

# 1. Introduzione

## 1.1 Scopo del sistema

L'obiettivo del progetto è quello di realizzare una piattaforma e-commerce dedita alla vendita di prodotti dedicati alla salute, che consenta quindi ai clienti di poter acquistare i farmaci e gli altri prodotti di cui necessitano direttamente dalla propria abitazione, consentendo anche ai gestori della piattaforma di gestirla facilmente.

La piattaforma deve permettere ai clienti di poter scegliere il prodotto più adatto alle loro esigenze anche tramite la visione di recensioni lasciate da altri clienti che devono però essere moderate al fine di evitarne un utilizzo scorretto.

## 1.2 Design Goals

Il sistema NetPharma deve avere un alto livello di usabilità garantendo quindi il corretto utilizzo della piattaforma anche da utenti meno esperti possedendo quindi interfacce grafiche intuitive. Inoltre trattandosi di un sistema di e-commerce deve garantire alti standard di affidabilità attuando dei sistemi che prevengono il malfunzionamento del sistema in caso di errori.

Infine devono essere considerati anche requisiti di performance e manutenibilità dovendo la piattaforma essere in grado di gestire correttamente un discreto numero di accessi simultanei e avere la possibilità di poter essere mantenuta osservando regole di buona programmazione.

Le classi da considerare sono quindi : Usabilità,Affidabilità,Performance e Manutenzione

### 1.2.1 – Criteri di usabilità

<b>GRAFICA</b>	NetParma deve rendere le operazioni accessibili a qualsiasi categoria di utente indipendentemente dalle loro abilità informatiche tramite l'utilizzo di una interfaccia grafica minimale e di facile comprensione, nello specifico verranno utilizzati fogli di stile css ed il framework BootStrap, la grafica ha priorità più elevata rispetto a tutto il resto, anche a discapito di performance o manutenibilità. Priorità : Alta
<b>ICONE</b>	Le opportune funzioni dell'interfaccia devono essere accompagnate dalle icone standard ad esse associate come quella del carrello,del log-in e del log-out. Le icone verranno implementate utilizzando BootStrap. Priorità : Alta
<b>EVIDENZIARE GLI ERRORI</b>	Il sistema deve riconoscere ed evidenziare gli errori di compilazione di un form con una apposita colorazione che consenta all'utente di capire la provenienza di un determinato errore, il sistema deve inoltre fornire anche

	suggerimenti utili a correggerlo utilizzando JavaScript. Priorità : Alta
<b>RESPONSIVENESS</b>	L'interfaccia grafica del sistema deve adattarsi alle grandezze dei principali dispositivi in commercio quali: Computer, Tablet e Smartphone. La responsiveness verrà implementata tramite framework Bootstrap. Priorità : Alta

#### 1.2.2 – Criteri di affidabilità

<b>FAULT AVOIDANCE</b>	Il sistema deve essere in grado di scartare eventuali dati errati inseriti, non memorizzarli, non andare in crash a seguito del loro inserimento e di rispondere in maniera adeguata. Questi controlli verranno implementati attraverso l'utilizzo di JavaScript e JQuery. Priorità : Alta
<b>SICUREZZA DEI DATI</b>	Le credenziali di accesso devono essere criptate all'interno del database e di queste non deve essere possibile il recupero ma soltanto la sostituzione o il reset. Questi dati sensibili saranno criptati in MD5 direttamente all'interno del database e manipolati nel codice utilizzando le apposite funzioni di libreria. La sicurezza dei dati ha priorità sulle performance. Priorità : Alta

#### 1.2.3 – Criteri di performance

<b>TEMPO DI RISPOSTA</b>	Il sistema deve essere reattivo per tutte le operazioni che riguardano l'acquisto e avere comunque sempre un tempo di risposta inferiore a 3 secondi indipendentemente dalla quantità di spazio occupata all'interno del database.
--------------------------	--

	<p>Verranno quindi ridotti al minimo i cicli e si preferiranno algoritmi con complessità asintotica lineare o inferiore dove possibile.</p> <p>Si provvederà anche ad utilizzare strutture dati opportune per minimizzare gli sprechi di memoria.</p> <p>Priorità : Media</p>
<b>THROUGHPUT</b>	<p>Il sistema deve essere in grado di gestire picchi di carico fino ad un massimo di 300 utenti connessi in simultanea che effettuano operazioni senza mostrare rallentamenti. Garantendo quindi in condizioni di normalità determinati tempi di latenza e una sensazione di fluidità.</p> <p>Verrà quindi utilizzato un ConnectionPool efficiente che sfrutterà la programmazione multithread.</p> <p>Priorità : Media</p>

#### 1.2.4 – Criteri di manutenzione

<b>DESCRIZIONE DELLA LOGICA</b>	<p>Il codice della piattaforma deve avere dei commenti utili alla comprensione delle principali funzionalità.</p> <p>I commenti verranno implementati utilizzando JavaDoc.</p> <p>Priorità : Media</p>
<b>IDENTIFICAZIONE</b>	<p>Le variabili utilizzate all'interno del codice devono avere nomi univoci e facilmente identificabili al fine di identificare facilmente il loro scopo</p> <p>Priorità : Alta</p>

#### 1.3 Acronimi e abbreviazioni

**FAULT** : Qualsiasi errore o malfunzionamento critico che nel caso non fosse adeguatamente gestito porterebbe all'arresto del sistema.

**RESPONSIVENESS** : Capacità di una pagina web di adattarsi al tipo di dispositivo in uso.

**E-COMMERCE** : Negozio online accessibile tramite un computer o qualsiasi altro dispositivo simile.

**RAD** : Documento di analisi dei requisiti .

**THROUGHPUT** : In termini generali indica la potenza effettiva di un canale di comunicazione o di un software applicativo.

#### 1.4 Riferimenti

- *Testo Object-Oriented Software Engineering Using UML, Patterns, and Java*
- *RAD (NetPharma)*
- *Problem statement (NetPharma)*

#### 1.5 Panoramica

Il documento si divide in cinque parti:

- Nella prima parte sono introdotti gli obiettivi di design e forniti i riferimenti utilizzati per la creazione
- Nella sezione successiva sono invece illustrate le caratteristiche del sistema corrente.
- La terza parte invece rappresenta il software proposto e vengono descritti i seguenti elementi :
  - **DECOMPOSIZIONE IN SOTTOSISTEMI** : Il sistema viene suddiviso in diversi sottosistemi dove ogni sottosistema è caratterizzato dai servizi che offre agli altri sottosistemi
  - **MAPPING HARDWARE/SOFTWARE** : Si decide l'hardware dove il sistema deve funzionare e si mappano le componenti su di essa
  - **GESTIONE DEI DATI PERSISTENTI** : Descrive i dati memorizzati dal sistema e il database usato per memorizzarli
  - **CONTROLLO DEGLI ACCESSI E SICUREZZA**: Descrive la matrice degli accessi
  - **CONTROLLO GLOBALE DEL SOFTWARE** : Descrive come si sincronizzano i sottosistemi
  - **CONDIZIONI DI BOUNDARY** : Descritte le funzionalità del gestore del sistema come start-up e shutdown
- La quarta parte è per i servizi dei sottosistemi che descrive in termini di operazioni i servizi forniti da singoli sottosistemi
- L'ultima parte del documento è il glossario che si occupa di illustrare il significato di alcuni termini utilizzati.