

Dipartimento di Ingegneria e Scienze dell’Informazione e Matematica

CORSO DI LAUREA IN INFORMATICA

Insegnamento Laboratorio di programmazione ad oggetti

PHARMAPP

|  |  |  |  |
| --- | --- | --- | --- |
| **Membri del team** | | | |
| **Cognome e nome** | **Matricola** | | **Indirizzo e-mail** |
| **Paolilli Giuseppe** | 242096 | | giuseppe.paolilli@student.univaq.it |
| **Corsi Antonio** | 236978 | | antonio.corsi@student.univaq.it |
| **Di Laudo Luca** | 248444 | | luca.dilaudo@student.univaq.it |
|  | |  | | |
|  | | | | |
|  | |  | | |

A.A. 2019/2020

Sommario

[Capitolo 1 – Caso di studio 2](#_Toc38393257)

[1.1 – Descrizione dell’applicazione 2](#_Toc38393258)

[1.2 - Funzionalità 3](#_Toc38393259)

[1.3 Domain Model 4](#_Toc38393260)

# Capitolo 1 – Caso di studio

## 1.1 – Descrizione dell’applicazione

Si vuole realizzare un software che permette la gestione di farmacie con diverse tipologie di utenti che vanno dall’amministratore al cliente. PHARMAPP è la piattaforma ideale per la gestione di tutto il lato tecnico-amministrativo dell’attività. Sviluppata in linguaggio Java, questa piattaforma si ispira al modo di gestire il lato back-end di Laravel, dall'esperienza passata con questo si è optato per un sistema di migrations e di seeder per il lavoro sul database,a questa logica abbiamo affiancato un da osche svolge operazioni base crud. PHARMAPP durante l’accesso distingue gli utenti in base al proprio ruolo, permettendo quindi una grande quantità di utilizzi, a partire dalla semplice visione di una ricetta prescritta dal proprio medico, fino alla gestione del tuo magazzino interno. Nella scrittura della piattaforma una delle priorità è stata rendere PHARMAPP user friendly fin dal primo utilizzo, interfacce semplici ed immediate portano l’utilizzatore ad accedere ai servizi richiesti in un solo click. Per come è stata implementata l’architettura, PHARMAPP non consente però un utilizzo serverless poiché deve gestire in ogni momento molte funzioni che difficilmente implementabili senza l’appoggio di un Database. Uno dei punti di forza dell’applicazione è proprio la creazione del Database sopracitato, il sistema infatti ha integrato tutto il necessario per la popolazione del database della tua farmacia, cosicché non sia necessario,in maniera preventiva al primo utilizzo la popolazione di una base di dati . L’unico prerequisito è quello di creare, in anticipo al primo avvio, una table completamente vuota nel proprio software (Nel caso di studio da noi effettuato MySQL) per poi inserire i propri dati di connessione nel file di testo di configurazione presente nel pacchetto del programma. Il risultato dello studio è la creazione di una piattaforma che concilia user friendliness, velocità di esecuzione e un buon utilizzo di risorse

## 1.2 – Funzionalità

Le funzionalità della piattaforma PHARMAPP vengono di seguito elencate in base al ruolo dell’utente nel sistema:

|  |  |  |
| --- | --- | --- |
| Ruolo dell’utente | Funzionalità | Descrizione |
| Paziente | Login | L’utente accede alla piattaforma utilizzando i propri dati anagrafici e le credenziali fornite dall’amministratore |
| Paziente | Storico delle prescrizioni | Effettuato l’accesso il paziente può visionare lo storico completo delle proprie prescrizioni, siano esse già evase o meno |
| Paziente | Visualizzazione bugiardini | Effettuato l’accesso il paziente ha la possibilità di visualizzare l’intera lista di farmaci presente nella farmacia, dandogli la possibilità di visualizzare i bugiardini di un farmaco |

|  |  |  |
| --- | --- | --- |
| Ruolo dell’utente | Funzionalità | Descrizione |
| Medico | Login | L’utente accede alla piattaforma utilizzando i propri dati anagrafici e le credenziali fornite dall’amministratore |
| Medico | Creare prescrizioni | Effettuato l’accesso il medico può creare una prescrizione al proprio paziente,questa verrà aggiunta allo storico delle prescrizioni del paziente |
| Medico | Modificare prescrizioni | Effettuato l’accesso il medico può modificare una prescrizione già inserita nello storico qualora essa non sia già stata evasa |
| Medico | Eliminare prescrizioni | Effettuato l’accesso il medico può cancellare una prescrizione precedentemente inserita nello storico qualora essa ancora non sia stata evaso |

