo motivo abbiamo adottato il controllo di flusso event-driven.

3.7 Boundary conditions

-Start-up: Per il primo start-up del sistema è necessario l’avvio di un web server e l’avvio di un database MySQL. Successivamente il gestore del catalogo potrà effettuare l’accesso inserendo le credenziali (e-mail e password) ed inserirà i prodotti all’interno del catalogo.

-Terminazione: Al momento della chiusura del software si ha la terminazione del sistema e verrà assicurata la consistenza dei dati.

-Fallimento: Possono verificarsi fallimenti nei seguenti casi:

* Nel caso in cui si verifichi un’interruzione dell’alimentazione. Non è previsto il ripristino dello stato del sistema prima del fallimento.
* Errore critico dell’hardware non è previsto una misura correttiva.
* Sovraccarico del database, non è previsto il salvataggio periodico dei dati.

4 Subsystem services

**PRESENTATION**

Interfacce che gestiscono l’interfaccia grafica e gli eventi generati dall’interazione dell’utente con il sistema.

**GESTORE PRODOTTO:**

* Aggiungi prodotto al catalogo
* Modifica prodotto al catalogo
* Elimina prodotto al catalogo
* Aggiungi prodotto al carrello
* Modifica prodotto al carrello
* Elimina prodotto al carrello

**GESTIONE UTENTE:**

* Login
* Registrazione
* Logout

**GESTIONE ORDINE**

* Checkout
* Procedi all'ordine
* Visualizza carrello

**STORAGE**

* Salvare dati
* Aggiornare dati
* Eliminare dati