|  |  |  |
| --- | --- | --- |
| Ruolo dell’utente | Funzionalità | Descrizione |
| Farmacista | Login | L’utente accede alla piattaforma utilizzando i propri dati anagrafici e le credenziali fornite dall’amministratore |
| Farmacista | Evasione delle prescrizioni | Effettuato l’accesso il farmacista può, simulando la vendita dei farmaci prescritti, cambiare lo stato di una prescrizione presente nello storico di un paziente in “evasa” |

|  |  |  |
| --- | --- | --- |
| Ruolo dell’utente | Funzionalità | Descrizione |
| Amministratore | Login | L’utente accede alla piattaforma utilizzando i propri dati anagrafici e le credenziali fornite dall’amministratore |
| Amministratore | Gestione Magazzino | Effettuato l’accesso l’amministratore ha il completo controllo, sul magazzino della farmacia, potendo ordinare farmaci qualora siano in esaurimento, aggiungere nuovi farmaci o rimuovere farmaci non più ordinabili |

Login: identificato il ruolo dell’utente che effettua l’accesso il DAO restituisce le viste in base al ruolo

Effettuato il riconoscimento si passa alle funzioni per ruolo:

Paziente:

vista delle prescrizioni: il DAO restituisce all’utente le prescrizioni legate al proprio ID

vista dei farmaci: il DAO restituisce al paziente una lista di tutti i farmaci presente nel magazzino

Dottore:

Creare/cancellare/modificare prescrizioni: il controller della funzione in questione (crea,update o home ) passa al DAO l’ordine di modifica della base di dati

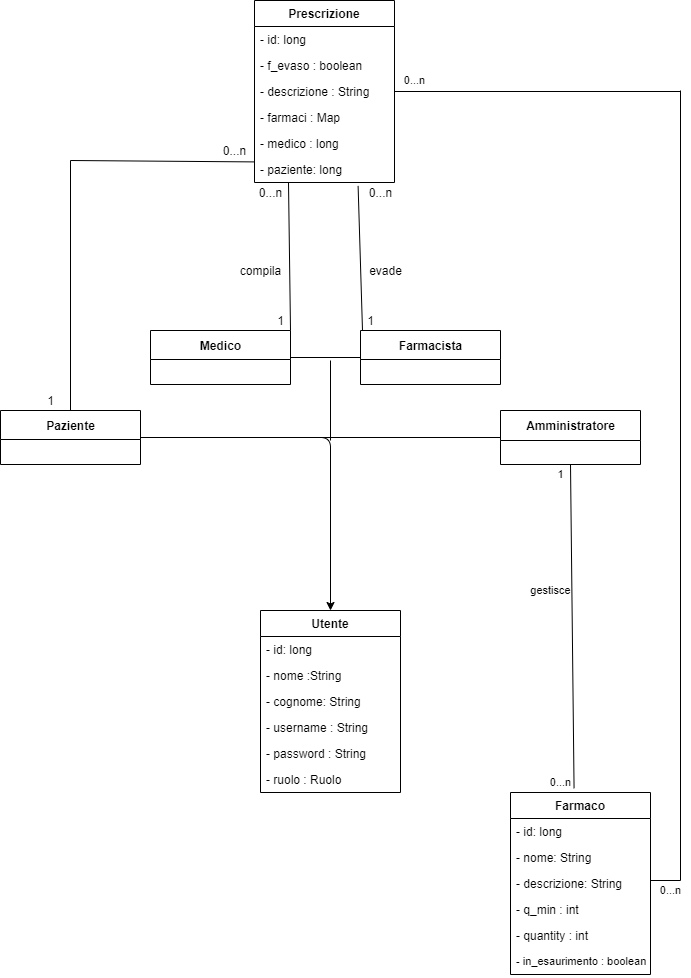
Medico:

evasione prescrizioni: tramite l’apposita view il controller evasioneprescrizione chiama il DAO al fine di effettuare la funzione di update sul database

Ammistratore:

Gestione farmaci: con una logica simile a quella del dottore con le prescrizioni l’amministratore può effettuare operazioni di creazione,update e cancellazione di farmaci in magazzino

## 1.3 – Domain Model



## 1.4: Entity–relationship model

